

Thèse de Doctorat de L'université Paris VI

Spécialité: INFORMATIQUE

Présentée pour obtenir le grade de Docteur de l'Université Paris VI

Par

ALEXIS DROGOUL

De La Simulation Multi-Agent A La
Résolution Collective de Problèmes

Une Étude De l'Émergence De Structures D'Organisation Dans Les
Systèmes Multi-Agents

Soutenue le **23 Novembre 1993**
devant le jury composé de:

MM.	LUC STEELS , Professeur (VUB), Bruxelles	Rapporteur
	ALAIN LENOIR , Professeur (Université Paris XIII)	Rapporteur
	YVES DEMAZEAU , Chargé de Recherches (CNRS), Grenoble	Rapporteur
	JEAN-FRANÇOIS PERROT , Professeur (Université Paris VI)	Examineur
	ALAIN PAVE , Professeur (Université Lyon I)	Examineur
	JACQUES FERBER , Professeur (Université Paris VI)	Directeur

De la Simulation Multi-Agents à la Résolution Collective de Problèmes: une Etude de l'Emergence de Structures d'Organisation dans les Systèmes Multi-Agents

Thèse de Doctorat de l'Université Paris VI présentée par Alexis DROGOULE le 23 Novembre 1993
LAFORIA, I.B.P., Université Paris VI, Boîte 169, 4 Place Jussieu, 75232 Paris Cedex 05

Résumé

La problématique générale qui oriente notre travail concerne la capacité de groupes d'agents à s'auto-organiser pour produire des réponses fonctionnelles collectives. Elle s'insère dans la problématique plus vaste de l'Intelligence Artificielle Distribuée, qui est de faire fonctionner de manière coopérative des agents informatiques au sein de *systèmes multi-agents*. Nous nous intéressons plus particulièrement à l'émergence de ces réponses au sein de groupes d'agents non-intelligents individuellement, encore appelés *agents réactifs*. Ce choix nécessite de comprendre les comportements individuels et inter-individuels qui permettent l'apparition et le maintien d'une dynamique collective alors même qu'aucun agent n'est capable de *penser* cette dynamique.

Comme les travaux théoriques portant sur l'auto-organisation ou l'émergence de propriétés sont encore loin d'être finalisés, le point de vue que nous défendons ici est qu'il est possible de s'inspirer de processus auto-organisés existants pour construire des logiciels exploitant ces propriétés. Pour cela, deux types d'études ont été réalisées:

- Une étude de simulation s'attachant à reproduire le fonctionnement d'une colonie de fourmis en utilisant un système réactif, où chaque fourmi est représentée dans le système par un agent possédant un comportement du type stimulus/réponse. Nous reproduisons fidèlement, avec ce système totalement décentralisé, la dynamique de fondation d'une société et la structuration de l'ensemble des agents-fourmis au sein d'une division globale du travail. Outre que ces résultats nous fournissent l'occasion de montrer le large champ d'application d'une approche multi-agents de la simulation, ils peuvent être interprétés comme la réponse fonctionnelle et auto-organisée d'un groupe d'agents aux problèmes que lui pose son environnement.
- La seconde étude développe dans le domaine de la résolution de problèmes cette capacité auto-organisatrice, en s'attachant tout d'abord à extraire de la simulation réalisée les paradigmes qui nous fournissent la base d'un système de Résolution Collective de Problèmes. Nous montrons ainsi comment concevoir des systèmes réactifs capables de résoudre des problèmes comme le Tri Collectif ou la recherche d'échantillons dans un environnement aléatoire, en s'inspirant directement des comportements des agents-fourmis de la simulation. Nous montrons ensuite qu'une approche réactive peut être appliquée à des problèmes classiques en Intelligence Artificielle, comme Pengi et le Taquin, en offrant de bien meilleurs résultats que toutes les approches précédemment employées. Nous en montrons cependant aussi les limites potentielles, notamment dans le cadre du Jeu d'Echecs.

Remerciements

Je tiens à remercier Luc Steels de m'avoir fait l'honneur de rapporter sur ce travail. Je lui suis redevable, par publications interposées, de bien des idées développées ici, ainsi que de la fascination pour les systèmes à fonctionnalité émergente qui ne m'a pas quitté depuis la lecture du premier article qu'il leur a consacré.

Je tiens également à remercier Alain Lenoir de rapporter sur ce travail d'informaticien, au milieu duquel, je l'espère, il a pu retrouver un peu de l'esprit éthologique, esprit observateur, vif et intègre, après lequel je cours depuis bientôt trois ans.

Je remercie Yves Demazeau d'avoir pu m'accorder une partie de son temps pour effectuer un rapport supplémentaire sur cette thèse. Je sais combien son temps est précieux, et je lui exprime toute ma gratitude et toute mon amitié.

Jean-François Perrot a su, en Licence, me faire découvrir que la programmation pouvait être esthétique et être abordée comme une œuvre d'art. Si je suis encore enchaîné à un ordinateur aujourd'hui, c'est sans doute en partie de sa faute. Qu'il ait accepté de faire partie de mon jury m'a profondément fait plaisir et je tiens ici à l'en remercier.

Je remercie Alain Pavé d'avoir accepté de faire partie de mon jury, et j'espère que les écosystèmes abordés ici ne lui paraîtront pas trop artificiels au regard de ceux qu'il a l'habitude d'étudier.

Jacques Ferber a été plus qu'un directeur de thèse pour moi. Son amitié et son soutien pendant ces trois années m'ont apporté plus que je ne l'aurais imaginé en abordant ce travail. La grande liberté dont j'ai bénéficié n'avait pas de prix, mais j'espère néanmoins que cet ouvrage comblera une partie de mon immense dette à son égard.

Le *nous* employé tout au long de la thèse n'est pas qu'académique. Il englobe dans son anonymat les personnes avec lesquels j'ai pu travailler pour réaliser les travaux présentés. Il serait sans doute trop long d'en faire la liste exhaustive, mais je tiens à remercier Bruno Corbara et Dominique Fresneau, qui m'ont fait découvrir le monde fascinant des insectes sociaux. Je tiens à leur exprimer ici toute mon amitié et toute mon admiration. Je tiens également à remercier Christophe Dubreuil, qui a initié la plupart des travaux de résolution de problèmes présentés ici. Les moments magiques que nous avons partagés lors de la conception du Taquin en Eco-Résolution resteront sans doute les meilleurs souvenirs de ces trois années. Enfin, j'ai pu compter sur l'apport de quatre étudiants du DEA IARFA, Steffen Lalande, Alain Touret, Nicolas Czternasty et Philippe Louise, qui ont contribué à l'avancée du projet MANTA et sans les idées desquels de nombreux résultats n'auraient pas vu le jour. Comme nous le verrons par la suite, un agent n'est plus grand-chose sans son environnement social. La rédaction de cette thèse, qui m'a coupé de l'équipe MIRIAD pendant de longs mois, m'a fait comprendre à quel point je vivais en symbiose (d'idées) avec ses membres. Qu'ils sachent que j'ai passé avec eux des moments inoubliables, et que les multiples échanges d'idées, interactions et confrontations que j'ai pu avoir avec tous m'ont fait comprendre la force de la notion d'*intelligence collective*.

Je tiens à remercier toute ma famille pour l'affection et la patience qu'elle m'a accordé depuis le début de mes études, et en particulier mes parents, qui m'ont toujours fait confiance même si je leur semblais quelquefois sorti tout droit d'un asile de fous.

Je tiens à faire savoir à tous mes amis combien précieuse a été leur présence et à quel point je leur sais gré de ne pas s'être trop plaint de l'être asocial que je suis devenu lors de la rédaction de cette thèse.

Enfin, j'exprime ici tout mon amour à Hélène qui a su, pendant ces trois ans, supporter avec tendresse et abnégation l'*homo informaticus* vivant à ses côtés. Je lui dédie cet ouvrage, ainsi qu'à Antoine, dont j'imagine à l'avance le rire lorsqu'il en découvrira le manuscrit, dans vingt ans, au fond d'une vieille malle poussiéreuse...

TABLE DES MATIERES

Avant-Propos en Forme d'Hommage a Douglas Hofstadter	vi
Introduction Présentation et (Auto-?) Organisation de la Thèse	xv
Première Partie: IAD réactive et Simulation	1
Chapitre I Introduction a la Problématique	2
I.1. L'Intelligence Artificielle Distribuée	2
I.2. Des Principes d'une Approche Réactive	5
I.3. Les Organisations Réactives	7
I.4. La Nécessité d'une "Démarche constructiviste"	11
Chapitre II La Simulation Multi-Agents	13
II.1. La Simulation	13
II.2. La Simulation Multi-Agents	20
Chapitre III l'EthoModélisation	23
III.1. Introduction.....	23
III.2. Principes Généraux du système de simulation	25
III.3. La programmation du Modèle Générique.....	30
III.4. Méthodologie de programmation d'une simulation	48
III.5. Discussion.....	50
Deuxième Partie: Projet Manta	55
Chapitre IV Simuler des Fourmis	57
IV.1. Des Fourmis à Manta.....	57
IV.2. Les Agents de Manta.....	65
IV.3. Programmation du Comportement des Agents	70
Chapitre V Expérimentations	76
V.1. Protocoles d'expérimentation.....	76
V.2. Sociogenèse	79
V.3. Auto-régulation de la population	88
V.4. Organisation Sociale de la Colonie.....	92
V.5. La Question du Polyéthisme d'Age.....	101
V.6. La question de la polygynie.....	106
V.7. Competition Spatiale entre Reines.....	108
Conclusion de la Deuxième Partie	112
Troisième Partie: Résolution Collective de Problèmes	115
Chapitre VI Résolution Collective de Problèmes	117
VI.1. Résolution Distribuée de Problèmes	117
VI.2. Résolution Collective de Problèmes	119
Chapitre VII Un Exemple de Résolution Collective de Problèmes: le Tri Collectif	125
VII.1. Présentation du problème.....	125
VII.2. Programmation des agents.....	126
VII.3. Expérimentations	127
VII.4. Conclusion.....	130
Chapitre VIII Les robots fourrageurs	131
VIII.1. Introduction	131

VIII.2. Présentation du problème.....	132
VIII.3. Programmation des premiers robots	133
VIII.4. La première génération de Petit Poucet.....	135
VIII.5. La seconde génération de Petit Poucet	139
VIII.6. La troisième génération de Petit Poucet.....	142
VIII.7. Les robots Dockers.....	146
VIII.8. Méthodologie de programmation	155
Conclusion de la Troisième Partie.....	159
Quatrième Partie: EcoRésolution	161
Chapitre IX L'éco-résolution.....	162
IX.1. Présentation	162
IX.2. Les éco-étho-agents.....	163
Chapitre X Pengi: Modélisation d'un Ecosystème Abstrait	165
X.1. Présentation	165
X.2. Le Choix de Pengi	165
X.3. La Programmation du Système.....	167
X.4. L'Emergence de Comportements	170
X.5. Conclusion	175
Chapitre XI Un Problème Classique: le Taquin	176
XI.1. Présentation du problème	176
XI.2. Programmation du problème.....	177
XI.3. Illustration.....	188
XI.4. Complétude et Complexité de la Résolution	190
XI.5. Résultats expérimentaux.....	194
XI.6. Conclusion	196
Conclusion de la Quatrième Partie: Quelles Limites ? Illustration par le Jeu d'Echecs	198
Conclusion Générale	203
Ce qui a été réalisé.....	204
... Ce qui n'a pas été réalisé... ..	204
... Ce qui devra être réalisé un jour.....	205
... Et Ce qui ne sera pas réalisé avant longtemps.....	205
Bibliographie.....	207

Avant-Propos

en Forme d'Homage a Douglas Hofstadter

(...) les nouveaux concepts sont tout d'abord présentés sous forme de métaphores dans un dialogue, afin de susciter un certain nombre d'images visuelles concrètes qui servent de base intuitive lors de la lecture du chapitre suivant, lequel présente le même concept sous une forme plus sérieuse et abstraite.

Douglas Hofstadter, *Gödel, Escher, Bach*, 1979
(Édition Française: InterEditions, Paris, 1985, p. 31)

(Achille est assis dans son salon lorsque la Tortue frappe à sa porte. Il se lève et vient lui ouvrir)

Achille: Ah! Bonjour Madame Tortue, comment allez-vous aujourd'hui ?

La Tortue: Très bien, Achille. Très bien. *(elle lui sourit)* N'était-ce pas votre anniversaire, récemment ?

Achille: *(qui rougit)* Oui. C'était avant-hier.

La Tortue: Eh bien, j'ai deux cadeaux pour vous. Tenez. *(elle lui tend deux paquets enveloppés)*. Allez-y, ouvrez-les.

Achille: Oh, merci! Merci beaucoup! *(il ouvre le premier paquet)* Un jeu d'échecs! C'est extraordinaire, je viens de perdre le mien il y a à peine une semaine. *(il pose le jeu sur la table basse)*. Je me demande d'ailleurs comment j'ai pu perdre un jeu d'échecs. Bah! N'y pensons plus, à présent j'en ai un tout neuf! Et le second cadeau ? Voyons. Un livre ? "De la Synthéthologie à l'Intellimergence" *(légèrement perplexe)* Eh bien... Ca m'a l'air intéressant.

La Tortue: Tout à fait. Je viens de le terminer et ça m'a beaucoup plu. C'est pourquoi je vous l'offre.

Achille: En fait, ça tombe bien! Je viens justement moi aussi de terminer le dernier livre que vous m'aviez conseillé, voyons *(il cherche au fond de son coffre)* Non, ce n'est pas la "Société de l'Esprit", non plus "Sciences des Systèmes, Sciences de l'Artificiel", voyons, où ai-je pu le mettre ? C'est incroyable! A croire que les objets s'amuse à disparaître dès que j'ai le dos tourné. Ah, le voici: "Gödel, Escher, Bach", vous savez, celui de monsieur Hofstadter.

La Tortue: Il vous a plu ?

Achille: Eh bien, je n'ai pas tout compris, mais dans l'ensemble, oui, il m'a plu. Certaines parties sont tout de même difficiles et j'avoue que je n'ai pas tout lu.

La Tortue: Mais alors, Achille, comment le livre a-t-il pu vous plaire si vous ne l'avez pas lu entièrement ?

Achille: En vérité, je l'ai lu entièrement... mais sans tout comprendre. Mais ceci ne m'a pas empêché d'arriver jusqu'à la fin, je vous rassure. Et je pense avoir compris le propos de votre ami, du moins dans ses grandes lignes.

La Tortue: Oui, cela est sans doute dû à la redondance de nombreux chapitres.

Achille: Comment dites-vous ? La redondance ?

La Tortue: Oui, la redondance. C'est une technique littéraire qui consiste à répéter de nombreuses fois la même pensée ou la même idée ou le même concept, mais sous une forme très légèrement différente, de manière, le plus souvent, à bien se faire comprendre.

Achille: Mais quel est l'intérêt de répéter plusieurs fois la même chose ?

La Tortue: Je vous l'ai dit, pour bien se faire comprendre. C'est normalement un défaut, mais ça peut être très utile. Prenons votre cas, par exemple. Vous lisez un livre dont certains chapitres sont difficiles à comprendre, non pas parce que vous n'êtes pas capable de comprendre les idées qu'ils contiennent, mais parce que c'est la manière dont elles sont présentées qui vous rebute.

Achille: C'est vrai, j'avoue que les chapitres sur la physique atomique ou l'ADN...

La Tortue: Bien. Or, l'intérêt de l'auteur, je dirais même le devoir de l'auteur, est de vous faire comprendre ces idées. Sans quoi ce n'est pas la peine d'écrire un livre. Il va donc lui falloir trouver un moyen d'exprimer le même contenu dans des formalismes qui puissent s'adapter aux différents lecteurs qu'il aura.

Achille: Je vois où vous voulez en venir. En présentant la même idée sous des formes différentes, le lecteur a plus de chances de la comprendre...

La Tortue: .. et peut éventuellement sauter quelques chapitres, comme ce que vous avez fait. Si vous ne sautez pas tous les chapitres traitant d'une même idée, vous avez ainsi une bonne chance de comprendre l'ensemble de l'ouvrage. C'est à ça que sert la redondance.

Achille: Ah! Tout de même, l'esprit humain est bien fait! Aucun animal n'aurait pu inventer ce mot extraordinaire: la redondance!

La Tortue: Certes, de même que tous les autres mots, car les animaux ne parlent pas, je vous le rappelle. Mais la redondance existe partout dans la nature.

Achille: Comment cela ? Vous venez de me dire que c'était une technique littéraire ! Si les animaux ne parlent pas, je ne vois pas comment ils pourraient écrire !

La Tortue: C'est effectivement une technique littéraire, mais le mot ne s'applique pas qu'à ça. On parle également de redondance pour qualifier des situations où un même travail est assuré de manière identique par plusieurs individus.

Achille: Et quel est le rapport avec les animaux ?

La Tortue: Eh bien, le rapport est que l'on trouve de la redondance chez de nombreuses espèces d'animaux.

Achille: Que voulez-vous dire ? Que tous les animaux d'une même espèce font tous la même chose ? Oui, ça me semble logique et même, à la réflexion, évident. Toutes les tortues... euh, tous les vers de terre, par exemple, doivent effectivement être identiques et faire à peu près la même chose. C'est cela que l'on appelle la redondance ?

La Tortue: Non, je me suis mal exprimée et vous m'avez mal comprise ... (*elle tousse*)... Auriez-vous la gentillesse de me faire une petite camomille pendant que nous poursuivons cette conversation ?

Achille: Bien sûr, avec plaisir. Juste un instant, le temps que je retrouve ma bouilloire. (*Il cherche dans son placard*) Pourtant, je suis persuadé de l'avoir rangée ici la dernière fois que vous êtes venue. (*il cherche de nouveau sous son évier*) Ce n'est pas grave, je vais chercher celle que le Fourmilier m'a offerte récemment. (*il revient avec la bouilloire*) Vous allez l'étreindre! (*il allume le feu, et pose la bouilloire dessus.*) Bien, nous pouvons continuer.

La Tortue: Où en étais-je ? Oui. La redondance va habituellement de pair avec la spécialisation. Prenons le cas des fourmis, que vous connaissez bien, si je ne me trompe*.

Achille: Ah non! Pas les fourmis! Prenez n'importe quoi d'autre, mais pas les fourmis.

La Tortue: Pourquoi donc ?

Achille: Mais parce que ma maison en est remplie! Elles se sont installées l'hiver dernier dans le mur du fond. Et depuis, j'ai tout essayé, la ruse, l'intimidation, la menace, la force, pour les déloger. Rien à faire. Elles prospèrent dans mes murs, et je ne sais pas comment m'en débarrasser.

La Tortue: Mais, Achille, tout le monde a des fourmis chez soi. Moi-même, dans ma carapace, il m'arrive d'héberger à la mauvaise saison la colonie qui habituellement réside dans mon jardin. Elles craignent le froid, vous savez ?

* Voir [Hofstadter 1985], p.347 à 376

Achille: (qui grommelle) Oui, mais quand elles auront fait s'effondrer les murs à force de creuser des trous dedans, elles ne seront plus seules à craindre le froid...

La Tortue: Allons, ne soyez pas rancunier! Je suis persuadée qu'elles vous rendent énormément de services. *(elle parcourt des yeux le salon)* Si je ne m'abuse, vous n'avez jamais été une fée du logis, n'est-ce pas ? Et pourtant, j'ai beau chercher des yeux, plus une miette, plus une poussière...

Achille: (qui regarde également le salon) Mais c'est vrai ce que vous dites-là! Voyons... Cela fait presque trois mois que je n'ai pas touché à mon balai et, effectivement, je ne me suis jamais posé la question. Vous croyez que ce sont...

La Tortue:... les fourmis ? Oui, bien sûr. Elles sont très efficaces de ce point de vue là. Une colonie est capable de ramasser plusieurs fois son poids en une journée de labeur. Très efficaces...

Achille: ... et très intelligentes! Regardez! Elles sont même allées faire le ménage dans des recoins que mon balai n'arrivait pas à atteindre!

La Tortue: Hum! *(elle toussote de nouveau discrètement)*. Achille, je crois que l'eau bout...

Achille: Oui, votre camomille! J'allais l'oublier! *(il prend la bouilloire et la pose sur la table)* Bien, maintenant, les tasses. *(il ouvre son placard et en sort deux tasses. Mais en attrapant le sucre, une des tasses tombe à terre et se brise)*. Oh! Mon service tout neuf! C'est le Crabe qui me l'avait offert!

La Tortue: J'en suis vraiment désolée. Voulez-vous un peu d'aide ?

Achille: Non, merci. Après tout, j'ai quatre autres tasses. Et je laisserai les fourmis faire le ménage. *(il revient avec ses tasses, verse la camomille et se rassoit)*. Alors, où en étions-nous ?

La Tortue: Je vous parlais de redondance et de spécialisation. Et nous avons pris l'exemple des fourmis.

Achille: Oui. Voilà. Alors, les fourmis ?

La Tortue: Eh bien, si vous vous souvenez de notre dialogue avec le Fourmilier, vous vous rappellerez sans doute que les fourmis sont organisées en castes, ou en groupes de travail qui sont chacun spécialisé dans une tâche donnée au sein de la fourmilière. Et vous vous rappellerez également qu'une fourmi prise individuellement peut être considérée comme un être très simple, disons un individu réactif.

Achille: Réactif ? Elles votent à droite ? Mais j'avais plutôt compris que le fonctionnement d'une fourmilière était communiste...*

La Tortue: Non, pas réactionnaire. Réactif. C'est le propre d'un individu qui réagit mécaniquement aux stimulations de son environnement, sans intentionnalité particulière. Dans un sens il ne fait que suivre un programme qui a été fixé par les millions d'années d'évolution de ses ancêtres. Je reprends. Pour que la colonie survive, il faut qu'elle soit bien adaptée à son environnement, c'est-à-dire que toutes les tâches à effectuer le soient de manière assez efficace.

Achille: J'entends bien. Mais quel est le rapport entre l'adaptation, la redondance, et la spécialisation ?

La Tortue: J'y viens justement. Pour que l'organisation de la colonie soit efficace, il faut d'abord que certaines fourmis soient capables de réagir plus rapidement que d'autres à certaines stimulations venant de l'extérieur...

Achille: Des gardiennes ?

La Tortue: Si vous voulez, mais ce n'est pas seulement ça. Vous avez également des individus réagissant aux besoins des œufs, des larves, des cocons, à la présence ou à l'absence de nourriture, aux soins nécessaires aux autres fourmis, à la nécessité de faire le ménage, etc.

Achille: Mon dieu! Mais elles ne sont pas simples que cela vos fourmis! Vous rendez-vous compte ? Coordonner toutes ces activités! Moi-même, qui ai déjà du mal à penser à la fois au ménage et à la vaisselle...

La Tortue: C'est justement là que le concept de spécialisation est important. N'oubliez pas que les fourmis vivent en groupe. Certaines fourmis vont donc se spécialiser dans l'une de ces tâches, et

* *ibid.* p.356 et 371

la prendre en charge dès qu'elle apparaîtra comme nécessaire. Un peu comme si vingt Achille vivaient ici, chacun s'occupant, qui de faire le ménage, qui de faire la vaisselle, qui de préparer le repas, etc.

Achille: Je vois. Mais ce que vous dites va précisément à l'inverse de la redondance dont nous parlions tout à l'heure. Je dirais même que c'est un excellent contre-exemple.

La Tortue: Non, Achille. Car n'oubliez pas que les fourmis sont des êtres fragiles. Que se passerait-il si la seule fourmi capable de trouver de la nourriture était dévorée par notre ami le fourmilier ?

Achille: Eh bien, je suppose que la colonie mourrait de faim.

La Tortue: Tout à fait. Il faut donc que plusieurs fourmis soient à même de la remplacer si elle disparaît. Et puis certaines tâches ont besoin d'être effectuées à plusieurs, si la charge de travail est trop importante pour une seule d'entre elles.

Achille: Je vois. (*il se met à réfléchir*) En somme, vous essayez de me faire comprendre qu'il faut un équilibre nécessaire entre redondance et spécialisation pour obtenir une organisation efficace ? Un peu comme... Ca y est, je comprends! C'est tout à fait comme les chapitres du livre que nous évoquions à l'instant. Tous ces chapitres sont spécialisés, pris un à un, mais je peux en sauter car il existe une redondance entre certains d'entre eux. Et donc, même en en sautant, l'ensemble de la tâche qui était de me faire comprendre le propos du livre aura été effectuée! (*il rit tout seul*) Et tenez! Encore un autre exemple! La bouilloire et la tasse sont chacun des individus spécialisés, non ? Or, s'il n'y avait pas eu redondance entre ma bouilloire et celle que le Fourmilier m'a offerte, ou entre les six tasses apportées par le Crabe, vous n'auriez jamais eu votre camomille!! Voyons, où puis-je trouver d'autres exemples...

La Tortue: Vous avez compris le mécanisme, Achille, mais...

Achille: Mais ?

La Tortue: ... mais j'ai peur que vous vous mépreniez sur un point essentiel.

Achille: Allons bon! Je ne vois pas ce qui cloche dans mon raisonnement!

La Tortue: Si je vous demandais, par exemple, de construire de A à Z un système qui reproduise exactement une fourmilière. Je veux dire, un système qui, si je l'observe, ne m'apparaisse pas différent d'une fourmilière dans des conditions normales.

Achille: Quelque chose que vous ne pourriez distinguer d'une fourmilière ? Un peu comme dans le test de Turing ?

La Tortue: Tout à fait. Appelons ce test le test de Fourming. Je prends d'un côté une fourmilière, de l'autre votre système. Puis je les regarde et, au besoin, j'interagis avec chacun d'entre eux, mais sans les voir. Disons, par l'intermédiaire d'un opérateur à qui je donne des ordres, "Mettez plus de nourriture", "Écrasez les fourmis qui sortent", etc. et qui me rapporte fidèlement le comportement de l'ensemble en réaction à chacun de ces événements. Votre système passera le test de Fourming si les réponses qu'il me fournit, par l'intermédiaire de l'opérateur, sont telles que je suis incapable de distinguer la vraie colonie de votre colonie artificielle.

Achille: Je vois. (*il prend la tête entre les deux mains*). Ce que vous me demandez-là est un peu compliqué. Laissez moi réfléchir un instant.

(*il prend une feuille de papier, et commence à noter certaines choses. Un long moment s'écoule. La Tortue, pendant ce temps, déguste la camomille*)

Bien! Je crois que j'ai votre système. En fait, ce n'est pas si compliqué. Vous m'avez d'ailleurs donné la réponse dans vos remarques précédentes.

La Tortue: Comment cela ?

Achille: Eh bien, vous ne m'avez parlé que de redondance, de spécialisation, d'adaptation et d'efficacité. Je suppose donc que le test de Fourming que vous voulez faire passer à mon système concernera ces aspects de la colonie. Je me trompe ?

La Tortue: Non. Vous avez raison. C'est effectivement la seule manière objective que j'ai de juger de la similarité de votre système avec une colonie réelle. Peu m'importe à vrai dire de savoir si vos

fourmis ont quatre ou six pattes. Ce qui m'intéresse, c'est que votre système agisse et se transforme de manière analogue à une fourmilière.

Achille: Parfait. C'est ce que je pensais. Alors, voyons... (*Il consulte ses notes*) Je construirais d'abord de petits robots capables de faire ce qu'une fourmi sait faire. Et capables, bien sûr, d'obéir aux ordres.

La Tortue: Aux ordres ? Quels ordres ?

Achille: Eh bien. Aux ordres de ... comment s'appelle la fourmi qui pond ? La reine! Aux ordres de la reine, pardi! Elle ne porte pas ce nom par hasard ! La reine qui va surveiller la situation et décider de placer plus d'ouvrières à tel endroit, ou pour s'occuper de telle tâche, afin que la redondance et la spécialisation soient équilibrées. Vous m'avez dit que les fourmis n'étaient pas intelligentes. Soit. Il faut donc bien que quelqu'un d'intelligent leur dise quoi faire.

La Tortue: Bien. Continuez. Comment construisez-vous cette reine intelligente ?

Achille: Je ne sais pas exactement. Je suppose que je ferais appel à un magicien de l'intelligence artificielle, ou quelque chose dans ce goût-là.

La Tortue: Très bien. Maintenant, admettons que le premier test de Fourming que je fasse passer à votre système soit de donner l'ordre à mon opérateur d'enlever la reine.

Achille: Ah non! Ce n'est pas de jeu!

La Tortue: Pourquoi donc ? Cela peut faire partie des conditions normales que peut rencontrer une fourmilière. Et il faut bien qu'elle se débrouille dans ce cas-là, non ? Les colonies naturelles sont tout à fait capables de survivre un certain temps sans reine.

Achille: (*il se plonge de nouveau dans ses notes*) Une deuxième reine ?

La Tortue: Je l'enlève aussi, ainsi que toutes les reines supplémentaires que vous pourriez rajouter. Oui, c'est cela. Je demande à mon opérateur d'enlever toutes les reines. Que devient votre système ?

Achille: (*il soupire bruyamment*) Un système incapable de passer le test de Fourming, je suppose...

La Tortue: Il y a en effet de fortes chances que les ordres donnés par vos reines à chacune des fourmis ne soient plus adaptés à la situation qui suivra l'enlèvement des reines. Elles continueront par exemple à faire le ménage dans le nid alors que tout est propre...

Achille: Mais alors, qu'est-ce qui ne va pas dans mon raisonnement ? J'ai pourtant bien mis de la redondance, de la spécialisation, du calcul d'efficacité, des fourmis et des reines dans ma fourmilière!

La Tortue: Ce qui ne va pas, Achille, c'est que vous avez mélangé deux choses. D'abord la vision que nous avons, vous et moi, de l'organisation d'une société de fourmis. Ce sont des concepts comme la redondance, la spécialisation, l'efficacité dont nous nous servons, de l'extérieur, pour qualifier le fonctionnement de la colonie. Ensuite, les composants mêmes de la colonie qui sont les fourmis, la reine, etc. Vous souvenez-vous de la discussion que nous avons eu à propos du réductionnisme et du holisme* ?

Achille: (*l'air sombre*) Oui, ce n'est pas un excellent souvenir.

La Tortue: Allons, allons. Vous avez bien été content que ce soit imprimé, non ? (*Achille grommelle et ne répond pas*) Le holisme stipule que le tout est irréductible à ses parties, ...

Achille: (*un peu vexé*)... et le réductionnisme qu'il y est réductible. D'où le nom. Oui, je me souviens bien de tout ça.

La Tortue: Ce qui est amusant, c'est que vous avez commencé par une approche réductionniste, en créant des fourmis alors que rien ne vous y obligeait, et que vous avez opté pour une approche holiste, en leur donnant la possibilité de se servir des concepts de haut niveau dont nous parlons depuis le début, alors que rien ne vous y obligeait non plus.

Achille: Comment ça, rien ne m'y obligeait ? Je comprends ce que vous voulez dire, mais je pense que mon approche est loin d'être fautive. Prenez une entreprise. Vous avez des ouvriers, groupés par spécialisation, convenablement répartis dans une division du travail efficace. Et vous avez un ou

* *ibid.*

plusieurs patrons qui définissent cette division, en fonction de la demande, de l'offre, de l'état de l'économie, etc.

La Tortue: Ce que vous dites est tout à fait exact, mais vous me semblez confondre O.S. et Ouvrières Spécialisées. N'oubliez pas que nous parlons des fourmis, et pas des êtres humains, mon cher Achille. Croyez-vous qu'une reine soit capable de savoir avec certitude en se levant le matin combien de proies devra récolter la colonie pour survivre ? Pensez-vous qu'elle puisse planifier l'organisation de la société sur une longue période ?

Achille: Eh bien... Non, vous avez sans doute raison. D'ailleurs, je me demande même si je serais capable de le faire... Il y a tellement de facteurs à prendre en compte. Et si je me souviens bien, beaucoup de fourmis se déplacent au hasard, n'est-ce pas ?

La Tortue: Oui. Du moins c'est ce que nous a affirmé le Fourmilier. Mais je pense que nous pouvons lui faire confiance quand il parle des fourmis.

Achille: Si c'est le cas, cela signifie que je ne serais jamais capable de prédire avec exactitude l'état de la colonie dans le futur, même proche... (*il reste un instant silencieux, pensif*) Et donc que les décisions que je prendrais risqueraient d'être totalement inadaptées à cet état. Aborder la construction d'une colonie d'un point de vue holiste n'est donc pas possible ?

La Tortue: Si. Mais uniquement à condition que vous fassiez abstraction des fourmis. Que vous ne considériez plus que les processus holistes.

Achille: Mais alors, je ne comprends plus rien à rien. Les fourmis existent bien dans la nature, n'est-ce pas ? Et vous me dites qu'elles créent de la redondance et de la spécialisation, bref des organisations, sans jamais les connaître ? Comment peut-on créer quelque chose sans le connaître ? Ca me semble absurde !

La Tortue: C'est que vous jouez encore sur les deux niveaux, mon cher Achille. Pour nous deux, observateurs naïfs, les fourmis créent effectivement des organisations qui nous semblent efficaces. Nous pouvons nous permettre de dire cela car, comme vous le dites, nous connaissons ces organisations et que nous sommes capables de les re-connaître dans ce que nous observons. Mais les fourmis, elles, ne les créent que parce qu'elles sont programmées pour ça.

Achille: Hum. Et qui les a programmées ?

La Tortue: Oh! Qui vous voulez! Zeus, l'Évolution, le Démon de Maxwell, bref, tout ce qui pour vous donne un sens à la vie. A la limite, ce n'est pas ce qui importe.

Achille: Ce n'est pas ce qui importe ? Malgré ma situation, je ne crois plus guère à Zeus, mais j'imagine que si un mauvais programmeur avait programmé vos fourmis...

La Tortue: ... elles n'auraient pas duré plusieurs centaines de millions d'années. C'est sûr. Comme je ne crois pas à l'infailibilité, je penche donc effectivement pour une hypothèse évolutionniste, avec beaucoup de brouillons en cours de route. Mais je vous le répète, là n'est pas le problème.

Achille: Et quel est-il, alors ?

La Tortue: Mon problème est de comprendre comment... Oh! regardez vos amies les fourmis ! Sous la table ! Elles forment une file.

Achille: Attendez un peu (*il se lève*). Je vais les guérir de faire des autoroutes dans mon salon...

La Tortue: Non, laissez! Laissez, Achille. Elles sont simplement en train de ramasser les miettes du gâteau que je viens de terminer. Laissez-les faire. Cela vous évitera de les ramasser vous-mêmes.

Achille: (*il se rassoit*) Tsss ! C'est uniquement pour vous faire plaisir, Madame Tortue. Sans cela j'aurais perpétré le plus grand massacre de l'histoire des fourmis...

La Tortue: A propos des fourmis, donc, le problème est de comprendre la notion d'é...

Achille: (*réveur*) Mais comment ont-elles fait pour savoir ? Il doit tout de même y avoir quelque part une intelli...

La Tortue: ...mergence. Tiens! Vous avez remarqué ?

Achille: ...gence. (*il sursaute et sort de sa rêverie*) Quoi ?

La Tortue: Les deux mots que nous avons prononcés ensemble. Moi, émergence et vous, intelligence. Vous avez remarqué ?

Achille: Mais quoi donc ?

La Tortue: Combinés tels que nous les avons prononcés, ils forment le mot *intelligence*.

Achille: Non, je n'avais pas remarqué. Et alors ? Ca n'a ni queue ni tête. C'est un nouveau jeu ?

La Tortue: Mais non, Achille. Rappelez-vous le titre du livre que je vous ai offert. "De la Synthéologie à l'*Intelligence*". Vous voyez bien que ça existe!

Achille: Reste à savoir ce que ça peut vouloir dire...

La Tortue: Mais enfin, je viens de vous le dire! *Intelligence* signifie tout simplement émergence d'intelligence.

Achille: Émergence d'intelligence ? Ca ne me parle guère plus.

La Tortue: C'est pourtant ce que je vous disais avant que vos fourmis ne nous interrompent. Tout le problème, au fond, est de comprendre comment des individus qui ne nous semblent pas intelligents arrivent ensemble à effectuer des tâches qui, elles, nous paraissent intelligentes. L'émergence d'intelligence, c'est l'émergence pour un observateur d'un phénomène auquel il prête une intelligence.

Achille: (*il réfléchit un instant*) Hum. C'est exactement la question que je me posais à propos de la file de fourmis. Et ce que vous dites... Voyons. Je cherchais quelle était l'intelligence qui dirigeait cette file de fourmis. Mais vous me dites que je prête de l'intelligence à cette file, alors que je n'en prête aucune à chacune des fourmis qui la composent... Il faut tout de même bien que l'intelligence soit quelque part, non ? (*il frissonne soudain*) On peut aller terriblement loin avec des idées pareilles... Je suis en train de vous parler, et je vous attribue une certaine intelligence. Mais...

La Tortue: ... existe-t-elle réellement? C'est une question difficile, mon brave Achille. Et je pense qu'il faut s'en remettre à la seule certitude que nous ayons sur le sujet.

Achille: Ah! Il en existe au moins une! Vous me rassurez, car je commençais à me poser des questions sur moi-même...

La Tortue: Hélas, je ne pense pas qu'elle y répondra. C'est un ami australien, Rodney Brooks, qui me l'a ainsi présentée: "L'intelligence est dans l'oeil de l'observateur".

Achille: Effectivement, ça ne m'aide en rien. Je dirais même que ça m'inquiète encore plus. (*il va de long en large et fait les cent pas dans le salon*) Mais enfin, il doit tout de même y avoir un moyen de séparer ce qui est intelligent de ce qui ne l'est pas ? Je ne sais pas, le raisonnement, par exemple ?

La Tortue: Je ne crois pas que ce soit en changeant de vocabulaire que vous parviendrez à résoudre ce problème. Quand vous "prêtez" de l'intelligence à la colonne de fourmis qui, d'ailleurs, monte maintenant sur la table, vous lui prêtez également une certaine capacité de raisonnement, n'est-ce pas ?

Achille: Oui. Bien sûr. Oooh! En fait, je ne sais plus. Mais tout de même, il y a bien des travaux qui requièrent de l'intelligence, non ? Par exemple, résoudre des problèmes ?

La Tortue: Certainement. Mais je puis vous assurer que la colonne de fourmis, qui, soit dit en passant, se dirige maintenant vers la table basse, a résolu un grand nombre de problèmes très complexes pour parvenir jusqu'à nous.

Achille: Donc, si je comprends bien, je dois qualifier d'intelligent tout phénomène qui m'apparaît intelligent, même si je suis intimement persuadé qu'il n'est pas intelligent ?

La Tortue: Rien ni personne ne vous oblige à le faire. Et de toutes façons, c'est très subjectif. Tenez, par exemple, pensez-vous être intelligent ?

Achille: Oui. Du moins... je le pense.

La Tortue: Grand benêt! C'est précisément ce que je vous demande. Pensez-vous que votre cerveau soit intelligent ?

Achille: (hésitant) Oui... Non! Je le connais mal, mais ma réponse est non. Un cerveau n'est pas intelligent, il est uniquement le siège de l'intelligence.

La Tortue: Uniquement ? Bien. Et si je prends un à un tous vos organes, sont-ils intelligents ?

Achille: Là, non! Pas plus que mon cerveau.

La Tortue: Comment pouvez-vous vous considérer comme intelligent, dans ces conditions ? Et pourquoi déniez-vous cette qualité à la colonne de fourmis ?

Achille: Je vous avoue que je ne le sais pas.

La Tortue: Est-ce parce que les fourmis ne vous semblent pas résoudre les problèmes à la manière de l'homme ?

(entre le Fourmilier)

Le Fourmilier: Bonjour mes amis! Je ne vous dérange pas au moins ?

Achille: Mais pas du tout! Nous parlions d'intelligence!

Le Fourmilier: A la bonne heure. *(il regarde autour de lui et avise le jeu d'échecs sur la table basse).* Mais si, je vous dérange, j'interromps votre partie !

La Tortue: Non, nous ne jouons pas, je viens juste d'offrir ce jeu à Achille. Mais asseyez-vous donc avec nous, nous parlions justement de fourmis !

Le Fourmilier: Comment cela, vous ne jouez pas ? *(il regarde de nouveau le jeu d'échecs)* Et qui donc est en train de reproduire la variante de l'ouverture indienne Spassky-Fisher de 1967?

Achille: (qui fixe le jeu d'échecs):

La Tortue: (qui fixe le jeu d'échecs):

Achille (d'une voix tremblante à la Tortue): Où m'aviez-vous dit que se trouvait la colonne ?



INTRODUCTION

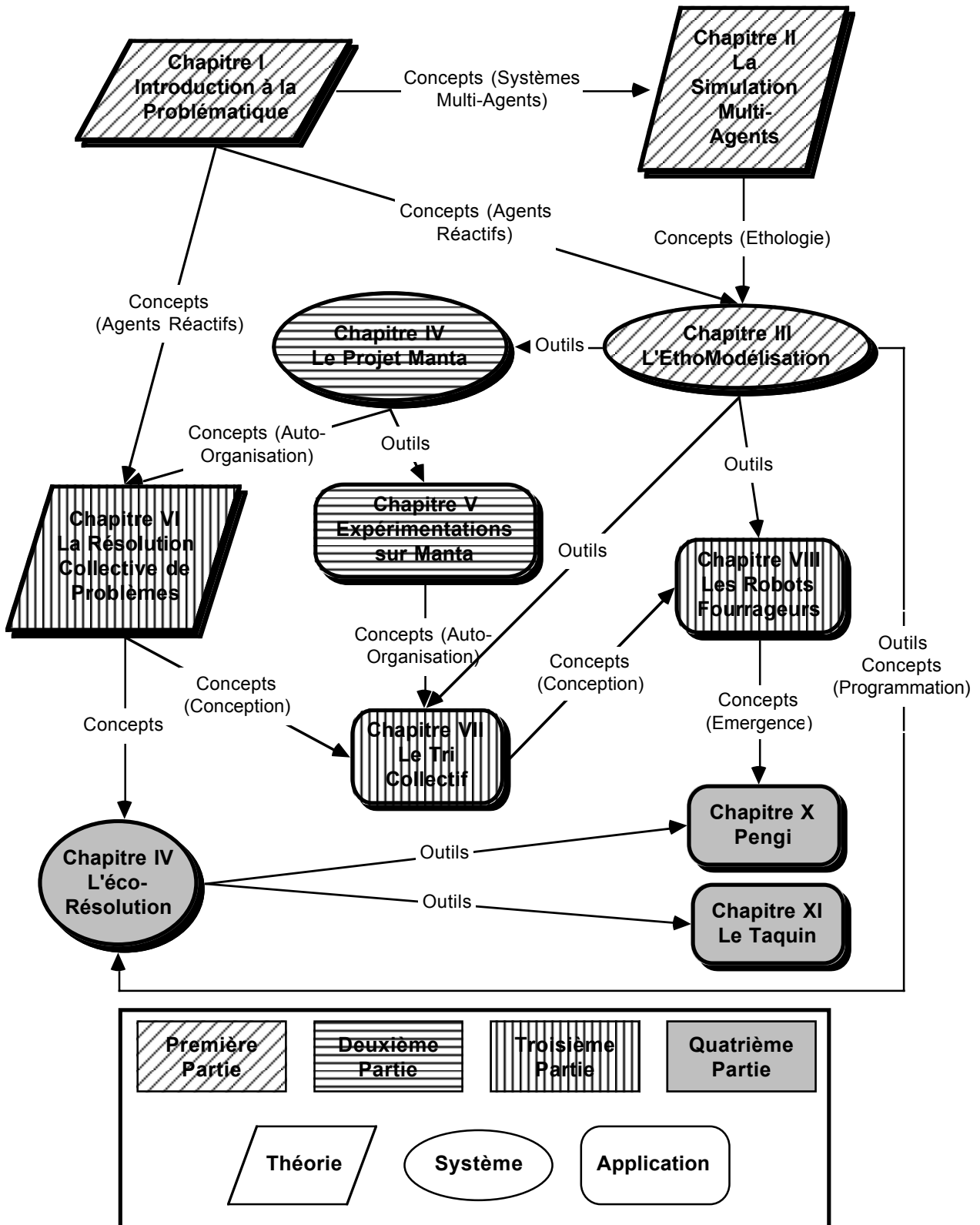
PRESENTATION ET (AUTO-?) ORGANISATION DE LA THESE

La thématique principale des recherches présentées dans cette thèse est la capacité que possèdent des entités non intelligentes individuellement de s'auto-organiser pour créer un tout dont le comportement peut être qualifié d'intelligent. Pour reprendre les mots-clés utilisés dans le sous-titre de la thèse, nous étudions donc l'émergence de propriétés intelligentes dans des systèmes multi-agents, propriétés caractérisées par l'apparition de structures d'organisation avec lesquelles elles entretiennent un lien de causalité circulaire. Cette thématique est illustrée par de nombreux travaux empiriques dont le déroulement n'a pas toujours été linéaire et qui se sont en conséquence enrichis mutuellement, les apports des uns venant nourrir l'inspiration des autres et inversement.

Malgré la linéarité imposée par tout processus de rédaction, qui nécessite une importante reconstruction du cheminement intellectuel effectué, nous nous sommes donc efforcés de préserver la forte interaction qui a existé depuis le début entre les deux grands thèmes exposés dans la thèse, qui sont d'une part la simulation de phénomènes naturels d'auto-organisation et d'autre part l'application de concepts d'auto-organisation au domaine de la résolution distribuée de problèmes. Cette interaction s'exprime par un dialogue entre des chapitres appartenant à des parties distinctes, dialogue en partie représenté dans l'organigramme par les flèches reliant les chapitres entre eux. Cet organigramme ne doit pas effrayer le lecteur: une lecture normale de la thèse débute par celle du Chapitre I et se termine par celle du Chapitre XI, et les interactions entre chapitres sont signalées par des références ou des rappels explicites. Il nous permet cependant d'en présenter l'organisation générale, caractérisée par les points suivants:

- La Première Partie est constituée de trois chapitres, qui sont des chapitres d'introduction. Le Chapitre I introduit la plupart des concepts informatiques manipulés dans les chapitres suivants et sa lecture constitue à ce titre un préalable indispensable à celle de la thèse. Le Chapitre II introduit la notion de simulation multi-agents en la situant par rapport aux travaux et aux besoins existants dans les sciences de l'homme et de la vie. Enfin, le Chapitre III introduit le système d'EthoModélisation, qui servira de support informatique aux applications développées dans les Deuxième et Troisième Parties, ainsi qu'au Chapitre X de la Quatrième Partie.
- La Deuxième Partie est constituée de deux chapitres, qui présentent l'ensemble du travail que nous avons réalisé sur la simulation multi-agents d'une société de fourmis. Elle s'ouvre sur le Chapitre IV, dans lequel est exposée la programmation du projet MANTA à partir du système développé au Chapitre III. Certains des concepts évoqués font explicitement référence à ceux développés dans le Chapitre II, mais il n'est pas obligatoire de l'avoir lu. Le Chapitre V nous sert quant à lui à analyser les résultats expérimentaux obtenus à partir des expériences menées sur ce projet, résultats qui nous servent à fonder la démarche de résolution de problèmes présentée dans les parties suivantes.
- La Troisième Partie est constituée de trois chapitres. Le Chapitre VI expose les concepts de résolution collective de problèmes que nous développons, au cours des Chapitre VII et VIII, dans deux applications inspirées des comportements collectifs des fourmis. Cette partie peut être abordée indépendamment de la partie précédente, mais il est utile d'avoir lu les Chapitres I et III pour bien l'appréhender, notamment en termes de programmation.
- Enfin, la Quatrième Partie traite de l'intégration des concepts de résolution collective de problèmes dans un système de résolution distribuée de problèmes existant, qui est l'Eco-Résolution. Le Chapitre IX introduit les notions essentielles de ce système, notions qui, dans les

Chapitres X et XI, nous servent à présenter la résolution de deux problèmes maintenant classiques, Pengi et le Taquin. Cette partie peut être lue indépendamment des précédentes, mais effectue de nombreux renvois aux concepts clés de la résolution collective de problèmes et d'émergence présentés dans la Troisième Partie.



Première Partie

Intelligence Artificielle Distribuée Réactive et Simulation Multi-Agent

[Nos travaux] reposent explicitement sur l'hypothèse que l'intelligence peut être représentée par l'activité de systèmes de symboles. Enoncée un peu plus formellement, l'hypothèse est qu'un système symbole physique (...) a les moyens nécessaires et suffisants pour une «action générale intelligente». *Cette hypothèse est tout à fait empirique, susceptible d'être jugée vraie ou fausse sur la base de l'évidence observée.*

H. Simon, *Sciences des Systèmes, Sciences de l'Artificiel*, 1969
(Edition française: Dunod, Paris, 1991, p.24, souligné par nous)

Ce n'est pas la conscience qui détermine la vie, mais la vie qui détermine la conscience. Dans la première façon de considérer les choses, on part de la conscience comme étant l'individu vivant, dans la seconde façon, qui correspond à la vie réelle, on part des individus réels et vivants eux-mêmes

K. Marx et F. Engels, *L'Idéologie Allemande*, 1845
(Edition Française: Editions Sociales, Paris, 1968, p. 37)

CHAPITRE I

INTRODUCTION A LA PROBLEMATIQUE

Avant-Propos

Ce chapitre se veut, comme son nom l'indique, un chapitre d'introduction à la problématique qui va sous-tendre l'ensemble du travail présenté dans cette thèse. Une introduction est d'abord l'occasion de familiariser le lecteur avec les concepts employés. Ces concepts ne sont pas très nombreux, mais communs à de nombreuses disciplines. Il importe donc que nous les définissions, sinon en toute rigueur, du moins le plus clairement possible, afin que la vision intuitive que chacun possède à leur sujet ne soit pas source de confusion pour la compréhension de l'ensemble. Cette introduction est également l'occasion de montrer comment nous nous positionnons dans le domaine de recherche, l'Intelligence Artificielle Distribuée, qui est le nôtre, et quelles sont les conséquences méthodologiques de cette position sur la conception de systèmes intelligents. Ces conséquences nous permettent alors de développer la méthodologie empirique et constructiviste que nous préconisons, et de montrer comment celle-ci structure l'ensemble de ce travail.

I.1. L'Intelligence Artificielle Distribuée

I.1.1. Définitions

La définition, dans son acceptation la plus commune, du domaine de recherche de l'Intelligence Artificielle Distribuée est celui de l'étude et de la conception d'*organisations d'agents artificiels* pour obtenir des *systèmes intelligents* [Erceau et Ferber 1991; Seel 1991]. La notion d'*agent artificiel* fait référence à tout artefact, logiciel aussi bien que matériel, capable d'exhiber une certaine autonomie, donc opérationnellement et informationnellement clos [Varela 1983] par rapport à l'environnement dans lequel il est plongé. L'environnement désigne ici aussi bien l'espace (topologique, temporel) que les autres agents (l'environnement social), avec lesquels l'agent forme un *système multi-agents*. Le couplage entre l'agent et son environnement se réalise par des capacités d'action¹ et de perception² qui lui sont propres. L'ensemble des actions qui s'actualisent dans un environnement donné forme le *comportement* de l'agent. Nous ne faisons ici aucune hypothèse quant aux mécanismes régissant ce comportement. La notion de *systèmes intelligents* doit être comprise, et ceci sera important par la suite, comme la référence à des systèmes *apparemment* intelligents, c'est-à-dire des systèmes auxquels un observateur humain peut *prêter* une intelligence dans la réalisation d'une tâche, dans la résolution d'un problème, ou dans son interaction avec lui-même, quels que soient, par ailleurs, les mécanismes sous-jacents conduisant à cette "intelligence". Elle n'est donc, dans cette perspective, pas très différente de celle habituellement employée en Intelligence Artificielle [Simon 1991], et nous ne discuterons pas plus avant des problèmes qu'elle a pu et peut encore soulever.

¹ L'étymologie du mot agent provient du latin *agens*, participe du verbe *agere*. Signifie tout ce qui agit. Le mot action a la même étymologie (*actum*).

² La communication entre agents est envisagée ici comme une action pour celui qui la transmet et comme une perception pour celui qui la reçoit. De même, l'interaction est simplement une action réciproque, ou une perception réciproque, selon le point de vue.

Une fois ceci défini, il reste que le terme IAD est un terme plutôt ambiguë, dont on ne sait pas très bien s'il désigne une "suite" logique à l'Intelligence Artificielle, ou un domaine de recherche autonome. Cette ambiguïté a permis la coexistence en son sein de deux courants de recherche opposés, qui, loin de l'affaiblir, ont au contraire, par leur existence, favorisé l'émergence d'une dialectique intéressante et potentiellement très riche. Cette dialectique oppose d'un côté les tenants d'une approche que nous appellerons par la suite *cognitive*, de l'autre ceux d'une approche que nous appellerons *réactive*. L'existence de cette dialectique n'est pas surprenante à nos yeux puisque, épistémologiquement, ces deux conceptions de l'IAD puisent leurs racines dans des domaines distincts, et résultent de conceptions très différentes de ce que doit être une *organisation*. Très schématiquement, l'IAD cognitive se base sur une conception *sociologique* de l'organisation, alors que l'IAD réactive a plutôt tendance à privilégier une approche *biologique* de l'organisation [Ferber 1989]. Il ne s'agit, la plupart du temps, que de métaphores, mais une métaphore n'est jamais neutre quant à ses choix et signifiants.

1.1.2. L'approche cognitive

Si nous évoquons la sociologie à son propos, c'est que l'IAD cognitive est pour une large part tributaire du projet cognitiviste de l'IA, et donc de l'anthropomorphisme inhérent à cette approche. Pour simplifier, nous pouvons décrire ce projet comme reposant presque exclusivement sur l'irréductibilité d'un traitement symbolique de la perception et de l'action par l'agent (la fameuse "symbol system hypothesis" de [Newell et Simon 1976]). En découlent les deux notions fondamentales qui sont la notion de représentation [Varela 1989] des connaissances, de soi, du monde, des autres, de ses buts, et la notion de contrôle par l'agent de son comportement, donc de sa perception et de ses actions, en fonction justement de ses représentations [Collinot 1988]. Comme le rappellent Alan Bond et Les Gasser (dans l'introduction de [Bond et Gasser 1988]), l'IAD cognitive est née de la volonté de faire évoluer les concepts et les outils de l'IA traditionnelle pour synthétiser la part *sociale* ou *interactionnelle* de la connaissance et de l'intelligence humaines. Pouvoir représenter et utiliser des points de vue et des expertises multiples, pouvoir résoudre des problèmes physiquement distribués nécessite en effet "autre chose" que le point de vue solipsiste qui a donné naissance aux systèmes experts. Et cet "autre chose", ce "plus" social entre individus rationnels se trouve précisément étudié par la sociologie, ou tout au moins par certaines branches de la sociologie, comme l'individualisme méthodologique (voir Section II.1.3.1), la sociologie des groupes restreints [Anzieu et Martin 1969] ou la psychosociologie [Levy 1965], pour ne citer que les plus connues. Il était donc naturel que l'IAD cognitive y puise son inspiration. Il serait sans doute trop long, et en-dehors du cadre cette étude, de développer les théories employées dans ces domaines. Nous ne saurions donc trop conseiller au lecteur de se reporter à [Bouron 1992] qui présente un panorama complet des différentes approches. On se reportera également au Chapitre VI pour une discussion sur l'approche cognitive en Résolution Distribuée de Problèmes. De par ses bases cognitivistes, et de par ses choix de modèles d'organisation, la thèse défendue en définitive par l'IAD cognitive est donc qu'un système multi-agents *intelligent* résulte de l'organisation *intelligente* d'agents *intelligents*.

1.1.3. L'approche réactive

L'IAD réactive postule tout simplement le contraire, du moins dans sa version la plus radicale. Les modèles d'agents réactifs sont particulièrement nombreux, et nous n'en retiendrons que deux traits distinctifs. Un agent réactif ne possède pas de représentations symboliques de son environnement, ni de lui-même. Par conséquent, il n'est pas censé pouvoir contrôler son propre comportement. Son contrôle est entièrement pris en charge par un mécanisme rigide qui se sert du résultat de la perception pour déclencher une action, et ce mécanisme n'est modifiable ni par l'action, ni par la perception de

l'agent. En ce sens, le contrôle du comportement est donc en partie effectué par l'environnement (c'est la non moins fameuse "physical grounding hypothesis" [Brooks 1991]).

L'IAD réactive a des bases épistémologiques plus floues et plus diffuses que l'IAD cognitive. Un lien certain la relie au courant auto-organisationnel de la cybernétique, représenté principalement par [Von Foerster 1960] et à ses résurgences récentes dans le cadre de la Vie Artificielle [Langton 1988; Langton et al. 1990]. Une autre source d'influence, également issue de la cybernétique [Wiener 1948], est constituée par les travaux effectués en robotique par [Brooks 1983]. Elle doit enfin beaucoup à deux approches de programmation, qui sont respectivement l'approche orientée objets [Ferber 1989], et l'approche orientée acteurs [Giroux 1993]. Le projet de l'IAD réactive peut être parfaitement résumé par la métaphore biologique que nous mentionnons plus haut, en ce qu'il postule qu'un comportement intelligent peut émerger de l'interaction d'un grand nombre d'agents qui, pris individuellement, ne disposent d'aucune intelligence propre [Hubbermann et Hogg 1988; Minsky 1988; Varela 1989]. Le système immunitaire, présenté par ce dernier auteur, les travaux de construction effectués par les insectes sociaux [Wilson 1985] en sont tous deux de bons exemples. Les modes d'organisation des agents empruntent donc leurs concepts aux travaux effectués en biologie sur les notions d'organisation et d'auto-organisation [Atlan 1972], en éthologie sur la socialité animale [Rabaud 1937; Tinbergen 1979], ou en écologie sur les notions d'écosystèmes [Carbiener 1991; De Rosnay 1975; Zeigler 1978].

1.1.4. Positionnement de ce Travail

Le travail que nous présentons dans cette thèse a pour ambition d'alimenter par une expérience pragmatique et phénoménologique (descriptive) la dialectique présentée, en proposant un certain nombre d'études dans lesquelles des systèmes perçus comme intelligents ont été obtenus par la programmation de systèmes multi-agents composés d'agents non intelligents. Nous nous positionnons donc clairement dans une perspective réactive. Nous ne ferons cependant, dans les chapitres qui suivent, aucun procès de l'approche cognitive, qui nous apparaît aussi utile à la connaissance scientifique que l'approche dont nous sommes les tenants. Nous pensons en effet *«que le mouvement dialectique de la pensée est le reflet du monde réel»* ([Marx et Engels 1968]), et que l'approche cognitive comme l'approche réactive analysent ce monde réel et le synthétisent sous deux formes différentes d'organisation, parce qu'il est lui-même dialectique, pris *«entre l'homme et la fourmi»* [Hadet 1993]. Et si, comme l'affirme Platon, *«toute dialectique se ramène à la construction de la connaissance vraie»* [Julia 1992], nous pouvons espérer que de la confrontation de la thèse cognitive et de l'anti-thèse réactive naîtra une synthèse englobant les divers aspects que nous avons mentionnés, identifiant les agents et les organisations à utiliser pour appréhender les problèmes qui lui seront fournis, sans préjuger de leur appartenance à telle ou telle école, mais uniquement en fonction de leur adéquation³. Dans cette perspective, une chose est certaine: nous avons besoin de précisément définir ce qu'apportent les deux approches, quelles sont leurs qualités, leurs défauts et leurs limites. Pour notre part, nous avons choisi le "champ" réactif, mais, à y regarder de plus près, les concepts abordés dans ce travail pourraient tout autant s'appliquer au "champ" cognitif, qui ne pourra longtemps s'affranchir d'une réflexion approfondie sur la part de l'auto-organisation dans l'organisation des groupe sociaux qu'il simule [Castelfranchi et Conte 1992].

³ Un mouvement en ce sens semble se dessiner, par exemple avec la science des organisations artificielles, la kénétique, que Jacques Ferber appelle de ses vœux [Ferber, en prép.], ou au sein d'une systémique qui se débarrasse peu à peu de ses démons holistes [Le Fur 1993]. Des travaux portant sur la combinaison de comportements réactifs et cognitifs au sein d'un agent sont également en cours [Ferguson 1991; Kiss 1991].

Une approche se juge à l'aune de ses domaines potentiels d'application et des méthodologies permettant de concevoir les systèmes qui les exploitent. Le problème de l'approche réactive à l'heure actuelle est qu'elle manque cruellement des deux. Il n'existe pas à proprement parler de domaines privilégiés d'application, et les méthodologies de conception sont encore rudimentaires et peu exploitables (e.g. [Miriad 1992]). Ces deux constatations font que notre contribution au domaine réactif doit à la fois être comprise comme applicative (*i.e.*, en simulation et en résolution de problèmes) et méthodologique (*i.e.*, comment s'inspirer de systèmes réactifs existants par le biais de la simulation pour concevoir des systèmes de résolution de problèmes). Les applications qui sont présentées dans cet ouvrage s'insèrent ainsi dans un cadre méthodologique général, cadre dans lequel nous répondons en partie à certains des problèmes de conception qui sont développés dans les sections suivantes.

I.2. Des Principes d'une Approche Réactive

Le modèle d'agent réactif que nous utiliserons pour programmer les applications présentées dans ce travail est décrit au Chapitre III. Cependant, il importe de poser avec clarté le cadre général de ce modèle, afin que l'enjeu de la thèse que nous défendons ne puisse être taxé d'ambiguïté. Pour ce faire, nous aurons recours à la définition de quelques *principes*, dont nous pensons qu'ils constituent une bonne approximation de ce qui fonde toute conception d'agents réactifs. Certains d'entre eux pourront apparaître comme des truismes, d'autres comme des postulats. Nous nous défendons cependant de tout dogmatisme, et ces principes ne sont pas à prendre pour la "vérité révélée" de la démarche réactive, mais bien plutôt comme le reflet de notre expérience en ce domaine. Le cadre qu'ils constituent nous permettra ainsi d'introduire notre problématique (Section I.4.) sans préjuger des connaissances du lecteur.

I.2.1. Le modèle d'agent

La relation d'un agent réactif à son environnement est un couplage par *stimulus/réaction*, proche du couplage par *entrée/sortie* tel qu'il est défini par [Deffuant 1992]. Dans ce cadre, la *perception* dénote le fait pour l'agent de ne considérer que certains éléments constitutifs de son environnement, qui seront les *entrées* ou les *stimuli*, auxquels correspondront des *réactions* sous la forme d'*actions*. Le comportement des agents réactifs est donc basé sur le principe de l'action comme réaction à un stimulus (sur l'affinité de ce modèle avec certains modèles éthologiques, voir Section III.2.2.).

Principe de réaction

Une réaction dénote l'exécution, selon un déterminisme strict, d'une action ou d'une séquence fixe d'actions en réponse à un stimulus. Le comportement potentiel d'un agent réactif est une liste de réactions.

Cependant, l'utilisation de ce principe ne peut être envisagé que si l'agent se montre capable de choisir certains stimuli par rapport à d'autres. Si son comportement n'est décrit que par une liste non ordonnée de réactions, que doit faire l'agent en présence de *n* stimuli conduisant à *n* réactions possibles dont certaines peuvent être contradictoires ? Nous sommes donc contraints de lui fournir un mécanisme lui permettant de choisir entre ces *n* stimuli pour aboutir à une réaction, tout en respectant le fait qu'il ne puisse pas modifier ce mécanisme. Nous parlons dans ce cas d'une discrimination passive, par opposition à une discrimination active⁴ qui résulterait de l'activité de l'agent.

⁴ Equivalent à la focalisation, qui implique une intentionnalité dans ce qu'il faut percevoir.

Principe de discrimination passive

Au mécanisme de perception est couplé un mécanisme de discrimination des stimuli. Le produit de cette discrimination consiste en un stimulus choisi selon des critères dont le contrôle échappe à l'agent, et qui servira à déclencher une et une seule réaction.

Il est à noter que la définition de ce processus de discrimination ne fait intervenir aucune contrainte quant au choix des critères de discrimination. Un grand nombre de cas de figure est alors possible et, de fait, la diversité constatée des modèles d'agents réactifs provient de la diversité des critères de discrimination employés. Dans le cas de critères fixes, nous pouvons ainsi envisager un critère aléatoire (*i.e.* n'importe quel stimulus est alors choisi), un critère basé sur la définition d'un ordre total de priorité entre les réactions [Brooks et Connell 1986] ou un critère basé sur la reconnaissance de certaines propriétés des stimuli (*e. g.* leur intensité, auquel cas le contrôle est entièrement effectué par l'environnement). Il est également envisageable d'utiliser des critères évolutifs dans le temps, mais à la condition expresse que leur évolution ne soit pas contrôlée par l'agent. Un ordre de priorité peut ainsi évoluer en fonction de la fréquence de déclenchement d'une réaction (voir Chapitre III).

1.2.2. Le principe de parcimonie

Les deux principes de conception d'un agent réactif fournis dans la section précédente peuvent laisser une impression troublante. Nous avons en effet omis toute référence à ce qui constitue une bonne partie de l'intérêt de la réactivité, à savoir la simplicité du comportement des agents. Cette omission provient du fait que la notion de simplicité est éminemment subjective, intimement contrainte par les choix d'organisation et d'agencement des agents entre eux, et donc par le choix de *décomposition* en agents de l'application ou du problème à résoudre (nous abordons cette notion de décomposition au Chapitre VI). Seule la connaissance de ces choix permettra d'établir une mesure de la simplicité requise. Il est cependant possible de percevoir une mesure intuitive de simplicité dans la notion de *granularité*. La granularité d'un agent dénote un ensemble de considérations informatiques statiques et dynamiques, pondérées par des coûts effectifs, qui prennent respectivement en compte la taille du code minimal nécessaire pour décrire complètement l'agent, la taille de la mémoire (à long terme comme à court terme) dont cet agent dispose et sa rapidité d'exécution. Il n'est guère envisageable d'établir une échelle absolue de granularité, dans la mesure où elle dépend, entre autres choses, du matériel employé (et donc des coûts respectifs de chacun de ces critères), mais on peut envisager, pour le même matériel, d'établir un ordre partiel entre différents agents chargés de tâches semblables. Ceci étant défini, nous pouvons ainsi poser comme fondement de toute démarche réactive le principe de *parcimonie*. Dans sa définition, nous positionnons ce principe par rapport à la résolution d'un problème par l'agent, le problème étant ici la réalisation de tâches pour lesquelles il a été conçu.

Le Principe de Parcimonie⁵

Il n'est pas nécessaire de faire appel à un agent de granularité élevée pour résoudre un problème s'il s'avère qu'un agent doté d'une granularité inférieure est capable de lui trouver une solution.

Autrement dit, l'idée de base est que seul compte le résultat et que nous nous efforçons de concevoir les agents les plus simples permettant de l'obtenir. On conçoit bien qu'un agent possédant une perception locale, une méthode de discrimination simple et peu de réactions possibles sera un meilleur candidat qu'un agent ayant une perception globale, une méthode de discrimination complexe, et un

⁵ voir Chapitre III.

grand nombre de réactions possibles, si, bien sûr, ils résolvent tous deux le problème qui leur est posé. Ce principe induit naturellement deux corollaires, qui définissent de manière générale une méthodologie à la fois réductionniste et incrémentale.

Corollaire 1: Le Principe de Réductionnisme

Avant de faire appel à un agent d'une granularité donnée pour résoudre un problème, il convient de vérifier qu'un agent de granularité inférieure n'est pas en mesure de lui donner une solution.

Corollaire 2: Le Principe d'Incrémentalité

On ne doit faire appel à un agent d'une certaine granularité que si les agents de granularité inférieure sont incapables de donner une solution au problème posé.

C'est cette méthodologie qui distingue radicalement l'approche réactive de l'approche cognitive, qui, elle, se place d'emblée à un niveau élevé de granularité en postulant des capacités de mémorisation, de perception et de délibération importantes. Ces principes peuvent être de plus récursivement employés pour la définition du comportement de l'agent, donc de chacune de ses actions, et surtout pour la définition de l'organisation d'agents (voir par exemple le Chapitre VI). Du fait de ce principe de parcimonie, la relation entre l'agent et son environnement sera beaucoup plus étroite que celle pouvant exister entre un agent cognitif et son environnement qui, elle, passe par un niveau irréductible de représentation. Il ne sera ainsi pas rare de voir l'agent utiliser l'environnement comme mémoire externe en y laissant des traces [Drogoul et Ferber 1993; Steels 1989], ou en en extrayant les informations de manière analogique [Connell 1990; Drogoul, Ferber et Jacopin 1991; Steels 1990].

1.3. Les Organisations Réactives

1.3.1. Définition

L'introduction du concept d'organisation après celui d'agent ne doit pas faire illusion. La raison d'être principale des agents réactifs réside précisément dans l'intérêt fondamental de ces organisations. Et les problèmes méthodologiques de conception de systèmes réactifs proviennent de la difficulté d'obtenir des organisations efficaces à partir d'agents incapables de les penser. Comme nous le verrons dans les sections suivantes, une organisation réactive est ainsi souvent plus efficace, plus adaptative et plus souple qu'une organisation cognitive, mais également beaucoup plus difficile à concevoir. Après tout, si l'on reprend dans notre perspective les arguments classiques de la biologie réductionniste, l'homme ne peut être objectivement décrit que comme une organisation particulière de cellules (et, si l'on voulait descendre plus bas, de molécules) qui, prises individuellement, peuvent difficilement être créditées d'intelligence. Et pourtant, nous attribuons à cette organisation de cellules des propriétés d'intelligence (voir à ce sujet [Hofstadter 1985], mais également [Beni et Hackwood 1992; Dumouchel et Dupuy 1983; Monod 1970; Morin 1977; Prigogine et Stengers 1985; Varela 1983; Von Foerster 1960]).

La nature nous fournit donc des exemples d'organisations d'entités simples qui parviennent à accomplir des fonctionnalités qu'aucune de ces entités, prise individuellement, ne serait en mesure d'accomplir. Cependant, outre cette constatation, qui ne suffirait pas à elle seule à fonder une démarche de recherche, la caractéristique la plus remarquable de ces organisations est qu'elles sont souvent extrêmement adaptatives, aussi bien à la transformation de leur environnement qu'à leur propre transformation, et manifestent ce que l'on appelle de manière générale une capacité homéostatique ou une capacité d'autorégulation [Dumouchel et Dupuy 1983]. Des milliers de neurones meurent chaque jour, mais le cerveau humain reste longtemps capable de fonctionner de manière égale. De même,

retirer à une colonie de fourmis un grand nombre d'ouvrières ne l'empêche pas de continuer à exister. Ce qui, dans ces systèmes, intéresse donc les chercheurs est la possibilité d'obtenir des comportements intelligents *et* adaptatifs, dont les performances ne se dégradent pas de manière catastrophique au premier dysfonctionnement de leurs composants. Les exemples les plus connus d'application se situent à l'heure actuelle en robotique [Goss et Deneubourg 1992; Jacopin 1991; Waldrop 1990], où il apparaît, pour un certain nombre de tâches (exploration d'un terrain inconnu, surveillance, etc.) qu'il est préférable, pour le même coût, d'avoir une population nombreuse de robots simples plutôt que quelques robots aux comportements plus complexes. La capacité homéostatique de la population, pour laquelle on emploie souvent le mot *robustesse*, assure dans ce cas que la population, grâce à la redondance existant entre les agents, continuera à effectuer la tâche pour laquelle elle a été conçue en dépit du dysfonctionnement de certains de ses membres. L'organisation elle-même peut être fondée sur de nombreuses caractéristiques. Certaines émergent par interaction directe entre les agents [Ferber 1990b; Mataric 1993], par interaction indirecte via l'environnement, comme le fait de déposer des traces (voir plus haut les relations de l'agent à son environnement) ou d'établir des influences par propagation d'information [Drogoul 1993], ou enfin par adaptation à la configuration même de l'environnement [Demazeau 1991]. En paraphrasant Francisco Varela [Varela 1989], nous pouvons donc décrire un système multi-agents réactifs de la manière suivante:

Question 1: Qu'est ce que l'intelligence d'un système multi-agent réactif ?

Réponse: L'émergence d'un ou plusieurs états globaux dans ce groupe d'agents réactifs, ou dans l'environnement dans lequel ils évoluent, comme résultat de l'activité des agents.

Question 2: Comment cela fonctionne-t-il ?

Réponse: Des "règles" locales gèrent les comportements individuels et leurs interactions, entre eux et leur environnement.

Question 3: Comment savoir qu'un système fonctionne de manière appropriée ?

Réponse: Quand les propriétés émergentes (et les structures résultantes) de ces états globaux sont identifiables à une solution adéquate pour une tâche donnée.

Comme on le comprend à la lecture de la définition précédente, le concept crucial permettant de concevoir et d'obtenir un système multi-agents réactif *intelligent* est la notion d'*émergence*. Bien que nous montrions au Chapitre VIII qu'il est possible d'envisager des systèmes réactifs dans lesquels aucune émergence n'intervient, et qui ne jouent que sur *l'effet de groupe* et la redondance individuelle pour fonctionner avec une efficacité *amplifiée* par rapport à un système mono-agent, nous verrons qu'ils ne correspondent pas à l'idée que nous défendons d'un comportement intelligent. Cette notion d'émergence est donc centrale, et il importe de la définir, ou du moins de mieux la cerner.

1.3.2. De l'émergence à l'auto-organisation

Atteindre le mécanisme causal d'une genèse consiste en premier lieu à reconstituer ce qui est donné au point de départ (...) et en second lieu à montrer de quelle manière (...) ces structures de départ se transforment en celles dont il s'agit de rendre compte. [Piaget 1967]

L'émergence est un concept flou, mal défini et souvent décrié. Il dénote d'une manière générale, dans les systèmes qualifiés de complexes ou de non-linéaires [Moyson et Manderick 1988; Prigogine et Stengers 1985], l'apparition d'un effet global qui n'est pas déductible de la connaissance des causes locales [Mataric 1993]. Dans un système multi-agent, il dénote donc l'apparition d'un phénomène global qui n'est pas explicitement programmée dans le comportement des agents [Beni et Hackwood 1992]. Nous pouvons ainsi concevoir la "quantité" d'émergence que l'on impute à un système comme la

"quantité" d'ignorance par l'observateur des mécanismes de causalité entre les causes locales et l'effet global. Le premier point est donc que "voir" de l'émergence implique au moins la disponibilité de deux niveaux⁶. Un niveau qualifié de *micro* (les causes locales) et un niveau qualifié de *macro* (le phénomène global). Le niveau micro est connu et/ou spécifié, le niveau macro observé [Assad et Packard 1992]. Mais cette définition est loin d'être satisfaisante, car, comme le souligne très justement [Castelfranchi 1993], toute composition d'effets accidentels, ou tout ce qui ressemble à un équilibre dynamique au niveau global, pourra à ce titre être qualifié d'émergent, y compris les erreurs d'exécution ou de conception conduisant à des comportements aberrants. C'est là que s'impose une première restriction quant à la définition de l'émergence, que nous appellerons restriction fonctionnelle. Car en définitive, que veut dire concevoir un système multi-agents *intelligent*, par exemple pour résoudre un problème? Dans l'absolu, rien. L'intelligence n'est que ce que nous, en tant qu'observateur, attribuons au système. Et deux conditions nous semblent indispensables pour que nous le fassions. Premièrement, que le système vienne à bout de problèmes que nous n'imaginerions pas pouvoir résoudre sans faire appel à *notre* intelligence. Deuxièmement, que le système innove en découvrant des solutions que nous ne lui avons pas explicitement fournies. Nous ne serons donc enclin à voir une émergence d'intelligence dans le fonctionnement du système que si son comportement nous apparaît fonctionnellement intéressant [Steels 1991].

Deux critères d'"intelligence" sont ainsi utilisés de manière concomitante: l'un facile à déterminer, qui est que la tâche a été effectuée, et l'autre plus difficile à évaluer, qui porte sur la manière dont elle a été effectuée. Or, en terme de méthodologie de programmation, le deuxième critère apparaît fondamental à évaluer, car comprendre la dynamique de résolution d'un problème donné nous permet de généraliser cette approche à d'autres problèmes équivalents. Simplement considérer l'effet final (le résultat) ne nous donne donc en fait aucune indication sur ce qui nous intéresse, à savoir la manière dont la société s'est organisée pour faire émerger une capacité fonctionnelle qui lui était jusqu'alors inaccessible, ce que nous laissait déjà présager la citation de Jean Piaget en tête de cette section. Nous devons donc considérer, pour comprendre cette émergence, et pour être certain qu'elle n'est pas simplement due à une distribution initiale particulière des agents, les processus organisationnels à l'œuvre dans notre groupe d'agents, ce que l'on note plus généralement, dans ce cas précis de systèmes où les agents n'ont pas les capacités individuelles d'organiser de manière intelligente leur groupe, sa capacité d'*auto-organisation*.

Le concept d'auto-organisation fait référence en biologie ou en chimie à la capacité que possède une population d'entités simples (particules, cellules, molécules) de se structurer spatio-temporellement [Dumouchel et Dupuy 1983]. Que signifie se structurer pour un groupe d'agents? Selon [Varela 1983], c'est que ce groupe acquière une autonomie, à la fois par rapport à ses constituants, et par rapport au substrat physique (donc pour nous, à l'environnement)⁷. La structure d'un groupe d'agents est donc un lieu autonome de transformation, dont la transformation s'exprime comme les interactions entre les agents qui la composent [Ferber 1989]. Puisque cette structure se "détache" du niveau micro et de l'environnement, et que de plus elle se déploie dans l'espace et dans le temps, nous pouvons donc supposer qu'elle peut être observée, dans son évolution, au niveau macro. Pour comprendre le

⁶ Nous nous limiterons dans notre travail à deux niveaux. Si cette option est suivie par la plupart des auteurs ayant travaillé la question de l'émergence (e.g. [Atlan 1972; Varela 1983]), un certain nombre de travaux récents mettent en avant la nécessité de considérer trois niveaux. Un premier niveau "cellulaire" dont la structure émergente constitue le second niveau, celui des agents, ces agents étant observés depuis le troisième niveau, qui est celui de l'observateur. Cependant, outre le fait qu'il n'y a pas de raisons dans ces conditions de ne pas rajouter un quatrième niveau "atomique", voire un cinquième niveau "sub-atomique", nous pensons que les mécanismes ou les processus d'émergence entre niveaux sont largement équivalents.

⁷ Une autre manière de la caractériser, formulée par [Atlan 1972], est que cette structuration fait diminuer la quantité d'information nécessaire à décrire le système tout entier.

processus d'émergence d'une fonctionnalité, il nous suffit ainsi d'observer la structuration du groupe d'agents qui l'effectue, car il est clair qu'il existe une causalité circulaire entre l'accomplissement de la tâche et la structuration de la société [Dumouchel et Dupuy 1983]. Le problème, bien entendu, est que cette auto-organisation de la société d'agents ne peut elle-même être comprise que comme une donnée *émergente* de l'ensemble des interactions qu'ont entre eux les agents, puisque, si nous revenons au début de cette section, elle constitue bien un effet global non déductible des causes locales. Il est de plus clair qu'il existe une causalité circulaire entre l'accomplissement de la tâche désirée (l'émergence de la fonctionnalité) et la structuration de la société (l'émergence d'une organisation) si l'accomplissement de cette tâche n'en est pas instantané. Programmer un système multi-agents réactif n'est donc pas aussi simple qu'il y paraît puisqu'il n'existe aucune prescription sur la manière de programmer les agents en vue d'obtenir, ou bien une organisation particulière de la population, ou bien l'accomplissement d'une tâche. Et il faut bien reconnaître que les travaux théoriques sur l'émergence ou l'auto-organisation, inspirés des mathématiques du chaos [Perez 1989], de l'étude des transitions de phase [Huberman et Hogg 1987] ou de celle des structures dissipatives [Prigogine 1982] ne sont pas encore en mesure d'apporter des solutions méthodologiques pratiques permettant de le faire. Avant d'aborder la voie pragmatique que nous proposons, il nous reste cependant à vérifier que ce que nous cherchons n'existe pas déjà dans d'autres domaines de l'informatique qui manipulent des concepts équivalents.

1.3.3. Connexionisme, automates cellulaires

Bien que la problématique générale soit similaire, puisque les aspects d'auto-organisation et d'émergence d'états globaux y sont également cruciaux [Deffuant 1992], un système multi-agents réactif diffère fondamentalement d'un réseau connexionniste [Corsi 1990; Fogelman 1985] sur trois points:

- Le premier tient à la plus grande amplitude autorisée dans la définition du comportement des entités de base, qui ne se limite pas à un simple traitement mathématique des informations perçues, ni à la simple propagation de valeurs numériques.
- Le second réside dans l'extrême dynamisme des relations pouvant exister entre les agents, qui ne sont pas liés entre eux par des liens fixes, et qui peuvent même ne jamais interagir directement [Drogoul et al. 1992].
- Le troisième, enfin, tient dans la définition explicite d'un environnement perceptible de manière locale par tous les agents qui, de par sa configuration, va souvent influencer fortement sur la configuration de l'organisation [Tyrrell et Mayhew 1991].

Les seuls ponts existant à l'heure actuelle, par le biais de la Vie Artificielle, entre les deux domaines vont dans le sens d'une programmation d'agents réactifs à base de réseaux de neurones, ce qui permet d'envisager des processus d'apprentissage sub-symboliques au niveau des agents [Atlan et Meyer 1991; Beer 1990; Holland et Snaith 1991], mais il semble plus délicat d'envisager la réutilisation de certains résultats sur l'émergence au sein des réseaux pour l'auto-organisation de nos populations d'agents, tant le fossé qui les sépare est encore grand. Les dynamiques d'auto-organisation obtenues par les méthodes connexionnistes sont en effet extrêmement contraintes par les postulats qui les fondent et que nous venons d'énumérer [Perez 1989]. Par contraste, la différence entre les systèmes multi-agents réactifs et les automates cellulaires [Von Neumann 1966] tend à s'estomper, en raison, sans doute, de la complexification progressive d'une partie de ces derniers. Il n'est maintenant plus rare de lire des travaux qui nous apparaissent typiques de l'approche réactive que nous proposons, et dont les auteurs disent s'inspirer des automates cellulaires (e. g., [Nowak et Latané 1992; Theraulaz 1991]). Mais la connaissance de la dynamique de ces automates, appliqués par exemple à la dynamique des fluides

[Perez 1989], reste encore pour une large part empirique et donc difficilement réutilisables, du moins dans l'immédiat.

I.4. La Nécessité d'une "Démarche constructiviste"

Qui Scribit, Bis Legit (Proverbe Latin)

I.4.1. Observation et construction

Nous sommes partout et tout le temps confrontés à des systèmes auto-organisés dans notre vie quotidienne. Qui n'a jamais observé de fourmis traverser son living-room ? Qui n'a jamais discuté avec un agrégat de cellules ayant forme humaine ? D'où vient alors que nous ne puissions comprendre et reproduire ces formes d'organisation dans nos mondes artificiels ? Tout simplement du fait que cette forme d'observation est *passive* car intégrée à notre système sensoriel et représentationnel. Nous ne discutons jamais avec un agrégat de cellules, nous discutons avec une *personne*. Nous ne regardons que rarement *la* fourmi, mais nous notons avec inquiétude la présence d'une *file* de fourmis. En un sens, cette démarche est naturelle et se ramène à celle de l'observateur d'un système multi-agent réactif. Ce qu'il note, ce sont les organisations et les fonctionnalités émergentes qu'il pourra observer - et utiliser.

Mais qu'en est-il du concepteur ? Il lui faut apprendre à dépasser cette vue holiste du monde pour, à l'instar du biologiste ou de l'éthologiste, découvrir les parties qui formeront le tout et la manière dont elles s'organisent. Et, en l'absence de toute théorie ou de toute méthodologie lui permettant de spécifier ces parties en fonction du tout qu'il souhaite obtenir, la seule démarche sensée consiste à utiliser une approche *constructiviste*⁸, démarche dont [Resnick 1992], après [Papert 1981; Piaget 1967], démontre la puissance pédagogique et inductive, même si elle n'est pas formelle, où «*la connaissance est processus avant d'être résultat*» [Le Moigne 1991]. Si nous nous replaçons dans le cadre qui est le nôtre, la construction d'un système auto-organisé permet ainsi d'obtenir une fonctionnalité donnée, mais nous permet surtout de construire notre compréhension du phénomène d'auto-organisation à l'œuvre, par l'interaction que nous menons en tant qu'observateur avec ce système. Il y a ainsi co-évolution du sujet (nous, en tant que concepteur/observateur) et de l'objet (le phénomène d'auto-organisation). La démarche *constructiviste* est une démarche empirique qui, selon [Resnick 1992], n'atteint toute sa puissance inductive que quand elle s'applique à la re-construction d'un objet préexistant, approche que nous illustrons avec le proverbe latin se trouvant en tête de cette section, qui signifie que pour analyser et mieux comprendre un texte, l'écrire (donc le reconstruire) revient à le lire deux fois.

I.4.2. Démarche suivie

Dans cette perspective, pour analyser et mieux comprendre les phénomènes d'auto-organisation, nous nous proposons de reconstruire des phénomènes existants, en utilisant bien entendu les outils dont nous souhaitons nous servir pour concevoir des applications, c'est-à-dire des systèmes multi-agents réactifs. Et une reconstruction, en informatique, est une *simulation*. Le travail que nous présentons dans les Chapitres suivants va donc s'articuler en deux axes, présentés de manière indépendante pour les besoins de l'écriture mais profondément liés et interagissant quant à la problématique.

Le premier axe, composé des Chapitres II à V va en quelque sorte récapituler la genèse de notre compréhension des systèmes auto-organisés en présentant une application des systèmes multi-agents

⁸ Nous différencions ici la démarche pédagogique constructiviste, propre à S. Papert et M. Resnick, de la philosophie constructiviste définie par J. Piaget.

réactifs à la simulation d'un phénomène naturel, en l'occurrence l'auto-organisation d'une société de fourmis. Nous montrerons à cette occasion le large champ d'applications que constitue la simulation de tels phénomènes pour une approche multi-agents.

Le second axe, qui commence au Chapitre VI et se termine au Chapitre XI, va développer dans le domaine de l'artificiel, et plus précisément dans le domaine de la résolution de problèmes, cette connaissance, sous la forme d'une description des systèmes multi-agents que nous avons utilisé pour résoudre des problèmes d'abord proches de ceux résolus par les fourmis, puis autonomes par rapport à la simulation réalisée. Nous tenterons parallèlement de définir de manière formelle la notion de Résolution Collective de Problèmes.

Il serait inutile de rendre compte d'une expérience constructiviste sans essayer de la faire partager, car la connaissance que constituent les processus de construction employés dans notre travail, indépendamment de celle que fournissent les résultats, serait ainsi perdue. Cependant, faire revivre sur le papier cette connaissance par essence dynamique n'est pas chose facile. Nous nous y sommes essayé, en employant un ton souvent plus didactique qu'académique, et nous espérons que de l'interaction entre le lecteur et cette thèse jailliront des idées encore plus audacieuses et ambitieuses que celles que nous défendons ici.



CHAPITRE II

LA SIMULATION MULTI-AGENTS

Avant-Propos

Suite à la définition générale des systèmes réactifs fournie dans le Chapitre précédent, et en raison de l'intérêt méthodologique que nous lui prêtons dans le cadre de notre méthode constructiviste, ce Chapitre se propose d'aborder la problématique de la modélisation et de la simulation multi-agents. Après une introduction aux principes généraux des méthodes classiques de simulation et aux problèmes d'application qu'elles peuvent rencontrer dans les sciences de l'homme et de la vie, nous montrerons comment une approche multi-agent peut remédier à cet état de fait, en proposant une nouvelle manière de concevoir à la fois modèles et théories de simulation. Cette présentation sera suivie d'une description des projets de simulation multi-agents existants et de leurs orientations.

II.1. La Simulation

II.1.1. Principes généraux

On nomme simulation la démarche scientifique qui consiste à réaliser une reproduction artificielle, appelée **modèle**, d'un phénomène réel que l'on désire étudier, à observer le comportement de cette reproduction lorsqu'on en fait varier certains paramètres, et à en induire ce qui se passerait dans la réalité sous l'influence de variations analogues. La démarche de simulation passe donc par trois étapes distinctes: l'étape de modélisation, qui consiste à construire le modèle du phénomène à étudier, l'étape d'expérimentation, qui consiste à soumettre ce modèle à un certain type de variations, et l'étape de validation, qui consiste à confronter les données expérimentales obtenues avec le modèle à la réalité.

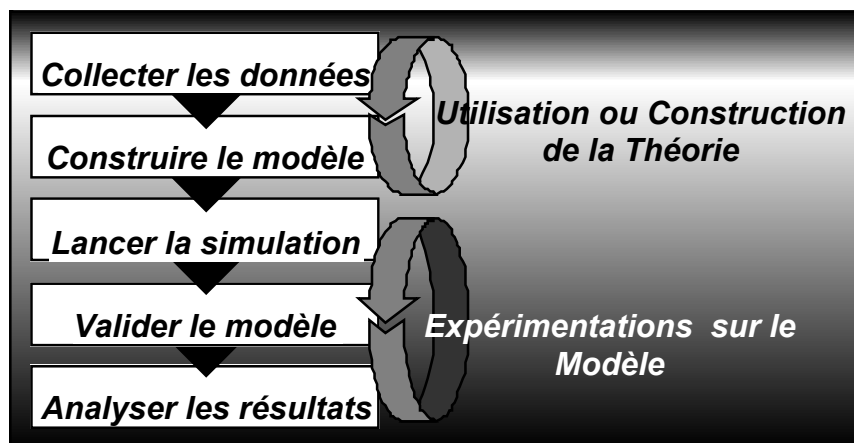


Figure I.1 - Les étapes du processus de simulation.

La construction d'un modèle se fonde toujours sur une **théorie**, c'est à dire, en simulation classique, une description abstraite de certains aspects du phénomène modélisé en termes de concepts ou de variables, et de relations ou de lois. Nous insistons sur ce point car c'est précisément cette théorie de modélisation qui différencie les différentes démarches de simulation. En effet, l'étape

d'expérimentation est le plus souvent dépendante de la manière dont le modèle a été construit, donc de la théorie, et l'étape de validation peut être soit dépendante de cette théorie (voir Section II.1.2.), soit indépendante, ce qui fait qu'aucune des deux ne peut servir à discriminer les méthodes de simulation. Construire une nouvelle méthode de simulation consiste donc à utiliser, voire inventer, une théorie de modélisation et c'est cette partie-là que nous développerons au cours de ce chapitre pour montrer l'apport des systèmes de modélisation multi-agents.

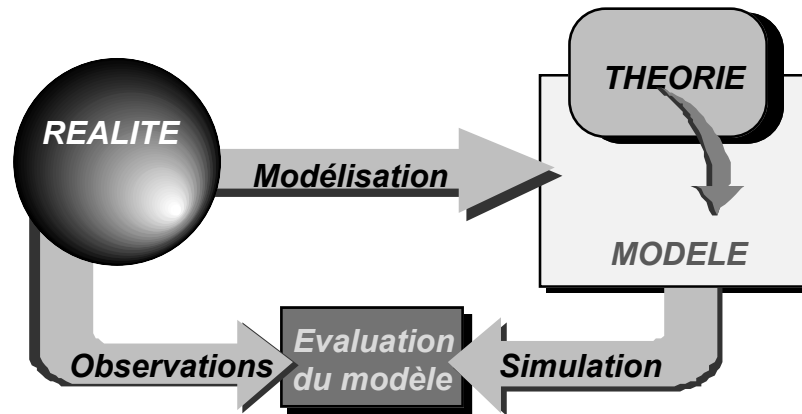


Figure I.2 - Intervention de la théorie dans la construction d'une simulation. A noter que la théorie retenue peut être partie prenante de l'étape de validation du modèle.

On a recours aux techniques de simulation essentiellement dans deux types de situation. Le premier type se définit par l'impossibilité de recourir à l'expérimentation directe, en raison, par exemple, de considérations éthiques ou d'impossibilités techniques. Un candidat aux élections n'a ainsi jamais la possibilité de mener de front dix campagnes différentes, de même qu'il n'est pas envisageable de construire dix autoroutes pour n'en garder qu'une ou de vérifier la portance des ailes d'un avion lors de son premier vol. Dans ces trois cas, la simulation permet de s'assurer *a priori* qu'une action dans le monde réel aura les conséquences que l'on souhaite qu'elle ait, en simulant le comportement des électeurs face à plusieurs campagnes, en calculant les effets de différents tracés d'autoroute sur un environnement simulé, ou en testant les réactions d'une maquette de l'avion dans une soufflerie. Le but de la simulation dans ces conditions est donc opératoire. Le second type de situation est celui où l'on ne dispose pas de bases théoriques solides sur un phénomène donné et où l'on cherche à élaborer une théorie, par voie de simulation, qui permette de rendre compte des données d'observation. Il s'agit là d'une utilisation *a posteriori* du processus de simulation dans le but de formaliser, souvent de manière mathématique, un ensemble de propriétés du phénomène réel.

II.1.2. Simulations analogique et stochastique

Bien que le nombre de méthodes de simulation soit relativement important (voir par exemple [Mosekilde 1991; Pavé 1993]), il est possible de distinguer parmi les méthodes classiques deux approches générales difficilement réductibles mais, comme nous le verrons par la suite, potentiellement complémentaires dans un cadre multi-agents. Quand on possède une bonne connaissance, en termes de théorie ou de pratique, du phénomène à simuler, il est en effet généralement possible soit de construire un mécanisme physique ou une maquette dont le fonctionnement présente une analogie avec le déroulement du phénomène, soit d'élaborer un modèle abstrait programmable sur ordinateur.

L'approche **analogique** de la simulation est la première, chronologiquement, à avoir été utilisée. De manière simplifiée, on peut dire qu'elle consiste en une reconstruction du phénomène à simuler, reconstruction qui permettra, au moment de l'expérimentation, de s'affranchir de certains problèmes

qui conduisaient à des impossibilités techniques dans cette même réalité. Il s'agit donc là de simuler un phénomène réel par un autre phénomène réel. La forme la plus rudimentaire de simulation analogique est bien entendu l'expérimentation sur maquette: maquette d'avions en soufflerie, maquette de digue ou de barrage, de navires, où l'analogie entre le phénomène réel et la maquette est évidente, à quelques détails près, comme l'échelle et parfois les matériaux employés.

On peut placer également dans cette catégorie les automates réalisés au XVIIe et XVIIIe siècles pour simuler certaines fonctions de l'homme et appuyer ainsi la doctrine cartésienne, les animaux cybernétiques de Norbert Wiener [Wiener 1948], les expériences chimiques ou physiques servant à simuler des phénomènes naturels comme les volcans ou les ouragans, etc.

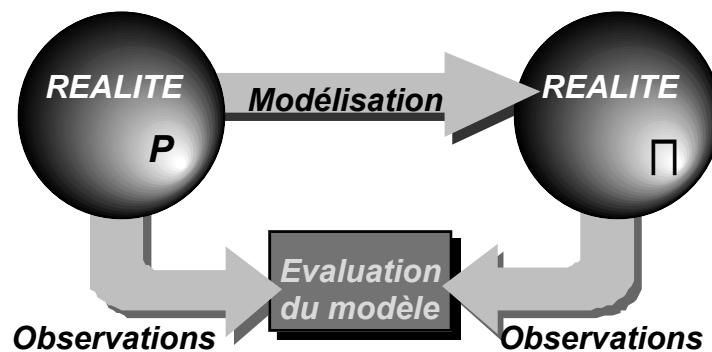


Figure I.3 - Principes de la simulation analogique. Le phénomène réel P se trouve modélisé par un autre phénomène du monde réel Π dont le fonctionnement présente une analogie avec celui de P. L'évaluation du modèle et l'obtention des résultats nécessitent une observation de chacun des deux phénomènes.

La simulation analogique est un exercice délicat et passablement difficile dès lors que l'on cherche à simuler des domaines où subsistent de nombreuses inconnues. Elle nécessite du chercheur qu'il connaisse parfaitement la nature et la forme des phénomènes à modéliser, afin de pouvoir les simuler par des dispositifs physiques analogues. Bien adaptée, dans l'esprit, à la simulation de systèmes complexes dont les éléments agissent et interagissent simultanément, elle s'avère, dans la pratique, peu employée en dehors de certains secteurs très particuliers comme l'aéronautique.

L'approche **stochastique** (ou numérique) de la simulation a pris son essor quand sont apparus des ordinateurs numériques capables de calculs mathématiques suffisamment puissants [Novels et Hackwell 1991], c'est-à-dire dans le cours des années 1950. Un modèle stochastique est une description de la réalité sous la forme d'un ensemble de paramètres numériques et d'un ensemble des relations mathématiques qui décrivent la manière dont certains de ces paramètres, appelés causes, agissent sur d'autres, appelés effets.

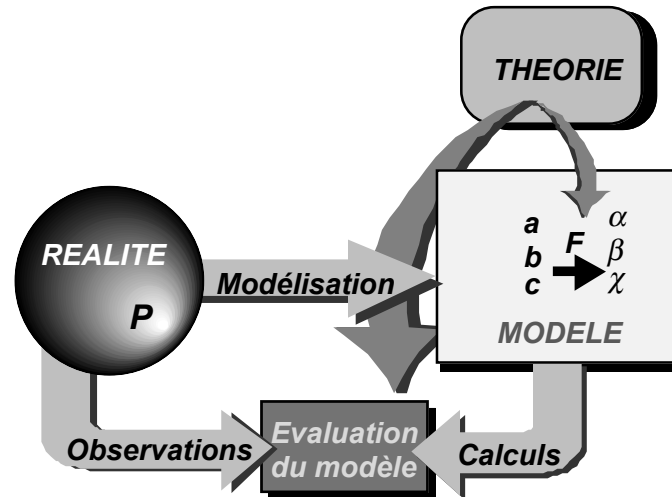


Figure I.4 - Principes de la simulation stochastique. Le phénomène réel se trouve modélisé sous la forme d'un certain nombre de variables-causes (a,b,c) et de variables-effets (α,β,χ), mises en relation par un ensemble de fonctions mathématiques F.

Le succès de cette approche s'explique d'une part par la possibilité qu'elle procure de prendre en compte les phénomènes aléatoires (par la méthode dite de *Monte-Carlo*, ou par des méthodes probabilistes semblables) et d'autre part par une maniabilité et une généralité bien plus importantes que celles offertes par l'approche analogique. Un des exemples les plus connus (et les plus anciens) de modélisation mathématique est celui de Lotka et Volterra [Volterra 1926] qui exprime par les équations suivantes (Formule II.1) les taux de croissance respectifs d'une population de prédateurs et d'une population de proies partageant un même environnement:

$$\frac{dN_1}{dt} = r_1 N_1 - P N_1 N_2 \quad \frac{dN_2}{dt} = a P N_1 N_2 - d_2 N_2$$

P est le coefficient de prédation, N_1 et N_2 les effectifs des deux populations, a l'efficacité avec laquelle les prédateurs convertissent la nourriture en descendance, r_1 le taux de fécondité des proies et d_2 le taux de mortalité des prédateurs.

Ce système est bien sûr simplifié, puisqu'il ne prend pas en compte, par exemple, les données climatiques ou saisonnières, qui peuvent intervenir sur les deux populations aussi bien au niveau démographique qu'au niveau comportemental, mais il permet de présenter de manière formelle un phénomène non-formalisé en lui-même et, du fait de sa grande abstraction, se présente comme potentiellement applicable à l'ensemble des situations analogues, puisqu'il ne fait aucune hypothèse sur les espèces d'animaux mis en jeu (voir également [Hogeweg et Hesper 1981], et leur étude sur les interactions entre deux populations de prédateurs qui chassent les mêmes proies, ainsi que leur discussion sur la validité du modèle de Lotka-Volterra). Ces deux raisons expliquent en partie l'engouement dont a bénéficié la simulation stochastique auprès des sciences dites "non-exactes", c'est-à-dire les sciences de l'homme et de la vie. De nombreux chercheurs y ont vu, et continuent à y voir, un moyen de rendre plus rigoureuses les théories de leur domaine de recherche en tentant de découvrir, grâce à la méthode par essais et erreurs que leur offre la simulation, certaines lois universelles propres au vivant ou au social. D'une certaine manière, et nous allons voir par la suite les difficultés que cela a posé, la simulation numérique stochastique est devenue l'outil des chercheurs qui perpétuent, dans leurs domaines respectifs, le projet positiviste et holiste d'Auguste Comte pour les sciences sociales. Il est cependant certain que l'usage de la simulation stochastique dans des domaines très formalisés, et donc naturellement candidats à la modélisation nécessaire, tels la mécanique, la physique ou la chimie,

a permis des avancées théoriques considérables par la valeur heuristique qu'elle apportait aux chercheurs dans la conceptualisation de phénomènes difficilement appréhendables de manière analytique; il en a été ainsi de la théorie du chaos [Gleick 1989] ou de celle des systèmes dissipatifs [Prigogine 1982]. Mais, comme nous allons le voir, son apport en tant qu'outil conceptuel dans les sciences humaines et naturelles s'est avéré décevant, en-dehors de quelques domaines restreints.

II.1.3. Application aux sciences de l'homme et de la vie

"Les circonstances font tout autant les hommes que les hommes font les circonstances" ([Marx et Engels 1968], p. 59)

A certains égards, le point de vue de la simulation stochastique sur le phénomène qu'elle doit simuler apparaît comme un point de vue *holiste* ou *macroscopique*, en ce qu'elle tente de faire apparaître des tendances, des rythmes ou des cycles dans le phénomène en question ou encore de déterminer l'influence sur ce phénomène de telle variable ou ensemble de variables macroscopiques (comme le taux de mortalité ou l'effectif d'une population), sans se préoccuper des médiations microscopiques (comme les comportements individuels, les interactions entre éléments du système) responsables de ces relations. Bien que cette approche méthodologique représente un aspect essentiel de la recherche, aussi bien en sociologie, en biologie qu'en économie, il est fondamental de ne pas perdre de vue qu'une très importante partie de cette même recherche se caractérise par des travaux relevant d'une perspective que l'on pourrait qualifier d'*actionniste* ou d'*individualiste*, lesquels se soucient d'analyser les causes microscopiques des relations observées au niveau global. Nous pouvons ainsi situer les recherches effectuées en éthologie, microéconomie ou sociologie actionniste, pour ne citer que quelques domaines, comme partie prenante de ce cadre conceptuel. De manière générale, nous pouvons même ajouter que cette conception de la réalité progresse d'une manière significative, dans la mesure où, dans chacun de ces domaines, l'on passe graduellement de la question *comment un phénomène se développe-t-il* à la question *pourquoi se produit-il* (voir par exemple [Lomborg 1992] et son analyse de l'évolution des sciences sociales, pp. 2-28). Si la simulation stochastique a pu servir d'outil conceptuel à la sociologie descriptive héritière de Durkheim, ou à l'étude des phénomènes non-linéaires en écologie (comme par exemple au sein du domaine défini par la *Behavioural Ecology* [Krebs et Davies 1978]), que peuvent en attendre les tenants d'une méthodologie *actionniste*? Sans trop entrer dans le détail, puisque ce n'est pas le but de cette thèse, nous allons parcourir quelques-uns des problèmes qui ont incité de nombreux chercheurs à demander ou proposer de nouveaux outils de simulation.

II.1.3.1. LES RELATIONS MICRO/MACRO

Afin d'illustrer ce problème, nous considérerons le cas d'une branche de la sociologie généralement désignée sous le nom d'*individualisme méthodologique* ou *sociologie actionniste* [Boudon 1992]. Née au début du XXe siècle des travaux de Weber [Weber 1964], elle représente aujourd'hui l'une des écoles de pensée les plus importantes et les plus fécondes en sociologie, par exemple dans la sociologie des organisations [Crozier et Friedberg 1977; Friedberg 1993; Simon 1991]. Son paradigme fondamental consiste à poser le fait que *«tout phénomène social, quel qu'il soit, est toujours le résultat d'actions, d'attitudes, de croyances, et généralement de comportements individuels»* [Boudon 1992]. Quels outils de simulation peuvent être adaptés à cette démarche de pensée? Il est en effet difficile, sinon impossible, en simulation stochastique, de mettre en relation des causes et des effets définis à des niveaux différents. Ainsi, l'étude de l'évolution du taux de fécondité au sein d'une population ne pourra faire intervenir que des paramètres relatifs à l'ensemble de la population, comme l'accroissement général du niveau de vie, la diffusion des moyens contraceptifs, etc. En aucun cas ne peuvent être pris en compte les prises de décision individuelles dont l'agrégation produit le taux de fécondité global

[Entwistle 1991]. Et la réponse au pourquoi d'une situation particulière ne pourra, au mieux, qu'indiquer un ensemble de corrélations entre des paramètres indépendants de même niveau, sans répondre à la question posée par la sociologie actionniste, qui tente de rechercher la causalité du phénomène dans le comportement des individus⁹.

II.1.3.2. COMPLEXITE DU PHENOMENE ET REALISME DES PARAMETRES

Prendre en compte dans un modèle stochastique la complexité de certains phénomènes amène souvent à définir des paramètres dont le rapport à la réalité n'est pas vraiment évident. Par exemple, dans les équations de la Formule II.1, le paramètre *a*, qui indique l'efficacité avec laquelle les prédateurs convertissent les prises en descendants, ne modélise pas, et c'est le moins que l'on puisse dire, la situation de manière très réaliste. La descendance d'un groupe d'animaux est en effet non seulement le résultat du succès de ce groupe dans son comportement de chasse, mais également et certainement autant le résultat d'un ensemble de comportements complexes qui mettent en jeu bien d'autres caractéristiques des individus. Que l'on songe simplement aux processus de recherche du partenaire ou de construction de l'abri ou du nid, et l'on voit déjà que le simple fait d'obtenir une descendance sera le résultat d'une longue séquence de comportements individuels ou sociaux, d'interactions avec les autres individus et d'actions sur l'environnement. Que signifie donc la variable *a* dans ces conditions pour un éthologiste? De manière similaire, et bien que ce soit dans un autre domaine, l'on ne peut que tomber d'accord avec [Lomborg 1992], dans sa critique du modèle économique du Club de Rome, dont seulement 1% des paramètres qu'il nécessitait pouvaient être estimés empiriquement!¹⁰

II.1.3.3. PRISE EN COMPTE DU COMPORTEMENT

Les méthodes numériques s'avèrent incapables, en général, de représenter correctement les actions, c'est à dire les activités individuelles qui ont pour résultat une modification du monde. Au mieux peuvent-elles prendre en compte les conséquences de l'ensemble des actions individuelles si cet ensemble agit effectivement de manière quantitative sur certains paramètres globaux du système considéré. Mais encore une fois, une simulation de ce type ne permettra jamais, par exemple, d'expliquer le *pourquoi* de certains comportements collectifs qui résultent de l'agrégation d'actions individuelles. On ne peut ainsi pas expliquer de manière macro-économique certains phénomènes de paniques bancaires ou de krach boursier qui sont dus à des comportements individuels de prudence ou de mimétisme. Et, par voie de conséquence, une simulation macroscopique des phénomènes économiques ne pourra *prévoir* ce type de phénomènes, de même qu'elle aura sans doute du mal à simplement les *expliquer* (voir à ce sujet l'étude réalisée par l'OCDE, citée dans [Wolfe 1989], qui compare un certain nombre de modèles économiques et analyse leurs erreurs de prédiction).

II.1.3.4. PARAMETRES QUANTITATIFS

Enfin, et c'est sans doute ce point qui résume le mieux la difficulté pour certains domaines de recherche de se satisfaire de la simulation stochastique, les paramètres considérés par ce type de

⁹«L'analyse des organisations et des actions organisées ne peut donc faire l'impasse sur les acteurs. Les espaces d'actions se composent d'acteurs qui pensent, même s'ils n'ont pas toutes les données; qui ont des intentions, même s'ils ne parviennent pas toujours, loin de là, à leurs fins; qui sont capables de choix (...) et qui peuvent s'ajuster intelligemment à une situation (...) et déployer leur action en conséquence» ([Friedberg 1993], p. 197).

¹⁰Une projection réalisée avec ce modèle sur l'évolution mondiale entre 1860 et 1900, après injection des paramètres relatifs à l'année 1860, a fait apparaître une baisse de la population mondiale de 2,6 milliards d'individus! ([Lomborg 1992]).

simulation ne peuvent être que des paramètres numériques, c'est à dire des données quantitatives. Il est en effet clair que la plupart des interactions entre individus réels, que ce soit en sociologie ou en éthologie, impliquent tant de non-linéarité qu'elles ne sont pas analysables avec les outils mathématiques dont nous disposons à l'heure actuelle. Dans ce cas, et si nous faisons l'hypothèse que ces paramètres ont pu être définis de manière relativement claire, que pourra-t-on en faire ? Il sera toujours possible de procéder à des simplifications, notamment dans les équations non-linéaires mises en oeuvre, mais le danger est grand de simplifier arbitrairement sur la seule base de propriétés mathématiques, car toute simplification augmentera le découplage entre le modèle et la réalité, sans garantie aucune que ce dernier reste toujours fidèle. On en prendra comme exemple la critique adressée au modèle de Lotka et Volterra par [Carbiener 1991], pour qui ce n'est pas seulement *« par le biais des ressources nutritionnelles, tel l'équilibre prédateur-proie qui relève pour une bonne part de la fable »* que la stabilité des populations vivant au sein d'un même écosystème est assurée, mais par un jeu beaucoup plus complexe de régulations qui font intervenir les besoins territoriaux, les habitudes comportementales et les modes de reproduction des différentes espèces. A trop simplifier les modèles mathématiques de phénomènes complexes, on prend le risque de transformer un modèle théorique en une simple abstraction conceptuelle, plaisante à manipuler mais dont la fécondité risque d'être faible.

II.1.3.5. QUELLE METHODOLOGIE EMPLOYER?

La définition d'une méthodologie permettant de pallier ces difficultés va donc faire intervenir une réflexion importante sur la notion même de modèle. Si l'on résume les quatre points précédemment cités, il apparaît clairement que le processus de modélisation lui-même n'est pas en cause, mais que c'est la théorie permettant d'aboutir à un modèle qui pose problème. Toutes les sciences concernées possèdent en effet un certain nombre de théories, plus ou moins formalisées, ou un certain nombre de paradigmes concernant leur objet qui leur permettent de manipuler des abstractions sans avoir à se référer constamment à des exemples empiriques. Quand ces théories sont exprimées sous la forme de relations mathématiques, l'usage de modèles numériques se révèle adéquat. C'est le cas, par exemple, de la sociologie descriptive. Quand elles ne sont pas exprimées sous une forme mathématique, ou quand elles rentrent mal dans un formalisme holiste, les chercheurs se trouvent confrontés à la nécessité de reformuler leurs théories dans un cadre qui n'arrivera jamais, pour toutes les raisons que nous avons évoquées, à exprimer pleinement la richesse de leurs observations. Pour pallier ce blocage réel dans les domaines sus-cités, il nous faut donc concevoir une méthodologie de modélisation et de simulation basée sur les points suivants:

- **Respect de la théorie du domaine.** Il est en effet essentiel pour les chercheurs de pouvoir exprimer leur modèle théorique sous une forme computationnelle, car les simplifications éventuelles que nécessitera le modèle informatique pourront dans ce cas-là être isomorphes aux simplifications effectuées par la théorie dans le domaine concerné.
- **Prise en compte des comportements individuels.** La possibilité de modéliser et de simuler les comportements individuels apparaît absolument essentielle dans bon nombre de domaines. On a en effet du mal à imaginer une démarche sociologique basée sur l'individualisme méthodologique qui ne prendrait pas en compte le concept individuel de rationalité limitée, de même qu'une simulation éthologique qui ne se baserait pas sur le comportement des animaux qu'elle modélise...
- **Prise en compte des actions et des interactions.** Enfin, offrir la possibilité de représenter les actions des individus sur leur environnement immédiat, de même que les interactions directes ou indirectes entre les individus.

La méthodologie multi-agents que nous proposons offre, si elle ne permet pas de résoudre intégralement tous les problèmes évoqués, un cadre conceptuel potentiellement applicable aux domaines des sciences de l'homme et de la nature. Le chapitre suivant se propose d'en faire un court exposé, qui sera suivi de la présentation du noyau d'EthoModélisation, qui constitue une application de cette nouvelle théorie de modélisation à l'éthologie.

II.2. La Simulation Multi-Agents

II.2.1. Principes Généraux

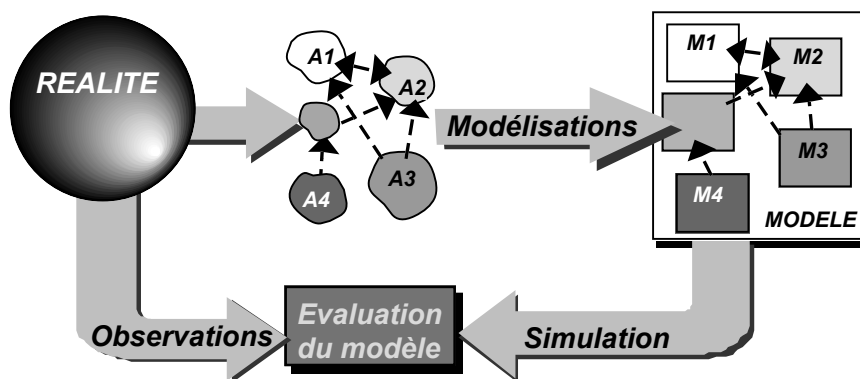


Figure 1.5 - Principe de la simulation multi-agents. Le phénomène réel est décomposé en un ensemble d'éléments qui agissent ou interagissent. Chacun de ces éléments est modélisé par un agent, et le modèle général est la résultante des interactions entre ces agents.

Comme il a été dit dans le chapitre II.1.1, la conception d'une nouvelle méthodologie de simulation se base sur la définition d'une théorie de modélisation. La modélisation d'un phénomène dans une perspective multi-agents se traduit par :

- **Une décomposition du phénomène** en un ensemble d'éléments discrets autonomes dont les interactions reproduisent le phénomène. Cette phase est semblable au processus de *réification* intervenant dans les méthodes de conception orientées objet. Il est à noter que ce préliminaire nécessite une vision déjà distribuée du phénomène à modéliser (à moins que, comme nous le faisons remarquer plus loin, l'un des objets de la simulation soit justement de déterminer les éléments ou les niveaux pertinents à étudier).
- **La modélisation de chacun de ces éléments par un agent.** Dans cette phase intervient un nécessaire choix quant à la théorie à employer pour définir les connaissances de l'agent, ses capacités fonctionnelles, ses comportements et les modes d'interaction qu'il adoptera à l'encontre des autres agents. On se reportera utilement au Chapitre I et aux modèles d'agents qui y sont présentés pour comprendre que ce choix peut être délicat. C'est au cours de cette étape que le modèle théorique de comportement individuel proposé par le domaine de recherche dans lequel s'effectue la modélisation est traduit en un modèle informatique.
- **La définition de l'espace** dans lequel évoluent ces agents et des lois qui le gouvernent. On appellera cet espace l'*environnement* des agents. Sa définition permettant d'affiner la description des actions possibles des agents, ainsi que celle de leurs moyens de communication, on définira souvent agents et environnement de manière concomitante.

- **La définition des objets inertes.** Seront considérés comme objets inertes les agents du système qui ne sont dotés d'aucune capacité d'action ni de communication.

Cette définition informelle, que nous compléterons lorsque nous aborderons la partie consacrée à la modélisation en éthologie, permet d'emblée de formuler un certain nombre de remarques préliminaires quant à l'utilisation potentielle de la simulation multi-agents comme outil conceptuel¹¹:

- L'IAD, en tant que méthodologie de simulation, n'a pas besoin de faire appel à une granularité donnée d'agents. Ainsi, contrairement à la simulation stochastique, elle n'attribue pas un comportement et une unité au phénomène dans son entier (ce qui correspondrait à la granularité maximale). Et, contrairement aux tendances néo-connexionnistes, elle n'a pas non plus besoin de se limiter à des composants aveugles capables des actions les plus élémentaires. Elle se donne les moyens de penser les situations intermédiaires, et c'est là que réside sa potentialité [Lestel, Grison et Drogoul 1993]. Elle n'éprouve nullement le besoin de se placer d'emblée à un niveau opératoire donné, mais elle se laisse au contraire la possibilité de passer d'un niveau à l'autre, au gré du contexte (voir par exemple [Bousquet et al. 1992] pour une excellente illustration de la coexistence d'agents de granularité variable au sein d'une même simulation).
- De cette manière, à partir d'une modélisation multi-agents, les chercheurs peuvent devenir maîtres des acteurs à prendre en compte dans la conceptualisation du phénomène et ainsi jouer avec le statut des objets pertinents pour un processus. De fait, comme le formule [Lapierre 1992] à propos de l'analyse systémique, «(...) *il faut s'arrêter au niveau d'analyse qui convient à la nature de l'objet qu'on a choisi d'étudier et aux questions que l'on s'est posées sur cet objet*». De plus, du fait de sa prise en compte des phénomènes interactionnels, la modélisation IAD offre la possibilité de s'intéresser au moins autant aux agencements qu'aux agents. Ainsi, des domaines comme la biologie ou la sociologie peuvent désormais apprendre à travailler avec l'idée que le statut des objets ne leur est pas d'emblée évident, et qu'atteindre les plus pertinents pour une problématique fixée est loin d'aller de soi. D'autant plus que leur objet n'est le plus souvent pas un *objet* donné (l'être humain, la structure, la cellule, etc....) mais des agencements d'objets dans des situations précises, niveaux intermédiaires d'organisation entre l'agent et le phénomène [Lestel, Grison et Drogoul 1993]. Ainsi, il est possible d'imaginer par exemple la simulation d'un même phénomène sociologique qui prenne successivement comme agents les individus, les structures et les organisations (voir la discussion dans [Rieu 1991] sur ces trois niveaux possibles de représentation) et qui compare l'adéquation de ces trois modèles à la réalité.
- Une approche multi-agent permet de représenter les systèmes complexes, complexes au sens de [Kolasa et Pickett 1991], c'est à dire des phénomènes dont le ou les modes d'organisation résultent d'interactions entre composants de bas niveau, dans la mesure où un modèle multi-agents ne présuppose aucun niveau donné d'organisation. Elle offre ainsi une aussi grande représentativité que la simulation stochastique munie d'outils mathématiques tels que la théorie du chaos [May 1976] ou la théorie des catastrophes et permet de plus ([Villa 1992], pour une illustration en biologie) de représenter des phénomènes complexes difficiles, voire impossibles, à modéliser en utilisant ces outils.

¹¹Cette partie doit beaucoup aux échanges d'idées que j'ai pu avoir avec D. Lestel et B. Grison, échanges concrétisés par un article [Lestel, Grison et Drogoul 1993]. Je les remercie ici de m'avoir autorisé à en utiliser certains passages.

II.2.2. Travaux existants

Bien évidemment, nous retrouvons dans les applications de simulation multi-agents existantes la dichotomie déjà mentionnée pour les systèmes multi-agents en général, entre les systèmes basés sur des agents réactifs et des systèmes basés sur les agents cognitifs. Ces derniers, qui d'une certaine manière apportent dans la simulation la vision anthropomorphe de l'Intelligence Artificielle¹², ont un champ d'application qui réside essentiellement dans les simulations réalisées en psychologie cognitive et sociale [Castelfranchi et Conte 1992] ou en sociologie actionniste appliquée à l'anthropologie [Doran et al. 1990; Doran, Palmer et Gilbert 1992], avec l'important projet EOS dont le but à long terme est de reconstituer les modes d'organisation des chasseurs/collecteurs du paléolithique afin de comprendre le passage progressif du groupe à la société. Les autres travaux existants consistent essentiellement en simulations prédictives conçus comme des aides à la décision [Kuhn, Müller et Müller 1993]. Le domaine qui apparaît cependant comme le plus fécond à l'heure actuelle est celui utilisant les agents réactifs¹³, qui prend le contre-pied de cette approche en tentant de simuler des comportements collectifs de haut niveau grâce à des interactions d'agents les plus simples possibles. Le nombre de travaux publiés commence à être important, aussi bien en sociologie (mécanismes d'influence sociale chez [Nowak et Latané 1992] ou [Lomborg 1992]), en biologie (simulation du processus d'hématopoïèse [Tabourdeau et Ferber 1993]), que dans d'autres domaines comme la simulation de trafic [Benz 1993] ou la démographie [Bura et al. 1993]. Avec le développement de la Vie Artificielle, le domaine qui a le plus bénéficié de ce type de simulations est l'éthologie, au point que la Vie Artificielle elle-même a pu être qualifiée d'"éthologie synthétique" [Mac Lennan 1991]. Un panorama des travaux de simulation multi-agents concernant l'éthologie des insectes sociaux est présenté Section IV.1.5. Cette approche réactive semble promise à un bel avenir en biologie, car, se basant expressément sur une métaphore biologique de l'intelligence (ou du comportement intelligent), elle se donne donc la possibilité de simuler des phénomènes ou des processus biologiques jusque-là difficilement accessibles aux outils théoriques de l'I.A.D. cognitive (et, à fortiori, de l'I.A.). Le système que nous présentons au Chapitre suivant en est une bonne illustration.



¹² Domaine dont la position vis-à-vis de la simulation ne laisse pas d'être ambiguë. Deux courants ont en effet toujours existé au sein de l'IA [Pomian 1993], le premier postulant qu'une réalisation informatique d'un comportement intelligent a une validité psychologique, le second considérant que l'ordinateur peut au plus reproduire un comportement sans avoir de valeur explicative [Winston 1984]. Le débat sur le fait de savoir si l'IA n'est qu'une science de la *simulation* du comportement humain est loin d'être clos. Et nombreux sont ceux qui, en psychologie, considèrent qu'elle n'est que ça (voir [Bonnet, Ghiglione et Richard 1989]).

¹³ Il s'agit ici, pour la plupart des travaux cités, d'une interprétation de notre part, les auteurs classant rarement eux-mêmes leurs systèmes dans la catégorie des agents réactifs, soit qu'ils ne connaissent pas son existence, soit qu'ils ne jugent pas utile de le mentionner.

CHAPITRE III

L'ETHOMODELISATION: UNE APPROCHE MULTI-AGENT REACTIVE DE LA SIMULATION EN ETHOLOGIE

Avant-Propos

Ce Chapitre se veut la jonction des deux Chapitres précédents. Nous y décrivons en effet un système de modélisation multi-agents réactif dont le domaine d'application originel est l'éthologie. Cette présentation va ainsi nous permettre de voir comment le modèle générique d'agent réactif présenté au Chapitre I peut être instancié en empruntant les principes de modélisation multi-agents que nous avons défini au Chapitre II. Après une brève introduction à l'éthologie en général, et à l'intérêt que possède, pour nous comme pour les éthologues, le fait d'effectuer des simulations dans ce domaine, nous allons passer en revue quelques-uns des modèles de comportement les plus marquants qui ont été ou qui sont toujours utilisés en éthologie. Nous montrerons ensuite comment définir un modèle comportemental réactif à partir du modèle comportemental d'activité instinctive, et comment le système qui en a résulté, appelé EthoModélisation, a été programmé. Nous fermerons enfin ce Chapitre sur une comparaison avec certains modèles existant en robotique et en Vie Artificielle, domaines dont bon nombre de travaux apparaissent à l'heure actuelle fortement influencés par l'éthologie.

III.1. Introduction

III.1.1. Un bref historique

L'étude du comportement animal a toujours été au centre des préoccupations humaines. La première cause a sans doute été purement liée aux activités humaines; connaître le comportement des animaux à chasser, à élever ou à domestiquer permettait aux hommes d'améliorer l'efficacité avec laquelle ils exploitaient les ressources animales de leur environnement. La seconde raison tenait à l'interrogation fondamentale sur la place de l'homme dans le monde et, notamment, sur son rang au sein des espèces animales. Étudier le comportement des animaux permettait ainsi de vérifier les différences et les analogies avec les comportements humains et d'établir des hiérarchies qui, en leur temps, ont joué un rôle important dans le rapport de l'homme à la nature.

Les études se sont accumulées au cours des siècles, d'Aristote à Descartes, de Saint Thomas d'Aquin à Réaumur, en passant par tous les grands naturalistes des XVIIe et XVIIIe siècles, tels Cuvier, Lamarck, Linné ou Buffon. Des monticules d'observations détaillées ont été réalisées, des milliers d'espèces observées, disséquées, cataloguées. Cependant, avant le développement de l'éthologie au début du XXe siècle, toutes ces observations ont été plus ou moins marquées par l'une des deux tendances suivantes:

- une tendance que l'on peut qualifier d'anthropomorphique, qui compare systématiquement les comportements animaux et les comportements humains. Les objectifs des travaux s'inscrivant dans cette lignée sont soit de marquer la séparation entre l'espèce humaine et les espèces animales, soit d'affirmer la place de l'homme dans le règne animal. Parmi les premiers peuvent être cités Aristote, Saint Thomas d'Aquin, Descartes ou Cuvier, pour qui l'animal est instinctif et l'homme intelligent. Parmi les seconds, on retrouve Montaigne,

Condillac ou Auguste Comte, pour qui les notions d'instinct et d'intelligence ne sont pas si éloignées l'une de l'autre.

- une tendance "naturaliste", basée sur l'observation pure, mais dont est absent le souci d'inclure les comportements ainsi découverts dans un système explicatif global portant sur leur causalité et leur organisation. Un bon exemple de cette tendance est constitué par les travaux de l'école entomologique française, avec les études de Réaumur ou de Fabre.

L'éthologie, telle qu'elle est définie aujourd'hui, se démarque de ces deux tendances (tout en en réalisant, d'une certaine manière, la synthèse) en ce qu'elle cherche à étudier *objectivement* le comportement animal, et qu'elle tente de combiner des explications causales et fonctionnelles à ce comportement. Elle se situe donc dans la lignée naturaliste, puisqu'elle privilégie l'observation du comportement, mais elle combine cette observation avec: (1) des hypothèses en ce qui concerne les causes proches du comportement, (2) des expériences qui interrogent la fonction de ce comportement, (3) le souci de toujours distinguer le comportement inné du comportement acquis¹⁴. On peut dater l'apparition de l'éthologie moderne des travaux de Konrad Lorenz et Nikolaas Tinbergen [Lorenz 1984; Tinbergen 1971]. Cependant, cette science n'a évidemment pas été créée *ex nihilo*, et de nombreux concepts clés sont apparus dans des travaux antérieurs, dont ceux des psychologues comparatistes [Loeb 1900; Morgan 1894] ou des behavioristes [Skinner 1938; Watson 1919]. Par souci de simplification, nous nous référerons dans la suite de cette partie à ces travaux en les englobant dans les recherches spécifiques à l'éthologie.

III.1.2. Pourquoi une approche multi-agents pour l'éthologie?

Deux raisons principales nous ont motivés pour réaliser un système de simulation multi-agents appliqué à l'éthologie. La première est le développement important dont a bénéficié la "sociologie animale" ces dernières années, notamment avec l'apparition de la sociobiologie [Jaisson 1993]. La seconde est la possibilité offerte par certaines théories éthologiques de manipuler des modèles de comportement débarrassés de toute référence anthropomorphe.

III.1.2.1. LE DEVELOPPEMENT DE LA "SOCIOLOGIE ANIMALE"

On a assisté, ces vingt dernières années, à une évolution croissante du nombre d'études menées en éthologie sur les animaux vivant en groupe ou en société. Sont concernés, bien évidemment, les insectes sociaux [Corbara, Fresneau et Lachaud 1988; Deneubourg et al. 1986; Theraulaz et al. 1991], mais aussi les primates [Chauvin 1989; Thierry 1990; Vauclair 1992], les oiseaux [Tinbergen 1979] ou les grands mammifères [Armitage 1987]. Alors qu'une bonne partie de l'éthologie envisageait et envisage toujours l'étude du comportement animal dans des situations mettant seulement en jeu un sujet avec des objets ou des stimuli de son environnement (la palme du radicalisme dans ce domaine revenant sans conteste à [Skinner 1938]), ces nouvelles études se trouvent confrontées à l'intense réseau d'échanges sociaux existant au sein des sociétés animales étudiées. Dans ces conditions, l'animal ne peut plus être appréhendé seul, mais comme partie intégrante d'un système social auquel il contribue et qui, à son tour, modèle son comportement. Et, inversement, le comportement global du groupe ou de la société ne peut être compris que par l'agencement, l'organisation, la collaboration et les interactions des individus entre eux. L'éthologie sociale (appelée aussi sociologie animale) apparaît

¹⁴ «L'éthologie (...) consiste à appliquer au comportement animal toutes les interrogations et les méthodes qu'il paraît naturel d'appliquer dans toutes les autres branches de la biologie depuis les découvertes de Charles Darwin» [Lorenz 1984], page 11.

donc comme un bon champ d'expérimentation d'une méthodologie multi-agents, d'autant plus que la demande de modèles alternatifs aux modèles mathématiques comme la théorie des jeux [Meyer 1980] se fait de plus en plus forte.

III.1.2.2. DES MODELES DE COMPORTEMENT SIMPLES

La seconde raison qui a motivé notre démarche est directement liée à l'intérêt de l'objet d'étude de l'éthologie dans une perspective informatique. La sociologie animale étudie en effet des organisations d'individus capables de s'adapter à leur environnement, capables de résoudre des problèmes souvent complexes d'approvisionnement, et capables de moduler de manière souvent fine leurs réactions aux modifications de cet environnement. Or, contrairement à la sociologie humaine, qui fait *a priori* la même chose, l'éthologie n'a pas besoin de postuler au départ l'existence d'une forte rationalité dans le comportement des individus mis en jeu. Des modèles quasi-mécanistes suffisent dans bien des cas à traduire fidèlement une bonne partie des comportements effectués par les animaux (voir par exemple [Rabaud 1937] cité dans [Theraulaz 1991]). Il ne faut bien sûr pas en tirer la conséquence que l'éthologie dans son ensemble considère les animaux comme des machines, car les primatologues par exemple seraient sans doute prompts à s'indigner d'une vision aussi restrictive [Vauclair 1992]. Mais, du fait que leur objet d'étude n'est pas l'homme (du moins pour la majeure partie d'entre eux, car il existe une éthologie humaine [Chauvin 1989]), les éthologistes peuvent se débarrasser de la vision anthropomorphe qui attribue à tout acte une intention, un plan ou une stratégie et suivre ainsi le "principe d'économie" (énoncé par [Morgan 1894] et cité dans [Vauclair 1992]) qui revient à affirmer «*qu'il n'est pas nécessaire de faire appel à des structures psychologiques d'ordre supérieur (...) quand des systèmes explicatifs plus simples (réflexes, par exemple) peuvent rendre compte de façon adéquate d'un comportement donné*». Nous verrons d'ailleurs dans l'analyse des différents modèles de comportements proposés en éthologie que ceci a permis l'émergence de visions aussi radicales que celle des tropismes ou du béhaviorisme. Ce principe a bien sûr été contesté, notamment par les chercheurs en cognition animale [Roitblat 1982], mais il n'en reste pas moins important à l'heure actuelle sur le plan méthodologique en éthologie. Or, qu'est donc ce principe d'économie si ce n'est l'application à l'éthologie du principe de parcimonie proposé (Section I.2.2.) pour la définition de systèmes réactifs ? Modéliser une organisation animale avec un système multi-agents devrait donc permettre de générer des sociétés artificielles d'agents en s'affranchissant des modèles individuels de haut niveau proposés, par exemple, par la psychologie cognitive. Nous insisterons d'ailleurs sur ce point lorsque nous passerons en revue les différents modèles comportementaux proposés en éthologie, puisque, nous plaçant délibérément dans une perspective réactive, nous mettrons de côté les modèles cognitivistes ou mentalistes pour nous focaliser sur les modèles plus mécanistes.

III.2. Principes Généraux du système de simulation

III.2.1. Concepts de base

Puisque c'est de modélisation appliquée à l'éthologie qu'il s'agit, les agents considérés seront donc le ou les animaux étudiés. Il y aura ainsi isomorphisme entre les agents de la modélisation et les objets étudiés par les éthologistes. Il est important de noter ce que nous avons déjà souligné, à savoir la grande faculté que possède la simulation multi-agents de pouvoir se placer à différents niveaux de modélisation en accord avec le domaine d'application choisi. Un modèle plus réductionniste aurait sans doute été utilisé s'il s'était agi de modéliser les comportements animaux à travers le prisme de la biologie moléculaire, de même que si nous avions travaillé dans une vision écologique, nous aurions sans doute opté pour la réification d'entités plus abstraites, comme les populations (voir par exemple

[Sekine, Nakanishi et Ukita 1991] ou [Bousquet et al. 1992] qui utilisent tous deux cette technique pour modéliser des bancs de poissons). Le système de modélisation proposé se compose, à la base, d'un noyau indépendant du domaine d'application (donc, dans notre cas, de l'espèce animale choisie) qui offre un ensemble d'outils de programmation et de visualisation permettant de concevoir et de suivre le déroulement des simulations. Conceptuellement parlant, cet ensemble se décompose en trois axes principaux: le modèle minimal et abstrait de comportement des agents, le modèle d'environnement choisi pour faire évoluer ces agents et l'interface utilisateur de programmation et de suivi de simulation. Chacun de ces axes va être décrit en détail dans les parties suivantes.

III.2.2. Choisir un modèle de comportement

Une grande partie de la recherche menée en éthologie, parallèlement aux travaux d'observation et d'expérimentation, a consisté à tenter de trouver des modèles de comportement qui puissent rendre compte des différents types de comportement observés chez les animaux¹⁵. Nous ne nous attarderons pas sur les caractéristiques de ces schèmes, et le lecteur intéressé pourra utilement se reporter à [McFarland 1990] pour une définition détaillée de chacun d'entre eux. Mais nous allons par contre nous intéresser à la manière dont les éthologistes ont tenté de comprendre et de formaliser les déclenchements (ou activations) de ces comportements et passer en revue les modèles existants en mentionnant explicitement ceux qui nous ont servi de support pour la conception du système.

III.2.2.1. LES PRINCIPAUX MODELES DE COMPORTEMENT

Nous nous limiterons, dans cette partie, aux études les plus importantes réalisées depuis la fin du XIXe siècle. Cette époque a en effet été marquée par la propagation des idées de Charles Darwin, qui contribuent d'une certaine manière à ré-humaniser les animaux jusque-là séparés de l'espèce humaine par les théories cartésiennes [Richard 1975]. En postulant une continuité naturelle entre l'animal et l'Homme, la théorie darwinienne fait *de facto* de l'étude du comportement animal un outil de compréhension possible pour les fondements biologiques du comportement humain. Et inversement, elle permet de faire de la "psychologie animale" un domaine à part entière, rendant possible l'étude des homologues entre les comportements d'animaux apparentés, sur laquelle se basera plus tard l'éthologie, mais qui débouche, dans un premier temps, sur la définition de modèles de comportements "universels", censés expliquer les formes de comportements observées chez les animaux. De manière simplifiée, on peut distinguer, si l'on ne tient pas compte des variantes qu'elles ont suscitées, trois familles de modèles. Il s'agit du modèle des tropismes, du modèle béhavioriste et du modèle objectiviste. Chacun d'entre eux aura son importance dans le modèle que nous proposons.

III.2.2.2. LE MODELE DES TROPISMES

Le modèle des tropismes¹⁶, proposé par Jacques Loeb [Loeb 1900], participe à une conception selon laquelle l'environnement forcerait l'animal "à se comporter". Dans cette optique, les mouvements des animaux sont vus comme conditionnés par des attirances ou des répulsions (tropisme positif ou négatif) à l'égard de certains facteurs (lumière, humidité, etc.) de leur milieu. L'organisme est considéré comme une coquille remplie d'instincts innés, coquille balancée au gré des tropismes auxquels elle

¹⁵ Bien que ce domaine découle directement de la psychologie comparative, on regroupe actuellement toutes ces recherches sous le nom de *cognition animale* [Vauclair 1992].

¹⁶ Le terme de *taxie* est également employé.

obéit aveuglément¹⁷. Si cette conception a connu un certain succès, c'est essentiellement à son opposition radicale aux thèses dites mentalistes (qui attribuaient un peu naïvement à tous les animaux des buts et des croyances) qu'elle le doit. Pour le reste, elle est aujourd'hui tombée en désuétude. Que faut-il en retenir ? Essentiellement le fait d'avoir inclus aussi bien les facteurs biotiques que les facteurs abiotiques de l'environnement au comportement en les rendant responsables du positionnement spatial et de la forme du déplacement des animaux. On retrouvera d'ailleurs cet apport fondamental quand sera abordé le comportement cinétique de base des agents du modèle (Section III.3.3.3.).

III.2.2.3. LE MODELE BEHAVIORISTE

Le modèle béhavioriste, proposé par J.B. Watson [Watson 1919] et popularisé par B.F. Skinner [Skinner 1938] a connu un plus grand succès en raison de son approche globale du comportement. Alors que les tropismes se limitaient aux comportements cinétiques, le béhaviorisme fournit un schéma explicatif de toutes les formes de comportement. L'idée de base du béhaviorisme est de réfuter les caractères classiquement attribués aux comportements innés: universalité, apparition précoce, apparition chez les individus isolés, absence d'apprentissage [Richard 1975]. Un organisme est vu comme une boîte opaque dont le seul rôle est d'apprendre, par conditionnement, à exécuter certaines actions, ou certaines séquences d'actions, en réponse à des stimulations de son environnement. L'animal possède bien à la naissance certaines potentialités, aussi bien sensorielles que physiologiques, mais ne peut développer ces potentialités que par l'apprentissage de liaisons entre les stimuli de son environnement et ses capacités physiologiques, liaisons qui portent le nom de liens S-R (stimulus-réponse). Le béhaviorisme est longtemps resté la référence obligée en psychologie expérimentale, mais a depuis été rejeté en raison de son refus de reconnaître la distinction entre inné et acquis et surtout de son opposition à vouloir étudier le contenu de cette fameuse boîte opaque¹⁸, qui constitue aujourd'hui le thème principal des recherches en cognition. Que faut-il en retenir ? Certainement le fait d'avoir posé le problème des effets du passé de l'animal sur son comportement présent en mettant l'accent sur les conséquences des répétitions de ce comportement. Le processus de renforcement que nous proposons comme partie intégrante du modèle d'agents (Section III.3.2.4.) en est une bonne illustration.

III.2.2.4. LE MODELE A BASE D'ACTIVITES INSTINCTIVES

La troisième théorie, appelée objectiviste, sur laquelle est basée l'éthologie, a donné naissance à de nombreux modèles de comportement. Nous n'analyserons ici que celui proposé par Konrad Lorenz, puisque notre modèle est basé pour une large part sur ses concepts¹⁹. Le concept principal, celui d'activité instinctive, part du principe, énoncé par ([Richard 1975] pp. 44-45), selon lequel

(...) il est possible de reconnaître dans les comportements des animaux des actes spécifiques définissables qui paraissent posséder en propre des facteurs de causalité, qui dépendent pour leur mise en train de stimuli auxquels on peut ne prêter aucune action modulante ultérieure sur le comportement engagé; qui enfin organisent leurs éléments en séquences temporelles dont l'observateur averti peut prédire le déroulement .

¹⁷ «Il était impossible de parler du comportement sans le faire en termes de tropismes; un animal ne creusait pas, il obéissait à un géotropisme positif; il ne grimait pas sans obéir à un géotropisme négatif» ([Chauvin 1989] p. 13).

¹⁸ «c'est un peu comme si l'on voulait comprendre le fonctionnement d'un distributeur automatique de tickets sans connaître d'autres éléments que le type de pièces que l'on peut y insérer et le type de tickets que l'on peut en obtenir» ([Lorenz 1984] , p.355)

¹⁹ On trouvera une description des modèles issus de cette école de pensée dans [Lorenz 1984]. Par ailleurs, les modèles hiérarchiques comme celui de Tinbergen [Tinbergen 1971] ne seront pas répertoriés ici bien qu'ils aient joués un grand rôle dans l'explication du comportement. Leur pertinence quant à la démarche "réactive" ne nous a pas semblé convaincante (voir à ce sujet [Maes 1991c], de même qu'une intéressante revue des modèles de comportement dans [Guillot 1986])

Ce qu'il faut ainsi retenir de fondamental dans ce concept, c'est:

- qu'il présente un aspect **stéréotypé** dont la marge de variation chez les individus d'une même espèce est très faible [Ricard 1970],
- qu'il est déclenché par un **stimulus externe** spécifique, et ceci par l'intermédiaire d'un Mécanisme Inné de Déclenchement (MID)²⁰,
- que ce stimulus externe spécifique est conjugué lors du déclenchement à une "**motivation interne**" à l'animal, souvent appelé Potentiel d'Action Spécifique (PAS) [Lorenz 1984],
- qu'il se traduit en un **enchaînement d'actes élémentaires**, immuables dans leur coordination, qui sont souvent constitués de mouvements instinctifs (MI) et de tropismes²¹[Ricard 1970],
- et qu'enfin l'intensité avec laquelle se déroule cette activité dépend à la fois du niveau du PAS et de la force du stimulus déclencheur.

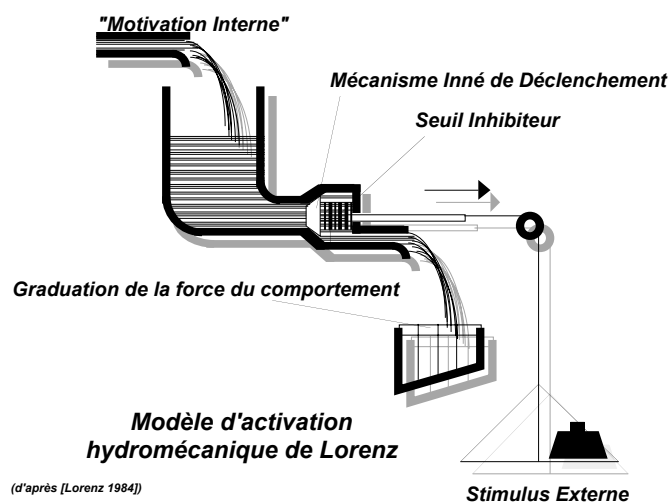


Figure I.6. - Le modèle d'activation de Lorenz. Pour chaque activité existe une "énergie" interne et un stimulus externe. Tous les deux ont le même rôle: repousser la valve qui ferme le récipient afin de déclencher le comportement.

Afin d'exprimer sous une forme compréhensible le déclenchement d'une activité spécifique, Lorenz a utilisé un modèle hydromécanique reproduit ici Figure I.6. S'il est certain que ce modèle général de comportement a depuis été revu, corrigé et critiqué même par les chercheurs qui se réclament de l'éthologie objectiviste, il n'en demeure pas moins qu'il continue à posséder, dans bien des cas, une valeur explicative essentielle. Les apports plus récents (influencés par la neurophysiologie) ont essentiellement portés sur l'assouplissement des concepts de MID et de coordination fixe, en mettant l'accent sur la variabilité individuelle et l'adaptabilité importante dont font preuve les animaux, ainsi que sur l'apport de l'apprentissage au déclenchement des comportements. On pourra se référer à [Richard 1975], qui tente de combiner l'approche béhavioriste et l'approche objectiviste, à [Vauclair 1992] qui s'intéresse plus directement à la cognition animale ou à [McFarland 1990] pour une vision générale des tendances actuelles en éthologie.

²⁰En anglais, IRM (Innate Releasing Mechanism). Le MID est un concept physiologiquement flou (difficile à situer chez l'animal) mais fonctionnellement important, qui postule chez l'animal une "connaissance innée" lui permettant de réagir sélectivement à certains stimuli.

²¹Ici, la notion de tropismes n'évoque que le phénomène d'orientation et non l'acte moteur.

III.2.2.5. LE MODELE DE COMPORTEMENT CHOISI: UN COMPROMIS

Choisir un modèle de comportement parmi le nombre important de modèles proposés par les éthologues ou les psychologues n'a pas été chose aisée. Le choix que nous avons fait repose donc sur une constatation simple: le noyau d'un système de modélisation comportementale doit s'efforcer de décrire le plus grand commun dénominateur des différents modèles de comportement existants. Il doit, tout à la fois, être synthétique, minimal et, bien sûr, évolutif. La construction du modèle s'est donc fondée sur une série d'hypothèses qui, à bien des égards, peuvent apparaître comme des truismes mais qui nous ont permis, parce qu'en éthologie elles constituent aujourd'hui la base minimale de toute description du comportement, de constituer les principes fondamentaux du noyau. Ces hypothèses sont les suivantes:

- Tous les modèles précédents s'accordent sur un point: il existe chez les animaux des comportements innés que l'on peut qualifier d'atomiques (ou primaires) dans le sens où ils sont difficilement réductibles à d'autres comportements directement observables. De manière générale, tous les actes moteurs (coordination d'un ensemble d'activités musculaires) des animaux sont compris dans cette catégorie. Ces actes sont généralement communs à tous les membres d'une même espèce, sous réserve, bien évidemment, d'anomalies physiologiques, et constituent les briques avec lesquelles des comportement de plus haut niveau peuvent être construits et décrits. Ces actes unitaires seront appelés **primitives de comportement** dans le modèle²².
- Une autre caractéristique commune aux modèles est que ces comportements atomiques peuvent être déclenchés seuls sous la forme de réflexes, ou bien faire partie de séquences comportementales. Dans un premier temps, nous nous sommes limités à la définition de séquences comportementales, sans envisager le cas des réflexes. De plus, nous avons considéré que ces séquences étaient figées, admettant ainsi que tous les comportements potentiels d'un agent étaient innés. Pour ce faire, nous avons choisi d'adopter un modèle proche de celui des activités instinctives, basées sur l'idée qu'une stimulation, aussi bien interne qu'externe, peut déclencher un ensemble figé d'actions dont l'activation individuelle ne dépend plus de conditions environnementales, mais de leur place dans la séquence. Les activités d'un agent seront donc décrites dans notre modèle par des séquences de primitives, que nous appelons **tâches**, indépendantes les unes des autres.
- Même s'il a depuis été critiqué et revu, le modèle de déclenchement des comportements établi par Lorenz [Lorenz 1984], qui se fonde sur la conjonction d'une motivation interne et d'un stimulus déclencheur présent dans l'environnement, apparaît, par sa généralité même, comme une bonne approximation de la manière dont sont activés les comportements. Il offre en effet un modèle alternatif au lien S-R des behavioristes en ce qu'il permet de prendre en compte aussi bien la "motivation" de l'agent à effectuer une tâche que la "stimulation" exercée sur l'animal par l'environnement. Le modèle de déclenchement que nous proposons se base donc sur la notion de **stimulus** (interne ou externe) qui servira de déclencheur à une tâche, la motivation interne étant représentée par le concept de **seuil inhibiteur** propre à chaque tâche²³.
- Enfin, il nous restait à inclure deux notions qui proviennent à la fois des modèles objectivistes et behavioristes: l'intensité avec laquelle une tâche est déclenchée et les effets de la répétition de cette tâche sur ses déclenchements futurs. Afin de modéliser ceci, les

²² Par référence aux primitives des langages de programmation.

²³ Pour être tout à fait exact, signalons que le seuil inhibiteur indique précisément l'inverse de la motivation de l'agent. Un seuil élevé représentera une motivation faible, d'où son nom de seuil inhibiteur.

stimuli ont une **force** et chaque tâche se voit attribuer un **poids** et un **niveau d'activation**. Le poids est la mesure de l'expérience²⁴ de l'agent par rapport à la tâche, mesure qui pourra varier positivement ou négativement, en permettant ainsi de représenter les phénomènes de renforcement ou d'habituation²⁵ observés chez les animaux. Le niveau d'activation d'une tâche résulte de la multiplication de la force du stimulus déclencheur et du poids de la tâche. Il est calculé à chaque excitation de la tâche et sert ainsi, par comparaison avec les niveaux d'activation des autres tâches, à les discriminer.

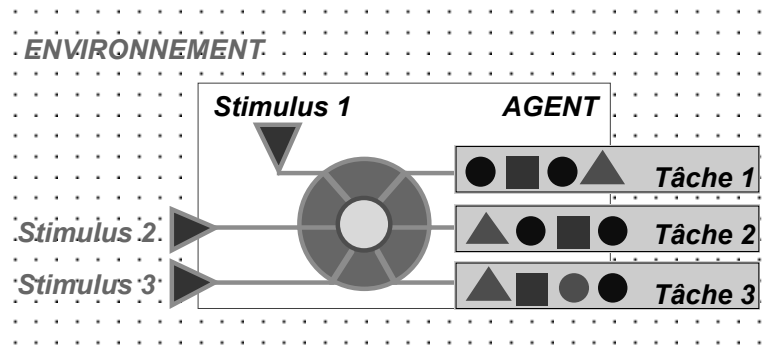


Figure I.7 - Un agent est constitué d'un ensemble de tâches, chaque tâche étant déclenchée par un stimulus particulier, qu'il soit interne ou externe. Une tâche est composée d'une séquence de primitives, représentées ici par des figures géométriques, qu'elle exécute lorsqu'elle est déclenchée.

Pour nous résumer (voir Figure I.7), nous proposons donc un modèle qui postule que le comportement d'un animal peut être caractérisé par un ensemble de *tâches indépendantes*, composées d'une séquence de comportements élémentaires moteurs appelés *primitives*. Chacune de ces tâches est exclusive et son déclenchement provient d'une stimulation externe ou interne, concrétisée par des *stimuli de force* variable, qui se combine à une motivation préexistante, exprimée sous le double aspect d'un *seuil inhibiteur* et d'un *poids*, reflet de l'expérience antérieure de l'animal dans cette tâche²⁶. Il nous reste maintenant à voir comment ces hypothèses ont été traduites en un modèle informatique. Nous allons donc examiner en détail la programmation du noyau.

III.3. La programmation du Modèle Générique

III.3.1. La programmation du noyau

Le système de modélisation qui sera utilisé dans la suite de ce chapitre a été écrit en Smalltalk-80 sur la base d'un noyau acteurs dérivé d'Actalk [Briot 1988]. Ce noyau minimal permet d'assurer un fonctionnement pseudo parallèle des agents, grâce à un système d'envoi de messages asynchrone. La différence avec Actalk réside dans l'utilisation d'un seul processus gérant une liste des acteurs en activité, là où ce dernier créait un processus par acteur. Ce choix est en partie motivé par une plus grande facilité de débogage, Smalltalk permettant difficilement de suivre des envois de messages entre

²⁴ Si deux tâches sont excitées par des stimuli de même force, et si leurs seuils sont égaux, alors c'est l'expérience de l'agent qui permettra de les discriminer, en donnant la préférence à celle qui possède le poids le plus important.

²⁵ Le renforcement est un processus qui conduit l'animal à être de plus en plus réceptif à un stimulus donné, c'est-à-dire à activer de plus en plus vite et avec une intensité croissante le comportement. L'habituation est exactement l'inverse.

²⁶ L'architecture présente de fortes analogies avec les systèmes à base de règles de production. La différence réside dans les possibilités d'interruption d'une tâche et dans l'absence de bases de faits autre que l'environnement.

processus. Les classes qui permettent d'assurer la gestion du pseudo parallélisme se nomment `EthoKernelAgent` et `EthoKernelBehavior` et jouent un rôle semblable à `KernelActor` et `KernelBehavior` dans `Actalk`²⁷. De la même manière, un nouveau gestionnaire de processus, nommé `EthoKernelScheduler` a été redéfini afin de permettre une meilleure gestion de l'interactivité avec l'utilisateur.

Le noyau du système est composé d'un ensemble de six classes qui nous permettent de réifier les concepts décrits précédemment. Nous allons ainsi passer en revue les classes `EthoBehavior`²⁸, `Stimulus`, `Task`, `Primitive`, ainsi que `Brick` et `Operator`, qui seront utilisées dans la définition des tâches.

III.3.1.1. ETHOBEHAVIOR

La classe `EthoBehavior` va servir, dans un premier temps, à décrire la structure minimale des agents d'une simulation. A la vue de ce qui a été dit précédemment, nous pouvons simplement dire qu'un agent est composé d'un ensemble de tâches et qu'il doit être capable de recevoir des stimuli provenant aussi bien de l'extérieur (son environnement, les autres agents) que de lui-même (à la manière des hormones, par exemple). Ces stimuli devant provenir de quelque part, nous supposons également que les agents auront besoin de pouvoir propager des stimuli dans l'environnement. La classe `EthoBehavior` s'écrit donc²⁹:

```
EthoKernelBehavior subclass: #EthoBehavior
  instanceVariableNames: '
    tasks
    currentTask
    environment
    internalStimuli
    personalStimuli'
  category: 'EthoKernel'
EthoBehavior class
  instanceVariableNames: 'type tasks'
```

La variable d'instance `tasks` représente l'ensemble des tâches que possède l'agent, `currentTask` sa tâche courante, `environment` l'environnement dans lequel il se trouve, `internalStimuli` l'ensemble des stimuli internes qu'il possède en propre et qu'il ne peut pas propager, `personalStimuli` l'ensemble des stimuli qu'il est capable de propager dans son environnement. La métaclasse de `EthoBehavior` enrichit la structure de toutes ses sous-classes en en précisant le `type` et les tâches par défaut (`tasks`). Lors du mécanisme d'initialisation, un individu (une instance) appartenant à cette classe sera pourvu d'une copie des tâches définies par défaut dans sa classe, copie qu'il pourra particulariser en fonction de son comportement (via les mécanismes d'habituation ou de renforcement) ou en fonction d'une instanciation particulière. La création d'une sous-classe d'`EthoBehavior` nécessitera donc d'initialiser les tâches par défaut que posséderont ses instances. Cependant, une tâche, qu'il est possible de redéfinir dans les sous-classes, est fournie de base à tous les agents; il s'agit de la tâche appelée `default`, qui spécifie le comportement de l'agent quand aucune autre tâche n'est activable.

²⁷ `EthoKernelBehavior` est la classe où est décrit le comportement de l'agent, `EthoKernelAgent` servant à assurer l'indirection asynchrone. Ainsi, toutes les classes d'agent que nous décrivons seront sous-classes directes ou indirectes de `EthoKernelBehavior`.

²⁸ Qui servira à décrire le *comportement* des agents, d'où son nom.

²⁹ La notation `Smalltalk` a ici été allégée dans un souci de simplification. Ne sont pas mentionnées les variables de classe, ni les dictionnaires partagés, sauf quand les informations qu'ils contiennent sont essentielles à la compréhension de la classe.

III.3.1.2. TASK

Les objets tâches possédés par les agents se décrivent de la manière suivante:

```

Object subclass: #Task
  instanceVariableNames: `
    name
    activationBricks
    endBrick
    activationLevel
    activityLevel
    threshold
    weight weightChange'
  category: 'EthoKernel'

```

où `name` représente le nom de la tâche³⁰, `activationBricks` la séquence de comportements élémentaires exécutées par la tâche quand celle-ci devient active, `endPrimitive` le comportement à exécuter quand la tâche est interrompue ou quand elle se termine, `threshold` le seuil inhibiteur de la tâche, `weight` son poids, c'est-à-dire son importance relative au sein du comportement global de l'agent, `weightChange` une valeur permettant de modéliser les phénomènes d'habituation et de renforcement (voir Section III.3.2.4), `activationLevel` son niveau d'activation lorsqu'elle est excitée par un stimulus, et `activityLevel` son niveau d'activité lorsqu'elle est active, qui représente en quelque sorte la motivation de l'agent à continuer cette tâche. Quand elle est déclenchée, cette motivation est au plus haut, c'est-à-dire qu'elle est égale à la valeur du niveau d'activation de la tâche. Au fur et à mesure que l'agent exécute les primitives de la tâche, cette valeur va être décrétementée. Cet indicateur permet en fait de modéliser le phénomène de *fatigue* des animaux vis-à-vis d'un comportement, phénomène qui semble indépendant de la fatigue physique de l'animal [Lorenz 1984] et qui permet à d'autres comportements de prendre le dessus au bout d'un certain temps.

III.3.1.3. BRICK, OPERATOR, PRIMITIVE

La séquence de comportements élémentaires représenté par `activationBricks` n'est pas directement décrite sous la forme d'une séquence de primitives mais sous la forme d'une séquence de ce que nous appelons des briques. Chaque brique est composée d'un opérateur et d'une primitive. Les primitives représentant des comportements moteurs atomiques, il est en effet nécessaire de contrôler leur exécution, soit en leur fournissant un argument particulier (par exemple, dans la cas d'une primitive d'interaction, savoir avec quel type d'agent ou quel agent interagir), soit en prévoyant les cas où la primitive ne fonctionne pas (par défaut, la tâche s'interrompt automatiquement), soit en leur permettant de fonctionner en boucle (par exemple, un animal fuyant un prédateur s'éloignera de lui *tant qu'il le percevra*). Les trois classes qui permettent d'implémenter cela ont comme définition:

```

Object subclass: #Brick
  instanceVariableNames: `
    operator
    primitive
    argument '
  category: 'Etho-Tools-Programming'

```

³⁰ Qui devra être semblable au nom du stimulus pouvant la déclencher.


```

Object subclass: #Operator
  instanceVariableNames: '
    name
    image
    numSucc '
  category: 'Etho-Tools-Programming'

Operator class
  instanceVariableNames: 'operators'

Object subclass: #Primitive
  instanceVariableNames: '
    name
    image
    time'
  category: 'Etho-Tools-Programming'

```

Une brique est donc composée d'un opérateur, d'une primitive et d'un argument (dépendant du contexte). Un opérateur possède un nom, une `image` (qui est sa représentation graphique, voir III.3.4.) et le nombre `numSucc` de successeurs directs qu'il a dans la séquence³¹. La classe `Operator` connaît un certain nombre d'opérateurs prédéfinis (dans la variable `operators`). Enfin, une primitive possède un nom, une image et un temps d'exécution (en nombre de cycles).

Les noms donnés aux primitives et aux opérateurs correspondent à des méthodes que doivent posséder les classes d'agents qui souhaitent utiliser ces tâches. Toutes les méthodes d'opérateurs de base sont définies dans la classe `EthoBehavior`. Les méthodes de primitives étant la plupart du temps très dépendantes du domaine d'application, elles seront programmées au niveau des classes concrètes d'agent ou au niveau de sous-classes abstraites d'`EthoBehavior` définies pour l'application. Enfin, le temps d'exécution spécifié lors de la création d'une primitive va permettre de compenser les différences entre les comportements de base décrits par ces primitives, comportements dont la durée peut varier.

Les opérateurs sont conçus pour gérer de manière transparente l'asynchronisme de fonctionnement des agents. On peut les considérer comme des macro-commandes qui encapsulent les primitives et offrent des opérations comme les boucles, sans que l'on n'ait besoin de les décrire sous la forme de continuations explicites, nécessaires dès lors que l'on programme dans un langage d'acteurs. Les méthodes d'opérateur sont au nombre de dix et permettent d'exécuter une primitive en spécifiant un argument et/ou des blocs de continuation³² dans le cas où la primitive réussit ou ne réussit pas son exécution. Voici leur liste³³:

Opérateur simple

```

exec: aPrim
  "Simple exécution d'une primitive et arrêt de la tâche"

```

Opérateurs avec continuation

```

exec: aPrim ifSucceed: aBlock
  "Exécution de la primitive. Si elle réussit, donne la main au bloc passé en argument. Sinon, arrête la tâche"

exec: aPrim then: aBlock

```

³¹ Pour le moment, les opérateurs ne peuvent avoir qu'un successeur au maximum. Cette variable n'est donc pas utilisée mais elle est prévue pour des versions ultérieures du programme où des choix pourront être codés dans les primitives.

³² On appelle bloc de continuation un texte Smalltalk qui contient une brique (opérateur + primitive).

³³ Pour des raisons évidentes de place, nous ne donnons ici que la liste des méthodes et non leur code.

```
"Même chose que exec:ifSucceed:"
```

Opérateurs simples avec argument

```
exec: aPrim with: argument  
"Même chose que exec: avec un argument"
```

Opérateurs avec argument et continuation

```
exec: aPrim with: argument ifSucceed: aBlock  
"Même chose que exec:ifSucceed: avec un argument"  
exec: aPrim with: argument then: aBlock  
"Même chose que exec:then: avec un argument"
```

Opérateurs de boucle

```
execWhileSucceed: aPrim  
"Exécute la primitive en boucle tant qu'elle réussit"  
execWhileSucceed: aPrim then: aBlock  
"Exécute la primitive en boucle tant qu'elle réussit, puis donne  
la main au bloc passé en paramètre"  
execWhileSucceed: aPrim with: argument  
"Même chose que execWhileSucceed: avec un argument"  
execWhileSucceed: aPrim with: argument then: aBlock  
"Même chose que execWhileSucceed:then: avec un argument"
```

Quant aux méthodes de primitive, qui peuvent contenir n'importe quel code Smalltalk, la seule contrainte imposée par le système est qu'elles renvoient `true` ou `false` selon que leur exécution a réussi ou non.

Ces composants (brique, opérateur, primitive) permettent de créer, de spécifier et de modifier des tâches (voir Section III.3.4., l'outil de programmation appelé `TaskBrowser`). L'exécution d'une tâche par un agent passe, quant à elle, par une forme compilée de la tâche qui sera générée si elle n'existe pas déjà. Cette forme est constituée de deux méthodes rangées dans la classe de l'agent, dont les noms respectifs sont composés du nom de la tâche suivis du suffixe `-Do` ou `-Stop` (la première correspondant au comportement d'activation, la seconde à son comportement d'interruption). Bien évidemment, cette forme compilée fait directement appel aux méthodes d'opérateurs et aux méthodes de primitives.

A titre d'exemple (voir également Figure I.9), supposons que soit définie pour une sous-classe d'`EthoBehavior` la tâche appelée `default`, dont la seule brique est composée de l'opérateur `execWhileSucceed:` et d'une primitive appelée `primMoveRandomly`³⁴, dont on supposera qu'elle a été précédemment définie sur cette classe. La tâche ainsi créée va automatiquement se compiler sous la forme des deux méthodes `defaultDo` et `defaultStop` qui s'écriront:

```
defaultDo  
"Automatically generated by the task compiler"  
    aself execWhileSucceed: #primMoveRandomly  
defaultStop  
"Automatically generated by the task compiler"
```

Dans le cas présent, la méthode de désactivation de la tâche ne fait rien. Elle doit normalement contenir une brique composée d'un opérateur simple (comme `exec:` ou `exec:with:`) et d'une primitive.

³⁴ Qui permet à l'agent de se déplacer aléatoirement. Voir Section III.3.3.3, pour la définition de cette primitive.

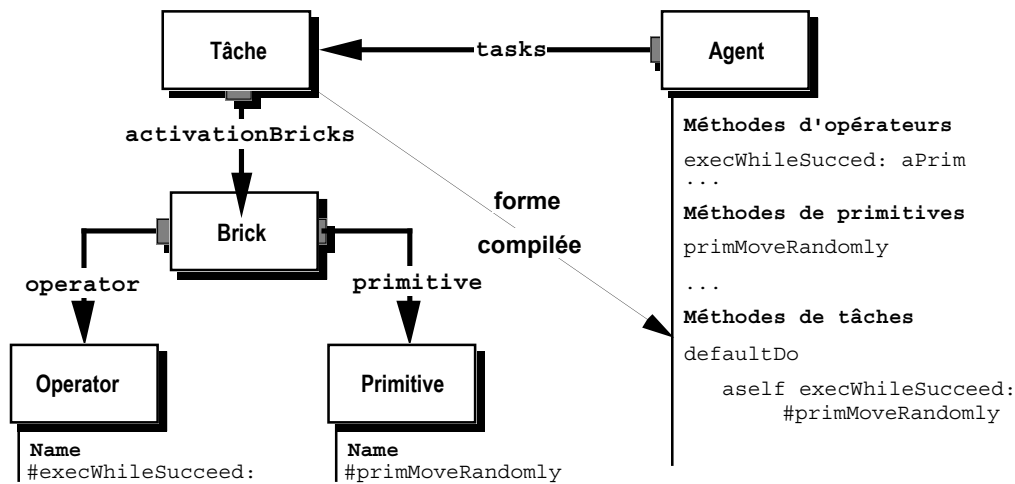


Figure 1.8 - Une tâche est créée sous la forme d'une séquence de briques. La tâche `default` décrite ici (qui indique de se déplacer aléatoirement) possède une brique dont l'opérateur s'appelle `execWhileSucceed:` et la primitive `primMoveRandomly`. Les deux font référence à deux méthodes décrites dans l'agent. La forme compilée de la tâche apparaît également dans l'agent comme une méthode, sous le nom de `defaultDo`.

Les deux objets qui se chargent de compiler la description d'une tâche en méthodes et de décompiler les méthodes pour générer des descriptions de tâche sont respectivement instances des classes `TaskCompiler` et `TaskDecompiler`. En mode de programmation, ils sont utilisés par le `TaskBrowser` (voir Section III.3.4.), mais le compilateur de tâches peut être automatiquement et dynamiquement appelé par une tâche lorsqu'elle ne trouve pas les méthodes qu'elle doit exécuter sur la classe de l'agent.

III.3.1.4. STIMULUS

La dernière composante du noyau d'Ethomodélisation est la classe `Stimulus` qui va représenter les informations externes et internes qui permettront de déclencher les tâches des agents.

```

Object subclass: #Stimulus
  instanceVariableNames: 'name intensity'
  classVariableNames: 'DefaultStimulus'
  category: 'Etho - Concepts'
  
```

Le nom des stimuli est utilisé pour déterminer les tâches qu'ils peuvent déclencher, et leur intensité pour déterminer l'urgence avec laquelle il faut les déclencher. La variable `intensity` contient soit une valeur, soit une fonction permettant de calculer cette valeur à partir d'autres éléments (voir par exemple les stimuli présentés Section IV.2.3.). La variable de classe `DefaultStimulus` contient un stimulus nommé `#default` et d'intensité minimale que les agents incluent automatiquement dans leur stimuli internes afin de pouvoir déclencher la tâche `default` quand aucun autre stimulus n'est présent.

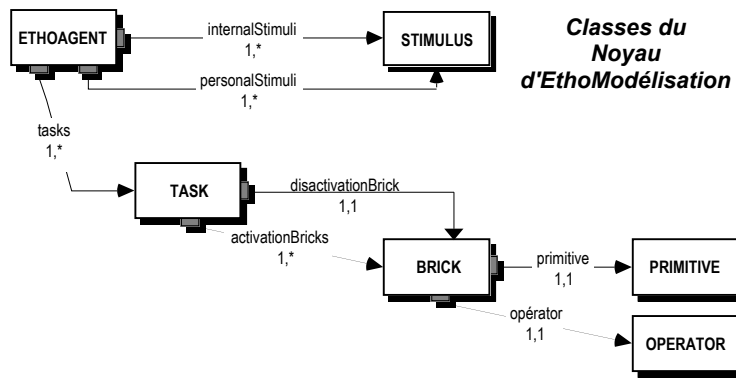


Figure 1.9 - Les classes du noyau d'éthomodélisation. Les liens symbolisent les relations qu'elles possèdent entre elles, chaque relation étant affectée d'une cardinalité minimale et d'une cardinalité maximale, * signifiant un nombre indéfini.

III.3.2. Le comportement des agents

Après avoir vu les différentes entités qui permettent de décrire le comportement d'un agent, il nous reste à étudier la manière dont l'agent détermine dynamiquement ce comportement en fonction de la situation dans laquelle il se trouve, c'est-à-dire la manière dont il choisit une tâche à exécuter. Nous ferons pour l'instant abstraction de la façon dont l'environnement fourni dans le noyau est codé, nous bornant à spécifier les messages auxquels il doit répondre pour que l'agent puisse y évoluer. Ses spécifications seront décrites dans la partie suivante. Comme les animaux de Lorenz qui possèdent un Mécanisme Inné de Déclenchement difficile à situer physiologiquement, nos agents vont posséder un *moteur de déclenchement* qui ne sera pas un objet particulier, mais un ensemble de fonctionnalités leur permettant de choisir un comportement. Comme le MID, ce moteur de déclenchement est immuable, c'est-à-dire que son fonctionnement ne peut être modifié par les activités de l'agent.

III.3.2.1. LE CYCLE DE VIE DES AGENTS

Le moteur de déclenchement peut-être vu comme composé de trois entités virtuelles indépendantes qui ont pour nom respectifs *StimuliSensor*, *TaskManager* et le *PrimitiveExecutor*. Ces trois entités fonctionnent de manière synchrone et sont donc actives à tour de rôle.

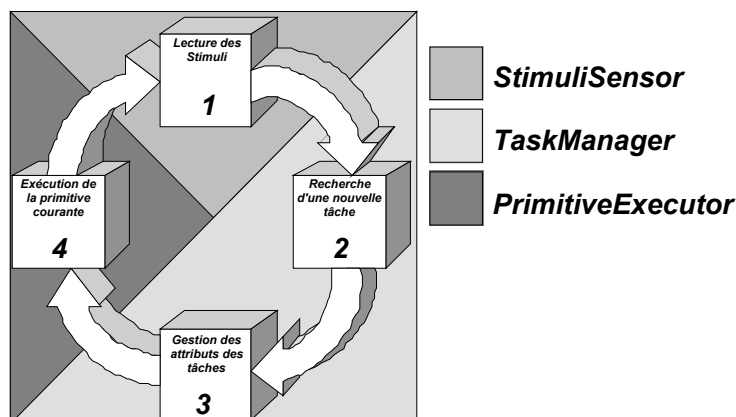


Figure 1.10 - Le cycle de vie d'un agent de base. Le *StimuliSensor* lit les stimuli puis active le *TaskManager* (1). Si une tâche se montre urgente, celui-ci interrompt la tâche en cours et déclenche la nouvelle tâche qui devient la tâche en cours (2). Les différents attributs des tâches de l'agent sont alors mis à jour (poids, seuil, etc.) (3) et la primitive courante exécutée par le *PrimitiveExecutor* (4).

Le comportement d'un agent est discrétisé par l'exécution des primitives. Autrement dit, le choix d'une tâche n'intervient que quand la primitive courante de la tâche en cours a été exécutée par le `PrimitiveExecutor`, ou bien quand le `PrimitiveExecutor` a effectué un cycle d'attente, ce qui signifie que la primitive en cours possède une durée d'exécution supérieure à 1. A ce moment, l'agent active le `StimulusSensor` qui recueille les stimuli internes et externes susceptibles d'activer une nouvelle tâche (voir Figure I.10). La main revient ensuite au `TaskManager` qui va ou non choisir de déclencher une nouvelle tâche en fonction de la situation que lui propose le `StimulusSensor`. Quel que soit sa décision, le `PrimitiveExecutor` reprend la main pour exécuter soit la primitive suivante de la précédente tâche en cours, soit la première primitive de la nouvelle tâche, soit un cycle d'attente (si la primitive précédente en spécifie un).

III.3.2.2. LE DECLENCHEMENT DES TACHES PAR LE TASKMANAGER

Le déclenchement des tâches se compose de trois étapes: (1) Recherche des tâches déclenchables; (2) Comparaison avec la tâche en cours; (3) Commutation éventuelle.

La recherche des tâches déclenchables se fait en trois temps.

<i>soit</i>	$S_A = (s_1, s_2, \dots, s_n)$	l'ensemble des stimuli lus par l'agent A ,	
	$T_A = (t_1, t_2, \dots, t_m)$	l'ensemble des tâches de l'agent A ,	
	tc_A	la tâche courante de l'agent A ,	
	na_c	le niveau d'activité de la tâche en cours,	
	$strength: S_A \rightarrow \mathfrak{R}^+$	l'application qui dénote la force d'un stimulus,	L
	$weight: T_A \rightarrow \mathfrak{R}^+$	l'application qui dénote le poids d'une tâche,	
	$threshold: T_A \rightarrow \mathfrak{R}^+$	l'application qui dénote le seuil inhibiteur d'une tâche,	

Le `TaskManager` va d'abord rechercher les tâches excitées par les stimuli recueillis. On appelle tâches excitées de A l'ensemble TE_A tel que:

$$TE_A = \{t_i \in T_A, t_i \neq tc_A, 1 \leq i \leq m \mid \exists sd_i \in S_A; name(sd_i) = name(t_i)\}$$

Cet ensemble représente toutes les tâches dont le nom correspond à celui d'un stimulus. Ce stimulus sd_i est appelé stimulus déclencheur de t_i . Cet ensemble de tâches va lui servir à détecter les tâches activables de A . On appelle tâches activables l'ensemble TA_A tel que:

$$\begin{cases} na_i = strength(sd_i) \cdot weight(t_i) \\ TA = \{t \in TE \mid na > threshold(t) \end{cases}$$

où na_i représente le niveau d'activation de t_i . Les tâches activables sont donc les tâches excitées dont le niveau d'activation est supérieur au seuil inhibiteur. Enfin, il va extraire de cet ensemble les tâches considérées comme déclenchables, qui feront partie dans l'ensemble TD_A qui est:

$$TD_A = \{t_i \in TA_A \mid na_i > na_c\}$$

c'est-à-dire toutes les tâches activables dont le niveau d'activation est supérieur au niveau d'activité de la tâche en cours. Du fait du stimulus `default` ajouté dans les stimuli recueillis et de la tâche `default` présente parmi les tâches, il y a toujours au moins une tâche activable que l'agent peut effectuer. Plusieurs cas de figures peuvent se présenter lors du calcul des tâches déclenchables:

- Si aucune n'est déclenchable, ce qui signifie que le niveau d'activité de la tâche en cours est toujours supérieur aux niveaux d'activations des tâches activables, le `TaskManager` passe la main au `PrimitiveExecutor`, après avoir décrétement le niveau d'activité de la tâche en cours et prévenu toutes les tâches activables qu'elles n'avaient pas été choisies.

- Si une tâche seulement est déclenchable, le `TaskManager` stoppe la tâche en cours et demande au `PrimitiveExecutor` d'exécuter le comportement d'interruption de cette tâche. Après quoi, il promeut la nouvelle tâche au rang de tâche en cours, initialise son niveau d'activité avec la valeur de son niveau d'activation et passe la main au `PrimitiveExecutor`.
- Si, enfin, plusieurs tâches sont considérées comme déclenchables, le `TaskManager` considère uniquement celle qui possède le plus haut niveau d'activation (ou choisit aléatoirement parmi les tâches de niveau d'activation égal) et applique la même méthode.

III.3.2.3. LES METHODES DU MOTEUR DE DECLENCHEMENT

La méthode automatiquement appelée après chaque exécution de primitive s'appelle `readEnvironment`. C'est elle qui va, dans `EthoBehavior`, séquencer l'exécution des trois composants³⁵ du moteur de déclenchement, en appelant les méthodes `collectStimuli`, `triggerTasksWith:` et `schedule`. Il faut noter que cette méthode peut être redéfinie dans les sous-classes d'`EthoBehavior`, par exemple pour calculer des paramètres ou pour faire varier les forces des stimuli internes, mais que cette surcharge doit faire appel à la méthode décrite ici.

```

readEnvironment
  self triggerTasksWith: self collectStimuli.
  self schedule

```

La méthode `collectStimuli` recueille les stimuli de l'environnement (par l'envoi du message `allStimuli`), puis leur rajoute le stimulus `default` et les stimuli internes de l'agent. Elle décrit entièrement le fonctionnement du `StimuliSensor`.

```

collectStimuli
  | c |
  c := self environment allStimuli.
  c at: #default put: Stimulus default.
  internalStimuli do: [:each | c at: each name put: each]. ^c

```

La méthode `triggerTasksWith:` recherche les tâches excitées et leur envoie le message `triggerWithStimulus:` afin qu'elles calculent leur niveau d'activation. On s'aperçoit à cette occasion que le `TaskManager` ne centralise pas l'activation des tâches mais qu'il leur transmet les informations susceptibles de les activer.

```

triggerTasksWith: stimuli
  stimuli keys do: [:n | task| task := tasks at: n ifAbsent: [nil].
  task isNil iffFalse: [task triggerWithStimulus: (stimuli at: n)]]].

```

Une fois que les stimuli ont été lus, et transmis aux tâches correspondantes, le `TaskManager` reprend la main afin de faire le tri parmi les tâches activables et de gérer la transition éventuelle d'une tâche à une autre. La méthode `schedule` regarde d'abord si il existe une tâche déclenchable. S'il n'en existe pas, elle appelle `decreaseCurrentTask`. S'il en existe une, elle appelle `executeTask:` pour effectuer la transition entre la tâche en cours et la nouvelle tâche.

```

schedule
  |activable|
  activable := self selectNextActivableTask.
  activable isNil ifTrue: [^self decreaseCurrentTask].
  self executeTask: activable.

```

³⁵ En réalité, le `PrimitiveExecutor` n'apparaît pas dans cette liste car il est entièrement pris en charge par la manière dont sont compilées les tâches.

La sélection d'une tâche déclenchable passe d'abord par la recherche des tâches activables (les tâches ayant été excitées précédemment ont calculé leur niveau d'activation et sont capables de répondre au message `isActivable`). La méthode extrait de ces tâches les tâches déclenchables par comparaison entre leur niveau d'activation et le niveau d'activité de la tâche en cours. Elle renvoie la tâche possédant le plus haut niveau d'activation, ou `nil` si aucune n'est déclenchable. Toutes les autres tâches reçoivent le message `hasNotBeenChosen`.

```
selectNextActivableTask
  | act decl newTask |
  act:=tasks values select:[:t | t isActivable].
  decl:=act
  select:[:t | t activationLevel > currentTask activityLevel].
  newTask:=decl isEmpty
    ifFalse: [(decl sortBySelector: #activationLevel) first].
  act remove: newTask. act do: [:each | each hasNotBeenChosen].
  ^newTask
```

La méthode `decreaseCurrentTask` demande simplement à la tâche en cours de décrémenter son niveau d'activité. Elle vérifie ensuite qu'elle est toujours active (c'est-à-dire que son niveau d'activité n'est pas tombé sous zéro) et, dans le cas contraire, fait appel à `purgeCurrentTask`.

```
decreaseCurrentTask
  self currentTask decrementActivityLevel.
  currentTask isActive ifFalse: [^self purgeCurrentTask].
```

La méthode `executeTask` : permet de déclencher une nouvelle tâche. Elle arrête d'abord la tâche courante, la remplace par la nouvelle tâche dont elle initialise le niveau d'activité, retire cette nouvelle tâche des tâches potentielles de l'agent, puis lui demande de s'activer en lui envoyant le message `activateOn`: (la tâche exécutera alors sa méthode d'activation sur l'agent).

```
executeTask: aTask
  self stopCurrentTask.
  tasks removeKey: aTask name.
  aTask activityLevel: aTask activationLevel.
  self currentTask: aTask.
  aTask activateOn: self.
```

Arrêter la tâche en cours nécessite simplement de la replacer dans les tâches potentielles de l'agent et d'activer son comportement d'interruption en lui envoyant le message `disactivateOn`:

```
stopCurrentTask
  currentTask isNil ifTrue: [^nil].
  tasks at: currentTask name put: currentTask.
  currentTask disactivateOn: self.
  currentTask := nil.
  self purgeMailBox
```

Enfin, la méthode `purgeCurrentTask` sert à arrêter la tâche courante et à relancer une nouvelle tâche par un simple appel à `readEnvironment`.

```
purgeCurrentTask
  self stopCurrentTask.
  self readEnvironment.
```

III.3.2.4. RENFORCEMENT, HABITUATION, ET MOTIVATION

Les mécanismes de renforcement et d'habituatation peuvent être modélisés dans la définition d'une tâche. Ces mécanismes semblent en effet jouer un rôle important dans les processus d'ontogénie individuelle, et donc dans l'adaptation d'un animal à son environnement, qu'il soit physique ou social.

Il faut noter que le terme renforcement est utilisé ici avec une signification différente de celle que lui donnent communément les psychologues, pour lesquels il est étroitement confondu avec les capacités d'apprentissage individuel (voir [McFarland 1990], notamment à l'entrée *Renforcement*). Nous ne modélisons pas de mécanismes d'apprentissage de liens entre stimuli et tâches, dans la mesure où nous les considérons comme innés, mais nous donnons la possibilité à l'agent de privilégier ou de négliger certaines tâches en fonction de son expérience passée dans leur accomplissement.

Ce renforcement, ou cette habituatation, n'est de plus absolument pas motivé par la réussite ou l'échec de la tâche³⁶, mais uniquement par sa durée relative. Qu'est ce que la durée relative ? C'est simplement le rapport entre le niveau d'activation de la tâche (qui définit au moment de son déclenchement sa durée théorique) et son niveau d'activité lorsqu'elle est arrêtée. Si la tâche s'accomplit jusqu'à son terme sans être interrompue, son niveau d'activité est normalement nul. Elle est alors récompensée au maximum de sa capacité de renforcement. Si, par contre, elle est interrompue en cours d'exécution, elle est récompensée au prorata de son niveau d'activité par rapport à son niveau d'activation originel. La capacité de renforcement d'une tâche est définie par son paramètre `weightChange` (voir Section III.3.1.2.), à qui l'on fournit une valeur positive ou négative (ordinairement 1 ou -1) si l'on souhaite qu'elle soit renforcée ou non. Ce paramètre est utilisé par la méthode `disactivateOn` : des tâches, appelée par `stopCurrentTask` (voir ci-dessus). La tâche se renforce en additionnant à son poids le nombre donné par la formule suivante:

$$i_w = \text{weightChange} * (1 - (\text{activityLevel} / \text{activationLevel}))$$

On voit bien que, si `weightChange` est positif, le poids aura tendance à croître, alors que s'il est négatif, il aura tendance à décroître. Ce qui influera sur la probabilité que la tâche soit ou non déclenchée dans le futur. Nous verrons que ce modèle simpliste de renforcement est capable de générer des spécialistes au sein d'une population d'agents (voir le projet MANTA Chapitre IV).

Nous avons vu, Section III.3.1.2., que le seuil inhibiteur d'une tâche pouvait dénoter la motivation de l'agent à l'effectuer. Dans le modèle de Konrad Lorenz, cette notion se traduit par une énergie interne qui augmente peu à peu et facilite le travail du stimulus lorsque celui-ci apparaît³⁷. Le modèle postule donc, qu'indépendamment de toute excitation, tout comportement reçoit une part plus ou moins importante de cette énergie interne, et, qu'une fois excité, la motivation qui le sous-tend a tendance à diminuer (le "réservoir à énergie" de la Figure I.6 ayant été vidé).

Ainsi, des tâches peu susceptibles d'être effectuées quand l'environnement est moyennement attractif (c'est-à-dire, pour replacer ceci dans les mots du modèle, quand le niveau d'activation d'une tâche

³⁶ Comme nos agents sont délibérément dépourvus de toute représentation interne de leur comportement, on voit mal comment ils pourraient intégrer et manipuler de telles notions.

³⁷ Konrad Lorenz a modifié son modèle en 1978 pour prendre en compte le fait que la motivation à effectuer une tâche peut être influencée par des stimuli externes ou internes qui ne déclenchent pas directement de comportement. De même, une intéressante critique du modèle de Lorenz est réalisée dans [Toates et Jensen 1991], qui présentent d'autres alternatives pour modéliser la motivation.

n'arrive pas à dépasser son seuil inhibiteur) peuvent se voir offrir une chance d'être déclenchées si nous abaissons le niveau de leur seuil inhibiteur (c'est-à-dire si nous augmentons la motivation interne les concernant). Comment, alors, décider de l'augmentation de cette motivation ?

L'hypothèse que nous faisons est que cette augmentation est liée à la chance qu'aurait cette tâche d'être déclenchée indépendamment de son seuil . Elle sera donc proportionnelle au rapport entre le niveau d'activation de la tâche lorsqu'elle est excitée et le niveau d'activité de la tâche en cours. Ce qui se traduit par la formule suivante, où d_t représente le décrétement appliqué à la valeur du seuil:

$$d_t = (\text{activationLevel} / \text{currentTask.activityLevel})$$

Ainsi, l'augmentation de la motivation est une fonction croissante du niveau d'activation de la tâche (plus elle est sollicitée, plus le seuil s'abaisse) et décroissante du niveau d'activité de la tâche en cours (plus elle est effectuée, plus les seuils des autres tâches s'abaissent)³⁸. C'est une manière de prendre en compte les sollicitations provenant de l'environnement qui ne sont pas liées à la tâche en cours. L'augmentation du seuil pour la tâches ayant été déclenchées se produit lorsqu'elles sont interrompues. L'hypothèse que nous faisons ici est que la motivation à effectuer de nouveau une tâche est inversement dépendante de la satisfaction de l'agent lors de son accomplissement. Comme nous n'avons aucun moyen de calculer la qualité de cette satisfaction, nous nous référons tout naturellement à sa durée relative, précédemment invoquée dans le mécanisme de renforcement. Nous postulons donc que plus la tâche a duré, moins la motivation à l'effectuer de nouveau augmente (donc, plus le seuil inhibiteur augmente). Cette variation est donné par la formule:

$$i_t = \min ((\text{activationLevel} / \text{activityLevel}), \text{activationLevel})$$

qui signifie que, si une tâche est allée jusqu'à son terme, son seuil sera incrémenté de son niveau d'activation lorsqu'elle a été déclenchée; sinon, il sera incrémenté du rapport entre son niveau d'activation et son niveau d'activité lorsqu'elle a été interrompue. Si l'on analyse la dynamique de la gestion des paramètres de tâches présentée ici, nous avons donc un ensemble de rétroactions positives ou négatives qui vont influencer sur le comportement de l'agent de manière significative. Les mécanismes de renforcement ou d'habituation vont intervenir dans le choix de son comportement courant par le biais de son histoire ou de son expérience, c'est-à-dire des activations précédentes de ce comportement. Les mécanismes de diminution ou augmentation de la motivation vont s'efforcer de prendre en compte les sollicitations de l'environnement quand celles-ci ne sont pas assez fortes pour déclencher de tâche. La différence entre ces deux types de rétroaction tient dans leur échelle temporelle. Les rétroactions de renforcement sont des rétroactions intervenant sur le long-terme, qui vont peu à peu modeler le comportement de l'animal en fonction des invariants de son environnement et faire en sorte que son profil comportemental épouse ces invariants. Comme le souligne [Theraulaz 1991], ces rétroactions (positives ou négatives) permettent l'accentuation d'une spécialisation comportementale et jouent en cela un rôle morphogénétique, au sens de [Morin 1977]. Les rétroactions sur la valeur du seuil sont par opposition des rétroactions de court-terme, qui vont refléter les modifications dynamiques de l'environnement et l'activité de l'agent. Les mécanismes présentés ici sont bien entendu des mécanismes par défaut qu'il est possible de modifier, voire de supprimer. L'ensemble de rétroactions qu'ils constituent étant programmé au niveau des tâches, il est ainsi possible d'imaginer des tâches faisant appel à d'autres mécanismes.

³⁸ De cette manière, si le niveau d'activation d'une tâche est supérieur au niveau d'activité de la tâche en cours mais que son seuil l'empêche de se déclencher, ce seuil sera brutalement réduit.

III.3.3. La programmation de l'environnement

Le modèle d'environnement fourni avec les autres classes du noyau représente un environnement minimal. Il sert à spécifier les méthodes que devront posséder les classes modélisant l'environnement dans les simulations. Il offre cependant quelques fonctionnalités que nous allons détailler dans cette partie. Cet environnement est considéré comme un maillage fini et discret en deux dimensions, composé de cases carrées qui représentent soit des emplacements libres, soit des obstacles. Le choix effectué a été de ne pas avoir d'objet central représentant l'environnement dans son entier, mais de le considérer comme un ensemble de cases possédant des relations de voisinage entre elles.

III.3.3.1. L'ENVIRONNEMENT COMME MEDIUM DE COMMUNICATION

La première fonctionnalité offerte par l'environnement est de pouvoir servir de médium de communication entre les agents. Les agents définis par `EthoBehavior` ne sachant que recueillir des stimuli dans l'environnement, toute communication entre eux ne pourra qu'emprunter ce canal. Un agent aura donc la possibilité de demander à son environnement de propager un certain nombre de stimuli afin qu'ils puissent être recueillis par d'autres agents. Il ne pourra cependant pas spécifier les destinataires de ces stimuli. On parle dans ce cas-là de communication non-intentionnelle. L'exemple le plus simple que l'on puisse donner de ce type de communication est celui de la communication chimique chez les insectes (voir le projet MANTA). La non-intentionnalité de ce type de communication apparaît surtout dans le fait que l'environnement tel qu'il est conçu va systématiquement chercher à propager les stimuli contenus dans la variable d'instance `personalStimuli` de chaque agent dès que cet agent apparaîtra ou à chaque fois qu'il changera de position. On peut grossièrement assimiler ce processus au fait pour un agent de se montrer, c'est à dire d'apparaître dans le champ de perception des autres agents capables de le voir. La fonctionnalité de propagation est assurée par les méthodes de deux classes: un ensemble de méthodes au niveau des cases, qui leur permettent de propager un stimulus vers leurs voisines (ce qui résultera globalement en une structure de gradient, voir [Steels 1989] ou [Shamos et Hoey 1975]); une nouvelle classe de stimuli (`PropagableStimulus`) capables d'indiquer la perte de force résultant de chacune des propagations locales (ils possèdent un décrement de force fixé au départ) et connaissant leur émetteur, ainsi que la place dont ils sont émis.

```

Model subclass: #EnvironmentalPlace
  instanceVariableNames: 'neighbours position image stimuli '
  category: 'Etho - Environment'

Stimulus subclass: #PropagableStimulus
  instanceVariableNames: 'origin decrement emitter '
  category: 'Etho - Concepts'

```

Pour une case instance d'`EnvironmentalPlace`, propager un stimulus revient à: (1) regarder si un stimulus de même nom et de même émetteur mais de force supérieure n'existe pas déjà; (2) si ce n'est pas le cas, ajouter ce stimulus au dictionnaire des stimuli qu'elle possède; (3) vérifier que le stimulus est encore propageable, c'est-à-dire que sa force est supérieure au décrement qu'il spécifie; (4) si c'est le cas, envoyer le message de propagation à ses voisines (variable `neighbours` initialisée au départ) avec comme argument un clone du stimulus précédent dont la force est amputée du décrement. La deuxième classe de places proposée dans le noyau est la classe `Obstacle`, qui va représenter les places ne pouvant pas propager de stimuli ni recevoir d'agents. Sa définition est assez simple et elle utilise le polymorphisme du modèle objet pour simplement redéfinir le message de propagation, de manière à l'accepter, mais sans rien faire.

```

EnvironmentalPlace subclass: #Obstacle
  instanceVariableNames: ''
  category: 'Etho - Environment'

```

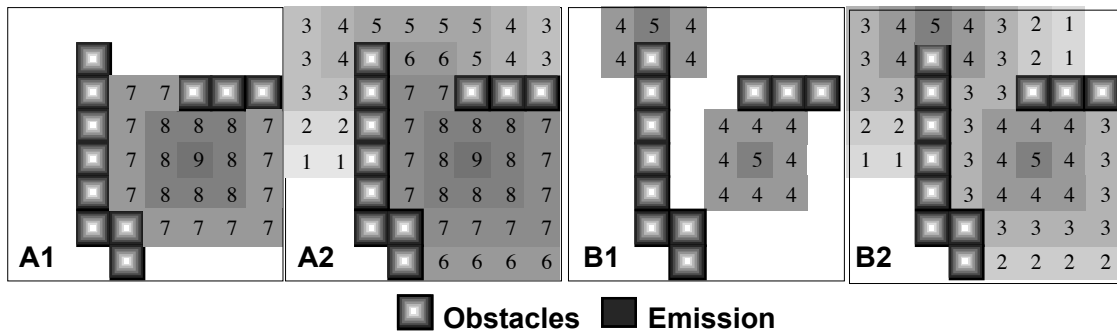


Figure 1.11 - Deux exemples de propagation. Le premier (A) provient de l'émission d'un stimulus de force 9 et de décrétement 1. Le second (B) de deux émissions simultanées de stimuli de même nom, de force 5 et de décrétement 1. Bien que les deux stimuli soient entièrement propagées jusqu'à leur extinction, n'est affichée sur les places que la force du plus grand stimulus (ce que l'agent perçoit s'il le demande à sa case).

Les places sont également capables de dépropager les stimuli ainsi propagés. Ceci est nécessaire quand l'agent disparaît, ou quand il se déplace (de manière à ce que ses gradients le suivent). La dépropagation est exactement semblable à la propagation, à la seule différence que les stimuli concernés sont retirés de chaque case au lieu d'être rajoutés. L'interface de ces cases avec les autres objets (agents ou cases) est composée de trois méthodes: `propagateStimulus:`, `depropagateStimulus:` et `findPlaceOfStrongestStimulus:`. Elles permettent respectivement de propager un stimulus, de le dépropager et de trouver la case voisine possédant, pour un nom donné de stimulus, celui dont la force est la plus grande.

III.3.3.2. L'ENVIRONNEMENT COMME ESPACE TOPOLOGIQUE

Les fonctionnalités de communication décrites sont complétées de manière naturelle par un ensemble de fonctionnalités permettant aux agents de se positionner et de se déplacer dans leur environnement. Celles-ci sont réparties entre les classes d'environnement et de nouvelles classes d'agent. Une nouvelle classe de case, appelée `AgentsPlace`, prend en charge les agents qui lui sont rajoutés ou enlevés, en effectuant automatiquement certaines opérations (propagation de leur stimuli par appel direct aux deux méthodes décrites plus haut, mise à jour de l'interface graphique, etc...).

```

EnvironmentalPlace subclass: #AgentsPlace
  instanceVariableNames: 'agents'
  category: 'Etho - Environment'

```

L'interface de ces cases avec les autres objets (dont les agents) est constituée de trois méthodes: `addAgent:`, `removeAgent:` et `whichAgentPropagates:`, qui permettent respectivement d'ajouter un agent à la case, de lui en retirer un et de récupérer la liste des agents qui propagent un stimulus donné.

Ces méthodes sont directement utilisées par la classe d'agents nommée `LocatedBehavior`, qui fédère tous les agents situés dans un environnement dérivé de celui du noyau.


```

InterfaceBehavior subclass: #LocatedBehavior
  instanceVariableNames: 'place '
  category: 'Etho - Agents'

```

Cette sous-classe indirecte³⁹ d'`EthoBehavior` va limiter l'environnement des agents à une place donnée, assurer la liaison entre eux et les instances d'`AgentsPlace` et implémenter deux primitives qui permettront aux agents de respectivement propager et dépropager leurs stimuli. Ces deux primitives se nomment⁴⁰:

```


primPropagateStimuli      "Propagation des stimuli de l'agent"
primDepropagateStimuli  "Dépropagation des stimuli de l'agent"

```

et peuvent être utilisées dans la construction des tâches des agents instances des sous-classes de `LocatedBehavior`.

III.3.3.3. SE DEPLACER DANS L'ENVIRONNEMENT

Deux autres classes d'agents ont été développées afin de tirer parti des fonctionnalités de l'environnement. La première fournit aux agents la capacité de se déplacer, la seconde la capacité d'orienter leurs déplacements en fonction de leur environnement. On retrouve là en partie les apports du modèle à base de tropismes que proposait [Loeb 1900]. Les éthologues considèrent en effet que l'activité de déplacement d'un animal est soumise à deux composantes: la composante dite *cinétique* et la composante dite *taxique* [Viaud 1951]. La composante cinétique, encore appelée *cinèse fondamentale*, consiste en un déplacement susceptible de se manifester seul dans des conditions isotropes de l'environnement vis-à-vis de tous les agents d'excitation. Elle se traduit par l'ensemble des mouvements effectués par l'animal en fonction de son état interne (donc sans interactions avec l'environnement) et est le plus souvent constituée d'un déplacement brownien. La composante taxique, quant à elle, apparaît après l'introduction d'une dissymétrie de stimuli dans l'environnement et produit une orientation de l'animal par rapport au facteur responsable de cette dissymétrie. Comme le note [Richard 1980], «*Parmi les facteurs externes jouant un rôle important dans la composante taxique, il faut insister sur deux caractéristiques du stimulus: direction et gradient. (...) Le stimulus déclenche et entretient une activité locomotrice orientée*». Ces deux composantes sont décrites dans le noyau par l'intermédiaire de deux classes abstraites, sous-classes indirectes d'`EthoBehavior`. La première, `MovingBehavior`, offre un semblant de cinèse fondamentale à tous les agents qui en héritent.

```

LocatedBehavior subclass: #MovingBehavior
  instanceVariableNames: 'angle '
  category: 'Etho - Agents'


```

Cette cinèse fondamentale est un déplacement aléatoire, restreint par la valeur de la variable d'instance `angle` qui spécifie l'angle maximal par rapport à la direction précédente que peut effectuer l'animal

³⁹ Elle est sous-classe d'`InterfaceBehavior`, elle-même sous-classe directe d'`EthoBehavior`, qui regroupe les agents possédant une interface visuelle.

⁴⁰ Dorénavant, toutes les primitives présentées seront précédées de leur représentation icônique, ceci afin de familiariser le lecteur à cette écriture et de lui permettre de lire plus facilement les tâches quand elles seront décrites. Cette représentation icônique est une image bitmap de 16 pixels sur 16.



lors d'un déplacement élémentaire (i.e. d'une case à une autre). Cette valeur est fixée arbitrairement dans `MovingBehavior` à 45° , mais peut être redéfinie dans les sous-classes concrètes au moment de l'instanciation. L'utilisation de la composante cinétique dans une tâche se fera donc par appel à la primitive:

```
|  primMoveRandomly "déplacement aléatoire d'une seule case"
```

Le déplacement proprement dit d'une case à une autre appelle les méthodes `addAgent:` et `removeAgent:` de `AgentsPlace`. Cela signifie également que toute la propagation et la dépropagation des stimuli de l'agent durant le déplacement est assurée. La deuxième composante du mouvement, la composante taxique, est elle implémentée dans la classe `SensingBehavior`, sous-classe de `MovingBehavior`, sous la forme de deux primitives.

```
| MovingBehavior subclass: #SensingBehavior
  instanceVariableNames: ''
  category: 'Etho - Agents'
```

Les primitives en question permettent de remonter (ou suivre) un gradient et de le descendre (ou de le fuir). Ces primitives fonctionnent en appelant la méthode `placeOfStrongestStimulus:` de la classe `EnvironmentalPlace` et prennent en argument soit un nom de stimulus (par exemple, `#ant`), soit un tableau composé de noms de stimuli et des poids associés (par exemple `#(#ant 1 #humidity 2)`), afin de permettre le suivi ou la fuite d'un gradient composite⁴¹.

```
|  primFollow: "Suivi d'un gradient simple ou composite"
   primFlee: "Fuite d'un gradient simple ou composite"
```

Avec cette classe, nous terminons l'étude des classes fournies dans le noyau et de leurs fonctionnalités. Il nous reste maintenant à voir quels sont les outils de programmation qui permettent de composer les tâches à partir des primitives existantes et quelle est la méthodologie à suivre pour effectuer la programmation d'une application de simulation complète.

III.3.4. Les outils de programmation

Bien que le système soit basé sur un modèle théorique relativement simple, il partage avec de nombreux autres systèmes informatiques le problème de l'acquisition de connaissances des experts (en l'occurrence, les éthologues). Etant donné qu'il est souvent difficile d'exiger de l'expert de savoir programmer dans le langage choisi (qui peut en plus changer), la méthode la plus simple est de trouver un accord sur un langage intermédiaire, souvent dépendant du domaine ou de la théorie utilisée. Les outils de programmation proposés par EMF se classent en deux catégories. Les outils de développement proprement dits, qui servent à définir la structure des classes d'agents et les primitives qui leur sont adjointes; ces outils reposent principalement sur les possibilités du langage choisi et sur le modèle opératoire fourni par le noyau. Ils en nécessitent donc une bonne connaissance. Et, deuxièmement, les outils d'extraction de connaissances qui servent à programmer le comportement des agents en donnant la possibilité à l'éthologue de contruire des tâches avec les primitives

⁴¹ Dans l'exemple, l'agent suivra les stimuli `#ant` et `#humidity` en suivant préférentiellement le second. Le calcul effectué revient à suivre un stimulus imaginaire dont la force sera égale à la force du stimulus `#ant` plus deux fois celle de `#humidity`.

disponibles, de manière à se limiter à la partie conceptuelle ou théorique du programme de simulation, sans avoir à rentrer dans les détails pratiques d'implémentation. Dans cette partie, nous nous limiterons à la seconde catégorie d'outils, la première ayant été déjà présentée dans les sections précédentes. L'outil principal de programmation des tâches s'appelle le `TaskBrowser`. Pour des raisons de facilité et de convivialité, il est basé sur des notions (simples) de programmation graphique qui permettent de manipuler les icônes de primitives et d'opérateurs. Le choix d'un outil de programmation graphique repose sur la constatation que des relations graphiques tangibles remplacent avantageusement les relations grammaticales abstraites que fournissent traditionnellement les langages (voir, par exemple, [Travers 1988]). La fenêtre de programmation (voir Figure I.13) offre à l'utilisateur différentes informations essentielles au développement des tâches, parmi lesquelles la liste des classes d'agents définies (vue `AGENTS`) dans l'application de simulation, la liste des stimuli (vue `STIMULI`) que leurs instances sont susceptibles de détecter, c'est-à-dire tous les stimuli propagés par ces instances et par les instances des autres classes, ainsi que la liste des stimuli internes que définit leur classe, la liste des primitives (vue `PRIMITIVES`) que définit ou dont hérite la classe sélectionnée, et la liste des opérateurs (vue `OPERATORS`) que définit ou dont hérite la classe sélectionnée. A priori, cette liste reprend tous les opérateurs définis dans `EthoBehavior`, mais il est possible d'en définir de nouveaux. La recherche de ces informations par le `TaskBrowser` s'effectue de la manière suivante:

- Les classes d'une application de simulation sont rangées dans une catégorie portant le nom de la simulation suffixé par `'-Agents'`. Il suffit alors de fournir au `TaskBrowser` le nom de la simulation choisie. Il est de plus possible d'incorporer à la main d'autres classes en utilisant le bouton `Add Class`.
- La liste des stimuli définis par une classe est décrite dans sa méta-classe. Il peut cependant arriver que des stimuli ne soient définis qu'au niveau de la classe. Auquel cas, le `TaskBrowser` parcourt la méthode standard d'initialisation `initializeStimuli` afin de récupérer les stimuli concernés.
- Les primitives définies par une classe ou héritées de classes supérieures doivent être rangées dans le protocole `'private-primitives'`. Le `TaskBrowser` parcourt la hiérarchie et renvoie la liste de toutes les primitives trouvées. Il est possible de rajouter à la main des primitives supplémentaires en utilisant le bouton `Add Primitive`.
- La recherche des opérateurs s'effectue de la même manière, mais dans les protocoles appelés `'private-operators'`.
- Enfin, les différentes tâches d'une classe d'agents sont décrites dans la méthode d'initialisation `initializeTasks` de sa méta-classe (d'où elles sont ensuite recopiées dans les instances). Le `TaskBrowser` parcourt alors cette liste, cherche les méthodes associées à chaque tâche (nom de la tâche suffixé par `-Do` ou `-Stop`), et fait appel au décompilateur de tâches afin de recréer, si besoin est, la tâche sous la forme de briques.

Comme il a déjà été indiqué, une primitive s'exprime dans le modèle à la fois par une méthode au sein de la classe de l'agent concerné et par la création d'une instance persistante conservée par la classe `Primitive`. Cette instance possède un nom (qui est le nom de la méthode associée) et une variable d'instance appelée `image` qui contient sa représentation icônique. Les opérateurs possèdent également cette double représentation en tant qu'instances de la classe `Operator`. Les représentations icôniques d'opérateurs sont au nombre de quatre et se différencient par deux critères: si l'opérateur possède une continuation et s'il s'agit d'une boucle (voir Figure I.12).

	<i>avec continuation</i>	<i>sans continuation</i>
<i>boucle</i>		
<i>simple</i>		

Figure I.12 - Les images des opérateurs. Les opérateurs sans continuation sont dits terminaux.

Nous pouvons donc décrire graphiquement une brique comme l'incrustation de l'image de sa primitive dans l'image de son opérateur. Ainsi, la brique qui s'écrit en Smalltalk `exec:#primPropagateStimuli` sera représentée par l'icône:



Et une tâche composée de plusieurs briques sera tout simplement représentée par une succession de ces images. La programmation d'une tâche avec le `TaskBrowser` va donc être un processus relativement intuitif. Il revient à sélectionner la classe d'agents voulue et choisir le stimulus qui déclenchera la tâche. Si la tâche existe déjà, sa représentation graphique vient s'afficher dans les deux vues prévues à cet effet, qui représentent respectivement le comportement adopté par l'agent lorsque la tâche est déclenchée et lorsqu'elle est interrompue. Les différents paramètres (poids, seuil et incrémentation du poids) susceptibles d'être modifiés sont quant à eux mis à jour dans la vue `PARAMETERS`. L'utilisateur peut alors intervenir sur chacun des composants de la tâche pris individuellement. Par exemple, modifier une brique revient à la sélectionner dans la vue `ACTIVATION`, puis à choisir dans les listes disponibles l'opérateur et/ou la primitive à modifier. Il est également possible de modifier le paramètre sur lequel s'applique la primitive (ce paramètre est un nom de stimulus ou un tableau, comme dans le cas de la primitive `primFollow`). Enfin, on peut modifier les paramètres propres à la tâche en spécifiant simplement de nouvelles valeurs.

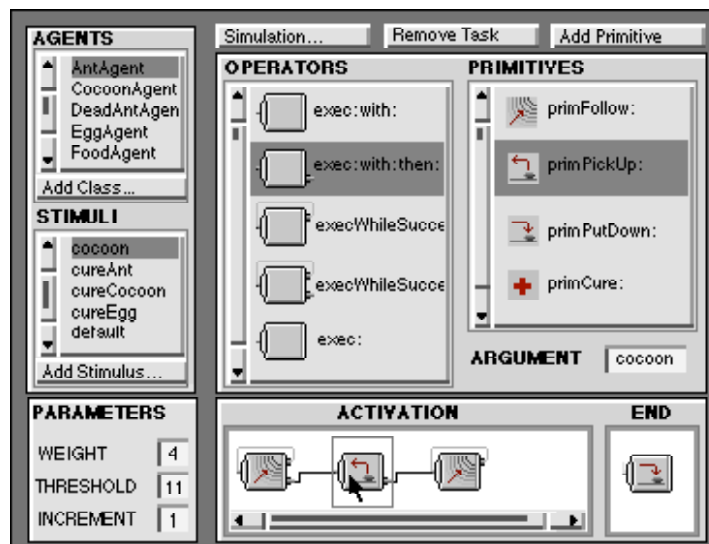


Figure I.13 - Le TaskBrowser et ses principales vues. A l'aide de cet ensemble de vues, il est possible de créer de nouvelles tâches ou de modifier des tâches existantes. L'exemple montré sur le dessin est tiré du projet MANTA. La sélection de la brique du centre (vue `ACTIVATION`) fait apparaître son opérateur, sa primitive, l'argument de sa primitive. Il est possible de les modifier individuellement.

Si la tâche n'existe pas, elle apparaîtra dans la fenêtre graphique sous la forme d'une brique indéfinie, c'est-à-dire un opérateur générique contenant une primitive inconnue (icônifiée sous la forme d'un point d'interrogation), invitant l'utilisateur à spécifier ceux qu'il souhaite définir. Dernier point, la construction d'une séquence se fait en choisissant un opérateur avec continuation, ce qui a pour effet d'obliger le système à le faire suivre de la brique indéfinie. La fin de la séquence, inversement, se marque par le choix d'un opérateur terminal. Les tâches ainsi créées ou modifiées sont ensuite compilées par le `TaskCompiler`, qui reproduit la séquence de briques sous forme de méthode. Soit par exemple la tâche décrite Figure I.13. Sa représentation graphique est:



Une fois compilée cette tâche aura la forme suivante:

```

cocoondo
  self execWhileSucceed: #primFollow: with: #cocoon then:
  [self exec: #primPickUp: with: #cocoon then:
  [self execWhileSucceed: #primFollow: with: #(cocoon 1 humidity
  1)]]]

```

On comprend en la lisant pourquoi nous proposons un outil de programmation graphique.

III.4. Méthodologie de programmation d'une simulation

III.4.1. Classes Abstraites et Classes Concrètes

La notion de classe dans le modèle recouvre deux concepts différents et partiellement indépendants, qui sont représentées par les classes dites abstraites et les classes dites concrètes. Une classe abstraite (comme `EthoBehavior`) est une classe qui ne possède pas d'instances directes ([Ferber 1990a]) et dont le rôle est simplement de fournir une description des comportements et des connaissances qui seront partagées par ses sous-classes. Elle recèle donc les *potentialités* qui seront exprimées (ou non) dans les niveaux plus bas de la hiérarchie. Inversement, une classe concrète est prévue pour être instanciée et fournit donc à ses instances des comportements et des connaissances directement utilisables dans l'application dont elles font partie. Une classe concrète est souvent une classe terminale de la hiérarchie qui représente directement un type d'agent particulier de l'application. Transposée dans le modèle de comportement choisi, cette distinction s'exprime par le fait que:

- C'est dans les classes abstraites que seront définis les primitives et/ou opérateurs et que seront décrites les connaissances (variables d'instance) qui leur sont nécessaires pour fonctionner. De la même manière tout le modèle de comportement ainsi que ses éventuelles additions (c'est à dire toute la "partie cachée" des agents) seront placés dans les classes abstraites.
- C'est dans les classes concrètes que seront définies les tâches et que seront instanciées les connaissances décrites plus haut, à partir des connaissances fournies par les experts du domaine. C'est également donc dans ces classes que seront décrits les stimuli que leurs instances propageront ou posséderont de manière interne⁴²

⁴² Certaines classes abstraites peuvent cependant, pour des raisons de commodité, définir des tâches et des stimuli abstraits. Un exemple en est la classe `MaturingBehavior`, décrite au Chapitre IV.

Créer un ensemble de classes concrètes pour une simulation consiste donc à choisir soigneusement les classes abstraites dont on souhaite hériter, et éventuellement à en créer de nouvelles si le comportement atomique souhaité n'existe pas; à créer les méthodes d'initialisation requises par ces classes abstraites (comme les méthodes d'initialisation de stimuli demandées par `EthoBehavior`) et enfin à doter ces classes de tâches.

III.4.2. Identifier les Stimuli et les Tâches

La création des tâches nécessite d'effectuer tout d'abord une recherche de tous les stimuli auxquels sont soumis les animaux réels, que ce soit des stimuli visuels, sonores, chimiques ou d'interaction. Cette liste faite, il reste maintenant à déterminer quels sont les comportements, donc les tâches, mises en oeuvre par l'animal en réponse à ces stimuli. Le point important est que les comportements doivent être définis *indépendamment les uns des autres*. Le modèle ne permet en effet pas de chaîner les tâches les unes aux autres⁴³, ni de définir une gamme de tâches possibles pour un même stimulus. Tous les comportements qui apparaissent comme chaînés doivent être considérés comme appartenant à la même tâche, de même que les différentes réponses possibles à un même stimulus doivent être différenciées en plusieurs tâches, à charge de trouver les stimuli adéquats.

Il n'existe aucune réelle heuristique pour la programmation de ces tâches, hormis la notion de *comportement consommateur*. Dans le modèle, un comportement est dit consommateur, ou une tâche dite consommatrice, si son objectif est de faire baisser le stimulus qui l'a déclenché (voir Chapitre VI le lien entre cette notion et la notion de téléonomie néguentropique). Ainsi, la tâche déclenchée par la faim de l'animal (stimulus interne), aura normalement pour objectif de faire baisser ce stimulus. Dûe à la programmation du noyau, il est même impératif que les tâches programmées soient consommatrices, dans la mesure où une tâche qui ne fait pas baisser le stimulus qui l'a déclenché a de fortes chances d'être déclenchée indéfiniment (surtout si elle est renforcée).

Une fois cette étape franchie, il reste à trouver la manière dont les stimuli sont propagés, c'est-à-dire à déterminer les agents qui les propagent; ce qui nécessite, également, de déterminer leur comportement.

Les étapes proposées ne constituent pas un processus linéaire, mais itératif, la définition des stimuli ne servant que comme point de départ arbitraire. Il est possible, et c'est ce qui a été fait dans MANTA, de partir des comportements observés plutôt que des stimuli. Quoiqu'il en soit, quand les stimuli environnementaux ont été épuisés, il peut rester un certain nombre de comportements qui n'ont pas trouvé de déclencheurs. Ces comportements vont alors être assimilés à des comportements déclenchés par des stimuli internes, qu'il faudra bien sûr décrire et ajouter aux agents. Ceci amènera, par exemple, à différencier le comportement appétitif de recherche de nourriture motivé par un stimulus interne (la faim) du comportement opportuniste de prise de nourriture motivé, lui, par un stimulus externe (l'odeur ou la vue de la nourriture).

Il reste enfin, et c'est sans doute le point le plus délicat, à paramétrer les agents en fonction des données biologiques fournies par les éthologues. Ce paramétrage inclut tout d'abord la définition de l'échelle temporelle et spatiale utilisée, qui consiste à fixer combien de temps va durer un cycle et de quelles dimensions sont les cases. Ensuite, en fonction de ces deux échelles, il faut fixer la durée (en nombre de cycles) de chaque primitive, puis la force initiale des stimuli, la valeur de leur incrément (si

⁴³ Bien qu'il soit possible, au sein d'une tâche, de propager un stimulus qui aura pour effet d'en déclencher une nouvelle. Mais il ne sera pas traité différemment des autres et il est possible que la tâche voulue ne soit pas déclenchée car d'autres sont prioritaires.

leur valeur est dynamique) ou de leur décrétement. Après quoi, restent les paramètres biologiques dépendants du domaine et le poids, seuil et incrément de chaque tâche. Le paramétrage de cette dernière valeur va être conditionné par la vision qu'ont les éthologues de l'évolution du comportement au cours du temps. Si un renforcement est observé, l'incrément sera positif; si une habitude se manifeste, il sera négatif (ce sera donc un décrétement); si, enfin, l'expérience de l'animal dans ce comportement ne semble pas influencer sur la probabilité qu'il a de le reproduire, l'incrément sera nul. Nous discutons d'ailleurs à la fin de la Deuxième Partie de la possibilité de différencier les tâches en sous-ensembles correspondant à ces trois types d'incrément (les tâches "normales" avec renforcement, les tâches "réflexes" avec habitude, et les tâches "internes" sans renforcement ni habitude).

III.5. Discussion

Nous confrontons dans cette section le modèle de comportement proposé aux critiques habituellement adressées aux modèles définis en éthologie. Puis, en tant que modèle de comportement d'agent réactif, nous le comparons avec quelques autres modèles apparus dans le domaine de la Vie Artificielle.

III.5.1. L'EthoModélisation et l'Ethologie

Pour ce qui est de l'éthologie, l'essentiel de ce que l'on peut dire est que le modèle proposé ici n'est bien sûr pas exempt de défauts et qu'il est fort probable qu'il prêtera le flanc aux mêmes critiques que celles adressées à l'encontre des approches dites "mécanistes" du comportement animal (comme celle de [Rabaud 1937]), ainsi qu'à celle de Lorenz quant au déclenchement et à la fixité des séquences comportementales. Nous en sommes conscients mais il convient de rappeler qu'il s'agit là d'un modèle *minimal* de comportement qui ne prétend pas à l'universalité. Nous pensons cependant qu'il constitue une base de travail potentielle à partir de laquelle l'on peut envisager de modéliser des comportements de plus haut niveau. Il y a d'ailleurs un parallèle intéressant à faire entre cette volonté incrémentale et les recherches actuelles menées en éthologie.

Les modèles de Tinbergen et Lorenz sont en effet toujours à la base de la plupart des modèles élaborés de nos jours. Et les modifications qui y sont apportées portent presque uniquement sur la plasticité de certains mécanismes (comme le MID) que l'on considérait comme figés et dont les observations les plus récentes tendent à montrer qu'ils sont capables d'une souplesse insoupçonnée. Ce qui était vu comme purement inné devient maintenant un mixte d'inné et d'acquis (voir par exemple [Chauvin 1989] ou [Lorenz 1984]). Mais ces nouvelles théories ne modifient pas les concepts énoncés dans ces deux modèles; elles viennent se greffer dessus, remplacer un élément trop rigide par un élément plus souple, ou compléter des mécanismes insuffisants. Un excellent exemple de cette manière de procéder, quoique extérieur à l'éthologie, peut être trouvé dans l'approche de Marvin Minsky de la psychologie humaine [Minsky 1988], qui ne fait rien d'autre que réutiliser les hiérarchies de comportement de Tinbergen en les enrichissant de manière à y inclure les capacités cognitives humaines.

Les faiblesses des modèles purement mécanistes résidaient dans le fait qu'ils déniaient à l'animal toute motivation ou inclination interne. Or, notre approche ne privilégie ni l'environnement, ni l'individu. Il est ainsi envisageable d'étendre les capacités cognitives des agents (par des mécanismes d'inhibition, ou même de planification de tâches) sans remettre en cause la notion même de tâches concurrentes. De même, la notion volontairement floue de stimulus peut être précisée pour y inclure des capacités sensorielles précises. Enfin, des pièces logicielles d'apprentissage de liens stimulus-tâches peuvent être

greffées au noyau existant sans que cela ne remette en cause son fondement⁴⁴. Il faut donc bien comprendre que le système présenté a été voulu minimal pour préserver sa généralité. L'avenir dira, bien entendu, si les bases choisies l'ont été judicieusement.

III.5.2. L'EthoModélisation comme modèle d'agent réactif

Les problématiques de l'Intelligence Artificielle Distribuée réactive ou de la Vie Artificielle (voir Chapitre I) sont bien sûr différentes de l'éthologie, dans la mesure où une part importante de la recherche dans ces deux domaines consiste à trouver des modèles de comportement qui possèdent à la fois une valeur explicative des mécanismes cognitifs du vivant et en même temps une valeur applicative. Dans cette perspective, si la plupart des chercheurs insistent sur l'intérêt essentiel de la réactivité au sens large, les différentes approches divergent sur l'aspect du *contrôle* de cette réactivité. Ce qui fait l'originalité de l'EthoModélisation par rapport à la plupart de ses *alter ego*, c'est que nous avons choisi de découpler totalement le contrôle du comportement du comportement lui-même, en faisant confiance d'une part au rôle organisateur, à court terme, de l'environnement immédiat, et au rôle régulateur, à long terme, de l'histoire de l'agent. En ce sens, notre démarche est parfaitement résumée par la phrase suivante, de [Lindauer 1986] (traduite par nos soins): «(...) *Le comportement est contrôlé par la situation. Chaque tâche effectuée génère une nouvelle situation qui est en elle-même un stimulus et un guide pour les activités futures*»⁴⁵. Concevoir des étho-agents implique donc de ne pas concevoir de système de contrôle de leur comportement⁴⁶. La raison pour laquelle nous avons choisi cette solution n'est pas fortuite. Si nous nous replaçons dans la perspective qui est la nôtre, à savoir celle énoncée Section I.3., et si nous quittons donc un instant le domaine de la simulation pour revenir à celui de la conception de systèmes multi-agents réactifs, il nous apparaît comme essentiel de rappeler le point suivant: un agent ne peut-être considéré comme adaptatif que (1) s'il est capable d'adapter son contrôle et son comportement aux situations auxquelles il fait face ou (2) si le contrôle de son comportement est fonction directe (sous une forme ou une autre) de la situation. Le premier point se situe clairement dans une perspective cognitive (voir par exemple [Collinot et Hayes-Roth 1991]), et nous ne l'aborderons pas. Mais nous allons passer en revue quelques architectures représentatives du champ réactif et analyser, du point de vue de la seconde exigence, leurs avantages et leurs inconvénients par rapport aux choix que nous avons faits.

III.5.2.1. L'ARCHITECTURE DE SUBSOMPTION

Ainsi, nous nous situons tout à fait dans la perspective de [Brooks et Connell 1986], qui décomposent le comportement global d'un robot en un ensemble de tâches parallèles ayant seulement accès aux capteurs et aux effecteurs qui leur sont nécessaires (voir Figure I.14 - Schéma A). Cependant, le contrôle de l'activité du robot est réalisé par une hiérarchisation des comportements et la définition de liens d'inhibition (théoriquement des niveaux les plus hauts vers les niveaux les plus bas) par l'intermédiaire d'une architecture dite de *subsumption* [Brooks 1990]. La priorité entre comportements doit donc être définie une fois pour toutes par le concepteur, ce qui pose le problème du choix de cette priorité.

⁴⁴ Nous discutons d'ailleurs, à la fin du chapitre, des moyens d'y inclure des mécanismes d'apprentissage tout en préservant sa minimalité.

⁴⁵ «(...) *the situation itself is the controlling agency. Each accomplished job creates a new situation which in itself is a stimulus and guide for the activities to be done next*».

⁴⁶ L'inconvénient que l'on pourrait trouver à cette approche est qu'elle nécessite donc de concevoir le comportement en fonction du contrôle imposé. Mais nous n'y voyons que des avantages. Ceci permet de restreindre le choix à effectuer au moment de coder le comportement, et surtout de se focaliser sur les différents comportements en eux-mêmes, plutôt que sur leur gestion, alors que de nombreuses approches imposent de concevoir contrôle et comportement parallèlement.

Certes, tous les robots fonctionnant sur la base de cette architecture ont fait la preuve que, pour des tâches simples, comme explorer un environnement inconnu, ou ramasser des boîtes de soda, ils étaient capables d'agir aussi bien et plus rapidement que les robots traditionnels et, de ce point de vue, Brooks a ouvert la voie que nous suivons en démontrant la crédibilité d'une approche basée sur le comportement plutôt que sur le raisonnement (voir [Jacopin 1991] pour un panorama des différents robots du MIT). Mais nous nous permettrons d'être sceptiques quant à la justesse de cette approche pour implémenter des comportements complexes, qui font intervenir, par exemple, plusieurs tâches de mêmes niveaux de priorité (une alternative est proposée par [Zeghal 1993], qui fusionne les sorties de tâches concurrentes au lieu de les inhiber, mais ceci n'est possible que quand les opérations de fusion peuvent être définies, ce qui n'est pas toujours le cas).

L'Architecture de Subsumption (R. Brooks, J. Connell)

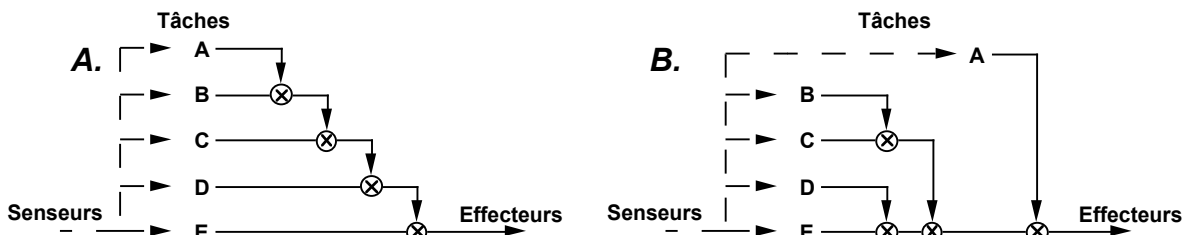


Figure I.14 - Les deux implémentations de l'architecture de subsumption. En A, l'architecture proposée par R. Brooks, qui définit un ordre hiérarchique total entre les tâches. En B, l'architecture proposée par J. Connell, qui se contente d'un ordre partiel.

Les limitations du modèle de subsumption proposé par Brooks ont bien entendu inspiré de nouvelles études, dont les plus représentatives sont certainement celles de Jonathan Connell et Patti Maes. La première, développée dans [Connell 1990], fait toujours appel aux liens d'inhibition et de suppression entre tâches, mais propose un ordre de priorité partiel, plutôt que l'ordre total défini par Brooks (voir Figure I.14 - Schéma B). Il est ainsi possible dans ce modèle d'avoir des comportements de même niveau hiérarchique, et donc des tâches concurrentes. Cependant, la gestion des conflits entre ces tâches, donc la gestion du contrôle, reste extrêmement délicate à effectuer, du fait qu'aucun mécanisme n'est prévu pour, et est entièrement laissée à la charge du concepteur, ce que Connell reconnaît d'ailleurs bien volontiers dans sa postface (*ibid.*, p. 141 à 151).

III.5.2.2. L'ARCHITECTURE ANA

Le troisième modèle de comportement reposant sur le concept de tâches est celui de Patti Maes, présenté dans [Maes 1990] et développé dans [Maes 1991a; Maes 1991b; Maes 1991c]. Ce modèle est intéressant dans la mesure où il prend en compte le facteur essentiel à nos yeux de compétition entre tâches par l'intermédiaire de niveaux d'activation différentiels. Ces niveaux d'activation résultent de la conjugaison de facteurs exogènes (stimuli) et de facteurs endogènes représentés par des motivations. Jusque-là, ce modèle est donc très proche du nôtre. La différence réside dans les liens que ces tâches ont les unes avec les autres. Trois types de liens sont possibles: les liens successeurs/prédécesseurs et les liens conflictuels. Les premiers servent à transférer une partie du niveau d'activation d'une tâche vers ses successeurs (si elle est déclenchée) ou vers son prédécesseur (si elle ne peut pas l'être). Les seconds permettent à une tâche de diminuer le niveau d'activation, proportionnellement à son propre niveau d'activation, des tâches avec lesquelles elle est considérée comme incompatible. La tâche choisie à chaque instant est bien entendu celle qui possède le plus haut niveau d'activation.

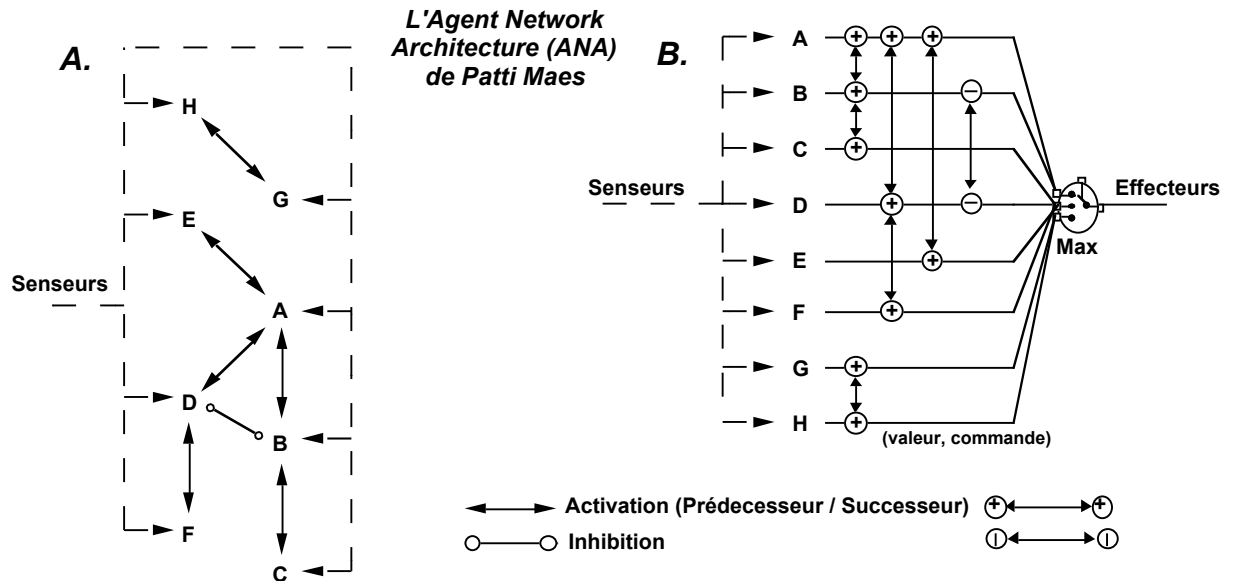


Figure I.15 - L'architecture ANA, proposée par Patti Maes. En A, la représentation qu'elle en donne, sous la forme d'un graphe. Les doubles flèches représentent les liens successeurs/prédécesseurs, par lesquels va passer le flot d'activation. En B, la reconstruction que nous en faisons, sous la forme d'un ensemble de tâches distinctes dont les sorties se renforcent ou s'inhibent.

Il nous est apparu, en étudiant ce modèle de plus près (voir Figure I.15, Schéma B.), que Patti Maes ne semble pas faire confiance aux capacités auto-organisatrices et adaptatives d'un système à base de tâches indépendantes. Appliquer le concept de liens d'activation revient exactement à réintroduire, par la petite porte, des notions de hiérarchies entre tâches qui, pour être partielles, n'en sont pas moins contraignantes. Et, alors que les conflits entre ces différentes tâches pourraient être gérés tout simplement par les situations auxquelles l'agent fait face, elle confie ce rôle au lien d'inhibition. Encore une fois, une part importante du contrôle est prise en charge lors de la conception du système.

III.5.2.3. LES AUTRES ARCHITECTURES

En réalité, le système d'EthoModélisation est assez proche de l'architecture de Uwe Schnepf, présentée dans [Schnepf 1991], également basée sur un modèle dérivé des activités instinctives. Les préoccupations de cet auteur étant d'ordre psychologique plutôt qu'éthologique, l'intérêt de ses travaux réside dans les solutions qu'il préconise pour passer progressivement et incrémentalement d'un modèle réactif à un modèle cognitif. Nul doute que si notre système devait évoluer vers des modèles d'agents plus rationnels, nous nous inspirerions de ses travaux. La plupart des systèmes basés sur le découpage du comportement en "classifieurs" sont également proches de ce que nous proposons, mais uniquement si nous faisons abstraction de la problématique évolutionniste qui les sous-tend (voir [Delaye et Ferber 1992]). Il existe enfin, dans "l'école réactive", un nombre important de travaux effectués dans le cadre du concept d'action située (voir, par exemple [Agré et Chapman 1987], [Wavish et Connah 1990], ou [Suchman 1987]). Nous comparerons ce modèle au nôtre et en analyserons les similitudes et les différences au cours du Chapitre X, consacré à la programmation de PENGI.



Deuxième Partie

Le projet **Manta**: la Simulation de l'Organisation Sociale d'une Colonie de Fourmis

"Peut-il y avoir une société de fourmis artificielle ?" ([Hofstadter 1985], p. 402)

"For example, consider modeling a colony of ants. We would provide simple specifications for the behavioral repertoires of different castes of ants, and create lots and lots of instances of each caste. (...) The behavior of the colony of automata would emerge out of the behaviors of the individual automata themselves, just as in a real colony of ants"
([Langton 1988], p. 4)

AVANT - PROPOS

Nous décrivons dans cette partie le projet MANTA⁴⁷, premier projet de simulation à avoir été programmé avec le noyau d'EthoModélisation. Après une brève introduction au monde des fourmis en général, et à l'espèce simulée dans le projet, *Ectatomma ruidum* Roger, en particulier, nous aborderons les objectifs de ce travail pluridisciplinaire, en les comparant à ceux des projets de simulation similaires développés dans d'autres laboratoires.

La décomposition d'une colonie de fourmis en agents et la définition de leurs comportements sont détaillées à partir de la Section IV.2., où nous indiquons pas à pas les différentes étapes franchies afin de bien montrer quelles ont été nos hypothèses. Cette description informatique est complétée, dans la partie suivante, par la présentation des protocoles expérimentaux sur lesquels sont basées toutes les expériences présentées.

Ces expériences portent, dans une suite logique, sur la sociogenèse, c'est-à-dire la naissance de la société, sur l'adaptabilité de cette sociogenèse à des conditions environnementales particulières, sur l'apparition progressive d'une organisation sociale dans la colonie au fur et à mesure de la naissance de nouvelles ouvrières, sur la division du travail (polyéthisme) à laquelle cette évolution aboutit quand la société devient adulte et, enfin, sur l'intérêt fonctionnel de la polygynie (situation où plusieurs reines cohabitent dans un nid). Bien entendu, chaque cas sera confronté aux données recueillies dans la nature ou en laboratoire, et analysé dans la perspective des objectifs que nous nous sommes fixés.

Nous terminerons cette partie par une courte conclusion sur l'avenir de ce projet, et sur celui de l'EthoModélisation. Nous effleurons à cette occasion quelques unes des améliorations que nous souhaitons apporter à son modèle de comportement.

⁴⁷ qui est l'acronyme de Modelling an ANTnest Activity (Modélisation de l'Activité d'une Colonie de Fourmis)

CHAPITRE IV

SIMULER DES FOURMIS

IV.1. Des Fourmis à Manta

IV.1.1. Généralités sur les fourmis⁴⁸

Le terme "fourmis" fait référence à l'ensemble des insectes membres de la famille des Formicidae, incluse dans l'ordre des Hyménoptères, qui comprend, entre autres, les abeilles et les guêpes. Les fourmis sont réparties en 11 sous-familles, 297 genres, et approximativement 10 000 espèces (8 800 répertoriées, et 20 000 probables pour [Hölldobler et Wilson 1990], 13 000 pour [Jaisson 1993]). On peut caractériser la fourmi générique, sans distinction d'espèce, par deux traits prédominants qui font d'elle un insecte à part. Premièrement, elle colonise tous les biotopes existants de la planète, à l'exception des zones glaciaires et des environnements marins (encore que certaines fourmis, voir [Hölldobler et Wilson 1990] p. 2, se montrent capables de survivre de nombreuses heures sous l'eau). Deuxièmement, tout comme le termite (qui appartient à l'ordre distinct des Isoptères), c'est un animal exclusivement social, ce qui signifie qu'il n'existe pas de fourmis solitaires. La coexistence de ces deux caractéristiques, ainsi que l'attestation de sa présence sur terre depuis au moins 100 millions d'années (fin du Crétacé inférieur), lui permet d'occuper une place prédominante à la fois parmi les insectes (les fourmis représentent environ 40% de la biomasse totale des insectes) mais également au sein d'écosystèmes où elle est particulièrement abondante (voir Figure II.1). Elle peut en effet y atteindre des densités de 120 millions d'individus par hectare et devient de fait le premier prédateur d'invertébrés ou le plus important herbivore.

Une des explications de cette réussite écologique particulièrement remarquable, tient, comme dans une certaine mesure pour l'homme, dans le phénomène de socialité. La totalité des espèces de fourmis vit en effet en communautés⁴⁹ plus ou moins importantes, qui sont qualifiées d'"eusociales" par [Wilson 1985], ce qui signifie qu'elles sont caractérisées par la présence simultanée et constante des trois critères suivants:

- La coopération des membres de la société pour élever les jeunes et le couvain.
- L'existence d'une division des rôles, d'abord au niveau de la fonction reproductrice, ensuite au niveau des tâches à accomplir. On appelle cette division du travail le polyéthisme.
- Le chevauchement d'au moins deux générations d'âge différent capables de contribuer aux travaux de la colonie.

On se reportera utilement à [Wilson 1992] ou à [Seger 1993] pour une démonstration intéressante du gain adaptatif que confère ce type de socialité, aussi bien par rapport aux individus solitaires (défense, fourrageage, etc...) que par rapport aux autres animaux sociaux exploitant la même niche écologique.

⁴⁸ L'essentiel des informations contenues dans cette partie sont tirées de [Hölldobler et Wilson 1990].

⁴⁹ Aussi appelées colonies, fourmilières, ou sociétés. Nous emploierons ces trois termes dans la suite du chapitre.

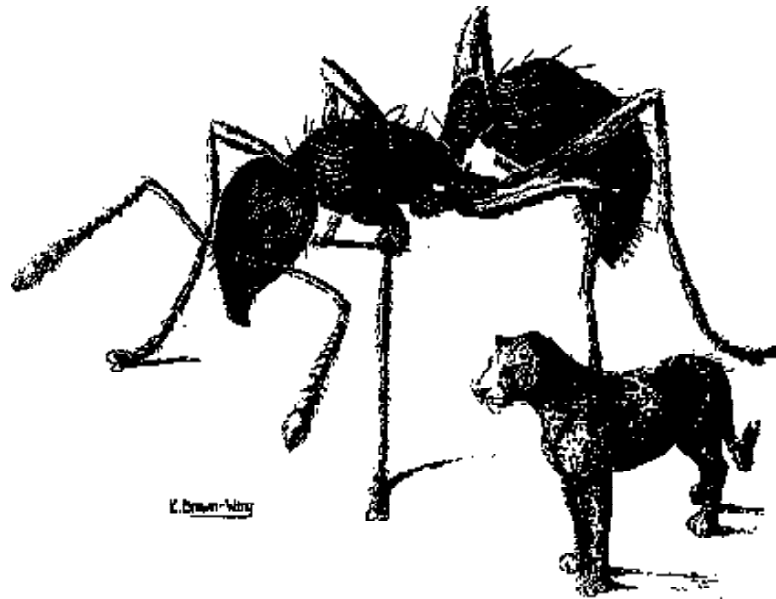


Figure II.1 - Poids comparé des fourmis et des vertébrés dans la forêt amazonienne. Leur biomasse est environ quatre fois supérieure! (d'après [Wilson 1990], reproduit dans [Jaisson 1993]).

On distingue dans une colonie de fourmis deux grandes catégories d'individus: les individus sexués, mâles et femelles, ces dernières étant ordinairement appelées les reines, et les femelles non-reproductrices, que l'on appelle les ouvrières (les "soldats", lorsqu'ils existent, n'étant que l'appellation donnée à des ouvrières de grande taille). La reproduction est donc assurée par la reine (ou les reines⁵⁰) faisant partie de la colonie selon un mode de reproduction, assez original dans le règne animal, qui porte le nom de parthénogenèse arrénotoque. Ce mode lui permet, quand elle est fécondée par un mâle, d'engendrer des femelles, donc des ouvrières, mais également d'engendrer des mâles si elle n'est pas fécondée, mâles qui pourront par la suite éventuellement la féconder... La colonie de fourmis est donc exclusivement matriarcale. Les stades de développement d'un oeuf fécondé sont au nombre de trois. Il devient tout d'abord larve (il existe plusieurs stades larvaires, de un à six selon les espèces), puis cocon (ou nymphe) et enfin ouvrière. Les trois stades préliminaires à sa transformation en ouvrière sont des stades où l'individu ne peut se suffire à lui-même. Il mène alors une vie d'assisté au sein du couvain. La société étant en règle générale monogyne, toutes les ouvrières d'une colonie sont filles de la même mère. Le premier distingo entre espèces "primitives" et espèces "évoluées" repose sur le caractère monomorphe (pas de distinctions morphologiques entre les ouvrières ou entre les ouvrières et la reine) ou polymorphe (des distinctions souvent importantes de poids, de taille ou de tout autre caractère) de la colonie. Les types d'organisation sociale sont peu variés, la seule variation étant liée à des contraintes environnementales particulières, et donc à des tâches spécifiques à certaines espèces. Ils consistent en une division de la population en un certain nombre de sous-groupes fonctionnels, appelés castes, dont la formation dépend des tâches à accomplir pour la survie de la colonie (soins au couvain, chasse, garde du nid, etc.). Le terme de polyéthisme, utilisé pour rendre compte de ce phénomène, indique donc la spécialisation d'individus dans des travaux parfois très différents, spécialisation qui peut apparaître chez des individus morphologiquement identiques. La participation des membres de la colonie à ces groupes de travail peut également dépendre, mais ce n'est pas une exclusive, de leurs caractéristiques morphologiques (si la société est polymorphe), ou de leur âge (polyéthisme d'âge). Cette division reste stable tout au long de la vie de la colonie, mais il existe une importante variabilité individuelle, que l'on nomme idiosyncrasie, les individus changeant de groupe s'ils en ont la possibilité morphologique. Dernier point, qui malmène la fourmi de la fable, les fourmis

⁵⁰ Chez les fourmis, la majorité des sociétés est monogyne, c'est-à-dire à une reine, et les cas de polygynie véritables sont plus rares que chez les autres hyménoptères (comme par exemple les abeilles).

prises dans leur ensemble consacrent de 15 à 70% de leur temps à ... ne rien faire. Les communications interindividuelles entre fourmis sont de plusieurs types et varient d'une espèce à l'autre. Il existe ainsi une communication chimique importante par le biais d'émissions de phéromones (hormones externes) qui peuvent servir à attirer, guider, déclencher une alarme, reconnaître les individus, etc. (voir [Passera 1984] ou [Chauvin 1989]). Il existe également d'autres formes de communication, comme les communications sonores par stridulations, tactiles, visuelles, etc.

En résumé, les fourmis composent une famille au sein de laquelle les différentes espèces présentent des caractères de similitude évidents (organisation sociale, développement embryonnaire, etc...), que l'on peut considérer comme des invariants phylogénétiques, d'autant plus que beaucoup sont partagés par d'autres Hyménoptères sociaux (comme les abeilles). Cependant, l'adaptation de ces invariants aux différents milieux environnementaux, mise en oeuvre au cours de leurs millions d'années d'évolution, nous font aujourd'hui découvrir un éventail de comportements, une diversité de formes et de fonctions, une *intelligence* dans la conquête de la planète qui nous apparaissent comme relativement fascinants. Que de telles créatures, pourtant insignifiantes par la taille, puissent jouer un rôle écologique aussi considérable (voir, par exemple, [Wilson 1992], sur leur importance dans les processus d'humification), comparable au moins quantitativement à celui de l'homme, ne peut que renforcer notre conviction dans l'intérêt de ce modèle pour les agents réactifs.

IV.1.2. L'espèce *Ectatomma ruidum* Roger⁵¹

Les fourmis simulées dans le cadre du projet MANTA ont pour nom *Ectatomma ruidum* Roger. Elles font partie de la sous-famille des Ponerinae, qui regroupe une bonne partie des fourmis considérées comme "primitives", chez lesquelles on observe donc un monomorphisme strict au sein des ouvrières. Le genre *Ectatomma* comprend douze espèces, toutes réparties en Amérique du Sud et en Amérique Centrale, dans les régions à dominante tropicale et subtropicale.

Ectatomma ruidum est une espèce terricole, dont la distribution géographique est indiquée sur la Figure II.2. L'effectif des nids dans la nature ne dépasse pas 200 individus, et ceux-ci sont considérés comme "adultes" quand ils atteignent une moyenne de 40 individus. Cette espèce semble essentiellement monogyne, même si quelques cas de polygynie ont été relevés.

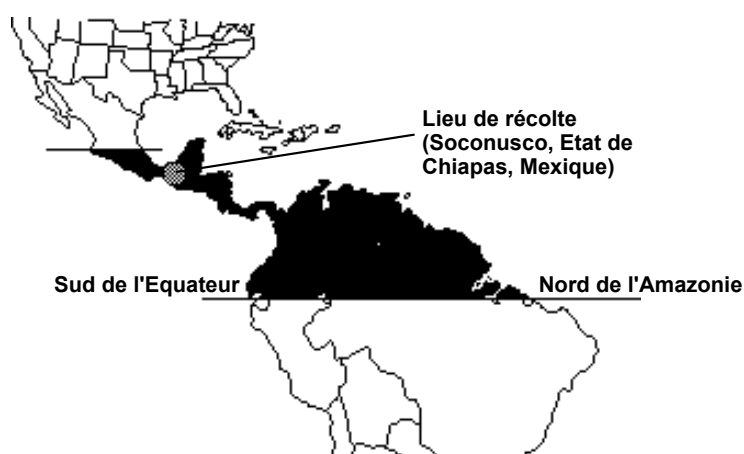


Figure II.2 - La localisation géographique d'*Ectatomma ruidum*. Sa présence est constatée depuis le Mexique jusqu'au Nord de l'Amazonie (Brésil). Les colonies élevées en laboratoire et sur lesquelles se fonde la simulation ont été récoltées dans l'Etat de Chiapas, au Mexique [Corbara 1991].

⁵¹ Les détails de la biologie et de l'organisation sociale d'*E. Ruidum* auxquels nous nous référons sont tirés de [Corbara 1991].

En laboratoire, les fourmis sont élevées dans des nids en plâtre (cf Section V.1.2) dans lesquels des chambres ont été creusées et qui permettent, par l'intermédiaire de dispositifs expérimentaux complexes (cf. [Corbara 1991] pour une description des divers dispositifs, qui incluent notamment un marquage individuel comme celui présenté Figure II.3, également utilisé par [Lenoir 1979], des techniques d'enregistrements photographiques, etc.), d'étudier l'organisation sociale de leurs colonies. Comment étudie-t-on une organisation sociale, c'est-à-dire, dans les cas des fourmis, le polyéthisme ? Comme l'indiquent [Corbara, Fresneau et Lachaud 1987], «*Les travaux qui se rapportent à [la problématique de l'organisation sociale] aboutissent le plus souvent à l'élaboration de profils comportementaux*», soit des listes de fréquences de comportements, relatives à un individu, ou relatives à un groupe d'individus, pour une période d'observation. Ce profil, qui va dépendre des comportements observés, est donc naturellement basé sur une notion de répertoire comportemental, «*liste d'actes ou situations s'excluant mutuellement*» [Corbara, Fresneau et Lachaud 1987] (souligné par nous). L'établissement de ce répertoire est soumis à une assez forte subjectivité, de même que celui de l'échantillonnage temporel utilisé pour la génération du profil, mais dans la mesure où il semble difficile de recourir à l'introspection pour analyser les comportements de fourmis, c'est encore cette méthode qui apparaît comme la plus objective. L'intérêt dans notre cas de figure est que ce répertoire comportemental présente une forte similitude avec une liste de tâches indépendantes.

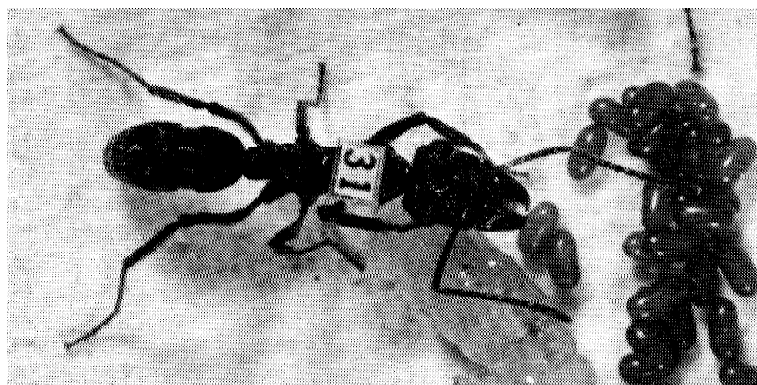


Figure II.3 - Un exemple de marquage des fourmis en vue de relevés photographiques. Chaque fourmi possède son numéro propre et il est possible d'identifier son comportement en fonction de la situation et de sa posture (colonie d'*Ectatomma ruidum*, détail d'une photographie publiée dans [Corbara et al. 1986]).

Ce répertoire comportemental va donc être utilisé comme une grille de lecture pour le dépouillement des enregistrements visuels. La technique utilisée pour distinguer les groupes fonctionnels est la suivante ([Corbara 1991], p. 25):

Les données, une fois recueillies, sont réorganisées sous la forme d'une matrice Individus/comportements qui, à chaque individu, assigne une série de fréquences d'occurrence pour chaque catégorie comportementale. Chaque fourmi est donc caractérisée par un profil comportemental individuel. Si on lit la matrice dans l'autre sens, à chaque catégorie comportementale correspond une série de fréquences des contributions individuelles sur la tâche considérée. (...) Les matrices de données ont le plus souvent été traitées par des programmes d'analyse multivariée. Nous avons fait appel, tout au long de ce travail, à deux techniques de taxonomie numérique: la classification hiérarchique ascendante et l'analyse factorielle des correspondances. (...) De cette façon, il nous est possible de ségréguer et de regrouper des individus ayant un profil comportemental voisin et de définir ainsi des groupes fonctionnels.

Un format usuel de description de ces groupes fonctionnels est le sociogramme regroupé, tel celui présenté Figure II.4.

Sociogramme Regroupé

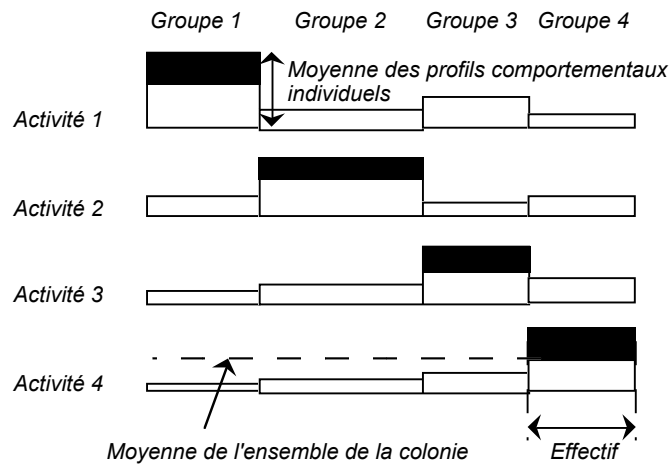


Figure II.4 - Description abstraite d'un sociogramme regroupé. Les activités sont en ligne, les groupes en colonne.

Si nous insistons sur les protocoles expérimentaux et sur les méthodes d'extraction de données mis en œuvre par les éthologues, c'est que, comme nous l'avons indiqué au Chapitre III, la reconstruction de la réalité que nous opérons va nous permettre de travailler avec des protocoles équivalents sur les fourmis artificielles de MANTA (voir par exemple Section V).

IV.1.3. Objectifs Généraux de la Simulation

Le projet MANTA est né de la rencontre de deux domaines de recherche, et donc de deux problématiques. D'un côté, la problématique de la compréhension des modes d'organisation du vivant, propre aux éthologues, de l'autre la problématique de la synthèse d'organisations artificielles fonctionnelles, qui nous est propre (voir Chapitre I). Ce sont deux problématiques complémentaires. La compréhension des modes d'organisation du vivant influe en effet sur notre manière particulière de concevoir les organisations artificielles, en proposant des modèles alternatifs à ceux de la psychosociologie humaine. L'exemple le plus éloquent de cet apport est constitué par le domaine de la Vie Artificielle. A l'inverse, la synthèse d'organisations artificielles permet d'appréhender les modes d'organisation du vivant de manière nouvelle, ce fait étant illustré par le développement important de la "pensée systémique" en biologie. Ces deux problématiques se sont donc naturellement confondues, l'espace d'un projet, en une problématique de simulation d'un mode particulier d'organisation, l'organisation sociale d'une colonie de fourmis, projet qui a le mérite d'aborder simultanément les deux aspects de synthèse et de compréhension, et surtout d'en faire ressortir clairement les interactions. En termes plus prosaïques, nous nous sommes ainsi fixés trois objectifs principaux qui sont les suivants :

- Le premier porte sur l'intérêt de la méthodologie multi-agents que nous défendons pour effectuer des simulations en éthologie, biologie ou écologie. Les questions auxquelles nous souhaitons apporter une réponse dans le cadre de ce projet portent sur les contraintes d'une approche multi-agents, sur ses possibilités, et sur ses apports tangibles par rapport aux outils de simulation classiques. En construisant un projet complet de simulation, notre but est donc de valider expérimentalement l'approche multi-agents de phénomènes semblables de manière générale, et notre instanciation de cette approche par le biais de l'EthoModélisation en particulier.

- Le second est directement lié à la problématique de la validité des descriptions du comportement des insectes sociaux⁵². Il consiste à vérifier que les hypothèses élaborées par les éthologues sur le comportement des individus (répertoire comportemental, mode de spécialisation, etc.) suffisent à expliquer la génération et la stabilité des formes sociales observées. En d'autres termes, est-ce que décrire les fourmis comme des individus simples dotés de comportements réactifs suffit à reproduire au niveau social les performances adaptatives complexes dont sont capables les sociétés de fourmis réelles ?
- Le troisième, qui se situe à la confluence des deux précédents et qui est lié à la problématique des agents réactifs, est d'explorer plus avant les relations entre le niveau micro (représenté par les fourmis) et le niveau macro (représenté par la colonie). Si nous sommes capables, par voie de simulation, de générer à partir d'interactions d'agents simples une structure fonctionnelle, adaptative et stable, apte à résoudre les problèmes posés par son environnement, nous serons capables, même si cette connaissance reste empirique, de mieux comprendre le lien morphogénétique qui existe entre les deux niveaux, dans le but de le reproduire dans des applications d'IAD.

Ces trois objectifs vont naturellement de pair, dans la mesure où l'émergence réussie d'une structure sociale comme celle de la division du travail permet de valider l'approche choisie (*i.e.*, nous sommes capables de le faire en utilisant cet outil), de valider les comportements fournis aux agents (*i.e.*, ces comportements individuels suffisent à générer la structure globale) et d'espérer reproduire ce phénomène en l'appliquant à un autre domaine (*i.e.*, des robots pourraient donc le faire dans un atelier de production, par exemple).

IV.1.4. Objectifs Ethologiques de Manta

L'organisation d'une colonie d'insectes sociaux et la répartition du travail entre les différents individus ne laisse pas d'être un phénomène surprenant. Si l'on abandonne les notions en partie dépassées qui tendent à idéaliser les colonies, soit comme des modèles miniatures de nos propres sociétés, soit en des termes suffisamment vagues et attrayants pour ne pas nécessiter de plus amples explications, comme la notion de super organisme ([Wheeler 1911], cité dans [Jaisson 1993]) qui n'est pas dénuée d'intérêt, mais dont la valeur explicative reste floue, l'on est confronté à une forme d'organisation assez étonnante, faite d'une combinaison dynamique de coopération et de compétition entre individus relativement simples, qui semble plus reposer sur une sorte de *confiance absolue* individuelle à l'égard des autres individus ([Jaisson 1993], p. 133) que sur une structure à caractère normatif.

Si l'on laisse de côté l'intérêt fonctionnel et écologique relativement évident des formes de socialité apparues chez les insectes, de même que si l'on ne s'interroge pas sur la manière dont elle se sont développées, en laissant ces deux questions à la sociobiologie, par exemple, dont c'est le terrain de prédilection (voir par exemple [Wilson 1985; Wilson 1990]), nous sommes confrontés à une question en apparence simple, mais terriblement complexe en réalité: *quels sont les mécanismes individuels qui permettent, par interaction, coopération et compétition avec les autres individus, de générer la structure sociale stable et reproductible que l'on observe aussi bien chez les fourmis que dans d'autres espèces d'hyménoptères ?* En d'autres termes, par quel jeu subtil s'opère le passage entre le niveau individuel et le niveau social, sachant que les individus concernés n'ont *a-priori* pas les capacités psychiques de *penser* le niveau social ?

Certains courants évolutionnistes de la biologie contemporaine (dont la sociobiologie) ont tendance à évacuer cette problématique ontogénétique en la dissolvant au sein d'une problématique plus

⁵² Cet objectif est décrit de manière plus détaillée en Section IV.1.4.

phylogénétique et énoncent, par un raisonnement purement tautologique, que les individus que nous observons à l'heure actuelle ont été sélectionnés, au cours de l'évolution, pour leur capacité d'adaptation à ce fait social, et qu'un double processus morphogénétique (d'élaboration de la société par les individus, et d'élaboration des individus par la société) a été à l'oeuvre durant des millions d'années pour finalement nous laisser des individus génétiquement programmés pour vivre en société. La société est donc ce qu'elle est parce que les individus sont ce qu'ils sont. Certes. Mais les théories évolutionnistes, qu'il ne s'agit pas de remettre en cause ici, répondent en définitive à la question *par quels mécanismes évolutionnaires sont apparues ces formes*. Par contre, elle ne répond pas à la question qui consiste à comprendre *par quels mécanismes élémentaires ces formes émergent de manière dynamique*.

Dans une perspective éthologique, la réponse à la deuxième question est au moins aussi importante que la réponse à la première, si ce n'est plus⁵³. Le projet MANTA, qui se place délibérément dans cette perspective, a pour but de fournir des résultats qui permettront de comprendre quels sont les comportements individuels nécessaires et suffisants à la génération d'une société artificielle de fourmis, et de quelle manière ils y contribuent. Il se base pour cela sur les hypothèses comportementales énoncées et défendues, entre autres, par [Corbara et al. 1993; Fresneau, Corbara et Lachaud 1989], hypothèses en partie basées sur l'affirmation que les processus ontogénétiques (ou éthogénétiques [Jaisson 1993]), comme la spécialisation, et interactionnels, comme la compétition entre individus, suffisent à expliquer l'émergence de configurations sociales complexes et auto-entretenuës, sans qu'il soit nécessaire de recourir à des explications phylogénétiques, telle la reconnaissance de parentèle (voir Section V.7.). Encore une fois, entendons-nous bien: l'aspect évolutionnaire est un aspect fondamental de la recherche en biologie, écologie et éthologie. Mais ce n'est tout simplement pas notre problématique.

Notre objectif "éthologique" avec MANTA, est donc, une fois le bagage comportemental et les processus d'ontogénie codés, d'essayer d'obtenir des phénomènes jugés comparables (par les éthologistes) à ceux observés dans la réalité. La simulation, et donc, dans une certaine mesure la théorie ayant servi à construire le modèle, seront ainsi considérées comme validées par l'obtention, premièrement, d'une dynamique de sociogenèse proche de celle des colonies naturelles (Section V.2.), deuxièmement par l'émergence d'une structure de division du travail entre les individus, sous la forme stable et adaptative d'un ensemble de groupes fonctionnels dont les membres sont spécialisés dans une ou plusieurs tâches, et enfin par l'apparition d'une division du travail en fonction de l'âge (qui porte le nom de polyéthisme d'âge).

Il est important de noter, donc, que l'ambition de MANTA n'est pas d'obtenir des fourmis artificielles, c'est-à-dire des automates se comportant de manière absolument semblable aux fourmis réelles, mais bien plutôt de générer des colonies artificielles dont la configuration sociale possède les caractéristiques structurelles et fonctionnelles d'une colonie réelle. En d'autres termes, si émergence il doit y avoir, ce n'est pas dans le comportement individuel qu'il faudra la chercher (même si cette problématique reste une de nos préoccupations, elle ne fait pas partie du projet), mais dans le comportement global résultant des interactions entre les individus et dans le rôle que ce comportement global attribuera, de manière décentralisée, à l'individu. Nous retrouvons donc bien ici la problématique du double processus morphogénétique évoquée dans le cadre des théories évolutionnistes, mais située sur une autre échelle temporelle et dans un contexte qui nous semble moins spéculatif.

⁵³ «(...) le comportement observable à un moment donné est toujours le résultat d'une histoire à deux termes interreliés: un terme phylogénétique décrivant l'appartenance (...) à une espèce; un terme ontogénétique décrivant l'état momentané (...) dans un continuum où s'affrontent et s'intriquent maturation et expérience» ([Richard 1975] p. 253)

IV.1.5. Travaux similaires de modélisation d'insectes sociaux

IV.1.5.1. LE MODELE MIRROR

Les premiers travaux que nous pouvons qualifier de similaires, bien qu'ils ne s'intègrent pas directement dans une problématique multi-agents, sont ceux de Pauline Hogeweg et Bernard Hesper [Hogeweg et Hesper 1983; Hogeweg et Hesper 1985; Hogeweg et Hesper 1991], qui nous ont servi de référence tout au long de notre travail, tant par l'acuité de la vision qu'ils ont développée que par l'intérêt indéniable de leurs résultats. Leur système de simulation, baptisé MIRROR (la méthodologie sur laquelle il s'appuie s'appelant MICMAC, pour MICro-MACro [Hogeweg et Hesper 1979]), a en effet pour finalité de générer des comportements globaux (sociaux) à partir de comportements individuels et locaux sans inclure aucune assomption sur les relations globales au niveau local.

On voit donc qu'il participe de la même philosophie que le nôtre. MIRROR a servi à développer plusieurs projets de simulation très divers, mais celui qui nous intéresse le plus concerne bien entendu le modèle développé pour rendre compte de l'ontogénie de la structure sociale des colonies de Bourdons (plus précisément décrit dans [Hogeweg et Hesper 1983]). Ce modèle, entièrement basé sur les comportements individuels des bourdons, arrive en effet à reproduire à la fois la composition et le développement démographique d'un nid, mais également des phénomènes cycliques complexes, comme le départ de la reine ou l'apparition de groupes d'ouvrières dominantes. Cependant, si la philosophie générale est la même, ce système diffère du nôtre en ce qu'il n'intègre aucun modèle de comportement explicitement établi, ces comportements, ainsi que leur déclenchement, pouvant prendre en fait n'importe quelle forme (fonctions LISP, règle de production, voir [Hogeweg 1988]). Néanmoins, la richesse des comportements fournis aux individus de même que l'attention portée à la modélisation de l'environnement spatial font que nous pouvons considérer que le modèle MIRROR a constitué notre plus grande source d'influence. Les travaux qui ont succédé à cette modélisation ont quitté le terrain des insectes sociaux pour se concentrer soit sur d'autres espèces animales (par exemple, le modèle CHIMP [Hogeweg et Hesper 1991]), soit sur des études relevant de l'intelligence en essaim, dont ils ont été parmi les précurseurs.

IV.1.5.2. LES AUTRES MODELES

En comparaison, les autres modèles "multi-agents" ou comportementaux consacrés aux insectes sociaux apparaissent un peu moins ambitieux. La plupart sont en effet dédiés à l'étude de certaines fonctionnalités effectuées par les insectes sociaux (*e.g.* construction, fourragement, etc...), ou, quand il s'agit d'analyser la structuration sociale, aux interactions entre des individus extrêmement simplifiés. Cependant, tous ceux que nous allons rapidement passer en revue ont, chacun à leur manière, influencé notre propre système.

Le premier est celui présenté dans [Chauvin et Callais-Hamonno 1972] et décrit de manière plus concise dans [Chauvin 1982], a consisté à trouver les règles élémentaires de base nécessaires à la construction du dôme chez les fourmis. Il est très intéressant historiquement dans la mesure où il préfigure de presque 10 ans les travaux réalisés par l'équipe de Jean-Louis Deneubourg à Bruxelles. Les deux conclusions auxquelles arrive Rémy Chauvin sont les suivantes: deux règles stochastiques, l'une contrôlant le déplacement, l'autre contrôlant la prise ou le dépôt de brindilles, le tout contrôlé par une probabilité plus grande de se déplacer en direction du nid une fois chargé, permettent de voir se construire un dôme sur l'écran de l'ordinateur, sans qu'il y ait nulle part de coopération entre les ouvrières. Ce système met de plus en évidence le rôle fondamental du hasard (ou des "actes absurdes", comme les appelle Rémy Chauvin) dans l'adaptation de la colonie à son environnement, ce qui n'est pas

sans rappeler les concepts d'ordre ou de complexité par le bruit [Atlan 1972; Von Foerster 1960], ou d'ordre par fluctuations [Glansdorff et Prigogine 1971]. C'est en effet sur des bases conceptuelles très proches que l'équipe de Deneubourg a analysé et simulé différents phénomènes collectifs: la construction des ruches [Belic et al. 1986], les stratégies de fourragement des fourmis [Deneubourg et al. 1986; Deneubourg et al. 1987; Deneubourg, Pasteels et Verhaeghe 1984; Goss et al. 1990], leurs méthodes de recrutement [Deneubourg et Goss 1989], les capacités d'aggrégation/ségrégation du couvain, appelées "tri collectif" [Deneubourg et al. 1991], etc.

Ces travaux ont longtemps constitué la référence obligée en matière de simulation d'insectes sociaux, et ont été repris, sous des formes plus ou moins semblables, par de nombreux chercheurs (par exemple, [Hatcher, Tofts et Francks 1992; Moyson et Manderick 1988], mais la liste est beaucoup plus longue). Les résultats obtenus et certaines méthodes utilisées ont permis des avancées conceptuelles considérables, parmi lesquelles les études sur les influences réciproques à l'aide de stimuli, sur la communication indirecte via l'environnement, sur l'effet du renforcement, toutes avancées dont nous sommes largement bénéficiaires. Cependant, l'attitude, sans doute volontaire, qui consiste dans tous ces travaux à gommer d'une certaine manière une large part du comportement pour ne se concentrer que sur les comportements qui apparaissent comme fonctionnellement (et subjectivement) "utiles", les autres intervenant sur leur déroulement par une simple influence stochastique, fait que nous ne pouvons pas vraiment les considérer comme des travaux de simulation aussi ambitieux que ceux de [Hogeweg et Hesper 1983], mais plutôt comme d'excellents travaux d'abstraction en termes de résolution de problèmes (cf. Chapitre VI).

On en voudra pour preuve la difficulté qu'ont eu certains de ces systèmes à simuler des environnements sociaux complexes, que ce soit dans le cas du modèle COLONY [Corbara 1991; Corbara et al. 1991], appliqué à *E. ruidum* ou dans celui du modèle qui en a été dérivé pour étudier les guêpes *Polistes*, décrit dans [Theraulaz 1991; Theraulaz et al. 1991]. L'extrême abstraction dont ils font preuve ne semble en effet pas convenir pour la description d'individus dont le répertoire comportemental comporte plusieurs tâches en compétition. Il nous paraît de plus extrêmement important de prendre à la fois la dimension spatiale (organisation du nid, par exemple) et la dimension sociale (division du travail) en compte, spécificité qu'aucun des deux modèles ne semble faire. Comme nous le verrons, en effet, la sensibilité différentielle des individus résultant de leur spécialisation est à la fois un effet de leur historique comportemental (par renforcement) et un effet de la structuration du nid, qui dans une certaine mesure, conditionne leur comportement présent et à venir. (voir à ce sujet [Fresneau, Corbara et Lachaud 1989]). L'on ne peut ainsi s'en tenir, comme le font les deux modèles cités, à la simple description d'un intérieur et d'un extérieur.

IV.2. Les Agents de Manta

IV.2.1. Décomposition en Agents

La première étape de notre modélisation a consisté à décomposer une colonie de fourmis *Ectatomma ruidum* en un ensemble d'agents et d'objets. Un premier tri rapide permet de conclure à l'existence de trois grandes catégories:

- Les "assistés": qui regroupent tous les membres du couvain, c'est-à-dire les œufs, les larves à leurs différents stades de maturation et les cocons.
- Les "assistants": qui comprennent la reine, les ouvrières et les mâles.
- Les autres: où sont regroupés les aliments, sous quelque forme qu'ils soient, les déchets et déjections en tout genre et les cadavres de fourmis.

Ce tri une fois effectué, il nous a fallu éliminer de ces catégories les agents qui, aux yeux des éthologues, n'apparaissent pas comme pertinents pour l'analyse que nous nous sommes fixée. Ont donc été supprimés:

- Les trois stades de larve, qui n'en forment plus qu'un seul. Le comportement des fourmis à leur égard est en effet le même. La durée de développement de ce super stade larvaire est bien entendu égale à la somme des durées de développement de chacun des trois stades.
- Les mâles, qui n'interviennent que rarement dans l'organisation de la société (cf. [Hölldobler et Wilson 1990]). Leur seule fonction utile semble être de remplir la spermathèque de la reine, et leur durée de vie n'excède pas quinze jours.
- Les déchets, qui ne semblent pas jouer de rôle hormis dans le soin aux cocons, durant lequel les fourmis les utilisent pour façonner des sortes de petits piédestaux. Ayant décidé de ne pas tenir compte du détail des soins, nous construirons donc une fourmilière propre.
- Et enfin, les cadavres. Bien que les ouvrières semblent en effet avoir des comportements dirigés vers ces cadavres et que la quasi-totalité des espèces de fourmis entretiennent des "cimetières" dans leurs nids [Chauvin et Callais-Hamonno 1972] ou à l'extérieur du nid [Corbara 1991], leur rôle dans le processus d'organisation sociale n'est pas patent.

Il est à noter que la suppression de ces agents n'est pas définitive et qu'à tout moment il nous est possible de les rajouter (ainsi que de rajouter les tâches nécessaires aux fourmis) si nous nous apercevons que leur absence pose problème. Cependant, les premiers résultats obtenus laissent à penser que ces agents n'interviennent que marginalement dans l'organisation de la colonie (voir Section V.4.) et que leur suppression ne remet pas en cause sa dynamique. Nous avons donc conservé dans le modèle les classes concrètes suivantes: **EggAgent** (les oeufs), **LarvaAgent** (les larves), **CocoonAgent** (les cocons), **AntAgent** (les ouvrières), **QueenAgent** (la ou les reines) et **FoodAgent** (la nourriture).

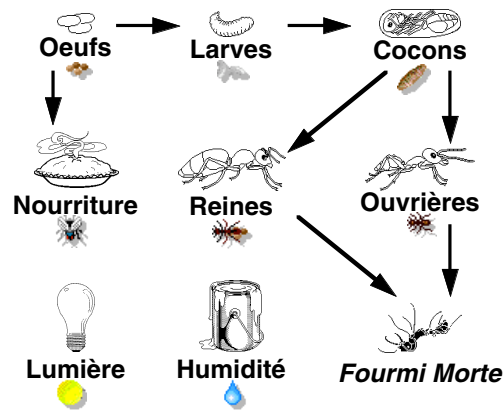


Figure II.5 - Les différents agents composant les colonies de MANTA. La classe *Fourmi Morte* y est représentée, mais n'est pas actuellement utilisée. A chaque type d'agent est associée une icône (sous son nom) qui indique sa représentation graphique dans la simulation.

De plus, suivant en cela les indications de [Fresneau, comm. Pers.], il apparaît que le nid n'est pas seulement structuré par sa configuration topologique (parois et obstacles), mais également par les gradients de deux agents environnementaux abiotiques qui sont l'humidité et la lumière. Tous deux jouent en effet un rôle essentiel dans la configuration spatiale de la colonie (certaines chambres vont être plus humides ou plus lumineuses que d'autres, ce qui va jouer sur la répartition du couvain), mais également sur le comportement des fourmis (on sait que les jeunes ouvrières sont photophobes, donc enclines à rester à l'intérieur du nid, mais qu'elles deviennent progressivement moins craintives, ce qui a un effet non négligeable sur l'évolution de leur mobilité). Nous avons donc inclus ces deux nouveaux

types d'agent dont le rôle sera, par l'intermédiaire de la diffusion de leur stimuli, de servir soit à l'orientation des fourmis, soit de déclencheurs de comportements spécifiques (fuite de la lumière), soit d'influer sur la force de certains autres stimuli. Ils seront représentés par les classes **LightAgent** (lumière) et **HumidityAgent** (humidité).

IV.2.2. Les Primitives Nécessaires

IV.2.2.1. DETERMINATION DES NOUVELLES PRIMITIVES

Ces nouvelles classes vont avoir besoin de primitives et de connaissances qui ne sont pas toutes définies dans le noyau, puisqu'elles sont pour la plupart dépendantes du domaine considéré. Ces primitives (et les connaissances qui vont avec) sont, bien entendu, une fois de plus déterminées grâce à l'observation des individus réels effectuée par les éthologistes et aux catégories abstraites de comportement qu'ils en dérivent. Ainsi, nous pouvons déterminer que:

- L'humidité et la lumière ne possèdent aucun comportement (ces deux facteurs restent inchangés en laboratoire), mais qu'il faut prévoir de faire varier la force de leur stimuli (pour simuler, par exemple, l'alternance jour/nuit).
- La nourriture (hormis dans le cas de proies mobiles, qui ne nous intéresse pas ici car les aliments fournis aux colonies en laboratoire sont généralement immobiles) n'a absolument aucun comportement, si ce n'est celui de perdre sa valeur nutritive au fil du temps.
- Les oeufs et les cocons n'ont aucun comportement particulier, si ce n'est qu'ils ont besoin d'être régulièrement déplacés, nettoyés et soignés et qu'ils le font savoir par l'intermédiaire de la diffusion de phéromones (voir [Chauvin 1982; McFarland 1990; Passera 1984]).
- Les larves nécessitent les mêmes soins que les œufs et les cocons, mais sont en plus capables de se nourrir, si la nourriture est proche (voir, par exemple, [Hölldobler et Wilson 1990] sur la notion de pseudotrophallaxie).
- Les ouvrières sont capables de se déplacer, de suivre des gradients, de porter, de soigner ou nettoyer des agents, de se nourrir et de tuer les oeufs et les larves pour se nourrir.
- Enfin, la reine, peu différente des ouvrières [Corbara 1991], est capable d'effectuer tous leurs comportements, mais possède en plus la capacité de pondre des œufs.

En outre, la caractéristique commune à ces agents est qu'ils sont capables de vieillir et de changer d'état (le fait de mourir n'étant ici envisagé que comme un changement d'état particulier). Bien entendu, ce changement peut être périodique et non définitif, comme dans le cas des agents lumière ou humidité.

IV.2.2.2. DE NOUVELLES CLASSES ABSTRAITES

La gestion de ces comportements nécessite de créer de nouvelles classes abstraites, et de modifier l'arbre d'héritage existant afin de créer un "noyau MANTA" à partir duquel seront dérivées les classes concrètes. Les principes sur lesquels nous nous sommes fondés sont (voir en parallèle la Figure II.6):

- Tous les agents du système possèdent le même modèle de comportement, ont une représentation graphique et sont localisés. Ils sont donc sous-classes de `EthoBehavior`, `InterfaceBehavior` et `LocatedBehavior`.
- Ils sont soumis à des variations temporelles et peuvent changer d'état en fonction de leur âge. Ils seront donc sous-classes de `MaturingBehavior`, appelée à gérer le temps, à proposer des stimuli génériques et des primitives permettant de grandir, de mourir et de changer d'état.

- La première bifurcation se produit sous `MaturingBehavior`, puisque certains agents ont besoin d'être soignés ou nettoyés, et d'autres non (`FoodAgent`, `HumidityAgent`, `LightAgent`). Ces trois dernières classes vont donc être dérivées à cet endroit, cependant que les autres vont hériter d'une classe abstraite appelée `CuringBehavior`⁵⁴.
- Les oeufs et les cocons n'ayant pas besoin d'autres comportements, leurs classes sont directement issues de `CuringBehavior`. Tous les agents restants ont par contre besoin de consommer de la nourriture et sont donc sous-classes de `FeedingBehavior`.
- La classe `LarvaAgent` est dérivée à cet endroit-là, cependant que les instances des deux dernières classes concrètes, `AntAgent` et `QueenAgent`, vont devoir se déplacer, suivre des gradients, porter d'autres agents et en tuer certains. On retrouve donc, sous `FeedingBehavior`, les classes `MovingBehavior` et `SensingBehavior`, décrites dans le noyau, ainsi que les classes `CarryingBehavior` et `KillingBehavior`, de laquelle est issue `AntAgent`.
- Enfin, les reines ayant besoin du comportement de ponte, on place la classe `LayingBehavior`, dont `QueenAgent` hérite, sous `KillingBehavior`.

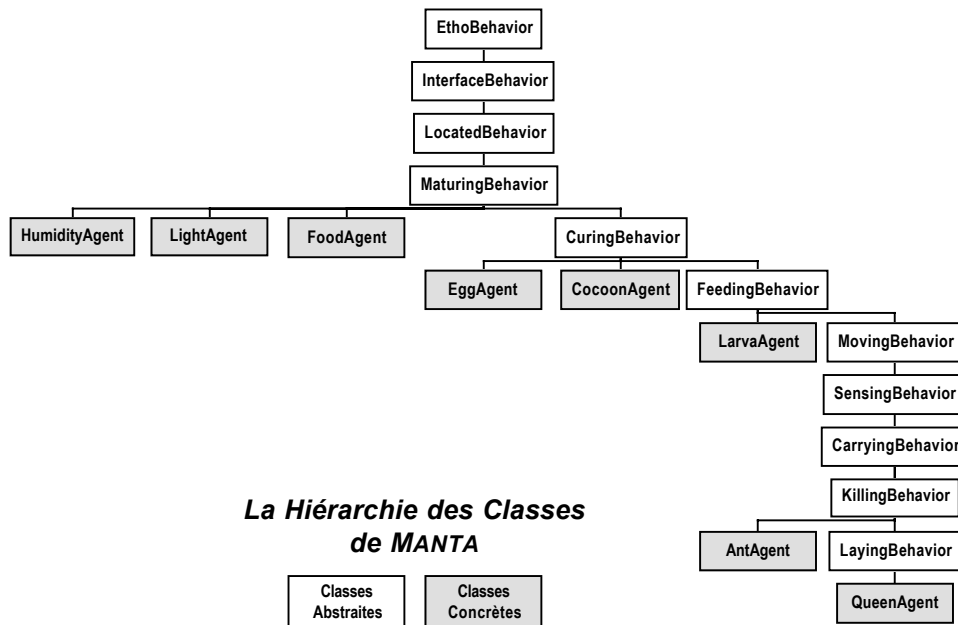


Figure II.6 - La hiérarchie des classes de MANTA. Sont indiquées en blanc les classes abstraites qui composent le nouveau noyau, et en grisé les classes concrètes qui font partie de la simulation.



IV.2.2.3. LA PROGRAMMATION DES CLASSES ABSTRAITES


Qu'apportent de nouveau ces classes abstraites ? Outre les primitives qui leur sont spécifiques, certaines d'entre elles fournissent aux agents des stimuli génériques, qui seront ensuite particularisés suivant le type de l'agent qui en hérite. La plupart redéfinissent en outre la méthode `readEnvironment` (voir Chapitre III), appelée à chaque cycle, afin d'effectuer certains traitements. Enfin, quelques classes vont nécessiter des agents qu'ils initialisent les variables d'instance qu'elles définissent par l'intermédiaire de méthodes génériques. Comme l'ensemble de ces fonctionnalités serait un peu long à décrire exhaustivement, nous avons regroupé les principales dans la liste qui suit. La classe `LayingBehavior` n'y figure pas en raison de son extrême spécificité. Elle ne définit qu'une primitive, appelée



⁵⁴ Qui regroupe les comportements des agents nécessitant et fournissant des soins.

primLayEggs, et un stimulus interne #layEggsQueen, dont l'intensité est incrémentée à chaque cycle et remise à zéro après chaque ponte.

```



LocatedBehavior subclass: #MaturingBehavior
  instanceVariableNames: 'time maturingTime isDead '
  time est l'âge de l'age, maturingTime son âge maximal.
  Primitives:
   primMature    "permet à l'agent de changer d'état"
   primDie      "permet à l'agent de mourir"
  readEnvironment: Incrément de time à chaque appel
  Stimuli: #maturing +type de l'agent force: [time - maturingTime]
  Méthodes à redéfinir:
    initializeMaturingTime    "Age maximal que peut atteindre l'agent"
    nextAgentClass           "Classe du nouvel agent après un
    changement d'état"


MaturingBehavior subclass: #CuringBehavior
  instanceVariableNames: 'cureLevel '
  cureLevel indique le niveau de soins actuel de l'agent.
  Primitives:
   primCure:    "permet à l'agent de prodiguer des soins"
  readEnvironment: Décrément du cureLevel à chaque appel.
  Stimuli: #cure +type de l'agent force: [minCureLevel - cureLevel]
  Méthodes à redéfinir:
    initializeCureLevel "initialisation du niveau de soins de l'agent"
    minCureLevel       "limite inférieure du niveau de soins"
    maxCureLevel       "limite supérieure du niveau de soins"
    cureDecrement      "Décrément du niveau de soins à chaque cycle"

CuringBehavior subclass: #FeedingBehavior
  instanceVariableNames: 'foodLevel '
  foodLevel indique le niveau des réserves de nourriture de l'agent.
  Primitives:
   primHasFood "permet à l'agent de savoir s'il a des réserves"
   primEat     "permet à l'agent de se nourrir"
  readEnvironment: Décrément du foodLevel à chaque appel.
  Stimuli: #hungry +type; force: [minFoodLevel - foodLevel]
  Méthodes à redéfinir:
    initializeFoodLevel "initialisation des réserves de nourriture"
    minFoodLevel       "limite inférieure des réserves"
    maxFoodLevel       "limite supérieure des réserves"
    foodDecrement      "décrément de la réserve à chaque cycle"

```

```

MovingBehavior subclass: #CarryingBehavior
  instanceVariableNames: 'carried carriedType '
  carried contient la liste des agents transportés, carriedType leur
  type.
  Primitives:
   primPickUp: "permet de saisir des agents (paramètre:leur type)"
   primPutDown: "permet à l'agent de poser les agents qu'il porte"

CarryingBehavior subclass: #KillingBehavior
  instanceVariableNames: ''
  Primitives:
   primKill: "permet à l'agent de tuer un autre agent"

  Stimuli Génériques: #killEgg, #killLarva. Leurs forces sont dépendantes
  de la force du stimulus dénotant la faim ('hungry'+type de l'agent) et
  de la force des stimuli egg et larva dans l'environnement.

```

IV.2.3. Une nouvelle classe de stimulus

Les stimuli précédemment définis dans le noyau d'EthoModélisation (instances des classes `Stimulus` et `PropagableStimulus`) étaient indépendants les uns des autres et, notamment dans le cas de ceux qui étaient propagés, n'influençaient en rien la propagation des autres stimuli de même type. Il nous est apparu cependant, en observant la manière dont les phéromones se diffusaient dans l'environnement, qu'un stimulus pouvait avoir un effet sur l'intensité des stimuli de même type émis par d'autres agents. Cet effet porte le nom de processus *auto-catalytique*. Un processus ou un phénomène auto-catalytique est un processus qui se renforce de lui-même, en créant à chaque amplification les conditions favorables à sa future amplification. Comme le note [Deneubourg et Goss 1989], ceci se réfère à l'idée que la probabilité d'un individu d'adopter un comportement ou un état donné est une fonction croissante du nombre d'individus qui ont déjà adopté ce comportement ou cet état (voir également [Grassé 1959], et sa théorie de la stigmergie).

Dans le cas de la propagation de stimuli par le couvain, l'hypothèse que nous faisons est que les stimuli, tout en se propageant indépendamment, influent les uns sur les autres pour créer un effet de concentration. Nous avons donc créé la classe `AutoCatalyticStimulus`, définie ci-après:

```

PropagableStimulus subclass: #AutoCatalyticStimulus
  instanceVariableNames: ''
  category: 'Etho - Concepts'

```

classe dont le principe de propagation est semblable à celui de sa super-classe `PropagableStimulus`, mais dont les instances multiplient leur intensité en fonction du nombre d'agents de même type que leur propogateur présents sur cette place.

IV.3. Programmation du Comportement des Agents

Sont décrits dans cette section les comportements associés à chacun des types d'agent existant dans la simulation. Nous allons donc décrire les tâches qui leur ont été fournies, ainsi que les stimuli qu'ils possèdent et qu'ils propagent dans l'environnement. Les détails de certains paramètres (poids et seuils des tâches, certains paramètres biologiques), dépendants des expérimentations réalisées, sont quant à eux décrits dans la section consacrée aux protocoles (Section V.1.).

IV.3.1. Les agents environnementaux et la nourriture

Les deux catégories d'agents environnementaux (respectivement, la lumière et l'humidité) n'ont pas, dans la version actuelle de MANTA, de comportement propre. Ils sont simplement initialisés avec un stimulus dont le nom est égal à leur type (`#humidity` et `#light`) et dont la force est arbitraire. La classe `FoodAgent` définit deux variables d'instance respectivement appelées `foodLevel` et `foodDecrement`. La première spécifie le nombre d'unités de nourriture que comporte l'agent et la seconde de combien d'unités elle se détériore à chaque cycle. Les agents nourritures émettent bien sûr un stimulus qui leur est propre (`#food`), dont l'intensité est fonction de la valeur de `foodLevel`.

IV.3.2. Le couvain

IV.3.2.1. LES OEUFS ET LES COCONS

Les oeufs sont, avec les cocons, les agents les plus simples du couvain. Ils héritent de trois stimuli génériques: le stimulus `#egg`, qui indique simplement leur présence, le stimulus `#cureEgg`, qui dénote leurs besoins de soins et le stimulus interne `#maturingEgg`, dont l'intensité devient positive à partir du moment où l'agent a dépassé son âge (fixé au départ). Les variations d'intensité du stimulus `#egg` sont liées à l'environnement dans lequel l'agent se trouve. S'il est posé à terre, il croît régulièrement en fonction de la valeur du stimulus `#humidity` et en fonction inverse de celle de `#light`. Si i_e représente l'intensité du stimulus `#egg`, i_h celle de `#humidity` et i_l celle de `#light`, sa variation est donnée par la formule:

$$i_e := i_e + (\max(0.005 / i_h, 0.001)) + 0.003 * i_l.$$

Si, par contre, l'agent est transporté (ce qu'il sait car sa place est alors à `nil`), l'intensité du stimulus décroît régulièrement de `0.001` unité/cycle. Il s'agit donc d'un stimulus qui dénote non seulement la présence des oeufs, mais également leur inconfort (s'ils se trouvent placés à un endroit trop lumineux ou pas assez humide). Nous verrons qu'il sera lié à la tâche de transport d'oeufs possédée par les fourmis. La variation de l'intensité du stimulus `#cureEgg` est quant à elle identique à celle présentée dans la définition de la classe `CuringBehavior`, c'est-à-dire que ce stimulus augmente quand le niveau de soins de l'agent diminue. Le stimulus `#maturingEgg` obéit également à ce qui a été défini dans la classe `MaturingBehavior`. Les tâches que peuvent entreprendre les agents oeufs sont au nombre de deux. La tâche `Default`, qui effectue simplement un appel à `primPropagateStimuli` pour repropager les stimuli de l'agent (si leur intensité a changé). Et enfin, la tâche `MaturingEgg`, déclenchée lorsque l'agent a dépassé son âge limite. Elle fait simplement appel à la primitive `primMature` qui, si l'agent possède un niveau de soins suffisant, le transforme en larve et sinon en nourriture (en oeuf alimentaire). Les agents cocons sont semblables en tous points aux agents oeufs, à savoir qu'ils possèdent les trois même stimuli (`#cocoon`, `#cureCocoon` et `#maturingCocoon`) et les deux tâches `Default` (propagation des stimuli) et `MaturingCocoon` (changement d'état). La seule différence provient du calcul de la variation d'intensité du stimulus `#cocoon`, car les cocons ont besoin de reposer dans un endroit sec et sombre. Cette intensité est calculée par:

$$i_c := i_c + (\max(0.005 * i_h, 0.001)) + 0.003 * i_l.$$

Comme le stimulus `#egg`, `#cocoon` dénote l'inconfort des larves dans leur environnement et est donc lié de la même manière à la tâche de transport des cocons que possèdent les fourmis.

IV.3.2.2. LES LARVES

Les agents larves sont différents des autres agents du couvain dans le sens où ils ont besoin, en plus d'être soignés et transportés, d'être nourris. Les oeufs et les cocons vivent sur leurs réserves énergétiques, les larves non. Comme ce sont des agents immobiles⁵⁵, il faut donc qu'ils puissent être nourris par les agents mobiles que sont les fourmis. Les agents larves héritent de leurs super-classes abstraites quatre stimuli différents. Les trois premiers, `#larva`, `#cureLarva` et `#maturingLarva`, ont un rôle identique à ceux fournis aux oeufs et aux cocons. La variation d'intensité de `#larva` est calculée de la même manière que celle de `#egg`. En revanche, la variation de la force de `#maturingLarva` est calculée à la fois en fonction du temps et du niveau de nourriture de l'agent. Si ce dernier est très bas, la tâche est déclenchée avant le moment théorique où il devrait devenir cocon et l'agent meurt. Le quatrième stimulus, qui provient de `FeedingBehavior`, porte le nom de `#hungryLarva` et son intensité évolue avec la faim de l'agent. Il a un rôle double; premièrement, attirer des agents mobiles afin qu'ils viennent lui apporter de la nourriture (pour peu que ces agents mobiles soient capables de reconnaître ce stimulus et d'y réagir de façon adéquate); deuxièmement, déclencher la tâche `◇hungryLarva` au sein de la larve, tâche dont la seule fonction est de faire appel à la primitive `primEat`. S'il n'y a rien à manger sur la place, la tâche est sans arrêt déclenchée jusqu'à ce que de la nourriture y soit déposée.

IV.3.3. Les ouvrières et la reine

IV.3.3.1. IDENTIFICATION DES TACHES

L'identification des tâches fournies aux ouvrières et à la reine a tout naturellement suivi la décomposition des comportements qui servent à décrire le répertoire comportemental des fourmis réelles. Tous les comportements n'ont pas été cependant modélisés, et certains ont été simplifiés, non parce que le modèle ne s'est pas montré capable de les représenter, mais parce que leur non-représentation même a fait partie des hypothèses de travail que nous avions au départ. La part la plus considérable du budget temps d'une fourmi est consacrée à l'ensemble des comportements orientés vers les agents du couvain. Ces comportements sont naturellement liés aux besoins manifestés par les oeufs, larves et cocons. Le premier est appelé comportement d'agrégation/ségrégation, qui consiste à opérer une différenciation spatiale entre les agents de différents types, et à regrouper ensemble les agents semblables, en tenant compte, bien évidemment, de leurs préférences quant à l'exposition lumineuse et aux conditions hygrométriques de leur lieu de regroupement. Ce comportement, selon [Hölldobler et Wilson 1990], existe chez toutes les espèces de fourmis. Le second est appelé comportement de soins et regroupe tous les actes de nettoyage, léchage et toilettage effectués par les fourmis à l'endroit des oeufs, larves et cocons, actes que l'on regroupe sous le nom d'actes trophiques et de soins. Le troisième, qui est le comportement d'alimentation, ne s'effectue qu'envers les larves et le plus souvent de manière indirecte (selon [Corbara 1991], la pseudotrophallaxie à l'égard des larves est rarissime chez *E. Ruidum*), la fourmi transportant de la nourriture la dépose près de la larve, qui sait se nourrir seule. Le quatrième, enfin, qui consiste à tuer des oeufs ou des larves afin de se procurer de la nourriture, est un comportement un peu plus hypothétique dans le sens où, s'il a été observé chez de nombreuses espèces à l'encontre d'oeufs viables ou alimentaires (cf. [Hölldobler et Wilson 1990]), il est rarement observé à l'encontre des larves. Les autres comportements comprennent le toilettage individuel, les différents actes de nutrition et l'activité de fourragement. Le toilettage individuel correspond à un besoin propre à chaque fourmi, qui est exprimé dans le modèle par le stimulus `#cureAnt` (ou `#cureQueen` pour la reine), hérité de la classe `CuringBehavior`.

⁵⁵ En réalité, les larves sont capables de se déplacer, mais de manière extrêmement lente. Elles ne sont donc pas capables d'aller fourrager et les fourmis doivent venir leur apporter la nourriture.

L'alimentation est motivée par le stimulus individuel de faim (`#hungryAnt` ou `#hungryQueen`), hérité, lui, de la classe `FeedingBehavior`. Enfin, le fourragement, qui consiste à sortir du nid pour aller chercher de la nourriture, est motivé par les besoins nutritionnels des larves, des ouvrières et par la présence de nourriture, c'est-à-dire par les stimuli `#food` et `#hungryLarva`. De manière générale, les relations de la fourmi à son environnement sont marquées par une très nette photophobie des jeunes ouvrières, réaction qui s'atténue au fil du temps pour presque disparaître chez les vétérans, comme le souligne [Chauvin 1982] à propos des fourmis rouges. Le comportement d'exploration dans le nid est le plus souvent constitué d'un déplacement aléatoire ou brownien ponctué de courtes haltes. Nous l'avons simplifié en le considérant comme uniformément aléatoire. L'essentiel des tâches fournies aux ouvrières est présenté Figure II.7, sous la forme exacte que possèdent ces tâches lorsqu'elles sont éditées avec le `TaskBrowser`. Les paramètres des différentes primitives sont eux aussi indiquées de manière graphique, en suivant la représentation icônique des agents de la simulation.

<i>Propagateur</i>	<i>Stimulus</i>	<i>Séquence de Primitives</i>	<i>Interruption</i>
	egg		
	cure Egg		
	larva		
	cure Larva		
	hungry Larva		
	maturing Larva		
	cocoon		
	cure Cocoon		
	cure Ant		
	hungry Ant		
	killEgg		
	kill Larva		
	food		
	light		

Figure II.7 - Les tâches fournies aux ouvrières. Le tableau se lit de gauche à droite, en commençant par le propagateur du stimulus, son nom, suivi de la séquence de primitives qui définissent le comportement et de la primitive à effectuer en cas d'interruption.

IV.3.3.2. LA GESTION DES STIMULI

Il nous reste deux points à préciser avant d'aborder les expériences et les résultats expérimentaux. Le premier concerne le calcul des stimuli internes effectué par les agents fourmis. $\#_{\text{cureAnt}}$ et $\#_{\text{hungryAnt}}$ (et leurs homologues pour les agents reines) sont calculés automatiquement par les variations respectives du niveau de soins et du niveau d'alimentation des agents (voir Section IV.3.3.1.). Les stimuli $\#_{\text{ant}}$ et $\#_{\text{queen}}$ n'étant pas utilisés normalement, ils ne sont pas calculés. Les simulations qui les utilisent (comme celle présentée Section V.6.) préciseront leur valeur. Enfin, $\#_{\text{killEgg}}$ et $\#_{\text{killLarva}}$ ont un mode de calcul qui fait intervenir le niveau de nourriture de l'agent, mais uniquement quand celui-ci devient extrêmement faible. Il s'agit d'un stimulus à seuil, ce qui signifie qu'il est à zéro la plupart du temps, mais qu'il devient brusquement positif quand le niveau de nourriture est inférieure au dixième de la normale. Puisque les stimuli externes sont assimilés aux phéromones, on peut assimiler ce calcul à une brusque décharge hormonale.

IV.3.3.3. LA QUESTION DES INTERACTIONS ENTRE INDIVIDUS

Le second point à préciser est la réponse à la question qui ne manquera d'être posée à propos de ces comportements: *quid* des interactions inter-individuelles ? C'est effectivement une question intéressante, dans la mesure où les actes trophiques réciproques semblent jouer un rôle non négligeable dans le maintien de la socialité et dans la transmission d'informations chimiques chez beaucoup d'espèces [Jaisson 1993], le plus important étant sans doute l'acte au nom barbare de trophallaxie, qui consiste pour une fourmi à alimenter un autre individu en déglutissant une partie de son propre repas.

Il se trouve en fait que ces actes sont très peu observés chez *Ectatomma ruidum*, notamment pour l'alimentation [Corbara 1991]. Cela ne veut pas dire qu'ils n'existent pas, mais il semble que leur rôle ne soit pas aussi important que dans d'autres espèces. Nos agents ne possèdent donc pas de tâche qui puisse les amener à interagir de manière directe avec d'autres agents. Nous avons cependant prévu, dans une des tâches déclenchées de manière interne ($\diamond_{\text{cureAnt}}$), le cas où elle serait déclenchée par un stimulus exogène provenant d'un autre agent (rappelons ici que les agents ne discriminent pas les stimuli qui leur sont propres des stimuli qui proviennent de l'environnement, et que la tâche $\diamond_{\text{cureAnt}}$ pourrait être aussi bien déclenchée par le stimulus $\#_{\text{cureAnt}}$ de l'agent que par le stimulus d'une autre fourmi, propagé dans l'environnement). En effet, si l'on regarde la séquence de primitives qui la composent, on s'aperçoit que la première primitive consiste à suivre le gradient généré par $\#_{\text{cureAnt}}$, puis, une fois arrivé à la source de ce gradient, à toiletter l'individu qui le propage. Si le stimulus est endogène, l'agent se découvrira lui-même comme propagateur du stimulus et se toiletera (il interagira avec lui-même sans le savoir, par un phénomène d'autostimulation proche de ceux présentés dans [Hölldobler et Wilson 1990]). S'il est exogène, il ira toiletter l'agent qui en a besoin et qui, pour une raison quelconque (autre tâche en cours, etc...), n'exécute pas son comportement de toilettage. Dans toutes les expériences présentées dans les sections suivantes, cependant, le stimulus $\#_{\text{cureAnt}}$ sera interne à l'agent pour la raison que nous avons invoquée plus haut. Si nous prenons par contre le sens originel du mot interaction, qui signifie tout autant *influence réciproque* qu'action réciproque, nous obtenons un système extrêmement interactif, mais d'une manière qui nous apparaît plus indirecte que la précédente, laquelle se base exclusivement sur le contact physique. En effet, quand un des agents prend en charge une des tâches pour lesquelles il est sollicité par son environnement, il va non seulement accomplir le rôle fonctionnel qui lui est demandé, mais en même temps *agir* (de manière non-intentionnelle, certes) sur le comportement des agents qui l'environnent en diminuant leurs chances de choisir cette tâche et en leur offrant la possibilité d'en accomplir d'autres, tout simplement parce qu'il en *consommer*a la demande.

Prenons un exemple concret. Supposons qu'à un instant donné dans la colonie, certains oeufs manifestent le besoin d'être transportés. Ils le font savoir aux fourmis en propageant dans l'environnement leur stimulus $\#_{egg}$. Admettons maintenant que trois fourmis déclenchent la tâche correspondante parce qu'elles se trouvent toutes trois proches du tas d'oeufs. La première arrivée, appelons-la A, va prendre en charge les oeufs et les soulever pour les transporter vers un lieu plus propice à leur épanouissement. Le stimulus $\#_{egg}$ va donc disparaître de l'environnement, ce qui aura comme conséquence immédiate que, n'ayant plus de gradients à remonter, les deux autres fourmis interrompent immédiatement leur tâche. A aura donc agi sur le comportement de ses deux congénères qui pourront maintenant vaquer à d'autres occupations, en leur faisant savoir, à distance et non-intentionnellement, que la tâche n'avait plus besoin d'être accomplie. Ce qui précède est bien entendu vérifié pour toutes les autres tâches, à la condition expresse qu'elles soient consommatrices (cf. Chapitre III).

Les sections suivantes, consacrées aux expérimentations effectuées avec MANTA vont nous permettre de démontrer que ce type d'interactions, réalisées de manière non-intentionnelle, est à la base d'une structure globale de coopération similaire à celle observée dans les colonies naturelles.



CHAPITRE V

EXPERIMENTATIONS

V.1. Protocoles d'expérimentation

V.1.1. Description de la plate-forme de simulation

La plate-forme de simulation utilisée est en fait composée de deux parties logicielles indépendantes. Une partie dite de modélisation (ou de maquettage), une partie de simulation proprement dite.

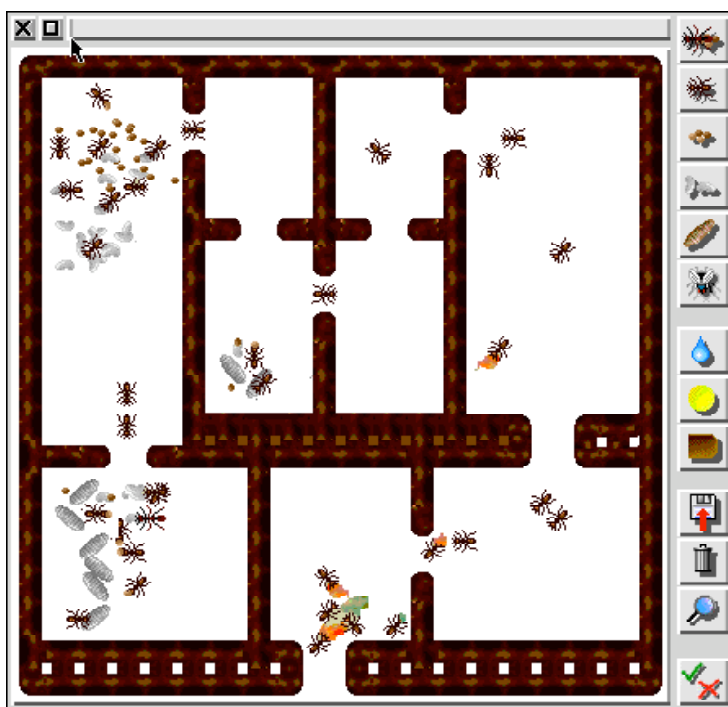


Figure II.8 - La fenêtre SimulationWindow sert à générer des situations initiales.

La partie de modélisation est la version écrite en Smalltalk 80 4.0, qui a été présentée dans les sections précédentes. Outre les outils de programmation comme le `TaskBrowser`, elle offre la possibilité de créer (également de manière graphique) des modèles d'expérimentation, c'est-à-dire des situations initiales dont l'évolution est ensuite simulée. La création de ces modèles se réalise par l'intermédiaire d'une interface appelée `SimulationWindow` (voir Figure II.8), qui permet de générer des configurations particulières à partir desquelles seront conduites les expérimentations, et également de faire fonctionner certaines expérimentations. Cependant, Smalltalk étant un langage interprété et non compilé, le processus de simulation prend énormément de temps. Dans certaines situations, nos fourmis artificielles deviennent même moins véloces que les fourmis réelles, ce qui induit des durées de simulation par trop contraignantes. Pour remédier à ce problème dans le cas de simulations importantes, une version écrite en Turbo-Pascal⁵⁶ et reprenant l'intégralité du code écrit en Smalltalk a été développée. Le facteur d'accélération obtenu oscille entre trente et soixante, et cette version permet

⁵⁶ Cette version a été écrite par S. Lalande et porte le nom de MANTA_P. La version écrite en Smalltalk a été rebaptisée MANTA_S. Il existe une version, MANTA_C, réalisée par A. Touret en C++, mais qui n'intègre qu'une partie de ce que nous présentons.

d'obtenir une fourmière adulte (voir Section V.2.) en environ deux jours et demi, ce qui reste raisonnable. Bien entendu, MANTA_P est presque dépourvue de toute interface graphique et ne permet ni de modifier les tâches, ni de générer dynamiquement de nouvelles situations initiales. Le compromis retenu pour bénéficier des facilités de Smalltalk et de la rapidité d'un langage compilé a donc consisté à utiliser MANTA_S comme outil de modélisation et MANTA_P comme outil de simulation. Le lien entre les deux est réalisé par un fichier⁵⁷, que l'on peut sauvegarder lors de la création d'une configuration avec la `SimulationWindow` et relire au démarrage de la version en Turbo-Pascal. Il ne fait pas de doute, qu'à l'avenir, l'intégration entre les deux devra être plus complète, pour aboutir très certainement à une fusion (peut-être par l'intermédiaire de la plate-forme de développement MALEVA, actuellement développée par Marc Lhuillier pour accueillir la plupart des travaux réactifs de l'équipe MIRIAD du LAFORIA, et qui intègre déjà EMF et MANTA_S).

V.1.2. Les protocoles d'expérimentation

Toutes les expériences qui vont être présentées dans la suite de ce chapitre ont été réalisées avec la même version du logiciel, sauf celles qui précisent les modifications qui y ont été apportées. L'environnement des agents est constitué de la reproduction des nids en plâtre utilisés en laboratoire (Figure II.9). Le gradient d'humidité est généré par le placement, dans l'angle supérieur gauche de la chambre du couvain, d'un agent `HumidityAgent` dont le stimulus possède une force de 40 et un décrétement de 1. La sortie est quant à elle marquée par un agent `LightAgent` dont le stimulus est de 30, mais dont le décrétement est de 5 (il ne se propage que sur six cases), ce qui laisse le nid suffisamment obscur. La colonie est approvisionnée en nourriture tous les 26000 cycles, ce qui signifie, sachant qu'un cycle est égal à 7,5 secondes en temps réel (voir Tableau II.1), tous les deux jours et demi. Le nombre d'unités de nourriture placées à l'extérieur du nid est calculé au prorata du nombre d'agents mobiles dans la colonie, ce qui est le cas en laboratoire. Théoriquement, une quantité de 50 unités de nourriture est suffisante pour nourrir un agent entre deux réapprovisionnements. La nourriture fournie dans la pratique est un plus abondante et consiste en 4 `FoodAgent` comportant chacun une capacité nutritive de 30 unités (ne pas oublier que la nourriture se détériore lentement).

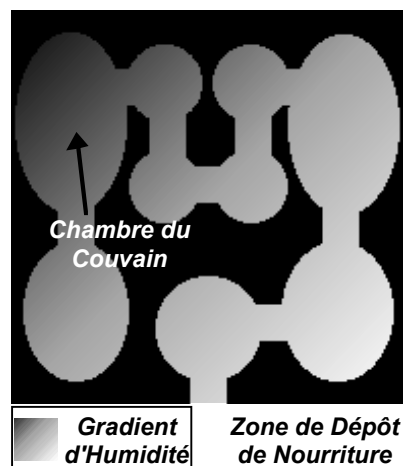


Figure II.9 - Les nids en plâtre utilisés en laboratoire. Le gradient d'humidité est généré par une réserve d'eau placée en haut à gauche de la chambre du couvain, la porosité du plâtre lui permettant de se distiller dans tout le nid de manière différentielle.

⁵⁷ Appelé fichier de configuration, il contient la description du nid et de tous les agents qui s'y trouvent.

Les paramètres dépendants de données biologiques sont présentés dans le Tableau II.1. Certains de ces paramètres sont directement dérivés de données physiologiques, d'autres ont été déduits, en accord toutefois avec les éthologues, d'après les résultats des expérimentations préliminaires, par un jeu d'essais et d'erreurs dont une description peut être trouvée dans [Lalande 1992]. Nous ne nous attarderons donc pas sur cette partie, qui est cependant délicate à maîtriser. Le seul déplacement aléatoire des agents lorsqu'ils effectuent la tâche $\diamond_{\text{default}}$ suffit en effet à rendre les expériences non-reproductibles.

Paramètres des Simulations Entreprises avec Manta

	Unité	FoodAgent	EggAgent	LarvaAgent	CocoonAgent	AntAgent	QueenAgent
maturingTime	cycle	25000	253440	212428	334080	4204800	12614400
	temps réel	2 j ± 5h	22 j ± 4h	18 j ± 24h	31 j ± 48h	1 an ± 20 j	3 ans ± 60 j
minCureLevel	unité	-	0	3	0	3	3
maxCureLevel	unité	-	30	30	30	30	30
cureDecrement	unité/cycle	-	0,003	0,002	0,0005	0,002	0,002
minFoodLevel	unité	-	-	0	-	0	0
maxFoodLevel	unité	30	-	30	-	30	30
foodDecrement	unité/cycle	0,0001	-	0,002	-	0,002	0,0025
layEggsTime	cycle	-	-	-	-	-	6776
Taille des Cases = 1 cm x 1 cm					Durée d'un Cycle = 7,5 secondes		

Tableau II.1 - Les paramètres biologiques et topographiques définis dans MANTA.

Dernier point, l'espérance de vie d'un agent est fixée à une valeur moyenne, également d'après les chiffres fournis par les éthologues. Elle n'est pas influencée par l'état de l'agent *précédent* (par exemple, celui de l'œuf qui a donné la larve). Elle est par contre dépendante de l'état de soins et de nutrition de l'agent considéré (par l'intermédiaire du stimulus $\#_{\text{maturing}}$ + type de l'agent). Les derniers paramètres qu'il nous reste à présenter sont relatifs aux poids, seuils et incréments des tâches, qui vont représenter le profil comportemental de nos agents. Ils sont rassemblés sur le Tableau II.2, où *AntAgent* et *QueenAgent* se taillent la part du lion (le couvain n'a en effet que très peu de tâches de définies). Comme nous l'avons vu (cf. Chapitre III), l'incrément de poids sert à indiquer un phénomène de renforcement, ou d'habituation à la tâche. Les tâches que nous avons choisi de renforcer sont toutes les tâches liées aux besoins physiologiques manifestées par le couvain et par les fourmis elles-mêmes, qui servent aux éthologues pour déterminer les profils comportementaux. Les tâches qui ne sont pas renforcées sont celles qui correspondent à des stimuli internes (comme $\#_{\text{killEgg}}$ ou $\#_{\text{killLarva}}$) dont l'existence est hypothétique. Enfin, la seule tâche envers laquelle les ouvrières manifestent une habituation est \diamond_{light} , ce qui permet de reproduire le phénomène de photophobie chez les jeunes ouvrières.

Les résultats expérimentaux présentés à partir de la section suivante proviennent de l'analyse de données brutes extraites des simulations. Ces données sont sauvegardées à intervalle régulier dans des fichiers prévus à cet effet. La version de base de MANTA ne sauvegarde que les données concernant l'évolution démographique de la colonie et le fichier de configuration. Il est cependant possible de sauvegarder d'autres informations, comme par exemple la liste contenant pour chaque agent la tâche qu'il effectue à un instant t , ou l'évolution des poids des tâches au sein des agents. L'ajout d'une nouvelle sauvegarde s'effectue dans MANTA_S en ajoutant à la simulation un nouvel agent instance de *SpyAgent*, classe qui offre la possibilité à ses instances de sauvegarder sur fichier les résultats de n'importe quel traitement demandé par l'utilisateur. Un espion est donc instancié avec un nom de

fichier, une fréquence d'activation (qui va servir à installer un démon dans le `ProcessManager`), et un bloc contenant le traitement à réaliser et prenant en argument la liste des agents du système. Dans MANTA_P, la correspondance n'est pas tout à fait réalisée, puisque l'ensemble des sauvegardes que l'on souhaite réaliser doit être programmé dans une fonction spéciale gérée par le noyau.

Paramètres des Tâches	EggAgent			LarvaAgent			CocoonAgent			AntAgent			QueenAgent		
	poids	seuil	incr.	poids	seuil	incr.	poids	seuil	incr.	poids	seuil	incr.	poids	seuil	incr.
egg	-	-	-	-	-	-	-	-	-	4	15	1	4	15	1
cureEgg	-	-	-	-	-	-	-	-	-	4	15	1	4	15	1
maturingEgg	4	15	0	-	-	-	-	-	-	-	-	-	-	-	-
killEgg	-	-	-	-	-	-	-	-	-	4	15	0	4	15	0
larva	-	-	-	-	-	-	-	-	-	4	15	1	4	15	1
cureLarva	-	-	-	-	-	-	-	-	-	8	30	1	8	30	1
maturingLarva	-	-	-	4	15	0	-	-	-	4	15	1	4	15	1
hungryLarva	-	-	-	4	15	0	-	-	-	4	15	1	4	15	1
killLarva	-	-	-	-	-	-	-	-	-	4	15	0	4	15	0
cocoon	-	-	-	-	-	-	-	-	-	4	15	1	4	15	1
cureCocoon	-	-	-	-	-	-	-	-	-	8	30	1	8	30	1
maturingCocoon	-	-	-	-	-	-	4	15	0	4	15	1	4	15	1
food	-	-	-	-	-	-	-	-	-	4	80	0	4	80	0
light	-	-	-	-	-	-	-	-	-	0,75	10	-1	0,75	10	-1
cureAnt	-	-	-	-	-	-	-	-	-	2	15	1	-	-	-
maturingAnt	-	-	-	-	-	-	-	-	-	4	15	0	-	-	-
hungryAnt	-	-	-	-	-	-	-	-	-	2	15	0	-	-	-
cureQueen	-	-	-	-	-	-	-	-	-	-	-	-	8	30	1
maturingQueen	-	-	-	-	-	-	-	-	-	-	-	-	4	15	0
hungryQueen	-	-	-	-	-	-	-	-	-	-	-	-	4	15	1
layEggsQueen	-	-	-	-	-	-	-	-	-	-	-	-	4	15	0

Tableau II.2 - Les poids des tâches définies pour toutes les classes d'agent de MANTA. Les cases grisées correspondent aux tâches qui n'existent pas.

V.2. Sociogenèse

V.2.1. Définition

On regroupe sous le nom de sociogenèse⁵⁸ l'ensemble des étapes qui caractérise la génération d'une nouvelle société jusqu'à ce qu'elle atteigne sa taille adulte, c'est-à-dire, pour *Ectatomma ruidum* entre 30 et 100 ouvrières. Chez les hyménoptères sociaux, la génération d'une nouvelle colonie peut se réaliser soit par essaimage, soit par fondation [Wilson 1985]. L'essaimage⁵⁹ est le processus durant lequel, lorsque l'effectif d'une colonie devient trop important, elle se divise en deux parties [Chauvin 1982] : la plus vieille reine part avec la moitié de la société et va fonder une autre colonie, cependant que l'autre moitié reprend le cycle avec une nouvelle reine élevée entre-temps. Dans le processus de fondation, qui semble majoritaire chez les fourmis [Jaisson 1993], la reine qui vient d'être fertilisée (par un mâle) quitte le nid afin de générer seule une nouvelle société. Cette fondation peut être à son tour de deux types différents: claustrale ou semi-claustrale. Dans le cas d'une fondation claustrale, la reine ne sort pas du nid et nourrit les larves, jusqu'à l'apparition des premières ouvrières, grâce aux réserves internes que lui fournit la dégénérescence des muscles de ses ailes. Dans le cas d'une fondation semi-claustrale, que l'on observe chez *Ectatomma ruidum* et chez la majorité des Ponérines, la reine est obligée de quitter le nid pour rechercher de la nourriture afin de se nourrir et de nourrir les larves. De plus, comme l'a montré [Corbara 1991], la reine continue souvent d'aller fourrager après l'apparition des premières ouvrières.

⁵⁸ Ce nom a été proposé pour la première fois par [Wilson 1985] dans l'idée de montrer l'équivalence entre ce phénomène et celui de morphogenèse.

⁵⁹ Qui existe essentiellement chez les abeilles, et semble pratiqué minoritairement chez les fourmis.

V.2.2. L'intérêt de la sociogenèse

D'un point de vue éthologique, la sociogenèse de sociétés naturelles permet l'étude de la génération des structures sociales présentes dans les colonies adultes, dans la mesure où le nombre d'individus est relativement faible, et que l'augmentation se fait lentement. Un exemple d'une telle étude se trouve dans [Corbara 1991]. De notre point de vue, elle offre deux grands intérêts:

- La génération de la société adulte, si nous ne nous préoccupons pas dans un premier temps de sa structuration, est la première occurrence d'"émergence" au niveau global. Il serait en effet surprenant d'étudier l'évolution d'une société sans d'abord essayer de comprendre comment elle est née et comment elle s'est développée. Nous pensons de plus (et ceci semble confirmé par les expériences menées sur la polygynie, voir Section V.6.) que les premières étapes de sa fondation conditionnent fortement son évolution subséquente.
- Le second intérêt à simuler le processus de sociogenèse réside dans le fait que nous obtenons de la sorte des colonies "réellement artificielles", dont la composition et la structure ne dépendent que des phénomènes d'auto-organisation qui se déroulent pendant leur fondation. De cette manière, les modèles de société adulte que nous obtenons ne sont ni "traduites" de sociétés adultes naturelles (ce qui introduirait un biais important dans leur paramétrage et réduirait considérablement nos possibilités d'expérimentation, dans la mesure où le nombre de sociétés d'*Ectatomma ruidum* totalement étudiées de cette manière est assez faible) ni arbitrairement fixées (à la différence de celles utilisées par exemple par [Theraulaz 1991], où tous les individus commencent, déjà adultes, avec la même expérience et le même profil comportemental).

D'un point de vue expérimental, nous allons aborder la comparaison des sociétés naturelles et artificielles sous l'angle de leurs évolutions démographiques respectives, à savoir l'évolution du nombre d'oeufs, de larves, de cocons et d'ouvrières. Ceci pour deux raisons principales. La première est que l'évolution démographique des sociétés *Ectatomma ruidum* (c'est également le cas pour d'autres espèces, dont *Neoponera Apicalis*, étudiée par [Fresneau et Dupuy 1988], qui fait partie de la même famille) est assez caractéristique et offre des ressemblances frappantes d'une société à l'autre. La dynamique de fondation apparaît dans ce cas comme une "signature" de l'espèce. La seconde est qu'il existe une relation circulaire entre l'évolution de la démographie du nid et la division du travail de la société. D'une part, les fluctuations du couvain ont un effet déterminant sur les profils comportementaux des ouvrières. D'autre part, le comportement de la reine et de ses filles va avoir un effet non moins déterminant sur le taux de production des oeufs et la réussite de leur développement. De cette manière, même si elle ne fournit aucune information sur la division du travail réalisée, l'étude démographique de la colonie fournit l'indication que les agents ont su effectuer les tâches nécessaires au développement biologique du couvain et à leur propre subsistance. De plus, contrairement aux sociogrammes qui présentent un instantané de la société pendant une courte période, l'évolution démographique nous permet de comparer la dynamique à long terme de cette genèse. Enfin, pour la raison que nous avons indiquée plus haut, elle offre la possibilité de vérifier immédiatement le gain quantitatif obtenu lors de l'arrivée des premières ouvrières, et ainsi de quantifier les phénomènes de coopération qui résultent de la division du travail entre tous les membres de la colonie. L'évolution quantitative du couvain dans les sociétés naturelles offre ainsi une dynamique typique qui peut se caractériser par une diminution de la quantité d'oeufs lorsque les larves apparaissent, et une augmentation des maxima de chacune des courbes du couvain au fur et à mesure que les ouvrières apparaissent. Un exemple d'une telle évolution est présentée Figure II.10, dans le cadre d'une société naturelle étudiée en laboratoire par Bruno Corbara.

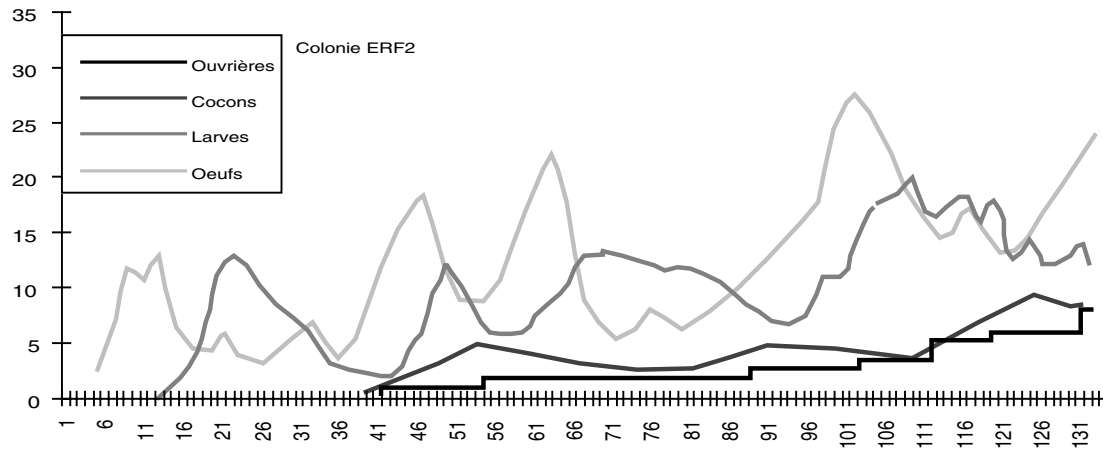


Figure II.10 - Évolution démographique de la colonie naturelle ERF2. L'axe des abscisses représente le temps (une unité valant deux jours). D'après les données de [Corbara 1991].

V.2.3. Les expériences de Sociogenèse

V.2.3.1. RESULTATS EXPERIMENTAUX

Comme indiqué dans la section consacrée aux protocoles d'expérimentation, les expériences de sociogenèse artificielle ont été conduites dans un nid qui reproduit exactement les nids en plâtre utilisés par les éthologues pour mener leurs expérimentations de sociogenèse naturelle (voir Figure II.9). Toutes les expériences ont été conduites à partir de paramètres égaux (les poids initiaux des tâches, les paramètres biologiques, etc.). Ce qui signifie que la seule source d'imprédictibilité est représentée par le déplacement aléatoire par défaut des agents fourmis et reines. La configuration initiale des différentes expériences consiste à placer une reine seule dans le nid vide (avec de la nourriture). Une expérience est arrêtée dès qu'une des deux situations suivantes est atteinte:

- La reine meurt (de faim). La fondation étant monogyne, ceci implique, à moyen terme, la mort de la colonie et donc la fin du processus.
- La population du nid dépasse trente ouvrières. Ce stade représente une bonne approximation du succès de la fondation, sachant qu'une colonie naturelle qui atteint ce stade continue normalement à se développer.

Cas d'Échecs	Composition	Nombre	Pourcentage
1	Oeufs	8	6,06%
	Oeufs, Larves	16	12,12%
	Larves	27	20,45%
	Oeufs, Larves, Cocons	24	18,18%
	Larves, Cocons	16	12,12%
	Oeufs, Cocons	2	1,52%
	Larves, Ouvrières	10	7,58%
2 + 3 + 4 + 5 + 7	Avec Larves	93	90,29%
Échecs		103	78,03%
Succès		29	21,97%
Expériences		132	100,00%

Tableau II.3 - Proportion d'échecs et de succès des sociogenèses. Le pourcentage des taux d'échecs en présence de larves est calculé par rapport au nombre total d'échecs.

Le Tableau II.3 montre, à partir des 132 expériences⁶⁰ que nous avons analysées, la proportion de réussites et d'échecs. Les cas d'échecs sont répertoriés en sept catégories, qui chacune correspondent à la composition de la population quand la reine vient à mourir. Dans ces expérimentations, le taux d'échecs, 78,3%, apparaît proche de celui observé dans les colonies de laboratoire (où il s'élève à 86%, soit 10 sociogénèses réussies sur 72 expériences). Il est dû en grande partie aux difficultés compréhensibles rencontrées par la reine dans l'accomplissement simultané de plusieurs tâches antinomiques ou difficilement conciliables, comme par exemple, s'occuper du couvain et aller fourrager. Ceci est corroboré par l'analyse de la composition de la population du nid lorsque la reine meurt. Dans 90,29% des cas d'échecs, en effet, la population du couvain comporte des larves.

Or, si l'on regarde les tâches nécessitées par le couvain, on s'aperçoit que, alors que les oeufs et les cocons ne déclenchent que deux tâches chacun, les larves en déclenchent trois, dont une (\Diamond hungryLarva) qui oblige l'agent qui l'exécute à sortir du nid pour chercher de la nourriture. La vie de la reine durant les premières étapes de la fondation est en effet entièrement consacrée au couvain et, de ce fait, se déroule presque exclusivement dans la chambre du couvain, ce qui lui permet de réagir rapidement à toutes les sollicitations dont elle est l'objet. Avant que les larves n'arrivent, les forces qui s'exercent sur son comportement (force étant compris au sens de directions spatiales pourvues d'intensités différentielles) sont ainsi quasi-exclusivement centripètes, c'est-à-dire à même de contraindre la reine à rester dans le voisinage de la chambre du couvain. L'arrivée des larves va donc déstabiliser cet état de fait en générant une nouvelle dynamique centrifuge.

Intensités des Forces Centrifuges et Centripètes s'exerçant sur la Reine

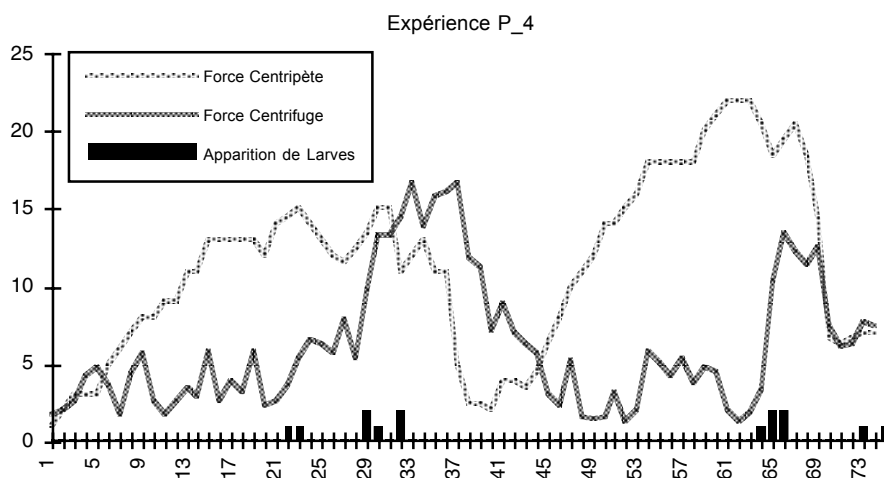


Figure II.11 - L'évolution respective des forces centripètes et centrifuges qui s'exercent sur la reine durant les deux premiers mois de la sociogénèse. L'exemple pris ici est celui d'une sociogénèse réussie (expérience P_4).

L'intensité de ces deux forces à chaque instant et en tout point où se trouve la reine peut être grossièrement approximée en additionnant les valeurs des niveaux d'activation de chacune des tâches qui y contribuent. En effet, ces valeurs nous fournissent à un instant donné l'intensité des sollicitations auxquelles doit répondre la reine conjuguée à ses propres prédispositions à y répondre. Pour la force centripète, ces tâches sont \Diamond egg, \Diamond cureEgg, \Diamond cocoon, \Diamond cureCocoon, \Diamond larva et \Diamond cureLarva.

⁶⁰ En réalité, le nombre d'expériences est de 300. La moitié a servi de "génératrices" de sociétés et n'a pas été analysée. Si le nombre d'expériences paraît faible, il est à comparer aux 72 expériences menées par [Corbara 1991].

Pour la force centrifuge, ce sont \Diamond hungryQueen, \Diamond hungryLarva et \Diamond food. Ont été laissées de côté les tâches qui n'intervenaient que marginalement dans ces intensités ou de manière trop périodique (comme \Diamond layEggsQueen, par exemple). La Figure II.11 fournit un exemple (calculé d'après l'évolution de la colonie P_4) des variations de l'intensité respective de chacune de ces forces.

En analysant ces variations, on s'aperçoit que tant que la colonie ne comporte que des oeufs (stade que nous appellerons stade 1, et qui dure, selon les expériences, de deux à trois mois), l'essentiel des forces qui s'exercent sur la reine est centripète, hormis sa propre faim qui va la pousser à sortir de temps à autre, mais qui pourra être également assouvie par cannibalisme. La composante centrifuge exercée par la nourriture est négligeable à l'intérieur du nid. Le stade 2, qui voit l'apparition des premières larves, et donc la diminution automatique du nombre d'oeufs, rééquilibre les deux forces et déstabilise en conséquence le comportement de la reine, écartelée entre la chambre du couvain, où elle doit soigner les larves, et l'extérieur du nid, où elle doit aller chercher de la nourriture.

L'exercice qu'elle mène alors est assez délicat, car toute augmentation du couvain (par ponte) augmente la force centripète, mais également la force centrifuge, dans la mesure où les larves non nourries réclament de plus en plus de nourriture que la reine, occupée dans la chambre à couvain, va de plus en plus rarement chercher. Et, quand elle obéit aux forces centrifuges, elle diminue les chances de survie du couvain. Nous verrons plus loin, dans l'analyse des résultats expérimentaux positifs, quelles mesures drastiques lui permettent d'échapper à ce cercle infernal. Le stade 3, marqué par l'apparition des premiers cocons, est caractérisé par un retour à la normale, c'est-à-dire une nette augmentation de la force centripète au détriment de la force centrifuge. En revanche, ce stade est caractérisé par un nombre assez important de tâches à effectuer, dans l'hypothèse où, la reine continuant à pondre, les trois familles du couvain sont présentes simultanément. Bien évidemment, la naissance des premières ouvrières va permettre à ces forces de se répartir sur un nombre plus important d'individus et, en conséquence, de moins faire sentir, sur chacun d'entre eux, leurs effets contradictoires. Un grand pas est également franchi quand les individus prennent en charge de manière exclusive chacune des composantes de ces forces, ce qui réduit leur effet déstabilisateur à néant (voir la Section V.4.).

Les cas d'échecs sont donc causés par l'obligation, pour l'individu qu'est la reine, d'effectuer seul et simultanément des tâches matériellement contradictoires mais absolument nécessaires au maintien en vie de tous les agents qui composent le couvain. Celles-ci peuvent être si urgentes que la reine en oubliera de se nourrir.

A l'inverse, le phénomène qui se déroule dans *toutes* les sociogenèses réussies peut être analysé de deux manières complémentaires. La première est celle de l'observateur, qui a toutes les chances d'y déceler une stratégie sous-jacente, stratégie que nous qualifierions d'émergente dans la mesure où elle n'est programmée dans aucun des comportements fournis aux agents de la simulation. La seconde est celle du concepteur, qui connaît les comportements programmés, et qui est capable d'expliquer quels sont, au niveau individuel, les comportements qui permettent à cette "stratégie" de survenir. Afin de comprendre en quoi elle consiste, nous allons regarder de près les résultats obtenus avec deux des sociétés qui ont réussi (voir Figure II.12) et qui présentent une dynamique d'évolution beaucoup plus marquées que les autres expériences, ce qui nous permet d'en montrer clairement les caractéristiques. Sur cette figure, la courbe en gris clair représente l'évolution de la population d'oeufs, celle en gris des larves, celle en gris foncé des cocons et celle en noir des ouvrières.

Contrairement à l'idée intuitive que nous pourrions avoir de l'évolution idéale du système, le nombre d'oeufs, de larves et de cocons ne reste absolument pas constant, mais semble au contraire dépendre de fonctions périodiques croissantes et alternées. La courbe des oeufs est une bonne illustration.

Elle croît de manière continue jusqu'à l'apparition des premières larves, se met alors à décroître assez brutalement⁶¹, et croît de nouveau lorsque les larves disparaissent pour laisser la place à des cocons. Ses phases successives de croissance semblent synchronisées avec les phases de décroissance de la courbe des larves, qui, elle-même, apparaît synchronisée avec la courbe des cocons. Et, au fur et à mesure de l'évolution de la colonie, les maxima atteints par les trois courbes du couvain augmentent, de même que leurs minima.

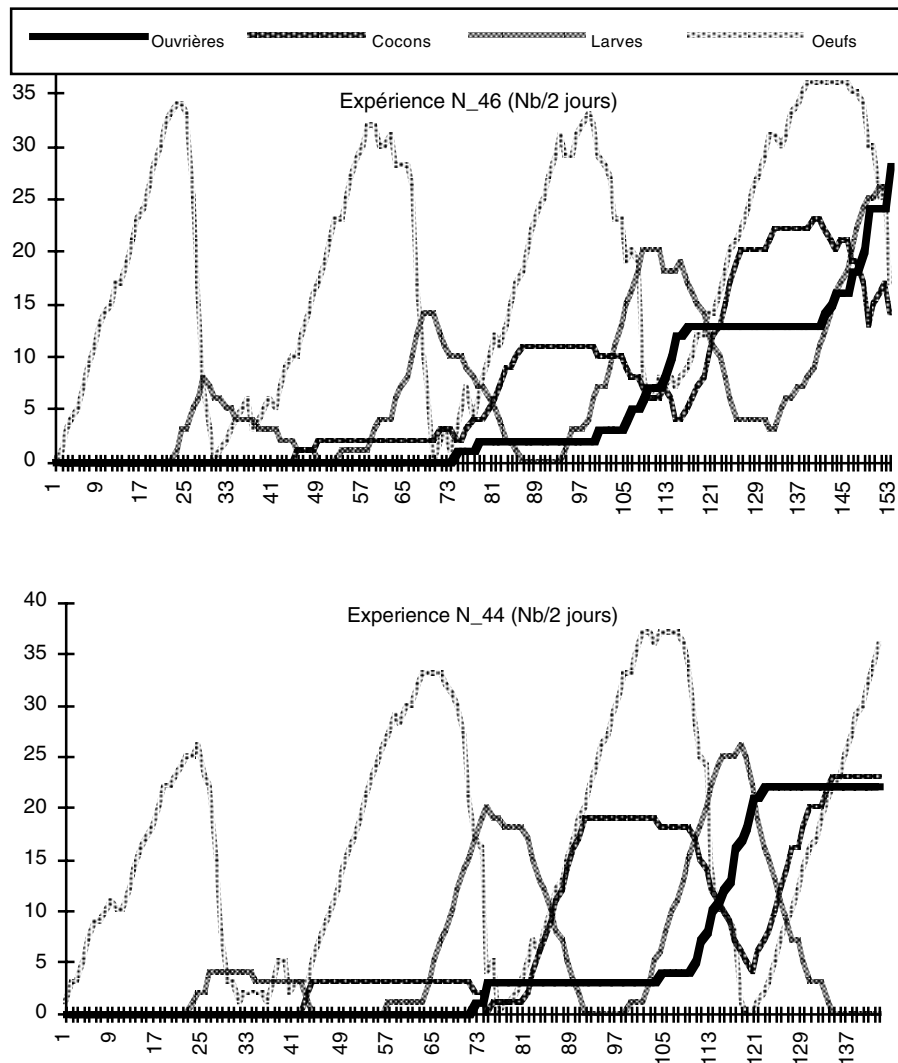


Figure II.12 - Evolution démographique des colonies N_46 et N_44. Une unité de l'axe des abscisses est égale à deux jours sur l'échelle de temps de la simulation.

Ce qui est intéressant dans cette évolution est que la configuration dynamique obtenue ici n'est à aucun moment présente de manière symbolique (c'est-à-dire, codée) dans le comportement des agents. Elle apparaît uniquement sur la base des interactions mutuelles existant entre les agents et leur environnement, aussi bien social que spatial.

⁶¹ Seuls 20% des oeufs de la première génération se transforment en larves. Ce pourcentage augmente régulièrement à partir de la deuxième génération pour atteindre 60 à 70%.

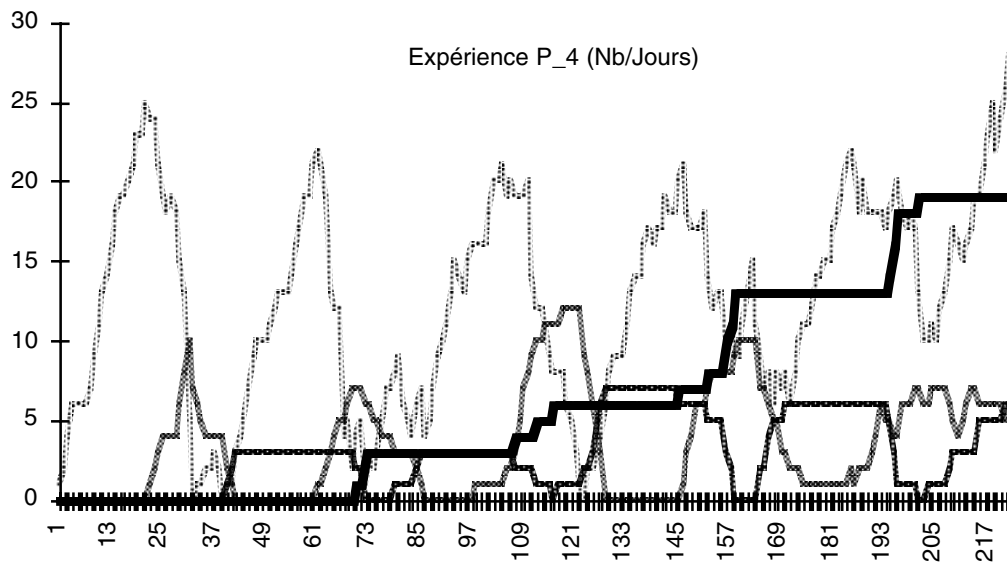


Figure II.13 - Evolution démographique de la colonie P_4. Une unité de l'axe des abscisses représente un jour sur l'échelle de temps de la simulation.

Evidemment, cette évolution, qui présente une similitude troublante avec les évolutions des colonies naturelles (cf. Figure II.10), peut être vue comme un effet de bord d'une situation particulière. Il est vrai que certaines expériences ne possèdent pas la même dynamique de fondation. A titre d'exemple, la Figure II.13 dépeint l'évolution de la colonie P_4, où l'augmentation des maxima atteints par chaque courbe du couvain est loin d'être aussi claire que dans les précédentes. Cependant, la Figure II.14, qui présente la moyenne, sur les 160 premiers jours, des évolutions de 29 colonies, est là pour attester que leur dynamique tend vers un même attracteur. Certes, les configurations ne sont pas aussi nettes que sur la Figure II.12, mais ceci est dû en partie aux effets déformants du calcul de moyenne.

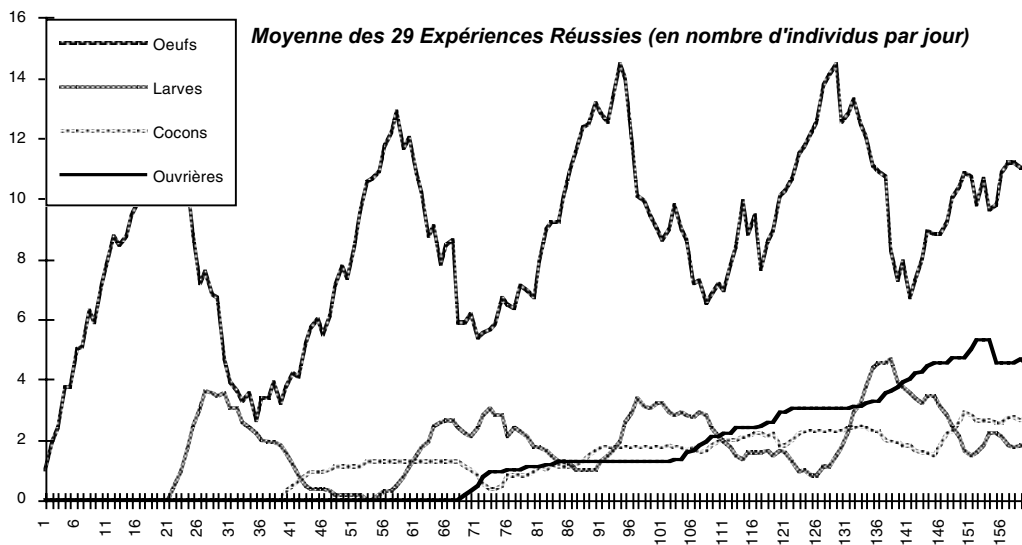


Figure II.14 - Courbes de population moyennes des 29 colonies dont la sociogenèse a été couronnée de succès.

Il nous reste maintenant à expliquer quels sont les comportements mis en oeuvre qui permettent à la reine de pourvoir aux besoins du couvain ainsi qu'aux siens pendant les premiers stades du développement. En effet, si l'on adopte le point de vue d'un observateur ignorant de la conception du système, il y a fort à parier que l'on détectera dans les évolutions de la population du couvain les effets d'une stratégie particulièrement habile de contrôle des naissances (voir notre article sur les stratégies émergentes [Drogoul 1993]). Pourquoi en sera-t-il presque certainement ainsi ? Tout simplement, selon [Resnick 1992], parce que l'homme ne peut s'empêcher, dans nombre de cas, de raisonner de manière centralisée et d'attribuer aux systèmes qu'il observe un contrôle "intelligent". Ce n'est d'ailleurs pas fortuitement, et nous ne sommes pas les seuls à le faire remarquer (lire l'introduction de [Seger 1993]) que l'individu reproducteur de la colonie s'appelle la *reine* (avec ce que cela implique comme structure hiérarchique, voir par exemple ce type d'analyse dans [Reeve et Gamboa 1987]).

Si la dynamique de fondation était différente d'une colonie à l'autre, nous ne serions certainement pas tenté d'y voir l'oeuvre d'un *deus ex machina*. Mais il se trouve que les variations inter-coloniales sont très peu significatives, ce qui ne laisse à l'observateur que deux solutions: ou bien il se trouve face à un comportement rationnel, capable de planifier sur le long terme une évolution démographique équilibrée, ou bien ce qu'il observe est le produit d'un comportement réactif, parfaitement adapté à la situation dans laquelle il opère. Si maintenant nous reprenons le rôle de concepteur, connaissant la manière dont ont été codés les agents qui évoluent dans la colonie, nous pouvons d'ores et déjà éliminer la première hypothèse. Mais nous la gardons en mémoire, car c'est la première démonstration... d'intelligemence (voir Introduction).

En effet, le plan, ou plutôt la résultante de la séquence d'actions mises en oeuvre par la reine durant la sociogenèse consiste, du moins jusqu'à l'arrivée des premières ouvrières, à faire en sorte de ne jamais avoir plus de deux types d'agents du couvain à élever simultanément. Ceci se traduit par un cannibalisme important (car la reine ne peut pas contrôler sa ponte), cannibalisme qui permet d'obtenir des générations (d'oeufs, de larves, de cocons et donc d'ouvrières) espacées d'environ un mois, ce qui n'est pas loin du décalage optimal. Bien sûr, l'acte de cannibalisme n'est pas choisi en fonction de ce critère, mais en fonction de la situation de manque de nourriture à laquelle est confrontée la reine.

En revenant aux données concernant les colonies prises une à une, il est aisé de détecter le gain de stabilité apporté par l'augmentation du nombre d'ouvrières. Bien que la fréquence de variation des différentes populations du couvain ne change pas réellement, l'arrivée des ouvrières modifie par contre radicalement l'amplitude de ces variations, surtout celles des larves et des cocons. Le nombre moyen d'oeufs/jour pondus par la reine n'évoluant pas, il faut donc en conclure que les oeufs ont une chance plus importante de devenir larve quand la colonie contient des ouvrières. L'explication est bien sûr liée à la répartition des tâches au sein de la colonie entre la reine et les ouvrières. Si nous reprenons la Figure II.11, qui s'arrêtait juste avant l'apparition de la première ouvrière dans la colonie P_4, et que nous l'étendons maintenant jusqu'au stade de 10 ouvrières, en centrant toujours le calcul sur les forces centripètes et centrifuges qui s'appliquent sur la reine, voici ce que l'on obtient:

Intensités des Forces Centrifuges et Centripètes s'exerçant sur la Reine

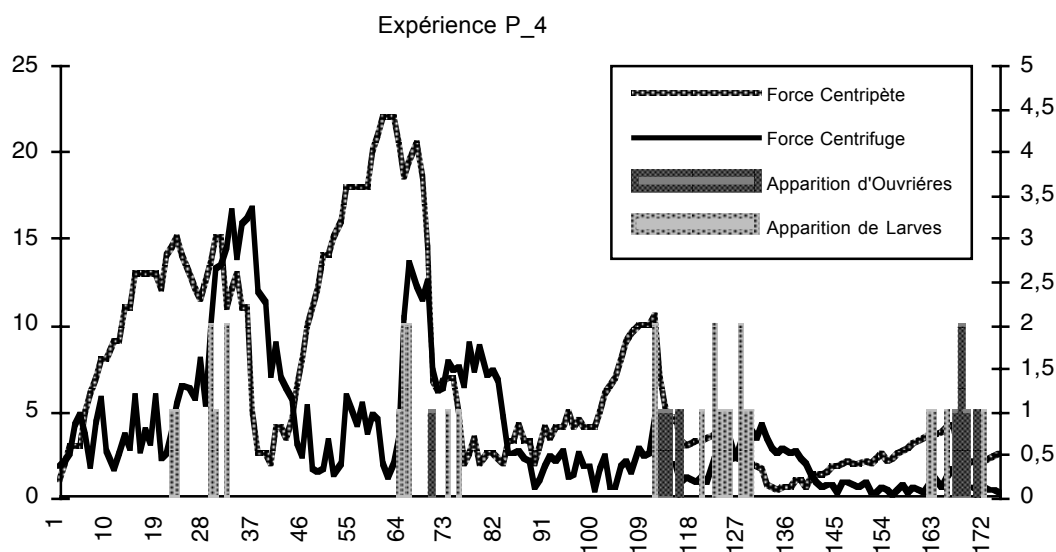


Figure II.15 - Intensité des forces centrifuges et centripètes s'exerçant sur la reine au cours des six premiers mois de la sociogénèse.

On voit clairement que l'intensité des forces qui s'exercent sur la reine diminue au fur et à mesure de l'arrivée de nouvelles ouvrières. Ce n'est bien sûr pas un résultat extrêmement suprenant mais il montre clairement qu'une bonne partie des tâches, qui auparavant déstabilisaient la reine, sont maintenant prises en charge par les nouvelles venues. Il faut cependant être conscient du fait que ceci ne signifie pas forcément qu'une structure sociale de *division* du travail soit à l'oeuvre. Il se peut en effet que les agents possèdent tous le même profil comportemental, et que la stabilisation que l'on observe ne soit que le résultat d'une *répartition* uniforme de la charge de travail (un point semblable est développé dans [Theraulaz 1991]). Mais nous y reviendrons lors de l'analyse de la structuration sociale de la colonie. Toujours est-il que, l'intensité des sollicitations diminuant, la reine ne se trouve plus dans une situation d'urgence et peut s'occuper pleinement du couvain ou aller fourrager. Les tâches qu'elle n'effectue plus (ou moins longtemps) sont prises en charge par d'autres individus (si ce n'était pas le cas, les niveaux d'activation de ses tâches seraient en effet toujours aussi hauts).

V.2.3.2. DISCUSSION

Que peut-on conclure de ce premier ensemble d'expériences ? Essentiellement deux points, qui reprennent en partie certaines des idées développées dans les paragraphes précédents.

Premièrement, la dynamique de sociogénèse caractéristique des colonies naturelles d'*Ectatomma ruidum* a pu être reproduite assez fidèlement dans MANTA. L'évolution quantitative est comparable, de même que l'évolution qualitative. Cette dernière est en effet marquée par des périodes de forts accroissements du couvain, suivies d'une brusque rétraction lors de l'apparition de larves. L'amplitude de ces deux phénomènes est tempérée par l'apparition des premières ouvrières, ce qui correspond parfaitement aux données expérimentales de laboratoire mises à notre disposition [Corbara, comm. pers.]. Il s'agit donc ici d'une validation partielle, d'une part de la justesse des comportements individuels fournis aux agents, et d'autre part du modèle comportemental choisi qui leur permet de s'actualiser, puisque la reine et les ouvrières se montrent visiblement capables de

prendre en charge l'ensemble des tâches nécessaires à la survie de la colonie (essentiellement: fourragement, soin du couvain). Cette validation ne demeure pour l'instant que partielle, la répartition des tâches entre les individus n'ayant pas encore été étudiée (voir Section V.4.).

Deuxièmement, la performance adaptative globale de la société à l'égard de son environnement est réalisée sans l'intervention d'aucun contrôle centralisé. Nous nous trouvons donc là devant un exemple de système adaptatif fonctionnant sur la base d'une intelligence en essaim [Beni et Hackwood 1992], ce qui est bien le moins, d'ailleurs, pour une colonie de fourmis. Mais la question que nous pourrions nous poser est la suivante: ce système est-il réellement *adaptatif*, ou n'est-il que particulièrement bien *adapté* à des contraintes environnementales particulières (apport de nourriture, configuration du nid, etc.) ? Le danger est grand, en effet, d'avoir tellement bien réalisé le couplage agents/environnement lors de l'initialisation des différents paramètres biologiques, pour que nous nous trouvions face à un système qui ne puisse actualiser ses capacités que dans ce type d'environnement précis.

L'exemple des fourmis réelles est en effet là pour nous montrer, aussi bien dans le cas particulier d'*Ectatomma ruidum* que dans d'autres espèces, qu'elles sont capables d'adapter le fonctionnement global de leurs colonies à des contraintes environnementales différentes, alors même que les comportements individuels ne changent pas. Il suffit de songer seulement à l'extraordinaire performance que constitue pour les colonies le fait de s'acclimater parfaitement à un nid de laboratoire en plâtre lorsqu'elles proviennent d'un milieu naturel riche comme la forêt amazonienne. Leur capacité à recouvrer en très peu de temps des comportements normaux est d'ailleurs soulignée par [Hölldobler et Wilson 1990].

Bien sûr, de même que pour la génération de formes sociales spécifiques, l'adaptabilité d'une population à son environnement peut-être abordée sous deux angles complémentaires; d'une part, les processus évolutifs qui permettent de comprendre la coévolution de la population et de son environnement et d'autre part les processus adaptatifs qui, sur une échelle de temps beaucoup moins importante, permettent d'étudier les phénomènes d'acclimatation (cf. l'exemple de l'acclimatation particulièrement réussie de la fourmi de feu en Amérique sur une période d'un peu moins de quarante ans, cité dans [Chauvin 1982]). Nous nous intéresserons uniquement, comme nous l'avons déjà précisé, à cette seconde forme d'adaptation, non pas que la première ne présente pas d'intérêt, mais parce qu'elle déborde du cadre purement ontogénétique de nos travaux. Il convenait de le préciser car le terme adaptation possède à l'heure actuelle une connotation majoritairement évolutionniste. Nous avons réalisé pour cela une expérimentation dont les résultats sont analysés dans la Section suivante.

V.3. Auto-régulation de la population en fonction de contraintes environnementales

V.3.1. Sur le concept d'éthologie synthétique

Le travail que nous menons avec le projet MANTA, et dont nous venons de passer en revue les premiers résultats, peut être à la fois compris comme faisant partie d'une problématique de *simulation éthologique*, ou, en inversant les perspectives, d'*éthologie synthétique*. L'éthologie synthétique (ou... synthéthologie), telle qu'elle est définie par [Mac Lennan 1991], consiste en effet à synthétiser un monde artificiel rempli d'animaux artificiels, permettant d'étudier de manière plus simple que dans la réalité certains phénomènes (Mac Lennan s'intéresse à la genèse des communications inter-individuelles), pour la simple raison que l'on en contrôle tous les paramètres et que l'on y observe de manière transparente les comportements mis en oeuvre. En fait, et bien que le projet puisse sembler utopique, il s'agit juste d'une extension "computationnelle" du travail des éthologues, qui formulent des

hypothèses, échafaudent des théories sans toujours pouvoir les vérifier expérimentalement, et qui créent eux-mêmes des situations totalement artificielles, comme par exemple les expériences de sociotomie présentées dans [Lachaud et Fresneau 1987]. Il n'est donc pas totalement irréaliste de penser que, d'un simple outil de simulation d'une réalité existante, l'on puisse passer à un outil de synthèse de situations hypothétiques, qui se baserait sur la validité du premier pour établir une sorte de "validité inverse", de l'artificiel vers le vivant. On se rend compte des formidables potentialités que conférerait cet outil aux éthologues, qui pourraient alors jouer au jeu du "et si...": et si les fourmis ne mangeaient pas leurs œufs ? et si les ouvrières naissaient plus vite ? etc. afin de vérifier l'influence de l'ajout ou du retrait de certains comportements sur la colonie. Cependant, et c'est pourquoi la phrase précédente est au conditionnel, il convient d'être très prudent quant aux interprétations que l'on pourrait éventuellement faire de telles expériences. Un tel outil est certainement dans la logique de notre approche, mais il nécessite de devoir avancer à petits pas, en jouant par exemple sur des paramètres que nous sommes certains de pouvoir faire varier de manière isomorphe dans la réalité afin de vérifier la similitude de dynamique entre le processus synthétisé et le processus réel.

V.3.2. Restriction de la nourriture

L'expérience que nous présentons dans cette section est une expérience extrêmement simple de synthéthologie dont le but est de montrer les facultés d'adaptation des colonies de MANTA à une contrainte environnementale qui consiste en une sévère restriction de la nourriture⁶². Comme nous l'avons en effet précisé, la quantité de nourriture fournie aux colonies dans les expériences est fonction du nombre d'ouvrières. C'est d'ailleurs ce qui se passe, avec peut-être moins de précision, en laboratoire. De cette manière, la colonie a théoriquement assez de ressources pour nourrir toutes ses ouvrières. La situation dans leur milieu naturel n'est cependant pas aussi idyllique. La quantité de nourriture est beaucoup plus variable et soumise à des variations (variations saisonnières, compétition avec d'autres colonies, etc.) qui ne peuvent être contrôlées ni par la colonie, ni par l'expérimentateur. Ce sont les conditions "normales" dans lesquelles vivent les colonies de fourmis et auxquelles elles doivent faire face, soit en multipliant les sorties, soit en régulant leur évolution démographique de manière à ajuster leur population aux ressources dont elles disposent. De manière à vérifier si notre modèle pouvait reproduire cette capacité de "contrôle des naissances", nous avons conduit un ensemble de sociogénèses dans lesquelles la nourriture est limitée à un certain nombre d'ouvrières. Elle augmente donc normalement à chaque naissance d'ouvrière (et diminue à chaque mort) jusqu'à ce que cette limite soit atteinte. Dans toutes les expériences présentées ici, cette limite a été fixée à 10, nombre qui n'a pas été choisi au hasard, mais qui résulte d'observations que nous n'avons pas la place de développer ici, qui montrent que le nombre de 10 ouvrières constitue un moment clé dans le développement de la colonie, dans le sens où elle commence à acquérir une certaine stabilité sociale.

Les résultats, illustrés par la Figure II.16, qui décrit l'évolution de la colonie P3_2, révèlent une tendance étonnante. Elle peut être interprétée comme une sorte de "sagesse" à l'égard des ressources environnantes, et est partagée par environ 90% des expériences réalisées. Elle se traduit par une augmentation normale des effectifs des ouvrières jusqu'à 10 (ou un peu plus), puis une décrue brutale jusqu'à 4, 5 ou 6 ouvrières, où la population se stabilise. Certaines expérimentations ont été stoppées prématurément et donnent donc, comme P3_2, une impression de stabilisation à la moitié des possibilités démographiques théoriques de la colonie.

⁶² La première version de cette expérimentation résultait d'une erreur de programmation dans une des versions de MANTAP. Nous nous en sommes rendus compte au bout d'un laps de temps assez important, quand nous a paru étrange le fait que les sociogénèses se répartissent de manière égale en deux catégories dont la dynamique respective d'évolution était très différente. D'autant plus que la première dynamique, sur l'ensemble des quatorze ordinateurs utilisés, n'était observée que sur les machines portant les numéros de 1 à 7, alors que la seconde ne consentait à apparaître que sur les numéros 8 à 14. Pour tout dire, ce cas d'adaptation nous semblait un peu au-delà des capacités du logiciel...

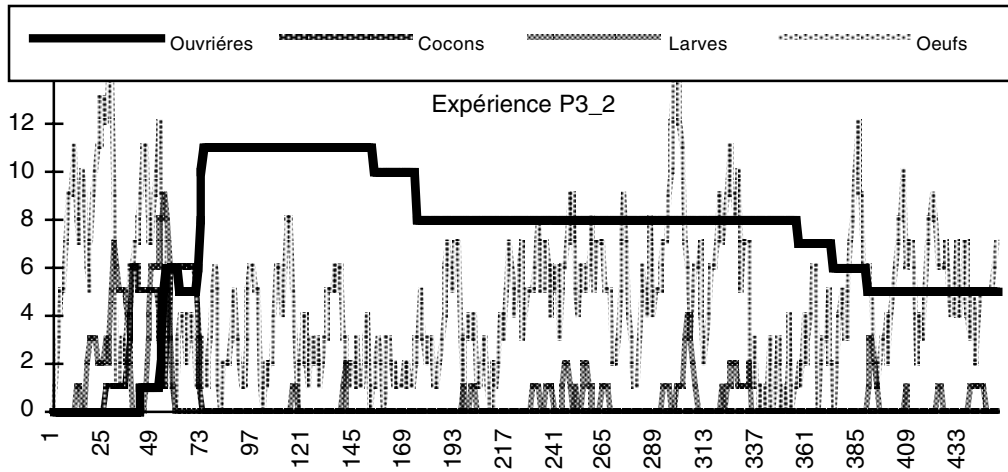


Figure II.16 - Evolution démographique de la société P3_2, dont l'approvisionnement en nourriture est limité à 10 ouvrières. Une unité sur l'axe des abscisses correspond à 2 jours.

Néanmoins, en poursuivant les expériences un peu plus loin, il apparaît que cette stabilisation est en trompe-l'œil et qu'il s'agit d'un équilibre dynamique. Ceci est illustré par la Figure II.17, sur laquelle sont représentées les évolutions démographiques respectives des colonies P11_2 et P11_4. Elles montrent clairement, sur une période d'environ un an et demi, les "tentatives" faites par la colonie pour poursuivre son évolution normale, et les retours immédiats à un effectif moindre.

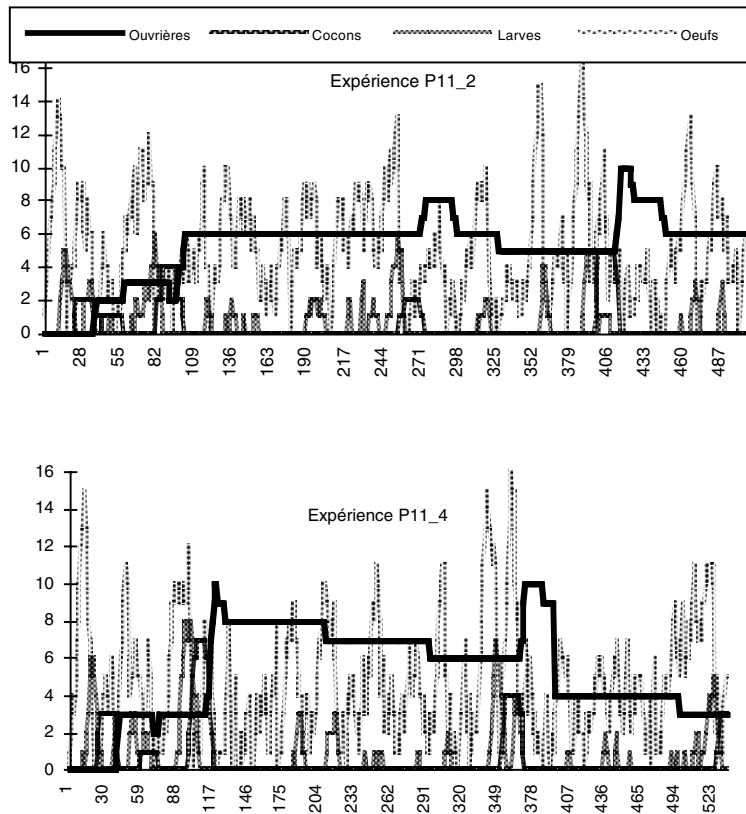


Figure II.17 - Les évolutions démographiques des colonies P11_2 et P11_4, dont l'approvisionnement en nourriture est limité à 10 ouvrières. Une unité sur l'axe des abscisses correspond à 2 jours.

Il est intéressant de noter que la population ouvrière reste stable sur d'assez longues périodes, mais que cette stabilité masque en fait une dynamique très importante de la population du couvain. Les effectifs des oeufs, larves et cocons varient en effet de manière substantielle d'un jour à l'autre. Ce qui est certain, en tout cas, c'est que le simple changement du paramètre d'approvisionnement a complètement bouleversé l'évolution normale de la colonie. L'explication comportementale qui peut en être donnée est la suivante:

- Tant que le nombre d'ouvrières ne dépasse pas la limite fixée par l'expérimentateur, l'évolution de la colonie reste la même que dans les sociogenèses normales. A dire vrai, il n'y a pas de raisons objectives qu'elle soit différente.
- Dès que cette limite est atteinte (ce qui signifie, dans notre cas de figure, qu'il y a dix ouvrières plus la reine pour dix rations de nourriture), un phénomène similaire à celui observé lors de l'arrivée des premières larves apparaît. Les besoins en nourriture de la société dépassent ses ressources et de nombreux oeufs et larves sont tués afin de servir d'aliments. Ceci réduit automatiquement la chance qu'avaient ces oeufs de devenir larves, puis cocons, et enfin ouvrières. Comme les oeufs alimentaires (qui sont consommés en priorité par rapport aux larves) ne suffisent pas, dans bien des cas, à assouvir les ouvrières, elles passent plus de temps à chercher de la nourriture qu'à prendre soin du couvain, ce qui en diminue également la quantité, et certaines finissent même par mourir de faim, en restant enfermées dans un processus de recherche de nourriture à travers tout le nid.
- L'arrêt brutal de l'approvisionnement constitue donc un phénomène catastrophique qui désorganise brutalement la colonie pendant une période importante, période suivie par une stabilisation en douceur. Ceci est sans doute dû à l'inertie importante de l'évolution du couvain, et également au fait que, en-deçà de 10 ouvrières, la colonie n'est pas très stable. Quand un nombre suffisant d'ouvrières a disparu, la demande alimentaire devient plus raisonnable (la plupart des larves étant également mortes). Mais les coupes effectuées dans la population du couvain durant cette période vont empêcher la colonie d'avoir de nouvelles ouvrières avant un certain temps, qui correspond à plusieurs cycles de maturation du couvain. Après quoi, l'arrivée des nouvelles ouvrières fait repartir le processus.

En dépit du fait que ces résultats ne constituent pas vraiment une surprise, ils confirment un point important sur lequel nous reviendrons dans la troisième partie, qui est la robustesse de tels systèmes décentralisés. Nous n'avons pas moyen, ici, de juger si le système est *plus* robuste qu'une organisation centralisée (autour de la reine, par exemple), mais nous voyons qu'à partir d'un certain seuil, il sait compenser les pertes individuelles et retrouver un état d'équilibre. De plus, nous pouvons constater que l'absence de prise de décision globale n'empêche pas l'apparition d'une réponse globale, qui affecte l'ensemble des individus. Le réajustement du "régime" de la société constitue un bon exemple d'une telle réponse; il n'est décidé nulle part, la contrainte qui le fait naître n'est ressentie qu'au niveau local par chacun des individus, et cependant, en comparant l'évolution de toutes les sociétés soumises à cette contrainte environnementale, on voit apparaître un schéma de réajustement similaire d'une société à l'autre, qui sait de plus s'adapter aux conditions particulières de chacune. Il nous reste donc à voir quelle est la forme de l'organisation sociale qui permet à la colonie de manifester une telle adaptabilité (voir plus précisément Section V.4.3.4). Comment cette organisation se crée à partir d'interactions entre individus qui ne peuvent la penser, comment elle se transforme en prenant en compte les nouveaux arrivants ou au contraire les départs, et surtout de quelle manière elle influe sur les comportements individuels pour assurer sa pérennité - et sa stabilité.

V.4. Organisation Sociale de la Colonie

L'étude de l'organisation sociale d'une colonie de fourmis est toujours centrée sur deux aspects: la division du travail et les relations de hiérarchie entre les individus, l'un et l'autre étant dans la plupart des cas interdépendants. Les colonies d'*Ectatomma ruidum* ne semblent pas posséder de structure hiérarchique (d'après [Corbara 1991]), nous nous limiterons donc dans cette partie à l'étude de la division du travail entre les individus faisant partie d'un même nid. Le phénomène de division du travail peut être abordé sous deux angles différents, tous deux complémentaires, puisqu'ils répondent respectivement au comment et au pourquoi de son existence. La première approche consiste à étudier la manière dont elle émerge et dont elle se maintient au sein du groupe de fourmis, en mettant l'accent sur les processus d'ontogénie individuels. La seconde approche consiste à étudier son intérêt fonctionnel, c'est-à-dire à mettre en évidence les problèmes qu'elle permet à la colonie de résoudre. Avant d'aborder ces questions essentielles à la compréhension de la dynamique sociale de nos colonies artificielles, et, partant, de celle des colonies naturelles, nous allons présenter, dans la section suivante, les formes de division du travail obtenues dans deux des sociétés que nous avons étudiées. Nous nous baserons ensuite sur ces exemples pour développer les deux approches.

V.4.1. Résultats Expérimentaux

La répartition des agents au sein de la structure globale de division du travail a été étudiée de manière isomorphe à celle utilisée par les éthologues pour étudier la répartition des ouvrières (voir Chapitre IV). Nous avons ainsi, pendant des périodes de durée fixe (typiquement, une semaine, soit 80650 cycles), enregistré à intervalles de temps réguliers (tous les 50 cycles, soit toutes les six minutes) les activités de chacune des fourmis (identifiable grâce à un numéro), en sauvegardant sur fichier le nom de la tâche que chacune d'entre elles effectuait. Cette sauvegarde nous a permis d'avoir en quelque sorte l'emploi du temps de chaque agent, et, donc, ce que les éthologues appellent son profil comportemental. Nous n'avons malheureusement pas la place de trop développer ce point ici, mais le passage du profil comportemental exprimé sous forme de tâches au profil comportemental manipulé par les éthologues a nécessité des mises au point très délicates. La sauvegarde des tâches que nous effectuons est en effet à la fois trop et pas assez détaillée. Trop, car un agent qui aura déclenché une tâche même s'il est encore éloigné du propagateur du stimulus sera considéré par nous comme effectuant cette tâche, et par les éthologues comme ne faisant rien, puisqu'ils n'ont pas accès à son "état interne". Ceci est particulièrement sensible pour les tâches nécessitant de grands déplacements dans le nid, comme les tâches de fourragement ou `hungryLarva`. Pas assez, car un agent effectuant la tâche `default` sera vu par nous comme ne faisant rien, alors qu'il pourra, sur une photographie, donner l'impression d'effectuer une tâche, s'il est par exemple très près du couvain. La mise au point des correspondances entre tâches et catégories comportementales nécessitera certainement dans l'avenir que nous sauvegardions une partie du contexte en plus du nom de la tâche. Quoi qu'il en soit, le profil comportemental individuel obtenu est ramené à quatre catégories d'activités (identiques, moins l'activité de garde du nid que nous n'avons pas prise en compte, à celles utilisées par [Corbara et al. 1991]) qui sont, respectivement, les activités de soins aux oeufs, de soins aux larves, de soins aux cocons et de fourragement. Chacune de ces activités regroupe un certain nombre de tâches comme, par exemple, `egg` et `careEgg` pour le soin aux oeufs. La première constatation, si nous mettons simplement les individus côte à côte, est qu'ils présentent des profils comportementaux hétérogènes, ce qui signifie que les tâches ne sont pas également réparties entre eux. En analysant la Figure II.18 (appelée sociogramme éclaté), qui représente par souci de clarté une époque de la société P_5 ne comportant que 7 ouvrières, on voit par exemple clairement que l'ouvrière n°7 possède un profil fortement orienté vers le soin aux oeufs (auquel elle consacre 45% de son budget-temps), alors que l'ouvrière n°1 ne lui consacre que 11% de son temps. Cette figure se lit comme suit: les activités

retenues sont placées en lignes, et les individus en colonnes (par ordre d'apparition dans la colonie). Le profil comportemental d'une ouvrière se lit donc verticalement. Chaque histogramme représente l'importance en pourcentage de l'activité dans le profil de l'individu considéré limité à ces quatre activités, la moyenne théorique pour chaque activité étant donc de 25%. La partie de l'histogramme dépassant cette moyenne, qui indique une spécialisation, est marquée en noir.

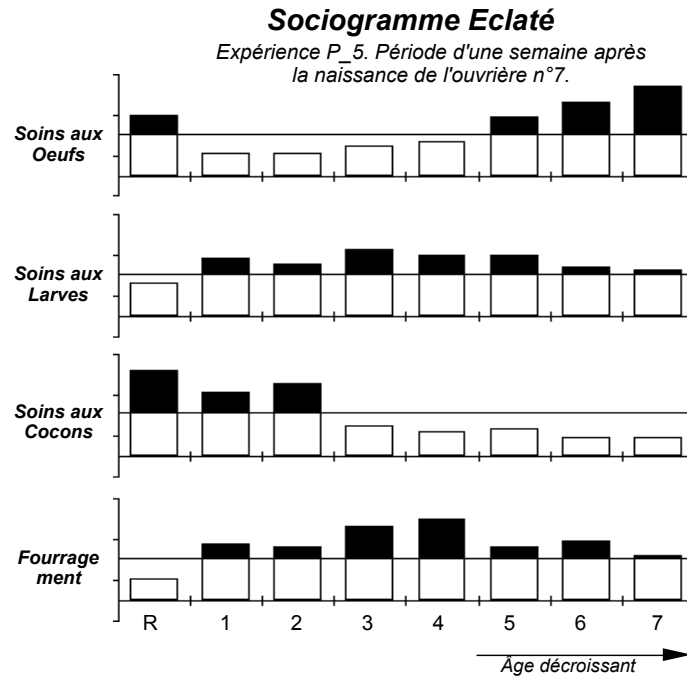


Figure II.18 - Sociogramme éclaté de la colonie P_5. Profils comportementaux des huit individus de la colonie par rapport aux activités de soins aux oeufs, aux larves, aux cocons et de fourrage ment.

La deuxième constatation est que, malgré l'hétérogénéité apparente qui se dégage de cette figure, il est possible de trouver des profils proches, comme ceux des ouvrières n°3 et 4, ou encore ceux des ouvrières n°1 et 2. Ce regroupement de profils, opéré à l'aide de méthodes d'analyse de données comme l'analyse factorielle des correspondances [Lenoir et Mardon 1978] (voir également Chapitre IV), va ainsi nous permettre de déterminer des groupes fonctionnels qui seront chacun liés à une ou plusieurs activités, groupes qui serviront à caractériser la structure de division du travail de la société. La Figure II.19 présente un exemple typique d'une telle répartition en groupes fonctionnels et correspond à l'étude de la colonie P_6 pendant une semaine à partir de la naissance de l'ouvrière n°14. Nous avons arbitrairement choisi de considérer quatre groupes à partir de la classification que nous a fourni l'analyse factorielle des correspondances (sur l'aspect méthodologique concernant la détermination des groupes, voir [Corbara et al. 1991]). Le profil de ces groupes est obtenu en moyennant les profils individuels des agents qui le composent. Nous pouvons les décrire comme suit:

- Groupe 1 (la reine + 2 ouvrières) - Soigneuses d'œufs: l'activité de soins aux œufs monopolise 45% du temps consacré aux activités considérées. Le reste des actes est à peu près réparti entre les autres activités.
- Groupe 2 (4 ouvrières) - Soigneuses de larves: cette activité accapare 47% du budget temps du groupe. Le reste est réparti entre le soin aux cocons et le fourrage ment.
- Groupe 3 (3 ouvrières) - Soigneuse de cocons: les ouvrières de ce groupe passent en moyenne 45% de leur temps à s'occuper des cocons.
- Groupe 4 (5 ouvrières) - Fourrageuses: contrairement aux groupes précédents, celui-ci consacre 43% de son temps en moyenne à aller fourrager.

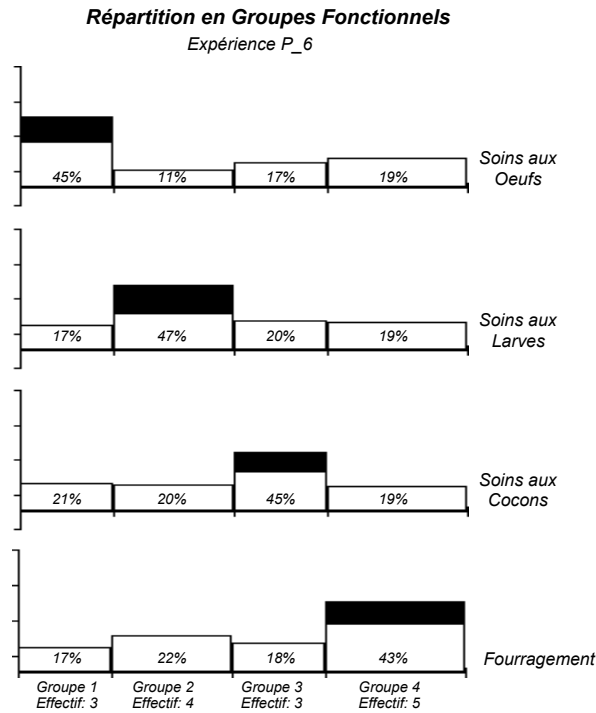


Figure II.19 - Sociogramme regroupé de la colonie P_6. Répartition de ses 15 membres en 4 groupes fonctionnels. Les activités sont disposées en lignes, et les profils comportementaux de chaque groupe en colonnes. Le pourcentage indiqué dans chaque histogramme indique son importance dans le profil comportemental du groupe.

Bien entendu, les répartitions en groupes fonctionnels des différentes colonies ne sont jamais tout à fait semblables. La Figure II.20 fournit la division du travail observée au sein de la colonie P_5 (cf. Figure II.18), pendant une semaine, après la naissance de l'ouvrière n°16.

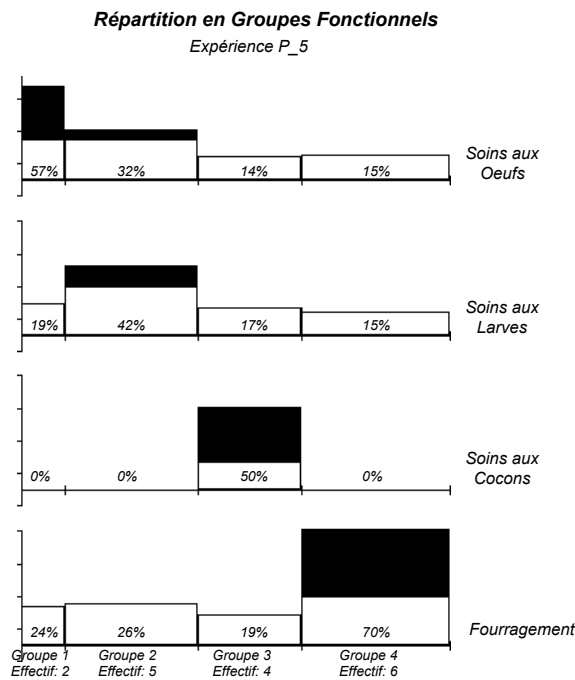


Figure II.20 - Sociogramme regroupé de la colonie P_5. Répartition de ses 17 membres en 4 groupes fonctionnels.

L'analyse de ce sociogramme nous fournit les groupes suivants:

- Groupe 1 (la Reine + 1 ouvrière) - Soigneuses d'oeufs: 57% de leur temps est consacré aux oeufs, le reste étant réparti entre le soin aux larves et le fourragement.
- Groupe 2 (4 ouvrières) - Généralistes/Soigneuses de larves: ce groupe consacre 32% de son temps au soin des oeufs, ce qui en fait le deuxième groupe spécialisé, 42% au soin des larves et 26% au fourragement.
- Groupe 3 - (4 ouvrières) - Soigneuses de cocons: ce groupe monopolise à lui seul tous les soins aux cocons, qui représentent 50% de son activité, le reste étant réparti de manière homogène..
- Groupe 4 - (6 ouvrières) - Fourrageuses: il se distingue des trois précédents par le fait qu'il consacre 70% de son temps au fourragement. Parmi les ouvrières qui le composent, certaines y consacrent plus de 90%. Le soins aux oeufs et le soin aux larves se partagent de manière égale les 30% restants.

V.4.2. Comparaison avec les sociétés naturelles

Sur l'ensemble des colonies obtenues par simulation, quatorze ont été analysées de cette manière (dont P_5 et P_6). Si le nombre paraît faible, il correspond au nombre de colonies naturelles à partir desquelles [Corbara 1991; Corbara, Fresneau et Lachaud 1986] ont étudié l'organisation sociale d'*Ectatomma ruidum*. En comparant leurs sociogrammes avec ceux obtenus à partir de colonies naturelles, nous pouvons souligner trois points importants.

Premier point, l'organisation générale de la colonie apparaît très proche, tout au moins pour les quatre activités considérées. Nous obtenons la même "diagonalisation" typique de la matrice groupes/activités, les effectifs de chaque groupe semblent cohérents avec les données expérimentales des éthologues, et, même si ceci ne figure pas sur les sociogrammes présentés, nous obtenons une proportion équivalente (entre 20% et 30%) d'ouvrières inactives. Enfin, le comportement de la reine, c'est-à-dire son appartenance systématique au groupe des soigneuses d'oeufs (dont elle constitue le pilier) et ses faibles performances dans les activités extérieures, est tout à fait semblable au comportement des reines des colonies naturelles.

Cependant, et c'est notre deuxième point, la distinction entre les groupes n'est pas toujours aussi franche que celle qui existe dans la nature. Si nous regardons le sociogramme regroupé de la colonie P_6 (Figure II.19), en effet, le temps alloué par chaque groupe aux activités dans lesquelles il n'est pas spécialisé représente systématiquement plus de la moitié de son budget-temps, et est de plus équitablement réparti entre ces trois activités. Si le sociogramme de P_5 vient apporter un bémol à cette constatation, il convient de préciser que les colonies dont la structure sociale est proche de celle de P_6 représentent presque 60% des cas que nous avons étudiés (exactement 8 colonies sur les 14 dont les données ont été dépouillées).

Le troisième et dernier point concerne la part très importante de l'activité de fourragement dans l'activité générale des ouvrières, part que l'on ne retrouve pas dans les colonies naturelles. Ce point peut cependant s'expliquer assez facilement par la distorsion qu'entraîne la traduction des tâches en activités. Il semble en effet que dans nombre de cas, nous considérons qu'un individu va fourrager parce qu'il effectue une des tâches regroupées dans cette activité, alors qu'une photographie le montrerait déambulant dans le nid sans but apparent. Il se peut également que les paramètres biologiques concernant les besoins en nourriture des agents aient été exagérés. Enfin, la mortalité des individus sortant du nid n'est pas plus importante, ce qui ne correspond pas à l'idée intuitive que l'on

peut se faire de ces colonies dans la nature. Les dangers de toutes sortes sont en effet beaucoup plus importants à l'extérieur qu'à l'intérieur du nid, et ceci n'a pas été pris en compte dans notre système. Il est probable que de nouvelles expérimentations devront être effectuées afin de vérifier l'incidence des ces trois points sur l'importance du fourragement.

Cependant, même en conservant ce dernier point, il apparaît clair que notre système se montre capable d'engendrer une structure sociale de division du travail très semblable à celle observée dans les colonies naturelles. Ce résultat ne doit pas être sous-estimé. En effet, rien dans le comportement des agents, ni dans les contraintes environnementales, ne permet de prédire une répartition des tâches hétérogène qui conduise à l'émergence de groupes fonctionnels. Comme le montre [Theraulaz 1991] sur un système plus simplifié que le nôtre, il aurait été possible d'obtenir une répartition uniforme d'où n'émerge aucun spécialiste, et, partant, aucun groupe. Nous allons tenter de déterminer, dans la section suivante, quels sont les mécanismes qui permettent d'expliquer l'émergence d'une division du travail.

V.4.3. *Comment se génère la division du travail ?*

La condition *sine qua non* de l'existence d'une division du travail entre les individus passe, comme chez l'homme, par la possibilité d'une spécialisation individuelle. Dans les sociétés dont les ouvrières sont polymorphes, cette spécialisation est en quelque sorte déterminée dès la naissance. Certaines ouvrières possèdent alors des attributs physiologiques leur interdisant, ou au contraire leur favorisant l'exécution d'une ou plusieurs tâches. Chez *Ectatomma ruidum*, et *a fortiori* dans MANTA, nous n'avons rien de tel puisque les ouvrières sont strictement monomorphes, donc toutes pourvues des mêmes capacités physiologiques et comportementales. La spécialisation individuelle va donc être le fruit de deux facteurs, qui sont respectivement les tâches à pourvoir dans la colonie et la possibilité de renforcer les comportements correspondant à ces tâches. Il peut sembler inutile d'inclure les tâches à effectuer comme facteur de la spécialisation, puisqu'à première vue, on ne peut se spécialiser que dans une activité que l'on pratique. Mais ce processus n'est pas si trivial, comme nous le verrons par la suite, et tenir compte de ce facteur permet d'expliquer la stabilité des formes d'organisation observées. Pour expliquer comment sont générées les structures de division du travail que nous avons présenté, il nous faut donc répondre à trois questions: comment se spécialise un agent, comment cette spécialisation influe-t-elle sur celle des autres agents et génère-t-elle une division des tâches, et comment cette organisation peut-elle rester stable ?

V.4.3.1. *SPECIALISATION INDIVIDUELLE*

La spécialisation individuelle passe bien entendu par la possibilité, incluse dans le modèle de comportement fourni aux agents, de renforcer les tâches souvent effectuées. Il nous faut d'abord souligner qu'il s'agit là d'un processus autocatalytique, dans le sens où le renforcement d'une tâche conduit à accroître la probabilité que cette tâche soit de nouveau effectuée prioritairement par l'agent quand ses préconditions seront vérifiées. Ceci s'explique tout naturellement par le fait que le renforcement du poids de la tâche va induire une sensibilité différentielle de l'agent par rapport aux stimuli qu'il sera amené à rencontrer.

Si nous laissons de côté la reine et que nous nous intéressons à la spécialisation d'une ouvrière qui vient de naître, qu'observons nous ? Tout d'abord que la première tâche qui sera choisie ne le sera pas en fonction du profil comportemental de l'agent mais en fonction de la situation dans laquelle il est plongé. Si l'agent est sollicité par plusieurs stimuli de forces équivalentes, le choix qu'il fera sera donc aléatoire. Et c'est à partir de ce moment-là que sa sensibilité deviendra différentielle, c'est-à-dire que son profil comportemental commencera à influencer sur son choix.

Trois cas de figures schématiques peuvent alors se présenter (voir Figure II.21):

- ou bien la tâche qu'il vient d'exécuter a temporairement consommé son stimulus déclencheur, ce qui signifie que ce n'était pas une tâche très urgente, auquel cas l'agent va pouvoir être soumis à d'autres influences et donc noyer sa spécialisation naissante en renforçant d'autres tâches (cas C.);
- ou bien l'exécution de la tâche n'a pas suffi à totalement éteindre son stimulus déclencheur (ce qui est souvent le cas au début car la durée théorique de la tâche n'est pas très importante, voir Chapitre III), auquel cas l'agent, de par la plus grande sensibilité qu'il vient de développer à son égard, aura une forte probabilité de l'effectuer de nouveau, et donc de la renforcer, ce qui conduira peu à peu à une spécialisation plus importante (cas A.);
- ou bien, enfin, l'environnement se montrant peu attractif après l'exécution de cette tâche, ce qui signifie qu'elle a consommé son stimulus et qu'aucune autre tâche n'apparaît urgente, l'agent effectuera sa tâche par défaut, en conservant la sensibilité qu'il vient de développer, et donc la probabilité, dans d'autres situations, de déclencher prioritairement cette tâche (cas B.).

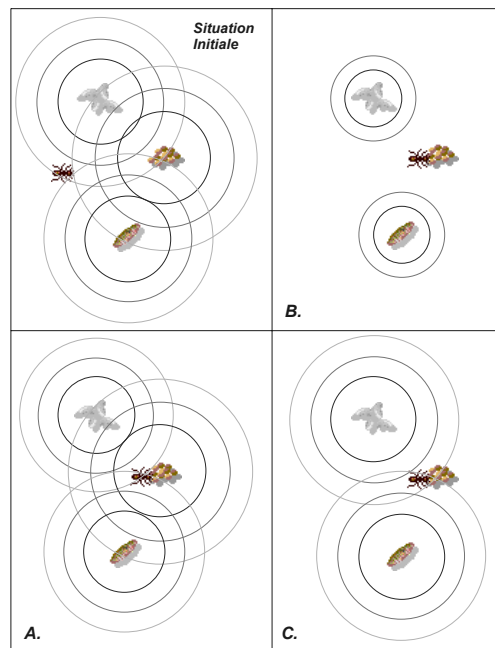


Figure II.21 - Quelques cas de figures jouant sur la spécialisation individuelle. En haut à gauche la situation initiale (choix de la tâche). En A, B et C les situations après que la tâche a été accomplie.

V.4.3.2. INFLUENCE INTER-INDIVIDUELLE ET DIFFERENCIATION

Comme nous l'avons déjà évoqué au Chapitre IV, le fait qu'un agent effectue une tâche n'est pas neutre sur le comportement des autres agents. Dans un sens, il va en effet "capturer" à son profit cette tâche et le stimulus déclencheur attendant, ce qui réduira fortement la probabilité que d'autres agents l'effectuent simultanément. Si nous faisons le lien avec la section précédente, nous voyons donc se dessiner la possibilité pour les agents d'influer mutuellement sur leurs sensibilités respectives et, partant, sur leurs spécialisations. Car, si nous plongeons maintenant l'agent évoqué plus haut dans une situation où il est en compétition avec d'autres agents pour effectuer une tâche, nous avons une bien plus forte probabilité d'obtenir les cas de figures A. ou B. que le cas C.

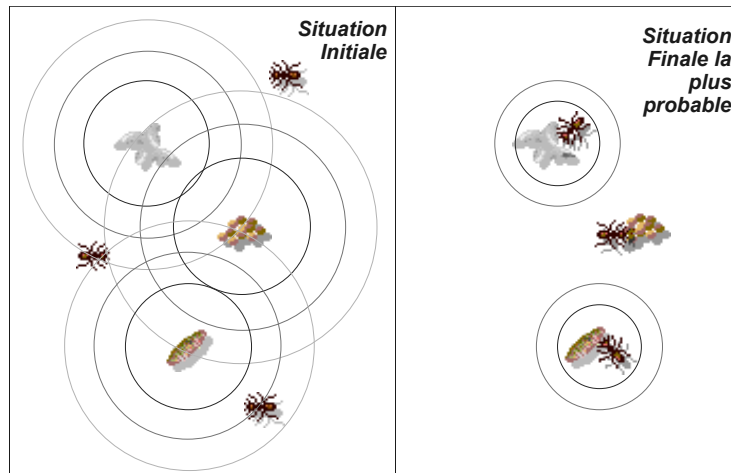


Figure II.22 - La même situation initiale que la celle de la Figure précédente, mais avec plusieurs agents. La situation finale la plus probable est équivalente à la situation B.

En effet, si nous regardons la Figure II.22, qui ne se veut pas représentative de l'ensemble des situations possibles, mais qui représente une situation suffisamment commune pour que nous puissions la prendre en exemple, dès qu'un agent aura choisi une des trois tâches, il fera fortement baisser la probabilité des deux autres agents de choisir la même, limitant leur choix aux deux autres tâches, pour lesquelles le même processus a de grandes chances de se répéter. Si les individus sont tous non-spécialistes au départ, ils acquerront une sensibilité différente à la fin. Quand une situation analogue se reproduira, la répartition des trois agents se fera de manière beaucoup plus rapide, chacun suivant le stimulus dans lequel il s'est spécialisé. La compétition entre les différents agents constitue donc un puissant catalyseur pour les processus de spécialisations individuelles, en générant, de manière globale, des phénomènes d'exclusion mutuelle répartis [Hatcher, Tofts et Francks 1992], et donc un phénomène général de différenciation des formes comportementales. Si nous considérons d'une part le niveau "micro" représenté par les ouvrières, et d'autre part le niveau "macro" constitué par l'ensemble des groupes fonctionnels, nous avons ainsi un double processus de structuration qui se fait jour: structuration du niveau "macro" par les niveaux "micro", dans le sens où les profils comportementaux individuels vont établir la forme et les profils des groupes fonctionnels, et structuration du niveau "micro" par le niveau "macro", dans le sens où la répartition globale des tâches résultant de cette division en groupes va confirmer l'agent dans son profil, en excluant la possibilité pour lui d'effectuer d'autres tâches, qui sont prises en charge par les autres agents. Il est par ailleurs intéressant de constater que la structuration de l'environnement, par exemple par l'agrégation des trois types d'agents appartenant au couvain dans différentes chambres, joue un rôle absolument essentiel dans le processus de structuration sociale, dans la mesure où il va également servir à conditionner les agents. Ainsi, un agent s'occupant des oeufs se trouvera, la plupart du temps, dans la chambre contenant les oeufs, ce qui exclura pratiquement toute possibilité pour lui d'être soumis à d'autres stimuli que ceux provenant de ces agents. [Fresneau, Corbara et Lachaud 1989] ont ainsi vérifié l'étroite corrélation existant entre le profil comportemental et la répartition spatiale des ouvrières dans les colonies de *Pachycondyla apicalis*, une autre espèce appartenant à la même sous-famille qu'*Ectatomma ruidum*. Nous n'avons, pour notre part, pas eu le temps d'effectuer des expériences similaires, mais nous avons vérifié cette propriété dans d'autres applications (comme celles concernant la Résolution Collective de Problèmes, voir Chapitre VI) directement dérivées de MANTA, ce qui nous permet de penser qu'elle joue également un rôle important dans la simulation.

V.4.3.3. STABILITE DE LA STRUCTURE SOCIALE

Tant que l'environnement reste relativement stable, ce qui est le cas dans les simulations que nous étudions ici, le seul facteur pouvant quelque peu perturber ce processus inéluctable de division du travail est l'arrivée de nouvelles ouvrières. Celles-ci doivent en effet s'intégrer à une structure existante, mais cette structure doit également s'adapter à elles. Ce que nous avons constaté, en parfait accord avec les données expérimentales à notre disposition (et avec notre intuition), est que la perturbation engendrée par la naissance de nouvelles ouvrières est inversement proportionnelle à l'effectif de la colonie. Une population importante en nombre est ainsi capable d'absorber ces nouvelles ouvrières sans pratiquement rien changer à sa structure de division du travail, alors qu'une population de taille restreinte verra sa structure se modifier de façon significative. Cela provient bien sûr de ce que nous avons déjà montré dans la section précédente, à savoir que dans le premier cas, le comportement de la nouvelle arrivée va très vite être canalisé par l'influence indirecte des autres ouvrières, alors que dans le second, elle sera plus "libre" et donc susceptible de choisir son profil sans subir les contraintes posées par la structure sociale existante. Il existe également une explication plus prosaïque, qui consiste à considérer que, statistiquement, l'importance du profil comportemental d'un individu dans celui de sa société est effectivement inversement proportionnel au nombre d'individus considérés. Il y a d'ailleurs toutes les chances que ces deux explications se combinent.

Cependant, même si la structure sociale, donc la répartition dynamique des individus en groupes fonctionnels, n'est pas affectée de manière globale (pour l'observateur) par l'apparition de nouvelles ouvrières, nous pouvons pratiquement affirmer que celle-ci conduit à un réajustement des comportements d'un grand nombre d'ouvrières, afin de "faire de la place" aux nouvelles arrivantes dans les groupes fonctionnels pré-existants. Ceci est confirmé par les résultats d'expériences sur le polyéthisme d'âge, résultats que nous présentons en Section V.5. Que cela signifie-t-il ? Tout simplement que la structure sociale, à partir d'un certain seuil de population que nous pouvons empiriquement estimer à dix individus, devient en quelque sorte indépendante des agents qui la forment, ceux-ci glissant de groupe en groupe au gré de leur âge et des arrivées de nouveaux agents, sans jamais remettre en cause l'existence même de ces groupes. L'ensemble des résultats et des explications que nous avons présentés à propos de la structuration sociale constitue donc un pas important, aussi bien pour la validation du modèle que pour notre problématique informatique. En ce qui concerne la validation du modèle, ils permettent de conclure provisoirement (car tous les résultats d'expérience ne sont pas dépouillés) que les hypothèses retenues par les éthologues pour décrire les individus, aussi bien dans la teneur des comportements que dans leur mode de déclenchement (stimuli, poids, etc.), suffisent, une fois codées dans le modèle que nous proposons, à générer des structures supra-individuelles (donc, sociales) isomorphes à celles obtenues dans la réalité. Cela ne veut pas dire que toutes sont nécessaires, et cela ne veut pas dire non plus qu'elles sont seules en cause. Nous reviendrons sur ce point dans l'étude des résultats expérimentaux concernant le polyéthisme d'âge (Section V.5.) et la polygynie (Section V.6.).

V.4.3.4. L'EMERGENCE D'UNE STRUCTURE DE COOPERATION

Du point de vue de notre problématique, ces résultats sont tout aussi importants car ils nous permettent d'envisager l'obtention de structures sociales par l'interaction d'agents réactifs. Dans nos colonies simulées, aucun agent ne possède en effet de représentation de la division du travail, et cette dernière ne nécessite non plus aucun contrôle de leur part. Cependant, le jeu en vaut-il la chandelle ? Autrement dit, la structure résultante est-elle comparable à des structures plus centralisées de répartition des tâches, notamment en terme d'efficacité ? Il semble que ce soit le cas (voir [Hölldobler et Wilson 1990] pour une démonstration partielle), et nous allons rapidement le montrer.

Dans un cadre comme celui de la fourmilière, où de nombreuses tâches doivent être exécutées en parallèle, et dans lequel il n'est pas concevable que certaines activités ne soient pas effectuées, une structure efficace de division du travail et de coopération doit exhiber les caractéristiques suivantes: être efficace, c'est-à-dire effectuer les tâches dans le minimum de temps requis, être robuste, c'est-à-dire s'adapter à la disparition d'un ou plusieurs individus, et être flexible, c'est-à-dire pouvoir s'adapter à des perturbations endogènes, comme la naissance de nouveaux individus, ou exogènes, comme des contraintes environnementales. Dans le modèle, ces trois caractéristiques s'expriment de la manière suivante: l'efficacité nécessite une importante spécialisation individuelle, source d'efficacité individuelle, et une non moins importante différenciation des profils comportementaux, source d'efficacité globale par non-redondance des groupes fonctionnels⁶³. La robustesse nécessite une certaine redondance entre ces individus spécialisés, de manière à pouvoir pallier rapidement à la disparition de l'un d'entre eux. Ces deux points sont d'ailleurs, selon [Atlan 1979], des facteurs nécessaires à l'obtention d'une organisation, «*qui consistera en un compromis optimal entre ces deux propriétés opposées*» que sont la différenciation et la redondance. Enfin, la flexibilité, qui n'est rien d'autre que la modification du rapport entre ces deux valeurs en réponse à une perturbation, exige que la structure sociale soit capable d'influer sur les comportements de ses composants.

La vérification des deux premières caractéristiques découle naturellement de ce que nous avons décrit dans la section précédente concernant la répartition en groupes fonctionnels. L'ensemble des groupes est l'expression de la variété ou de la différenciation de la société, donc de ses spécialisations, et les individus composant ces groupes de sa redondance, dans la mesure où ceux-ci sont fonctionnellement interchangeables. Pour ce qui est de la flexibilité, nous savons que la répartition des individus entre ces groupes est directement fonction de la demande sociale. Ainsi, quand trop d'individus sont spécialisés dans une tâche par rapport aux besoins de la société, ceci se traduit forcément par l'inactivité de certains d'entre eux, qui seront ainsi susceptibles, s'ils sont soumis à de nouvelles influences, de se spécialiser dans une autre tâche, et donc de changer de groupe. La plasticité comportementale des agents permet ainsi d'envisager une flexibilité potentielle, qui s'actualise de manière nette dans les sociétés simulées lors, par exemple, de l'arrivée massive d'une catégorie de couvain (quand les larves deviennent des cocons, par exemple), ou comme nous pouvons le voir sur la Figure II.23, qui représente le sociogramme de la colonie P11_4 (voir Section V.3.) lorsque la nourriture est épuisée.

Ce sociogramme montre en effet que l'activité de fourragement, jusque-là importante mais cantonnée dans les limites présentées plus haut, devient majoritaire chez toutes les ouvrières. De plus, il apparaît que ce sont les jeunes qui s'y consacrent le plus (75% et 80% du budget-temps des ouvrières n°8 et 9), ce qui crée une situation tout à fait exceptionnelle au regard de celles habituellement constatées (voir Section V.5.) dans des environnements plus stables. Dans une telle situation, les jeunes ouvrières, moins spécialisées que les plus âgées, semblent avoir directement répondu à la demande sociale diffuse concernant le manque de nourriture, et ont ainsi généré une réponse collective apte à rétablir l'équilibre nutritionnel. Les tâches essentielles à la survie de la colonie ne sont pas pour autant négligées car les ouvrières plus âgées, ainsi que la reine, continuent de s'occuper du couvain.

⁶³ C'est l'équivalent de la «loi de la variété requise» de [Ashby 1962] pour assurer une autonomie fonctionnelle au système.

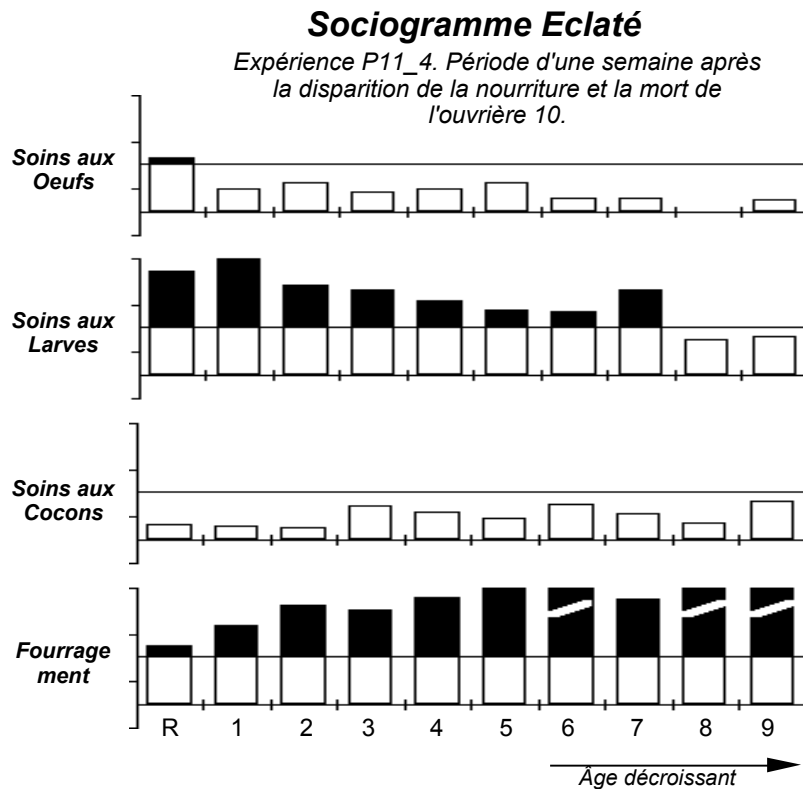


Figure II.23 - Sociogramme Eclaté de la colonie P11_4, soumise à la restriction de la nourriture.

La division du travail que nous obtenons n'a donc rien à envier à des organisations sociales plus centralisées ou plus normatives, comme par exemple celle d'une entreprise. Comme le souligne [Hölldobler et Wilson 1990], elle est même, du fait de sa décentralisation, apte à répondre beaucoup plus rapidement à des changements majeurs qu'une organisation hiérarchisée, où l'information perçue localement doit d'abord remonter, puis redescendre sous la forme de directives. Ici, chacune de ses composantes va réagir localement aux informations qu'elle perçoit, et cette réaction va influencer sur la réaction des autres composantes pour générer une réponse globale qui apparaît coordonnée, alors qu'elle ne l'est nullement. Nous montrerons, dans la troisième et la quatrième partie, que l'obtention de cette propriété globale nous a été précieuse, puisqu'elle a permis d'imaginer des solutions logicielles basées sur les mêmes concepts dans le cadre de la résolution distribuée de problèmes (Chapitre VI).

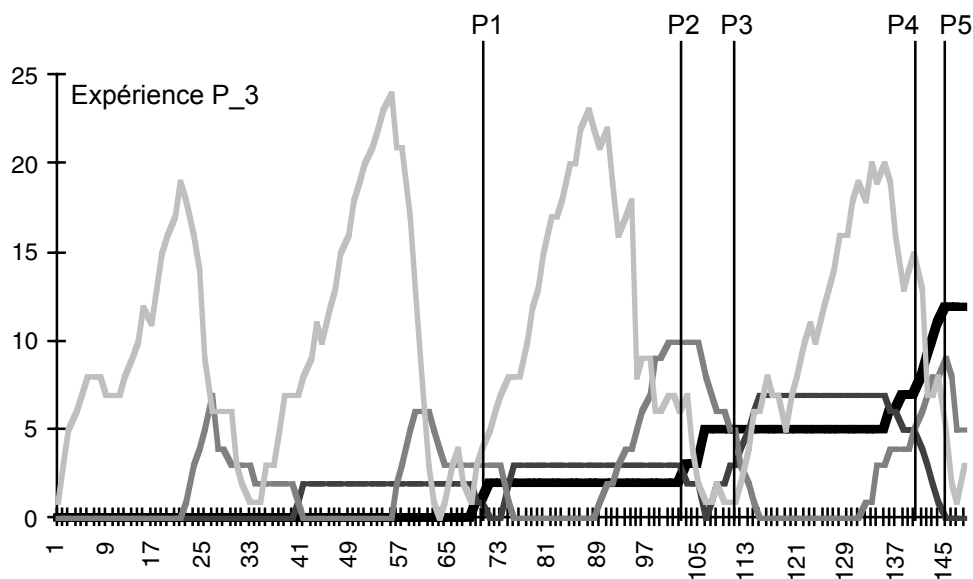
V.5. La Question du Polyéthisme d'Age

Nous abordons, avec cette section, deux séries d'expérimentations qui ont pour but de tester la validité des hypothèses ontogénétiques avec lesquelles nous travaillons en vue de reproduire des phénomènes qui sont habituellement attribués à des influences maturationnelles ou génétiques. Il ne s'agit donc plus seulement de simuler la dynamique sociale des colonies, mais également de confronter la théorie qui nous a servi à construire le modèle à d'autres théories existantes, afin d'en détecter les faiblesses qui nous obligeraient à prendre de nouveaux facteurs en compte. La première porte sur la notion de polyéthisme d'âge, et est décrite dans cette section. La seconde, présentée dans la section suivante (Section V.7.), s'intéresse à l'étude des sociétés polygynes.

Le terme "polyéthisme d'âge" sert à désigner un phénomène typique des sociétés d'insectes sociaux, qui voit les individus changer de spécialisation au fur et à mesure qu'ils vieillissent, selon un ordre qui suit celui des activités utilisées pour les sociogrammes (du soin aux oeufs jusqu'au fourrage ment). Chez

E. ruidum, comme le souligne [Corbara 1991], cette évolution n'est pas aussi marquée, puisque des individus de même âge peuvent varier dans leurs spécialisations, mais il est possible d'observer une certaine homogénéité d'âge à l'intérieur des groupes fonctionnels. Ainsi, les soigneuses d'oeufs sont généralement recrutées parmi les individus les plus jeunes, et les fourrageuses parmi les plus vieilles. Une des hypothèses les plus utilisées pour expliquer ce phénomène se base sur l'atrophie des organes reproducteurs des ouvrières au cours de leur vie [Hölldobler et Wilson 1990], ce qui aurait tendance à diminuer leur sensibilité au couvain, et par là même à accroître leur probabilité d'effectuer le fourrage. Cependant, cette hypothèse ne permet pas d'expliquer de manière satisfaisante l'idiosyncrasie individuelle d'*E. ruidum*. Nous avons donc voulu étudier dans quelle mesure le modèle de comportement que nous avons programmé suffisait à générer un phénomène de polyéthisme d'âge. Pour cela, nous avons analysé l'évolution de deux colonies depuis leur fondation jusqu'à un stade de 10 ouvrières pour la première (P_3) et de 12 ouvrières pour la seconde (P_5).

Pour chacune de ces deux colonies, nous avons isolé un certain nombre de périodes clés (notées de P1 à Px) correspondant à des effectifs différents (voir Figure II.24 et II.25). Ainsi, l'expérience P_3 a été analysée au travers de 5 périodes où la colonie contenait respectivement la reine et une ouvrière (P1), la reine et trois ouvrières (P2), la reine et cinq ouvrières (P3), le reine et neuf ouvrières (P4) et enfin la reine et dix ouvrières (P5). De la même manière, P_5 a été étudiée sur six périodes pendant lesquelles la société contenait la reine et une ouvrière (P1), la reine et deux ouvrières (P2), la reine et trois ouvrières (P3), la reine et six ouvrières (P4), la reine et sept ouvrières (P5) et enfin la reine et douze ouvrières (P6). Comme nous n'avons eu aucune mortalité d'ouvrières durant ces deux études, les numéros d'ordre de chaque ouvrière restent inchangés.



Durant ces périodes, dont la durée est d'une semaine chacune, les profils de chacun des individus par rapport aux quatre activités (Soins aux oeufs, aux larves, aux cocons et Fourrage) ont été calculés et comparés. Nous aboutissons ainsi aux deux Figures (II.26 et II.27), qui se lisent de la même manière que les sociogrammes éclatés précédemment présentés.

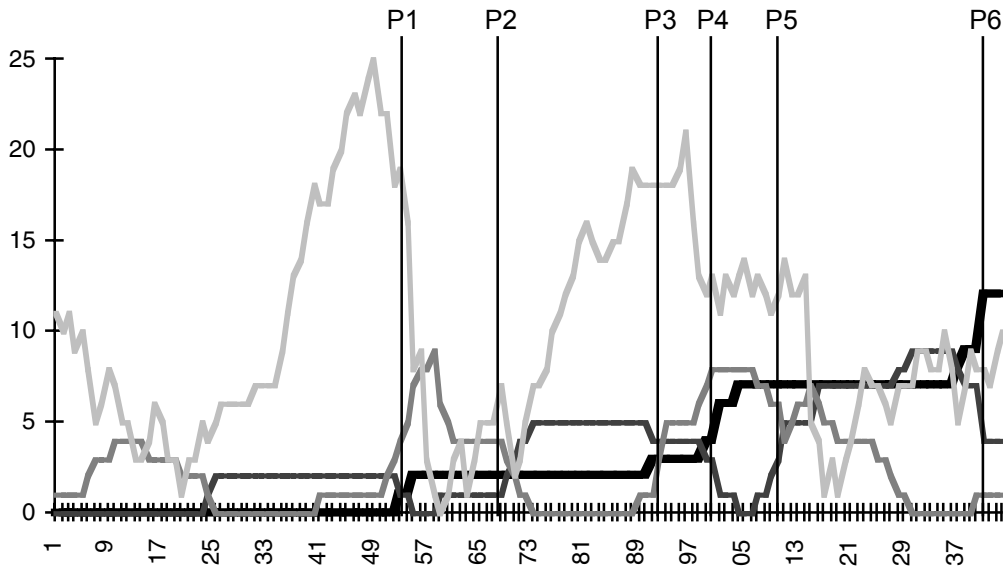


Figure II.25 - Répartition du début des périodes d'études de la colonie P_5. Chaque période dure une semaine. Echelle des abscisses: 2 jours / unité.

Sociogrammes Eclatés sur Cinq Périodes Expérience P_3

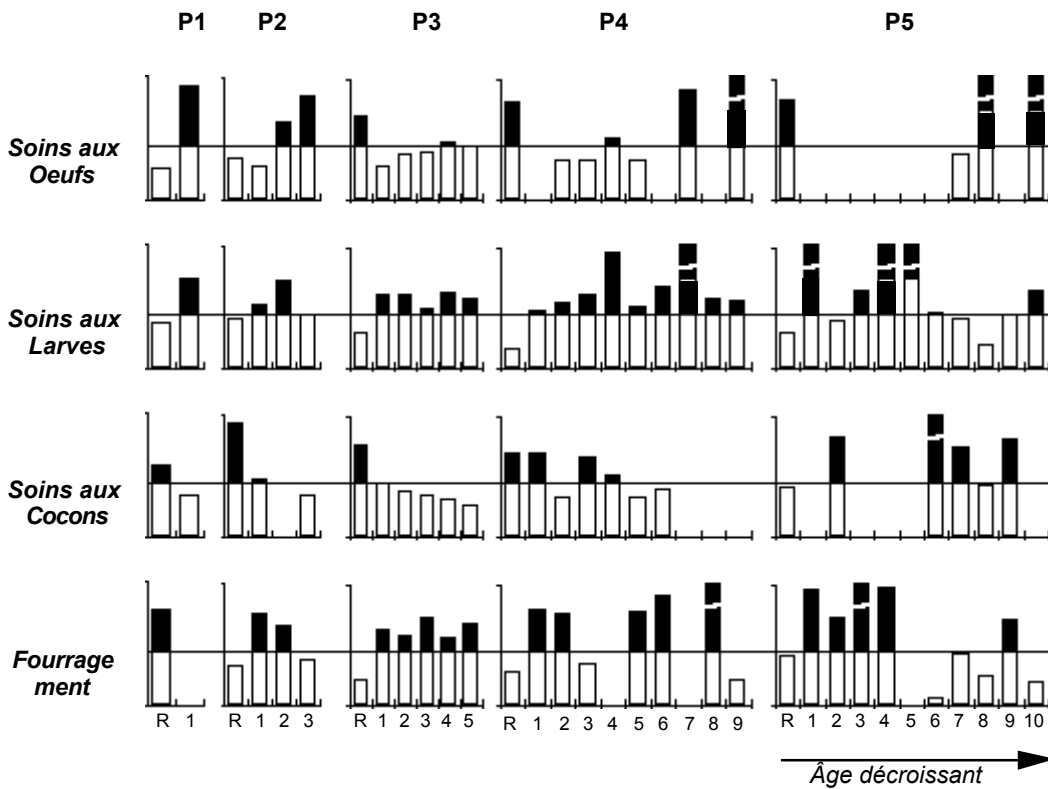


Figure II.26 - Evolution des sociogrammes éclatés de la colonie P_3.

Sociogrammes Eclatés sur Six Périodes

Expérience P_5

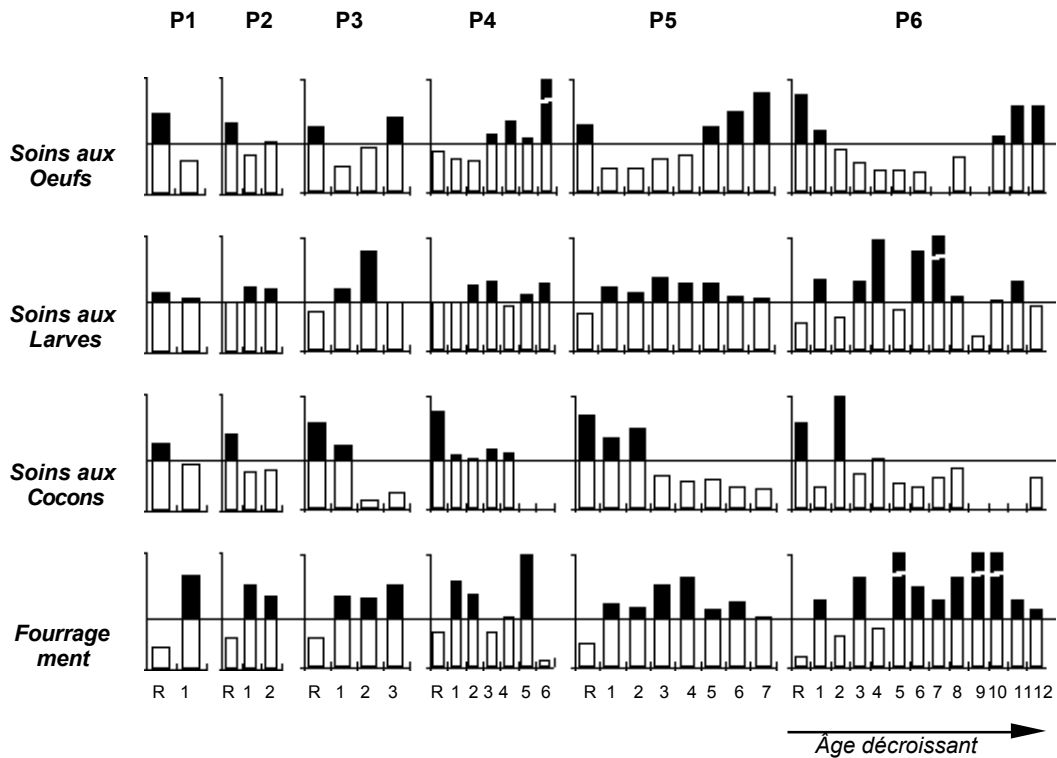


Figure II.27 - Evolution des sociogrammes éclatés de la colonie P_5.

L'étude de ces deux Figures permet de faire plusieurs constatations intéressantes par rapport à la notion de polyéthisme d'âge. Si nous suivons un individu, par exemple l'ouvrière n°1 de la colonie P_3 (qui est l'ouvrière la plus âgée), nous voyons clairement que l'évolution de son profil se traduit par un abandon progressif des comportements liés au couvain, et surtout aux oeufs, pour se concentrer sur le fourrage ment. Si nous suivons par contre l'ouvrière n°1 de l'expérience P_5, qui commence à fourrager très tôt, cette évolution est beaucoup moins nette, et, pour tout dire, pratiquement inexistante. Nous n'obtenons donc pas de chronologie stricte de spécialisation au niveau individuel. Plusieurs ouvrières présentent d'ailleurs des profils totalement atypiques, comme l'ouvrière n°9 de l'expérience P_3, qui possède un profil ordinairement propre aux ouvrières âgées (prédominance du fourrage ment et du soin aux cocons) alors qu'elle est très jeune. Si maintenant nous analysons, activité par activité, la répartition des spécialisations en fonction de l'âge des individus, voici ce que nous obtenons :

- *Soins aux Oeufs*: dans les deux colonies, quelle que soit la période considérée, cette activité est majoritairement assurée par la reine et par les plus jeunes ouvrières, ce qui est en parfaite adéquation avec les résultats expérimentaux obtenus par [Corbara 1991]. L'âge semble donc avoir un effet important sur la spécialisation dans cette activité.
- *Soins aux Larves*: dans les deux colonies, cette activité est majoritairement assurée par les ouvrières situées au milieu de la classe d'âge, avec une prédominance de jeunes. Cependant, les phénomènes d'idiosyncrasie sont plus importants dans cette activité que dans la précédente et il est difficile de conclure à l'influence de l'âge sur cette spécialisation.
- *Soins aux Cocons*: les résultats diffèrent entre les deux colonies. Dans P_3, l'évolution jusqu'à la dernière période est conforme au polyéthisme d'âge, à savoir que ce sont les ouvrières les plus âgées qui apparaissent spécialisées dans cette activité, mais la tendance s'inverse brusquement pour la période P5. Il faut sans doute y voir l'effet du bouleversement (arrivée d'un nombre important de larves) perceptible au début de cette

période sur la Figure II.26. La nombre d'actes de soin aux cocons est en effet extrêmement faible. Pour la colonie P_5, en revanche, les spécialisations sont conformes à ce qui était attendu, les ouvrières les plus âgées prenant en charge les cocons, quelle que soit l'effectif de la société. Si nous mettons de côté la dernière période de P_3 pour les raisons que nous venons d'évoquer, l'âge semble donc jouer un rôle important sur la spécialisation des individus dans cette activité.

- *Fourragement*: les résultats sont ici plus contrastés, et de plus certainement faussés par la grande importance de cette activité. En fait, le fourragement semble être réparti sur l'ensemble des classes d'âge (hormis la reine, mais son profil atypique ne rentre pas en ligne de compte ici), avec, ici et là, des spécialisations conformes au polyéthisme d'âge (par exemple, la dernière période de P_3), mais loin d'être majoritaires. Si l'on compare les sociogrammes des deux colonies avec ceux obtenus pour les sociétés ERF1, ERF2 et ERF3 de [Corbara 1991] (Chap. IV, pp. 92 à 100), la différence est très nette.

Que pouvons nous donc conclure de cette série d'observations ? Pour deux activités au moins (soins aux œufs et aux larves), nous pouvons affirmer sans trop de risques que la spécialisation dépend de l'âge. Pour les deux restantes, l'idiosyncrasie individuelle semble de règle. Or, ces deux activités sont précisément celles pour lesquelles nous avons eu le plus de mal à établir une correspondance directe avec les tâches effectuées par les ouvrières, dans la mesure où elles nécessitent d'importants déplacements dans le nid (à la différence des tâches intervenant dans les soins aux cocons et aux œufs). Il est donc possible que nous spécialisons certains individus à tort, du moins par rapport à ce que ferait un éthologue observant le système. Le résultat est donc contrasté, et nécessitera sans doute d'être révisé par de nouvelles expérimentations.

Cependant, nous pouvons d'ores et déjà nous pencher sur le cas des deux activités pour lesquelles le polyéthisme d'âge apparaît. Elles constituent en effet un résultat plutôt surprenant, car, comme nous le rappelions au début de cette section, aucun facteur maturationnel dans le comportement des agents ne peut l'expliquer. Nous nous trouvons donc confronté là à une dynamique purement sociale, qui illustre bien la flexibilité de la division du travail entre les individus, puisque ceux-ci, peu à peu "repoussés" par les nouvelles arrivantes, vont abandonner le soin aux œufs pour se consacrer à d'autres activités. Le terme "repousser" peut ici être pris au pied de la lettre, puisque, en l'absence d'autres facteurs, la seule explication que nous pouvons fournir d'une telle dynamique porte précisément sur les contraintes spatiales intervenant dans le choix du comportement individuel. Les ouvrières nouvellement nées sont en effet en contact étroit avec le couvain durant leurs premiers jours ([Chauvin 1982] rapproche cette période de la période d'imprégnation décrite par [Lorenz 1984]) et donc susceptibles de rentrer en compétition avec les ouvrières habituellement chargées de cette tâche, surtout si celles-ci s'éloignent, ne serait-ce que temporairement. Or, comme nous l'avons déjà expliqué, le simple fait pour les plus jeunes de s'occuper du couvain va réduire la probabilité que les ouvrières plus âgées reviennent dans la chambre à couvain, en réduisant les stimuli qui y sont propagés. Dans un schéma idéal, les jeunes vont donc peu à peu se spécialiser dans le soin aux œufs (desquels elles sont les plus proches), les plus âgées, quant à elles, repoussées de la chambre des œufs, vont s'occuper des larves (placées dans une chambre plus excentrée), puis, dans la même dynamique, des cocons (placés dans une chambre encore plus excentrée) et enfin du fourragement. Evidemment, ce schéma idéal n'est jamais tout à fait respecté, puisqu'il dépend de nombreux facteurs, comme par exemple les effectifs respectifs des populations d'œufs, de larves et de cocons, mais également de la mobilité des ouvrières. Mais il constitue une trame intéressante qui ne prend en compte aucun facteur maturationnel et permet d'expliquer l'influence de l'âge au moins sur la spécialisation (ou la dé-spécialisation) dans le soin aux œufs. Il lui reste cependant à être confirmé par de nouvelles expériences, à la fois en milieu naturel et artificiel.

V.6. La question de la polygynie

Parmi les diverses stratégies utilisées par les fourmis pour s'adapter, notamment lors de sociogenèses, à leur environnement, la *polygynie* soulève de nombreuses questions d'ordre fonctionnel. Ce phénomène est relativement courant chez les Hyménoptères sociaux autres que les fourmis (notamment chez les guêpes sociales, où cette option est majoritaire), mais n'apparaît que minoritairement chez les fourmis. Pourtant, il semble évident, au vu de la difficulté que constitue la fondation d'une colonie (Section V.1.), qu'une société démarrant avec plusieurs reines se donne plus de chances de réussir qu'une société démarrant avec une seule reine, ne serait-ce qu'en raison de l'apport de robustesse que constitue, dans la colonie, la redondance individuelle. La question est de savoir pourquoi les sociétés polygynes ne sont pas plus nombreuses dans l'espèce que nous étudions, à savoir *Ectatomma ruidum*, où elles sont rarissimes [Corbara 1991].

Il importait donc de vérifier si, effectivement, une fondation polygyne était plus efficace, plus robuste, et donc moins aléatoire qu'une fondation monogyne. Nous avons à cet effet réalisé une série d'expérimentations qui consistaient simplement à démarrer la sociogenèse de colonies avec plusieurs reines (les expériences ont été réalisées avec quatre reines) et à observer leur dynamique de développement en la comparant à la dynamique de fondation des sociogenèses présentées Section V.1. Comme il nous apparaissait important de suivre, non seulement l'évolution globale, mais également l'évolution individuelle de chacune des reines et de leurs lignées respectives, nous avons rajouté à tous les agents de MANTA la variable d'instance `lineage` (lignée) dont l'initialisation est réalisée lors de la création de reines (chacune a un numéro différent), et dont la valeur est transmise lors de la ponte de la reine aux oeufs, puis des oeufs aux larves lors de leur changement d'état, et ainsi de suite jusqu'aux ouvrières. Ceci nous permet de connaître avec exactitude la répartition démographique de la colonie entre les différentes lignées. Bien entendu, les expériences ont été menées avec des agents identiques à ceux utilisés précédemment, ce qui signifie que, contrairement à ce qui peut être observé dans la nature [Jaisson 1993], les reines n'ont aucun comportement particulier les unes à l'égard des autres, non plus qu'à l'égard du couvain (elles ne différencient pas leur propre couvain du couvain des autres reines).

Ce premier ensemble de quinze expérimentations a fourni des résultats assez surprenants, car comparables en tout point aux résultats obtenus avec des fondations monogynes. Le taux de réussite moyen est en effet de 20% (contre 20,97%), et les échecs se produisent approximativement dans les mêmes situations, à savoir lorsque la population de larves est importante. Il semble donc que la présence de quatre reines ne soit pas vraiment bénéfique du point de vue fonctionnel, alors que la présence dans une colonie monogyne de la reine et de trois ouvrières, donc également de quatre individus capables d'effectuer les tâches nécessaires conduit dans 9 cas sur 10 à un succès. A quoi cela est-il dû ? Tout simplement à la seule différence existant reines et ouvrières, à savoir que seules les reines pondent des oeufs fécondés. Lors de la fondation, les quatre reines vont donc générer quatre fois plus de couvain que dans une situation comprenant une reine et trois ouvrières. La force de travail des quatre reines ne peut ainsi se répartir sur les différentes tâches à accomplir, car toute la demande en soins et en nourriture des oeufs, larves et cocons est également multipliée par quatre. L'évolution démographique de la colonie N_8, présentée Figure II.29, est ainsi exactement semblable à celle d'une colonie monogyne, mis à part le facteur d'échelle qui gravite autour de 4.

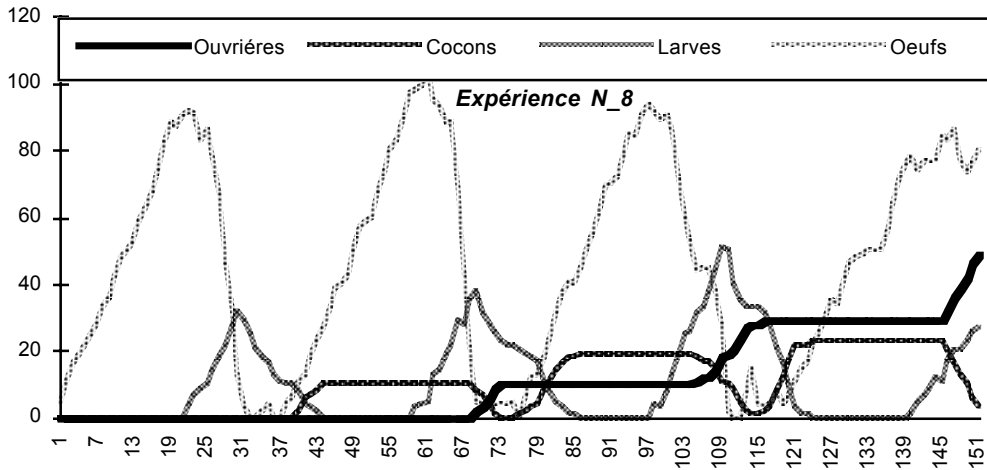


Figure II.29 - Evolution (sans distinction de lignée) de la colonie N_8.

Si l'on suit, lignée par lignée (Figure II.30), l'évolution démographique de la même colonie, on s'aperçoit qu'aucune reine ne se "détache" des autres, ce qui donne l'illusion du déroulement de quatre sociogénèses à l'intérieur du même nid.

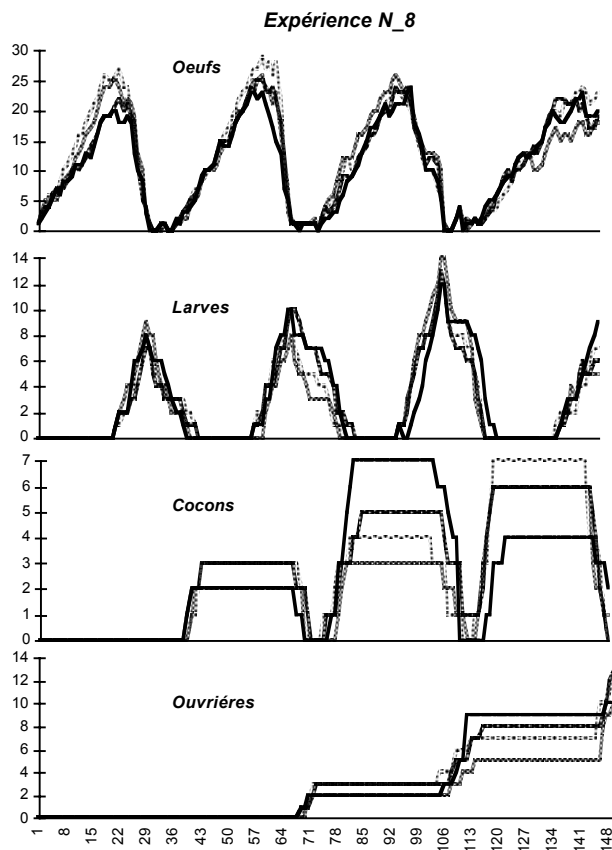


Figure II.30 - Effectifs des 4 lignées de l'expérience N_8.

Au vu de ces résultats, il apparaît clairement que la polygynie n'est guère avantageuse pour nos fourmis artificielles. Ou, plus précisément, qu'elle n'est pas plus avantageuse que la monogynie. En un sens, ce

résultat est tout à fait normal, et même plutôt rassurant du point de vue de la simulation. Car, si, dans nos simulations, les fondations polygynes s'étaient avérées beaucoup plus avantageuses que les fondations monogynes, nous aurions eu des difficultés à expliquer pourquoi, dans la nature, elles ne constituaient pas la norme d'*Ectatomma ruidum*. L'une des raisons essentielles paraît en être le comportement solipsiste des reines, qui continuent toutes à pondre alors que la situation exigerait d'elles que certaines se restreignent et assument d'autres tâches pour garantir le développement de la colonie. Et il est plutôt normal que, aucun comportement particulier ne leur ayant été fourni, cette différenciation ne soit pas observée. Les expérimentations présentées Section suivante explorent, dans une pure optique d'éthologie synthétique, l'une des hypothèses comportementales plausibles pouvant aboutir à ce type de scénario.

V.7. Competition Spatiale entre Reines

V.7.1. Ajout d'un nouveau comportement aux reines

Bien que nous n'ayons malheureusement pas encore de résultats des simulations concernant la répartition spatiale des ouvrières dans une colonie adulte, il est indéniable qu'elle joue dans la nature un grand rôle dans la différenciation individuelle, et donc dans la formation des groupes fonctionnels. Elle semble être assurée par un phénomène d'exclusion mutuelle, exclusion réalisée par une influence indirecte *via* l'environnement (voir Section V.4.). Or, dans le cas précédent des fondations polygynes, il nous a été impossible de constater le moindre phénomène de répartition spatiale entre les reines, mis à part les sorties ponctuelles effectuées par certaines d'entre elles pour aller fourrager. Toutes restent dans le voisinage les unes des autres, ce qui explique sans doute la similitude d'évolution de leurs lignées, mais également les résultats en "tout ou rien" que nous obtenons (*i.e.* soit la mort de la société, soit le maintien des quatre reines au sein d'une "vraie polygynie" [Hölldobler et Wilson 1990]). Ce fait peut se comprendre assez facilement, étant donné que les reines pondent au même endroit et que, même quand l'une d'elles manifeste la velléité de s'éloigner, elle est forcée, lors du déclenchement de la tâche de ponte, de revenir auprès des autres. Les reines sont donc, à la différence des ouvrières, soumises à une force centripète cyclique indépendante de toute structuration spatiale et/ou sociale, ce qui réduit à néant toute tentative de répartition spatiale, et, par là même, toute possibilité de division du travail entre elles.

L'hypothèse que nous avons alors formulée est que l'ajout d'un nouveau comportement, forçant les reines à s'éviter au sein du nid, pourrait améliorer cet état de fait. Il s'agissait là, bien entendu, d'une hypothèse d'école, dans la mesure où ce comportement n'avait jamais été observé chez *Ectatomma ruidum*. Celui-ci était néanmoins basé sur certaines observations réalisées dans d'autres espèces d'hyménoptères sociaux, essentiellement dans le cadre des rapports de dominance entre individus (voir par exemple [Drogoul, Corbara et Fresneau 1993; Hölldobler et Wilson 1990; Theraulaz 1991], ce dernier article développant un peu plus cet aspect). L'ajout de ce comportement a consisté à faire propager par les reines leur stimulus propre (`#queen`, qui auparavant n'était pas utilisé), et à leur fournir une tâche (`queen`) déclenchée par ce stimulus et consistant à fuir le gradient formé par sa propagation. Pour s'assurer de son déclenchement en présence du stimulus, cette tâche a été dotée d'un poids relativement important (*i.e.* 100, soit 25 fois plus qu'une tâche comme `egg`)⁶⁴.

⁶⁴ [Czternasty 1993] a étudié l'impact de la variation de ce poids sur la dynamique de la sociogenèse. Faute de place, nous ne parlerons pas ici des résultats qu'il obtient, mais le lecteur pourra se reporter à ce travail, qui complète parfaitement le nôtre en ce qu'il analyse le comportement de répulsion avec spécialisation comme un facteur de dominance.

V.7.2. Expérimentations et résultats expérimentaux

Nous avons relancé une série d'expérimentations identique à celle de la section précédente, mais avec cette fois les nouvelles reines. Et, si les résultats, présentés dans le Tableau II.4, ont dépassé nos espérances, ce n'est pas tout à fait dans le sens où nous les attendions. Il s'avère en effet que l'introduction de ce nouveau comportement a extraordinairement réduit la mortalité des jeunes colonies en faisant passer leur taux d'échec de 80% (cas précédent) à 15,15%.

Résultats	Composition	Nombre	Proportion
Nombre Total d'Echecs		5	15,15%
Nombre Total de Succès		28	84,85%
Succès avec...	une reine	20	71,43%
	deux reines	3	10,71%
	trois reines	3	10,71%
	quatre reines	2	7,14%
Nombre Total d'Expériences		33	100,00%

Tableau II.4 - Proportion d'échecs et de succès des colonies polygynes avec comportement de fuite. Les pourcentages de la rubrique "Succès..." sont calculés par rapport au nombre total de succès.

Ce qui est en revanche inattendu, c'est que presque trois succès sur quatre se caractérisent par une monogynie secondaire (ce terme désigne les sociétés adultes monogynes obtenues à partir de fondations polygynes), et non par une vraie polygynie, qui ne représente que 7% des succès. La dynamique d'évolution des sociétés a donc été complètement bouleversée par le simple ajout de ce comportement de fuite, et ne correspond plus du tout à celle obtenue avec le modèle de comportement d'*Ectatomma ruidum*. Mais le plus surprenant, maintenant, est que l'obtention de cette monogynie secondaire correspond à un cas de figure très fréquent chez les autres espèces de fourmis, beaucoup plus fréquent que les cas de vraie polygynie ou de polygynie secondaire [Hölldobler et Wilson 1990]. En nous éloignant du modèle initial dérivé d'*E. ruidum*, nous sommes donc retombés sur des modèles d'évolution existants. Il convient cependant d'étudier la manière dont cette monogynie secondaire apparaît. En effet, selon [Jaisson 1993], les deux situations les plus caractéristiques, si l'on exclut les situations de dominance entre reines (dominance qui se traduit dans les faits par une inhibition du développement ovarien des reines dominées), sont: (1) la mort rapide, après deux ou trois semaines, de toutes les reines sauf une en raison d'agressions réciproques (l'exemple donné est l'espèce *Pheidole pallidula*); (2) l'écartement progressif et la mort (de faim et par manque de soins) de toutes les reines sauf une lorsque les ouvrières apparaissent (l'exemple donné est la fourmi *Myrmecocystus mimicus*). Or, ces deux cas de figure se retrouvent presque exactement dans nos expériences menant à une monogynie secondaire, le premier étant cependant largement majoritaire (14 cas contre 3 au second, et 3 non déterminés). On en prendra comme exemple les deux courbes présentées sur les Figure II.31 et II.32, qui correspondent respectivement aux cas (1) et (2).

V.7.3. Interprétation des résultats

Dans notre simulation, les agressions, qui ne correspondent pas à un comportement, sont remplacées par les fuites "en boucle" des reines les unes vis-à-vis des autres, ce qui, en soit, revient au même puisque, à force de se faire fuir mutuellement (donc de ne déclencher que cette tâche à l'exclusion de toute autre), elles meurent de faim. Nous sommes donc en mesure de donner une explication comportementale et interactionniste à ce phénomène global.

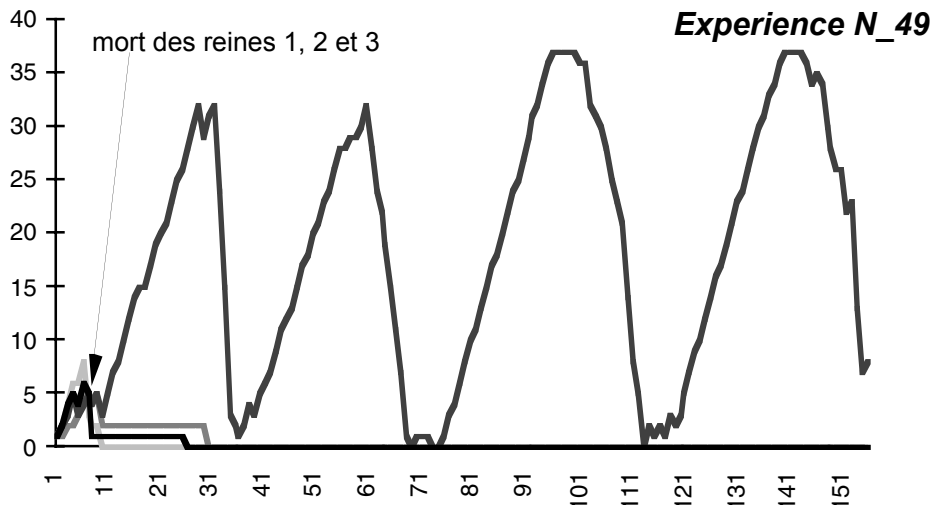


Figure II.31 - Evolution des lignées d'oeufs de l'expérience N_49.

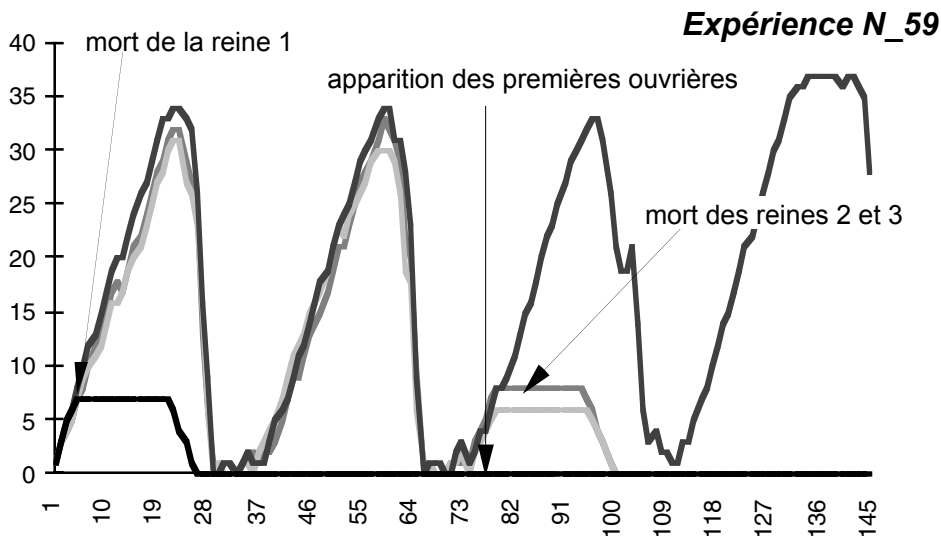


Figure II.32 - Evolution des populations d'oeufs des quatre reines de l'expérience N_59.

La sociobiologie, qui s'est intéressée de près à ces phénomènes [Hölldobler et Wilson 1990; Jaisson 1993], fournit une tout autre interprétation des mêmes situations, y voyant la manifestation d'un comportement *altruiste* de la part des reines sacrifiées dû à la proximité génétique qu'ont généralement entre elles les reines fondatrices. Les sociobiologistes attribuent en effet à tout être vivant un rôle de "vecteur" de ses propres gènes, qu'il va chercher à propager dans son espèce en se reproduisant, ou en aidant ses apparentés à se reproduire (stratégie formalisée par [Hamilton 1964] sous le nom de *théorie de la parentèle*). Comme nous l'avons vu plus haut, la mort de 3 des 4 reines permet à la société, dans 85% des cas, de réussir sa sociogenèse, alors que ce pourcentage serait de 20% si elles survivaient. En mourant, elles contribuent donc à augmenter la probabilité que les gènes de la reine survivante se propagent, et, comme elles lui sont apparentées, qu'une grande partie de leurs propres gènes soient également propagés. Que fournissent nos résultats expérimentaux dans ce cadre ? Ils dévoilent des individus incapables de discriminer les différentes lignées, donc incapables de reconnaître les individus qui leur sont apparentés, arrivant, par un simple jeu d'interactions non-intentionnelles, à reproduire des situations pouvant être interprétées sans problème aucun par la théorie de la parentèle. Remettent-ils

en cause cette théorie ? Non, mais ils remettent en cause l'un de ses postulats, selon lequel la *reconnaissance de parentèle* est une précondition nécessaire à tout acte d'altruisme. Comme le dit [Jaisson 1993]: «*La parentèle n'est donc pas la clé exclusive de l'interprétation évolutive de toutes les sociétés d'hyménoptères. Il est possible qu'elle ne joue qu'un rôle secondaire pour la poursuite de l'évolution des sociétés, une fois la socialité apparue*» (p. 135). Le modèle de comportement que nous utilisons apparaît donc comme un modèle alternatif, en ce qu'il permet d'expliquer l'apparition synchronique des formes observées au niveau social. Et la théorie de la parentèle conserve toute sa valeur explicative pour rendre compte, par exemple, de l'émergence diachronique de ce modèle, peu à peu compilé sous la forme de comportements élémentaires basés sur la *confiance absolue* à l'égard des autres.

Cette série d'expérimentations nous a permis de montrer la puissance potentielle de la notion d'éthologie synthétique, mais également ses limites. Elle ne peut, en effet, se permettre de remettre en cause des théories appliquées au monde vivant sur la simple observation de mondes artificiels, aussi bien construits soient-ils. Elle peut en revanche servir à édifier et vérifier des hypothèses qui nécessiteraient de longues et difficiles expérimentations dans la nature, hypothèses dont les éventuelles implications théoriques doivent cependant être validées *in vivo*. C'est en ce sens qu'il faut la comprendre, et c'est en ce sens que nous avons utilisé MANTA sur la question de la polygynie. On en voudra pour preuve le fait que des expériences sont actuellement effectués au Laboratoire d'Ethologie de l'Université Paris XIII, sous la direction de Dominique Fresneau, sur des fondations polygynes de sociétés qui reprennent les mêmes conditions initiales et les mêmes hypothèses que celles utilisées dans la plate-forme de simulation, afin de vérifier le bien-fondé biologique des expériences que nous venons de décrire et des résultats que nous avons obtenu.



CONCLUSION DE LA DEUXIEME PARTIE

Généralités

L'ensemble des expériences que nous avons décrites et commentées dans cette partie constitue la majeure partie du travail réalisé à l'heure actuelle avec MANTA. Elles nous ont permis de mettre en évidence l'aspect radicalement novateur de ce type de simulation, et le gain conceptuel que l'on pouvait en attendre. Il est en effet nécessaire de préciser ici, sans pour autant en faire une publicité démesurée, que MANTA constituait au départ le projet de simulation éthologique le plus ambitieux de tous ceux que nous avons eu l'occasion d'étudier, et qu'il est à l'heure actuelle le plus abouti, à la fois conceptuellement et expérimentalement. Comme nous le soulignons déjà au Chapitre II, une bonne théorie de modélisation est une théorie qui permet au chercheur d'exprimer le modèle sous la forme abstraite qu'il a l'habitude d'employer pour décrire la réalité. Avec le noyau d'EthoModélisation, nous pensons avoir contribué à faire avancer cette perspective dans le domaine éthologique, en permettant aux éthologues d'exprimer leur connaissance du comportement sous la forme de comportements. Qui plus est, il a permis de faire de cette manière d'un projet de simulation un véritable projet interdisciplinaire, où les aller-retours constants entre réalité et virtualité ont sans doute permis d'affiner aussi bien la vision des éthologues sur les fourmis que la vision que nous avons sur les agents réactifs (voir plus bas).

Le second point fondamental est que nous avons effectué avec MANTA une "réelle" reconstruction d'une société de fourmis. Cette reconstruction nous a permis de l'étudier avec les outils dont disposent habituellement les éthologues, tout en nous donnant la possibilité de faire varier de manière impensable dans la réalité des paramètres comportementaux. Elle nous a également permis de suivre pas à pas la naissance de ces colonies artificielles et donc de mieux comprendre les premiers stades des processus d'auto-organisation à l'oeuvre dans chacune d'entre elles.

Validation de l'approche Multi-Agents

Les objectifs que nous nous étions fixés ont-ils été pour autant atteints ? Le premier, qui consistait à valider l'approche multi-agents dans le domaine des sciences de la vie, a sans nul doute été partiellement atteint, mais MANTA ne constitue pas une démonstration en soi de la validité générale du modèle. Une généralisation à d'autres domaines de l'éthologie, de la biologie, ou même de la sociologie est donc un préalable indispensable à toute validation de cet aspect. A cet égard, l'intérêt manifesté par de nombreux laboratoires d'éthologie, en France et à l'étranger, à propos de MANTA, et l'utilisation probable de l'EthoModélisation comme support d'un projet européen de modélisation économique (MAGENTA), nous permet d'envisager l'avenir avec sérénité.

Le travail effectué par plusieurs étudiants de DEA (Steffen Lalande, Nicoals Czernasty, Alain Touret et Philippe Louise) sur les expérimentations que nous avons présentées ici nous a également permis de constater certaines limites de l'EthoModélisation, notamment en terme de facilité de développement. Ce point nous importe, car nous souhaitons, un peu à la manière de [Resnick 1991] avec *Logo, faire de ce système un véritable système de développement accessible au plus grand nombre, et notamment aux éthologues, souvent peu familiers des langages de programmation. Une deuxième version est actuellement en cours de programmation, version dans laquelle nous espérons inclure des possibilités d'apprentissage de tâches par les agents au moyen d'un mécanisme à base de classifieurs (voir par exemple [Delaye et Ferber 1992]), dans une optique d'aide à la programmation des tâches. L'idée

principale consisterait à programmer les primitives ainsi que les stimuli, en laissant ensuite le soin au système de proposer des ensembles de tâches consommatrices plausibles. Cette version, si elle s'avérait productive, nous serait alors d'un grand secours pour le développement de simulations ambitieuses et permettrait de faire du système d'EthoModélisation un véritable laboratoire d'éthologie synthétique.

Validation du modèle de Comportement

Le deuxième objectif, qui consistait à valider le modèle de comportement et le répertoire comportemental employés par les éthologues, a été largement atteint. Mis à part le poids par trop important du fourragement dans le profil comportemental de la société, poids que nous pouvons expliquer par des problèmes d'interprétation du comportement des agents, lesquels devraient être sans doute rapidement résolus (voir Section V.4.), nous arrivons à reproduire sans aucune difficulté la dynamique d'évolution des sociétés, aussi bien au niveau de la démographie qu'au niveau du polyéthisme. Ceci nous permet donc d'affirmer que les mécanismes purement ontogénétiques qui ont été retenus peuvent constituer de bons modèles explicatifs de l'évolution des sociétés naturelles. Mais nous ne remettons pas en cause, avec une telle simulation, la validité des modèles théoriques existants, comme ceux proposés par la sociobiologie. Elle nous autorise, tout au plus, à considérer sous un nouvel angle certains des postulats sur lesquels se basent ces théories, et à proposer de prudentes alternatives, sachant qu'il revient, en dernier lieu, aux éthologues de les valider ou non.

Le travail dans ce sens devrait être poursuivi d'ici peu avec le lancement d'une nouvelle campagne d'expérimentations portant sur la corrélation entre la répartition sociale et la répartition spatiale des individus, sur l'évolution à très long terme (plusieurs années) des sociétés artificielles déjà obtenues, et enfin sur l'aspect écologique des colonies de fourmis, par l'intermédiaire de leurs relations avec des environnements plus riches et compétitifs. Nous envisageons ainsi d'étudier les modalités d'interaction pouvant exister entre deux colonies qui se partagent un même environnement. Enfin, l'étude faite sur la polygynie nous permet d'envisager la modélisation de sociétés d'autres espèces de fourmis (ce qui a déjà été fait en partie par Alain Touret avec un modèle des stratégies de fourragement de l'espèce *Neoponera apicalis*), et de sans doute d'autres insectes sociaux, ce qui rejoindrait ainsi la nécessité de généralisation exprimée à la Section précédente.

Etude des Phénomènes d'Auto-Organisation

Le dernier point de cette conclusion nous permet de faire la liaison avec la partie suivante, partie dans laquelle nous allons abandonner nos fourmis artificielles pour nous pencher sur les capacités de résolution de problèmes des systèmes multi-agents réactifs. La réussite du projet MANTA nous a prouvé qu'il était tout à fait sensé d'espérer concevoir des organisations réactives fonctionnelles, même si la fonctionnalité obtenue ici n'a que peu à voir avec les fonctionnalités que nous avons l'habitude d'exiger de nos outils logiciels. Ce résultat n'allait en effet pas de soi, et nous a donc sérieusement conforté dans notre approche. Cependant, l'apport le plus intéressant de ce projet à notre problématique réactive ne réside pas tant dans l'obtention de ce résultat que dans la manière dont il a été obtenu. La démarche constructiviste que nous présentions au Chapitre I a en effet pris tout son sens dans l'interaction forte que nous avons eu, en tant que concepteur *et* observateur, avec le système. *Construire* le système et le voir *construire* sa propre organisation nous a permis de réellement *construire* les connaissances nécessaires à sa compréhension. En ce sens, nous pouvons tout à fait estimer que notre démarche a autant façonné MANTA qu'il l'a façonnée. La manière dont nous envisageons maintenant la résolution distribuée de problèmes (Chapitre VI) est à cet égard éclairante, puisque nous passerons insensiblement de problèmes pour lesquels la solution se trouve dans le comportement des fourmis simulées à des problèmes plus abstraits pour lesquels nous mettrons à profit les connaissances sur les

modes d'auto-organisation d'agents réactifs, alors que les agents utilisés n'auront plus rien à voir avec des fourmis. La démarche eût sans doute été plus délicate si nous avions étudié des colonies réelles, car nous n'aurions pas eu connaissance, en l'occurrence, du fonctionnement intime de chacun de leurs composants. De même, la construction arbitraire d'un système à base d'auto-organisation ne nous aurait pas permis de valider son fonctionnement en le comparant à des phénomènes existants.

Un important travail de formalisation reste à effectuer pour dégager de ces connaissances des règles applicables à d'autres cas de figures. C'est sur ce point que nos efforts se porteront à l'avenir, en complément des perspectives expérimentales évoquées plus haut.



Troisième Partie

La Résolution Collective de Problèmes

We need to move away from state as the primary abstraction for thinking about the world.
Rather we should think about processes which implement the right thing [Brooks 1987]

AVANT-PROPOS

Nous allons nous attacher, dans cette partie, à intégrer le modèle d'agents que nous avons utilisé pour la simulation et à exploiter les propriétés d'auto-organisation et d'émergence que nous a fourni MANTA dans des applications de résolution de problèmes.

Notre but est de démontrer que la *résolution collective de problèmes*, qui est une manière d'interpréter fonctionnellement l'activité d'une colonie de fourmis, terme que nous distinguerons dans notre introduction de la résolution distribuée de problèmes dans son acception actuelle, est capable de s'appliquer aussi bien à des problèmes naturellement distribués et potentiellement susceptibles d'une modélisation réactive qu'à des problèmes dont la centralisation semble tellement aller de soi que personne ne s'est jamais penché sur leur distribution. Nous partirons ainsi de problèmes maintenant classiques en Vie Artificielle, qui sont le Tri Collectif et la programmation de Robots Fourrageurs, dans cette Partie, pour aborder les domaines de la planification et de la recherche heuristique, par le biais du Taquin. Nous marquerons la transition entre ces deux domaines par l'étude de Pengi (Quatrième Partie).

Cette partie, ainsi que la suivante, peuvent se lire à plusieurs niveaux, et nous n'en privilégions aucun de manière à laisser le lecteur libre de son choix. La première lecture est conceptuelle. Au gré des applications, nous verrons comment intégrer les propriétés d'auto-organisation que nous avons évoquées au chapitre précédent à propos des fourmis dans des applications de résolution de problèmes. Nous en montrerons les potentialités, de même que les limites. Pour chacun des exemples proposés, nous insisterons sur l'instanciation des concepts de résolution collective de problèmes, afin de familiariser le lecteur à la fois à la terminologie que nous employons et à la méthodologie que nous suivons. Nous reviendrons d'ailleurs sur certains aspects méthodologiques en conclusion de cette Partie.

La seconde lecture est logicielle. Nous montrons peu à peu comment passer d'un système "ouvert" au niveau des comportements comme l'EthoModélisation, dont la résultante est difficilement formalisable, à un système déjà existant, plus contraint au niveau comportemental, qui est l'EcoRésolution (Quatrième Partie), tout en conservant les propriétés qui nous semblent essentielles.

La troisième lecture est enfin applicative. Nous montrerons ainsi qu'une approche réactive constitue pour certains problèmes, comme le Taquin, par exemple, une approche à la fois élégante et plus efficace que toutes les approches envisagées jusqu'à présent. Et nous terminerons cette étude en envisageant les applications futures de notre méthodologie de résolution collective de problèmes.



CHAPITRE VI

RESOLUTION COLLECTIVE DE PROBLEMES

VI.1. Résolution Distribuée de Problèmes

VI.1.1. Intelligence et Résolution (Distribuée) de Problèmes

D'une certaine manière, tout comportement d'un agent isolé ou d'un groupe d'agents peut être *décrit* comme une réponse fonctionnelle faisant partie de la résolution d'un problème qui lui est posé. En ce sens, un comportement quel qu'il soit fait toujours partie d'une activité plus générale de résolution. Nous définissons ainsi par problème la *description* d'une situation qui peut être caractérisée, de l'extérieur, par une ou plusieurs situations connexes qui consistent en des buts à atteindre et par un ensemble de facteurs qui représentent des contraintes.

L'intelligence, dans ce cadre, est alors perçue comme la capacité d'un agent à adapter au mieux sa réponse fonctionnelle à la situation en cours, même s'il ne l'a jamais auparavant rencontrée, afin d'atteindre une des situations connexes, que ce soit par exploration d'espaces d'états, apprentissage ou n'importe quelle autre technique manipulant des représentations de ces situations [Jacopin 1992]. Un agent ou un groupe d'agents sera donc considéré comme intelligent si et seulement si il se montre capable d'appréhender des problèmes pour lesquels on ne lui a pas fourni de solution immédiatement exploitable (dans le cas contraire, c'est plutôt l'intelligence du concepteur qui sera louée). On peut donc dire que la forme opératoire et observable de l'intelligence se caractérise par la capacité à résoudre des problèmes considérés comme complexes par un observateur. Comme nous l'avons déjà rapidement évoqué dans la première partie, la vision cognitive de cette capacité consiste à subodorer, qu'au sein d'un agent, l'actualisation des capacités fonctionnelles dans une situation donnée est *contrôlée* par un mécanisme (le plan de contrôle) qui tire partie d'un ensemble de connaissances, qui sont le plus souvent des expertises, c'est-à-dire des représentations abstraites de situations et de processus de résolution afférents. Comme le dit [Collinot 1988] : «(...) *pour bien résoudre [un problème] il faut effectuer les bons choix quand aucun algorithme ne précise la marche à suivre. Pour cela il faut utiliser des connaissances à bon escient - et donc des connaissances sur l'opportunité d'utiliser des connaissances*».

Par une démarche qui semble naturelle, la vision cognitive de la résolution distribuée ou multi-agents de problèmes ne travaille donc que sur des agents individuellement intelligents qui, une fois réunis, tenteront (en utilisant leur propres connaissances) de générer un plan de contrôle pour l'ensemble du groupe, par des processus de négociation et de partage d'informations, plan de contrôle qui ne coordonnera plus seulement l'activité d'un agent, mais l'activité du groupe tout entier (cf. par exemple la notion de Plan Partiellement Global de [Durfée 1989]). A l'heure actuelle, cette conception occupe tout le champ (ou presque, nous le verrons par la suite) de la résolution distribuée de problèmes (voir par exemple les Actes des derniers DAI Workshop). Il y a certes des raisons historiques à cette prédominance, raisons historiques qu'il faut sans doute rechercher dans le fait que l'IAD est plus

souvent perçue comme une extension de l'IA que comme une voie alternative (voir Chapitre I)⁶⁵. Cette vision se retrouve par exemple dans l'ambiguïté même de l'appellation *Distributed Problem Solving*, qui peut aussi bien se traduire par *Résolution Distribuée de Problèmes* que par *Résolution de Problèmes Distribués* - si l'on y met un peu de mauvaise foi.

VI.1.2. *Résolution de Problèmes et Intelligence*

Or, dès le moment où l'on ouvre la boîte de Pandore de l'intelligence distribuée, c'est-à-dire dès le moment où l'on *pense* le fait social en tant que tel et non plus comme une simple agrégation de processus distincts, l'on se trouve à notre avis confronté à une multitude de solutions quant au choix des individus qui composent la société ou le collectif confronté à un problème. La prédominance des choix influencés par la psychologie cognitive repose vraisemblablement sur la difficulté de l'homme à concevoir d'autres formes d'intelligence collective que la sienne. A ce titre, le postulat de base qui sous-tend à l'heure actuelle tout le domaine de la résolution distribuée de problèmes apparaît comme un parfait sophisme. Postulant d'une part que l'intelligence collective se manifeste par la capacité de groupes d'entités à résoudre des problèmes considérés comme complexes (c'est la *majeure* du syllogisme), et d'autre part que les groupes humains sont capables de résoudre des problèmes considérés comme complexes (voici la *mineure*), l'on postule tout naturellement qu'une intelligence collective ne peut être exprimée que par des groupes humains - ou anthropomorphes, bien sûr, si l'on parle d'agents (c'est la *conclusion*). Sachant donc que tous les hommes sont mortels et que tous les grecs sont mortels, l'on en déduit tous les hommes sont grecs... Ce sophisme, que l'on retrouve dans un nombre élevé de publications sous des formes diverses, a fait que l'essentiel de la recherche actuelle en *Résolution Distribuée de Problèmes* se concentre à l'heure actuelle sur l'implémentation des moyens cognitifs mis en oeuvre par l'homme pour résoudre des problèmes, en négligeant ainsi la plupart des autres voies, dont la voie biologique et, bien sûr, éthologique.

Si l'on en juge par la teneur de la plupart des publications qui y sont consacrées, le danger qui guette cette approche, dans une vision de résolution de problèmes, est de tomber dans le travers *idéaliste* (auquel nous nous opposons en préconisant une approche sinon matérialiste, du moins pragmatique) qui consiste à concentrer à la fois le raisonnement des agents et l'effort de conception du système, non pas sur le problème que l'on souhaite qu'ils résolvent, mais sur les problèmes (de coopération, d'organisation) que leur pose cette résolution. Ainsi, comme le note [Bouron 1992], à propos du concept de Plan Globalement Partiel de [Durfee 1989], «*Deux problèmes sont résolus simultanément [par les agents]: le problème lié à l'application et le problème de la coordination*» (p. 72). Il n'y a bien sûr pas lieu de critiquer ici cette approche, car elle est certainement justifiée dans le cadre d'une vision anthropomorphe de la résolution distribuée de problèmes. L'être humain, tout comme d'ailleurs certains animaux, dont essentiellement les primates [Thierry 1991], passe clairement une partie de son temps à adapter et à modifier les organisations auxquels il participe en vue de résoudre certains problèmes⁶⁶.

⁶⁵ La conception américaine de l'IAD, reprise en partie dans [Sichman, Demazeau et Boissier 1992], consiste à distinguer deux champs de recherche, qui sont respectivement la *Résolution Distribuée de Problèmes* et les *Systèmes Multi-Agents* (ou *Systèmes Ouverts*), le premier étant vu comme une extension de l'IA aux problèmes distribués par nature, le second comme ce que nous appelons nous l'IAD.

⁶⁶ L'exemple de l'organisation d'un dîner entre amis où chacun doit apporter une partie du repas est à cet égard éclairant. Le problème étant d'obtenir un repas complet, chacun des invités passe certainement plus de temps au téléphone à tenter de coordonner avec les autres invités ses actions et ses choix liés à un sous-problème particulier (préparation du dessert, des entrées, du plat, etc.), qu'à le résoudre. (exemple proposé par Jacques Ferber, comm. pers.)

Ce que nous critiquons par contre est la focalisation sur la construction *intentionnelle* de cette coordination, dont nous pensons qu'elle ne se justifie que rarement, en tout cas pour les problèmes jusqu'à présent posés aux agents. Nous avons vu avec les fourmis qu'une organisation flexible d'allocation des tâches pouvait émerger des interactions entre un groupe d'agents et son environnement. Il a été également montré, dans le cadre de l'application proie-prédateurs, maintenant bien connue en Intelligence Artificielle Distribuée, à la fois dans [Korf 1991], dans [Drogoul en prép.] et, dans une moindre mesure, dans [Jacopin 1992], qu'une organisation d'agents basée sur de simples mécanismes de perception, d'attraction et de répulsion pouvait largement concurrencer les organisations intentionnelles complexes envisagées dans d'autres travaux (ex. [Stephens et Merx 1990] ou [Bouron 1991]). Nous pouvons donc raisonnablement espérer que bon nombre de problèmes soient résolus collectivement par des organisations d'agents, sans que ces organisations aient besoin d'être *pensées* par les agents. Nous appliquons ainsi à l'organisation les principes de parcimonie et d'incrémentalité que nous proposons au Chapitre I pour les agents, à savoir que des organisations intentionnelles d'agents ne doivent être envisagées que quand tous les autres moyens nécessitant des capacités cognitives moindres de parvenir à des organisations efficaces et adaptatives ont été épuisés.

Ce que nous postulons dans cette thèse, et ce que nous espérons démontrer dans les sections suivantes, est qu'il existe des voies alternatives à l'approche cognitive de la résolution distribuée de problèmes. Nous faisons de plus clairement nôtre une vision plus pragmatique, donc empirique, que théorique de la résolution distribuée de problèmes, en soutenant, à tort ou à raison, et l'avenir seul le dira, que c'est en cherchant à résoudre des problèmes complexes que nous trouverons comment concevoir des systèmes intelligents. Voici pourquoi nous appellerons dans la suite de ce chapitre notre démarche Résolution Collective de Problèmes. Ce terme ne possède pas de connotations cognitives particulières (à l'inverse de la Résolution Distribuée de Problèmes actuelle), et les mentions qui y ont été déjà faites, notamment dans [Deneubourg, Theraulaz et Beckers 1992] (qui reprend l'essentiel de ce qui est développé dans [Theraulaz 1991]) correspondent tout à fait à notre projet de recherche. La section suivante expose quelques-uns de ses postulats de base, postulats dont nous nous servirons dans la suite du chapitre.

VI.2. Résolution Collective de Problèmes

VI.2.1. Les Travaux existants

C'est depuis l'apparition de la Vie Artificielle comme domaine que des auteurs ont commencé à se pencher sur les capacités collectives de résolution de problèmes de certaines espèces d'animaux sociaux en abordant simultanément les applications qui pouvaient en être dérivées en informatique comme en robotique.

Le mérite en revient à J-L. Deneubourg et à son équipe, qui ont mis l'accent sur l'intérêt fonctionnel de certains processus distribués qui prennent place au sein des sociétés d'insectes sociaux. Ainsi, les processus de construction [Deneubourg, Theraulaz et Beckers 1992], processus qui nous intéressent car certains mêlent étroitement la résolution de deux problèmes distincts qui sont la construction elle-même et l'allocation de ressources par des phénomènes de gestion distribuée de files d'attente [Corbara, comm. pers.]. Mais également, toutes les problématiques liées au fourragement (voir par exemple [Deneubourg et al. 1986], et voir également Chapitre VIII) ou certaines fonctionnalités transposables à des problématiques logicielles, comme celle du "Tri Collectif" [Deneubourg et al. 1991] (voir Chapitre VII). Puisque nous avons intitulé notre démarche Résolution Collective de Problèmes, il est juste de mentionner également l'excellente contribution de [Theraulaz 1991], à qui nous avons

emprunté le terme. Dans l'ensemble, cependant, cette démarche reste expérimentale et les tentatives de formalisation du concept d'"intelligence en essaim" [Deneubourg, Theraulaz et Beckers 1992], par exemple, ne sont pas encore abouties. Si, en termes de simulation multi-agents, nous reprochions à ces travaux une abstraction trop importante pour pouvoir prendre en compte la complexité des phénomènes étudiés, nous pouvons, dans une vision multi-agents, formuler la critique inverse en leur reprochant de ne pas suffisamment s'abstraire du domaine étudié. C'est en cela qu'ils diffèrent par exemple de l'approche de Luc Steels, présentée dans [Steels 1989] et reprise dans [Steels 1991], qui, tout en se basant sur le comportement de fourrageage des fourmis, parvient à intégrer son travail dans une problématique délivrée de tout "éthologisme", centrée autour des *fonctionnalités émergentes* dont nous donnons un aperçu, et quelques critiques, à la fin du Chapitre VIII. Cette démarche est d'ailleurs similaire à celle proposée par [Connah 1991], l'exemple des fourmis (déjà très populaire) servant d'analogie pour justifier une démarche réactive de résolution de problèmes.

Il y a encore peu de travaux portant sur l'utilisation de telles analogies pour résoudre des problèmes "classiques" en informatique. A ce titre, il convient donc de citer [Colomi, Dorigo et Maniezzo 1992a; Colomi, Dorigo et Maniezzo 1992b] qui utilisent un "Ant Algorithm" pour résoudre le fameux problème du voyageur de commerce, même si nous estimons que dans ce cas la ficelle est un peu grosse, puisque les agents utilisés n'ont guère de fourmi que le nom⁶⁷. Même s'il n'est pas pour l'instant terminé, le travail de [Hatcher, Tofts et Francks 1992] sur les problèmes de synchronisation, d'allocation de tâches et d'exclusion mutuelle au sein d'une colonie semble par contre plus prometteur.

VI.2.2. Caractéristiques

VI.2.2.1. LE POSTULAT DE DISTRIBUTION REACTIVE

De prime abord, nous pouvons caractériser cette démarche par un ensemble de postulats, qui sont à prendre pour ce qu'ils sont, c'est-à-dire des assertions dont nous ne cherchons pas à établir la validité, mais dont nous montrerons l'importance dans les sections suivantes. Ces postulats vont nous permettre d'insérer progressivement les différents concepts que nous avons manipulés au sein du projet MANTA dans une démarche de résolution de problèmes.

Le premier postulat est le postulat de **distribution réactive**. C'est un postulat fort, mais, à y bien réfléchir, guère différent de ceux de réification et de décomposition utilisés en programmation orientée objets [Ferber 1990; Perrot 1990]. Comme le dit [Ferber 1989] à propos de la réification, il s'agit **du** postulat, qui ne peut être ni réfuté ni prouvé et est à prendre comme tel.

Postulat de Distribution Réactive

Tout problème peut être décomposé en un environnement et un ensemble d'agents réactifs qui évoluent et peuvent agir dans cet environnement.

Le terme "agents réactifs" employé dans ce postulat fait bien entendu référence aux définitions données au Chapitre I. Précisons tout de suite que la difficulté d'application de ce postulat provient pour une large part de l'absence de toute méthodologie de "distribution orientée agents" d'un problème. L'état

⁶⁷ Cet algorithme se base sur la capacité qu'a une colonie de choisir, parmi deux chemins possibles d'un point à un autre, le plus court. Cette capacité est basée sur le temps mis à parcourir ces chemins, et donc sur le renforcement plus rapide (par dépôt de phéromones) du plus court des deux. Or, si l'on analyse de plus près la manière dont les agents fourmis voyagent de ville en ville dans le problème, l'on s'aperçoit qu'ils le font instantanément... et qu'ils laissent dans chaque ville une trace contenant la distance aux villes voisines afin d'orienter les autres agents. Il s'agit donc plus d'un algorithme parallèle de recherche que d'une utilisation du mécanisme auto-catalytique de recrutement des fourmis.

actuel de nos travaux ne nous permet pas d'en proposer une, mais il ne fait guère de doute qu'avec le développement actuel des méthodologies de décomposition orientée objet, nous ne soyons en mesure d'en disposer d'ici peu. Ce postulat induit tout naturellement le corollaire suivant:

Corollaire 1.1

L'état initial d'un problème, son état final et la suite des états intermédiaires entre ces deux états peuvent donc être chacun décrits par une configuration particulière de l'environnement et/ou une configuration particulière du groupe d'agents.

VI.2.2.2. DE L'ETAT A LA STRUCTURE

Le deuxième postulat est le postulat de **congruence structurelle**. Il convient ici d'être particulièrement attentif au glissement que nous opérons, puisque nous ne parlons plus de l'état d'un problème mais de sa *structure* (les deux mots *configuration* et *organisation* seront employés indifféremment dans la suite de ce chapitre avec le même sens que *structure*). Là où l'état d'un problème va par exemple être représenté par la liste des positions absolues de chacun des agents sur un échiquier, la *structure* va dénoter la manière dont ces agents sont agencés entre eux. Plutôt qu'une description quantitative, une structure va apparaître comme une description qualitative d'une situation, avec, bien évidemment, la subjectivité que cela risque d'entraîner. C'est un phénomène observable qui se reconnaît à certains invariants et qui ne tient pas compte, en dessous d'un certain seuil, de la variabilité interindividuelle (voir Chapitre I). Ce n'est donc pas une propriété de l'agent mais du groupe d'agents. Ainsi, toute modification de la position d'un agent fera passer le problème d'un état à un autre, mais pas forcément d'une structure à une autre. Comme les états, les structures observées appartiennent à un espace de structures possibles. Cependant, à une structure donnée ne correspondra pas forcément un point dans cet espace, car il pourra s'agir d'un attracteur dénotant sa dynamique. Le passage d'un état à un autre se notant *transition*, nous emploierons également le terme *transition* pour qualifier le passage d'une structure à une autre et le terme *structuration* pour qualifier l'apparition ou la modification progressive d'une structure.

Postulat 2 (dit de Congruence Structurelle)

La dynamique de la résolution du problème résulte d'un double processus de structuration d'une part du groupe d'agents par leur environnement et par leurs interactions, et d'autre part de l'environnement par les activités des agents.

Ce postulat est identique dans l'esprit au postulat de *causalité circulaire*, souvent employé dans la description de phénomènes d'auto-organisation [Dumouchel et Dupuy 1983], mais en diffère en ce qu'il ne se limite pas à des niveaux auto-organisés hiérarchiques. La structuration du groupe d'agents peut prendre différentes formes, qu'elle soit sociale (répartition des tâches, des rôles, des compétences), physique (configuration particulière qui se maintient dans l'environnement), temporelle (séquençement d'exécutions, synchronisation entre agents). Puisque nous employons des agents réactifs, qui a priori n'ont pas la possibilité de contrôler eux-mêmes l'activité du groupe dont ils font partie, nous obtenons le corollaire suivant:

Corollaire 2.1 (dit de Mémorisation Structurelle)

Seules la structure de l'environnement et la structure du groupe d'agents peuvent orienter par leur influence locale le comportement de chacun des agents. Ces deux structures réunies constituent donc au niveau global la mémoire de la résolution en cours.

Puisque d'autre part ces agents n'ont pas de vue globale de leur environnement, et donc de la résolution du système, l'état final du système (la solution du problème) est atteint soit par l'adéquation d'une des deux configurations, ou des deux configurations simultanément, au souhait de l'utilisateur (la définition d'un critère d'arrêt se fait donc sur l'apparition ou la disparition d'une structure particulière du système), soit par un état stable où le double processus de structuration ne fait plus que reproduire les mêmes structures (ce peut être un état d'équilibre dynamique, ou bien un attracteur de la dynamique de structuration).

Corollaire 2.2

L'état final du problème, donc sa solution, est caractérisé par l'arrêt de ce double processus. Il correspond à une configuration stable ou stationnaire, soit de l'environnement, soit du groupe d'agents, configuration qui se maintient d'elle-même, ou configuration recherchée par le concepteur.

VI.2.2.3. TELEONOMIE ET ENTROPIE

Résoudre un problème, c'est donc obtenir une structure particulière, soit du groupe d'agents, soit de l'environnement, soit des deux simultanément (voir le point intéressant fait sur ce sujet par [Theraulaz 1991], à propos du concept d'intelligence en essaim). Si maintenant nous assimilons les concepts de *structure* et d'*ordre* (sans impliquer autre chose qu'un simple point de vue analogique), la résolution devient alors la recherche de l'*entropie minimale* du problème, ce qui n'est pas loin de nous ramener aux concepts déjà exploités de manière remarquable dans un cadre multi-agents de résolution de problèmes par Khaled Ghedira [Ghedira et Verfaillie 1991]. Que nous apporte de plus cette constatation? Si nous lisons le Postulat 2 et son Corollaire 2.1, nous nous apercevons que la seule manière de faire baisser l'entropie du problème est de passer par l'intermédiaire de l'activité individuelle de chacun des agents, activité qui va être déclenchée et guidée par la configuration de l'environnement et/ou du groupe auquel appartient cet agent. Si nous voulons que les activités conjuguées des agents aboutissent à une diminution de l'entropie, il faut donc au moins que chacun des agents soit capable, par son comportement de la faire baisser localement⁶⁸. Nous admettrons ainsi que l'entropie globale du système se décompose en un ensemble de *températures locales* que les agents sont capables de percevoir. Nous pouvons donc énoncer le postulat suivant, postulat qui nous fournit quelques indices de programmation des agents et de leur environnement, à défaut de nous en fournir une méthodologie complète.

Postulat 3 (dit de téléonomie néguentropique)

Toute activité d'un agent est déclenchée et entretenue par la perception d'une "température locale". La finalité de tous ses comportements est de réduire la température locale qui les a déclenchés.

Le nom de ce postulat fait référence au concept de téléonomie défini par [Monod 1970]. Nous l'utilisons ici dans sa version la plus neutre et la plus répandue, dans laquelle un processus téléonomique est un processus qui matérialise une finalité *apparente* pour un observateur qui lui est extérieur. Un comportement téléonomique est donc un comportement dont la finalité n'est pas forcément connue de l'agent qui l'effectue, mais résulte de l'interaction entre le programme qui code ce

⁶⁸ Nous ne discuterons pas ici le fait que le système risque de se retrouver dans un état métastable, c'est-à-dire dans un minimum local d'entropie. Nous énonçons simplement que cette condition, à défaut d'être suffisante, est nécessaire, ce que personne ne contestera.

comportement et l'environnement dans lequel il est effectué (en ce sens, nous dépassons la définition de *téléonomie* donnée par [Ferber 1989], qui y voit seulement la conséquence de la manipulation par l'agent de buts explicites). Dans la décomposition que nous faisons, ce qui va pousser un agent à agir dans son environnement est donc la traduction locale dans cet environnement ou dans le groupe d'agents de l'entropie globale du système telle qu'elle est définie par le concepteur, par rapport à l'état final que lui, et lui seul, connaît. Pour comprendre l'adéquation de ce postulat, il suffit de se reporter à la manière dont les fourmis de MANTA structurent et organisent le couvain. Par l'intermédiaire des stimuli en provenance des oeufs, larves et cocons, elles sont en effet sensibles au "désordre structurel" qui affecte l'ensemble de ces agents (mauvaise localisation dans le nid, etc.) et leur comportement va être constitué de tâches consommatrices, c'est-à-dire de tâches dont la finalité sera de faire baisser l'intensité des stimuli qui les ont déclenchées (en déplaçant les agents propagateurs, par exemple). Nous nous trouvons ainsi face à un comportement téléonomique et négentropique tel que celui que nous préconisons pour les agents réactifs.

VI.2.2.4. CONCLUSION

En résumé, nous proposons donc de décrire un problème sous la forme d'une population d'agents réactifs et d'un environnement. Cette description est orientée par le fait que la solution du problème doit pouvoir s'exprimer comme l'obtention d'une ou de plusieurs structures de l'environnement ou du groupe d'agents. L'absence initiale de ces structures doit se manifester par une entropie élevée du système, entropie dont la traduction locale est effectuée par l'intermédiaire d'une température perceptible par les agents. L'intensité de cette température locale déclenche et guide les comportements de chacun des agents, comportements téléonomiques dont la finalité est de faire baisser la température qui les a déclenchés. Ainsi, l'activité de l'ensemble du groupe d'agents doit pouvoir se traduire par une baisse de l'entropie du problème, ce qui reviendra à obtenir la ou les structure(s) désirée(s), puisque c'est du fait de leur absence que l'entropie était élevée. Nous pouvons noter que la définition d'une résolution de problème en ces termes élimine de manière radicale la notion de *révision* de plan du fait de l'inadéquation entre le processus de résolution et l'état de l'environnement, question essentielle en résolution de problèmes classique. Or, toute *révision* suppose qu'au préalable existe une *vision*, ou une représentation du problème et du processus de résolution afférent par les acteurs de cette résolution. Ce n'est pas le cas ici, aucun agent ne possédant de représentation ni de son environnement, ni du groupe auquel il appartient. Toute perturbation sera donc prise en compte parmi les données "naturelles" du problème, au même titre que l'activité des agents (voir [Bura et al. 1991] qui fournit un bon exemple de cette situation).

Nous n'avons inclus dans ces propositions aucun postulat relatif à la vitesse de convergence du système vers l'état stable souhaité⁶⁹. Si nous n'envisageons pas pour l'instant cette question, c'est qu'elle nous semble relever pour une bonne part d'une volonté d'optimisation qui ne doit pas intervenir, à notre sens, dans la conception même du système, mais au sein d'un processus dynamique de co-évolution entre le concepteur/observateur et le système. Pour optimiser la réponse d'un système dynamique, il faut en effet le comprendre, et pour le comprendre, il faut l'avoir créé - et/ou observé. Et, dans la méthode constructiviste que nous suivons depuis le début de ce travail, nous en apprenons certainement autant à le programmer qu'à l'observer. Notre démarche s'inscrit donc essentiellement dans la première étape de ce processus, ce qui ne nous empêchera pas d'aborder, en aval, certains problèmes d'optimisation (cf. Chapitre VIII). Mais la manière dont nous les aborderons sera isomorphe

⁶⁹ A la différence, notamment, des chercheurs utilisant l'ingénierie génétique pour programmer et faire évoluer des populations d'agents (cf. par exemple [Atlan et Meyer 1991; Collins et Jefferson 1991; Delaye et Ferber 1992]), pour qui ce point est crucial.

à la manière dont nous abordons la résolution du problème, à savoir que l'amélioration de l'efficacité d'un système de résolution deviendra en soi un problème à résoudre, qui nécessitera l'émergence de certaines structures sociales (coopération, division du travail, coordination) ou environnementales (indications, chemins, etc.) supplémentaires. Nous insisterons sur ce point durant l'exemple du Tri Collectif (présenté Chapitre VII), et celui des robots fourrageurs (présenté Chapitre VIII). Nous ne prêtons ainsi qu'une attention distraite à l'optimalité qui différencie les "bons" solveurs de problèmes des "mauvais", car, comme nous l'avons montré dans [Drogoul et Dubreuil 1993], la difficulté que rencontrent la plupart des systèmes de résolution de problèmes n'est pas de trouver la solution optimale à partir d'une configuration initiale donnée, ce que A* par exemple est capable de faire théoriquement, mais de trouver une solution tout court, ce qui, pratiquement, pose de nombreux problèmes, à commencer par celui de l'explosion combinatoire engendrée par les systèmes dynamiques. Notre approche pragmatique consiste à d'abord trouver une décomposition adéquate pour résoudre le problème, quand bien même le coût réel de la solution apparaît intuitivement catastrophique. Et c'est à partir d'elle que nous pouvons construire, incrémentalement, des systèmes plus performants si le besoin s'en fait sentir. N'oublions jamais, si nous revenons à l'être humain, que la plupart des solutions que nous apportons à nos problèmes quotidiens sont largement sous-optimales, et que l'optimalité, dans ce cadre-là, n'est qu'une mesure subjective que nous serions bien en peine de calculer scientifiquement tant le nombre de variables à prendre en compte est important.



CHAPITRE VII

UN EXEMPLE DE RESOLUTION COLLECTIVE DE PROBLEMES: LE TRI COLLECTIF

VII.1. Présentation du problème

VII.1.1 Introduction

L'expression "tri collectif", introduite par [Deneubourg et al. 1991], désigne, comme nous l'indiquons déjà au Chapitre IV, le comportement collectif d'agrégation/ségrégation des fourmis envers le couvain. Celles-ci séparent en effet les oeufs, les larves et les cocons, en les agrégeant par type en différents emplacements du nid. Ce comportement correspond parfaitement à un comportement collectif de "résolution de problèmes" tel que nous le décrivons dans la section précédente. Le problème à résoudre, qui consiste à trier des objets de types différents, a un critère d'arrêt qui consiste à obtenir une structure particulière de l'environnement avec des tas homogènes de chaque type d'objet. Il y a enfin une manière relativement simple de distribuer le problème et sa résolution sur un certain nombre d'agents, manière que nous obtenons dans ce cas précis d'une simplification du comportement des fourmis de MANTA. Cet exemple est donc à prendre comme un exemple simple d'introduction aux systèmes de RCP que nous développons dans la suite de ce chapitre. Il nous permettra d'instancier assez facilement les concepts vus au Chapitre précédent.

VII.1.2. Travaux existants

A notre connaissance, le seul travail réalisé sur ce sujet est dû à [Deneubourg et al. 1991], qui proposent un modèle de tri collectif adaptable à des robots. Les principes sur lesquels se base leur système, déjà évoqués dans [Drogoul 1992], sont: (1) un déplacement aléatoire des agents, (2) une capacité de discrimination des objets à trier, (3) une mémoire à court ou moyen terme des dernières cases qu'ils ont parcourues, (4) et la possibilité de ramasser ou de déposer ces objets en fonction de la densité d'objets du même type (dans leur mémoire). Plusieurs de ces agents, placés ensemble dans un environnement où se trouvent disséminés de nombreux objets de deux types sont ainsi capables, au bout d'un certain temps, de former des tas d'objets de même type. Ce tri n'est certes pas optimal, mais la simplicité du système et sa robustesse compensent sa modique efficacité. On voit bien dans ce cas-là que l'activité des agents va avoir pour conséquence de structurer leur environnement, et que la structuration de l'environnement (du fait des différentiels de densité) va jouer un rôle dans la structuration du groupe d'agents (en focalisant un nombre plus important d'agents sur les objets non encore triés), constituant ainsi la mémoire de la résolution. Le système que nous proposons reprend les grandes lignes de celui de [Deneubourg et al. 1991], à savoir des agents simples et initialement redondants, mais nous nous intéresserons de manière plus approfondie aux mécanismes d'organisation des agents, notamment les mécanismes de division du travail qui peuvent intervenir dans une telle application et à leurs conséquences sur la résolution du problème.

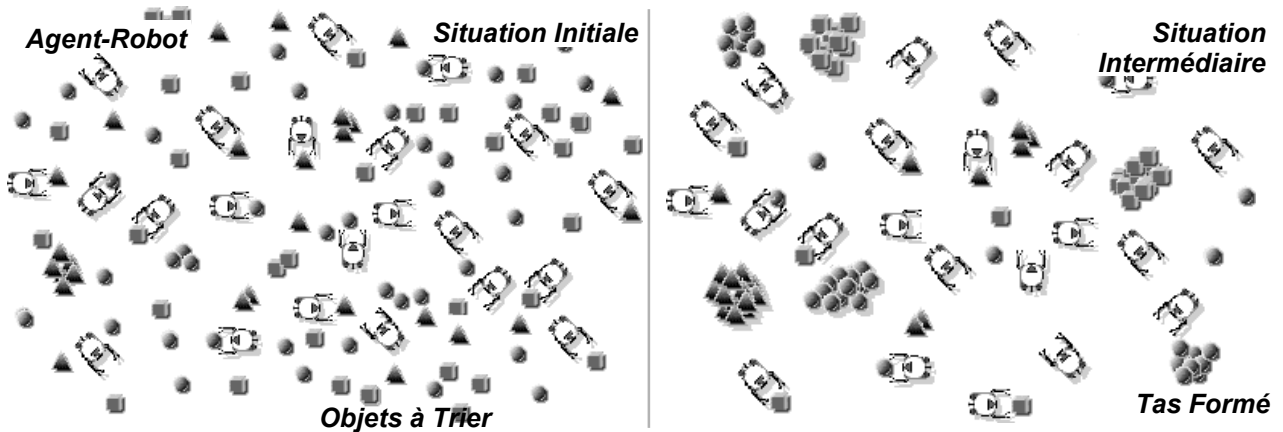


Figure III.1 - Deux exemples de situation de Tri Collectif. A gauche, la situation initiale, où les objets à trier sont tous disséminés dans l'environnement. A droite, une situation intermédiaire où des tas d'objets ont été formés.

VII.2. Programmation des agents

Les agents présents dans cette application sont de quatre types différents. Trois types d'objet à transporter et un type d'agent transporteur. Puisque leurs comportements ne font pas intervenir d'autres compétences que celles des fourmis, ces différents agents vont hériter des classes déjà programmées pour les agents de MANTA, à savoir celles présentées sur la Figure III.2.

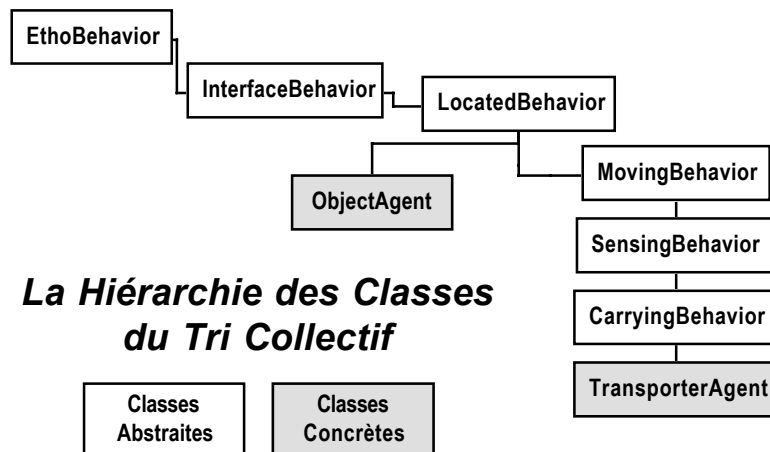


Figure III.2 - La hiérarchie des classes pour l'application du tri collectif. On note que cette hiérarchie est une simplification par rapport à celle de MANTA.

Les agents de la classe `ObjectAgent` ne possèdent qu'un comportement par défaut qui consiste à propager leur stimulus propre (ils sont initialisés avec un nom qui peut être `#a`, `#b` ou `#c`) afin que les agents transporteurs les sentent (ou les voient). Ce stimulus est instance de la classe `InverseAutoCatalyticStimulus` dont le principe de propagation est semblable à sa super-classe `AutoCatalyticStimulus`, mais qui divise son intensité en fonction du nombre d'agents de même type que son propagateur présents sur cette place.

Pourquoi ce choix? Tout simplement pour tenir compte du fait que, l'environnement étant la mémoire de la résolution, les agents doivent avoir tendance à moins se préoccuper des tas déjà formés (stimulus plus faible pour chacun des objets présents), que des objets seuls (stimulus plus fort). Ce type de stimulus joue donc le rôle, décrit au Chapitre précédent, de "température locale", et sert à la fois de déclencheur de comportement, d'indicateur de densité et d'indicateur de structuration de l'environnement en piles d'objets. Les agents transporteurs ne possèdent pas de stimuli propres et définissent seulement trois tâches (plus la tâche $\Diamond_{default}$ qui les fait se déplacer aléatoirement), qui sont décrites sur la Figure III.3. On voit qu'elles sont identiques aux tâches \Diamond_{egg} , \Diamond_{larva} et \Diamond_{cocoon} possédées par les fourmis, ce qui signifie qu'elles sont des comportements organisateurs dont la finalité sera de réduire la température locale qui les a déclenchés. Le poids de chacune de ces tâches est de 4, leur seuil de 10 et leur incrément de poids de 2.

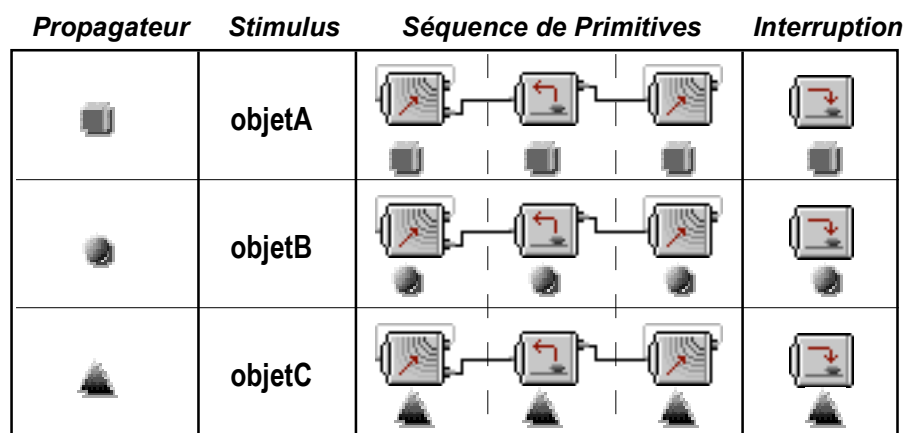


Figure III.3 - Les tâches fournies aux agents transporteurs. Elles sont équivalentes aux tâches \Diamond_{egg} , \Diamond_{larva} et \Diamond_{cocoon} des agents fourmis.

VII.3. Expérimentations

VII.3.1. Structuration de l'environnement

Les expérimentations que nous avons effectuées ont essentiellement eu pour objectif de vérifier que le système fonctionnait et pouvait trier les trois types d'objet en trois tas séparés. Toutes les expériences ont été réalisées avec un effectif de 30 agents trieurs, pour 20 objets de chaque type disséminés aléatoirement au départ dans l'environnement. Pour ce qui est du fonctionnement, les premiers temps du système sont marqués par une assez grande incertitude. Les agents changent fréquemment de tâche, aucun tas ne se forme réellement et le système donne plutôt l'impression de chercher à éparpiller encore un peu plus les objets. Il suffit en fait que deux objets du même type soient agrégés pour que le tri démarre réellement. Ces petits tas devenant moins attractifs, ils laissent les agents s'occuper des objets seuls qui, à leur tour, sont agrégés à d'autres objets. De plus, une focalisation de certains agents sur certains types d'objets apparaît, focalisation due au renforcement des tâches exécutées. Ainsi, dans l'expérience TC_8, après 570 relevés de l'état de l'environnement (échantillonnage à 500 cycles), les agents réussissent à créer deux tas pour chaque type d'objet (Figure III.4).

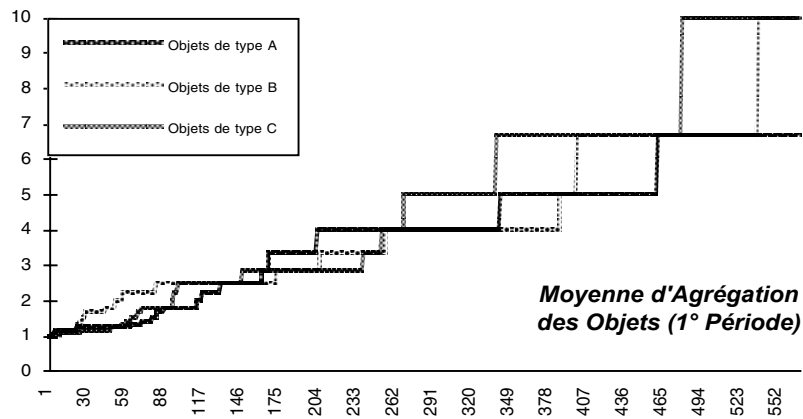


Figure III.4 - Taille moyenne des tas d'objets (en ordonnée) après 570 relevés (en abscisse, où une unité représente 500 cycles). Une taille moyenne de 10 indique la ségrégation en deux tas distincts sans préjuger de leurs tailles respectives. Expérience TC_8 (première période)

VII.3.2. Structuration sociale

Nous avons ainsi la preuve que l'environnement se structure dans la direction voulue, c'est-à-dire vers une diminution du nombre de tas pour chaque type d'objet. Mais qu'en est-il de la structure sociale de la population d'agents ? La structuration de l'environnement a-t-elle une conséquence sur celle de la population ? La réponse à ces deux questions se trouvent représentée sur la Figure III.5. Les courbes qu'elle montre ont été obtenues par relevés sur l'ensemble de la période de tri, où, tous les 10 cycles, était noté le comportement en cours de l'agent. On voit clairement que les profils comportementaux des agents par rapport aux trois tâches à effectuer sont très dissemblables.

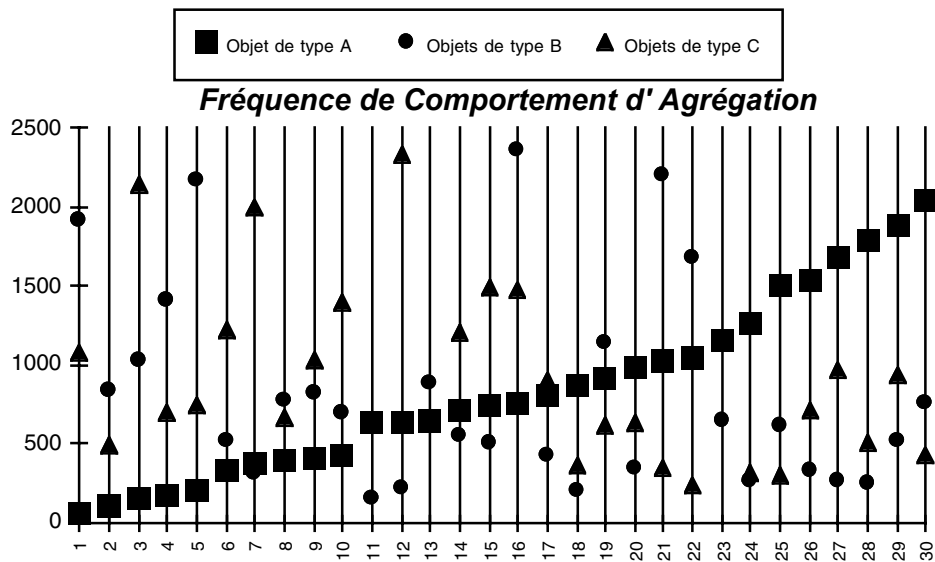


Figure III.5 - Fréquence de ramassage (en ordonnée) d'un type d'objet par un agent (numéro en abscisse). Expérience TC_8.

On perçoit très nettement sur cette courbe que les occupations des agents, premièrement, ne sont pas semblables (par de répartition d'un tiers du temps de travail pour chaque agent sur chaque type d'objet), et que certains individus sont systématiquement (ou presque) pris sur chaque relevé en train d'effectuer la même tâche que précédemment (voir par exemple le profil comportemental de l'agent

n°12 sur la Figure III.5). La spécialisation de chacun des agents (Figure III.6) vient nous confirmer qu'au cours de la résolution s'établit une réactivité différentielle des agents par rapport aux trois tâches qu'ils possèdent, et que cette réactivité, même si elle ne suit pas exactement leurs profils comportementaux, en donne une bonne approximation. En somme, nous obtenons le même résultat que pour les fourmis.

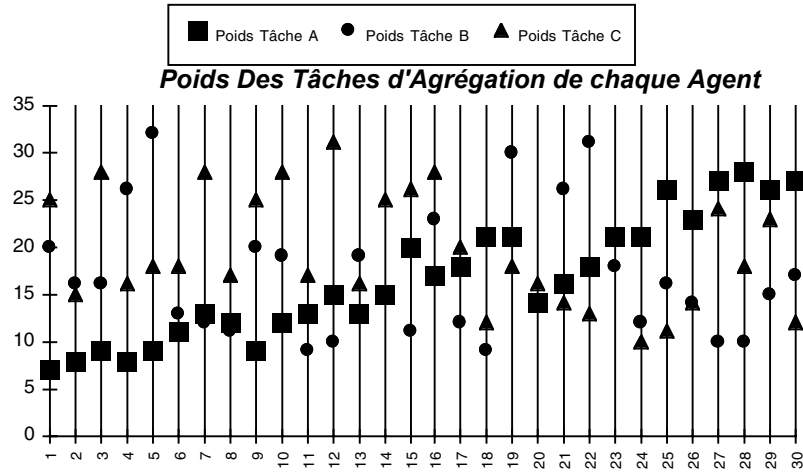


Figure III.6 - Répartition du poids des trois tâches d'agrégation (en ordonnée) pour chaque agent (numéro en abscisse)

VII.3.3. Apport de la structure sociale à la résolution

Cette spécialisation progressive de chaque agent, donc cette structuration progressive de l'espace social, où chaque individu se voit peu à peu assigner un rôle par l'activité totale du groupe, bénéficie-t-elle à la résolution du tri, c'est-à-dire à la structuration de l'environnement ? Si nous reprenons les agents à l'endroit où nous les avons laissés, avec deux tas de chaque type d'objet, et que nous supprimons ces tas pour les remplacer par le même nombre d'objets mais de nouveau disséminés dans l'environnement (c'est-à-dire, si nous initions de nouveau un tri avec les individus utilisés dans le tri précédent), voici les résultats que nous obtenons (Figure III.7):

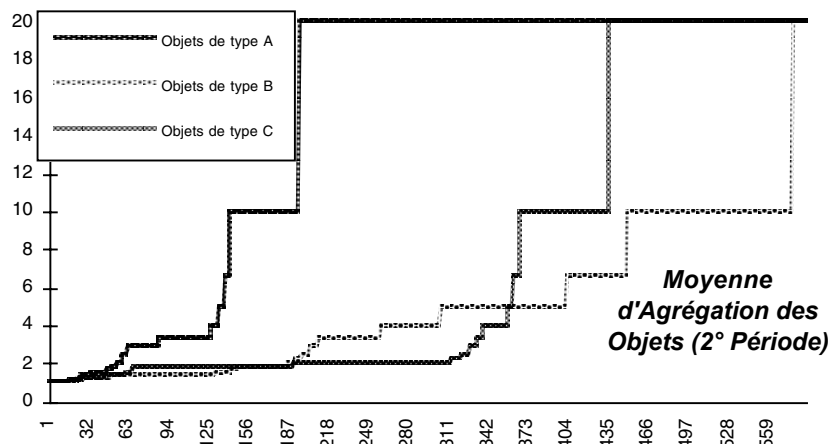


Figure III.7 - Effectifs moyens des tas d'objets (en ordonnée) après 570 relevés de la résolution (en abscisse, où une unité représente 500 cycles). Expérience TC_8 (seconde période)

Nous voyons clairement que le tri effectué converge beaucoup plus vite que le tri réalisé précédemment. A temps de résolution égal, les agents arrivent dans le second cas à créer un et un seul tas par type d'objet. La structuration sociale pré-établie a donc opéré à la manière d'un catalyseur.

VII.4. Conclusion

Ce premier exemple de résolution collective de problèmes nous a permis de voir que des agents simples pouvaient prendre en charge un problème assez complexe et mener sa résolution à terme. Chacun des agents est doté d'un ensemble de tâches qui lui permettent, de son point de vue local, de réduire l'"entropie locale" du monde qui l'entoure (en réduisant la force des stimuli des différents objets), et donc de favoriser la structuration de l'environnement. L'addition, au niveau global, de tous ces comportements locaux finit par générer une configuration particulière du système population/environnement qui prend la forme d'un équilibre dynamique représentant la solution du problème, équilibre caractérisé par l'entropie la plus faible que l'on puisse obtenir.

Dans un système tel que celui-ci, il n'y pas de comportement émergent. La fonctionnalité désirée peut tout à fait être obtenue avec un seul agent et, d'une certaine manière, c'est ce qui fait la robustesse du système. Cette fonctionnalité est par contre *amplifiée* (voir Section VIII.8.2.1) par une structure sociale qui, elle, est émergente. Elle consiste essentiellement dans cette application en une répartition dynamique des tâches entre agents, répartition qui se génère de manière décentralisée (c'est-à-dire que personne ne décide de répartir les agents en groupes fonctionnels), et qui apparaît fortement contingentée par la structuration opérée au sein de l'environnement par les agents. Un agent agrégeant deux objets de même type va en effet agir sur l'environnement en faisant décroître le stimulus associé à ces objets, et du même coup agir sur les autres agents. Il s'approprie (très partiellement) la tâche de cette manière, et fait ainsi diminuer la probabilité qu'ont les autres agents de l'effectuer en diminuant la température locale et donc l'entropie perceptible par ces agents (voir l'explication du comportement d'exclusion mutuelle chez les fourmis, Chapitre V).

Pour nous résumer, nous avons donc mis en évidence le double processus de structuration dont nous parlions au Chapitre VI, avec, d'une part, une structuration de l'environnement (en tas et en objets isolés) qui va influencer sur la structuration sociale du groupe d'agents, c'est ce que nous fournissent les résultats de la première période, et d'autre part cette même structuration sociale qui va catalyser la structuration de l'environnement - c'est ce que nous fournissent les résultats de la seconde période.

Comme nous employons, et que nous continuerons à employer abondamment dans cette section le mot de structure sociale pour des phénomènes comme la division du travail qui ne sont des structures appréhensibles qu'après un travail d'analyse comme celui effectué par les éthologues sur les fourmis, le second exemple que nous proposons va nous permettre de générer des structures sociales immédiatement appréhensibles, parce que reflétées dans l'environnement.



CHAPITRE VIII

LES ROBOTS FOURRAGEURS

VIII.1. Introduction

Dans le domaine de la résolution collective de problèmes, et ce depuis environ 1989, date de la parution de [Steels 1989], une large part est réservée aux applications mettant en jeu des robots explorateurs d'on ne sait quelle contrée lointaine, qui sont souvent appelés, par analogie une fois de plus avec le comportement de fourragement des fourmis, *robots fourrageurs*. L'application type, que nous utiliserons également dans cette section, consiste à se servir d'une équipe de petits robots autonomes pour récolter des échantillons de minerai (ou de n'importe quoi d'autre) dans un environnement inconnu, incertain, et éventuellement hostile afin de les ramener au sein d'une base centrale (où, suppose-t-on, ils seront analysés en vue d'une exploitation future). La définition de cet exemple fait toujours intervenir des robots qui agissent de manière indépendante et si possible simple. Il a pour but de mettre en valeur la capacité d'une population composée de créatures peu intelligentes à résoudre un problème double, qui est à la fois celui de la collecte d'échantillons *per se* et celui d'une recherche efficace dans l'environnement, donc d'une structuration de l'environnement en termes d'espaces explorés et non-explorés. Puisque cette application semble constituer l'exemple de référence pour quiconque veut concevoir des robots ou des agents autonomes, et qu'il a le mérite de renouveler agréablement le problème des proies-prédateurs [Bouron 1992; Korf 1991; Stephens et Merx 1990], nous avons décidé de la reprogrammer, afin d'évaluer tout d'abord la pertinence de nos postulats, et deuxièmement l'efficacité collective de certains comportements. La teneur des travaux [Arkin et Hobbs 1993; Goss et Deneubourg 1992; Steels 1989] consacrés aux systèmes multi-robots a essentiellement mis en valeur les capacités auto-organisationnelles que l'on pouvait découvrir avec des agents simples, ainsi que l'intérêt à utiliser des concepts venant d'autres sciences, comme l'éthologie, pour obtenir des comportements artificiels intéressants. Mais, curieusement, aucun d'entre eux n'a abordé le problème de l'efficacité de ces systèmes, privilégiant plutôt l'étude de leur robustesse ou leur adaptabilité. Par conséquent, même si l'idée générale est intéressante et certainement pleine d'avenir dans des domaines applicatifs variés [Flynn et Brooks 1988], il est toujours difficile de faire le bon choix parmi toutes les solutions proposées dès que l'on souhaite programmer une application ou construire un système à base de fonctionnalités amplifiées comme celui-ci. C'est pourquoi nous nous sommes intéressés à comparer certains de ces systèmes du point de vue de leur efficacité. Le but est de permettre, grâce aux résultats de cette comparaison, d'éliminer certaines solutions, manifestement trop inefficaces, et d'en privilégier d'autres, dont nous espérons que l'étude sera poursuivie.

Nous nous servirons de cet exemple pour également mettre en valeur les capacités incrémentales du langage à base de tâches défini par EMF et pour aborder la structuration du groupe d'agents sous l'angle de la coopération et de la coordination, en mettant en avant l'importance des comportements individuels et le rôle amplificateur du groupe dans l'influence de ces comportements sur la structuration de l'environnement et donc sur l'efficacité subséquente des agents.

VIII.2. Présentation du problème

La première simulation de "collecte d'échantillons distribuée" par un groupe de robots a été décrite dans [Steels 1989], comme l'illustration du concept de fonctionnalité émergente. L'idée de base, que nous conserverons tout au long de cette section, est de faire collecter des échantillons de minerai par un ensemble de robots dans un environnement inconnu afin qu'ils les ramènent à une base. Toutes les expériences que nous décrivons dans les prochaines sections se dérouleront donc dans un monde (voir Figure III.8) constitué d'une population de robots, d'une base centrale qui émet un signal d'intensité décroissante avec la distance et de trois tas de cent échantillons chacun. La distance entre les tas et la base a été fixée à quarante cases et la vitesse des robots est de 1 case/cycle. Les robots peuvent détecter les échantillons à une courte distance de 2 cases, ce qui correspond à l'aire de propagation de leurs gradients. Le gradient créé par le signal de la base définit quant à lui une aire de 80 cases de diamètre (de manière à ce qu'il soit perceptible par les robots se trouvant près des tas de minerai).

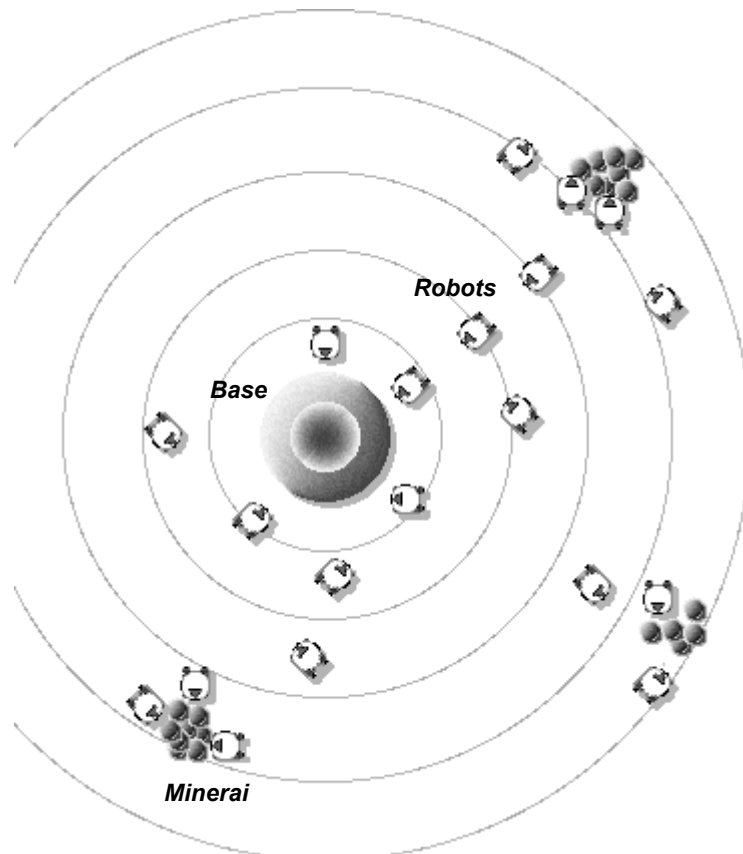


Figure III.8 - Une situation type de collecte d'échantillons par des robots.

VIII.3. Programmation des premiers robots

Les types d'agents présents dans l'application sont au nombre de trois. Des agents minerais, un agent base, et des agents robots. Dans les différentes instances du problème que nous présenterons par la suite, les agents minerais et base resteront les mêmes.

Un agent minerais est semblable aux agents objets de l'application précédente, à savoir qu'il ne peut que propager son stimulus et qu'il n'a pas la possibilité de se déplacer. Il sera donc sous-classe immédiate de `LocatedBehavior`. Il n'est pas non plus prévu que les agents base se déplacent (du moins dans l'immédiat), mais leurs capacités sont un peu plus importantes que celle des agents minerais, à savoir qu'ils possèdent deux stimuli, respectivement nommés `#base` et `#arrivée` et une tâche déclenchée par le stimulus de minerais. Le premier des stimuli sert à générer le gradient qui permettra ensuite aux robots de revenir, le second, qui n'est pas propagé, à indiquer l'emplacement exact de la base (il est "posé" sur sa case). La tâche déclenchée par le stimulus de minerais consiste simplement à ramasser le minerais (dont on suppose qu'il a été placé là par les robots). La classe des agents base sera donc sous-classe de `CarryingBehavior`.

Quant aux robots, ils doivent pouvoir se déplacer, suivre des gradients et ramasser le minerais, ce qui nous donne la hiérarchie suivante (Figure III.9):

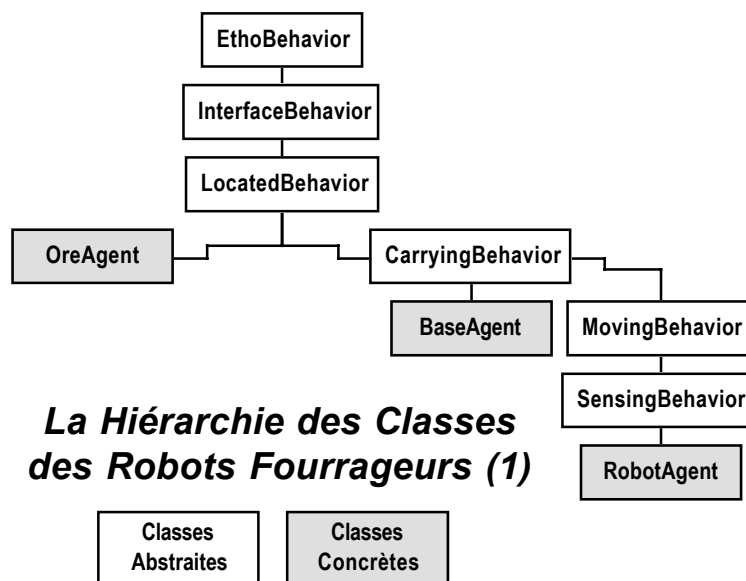


Figure III.9 - Hiérarchie des classes de l'application des robots fourrageurs. `OreAgent` représente les agents minerais.

Comme nous le disions dans l'introduction, l'exemple des robots fourrageurs nous a permis de tester différents comportements individuels et de comparer leur efficacité une fois fournis aux agents qui forment le groupe de robots. La vision que nous avons de ce test est une vision incrémentale. Nous allons partir du comportement le plus simple possible qui réponde aux exigences de l'application et lui rajouter peu à peu des fonctionnalités afin de créer différentes générations de robots. En cela, nous adoptons une méthode proche de celle adoptée par [Bouron 1992], méthode qui permet de retracer, du point de vue du concepteur, l'historique des modifications et permet ainsi d'expliquer le cheminement intellectuel et logiciel jusqu'à une solution satisfaisante.




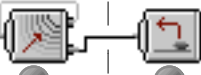


<i>Propagateur Stimulus</i>	<i>Séquences</i>	<i>Remarques</i>
	arrivée 	Stimulus présent sur l'emplacement de la base (non propagé)
	minerai 	
	chargé 	Stimulus interne à seuil (zéro ou n)

Figure III.10 - Tableau des tâches de la première classe de robots.

La première classe de robots, `RobotAgent`, possède trois tâches (plus la tâche `Default`, qui le fait se déplacer aléatoirement), respectivement déclenchées par la présence de minerai, par le stimulus interne `#chargé` et par le stimulus `#arrivée` fourni par la base. Son comportement est très simple (voir Figure III.11); détecter un tas de minerai consiste à se déplacer vers lui et à ramasser un échantillon. Ceci place à une certaine valeur le stimulus `#chargé`, qui va déclencher la tâche de retour vers la base (en remontant le gradient créé par le stimulus `#base`). Une fois arrivé à la base, le stimulus `#arrivée` déclenchera la tâche de dépôt du minerai (après quoi, la base se débrouillera pour le ramasser). Il ne restera plus alors au robot qu'à reprendre une recherche aléatoire.

Comportement du Robot Basique

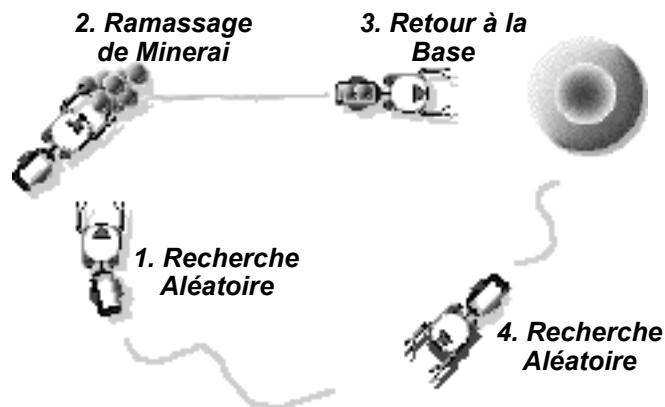


Figure III.11 - Les quatre phases de comportement de la première classe de robots.

Ces robots un peu stupides ont été introduits car ils vont servir, tout au long de cette section, de base commune à toutes les générations de robots suivantes. En effet, et malgré leur incompétence criante, ils exhibent une autonomie et des comportements largement suffisants pour mener à bien la tâche globale de fourragement. Nous verrons d'ailleurs dans les sections suivantes que des améliorations ou transformations mineures au niveau des individus permettront d'obtenir des changements importants à

l'échelle de la population, et de parvenir, puisque c'est là notre but, à l'émergence de processus d'organisation et de coopération entre ces robots qui les rendent plus efficaces collectivement.

Pourquoi ces robots sont-ils totalement incompetents ? Tout simplement parce qu'ils ne disposent d'aucun mécanisme leur permettant, en se servant par exemple de l'environnement comme mémoire, de retourner sur le lieu de leur dernière récupération de minerai. Ils sont donc inefficaces non seulement individuellement, mais aussi collectivement, car il n'y a aucune circulation d'information sur la présence de minerai entre les membres d'un groupe. Et qui dit absence de partage d'information dit absence de coopération. Pour les améliorer, nous devons donc trouver un mécanisme simple de maintien et de partage d'information qui, puisque les robots ne possèdent pas de mémoire, utilise l'environnement comme mémoire.

VIII.4. La première génération de Petit Poucet

VIII.4.1. Présentation des robots

La première génération de robots capables de structurer leur environnement en liaison avec leur histoire et d'exhiber une forme, certes primitive, mais réelle, de partage d'informations s'appelle **Petit Poucet 1**. La conception de ces robots puise son inspiration, nous y revenons toujours (mais n'est-ce pas le propre de notre démarche ?), dans le comportement de fourragement de certaines espèces de fourmis⁷⁰ qui construisent des pistes de phéromones entre la colonie et les sources de nourriture (voir à ce sujet [Chauvin 1982; Deneubourg et al. 1986; Hölldobler et Wilson 1990]).



Figure III.12 - Ouvrière déposant des phéromones de recrutement par l'intermédiaire de sa glande rectale. La piste ainsi créée, d'une source de nourriture au nid, attirera ses congénères.

En effet, sachant que les robots sont capables de rentrer à leur base en ligne droite en suivant le signal qu'elle émet, la manière la plus évidente, compte tenu de leurs capacités limitées, de tracer des chemins depuis la base vers les tas de minerai est de les créer, non pas quand ils partent fourrager (ce qui résulterait en un ensemble de chemins assez complexes ne menant d'ailleurs pas forcément au minerai), mais quand ils reviennent après avoir trouvé du minerai. De cette manière, le chemin créé a de grandes chances d'être rectiligne⁷¹, donc plutôt optimal, et les robots sont assurés de trouver du minerai à l'une de ses deux extrémités. Il faut alors que nous munissions nos robots précédents de la capacité à déposer des "miettes", des "balises", des "marqueurs" ou n'importe quoi d'autre, quand ils reviennent à la base, et de la capacité à suivre un chemin déjà tracé (soit par eux-mêmes, soit par un autre individu), ce qui leur permettra de localiser beaucoup plus rapidement les gisements de minerai. Un point intéressant par rapport à nos prescriptions de résolution collective de problèmes peut être fait

⁷⁰ Quant à leur nom, il fait bien entendu référence au jeune héros de Charles Perrault, dont le comportement de dépôt de miettes ressemble à s'y méprendre à celui de certaines espèces de fourmis...

⁷¹ Il peut en effet ne pas l'être si un robot rentrant à la base doit faire des détours pour éviter d'autres robots, qui eux vont en sens inverse. Toutefois, il sera toujours plus court qu'un chemin aléatoire.

ici. La structuration de l'environnement à laquelle nous souhaitons arriver ici consiste en la disparition du minerai qui s'y trouve. L'entropie maximale du problème correspond donc à la situation initiale, et devient nulle quand tout stimulus de minerai (qui représente la température locale perceptible par les agents) a disparu de l'environnement⁷². Cependant, déposer des miettes dans l'environnement va accroître l'entropie du système, en élevant la température locale le long de la piste. Nous serions donc en droit de nous demander si cette solution est intéressante. Mais en fait, l'accroissement de l'entropie globale du problème par la création des pistes est contrebalancée par le fait qu'elle apporte aux agents une *information organisatrice* (au sens de [Lapierre 1992]), information qui aura tendance à faire baisser cette entropie par l'efficacité accrue qu'elle leur procure.

Il convient donc de bien séparer les deux niveaux, celui de l'observateur et celui de l'agent. Pour l'observateur, l'accroissement d'entropie équivaut à une perte d'information au sujet du système (voir [Atlan 1979]), alors que pour l'agent il représente un gain d'information lui permettant d'orienter son comportement. Comme nous le verrons par la suite, tout le problème est de trouver un bon compromis entre le gain d'information locale et le gain de structuration globale.

VIII.4.2. Programmation du système

La métamorphose de nos précédents robots en robots **Petit Poucet** passe par l'ajout d'une nouvelle classe `CrumbAgent`⁷³ dans l'application, classe qui va représenter les marqueurs (ils sont symbolisés sur la Figure III.14 par un petit drapeau). Cet objet sera sous-classe directe de `LocatedBehavior`, comme le minerai, puisqu'il n'aura pas d'autre capacité que de déposer un stimulus `#piste` à l'endroit où il aura été placé. Quant à la classe des nouveaux robots, elle va être sous-classe de `RobotAgent`, et s'appellera `TomThumbAgent`.

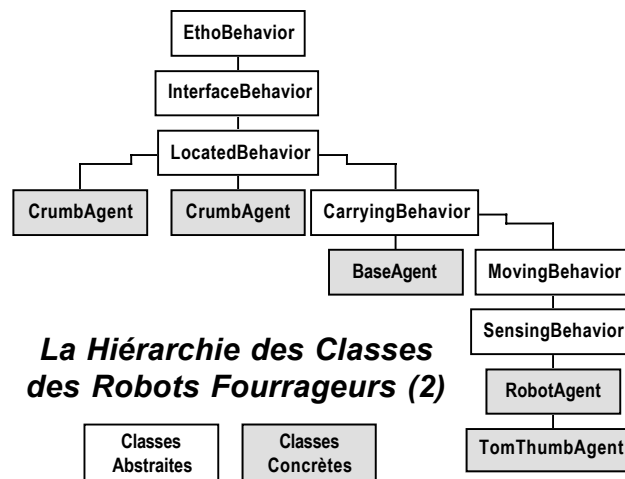


Figure III.13 - Hiérarchie des classes de l'application des robots fourrageurs incluant les robots **Petit Poucet**. `CrumbAgent` représente les agents matérialisant les chemins.

La programmation de la classe `TomThumbAgent` consiste en l'ajout d'une nouvelle tâche et en la modification d'une tâche existante. La nouvelle tâche correspond au comportement déclenché par le stimulus `#piste`. Comme indiqué sur la Figure III.14, elle oblige le robot à suivre ce stimulus tout en fuyant le stimulus `#base`. De cette manière, le robot cherchera une autre miette autour de son

⁷² Une conséquence de ceci est que, en l'absence de tout contrôle, les robots ne sauront jamais que le problème est résolu.

⁷³ *crumb* signifie miette en anglais.

emplacement, dans la direction opposée à celle de la base, c'est-à-dire dans la direction du minerai. Avec ce comportement, on voit donc qu'il n'est nul besoin d'avoir de mécanismes au niveau des miettes afin d'indiquer la direction à suivre. Quant à la modification, elle s'effectue sur la tâche déclenchée par le stimulus $\#$ chargé. Au lieu de simplement remonter le gradient émis par la base, le robot dépose en plus, sur chaque case rencontrée, un nouvel agent miette. En termes de programmation, ces tâches font intervenir une particularité que nous n'avons fait qu'effleurer lors de la présentation du noyau, qui est l'écriture de boucle... sans boucle. La tâche déclenchée par les miettes, par exemple, ne stipule qu'un seul déplacement alors que l'on s'attendrait plutôt à une encapsulation de cette primitive dans un opérateur de boucle (comme dans la tâche \diamond chargé des précédents robots). En fait, il s'agit d'une manière alternative de programmer les boucles, manière qui consiste à faire en sorte que ce soit l'environnement qui maintienne la boucle, et non le comportement lui-même. Son avantage est qu'elle permet de faire effectuer plusieurs primitives à l'intérieur de la boucle (voir par exemple la tâche \diamond chargé de la Figure III.14), ce que ne permettent pas les opérateurs.









	<i>Stimulus</i>	<i>Séquences</i>	<i>Remarques</i>
	arrivée		
	minerai		
	chargé		Pas de boucle. Tâche déclenchée tant que le robot est chargé.
	piste		Même remarque, tant qu'il y a des miettes.

Figure III.14 - Les tâches fournies aux robots *Petit Poucet 1*. Les briques entourées d'un filet gras correspondent aux ajouts effectués par rapport aux robots précédents.

Le comportement global de ces nouveaux robots est indiqué Figure III.15. Après avoir trouvé du minerai, le robot revient à la base en déposant sur son chemin un certain nombre de miettes, qui constitueront ainsi une piste. Lors de son voyage retour, il n'aura plus qu'à suivre cette piste pour retourner à l'emplacement du tas de minerai. De la même manière, tout robot peut accidentellement croiser une de ces pistes, et donc ainsi minimiser son temps de recherche en allant directement à l'emplacement indiqué.

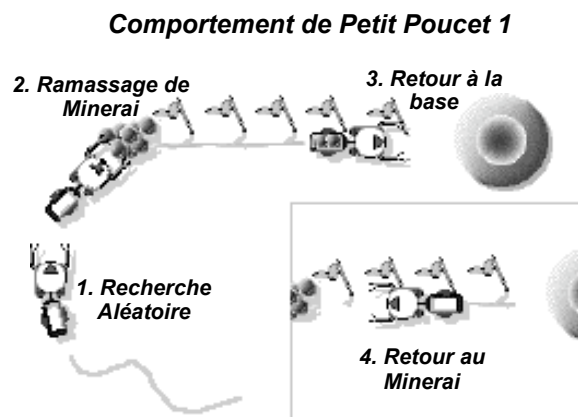


Figure III.15 - Comportements des robots *Petit Poucet 1*. La cadre grisé autour du dernier comportement indique une séquence temporelle différente.

VIII.4.3. Expérimentations et résultats expérimentaux

Afin de juger de l'efficacité de ces robots, nous avons mené un certain nombre d'expérimentations dans l'environnement décrit Section VIII.2. Ces expérimentations consistent à analyser, dans des populations dont l'effectif varie de 1 à 100, le temps mis par les robots à rapporter à la base les 300 échantillons de minerai. Pourquoi faire ainsi varier la taille de la population ? Essentiellement parce qu'elle constitue un paramètre important des phénomènes collectifs émergents (voir [Glance et Huberman 1992]). Pour les sociétés de fourmis comme pour nos robots, nous pensons en effet que ce paramètre possède deux seuils. Le premier, le seuil d'émergence, permet l'apparition de phénomènes de groupes et de processus auto catalytiques. En-deçà, nous avons une collection d'individus, au-delà une densité suffisante pour l'apparition de structures collectives. Le second, le seuil plafond, constitue le moment où la densité devient trop importante pour que la population puisse continuer à s'autoréguler. Des phénomènes locaux interviennent qui perturbent le fonctionnement global du système, et le font diverger de son objectif. Chez les insectes sociaux, l'apparition de ce seuil est souvent évité par la fondation de nouvelles colonies, dont il est en partie responsable. Mais il peut-être également résolu par une plus grande intégration des individus (l'évolution nous donne ainsi l'exemple de sociétés de plusieurs centaines de millions de fourmis, cf. [Hölldobler et Wilson 1990]).

Les expérimentations que nous avons donc menées sur cette population de robots, ainsi que celles que nous mènerons de manière identique sur les robots suivants, consistent à tester 100 populations de robots (dont le numéro d'ordre correspond à l'effectif dix fois de suite. Pour chacune des expériences nous calculons le temps nécessaire à la population pour accomplir la tâche requise. Nous rassemblons ensuite ces données sur une courbe identique à celle de la Figure III.16, où l'axe des abscisses indique le numéro d'ordre de la population et l'axe des ordonnées le temps nécessaire. Nous avons ainsi la possibilité de saisir l'évolution de l'efficacité par rapport au nombre de robots, ainsi que sa dynamique. Comme il est possible de le voir, la courbe dénotant le temps requis décroît très fortement pour les premières populations, ce qui signifie que leur efficacité augmente en conséquence, jusqu'à un minimum de 1113 cycles (pour les groupes dont l'effectif est de 64 robots), puis se remet à croître assez irrégulièrement jusqu'à un maximum, pour 99 robots, de 3500 cycles.

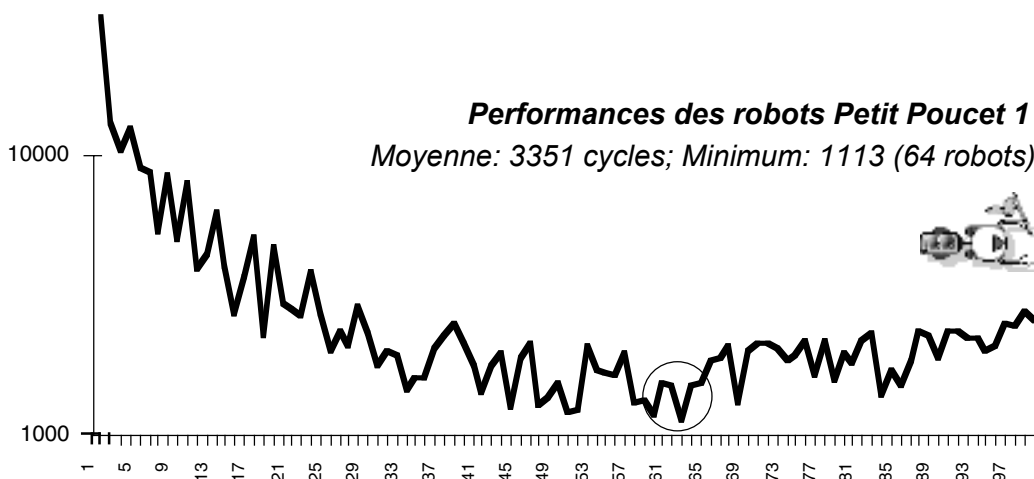


Figure III.16 - Evolution des performances de 100 populations de robots Petit Poucet 1. En abscisse sont indiqués les effectifs de chaque population et en ordonnée le temps mis à ramener tout le minerai à la base (en nombre de cycles). Echelle logarithmique.

VIII.4.4. *Interprétation des résultats*

Comment interpréter ces résultats ? Tout d'abord, l'amélioration de l'efficacité pour les populations variant de 1 à 63 robots apparaît de manière assez claire et le "seuil d'émergence" à partir duquel les résultats des populations se situent dans une fourchette plus étroite se situe aux environs de 33 robots. Cette amélioration peut s'expliquer par deux facteurs indépendants qui sont, premièrement, l'augmentation du nombre de robots, et donc l'augmentation de la probabilité globale qu'a la population de trouver plus rapidement les tas de minerai et, deuxièmement, le mécanisme de création de pistes qui permet au groupe de se focaliser sur le minerai dès lors qu'il a été découvert. A ce titre, nous pouvons donc estimer qu'il joue le rôle d'un catalyseur, et que le partage d'informations qu'il génère a l'effet d'une rétroaction positive sur le comportement global de fourragement⁷⁴.

Cependant, comment, dans ce cadre, expliquer les variations assez brutales observées entre deux populations proches ? Il suffit de se représenter ce qui se déroule lorsque des gisements de minerai sont épuisés. La rétroaction positive continue en effet à focaliser l'attention des robots sur les pistes qui y mènent, mais ces pistes ont perdu la signification qu'elles possédaient. Une information est donc toujours transmise, mais cette information est maintenant vide de sens, et rien dans le comportement des robots ne permet de l'invalider. Les variations entre des populations pourtant quasiment équivalentes peuvent donc être expliquées par les choix aléatoires effectués par les robots quand ils ressortent de la base. Quand beaucoup de robots choisissent un chemin menant à un tas vide, l'efficacité globale de la population s'en ressent, d'autant plus qu'ils ont toutes les chances, non pas de revenir à la base, mais de reprendre un comportement de recherche aléatoire une fois arrivés au bout de la piste.

Afin de réduire ces distorsions, qui sont gênantes car il n'est pas possible de prédire l'efficacité d'une population d'après celles qui la précèdent, il faut donc envisager de rendre dynamique l'information qui, jusqu'à présent, n'était que statique, et donc contrebalancer la rétroaction positive globale par une rétroaction négative qui tiendrait compte de la disponibilité du minerai en bout de piste.

VIII.5. **La seconde génération de Petit Poucet**

VIII.5.1. *Présentation*

De nombreuses possibilités existent pour implémenter une telle rétroaction. Les insectes sociaux nous en donnent une, dans la mesure où les pistes chimiques qu'ils créent sont volatiles et disparaissent donc au bout d'un certain temps, quand la piste n'est plus fréquentée par les individus revenant au nid (ce qui signifie qu'il n'y a plus rien au bout). Mais utiliser cette solution impliquerait de rajouter un paramètre supplémentaire à l'application, la persistance des miettes jouant en effet un rôle certain dans l'accomplissement de la tâche globale, et nous préférons nous concentrer sur les deux paramètres que sont le comportement local et l'effectif du groupe.

Il serait également possible de rajouter un comportement aux robots qui, lorsqu'ils atteignent le bout de la piste et que le gisement est épuisé, les fasse revenir en enlevant toutes les miettes. Mais la

⁷⁴ Ce mécanisme est même d'une finesse assez étonnante. Admettons que nos robots transportent des matières volatiles, qui s'évaporent assez rapidement. Si un gisement est découvert mais que le temps mis par le robot pour rentrer à la base à partir de ce gisement est supérieur au temps d'évaporation du produit transporté, il y aura création et partage d'une information partielle, c'est-à-dire d'une piste qui sera tronquée à l'emplacement où la tâche chargée aura été interrompue par la disparition de la charge. Le chemin sera donc indiqué, mais pas à partir de la base.

question serait alors de savoir comment détecter une telle situation, notamment dans le cas d'agents réactifs qui peuvent réagir à la *présence* d'un déclencheur mais difficilement à son *absence* (et encore moins à la conjonction de la présence et de l'absence simultanées de deux déclencheurs)⁷⁵.

La solution que nous avons adoptée n'est donc pas unique, mais elle respecte, à notre sens, le caractère réactif des robots et le fonctionnement totalement décentralisé du groupe qu'ils forment (personne ne "prend la décision" de supprimer toute marque de l'environnement).

Nous nous sommes encore une fois inspirés du conte de Charles Perrault. Qu'advient-il au chemin de miettes que forme le petit poucet? Il est picoré par les oiseaux. La rétroaction positive que crée le Petit Poucet pour rentrer chez lui est donc contrebalancée par la destruction de la piste, non pas par ses parents s'ils avaient découverts le stratagème (ce serait une rétroaction négative "absolue" comme celle préconisée un peu plus haut), mais petit à petit, en cours de route, par les oiseaux. Ce qui nous intéresse ici, c'est que les deux rétroactions, à la fois la création d'information et la destruction de cette information, se déroulent dans le même temps. Si nous revenons à nos robots, nous avons donc tout intérêt à leur donner à la fois le rôle du Petit Poucet et des oiseaux.

VIII.5.2. Programmation du système

Un robot seul va ainsi créer une piste en revenant à la base chargé de minerai (je crée de l'information qui me servira par la suite) et détruire la piste qu'il suit en revenant vers le minerai (je consomme l'information qui ne me servira peut-être plus). Pour ce faire, il suffit simplement de leur rajouter une brique dans la tâche \diamond_{piste} , brique qui leur fera ramasser l'agent miette présent sur leur case avant d'aller en chercher un autre (voir Figure III.17)




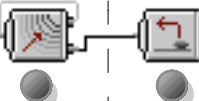

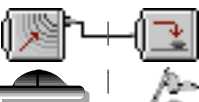

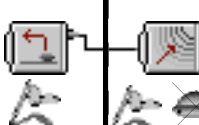
Propagateur Stimulus		Séquences	Remarques
	arrivée		
	minerai		
	chargé		Pas de boucle. Tâche déclenchée tant que le robot est chargé.
	piste		Même remarque, tant qu'il y a des miettes.

Figure III.17 - Tâches fournies aux robots Petit Poucet 2.

Le comportement d'un robot seul est indiqué Figure III.18. Au niveau collectif, Les robots vont donc avoir la possibilité de détruire les pistes créées par d'autres robots, mais ces pistes seront réétablies dès

⁷⁵ Qui implique, d'une certaine manière, une représentation plus fine que celle que nous avons voulu donner à nos agents. Une telle solution pourrait être envisageable dans le cadre des actions situées où la représentation d'une telle situation pourrait se faire sans problème. Mais en avons nous vraiment besoin ?

que l'un d'eux aura trouvé du minerai. Quand, par contre, une piste ne mène plus qu'à un gisement vide, elle ne sera pas réétablie, puisque les robots ne déclencheront pas la tâche \diamond chargé.

Comportement de Petit Poucet 2

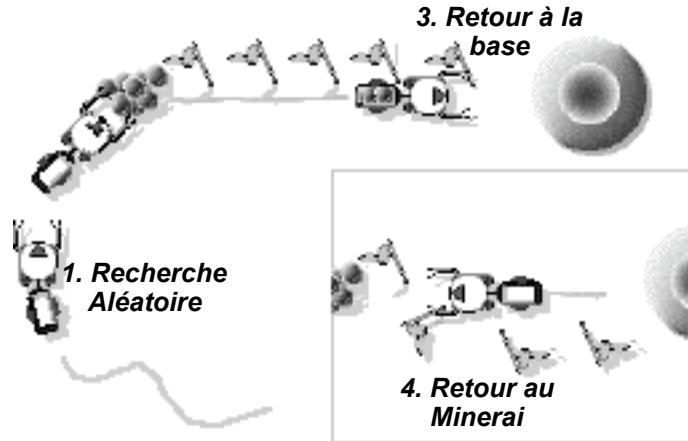


Figure III.18 - Le comportement individuel des robots de la seconde génération de Petit Poucet. Le filet gris indique une séquence temporelle différente.

VIII.5.3. Expérimentations et résultats expérimentaux

La même série d'expérimentations que pour les robots précédents nous fournit les résultats décrits Figure III.19. La ligne grisée représente les résultats précédents, et la ligne en trait plein les nouveaux.

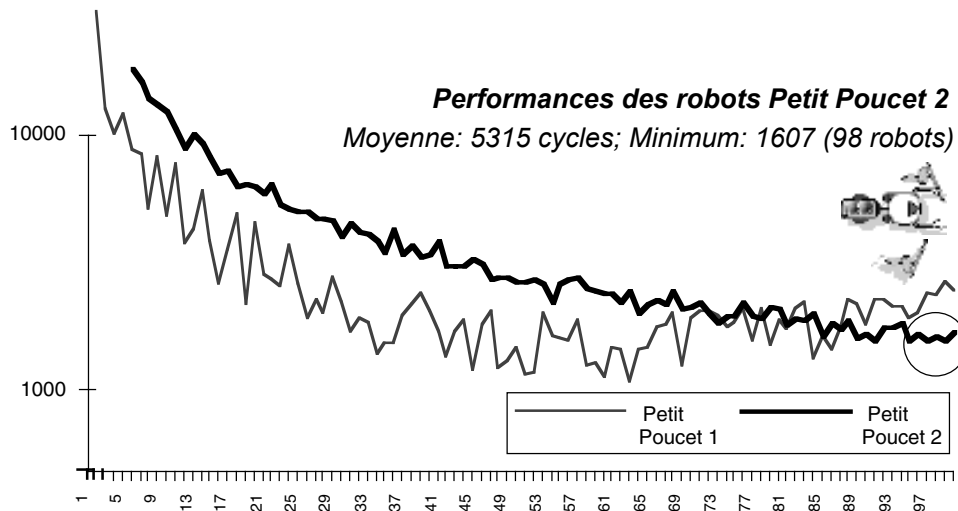


Figure III.19 - Evolution des performances des robots Petit Poucet 2, comparée à celle de la précédente génération.

La première constatation que nous pouvons faire est que ce nouveau comportement local a comme prévu supprimé les variations brutales d'efficacité globale entre populations, et que la performance d'une population donnée peut être approximativement déduite de la performance de celle qui précède par la formule empirique suivante:

$$E(x) = E(x-1) * (1 - \log(2x) / 2x)$$

où x représente la taille de la population, et $E(x)$ son efficacité. Les deux courbes sont représentées sur la Figure III.20. On est en droit de se demander à quoi sert une telle information. En fait, elle est essentielle pour deux raisons. La première est que nous sommes ainsi assurés de la robustesse du système, et notamment du fait que la dégradation de ses performances en cas de retrait de certains robots n'est pas catastrophique. La seconde est que, si une telle application doit réellement voir le jour sous la forme de vrais robots, le maître d'oeuvre n'aura pas à se soucier de mettre 10, 11 ou 12 robots, et ceci à la différence des robots précédents, puisque leurs performances seront voisines.

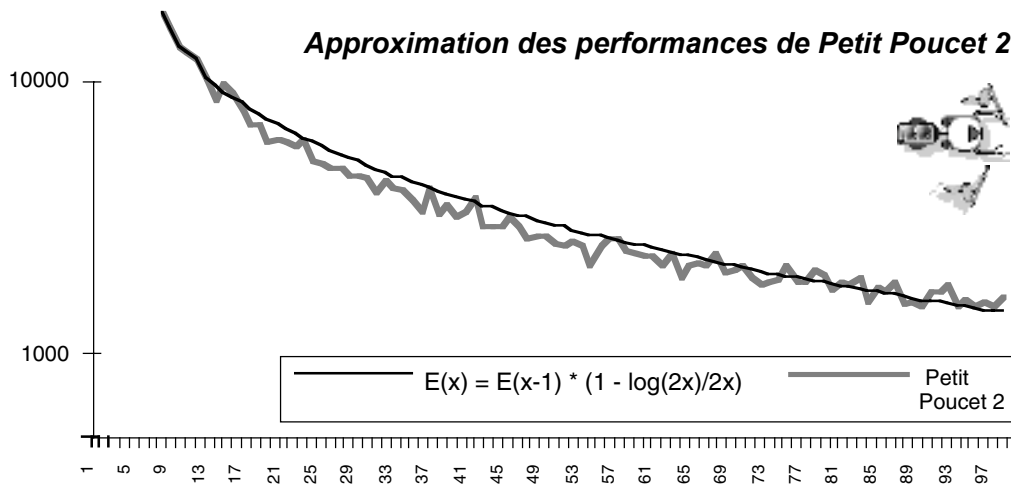


Figure III.20 - L'approximation par une suite de l'efficacité des populations de robots Petit Poucet 2. Ceci permet de déduire l'efficacité d'une population de l'efficacité de la population précédente.

VIII.5.4. Interprétation des résultats

Néanmoins, si l'on compare les performances de nouveaux robots avec celles des précédents, nous voyons que nous avons perdu en efficacité ce que nous avons gagné en prédiction. Pour des populations de tailles réduites (moins de 60 robots), le nouveau système est clairement moins performant que l'ancien. Pourquoi en est-il ainsi ? En fait, il semble bien que nous ayons introduit une trop grande dynamique dans le partage de l'information. Les robots créent et consomment cette information au même rythme, ce qui signifie qu'elle ne se propage pas assez dans la population pour créer un effet amplificateur. Ainsi, si un robot établit un chemin en revenant à la base, ce chemin aura une forte probabilité d'être partiellement détruit par un autre robot qui l'aura rencontré en se déplaçant aléatoirement. Il n'y a dans ce cas-là aucun mécanisme d'amplification ou de propagation de l'information dans la population mais simplement un mécanisme de transfert de cette information, d'un robot à un autre. D'un côté, donc, les robots ne sont plus attirés par des chemins inutiles, mais de l'autre, ils perdent trop souvent des chemins déjà établis. Il nous faut donc limiter cette perte d'information en implémentant de nouveaux comportements locaux, ce qui nous amène à la troisième génération de Petit Poucet.

VIII.6. La troisième génération de Petit Poucet

VIII.6.1. Présentation et Programmation

Il s'agit ici de trouver un comportement permettant d'allier une dynamique de mémoire et une dynamique d'oubli, tout en privilégiant la rétroaction positive entretenue par la mémoire des chemins menant au minerai. En fait, c'est simplement une affaire de paramétrage et la solution, déjà proposée,

mais de manière abrupte, par [Steels 1989], est de faire déposer *deux* miettes par un robot rentrant à la base, et d'en faire ramasser *une* par les robots suivant le chemin. Il y a donc possibilité de recrutement, puisque maintenant les robots recevant l'information ne la consomment pas entièrement et en laissent une partie qui reste disponible pour les autres robots, à commencer par celui qui a établi le chemin.

Propagateur Stimulus		Séquences		
	arrivée			
	minerai			
	chargé			
	piste			

Figure III.21 - Tâches fournies aux robots Petit Poucet 3. Le seul ajout (entouré d'un filet gras) concerne la tâche \diamond chargé.

Comme on le voit sur la Figure III.21, la définition de ce comportement ne nécessite que l'ajout d'une brique dans la tâche \diamond chargé des robots. Le comportement qui en résulte est décrit Figure III.22.

Comportement de Petit Poucet 3

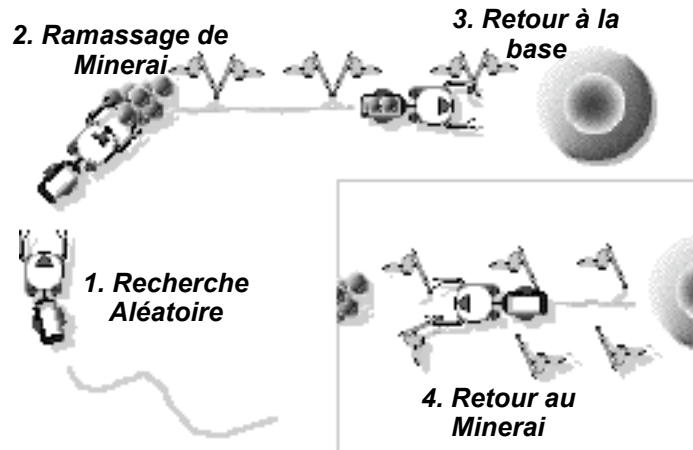


Figure III.22 - Le comportement individuel des robots de la troisième génération de Petit Poucet. Le filet grisé indique une séquence temporelle différente.

VIII.6.2. Expérimentations et résultats expérimentaux

Sur la base des mêmes expérimentations, les résultats obtenus, illustrés par la Figure III.23, montrent clairement deux choses. Premièrement, l'efficacité de ce type de robots est la plupart du temps supérieure, ou tout au moins égale, à celle des précédents. Nous avons donc globalement gagné en efficacité.

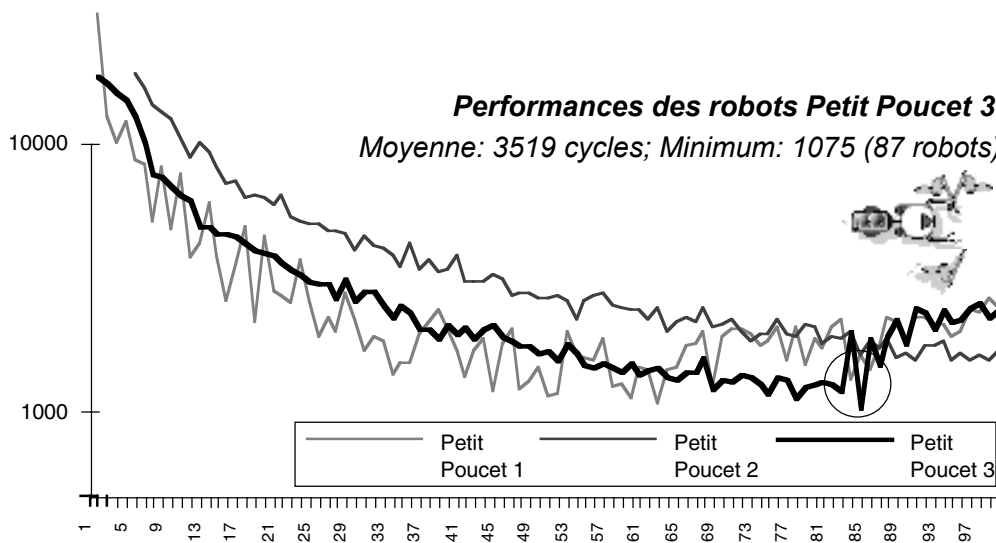


Figure III.23 - Evolution des performances des robots Petit Poucet 3, comparée à celles des deux précédentes générations.

Deuxièmement, la courbe décroît de manière très régulière tant que la population n'excède pas 85 robots, ce qui signifie que nous n'avons rien perdu en prévisibilité par rapport à la deuxième génération. Dans la plupart des cas, donc, cette solution apparaît comme un bon compromis entre les deux précédents **Petits Poucets**, en combinant efficacité et régularité.

VIII.6.3. Interprétation des résultats

Mais si cette solution apparaît comme la meilleure, en permettant un partage d'informations souple qui assure une coopération globale inégalée par les autres robots, comment se fait-il que nous observions un brusque accroissement du nombre de cycles nécessaires pour les 15 dernières populations ? Rien dans leur comportement ne semble en effet les prédisposer à de tels écarts.

Pour avoir la réponse à cette question, il est nécessaire d'*observer* le système en train de fonctionner, afin de se rendre compte qu'elle est en fait d'une simplicité enfantine. Regardons la Figure III.24, qui présente quelques captures d'écran successives d'une même simulation. Que voyons nous dans ces quatre situations ? Des pistes assez larges (ce ne sont plus des chemins, mais des autoroutes), et surtout une groupement de robots d'une densité incroyable autour de la base, groupement qui se maintient jusqu'à de l'épuisement des sites.

Et la réponse se trouve précisément là. Dans notre quête d'une meilleure structuration de l'environnement par les robots, nous avons oublié une chose essentielle, qui est que cet environnement possède également des contraintes qui vont influencer sur le comportement de ces robots. En l'occurrence, ces contraintes se réduisent à l'impossibilité spatiale pour les robots de se déplacer sur une case déjà occupée par l'un de leurs congénères. Et c'est somme toute une contrainte relativement évidente, du moins dans notre monde physique. Mais cette contrainte va générer des effets absolument inattendus sur le fonctionnement du système. Le premier de ces effets consiste en un éparpillement des pistes. Il est relativement simple à comprendre si l'on imagine une situation où un robot revenant à la base se trouve face à face avec trois robots allant, eux, vers le minerai. Il faudra soit qu'il les évite, et de ce fait, qu'il génère une piste qui aura perdu sa rectitude, soit que eux l'évitent, et ils auront de fortes chances

de perdre la piste. Le deuxième effet est lié, mais provoque plus de dégâts. Il peut en effet aboutir, puisqu'il n'y aucune coordination entre les robots, à de véritables blocages, exactement semblables aux embouteillages que nous connaissons, où un premier groupe de robots tente de remonter vers le nid, se trouve face à d'autres robots qui veulent aller vers le minerai, ce qui les fait tous prendre des directions opposées, et donc gêner d'autres robots qui arrivent à ce moment-là, et ainsi de suite. On obtient le même type de situations, totalement ridicules, que celles présentées Figure III.24.

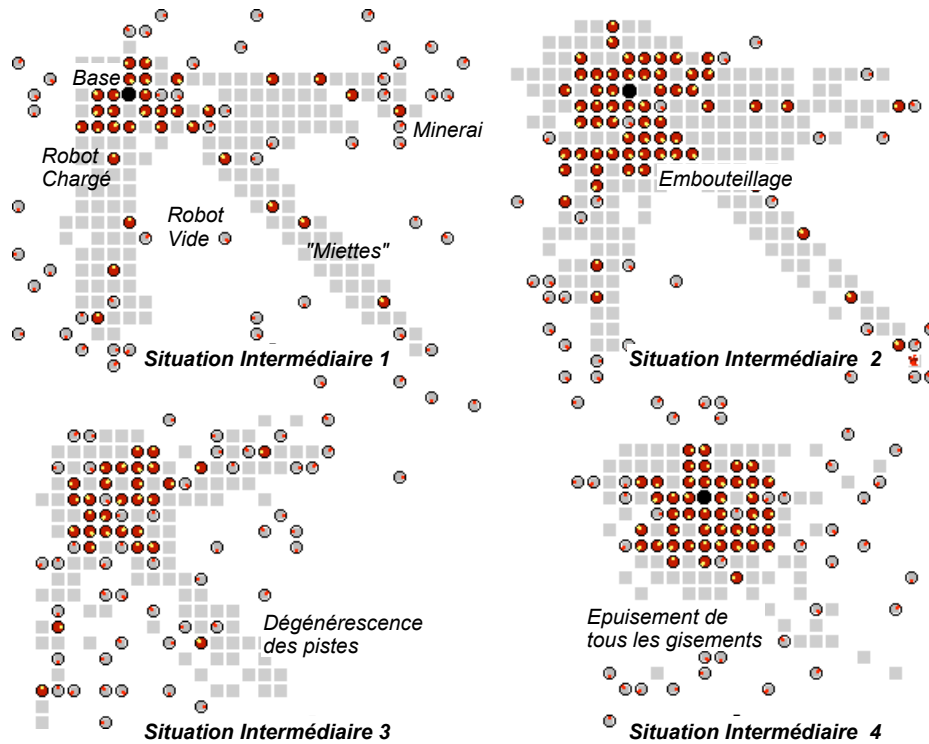


Figure III.24 - Quatre captures d'écran du fourragement effectué par les robots Petit Poucet 3. L'absence de coordination conduit à des embouteillages, et à un élargissement des pistes.

Dans ce cas précis, et donc pour des populations susceptibles d'atteindre une densité importante dans l'espace, cette minuscule contrainte de déplacement provoque une rétroaction négative absolument étonnante sur l'ensemble de la tâche, puisque, si nous regardons de nouveau les résultats Figure III.23, nous nous apercevons qu'une population de 100 robots est manifestement moins efficace qu'une population de 50, voire de 30 robots, ce qui est tout de même un comble pour un système où un nombre élevé d'interactions devraient au minimum garantir une bonne circulation de l'information et donc favoriser les populations d'effectif élevé.

VIII.6.4. Discussion

La littérature consacrée aux systèmes collectifs comme celui que nous présentons ici [Arkin 1988; Arkin et Hobbs 1993; Atlan et Meyer 1991; Delaye et Ferber 1992; Goss et Deneubourg 1992; Kagawa 1992; Waldrop 1990] ne s'est jamais vraiment penchée sur la question de leur efficacité, évitant sans doute, ce qui est compréhensible, de remettre prématurément en cause de manière expérimentale des principes mal définis théoriquement. Or, si l'on souhaite que l'auto-organisation ou l'émergence fasse demain partie du vocabulaire des chercheurs en résolution de problèmes, il est nécessaire d'admettre que tout paradigme, pour aussi intéressant qu'il soit, a toujours ses limites.

Faire une analogie avec le comportement des fourmis pour effectuer des tâches, résoudre des problèmes qui apparaissent comme proches de ceux résolus par les fourmis est louable, et nous ne sommes pas les derniers à emprunter cette approche. Mais cette analogie a ses limites, limites souvent définies par les capacités des fourmis elles-mêmes, qui résident par exemple dans la possibilité qu'elles ont de se moquer des contraintes spatiales qui restreignent nos robots, en grimpant allégrement les unes sur les autres quand l'environnement devient trop encombré.

Elles ne sont donc, par conséquent, absolument pas soumises aux mêmes types de contraintes, et de plus, comme le souligne [Chauvin 1982], certainement peu concernées par des problèmes d'optimalité dans le fourragement (voir également Section VIII.8.), alors que dans une perspective de résolution de problèmes, nous pouvons l'être tout autant que par la robustesse ou l'adaptabilité du système.

Quand nous serons capables de fabriquer des robots aussi adaptatifs individuellement que des fourmis, l'analogie pourra alors fonctionner. Mais, pour l'heure, il nous faut trouver de nouvelles stratégies originales, et, pour une fois, ce n'est pas chez les fourmis que nous les trouverons car elles ne sont pas, loin s'en faut, les dépositaires de toute l'auto-organisation du monde naturel.

VIII.7. Les robots Dockers

VIII.7.1. Présentation

Le problème que nous devons résoudre est donc de continuer à faire coopérer les robots comme précédemment, en évitant une trop forte compétition spatiale. Nous devons donc conserver le partage d'informations qui forme la base de cette coopération, et lui rajouter des mécanismes de coordination d'actions inter-robots qui leur permettent de ne pas se gêner mutuellement. Puisqu'il n'y a qu'une tâche à accomplir (aller chercher du minerai), nous aurons du mal à réaliser cette coordination d'actions par le biais d'une division du travail, comme c'est le cas chez les fourmis ou chez les agents trieurs du système de Tri Collectif. Il nous faut donc trouver autre chose.

Prenons un exemple tiré de la vie de tous les jours (ou presque)⁷⁶. Quand un bateau accoste un quai pour être déchargé, les dockers ne se précipitent pas tous ensemble vers lui, et ne se battent pas pour avoir accès à sa passerelle (comme ce qui est montré sur le schéma A de la Figure III.25). Bien au contraire, ils s'organisent en formant une chaîne depuis le bateau jusqu'aux entrepôts et, de ce fait, minimisent leur perte d'énergie individuelle et maximisent l'efficacité du groupe (schéma B).

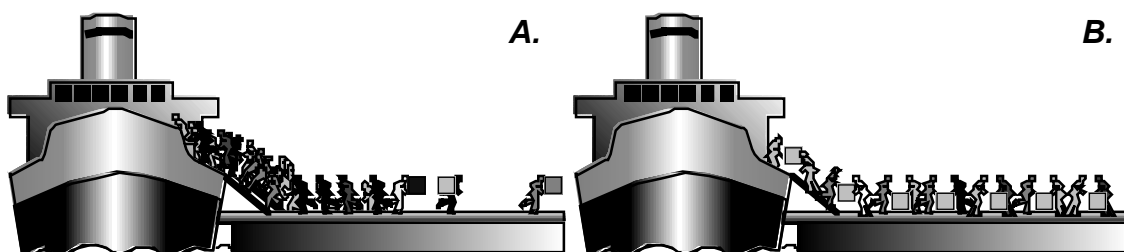


Figure III.25 - Dockers déchargeant un bateau. A. Un déchargement anarchique, sans coordinations interindividuelles; B. Formation d'une chaîne, qui suppose une coordination.

⁷⁶ Exemple fourni par Jacques Ferber, comm. pers.

Pourquoi avoir pris cet exemple ? En fait, la première situation de la Figure III.25 ressemble étrangement aux situations créées par nos robots **Petit Poucet**: un attroupement autour du point d'étranglement de la tâche à accomplir (la base, ou un tas de minerai), chacun des individus le composant essayant de s'en échapper pour aller accomplir son travail. La situation B, où se forme une chaîne, offre par contre un exemple de génération plus ou moins spontanée d'une structure de coopération qui démultiplie l'efficacité des intervenants et leur fait économiser mouvement et énergie. En termes de comportements, les différences entre ces deux situations sont assez simples à percevoir. Dans la première, les dockers agissent comme s'ils étaient seuls, sans prêter le moins du monde attention aux autres agents. Ils vivent chacun dans leur «*petit monde solipsiste*», pour reprendre une expression de Rodney Brooks [Brooks 1991], et poursuivent leur but individuel sans se soucier de savoir s'il interfère avec celui des autres. Dans la seconde situation, la création de la chaîne exige d'eux un minimum de coordination d'actions et nécessite donc de la part des individus de prêter un minimum d'attention à l'activité des autres. Ceci implique de savoir percevoir ce comportement, et de savoir également y réagir en conséquence.

VIII.7.2. *L'intérêt de la chaîne*

Une structure de coopération comme celles présentées dans les deux exemples précédents nous intéresse à plus d'un titre pour notre application. En premier lieu, réside la possibilité, manifeste ici chez les dockers, de la générer de manière totalement décentralisée. Dans l'un et l'autre cas, en effet, elle semble tellement intégrée à la culture, donc au comportement des individus, qu'aucun plan n'est nécessaire pour qu'elle se forme spontanément. D'une certaine manière, la chaîne est déjà virtuellement inscrite à la fois dans l'environnement et dans chacun des individus et ceux-ci ne font que s'y insérer. Qui plus est, l'évolution et la dislocation d'une chaîne semblent plus liées à la permanence et à la disparition des conditions environnementales même qui l'ont nécessitée qu'à un mécanisme global de planification. Ainsi, la chaîne formée par des pompiers perdurera tant qu'il y aura un feu - et tant qu'il y aura de l'eau. De même, tant que des colis resteront à bord du cargo et que le hangar ne sera pas plein, les dockers continueront à maintenir leur chaîne. L'apparition d'une structure de chaîne dans un environnement donné ne semble donc reposer que sur deux éléments essentiels, qui sont, respectivement:

- un lieu-source produisant des objets, ou quoi que ce soit d'autre, qui puissent être consommés en un lieu-arrivée. Ceci constitue une des préconditions nécessaires à l'établissement de la chaîne, et suffit à en assurer la stabilité.
- un ensemble d'entités capables de se positionner entre le lieu source et le lieu d'arrivée et d'interagir deux à deux pour se transmettre ces objets. C'est la deuxième précondition, et certainement la plus délicate.

Ce qui en fait néanmoins tout son intérêt dans une perspective réactive, c'est que les prérequis cognitifs au niveau du comportement des individus sont très peu importants. Il y a certes, notamment dans l'exemple que nous avons fourni, le problème de la reconnaissance initiale par les individus des situations qui semblent nécessiter la formation d'une chaîne. Mais nous pensons que de simples attractions inter-individuelles suffisent largement dans le cas de nos robots. En revanche, une fois la chaîne établie, il semble bien que la structure dans laquelle ils sont engagés transcende les individus qui la composent, qui ne deviennent alors, quelles que soient leurs capacités cognitives par ailleurs, que les vecteurs de l'objet à transporter, un peu à la manière de ce que sont les molécules de métal d'un fil électrique pour les électrons. Par rapport à d'autres structures de coopération, comme la division du travail, la chaîne apparaît de plus comme un processus totalement coopératif, c'est-à-dire qu'elle est

l'expression téléonomique d'un ensemble d'individus dont les buts apparents sont les mêmes. La chaîne prise comme un agent a le même but que les agents qui la composent, mais elle leur permet de l'atteindre de manière plus adaptative, robuste et économique que par eux-mêmes. Pour nos robots qui semblent souffrir d'une trop grande compétitivité spatiale, justement parce qu'ils poursuivent tous le même but et que par conséquent ils utilisent les mêmes ressources, une structure de chaîne semble donc être le mécanisme idéal.

VIII.7.3. Programmation des robots Dockers

La plupart des prérequis à l'apparition de chaînes dans notre application de robots fourrageurs sont déjà en place: des gisements qui produisent du minerai, une base qui le consomme, et des robots capables de se positionner entre les deux, par l'intermédiaire de pistes balisées. Il ne nous reste plus qu'un mécanisme d'interaction et de reconnaissance mutuelles suffisamment bien conçu. Restons dans le paradigme de la chaîne humaine. En vertu du principe de parcimonie énoncé au Chapitre I, nous allons rechercher quels sont les mécanismes cognitifs ou comportementaux de plus bas niveaux permettant d'expliquer la formation d'une chaîne. En choisissant cette option, nous éliminons d'emblée la transmission d'informations symboliques par le langage. Nous éliminons également tout mécanisme hiérarchique intentionnel qui fasse intervenir soit des représentations, soit des engagements (cf. la formation de groupes chez [Bouron 1992]). Nous n'éliminons par contre pas la notion de but, encore que nous la restreignons à un concept qui s'apparente plutôt à un tropisme. Est-on encore capable de former une chaîne avec ce qui reste ? Nous pensons que oui et nous allons le démontrer en programmant nos robots de manière adéquate.

La programmation du comportement des robots va ainsi reposer sur deux principes simples dont nous verrons qu'ils s'intègrent parfaitement au comportement de **Petit Poucet 3**. Le premier principe consiste à leur permettre de détecter les échantillons de minerai transportés par d'autres robots comme si ces échantillons reposaient sur le sol. Un robot transportant du minerai peut, par exemple, allumer une lampe ou émettre un signal acoustique. Ce signal déclenchera chez les autres robots un comportement semblable à celui déclenché par la détection de minerai. De cette manière, les robots deviennent capables de prêter attention aux autres robots s'ils transportent du minerai. Le deuxième principe consiste à les rendre capables de saisir les échantillons transportés par d'autres, toujours comme si ce minerai reposait sur le sol. Il n'y pas lieu ici de disserter sur les difficultés techniques qu'un tel comportement soulèverait pour des robots réels. Nous faisons simplement l'hypothèse que tous sont pourvus d'un bras et d'une benne, et que leur bras est capable de saisir des objets se trouvant dans la benne d'autres robots (voir par exemple Figure III.26).

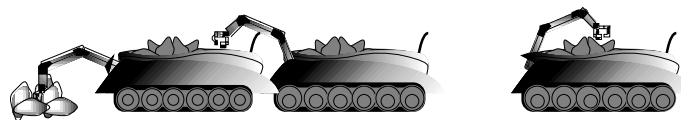


Figure III.26 - Une architecture envisageable pour les robots Dockers

Comment ces deux nouvelles capacités sont-elles traduites dans la liste de tâches des robots précédents? En fait, très simplement (voir Figure III.27). Il y a certes la primitive `primPickUp`: à réécrire, afin de prendre en compte la possibilité de saisir des objets ne se trouvant pas forcément sur la place, mais sur un autre agent. Sinon, les modifications comportementales se réduisent à la propagation, pendant le déroulement de la tâche \diamond `chargé`, du stimulus du minerai. Ainsi, comme on le voit sur la Figure III.27, les robots seront attirés par les robots chargés de la même manière que par le minerai normal.

<i>Propagateur Stimulus</i>		<i>Séquences</i>			
	arrivée				
	mineral				
	chargé				
	piste				

Figure III.27 - La liste des tâches fournies aux robots Dockers. Le stimulus #mineral peut être maintenant émis par deux types d'agents, le mineral et les robots. Cette émission volontaire par les robots correspond au seul ajout effectué sur la tâche \diamond chargé

De cette manière, un robot qui revient à la base chargé de mineral, s'il est détrossé en route, ne déclenchera plus la tâche \diamond chargé et se retrouvera forcément sur une piste (au pire, celle qu'il est en train de construire). Il retournera donc au mineral. Et le robot qui l'a détrossé ira, lui, à la base (puisque'il sera maintenant chargé). Si la piste qu'ils empruntent est la seule existante, ils auront toutes les chances d'interagir de nouveau, puisqu'ils devront se croiser. Et cette interaction peut se répéter successivement entre plusieurs individus qui fréquentent la même piste, jusqu'à ce que, le nombre de robots devenant suffisant, ils n'aient même plus à se déplacer, du mineral vers la dernière interaction, ou de celle-ci vers la base.

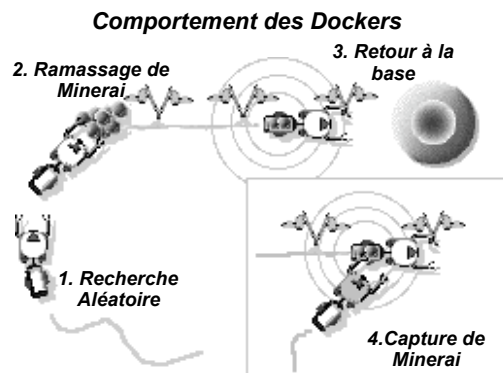


Figure III.28 - Le comportement individuel d'un robot Docker. L'émission du stimulus #mineral est ici représentée par les cercles concentriques.

VIII.7.4. Illustration

L'émergence d'une structure collective prenant la forme d'une chaîne est illustrée sur la Figure III.29, qui contient quatre copies d'écrans d'une même exécution. Il est ainsi possible de voir les trois chaînes qui se construisent depuis les trois tas de mineral jusqu'à la base. On constate également que le gisement pour lequel s'est constitué le plus tôt une chaîne est le premier à être épuisé, ce qui n'est pas très surprenant car la chaîne ainsi créée capture littéralement les robots qui en font partie.

Nous pouvons également voir sur cette figure (Situation Intermédiaire 1 et 2) que des chaînes non complètes peuvent tout à fait fonctionner. La seule différence est que les robots sont à tour de rôle obligés de se déplacer, ce qui en limite les performances.

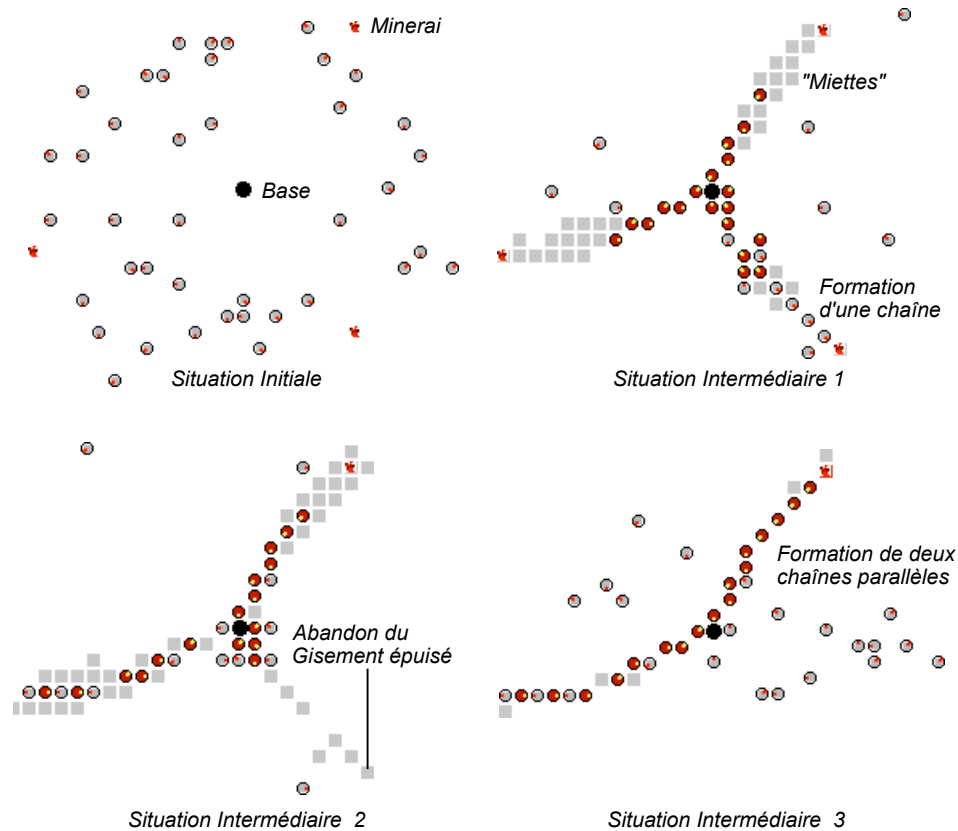


Figure III.29 - Quatre captures d'écran successives du fourragement effectué par les robots dockers. Pour des raisons de lisibilité, les conditions ne sont pas exactement semblables à celles décrites dans les expériences effectuées (3 tas de 50 échantillons chacun, et un environnement réduit à 20 sur 20 cases).

Il est clair que la qualité de la solution (pour ce qui est de l'efficacité, nous l'aborderons avec l'analyse des résultats expérimentaux) est meilleure, dans la mesure où l'éparpillement des pistes est marginal comparé à celui constaté pour les robots **Petit Poucet 3**. Ceci s'explique tout d'abord par le fait que les robots n'ont plus besoin de s'éviter, donc que les chemins sont plus ou moins rectilignes. Ensuite, les pistes ont un taux de fréquentation bien supérieur à celles des précédents robots, ce qui signifie que l'épuisement d'un gisement sera presque immédiatement répercuté dans l'environnement par la disparition de la piste (Situation Intermédiaire 3 et 4).

L'effet second d'une structure de chaîne consiste donc à concentrer l'information aux emplacements où elle est nécessaire, tout en améliorant sa représentativité de la réalité du monde. Nous voyons de plus encore une fois l'efficacité que procure un double processus de structuration sociale et spatiale. La forme de structuration sociale qu'est la chaîne agit en effet sur la structuration de l'espace où la société est inscrite en créant des chemins optimaux, chemins qui vont eux-mêmes agir sur la société en permettant à des robots de s'intercaler dans les chaînes existantes.

De plus, il apparaît (Situation Intermédiaire 4) que seuls les robots essentiels au fonctionnement optimal des chaînes sont conservés, les autres ayant ainsi la possibilité d'explorer les alentours en quête de nouveaux gisements, ce qui s'apparente, bien que rien ne le traduise de manière interne, à une répartition des tâches extrêmement adaptative.

VIII.7.5. Expérimentations et résultats expérimentaux

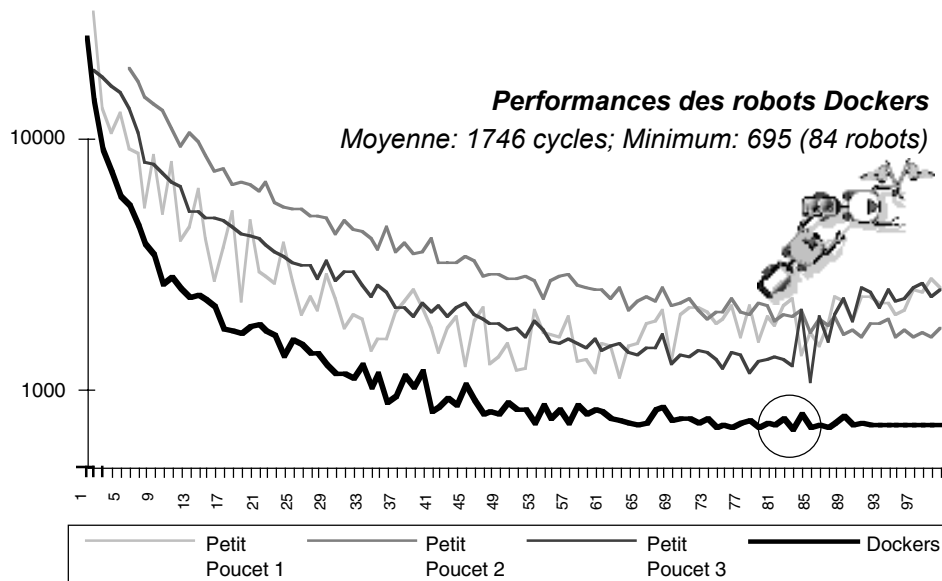


Figure III.30 - Evolution des performances des robots Dockers, comparée à celles des trois précédentes générations.

Le mécanisme de coopération introduit avec cette nouvelle génération de robots, qui combine à la fois le partage d'informations déjà évoqué pour les précédentes générations et une coordination d'actions entre robots, a pour conséquence une amélioration plus que significative de l'efficacité des différentes populations de robots. La courbe obtenue est tout d'abord assez régulière, du moins par rapport aux courbes précédentes, ce qui laisse à penser que nous n'avons rien perdu en prédictibilité depuis les robots **Petit Poucet 3**. La plupart des populations établissent également des performances qui avoisinent le double de celles des précédents robots et, surtout, nous n'observons plus de rétroaction négative pour des densités importante de robots, ce qui montre que notre hypothèse au sujet de la coopération était la bonne, à savoir qu'il nous fallait réduire la compétition spatiale pour faire pleinement jouer son rôle au partage d'informations par les pistes. Enfin, la courbe obtenue semble converger vers la direction asymptotique $\text{Nombre de cycles} = 690$, ce qui a été confirmé par deux expériences respectivement menées avec 150 et 200 robots, dont les résultats sont de 695 et 694 cycles.

VIII.8.1 Analyse des Résultats

Les explications de cet ensemble de résultats tout à fait éloquentes peuvent se faire au niveau local et au niveau global. Au niveau local, c'est-à-dire pour chacun des robots, le fait d'appartenir à une chaîne lui permet de parcourir moins de distance que les robots précédents entre le minerai et la base.

Toutes les générations de **Petit Poucet** devaient en effet effectuer le trajet complet depuis les gisements jusqu'à la base, alors qu'il suffit à nos nouveaux robots de se déplacer entre la position où ils ont trouvé du minerai et la position où ils en sont déchargés. Si cette distance est égale à la précédente dans le cas d'un robot solitaire, elle est rapidement divisée par le nombre de robots qui fréquentent la piste. Au niveau global, ensuite, la réduction du nombre de mouvements de chaque robot et, par suite, sa spécialisation spatiale au sein d'une portion de chaîne diminue fortement la probabilité qu'il vienne déranger les autres, et permet donc d'éviter pour une grande part les problèmes de surdensité autour du nid et des gisements. Et dans le cas extrême où la densité de robots dans l'environnement est

maximale (soit un robot par case), le minerai est convoyé sans aucune difficulté, d'un robot à l'autre, sans que les porteurs aient la nécessité de s'échapper de l'atroupement. Ceci explique bien la direction asymptotique prise par la courbe, qui signifie simplement que la résolution du problème est effectuée de manière semblable par les populations dont l'effectif est suffisant pour former rapidement des chaînes presque complètes. Sont ainsi évitées les brutales chutes de performances des générations précédentes. Qui plus est, la chaîne apparaît comme une structure fondamentalement robuste aux défaillances individuelles qui peuvent survenir, et de ce fait compense largement la plus grande complexité comportementale des robots⁷⁷.

VIII.8.2. *Les fourmis ne créent pas de chaînes*

Nous avons donc montré, en programmant cette application, que nous étions capables, pour résoudre un problème complexe, de générer des organisations d'agents qui s'adaptent à cette complexité par les simples interactions de comportements individuels réactifs, à la fois basés sur les observations éthologiques du comportement de fourrageur des fourmis et sur l'observation de certaines capacités auto-organisatrices de l'être humain (ou, du moins, supposées telles). Exactement comme pour la démarche qualifiée d'éthologie synthétique au Chapitre V, ce système, qui exhibe un comportement collectif assez fascinant, n'est cependant pas neutre vis-à-vis de ces emprunts socio-éthologiques, auxquels il pose des questions relativement intéressantes. Que l'on nous pardonne donc cette digression par rapport à la résolution de problèmes, mais il est un fait absolument étonnant, évoqué dans [Lestel, Grison et Drogoul 1993], qui est que la chaîne en tant que structure de coopération n'est, semble-t-il, observée chez aucune autre espèce animale que l'homme. Ni les insectes sociaux, ni les mammifères supérieurs sociaux ne semblent s'adonner à ce type de pratiques collectives. Et pourtant, il n'est que de voir certaines fourmis véhiculer leur charge sur des centaines de mètres pour se rendre compte de l'apport que constituerait une chaîne dans certains cas. D'où la question: pourquoi n'existe-t-elle pas dans la nature ? Une première raison serait que l'utilisation d'une chaîne n'est vraiment efficace que quand la source à laquelle elle puise les objets à véhiculer est d'une part abondante, et d'autre part relativement fixe. Mais les fourmis champignonnistes du genre *Atta*, par exemple (cf. [Hölldobler et Wilson 1990]), qui transportent des feuilles sur des centaines de mètres à partir d'arbres de leurs environs sont bien dans cette situation et elles ne forment pourtant pas de chaînes.

Une seconde raison, abondamment exploitée dans [Deneubourg et al. 1986; Deneubourg, Pasteels et Verhaeghe 1984], pourrait être qu'un comportement collectif comme le fourrageur a besoin d'un certain nombre d'"erreurs" individuelles pour pouvoir développer de nouvelles pistes. La chaîne apparaîtrait dans ce cas-là comme un processus trop "exact" qui aurait tendance à focaliser les individus sur une source de nourriture au détriment d'autres potentielles. Ceci cadrerait en tout cas parfaitement avec l'hypothèse, peu évoquée dans la littérature de biologie théorique consacrée aux modèles de fourrageur (voir, par exemple, [Krebs 1978]), hypothèse qui tend à prouver, au moins pour les insectes sociaux, que le fourrageur n'est absolument pas un processus optimal (cf. [Chauvin 1989], chapitre I) au sens où nous l'entendons. Les objets véhiculés jusqu'à la fourmillière ne sont pas tous forcément comestibles, et les proies choisies ne sont pas obligatoirement les plus proches, quand bien même elles ont été perçues en premier.

Comme le font remarquer [Chauvin 1982; Hölldobler et Wilson 1990; Passera 1984], les espèces de fourmis pratiquant le fourrageur collectif ont plus tendance à réduire les écarts de densité en nourriture existant au sein de leur environnement qu'à tenter d'épuiser le plus rapidement certaines

⁷⁷ Ceci dans la mesure où leur comportement nécessite un certain degré d'interaction et que les phénomènes d'interaction en environnement ouvert sont encore imparfaitement maîtrisés par les roboticiens.

sources. Il convient donc d'être très prudent dans l'emploi des analogies, surtout si nous souhaitons construire des systèmes dont les objectifs et les contraintes ne sont pas tout à fait semblables à ceux que rencontrent les systèmes dont nous nous inspirons dans la réalité.

Mais au-delà des comportements de fourragement, la génération dynamique de chaînes que nous avons obtenues pose également d'autres questions et peut vraisemblablement apporter de nouvelles hypothèses à la biologie. Elle peut en effet être considérée sous deux angles particuliers. Le premier est celui de la propagation d'informations ou de signaux entre animaux, propagation qui prend parfois des formes difficiles à comprendre comme, par exemple, les vocalisations en cascade chez les batraciens pendant la période de reproduction, pour lesquels, note [Lestel, Grison et Drogoul 1993], «*on évoque (...) un modèle épidémiologique qui n'est pourtant guère satisfaisant, (...) un modèle de chaîne [étant] peut-être plus approprié*». Le deuxième angle sous lequel nous pouvons considérer une chaîne est celui que nous évoquons avec insistance depuis le début de ce chapitre, à savoir la morphogénèse. D'un point de vue global, en effet, la chaîne peut être considérée comme une sorte de membrane, qui structure des entités jusque-là éparpillées par l'intermédiaire d'un "gradient" (équivalent à une différence de potentiel) entre deux lieux de leur environnement. Cette membrane est capable de suivre de manière souple les déplacements de ces lieux, capable de s'interconnecter avec d'autres membranes du même type⁷⁸, et perdure tant que l'information circule. Il serait alors intéressant (toujours selon [Lestel, Grison et Drogoul 1993]) de se demander dans quelle mesure de tels processus peuvent expliquer l'épigenèse des structures en embryologie, c'est-à-dire le processus le plus mal expliqué de l'embryogénèse. Comme le notait déjà Jacques Monod en 1970 [Monod 1970],

la construction d'un tissu ou la différenciation d'un organe, phénomènes macroscopiques, doivent être considérés comme la résultante intégrée d'interactions microscopiques multiples (...). Les embryologistes, pour rendre compte notamment des phénomènes de régénération, ont introduit la notion de « champ morphogénétique » ou de « gradient ». Notion qui paraît au premier abord dépasser de loin celle d'interaction stéréospécifique [des protéines] à l'échelle de quelques Angström. Il reste que cette dernière est la seule à présenter un sens physique précis, et qu'il n'est nullement inconcevable que de telles interactions, multipliées et répétées de proche en proche, ne puissent créer ou définir une organisation à l'échelle millimétrique, ou centimétrique par exemple. (...) Il est assez vraisemblable que la notion d'interactions stéréospécifiques purement statiques s'avérera insuffisante pour l'interprétation du « champ » ou des gradients morphogénétiques. Il faudra l'enrichir d'hypothèses cinétiques. (p. 119 - 120)

Tout ceci n'est bien sûr que prospective, mais laisse entrevoir que de telles applications, totalement artificielles, peuvent questionner la biologie comme l'IA a questionné la psychologie ou le connexionisme la neurobiologie.

VIII.8.3. Une chaîne est une structure de coopération

Si nous revenons à notre problématique réactive, nous pouvons affirmer que nous avons obtenu avec les chaînes une véritable structure coopérative de résolution collective de problèmes, au moins au sens où l'entend [Bouron 1992], c'est-à-dire qu'elle présente aux yeux d'un observateur les six indices nécessaires et suffisants à cette qualité. D'autres critères existent dans la littérature consacrée à la coopération, mais ceux-ci présentent à notre avis les plus grands caractères d'exhaustivité. Nous allons ici les récapituler:

⁷⁸ Nous évoquons ici des résultats empiriques portant sur un très petit nombre d'expériences durant lesquelles les robots pouvaient choisir entre trois bases, qui pouvaient elles-mêmes se déplacer lentement. Les chaînes formées suivaient bien ces déplacements, se montraient capables d'adapter leur taille à la distance séparant la base du minerai et pouvaient s'interconnecter si leurs chemins se croisaient. Mais tout ceci reste à confirmer par de plus amples expérimentations.

- **La coordination d'actions:** nous obtenons bien à la fois une synchronisation et un ajustement spatial des agents, sans laquelle, de toute façon, le transfert de minerai ne serait pas possible.
- **La parallélisation:** Nous observons effectivement un mécanisme de répartition de tâches concurrentes, comme par exemple l'exploitation de plusieurs gisements en même temps, ou une répartition quasi-optimale entre robots explorateurs et robots transporteurs.
- **Le partage des ressources:** il est ici matérialisé par le partage d'informations (via les miettes) au sujet de la localisation du minerai. Les robots ayant chacun les compétences requises pour l'extraction, puis pour le transport du minerai, aucun autre partage de compétences n'est nécessaire.
- **La robustesse:** une chaîne supplée sans aucune difficulté à la défaillance d'un ou plusieurs agents, soit en intégrant d'autres, soit en continuant à fonctionner, mais avec une efficacité moindre.
- **La non-redondance:** la redondance dans le système semble être limitée au facteur minimal nécessaire à la génération de chaînes à maillons identiques. Comme le souligne d'ailleurs Thierry Bouron, cet indice peut sembler contradictoire avec l'indice précédent, et c'est plutôt à un équilibre satisfaisant entre les deux que l'on doit se référer (voir par exemple la solution proposée par [Hölldobler et Wilson 1990] et voir également le Chap. V).
- **La non-persistance des conflits:** les conflits d'ordre spatial deviennent relativement rares avec l'émergence des chaînes et ne durent jamais très longtemps.

L'obtention de cette structure coopérative, qui peut donc être *interprétée* comme une *vraie* structure de coopération (autrement dit, intentionnelle) par un observateur, a pu se réaliser grâce à l'auto-organisation d'un groupe d'agents réactifs. Cet exemple constitue donc en soi un excellent argument quant aux points développés aux Chapitres I et VI, à savoir l'intérêt de voies alternatives pour l'obtention de structures intelligentes dans le cadre de la résolution distribuée de problèmes.

VIII.8.4. Perspectives

Cette application de fourrageur distribué en est bien sûr à ses premiers balbutiements et il est assez difficile de prédire ce qu'il en découlera. Néanmoins, quelques perspectives peuvent être envisagées. Premièrement, étendre l'application par l'adjonction de plusieurs bases mobiles (cf. note 1 du précédent paragraphe). Il est facile de voir l'intérêt d'une telle extension, par exemple pour l'agriculture (avec des bases-tracteurs et des robots-moissonneurs), ou dans le domaine militaire (afin d'assurer les lignes de ravitaillement entre les différents corps en mouvement). Deuxièmement, exploiter les analogies que ce système offre avec d'autres systèmes réactifs, dont essentiellement le système PACO, présenté dans [Demazeau 1991], qui utilise la coordination d'agents réactifs dans une application de vision dédiée à l'extraction de contours et la segmentation d'images en régions. Dans ce système, en effet, les agents forment également des chaînes, mais de manière différente, en suivant les lignes de fractures entre les motifs et/ou les couleurs de l'image. Dans une optique d'extension de notre application multi-agents pour la gestion d'environnements plus complexes⁷⁹, il serait sans doute intéressant d'envisager un regroupement des deux applications. Ceci nous permettrait, entre autres choses, d'échapper au travers habituel des applications de Vie Artificielle, qui semble être de construire des micro-mondes excessivement bien adaptés aux agents qui les peuplent (cf. [Tyrrell et Mayhew 1991]) - et non l'inverse.

⁷⁹ Ou, si l'on préfère, plus réalistes, c'est-à-dire incluant des obstacles, différentes altitudes ou différents types de terrain.

VIII.8. Méthodologie de programmation

Nous souhaitons, dans cette section, approfondir quelques points méthodologiques en rapport avec notre approche de la résolution de problèmes. Nous allons tout d'abord, par l'intermédiaire de la programmation et de l'observation des systèmes présentés au cours de ce chapitre, esquisser quelques lignes de notre démarche de conception du comportement des agents. Nous aborderons ensuite la question de l'émergence de fonctionnalité, où nous établirons une distinction entre fonctionnalités émergentes et amplifiées, et enfin la question de l'émergence de comportements individuels.

VIII.8.1. Programmation par complexité incrémentale

Dans ce type de systèmes dynamiques, de légères modifications au niveau individuel peuvent totalement modifier le comportement global de la population. Si nous comparons les trois générations de robots **Petit Poucet** en termes de capacités comportementales, leurs différences sont infimes. Cependant, les résultats obtenus par leurs populations respectives sont complètement différents en termes d'efficacité. Ceci illustre bien la difficulté de conception des systèmes basés sur l'intelligence en essaim ou de n'importe quel système distribué dans lequel l'"intelligence" est la résultante du comportement collectif d'entités "non-intelligentes". La méthodologie suivie dans cette section fournit un exemple de ce que l'on pourrait appeler "la programmation par complexité incrémentale" (également évoquée par [Beer 1990]), qui est le pendant méthodologique de notre principe de parcimonie. L'idée de base est de commencer par programmer des agents ne possédant aucun comportement social, c'est-à-dire aucune capacité d'interaction ni de communication, et de leur adjoindre progressivement, en fonction de l'analyse des structures (sociales et environnementales) obtenues, de telles capacités. Les premiers robots étudiés fournissent un bon exemple de "première génération": aucune mémorisation, aucune interaction, aucune communication. Au vu de leur comportement, que leur manque-t-il ? Une capacité même limitée de mémorisation, qui leur permettrait de retourner sur un gisement déjà découvert, et une capacité à partager l'information, afin d'inciter d'autres robots à les suivre. Cette génération zéro ne crée donc aucune structure et ne peut espérer résoudre le problème que fortuitement, la plupart du temps après un temps excessivement long. La première génération de **Petit Poucet** pallie donc ces deux défauts en donnant la possibilité aux robots de mémoriser dans l'environnement le chemin parcouru et d'en faire par la même occasion un signal pour les autres. Le problème devient alors que le partage d'informations réalisé n'évolue pas en fonction de la validité des informations. La deuxième génération de **Petit Poucet** a alors la possibilité de manipuler l'information (en la détruisant), mais il apparaît rapidement qu'ils le font de manière un peu trop... dynamique. Une simple modification de paramètres dans la troisième génération permet d'ajuster ce comportement en préservant suffisamment longtemps les pistes déjà créées. Finalement, des capacités d'interactions directes viennent équiper les robots **Dockers** afin de les autoriser à coordonner leurs actions.

Ce type de programmation est largement influencé par les paradigmes que nous employons, et notamment ceux provenant de l'éthologie. Outre sa partie informatique, qui incite déjà à adopter un point de vue particulier (le point de vue de l'agent) sur l'application, notre méthodologie est basée sur un important travail empirique d'expérimentation, d'observation, de description et d'interprétation des systèmes sur lesquels nous travaillons. En somme, ce qui pourrait le mieux sans doute caractériser notre démarche constructiviste, c'est, pour paraphraser le joli mot de [Chauvin 1989]⁸⁰, que lorsque nous

⁸⁰ Par opposition au behavioriste, «[l'éthologue] ne pose pas de problème au rat, il laisse le rat lui en poser», p. 18.

observons le déroulement d'une expérimentation, *nous ne posons pas de problèmes aux agents, mais nous laissons les agents nous en poser.*

VIII.8.2. La question de l'émergence

VIII.8.2.1. FONCTIONNALITES EMERGENTES ET AMPLIFIEES

Le passage du niveau micro au niveau macro (i.e., de l'individu au collectif) s'exprime quelquefois en Intelligence Artificielle Distribuée par l'intermédiaire de la notion de fonctionnalité émergente [Steels 1989] (cf. Chapitre I). Celle-ci fait référence à l'apparition d'une fonctionnalité effectuée par un groupe d'agents qui n'apparaît pas explicitement dans la programmation du comportement des agents, ce qui signifie qu'un observateur, ou que le concepteur, n'est pas en mesure de décrire le *lien causal* entre l'*effet global* observé et les *causes locales* qui sont connues. Une fonctionnalité émergente représente donc, à un moment donné, un saut qualitatif dans l'espace des fonctionnalités possibles du système telles qu'elles sont déductibles de la connaissance des comportements individuels. Or, il nous apparaît que les deux applications présentées dans cette section ne cadrent pas réellement avec cette définition. Si nous admettons, en effet, que le comportement d'un robot est dérivable de la connaissance de chacune des tâches qui le composent (voir par exemple [Drogoul, Ferber et Lalande 1993], où nous décrivons ces comportements sous une forme hiérarchique), la fonctionnalité de la population de robots sera aisément dérivable de ce comportement.

Aussi bien dans le cas du Tri Collectif que dans le cas des robots Petit Poucet, la tâche à effectuer est accomplie de manière similaire par un robot solitaire et par un groupe de robots (on en voudra pour preuve à la fois les expériences réalisées dans le cadre des robots Petit Poucet et les affirmations de [Deneubourg et al. 1991] concernant le Tri Collectif). Ce qui différencie dans les deux cas ces deux situations, c'est l'efficacité du système, et non sa capacité à appréhender la tâche. Nous n'obtenons donc pas de saut qualitatif lors du passage de l'individu au collectif, mais un saut quantitatif caractéristique d'un effet de groupe. Nous pouvons ainsi sans trop nous fourvoyer qualifier ces systèmes de systèmes à *fonctionnalités amplifiées* et non émergentes.

Ce qui constitue à nos yeux la différence essentielle entre ces deux types de fonctionnalités, ou de propriétés, c'est la nécessité dans le cas des secondes d'un seuil minimal d'émergence, c'est-à-dire d'un effectif minimal, ou d'une masse critique, nécessaire à la population d'agents pour atteindre un objectif. On dira dans ce cadre que l'obtention d'une chaîne est une propriété émergente (elle nécessite un certain nombre d'individus), alors que l'obtention de la piste le long de laquelle se forme la chaîne n'est qu'une propriété amplifiée (elle peut être créée par un seul robot). De la même manière, la division du travail entre individus semblables dans le Tri Collectif est une propriété émergente, alors que le tri lui-même est une propriété amplifiée. D'un point de vue incrémental et constructiviste, les fonctionnalités amplifiées ne sont pas inintéressantes à étudier, car, en observant le comportement d'une population d'agents comme les **Petit Poucet**, l'on sait de quoi il s'écarte (le comportement global idéal est normalement identique aux comportements individuels) et l'on sait pourquoi il s'en écarte (à cause d'un trop grand nombre d'agents, puisque le système est censé exhiber le comportement normal avec seulement un agent). On peut ainsi étudier *comment* il s'en écarte, ce qui ne manquera pas de fournir des hypothèses sur les comportements locaux aptes à le faire revenir dans le droit chemin. En somme, c'est en observant le système se désorganiser que l'on peut comprendre comment le faire s'organiser.

VIII.8.2.2. COMPORTEMENT EMERGENT ET CONTROLE EMERGENT

Toujours dans le cadre d'une approche méthodologique de la programmation de systèmes basés sur la Résolution Collective de Problèmes, il peut être intéressant de s'interroger sur la notion d'émergence

de comportements au sein d'un agent. Cette notion fait en effet l'objet d'un certain nombre de débats au sein des différentes communautés qui accueillent ce type de travaux, que ce soit l'Intelligence Artificielle Distribuée (voir par exemple [Wavish et Connah 1990]) ou la Vie Artificielle ([Brooks 1990]). Or, il apparaît, si l'on effectue un rapide tour d'horizon de ce qui est défini comme comportement émergent, qu'il s'agit d'un comportement observable (par une personne extérieure), non programmé dans le comportement de l'agent, et surtout généré par les interactions de deux, voire trois, comportements basiques qui sont, eux, programmés. Par comportements basiques, il faut comprendre un réflexe (du moins, dans tous les travaux portés à notre connaissance), c'est-à-dire l'exécution d'une action (une primitive, dans notre cas) en réponse immédiate et intangible à une perception (à un stimulus)⁸¹. Il convient de noter que nous ne parlons pas ici des systèmes utilisant une ingénierie génétique de sélection et d'évolution des comportements (algorithmes et programmation génétiques, systèmes de classificateurs, etc.), pour qui la notion d'émergence est dans ce cas différente, puisqu'elle se traduit par l'inscription automatique au sein de l'agent d'un nouveau comportement dérivé de l'évaluation des comportements de générations antérieures d'individus (voir par exemple [Delaye et Ferber 1992]). Nous ne citons que les travaux portant sur l'émergence synchronique de comportements au sein d'un même individu, par interaction entre les comportements qu'ils possèdent, ou par interaction entre lui-même et son environnement.

Comme nous le disions déjà au Chapitre IV, notre approche ne privilégie pas l'émergence de comportements individuels dans ce sens, puisque nous prescrivons l'utilisation de tâches qui sont, pour la plupart, loin de s'apparenter à des réflexes. Cependant, nous pouvons observer une autre forme d'émergence, qui ne consiste plus en un comportement émergent, mais en un *contrôle émergent du comportement* (voir discussion du Chapitre III). Dans le cas du Tri Collectif, l'apparition d'une sensibilité différentielle de chaque individu à chaque type d'objet va induire une spécialisation fonctionnelle de cet individu ou, si l'on préfère, une focalisation de son comportement, alors que les agents sont tous semblables au départ. De la même manière, et bien que ceci n'apparaisse pas dans les variables de contrôle que sont les poids des tâches, un robot faisant partie d'une chaîne sera en quelque sorte amputé d'un certain nombre de ses fonctions (par exemple, l'exploration) tant que la chaîne se maintiendra. Dans ces exemples, c'est l'organisation émergente de la population d'agents qui va orienter le comportement de chaque agent en influant sur son contrôle. Notre approche est donc différente de celle basée sur l'émergence de comportements, puisque nous postulons que les robots sont équipés de tous les comportements nécessaires au départ et que nous tentons par la suite d'obtenir au niveau de la société le meilleur ratio entre redondance et spécialisation via l'émergence d'un contrôle collectif qui se traduit en un contrôle individuel de ces comportements.

La "quantité absolue d'émergence individuelle" dans nos systèmes est donc comparativement moindre que dans les systèmes à bases de comportements émergents, mais ceci nous permet sans doute de mieux prescrire les comportements individuels et de mieux observer et comprendre leurs effets au niveau collectif, tout en nous permettant également d'obtenir des agents réactifs pourvus de comportements assez complexes (la complexité s'exprimant ici par la longueur des séquences comportementales), ce qui, pour l'instant, ne semble pas être le cas de l'approche émergente, du fait sans doute de l'extrême difficulté de réification des comportements émergents [Wavish et Connah 1990]. Cependant, l'approche émergente du comportement fait tout de même partie de nos préoccupations dans la mesure où elle peut conduire à une encore plus grande simplification du comportement réellement requis par un agent réactif pour fonctionner de manière satisfaisante. L'implémentation de PENGU présentée dans la

⁸¹ Un bon exemple de comportement émergent est fourni par [Brooks 1987] dans le cas du "suivi de murs", qui résulte de l'interaction de deux réflexes élémentaires d'attraction et de répulsion vis-à-vis des murs.

partie suivante (Chapitre X) nous en fournira une bonne illustration dans le cas de l'adaptation continue d'un agent à un environnement extrêmement dynamique.

VIII.8.2.3. L'EMERGENCE POUR QUOI FAIRE ?

En résumé, et ceci est valable pour les deux sous-sections précédentes, nous ne recherchons pas l'émergence (de fonctionnalités, de comportements) pour l'émergence. Un système d'agents réactifs peut parfaitement fonctionner sans aucune émergence à aucun niveau, ni individuel, ni collectif. Et un agent réactif peut très bien se passer de comportements émergents s'il possède ces comportements sous une forme pré-programmée (avec la réserve émise à la fin de la section précédente).

Quoi qu'il en soit, tout comportement émergent a de grandes chances d'être, un jour ou l'autre, compilé sous une forme définitive, à la condition bien évidemment qu'il soit réifiable et plus facilement manipulable que sous sa forme naturelle. Et il est faux de prétendre qu'un comportement émergent est plus robuste qu'un comportement programmé, puisque celui-là dépend étroitement de la bonne marche de ses différentes composantes comportementales. Si le comportement de répulsion vis-à-vis des murs ne fonctionne plus, le robot décrit par Rodney Brooks dans [Brooks et Connell 1986] ira tout simplement s'écraser contre le mur le plus proche. Un comportement émergent est donc sans doute plus adaptatif qu'un comportement rigide, mais il possède aussi ses faiblesses. De la même manière, la création de la chaîne de robots **Dockers** ne résout pas le problème du fourragement, qui était déjà résolu par les générations antérieures de **Petit Poucet**. En matière de robustesse et de tolérance aux pannes, elle n'accroît pas non plus les capacités du système, celles-ci étant essentiellement basées sur une grande redondance entre les individus. Elle permet d'améliorer la qualité de la solution, tout comme l'organisation sociale de nos agents trieurs. Mais il est fondamental de comprendre qu'une organisation pré-spécifiée (même si elle peut apparaître comme émergente parce qu'elle est distribuée au sein des agents, ce qui serait un non-sens puisque tout programme orienté objet devrait alors être considéré comme un programme à base de fonctionnalités émergentes) peut fonctionner tout aussi bien, dans certaines limites, qu'une organisation dont la structure émerge des interactions de ses composants entre eux et avec leur environnement.

C'est quand ces limites sont dues à un défaut d'adaptation de l'organisation vis-à-vis de nouvelles contraintes (par exemple, la densité de **Petit Poucet**) qu'il convient de s'interroger sur des formes plus adaptatives d'organisation. Et la remarque vaut également pour les comportements émergents.



CONCLUSION DE LA TROISIEME PARTIE

Au terme de cette partie, nous allons discuter de la valeur des deux applications présentées en termes de résolution de problèmes. A première vue, en effet, elles constituent des exemples d'une classe de problèmes pour lesquels la Résolution Collective de Problèmes semble parfaitement convenir. Cette classe de problèmes a, à vrai dire, co-évolué avec les concepts qui ont basé notre démarche, et leur adéquation l'une à l'autre n'est donc pas très étonnante. Toutes deux sont en effet issues de l'observation de comportements collectifs de résolution de problèmes, essentiellement par les insectes sociaux. Nous avons cependant montré que, loin de tomber dans le piège qui consisterait à *programmer des fourmis pour résoudre des problèmes de fourmis*, ce qui serait d'un intérêt certain pour les éthologues, moins sans doute pour les informaticiens, nous pouvons non seulement intégrer des paradigmes venant d'autres exemples de comportement collectif, mais également obtenir des structures dont les potentialités applicatives dépassent largement le cadre du fourragement que nous nous étions fixés au départ. Notre ambition est donc exprimée dans ces deux applications: nous souhaitons nous inspirer d'un paradigme donné, paradigme qui présente une approche de la résolution de problèmes différente de celles habituellement exploitées, et nous envisageons de construire en nous aidant des systèmes qui, aussi bien du point de vue applicatif que du point de vue de l'efficacité, soient sinon plus performants, du moins équivalents aux systèmes provenant d'autres approches. Cependant, si nous les comparons aux problèmes habituellement traités par l'Intelligence Artificielle, distribuée ou non, il est vrai que ces deux problèmes font partie d'une classe de problèmes assez particulière, classe caractérisée par les deux points suivants:

- Les problèmes sont par essence distribués. Il sont caractérisés par la configuration initiale d'un environnement dans lequel évolue un groupe d'agents, et par un point d'arrêt consistant à obtenir une nouvelle configuration qui vérifie certaines propriétés. Dans cette optique, l'organisation du groupe d'agents n'est pas en soi le but, mais constitue le moyen d'obtenir cette configuration.
- De par la dynamique de l'environnement et le fonctionnement semi-aléatoire de certains agents, il n'est pas envisageable d'appliquer au problème des techniques de planification ou de recherche heuristique, l'exploration de l'espace d'états ayant toutes les probabilités de conduire à une explosion combinatoire.

Afin de ne pas nous limiter à cette classe, nous allons donc étendre notre approche à des problèmes étudiés par d'autres techniques, dont la recherche heuristique et la planification. Ceci dans le but de montrer que, loin de s'appliquer exclusivement à des problèmes que l'IA et l'IAD cognitive n'ont pas l'habitude d'appréhender, la résolution collective de problèmes peut également viser la résolution de problèmes pour lesquels un comportement "intelligent" semble nécessaire. Cette extension se fera par l'intermédiaire d'un système appelé Eco-Résolution (dont la présentation se trouve dans [Ferber 1989]), qui possède l'avantage de se prêter plus aisément à une formalisation que notre système d'EthoModélisation. Nous conserverons cependant les points principaux de notre approche et montrerons comment ils s'intègrent dans ce nouveau système.



Quatrième Partie

Intégration des Concepts de Résolution
Collective de Problèmes et d'EcoRésolution

CHAPITRE IX

L'ECO-RESOLUTION

IX.1. Présentation

Dans le domaine de la Résolution Collective de Problèmes, un système occupe à l'heure actuelle une place originale: il s'agit de l'éco-résolution. Ce système, présenté en 1988 par Jacques Ferber [Ferber 1989], n'est pas issu, contrairement aux différents modèles présentés (voir Chapitre VI), d'une analogie avec certains phénomènes naturels d'auto-organisation (encore que son nom puisse le laisser penser), mais d'une approche orientée objets et acteurs de la résolution de problèmes.

Son but est de proposer la même alternative à la résolution classique de problèmes que celle fournie par une approche orientée objets à la programmation classique, c'est-à-dire avant tout une autre manière de penser le problème et sa résolution. Un problème est en effet décomposé en un ensemble d'agents réactifs seulement dotés de deux comportements qualifiés par Jacques Ferber de "tropiques", qui peuvent être interprétés comme des comportements d'attraction et de répulsion à l'égard d'autres agents ou d'autres composants de leur environnement. L'activité d'un agent durant sa courte vie est purement téléonomique et consiste à rechercher un état de "plaisir" vers lequel il sera attiré (et qui correspondra à un but que le concepteur lui a fixé, par exemple) en fuyant les états de "déplaisir" (essentiellement tous les états qui le précèdent, et plus particulièrement ceux où il empêche un autre agent de se faire plaisir), par lesquels il sera repoussé (voir également [Ribeiro, Barthès et Oliveira 1993], qui effectuent une modélisation énergétique de ces comportements, et [Giroux 1993], dont le système SAGE de gestion d'agendas distribuées s'inspire de l'éco-résolution).

Si nous nous reportons au postulat de téléonomie néguentropique décrit au Chapitre VI, nous voyons donc qu'il nous suffit de décrire l'état de plaisir maximal comme étant celui possédant la température locale la plus basse (pour l'agent) et les états de déplaisir comme ceux dotés d'une température plus importante pour retomber sur ce découpage binaire. Et de la même manière que la solution du problème correspondait pour nous à l'obtention d'une structure minimisant l'entropie globale (et donc toutes les températures locales), la solution du problème est atteinte en éco-résolution lorsque chaque agent a atteint son état de plaisir maximal, ce qui doit normalement correspondre à un état stable du problème. La gestion des états métastables ou des minima locaux n'est cependant pas plus prévue que chez nous. De plus, l'éco-résolution s'intègre naturellement à notre démarche de résolution collective de problèmes en prescrivant un modèle très contraint de comportement réactif pour les agents peuplant le problème, ce qui, dans une optique méthodologique, permet de diminuer les choix du concepteur en orientant son travail. Du fait de son comportement hédoniste, un éco-agent n'est en effet mû que par deux influences principales qui sont respectivement sa recherche de satisfaction et deuxièmement la pression que les autres agents font peser sur lui s'il constitue un obstacle à leur satisfaction. Le concepteur devra ainsi se restreindre à la programmation de deux tâches⁸² par agent, respectivement nommées $\diamond_{\text{satisfaction}}$ et \diamond_{flee} , la première décrivant le comportement d'un agent qui cherche à se satisfaire, la seconde le comportement d'un agent qui doit fuir son état actuel

⁸² Bien que le mécanisme de contrôle sous-jacent ne soit plus le même que celui que nous proposons dans le cadre de l'EthoModélisation, nous conservons à ces comportements le nom de tâches.

sous la pression d'autres agents. La section suivante va nous permettre de décrire comment nous avons intégré ce modèle de comportement au sein de notre système d'agents réactifs, intégration qui nous a permis de programmer PENGU, que nous présentons par la suite (Chapitre X). Le système présenté ici n'est pas exactement semblable au système ECOTALK présenté dans [Drogoul 1990], ni au système ECO présenté dans [Bura et al. 1991] et [Ferber et Drogoul 1992]. Les différences principales résident dans l'emploi du mécanisme stimulus/tâche couplé à l'activation par envoi de message, et dans la disparition de la notion de "dépendances" et de but explicite (voir [Bura et al. 1991] pour plus de détails). À bien des égards, cette version représente donc une version édulcorée de l'éco-résolution de base. Mais elle nous permettra d'aborder ce système de résolution de problèmes par la "petite porte", avant d'instancier toutes ses potentialités dans l'exemple du Taquin (Chapitre XI) qui utilisera, lui, ECOTALK.

IX.2. Les éco-étho-agents

IX.2.1. Structure des éco-agents

Le système d'éco-résolution se décompose en un noyau minimal indépendant du domaine d'application, sous la forme de la classe `ECOBehavior`, qui fournit une représentation abstraite des éco-agents (leur structure minimale ainsi que le contrôle de leur comportement), et une partie dépendante de chaque domaine d'application, contenant des classes abstraites et concrètes qui héritent d'`ECOBehavior`. Cette classe est prévue pour être placée n'importe où dans la hiérarchie des classes existantes d'EthoModélisation, afin de tirer parti des primitives de comportements (appelées *actions* dans la terminologie éco) déjà programmées. Elle ne repose donc sur aucune structure particulière, si ce n'est celle d'`EthoBehavior`, et se contente de définir (1) un nouveau mécanisme de contrôle des comportements qui surcharge certaines méthodes définies dans cette dernière, (2) un ensemble de primitives et (3) deux tâches génériques. La différence entre le mécanisme de contrôle du comportement utilisé par les étho-agents et celui utilisé par les éco-agents réside dans les points suivants:

- Un agent possède seulement deux tâches $\Diamond_{\text{satisfaction}}$ et \Diamond_{flee} , qui sont définies de manière générique dans la classe `ECOBehavior` (voir plus bas la définition de ces tâches). La création d'une nouvelle classe d'éco-agent ne nécessite donc pas de créer de nouvelles tâches (à moins qu'elles ne soient nécessaires), mais simplement la redéfinition de certaines des primitives qu'elles utilisent. De plus, les méthodes compilées correspondant à ces deux tâches ne suivent pas les préceptes que nous avons énoncés au Chapitre III, à savoir qu'elles contiennent chacune une partie de code exécutable qui ne s'exprime pas sous la forme de briques, et que la méthode `fleeDo` (implémentant le comportement de la tâche \Diamond_{flee}) peut accepter un argument.
- La tâche $\Diamond_{\text{satisfaction}}$ est semblable à la tâche $\Diamond_{\text{default}}$ des étho-agents, à savoir qu'elle est prévue pour être déclenchée quand aucune autre tâche n'est activable, grâce à la présence constante dans la liste des stimuli de l'agent du stimulus `#satisfaction`.
- La tâche \Diamond_{flee} est un peu différente des tâches que nous avons vu jusqu'à présent puisqu'elle peut être déclenchée de manière normale par la présence du stimulus `#flee` dans l'environnement, mais également de manière totalement préemptive par l'intermédiaire d'une méthode dédiée qui porte le nom de `flee:.` Le mécanisme implémenté dans cette méthode court-circuite une bonne partie du `TaskManager` (voir Chapitre III) en faisant un appel direct au `PrimitiveExecutor` et permet donc l'activation immédiate de la tâche, qui restera prioritaire tant qu'elle n'est pas terminée.

IX.2.2. Comportement des éco-agents

Les deux tâches génériques que définit `EcoBehavior` sont les suivantes (nous ne présentons ici que le code de la méthode d'activation):

```

satisfactionDo
  | aPlace |
  self exec: #primCanBeSatisfied
    ifSucceed: [self exec: #primActionSatisfaction]
    ifFail: [aPlace := self findPlaceForSatisfaction.
      self exec: #primAgress: with: aPlace then:
        [self exec: #primMoveTo: with aPlace]]

```

Si l'agent peut se satisfaire immédiatement (`primCanBeSatisfied`), il fait appel à `primActionSatisfaction` (voir plus bas) et sinon tente d'augmenter sa satisfaction en cherchant un nouvel emplacement (`findPlaceForSatisfaction`). Il agresse alors cet emplacement pour en faire fuir les éventuels occupants et se déplace dessus (`primMoveTo`).

```

fleeDo
  ^self fleeDo: nil

fleeDo: aConstraint
  | aPlace |
  aPlace := self findPlaceForFleeing: aConstraint.
  aPlace isNil ifTrue: [^nil].
  self exec: #primAgress: with: aPlace then:
    [self exec: #primMoveTo: with: aPlace]

```

Du fait du double mécanisme d'activation pouvant déclencher la tâche `flee`, nous avons deux méthodes implémentant son comportement. La méthode `fleeDo` (sans argument), utilisée lorsque la tâche est déclenchée par un stimulus, et la méthode `fleeDo:` (avec argument) appelée directement par la méthode `flee`. L'argument est habituellement une contrainte pouvant être transmise par l'agent souhaitant faire fuir le receveur du message. Le receveur cherche donc un emplacement où fuir, différent de celui passé en contrainte, par l'intermédiaire de la méthode `findPlaceForFleeing`. Si cet emplacement est `nil`, il arrête la tâche (il ne peut pas fuir) et sinon agresse l'emplacement et se déplace dessus.

Les primitives et méthodes dépendantes du domaine qu'il convient de coder pour rendre ces comportements opérants sont donc au nombre de six, ce qui n'est pas très important. Les deux premières, `findPlaceForSatisfaction` et `findPlaceForFleeing`, sont censées retourner un emplacement ou un autre agent sur lequel l'agent pourra se déplacer. `primAgress:` prend un emplacement ou un autre agent en argument et doit normalement le "libérer" afin que l'expéditeur puisse se déplacer dessus. C'est ordinairement dans cette méthode que le message `flee` est envoyé à un agent. `primMoveTo:` qui prend le même argument que la précédente, codera l'action consistant à se déplacer sur l'emplacement libéré. Enfin, `primActionSatisfaction` indique l'action à effectuer quand l'agent s'estime sur le point d'être satisfait, ce qu'il sait par l'intermédiaire de `primCanBeSatisfied`.

L'exemple de `PENGI`, à la section suivante, va nous permettre d'instancier ce système et de mettre en valeur à la fois sa simplicité, et sa puissance pour la conception d'agents réactifs.



CHAPITRE X

PENGI: MODELISATION D'UN ECOSYSTEME ABSTRAIT

X.1. Présentation

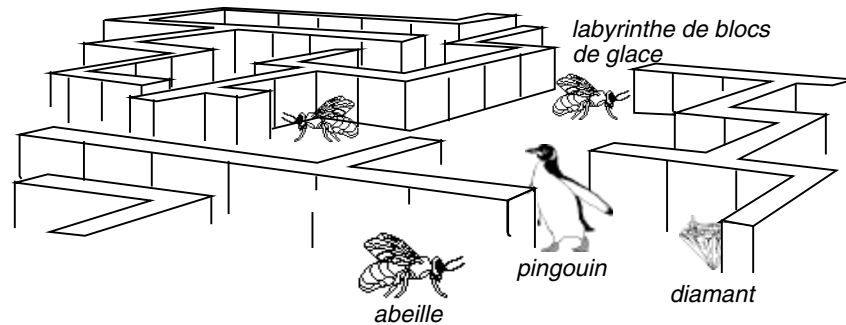


Figure IV.1 - Une représentation en perspective de PENGI, du point de vue du pingouin.

PENGI est dérivé d'un jeu vidéo nommé PENGU, qui a connu un certain succès il y a de cela quelques années. Ce jeu se déroule dans un micro-monde peuplé d'un pingouin, d'abeilles, de diamants et de blocs de glace (voir Figure IV.1). Le pingouin, contrôlé par le joueur, se déplace dans un labyrinthe dont les parois sont faites des blocs de glace qui peuvent être poussés afin de libérer un chemin ou d'écraser les individus qui se trouvent de l'autre côté. Le but du jeu est de faire récolter par le pingouin un certain nombre de diamants, tout en lui évitant d'être piqué par les abeilles, ou écrasé par un bloc de glace poussé par l'une d'entre elles. Celles-ci ont un déplacement semi-aléatoire, ce qui signifie qu'elles suivent le pingouin quand elles en sont assez proches, mais qu'elles peuvent tout à coup décider de l'abandonner, la probabilité pour qu'elles le fassent étant fixée une fois pour toutes au commencement de la partie. Un bloc de glace ne peut être poussé que dans les directions où il n'est pas immédiatement adjacent à un autre bloc. Après avoir été poussé, il glisse dans la direction choisie jusqu'à ce qu'il heurte un autre bloc de glace (ou les bords de l'écran). Tout animal, abeille ou pingouin, qui croise son chemin est irrémédiablement écrasé. La stratégie que doit utiliser le joueur s'appuie donc sur deux points:

- Il doit être capable de faire réagir le pingouin à toute attaque d'abeilles en le faisant fuir ou contre-attaquer s'il en a la possibilité. Il doit ainsi faire attention à ne pas le laisser se faire enfermer par un essaim d'abeilles, dont il a peu de chance de sortir vivant.
- Il doit en même temps veiller à lui fournir un chemin correct dans la labyrinthe de manière à ce qu'il ramasse les diamants le plus vite possible. Dans la version originale du jeu, le temps imparti est en effet strictement limité.

X.2. Le Choix de Pengi

X.2.1. Une application de la notion d'actions situées

Le choix de PENGI comme application n'est pas fortuit. Il a en effet servi en 1987 à Phil Agre et David Chapman [Agre et Chapman 1987] pour présenter une démarche alternative à la planification classique que nous avons déjà évoquée à propos des modèles d'agents (Chapitre III), qui est la notion d'actions

situées. Les arguments mis en avant par ces deux auteurs dans cet article maintenant fameux concernaient essentiellement l'incapacité des planificateurs à appréhender des situations extrêmement dynamiques, et choisir un jeu vidéo (donc une réalisation informatique) permettait de mettre clairement en évidence les lacunes de ces systèmes face à des problèmes qu'un enfant de quatre ans savait résoudre [Pêcheux 1990].

Le postulat de base des actions situées est qu'une action n'est jamais indépendante de son domaine d'exécution, à savoir qu'elle ne peut être séparée de ce qui, à un moment donné, l'a motivé dans l'environnement de l'agent. Toute action est vue comme une réponse (une réaction) à un aspect particulier de l'environnement, ou à une situation, c'est-à-dire une conjonction d'indices environnementaux élémentaires. Conjonction n'est d'ailleurs pas tout à fait le terme exact, puisqu'il ne s'agit pas pour l'agent de répondre à la présence simultanée d'un ensemble d'indices (une abeille, un bloc de glace, un diamant), mais à la présence simultanée d'indices mis en relation les uns avec les autres au sein d'une structure sémantique (dite *indexicale-fonctionnelle*) manipulée par le système de perception de l'agent (par exemple, l' -abeille-derrière-le-bloc-de-glace-auquel-je-fais-face). Le couple aspect/action forme une *routine* et le comportement d'un agent est décrit par la liste de routines qu'il possède.

La programmation des différentes routines de PENGİ par Agre et Chapman ne s'est pas faite de manière intuitive, mais après une longue observation des tactiques de plusieurs joueurs, qui ont été filmés pour l'occasion. Il y a lieu de noter d'ailleurs là une certaine ambiguïté quant au statut exact du pingouin dans le problème. Il est à la fois décrit comme un agent autonome, avec les conséquences que cela entraîne, comme une perception locale de son environnement, et comme l'incarnation du joueur, qui est censé en avoir une vue plus globale.

X.2.2. Pourquoi avoir reprogrammé Pengi ?

L'aspect radical de la thèse d'Agre et de Chapman repose donc sur des principes proches de ceux que nous formulons au sujet des agents réactifs, à savoir que l'environnement déclenche, mais également maintient intégralement l'activité de l'agent. Cependant cette prise de position diffère de nos conceptions par deux aspects essentiels. Le premier est qu'elle dénie à l'agent toute possibilité de posséder un état interne, que ce soit un but, une motivation ou même une mémoire quelconque⁸³. L'"agent situé" qu'est PENGİ est ainsi perpétuellement amnésique, et ne peut donc pas orienter sa perception en fonction de ses buts, ou de sa motivation.

Le deuxième est que le codage de l'information de perception sous la forme indexicale qu'il revêt dans PENGİ est extrêmement complexe. Le pingouin est certes réactif, mais la génération des aspects pouvant déclencher ses actions nécessite un système de fusion de données et de reconnaissance de situations très important, qui s'apparente à une véritable délibération, non plus sur l'action [Jacopin 1992], mais sur la perception. De plus, certaines questions restent en suspens et ne sont pas abordées franchement dans [Agre et Chapman 1987] à propos de PENGİ ou dans [Chapman 1990] à propos de SONJA, un autre jeu vidéo implémenté de la même manière. Il s'agit par exemple de la question de l'orthogonalité des situations reconnues. En effet, si deux situations déclenchent chacune deux actions différentes, que se passe-t-il quand l'agent rencontre ces deux situations simultanément ? Soit la conception du système oblige à ne définir que des *aspects* orthogonaux ou disjoints de l'environnement, auquel cas le problème d'exécution est résolu, mais le problème de conception devient un vrai casse-tête. Soit ces

⁸³ Bien qu'ils reconnaissent que la notion d'état interne joue certainement un rôle important dans le comportement des êtres humains, elle n'est pas employée pour PENGİ.

deux aspects sont réunis pour n'en former plus qu'un seul, auquel on fait correspondre une nouvelle action, ce qui est inenvisageable dans le cas d'un système fortement dynamique, car la bibliothèque de routines de l'agent aurait de fortes chances de grandir exponentiellement (lire une critique similaire formulée par [Maes 1990]). Soit, enfin, un ordre partiel ou total de priorité est définie entre les routines, ce qui serait sans doute la meilleure des solutions, à condition que les priorités puissent évoluer dynamiquement; mais nous réintroduirions ici un état interne implicite de l'agent, ce que Agre et Chapman se refusent à faire (tout en admettant, notamment dans [Agre et Chapman 1990], qu'un arbitrage est nécessaire). Toutes ces ambiguïtés, combinées au fait que nous voulions tester les capacités d'un agent réactif dans une application comme celle-ci, ont fait que nous avons décidé de programmer PENG1 en essayant d'en faire le système le plus simple possible (toujours selon notre principe de parcimonie).

X.3. La Programmation du Système

X.3.1. Les classes d'agents

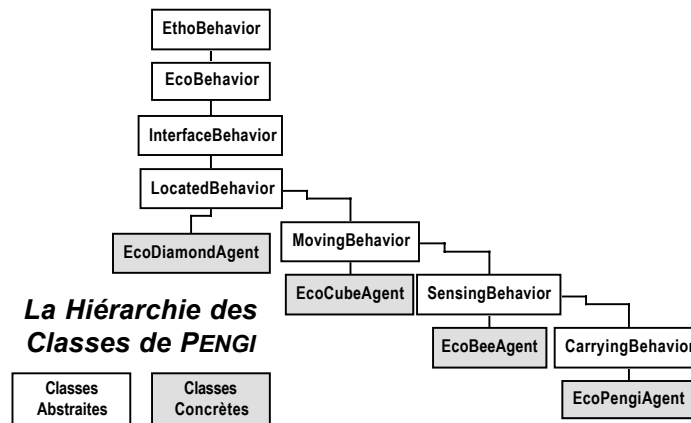


Figure IV.2 - La hiérarchie des classes de l'application PENG1. Plusieurs classes abstraites déjà utilisées pour MANTA sont ici réutilisées.

Les types d'agents présents dans PENG1 sont au nombre de quatre. Les diamants (classe `EcoDiamondAgent`), les cubes de glace (classe `EcoCubeAgent`), les abeilles (`EcoBeeAgent`) et enfin le pingouin (`EcoPengiAgent`). La racine de la hiérarchie de classes que nous avons utilisée dans cette application est composée de `EthoBehavior`, `EcoBehavior`, `InterfaceBehavior` et `LocatedBehavior`. Sachant que les diamants n'ont aucun comportement, ils sont directement sous-classe de `LocatedBehavior`. Les cubes de glace n'ont comme capacité que celle de se déplacer quand ils sont poussés par d'autres agents, ce qui les place sous `MovingBehavior`. Les abeilles savent se déplacer, mais également suivre le gradient émis par le pingouin quand elles en ont la possibilité; leur classe hérite donc de `SensingBehavior`. Enfin, le pingouin pouvant à la fois se déplacer, suivre des gradients et ramasser les diamants, il se place sous `CarryingBehavior`.

De manière générale, voici comment a été conçu le système. Le pingouin est guidé dans sa recherche de satisfaction par un gradient émis par les diamants et propagé dans l'environnement. Il émet lui-même un gradient, de rayon plus limité, qui a pour effet d'attirer les abeilles quand elles le sentent (leur recherche de satisfaction est entachée d'une erreur aléatoire). A leur tour, les abeilles propagent un stimulus qui déclenche chez le pingouin son comportement de fuite. Enfin, les cubes de glace ne sont sensibles qu'au fait d'être poussés par les abeilles ou le pingouin, ce qui déclenche leur comportement

de fuite. Lors de cette fuite, ils propagent un stimulus activant le comportement de fuite du pingouin. Si l'on analyse ces choix à la lumière de notre paradigme de résolution collective de problèmes, nous voyons que les deux buts informels que doit atteindre le pingouin sont, d'une part de supprimer tous les diamants de l'environnement, d'autre part de rester en vie le plus longtemps possible. Et ces buts à atteindre sont bien sûr transcrits dans l'environnement sous la forme d'une entropie globale (le gradient créé par les diamants) et de plusieurs températures locales (celles générées par les abeilles et les cubes de glace en translation), dont la perception par le pingouin déclenche ses comportements téléonomiques respectifs de satisfaction et de fuite, qui doivent avoir des propriétés néguentropiques (disparition ou évitement des gradients mentionnés). Nous allons donc détailler ces comportements dans les sections suivantes, puis nous analyserons les résultats que nous avons obtenus sur différentes instances du problème.

X.3.2. Le comportement des cubes de glace

Les cubes de glace ne possèdent aucun comportement de satisfaction, mais possèdent un comportement de fuite, déclenchable par une interaction directe entre eux et un autre agent, abeille ou pingouin. Les primitives redéfinies de manière à faire fonctionner cette tâche sont les suivantes:

findPlaceForFleeing: aConstraint

cherche une place dans la direction opposée à la contrainte (la place de l'agent qui fait fuir le cube). Si cette place contient un autre cube, renvoie nil.

primAgress: aPlace

ne fait rien

primMoveTo: aPlace

propage un stimulus #flee dans la direction de déplacement. Puis regarde si la place contient un agent, et auquel cas tue cet agent (par appel à `primKill`). Le cube est alors déplacé sur la place passée en argument et s'envoie `flee`: avec en argument sa place antérieure.

Le comportement des cubes de glace qui résulte de la programmation de ces deux primitives est décrit Figure IV.3. Les cubes se déplacent à la même vitesse que les autres agents.

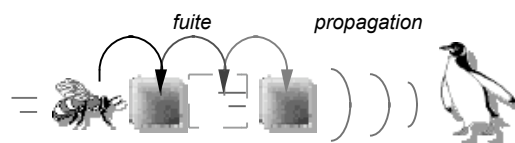


Figure IV.3 - Le comportement d'un cube en fuite. Il se renvoie récursivement le message de fuite jusqu'à ce qu'il bute sur un autre cube et prévient devant lui qu'il se déplace (en propageant #flee).

X.3.3. Le comportement des abeilles

Les abeilles possèdent un comportement de satisfaction et pas de comportement de fuite (le poids de la tâche est à zéro). La finalité de leur comportement de satisfaction est de piquer le pingouin (*i.e.* se trouver sur une case adjacente). Les primitives nécessaires à la tâche de satisfaction sont programmées comme suit:

primCanBeSatisfied

retourne vrai si le pingouin se trouve sur une des cases adjacentes à la case sur laquelle se trouve l'abeille.

primActionSatisfaction

consiste à tuer le pingouin par un appel à primKill:.

findPlaceForSatisfaction

renvoie la case adjacente à la case de l'abeille contenant le stimulus #pengi le plus important. Mais peut également renvoyer n'importe quelle case adjacente selon une probabilité fixée à l'instanciation. Si aucun stimulus #pengi n'est disponible, renvoie une des cases adjacentes choisie au hasard.

primAgress: aPlace

envoie le message flee: à l'agent se trouvant sur la case passée en paramètre, avec la case de l'abeille comme argument. Sert à pousser les cubes.

primMoveTo: aPlace

déplace simplement l'agent sur la place passée en argument.

Pendant leurs mouvements, les abeilles propagent également le stimulus #flee, qui sert à indiquer au pingouin qu'elles sont proches. Ceci est décrit sur la Figure ci-dessous.

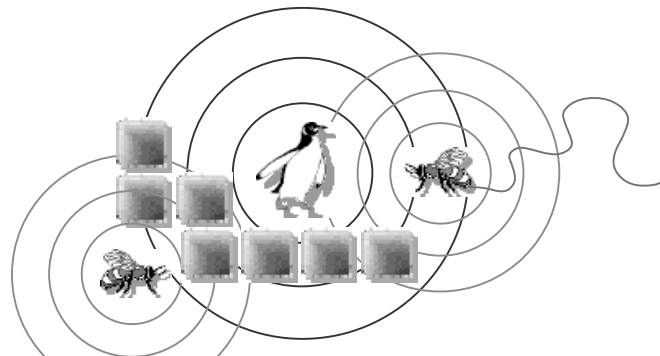


Figure IV.4 - Le comportement des abeilles face au pingouin. Elles sont attirées par le stimulus qu'il propage et émettent dans le même temps un stimulus qui l'éloigne.

X.3.4. Le comportement du pingouin

Enfin, le pingouin possède à la fois un comportement de satisfaction et un comportement de fuite, ce dernier pouvant être déclenché aussi bien par un stimulus en provenance d'un cube de glace que d'une abeille. Son comportement de satisfaction consiste à suivre le gradient émis par les diamants (voir Section suivante comment ce gradient est propagé) et à les ramasser, et son comportement de fuite à s'écarter le plus possible des cases contenant un stimulus #flee, ou au contraire à s'en approcher si la possibilité de pousser un cube sur l'agresseur existe. Dans les deux cas, nous avons bien un comportement négentropique, puisque ramasser les diamants fera disparaître le gradient qu'ils émettent et que fuir les abeilles ou les écraser avec un cube aura le même résultat. Ceci nous fournit les primitives suivantes:

primCanBeSatisfied

retourne vrai si le pingouin se trouve sur une case contenant un diamant.

primActionSatisfaction

consiste à ramasser le diamant par un appel à primPickUp:.

findPlaceForSatisfaction

renvoie la case adjacente à celle où se trouve le pingouin contenant le stimulus #diamond le plus faible. Si aucun stimulus n'est perceptible, stoppe la partie en cours, car ceci signifie soit

que tous les diamants ont été ramassés, soit que les derniers diamants sont bloqués par des cubes de glace.

findPlaceForFleeing

renvoie la case adjacente à celle où se trouve le pingouin contenant le stimulus #flee le plus faible ou pas de stimulus du tout, avec une préférence pour les cases ne contenant pas de cubes de glace. Ou bien la case contenant le stimulus le plus fort, à condition que soit posé dessus un cube de glace.

primAgress: aPlace

envoie le message flee: à l'agent se trouvant sur la case passée en paramètre, avec la case de l'abeille comme argument. Sert à pousser les cubes de glace.

primMoveTo: aPlace

déplace simplement l'agent sur la place passée en argument.

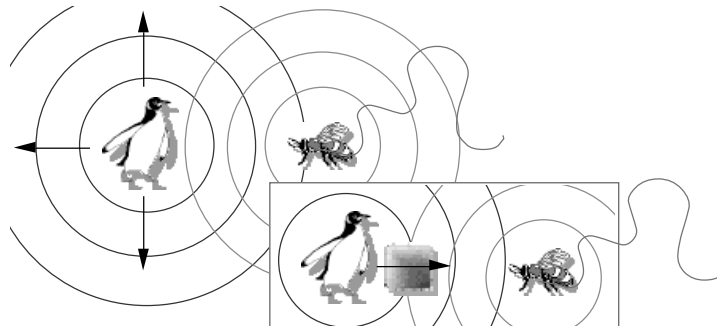


Figure IV.5 - Comportement de fuite du pingouin devant une abeille. Sa réaction est la même s'il est menacé par un cube de glace en mouvement (hormis l'encadré).

X.4. L'Emergence de Comportements

X.4.1. Propagation des stimuli et Planification

La propagation des différents stimuli (#diamond, #flee, #pengi) est différente dans cette application de celle qui est utilisée dans MANTA. Ici, en effet, les obstacles que constituent les cubes de glace ne sont pas totalement imperméables aux stimuli, pour la simple raison qu'ils peuvent être poussés par les agents. Ce qui complique cette propagation, c'est également que la direction selon laquelle les cubes peuvent être poussés va jouer sur cette perméabilité. Comme un cube ne peut en effet être poussé que dans l'une des directions où il n'est pas accolé à un autre cube, les assemblages complexes de cubes nécessitent d'être abordés de manière très fine. Et, enfin, le moindre mouvement d'un cube remet en cause l'ensemble de la propagation (ce cube peut en effet maintenant fermer un accès, ou au contraire en libérer un). Nous avons donc tout un ensemble de contraintes géométriques et dynamiques nouvelles qui vont faire leur apparition dans la propagation des stimuli, ce qui n'est pas très étonnant dans la mesure où ces contraintes définissent en fait des voies d'accès ou de fuite possibles pour le pingouin, informations sans lesquelles son comportement basique est totalement inefficace (voir l'exemple proposé dans [Drogoul, Ferber et Jacopin 1991]). Bien entendu, ces contraintes ne sont pas calculées de manière globale, mais distribuées dans les méthodes de propagation de chacune des cases (qui savent en outre remettre localement en cause un gradient en cas de déplacement d'une cube). Un exemple succinct en est donné Figure IV.6. Une critique souvent formulée à l'encontre de cette technique de propagation est que, d'une certaine manière, elle constitue une planification déguisée du chemin à suivre pour trouver les diamants. Nous ne mettons pas en doute

qu'il s'agit effectivement d'une information permettant au pingouin de notre application de trouver différents chemins le conduisant à des diamants, mais nous insistons sur deux points essentiels qui sont:

- La définition même du jeu implique que le joueur, lorsqu'il manoeuvre le pingouin, soit parfaitement à même de connaître la position des diamants dans l'environnement. Il serait absurde de programmer ce jeu en en compliquant encore plus les règles, par exemple en obligeant le pingouin à rechercher au hasard les diamants.
- Les gradients structurent l'espace de manière à fournir au pingouin un relevé topographique (donc une représentation analogique externe, voir Section X.5) de son environnement immédiat. Pour autant, ils ne lui fournissent pas la suite des actions à entreprendre pour aller d'un point à un autre. Une discussion sur la différence entre cette notion et la notion de plan habituellement utilisée en Intelligence Artificielle peut être trouvée dans [Payton 1990].

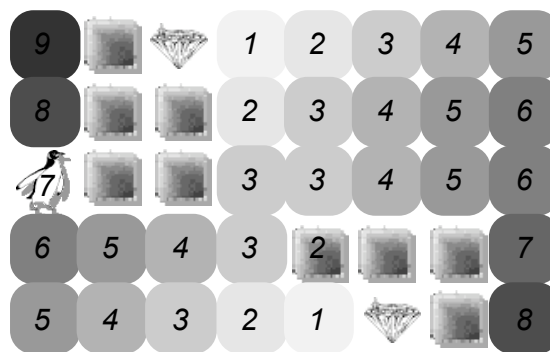


Figure IV.6 - Exemple de propagation du gradient de deux diamants. La perméabilité des cubes n'apparaît qu'à un seul emplacement, correspondant au cube pouvant être déplacé. Le pingouin fuit les valeurs les plus fortes pour aller vers les plus faibles, qui aboutissent à un des deux diamants.

Admettons qu'un explorateur ait à traverser une contrée dont il possède la carte. Dans cette contrée vivent des tribus (anthropomorphes), des animaux sauvages, et les coulées de boues, tremblements de terre et autres catastrophes naturelles n'y sont pas rares. Une fois la traversée accomplie, avec les multiples détours causés par ces menus désagréments, aura-t-on l'idée d'assimiler la suite d'actions effectuées par l'explorateur à un plan entièrement généré par la carte ? Certainement pas. La carte y aura contribué en tant que repère statique, mais n'aura fourni aucune solution aux problèmes dynamiques rencontrés par l'explorateur. Les gradients ne sont pas autre chose pour notre pingouin.

X.4.2. L'Emergence de séquences comportementales

X.4.2.1. DES PLANS EMERGENTS ?

Du fait de la présence de ce guide et de son comportement réactif, l'agent pingouin va donc combiner un bon sens de l'orientation avec un opportunisme de bon aloi (voir Figure IV.7). C'est peut-être ce qui fait la plus grande différence d'avec le système décrit par Agre et Chapman. Leur but était de faire jouer le pingouin en lui fournissant de nombreuses stratégies utilisées par les joueurs (attirer les abeilles dans des pièges, créer ses propres pièges, etc...), et nous montrons que des comportements aussi sophistiqués ne sont pas nécessaires pour le faire gagner. Nous avons ainsi par exemple clairement proscrit de son activité toute "chasse aux abeilles" quand il n'est pas lui-même attaqué.

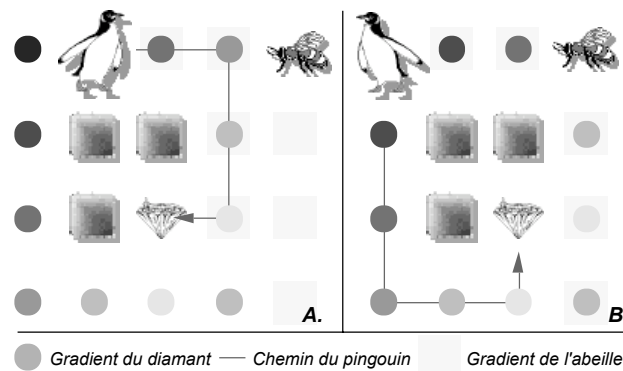


Figure IV.7 - Exemple de comportement opportuniste. Situation A, le pingouin suit un gradient qui le mène au diamant. Mais sa tâche de fuite est déclenchée par la présence d'une abeille. Situation B., le pingouin a fui l'abeille. Le gradient du diamant lui fournit un chemin alternatif, auparavant plus long, mais maintenant de longueur égal au chemin précédent.

De plus, et c'est là qu'un tel système s'intègre tout à fait dans notre démarche générale, nous avons vérifié empiriquement au cours des nombreuses démonstrations effectuées la grande relativité de la notion de "comportement intelligent". En regardant la résolution de PENG1, tout observateur non averti de son comportement réactif se trouvera en face d'un agent qui se déplace à bon escient, qui arrive à ramasser les diamants, à fuir les abeilles et quelquefois à les tuer. Que pourra-t-il conclure d'autre si ce n'est que le pingouin est gouverné par un système "intelligent", déroulant une stratégie invisible à ses yeux mais dont il pensera avoir la preuve de l'existence par le fait même que le problème est résolu ? Il sera de plus conforté dans cette opinion par l'apparition à certains moments de tactiques tout à fait originales et parfaitement adaptées à la situation, tactiques auxquelles il n'avait pas pensé lui-même⁸⁴.

Pour certains problèmes comme le monde des cubes ou les tours de Hanoï [Bura et al. 1991; Drogoul 1990; Ferber 1989; Ferber et Jacopin 1990], on a pu parler d'*éco-planification*, dans le sens où, pour une situation donnée, le système génère une suite d'actions capable de résoudre le problème. L'idée sous-jacente était donc d'utiliser l'éco-résolution dans une simulation du problème, puis de générer à partir de cette simulation la liste d'actions permettant de passer de l'état actuel à l'état final du problème, actions qui devaient être entreprises, bien évidemment, dans l'espace du problème réel et non plus simulé. Or, cette notion n'est à notre sens valable que pour des univers peu dynamiques dans lesquels la solution peut être atteinte par la même suite d'actions que celles obtenues dans la simulation. Si l'ensemble des actions effectuées par le pingouin peut en effet être assimilé à un plan (voir [Drogoul 1993]), celui-ci pourra difficilement être réutilisé pour la résolution d'une autre instance du jeu, la raison étant le comportement semi-aléatoire des abeilles. Un plan global nous paraît à cet égard de peu d'intérêt pour la résolution de ce problème (voir à ce propos [Agre et Chapman 1990])

X.4.2.2. EMERGENCE D'INVARIANTS COMPORTEMENTAUX

Ce qui nous apparaît par contre beaucoup plus intéressant, et qui se rapproche très nettement des travaux de [Connah 1991; Graham et Wavish 1991; Wavish 1992] sur le concept de comportements émergents, est l'apparition d'invariants comportementaux, ou de tactiques, dans des situations précises.

⁸⁴ Les observations énoncées ici proviennent des démonstrations du logiciel organisées sur le stand de l'UPMC au SITEF (Toulouse 1991), et sur le stand de la DGA au Salon d'Avignon, également en 1991. Ces démonstrations nous ont permis d'observer *in vivo* la réaction des personnes au comportement du pingouin, réactions qui toutes confirmaient de manière éclatante les expériences similaires menées par Mitchell Resnick avec son logiciel *LOGO [Resnick 1992].

Ainsi, l'exemple fourni par la Figure IV.8 est l'illustration d'une tactique fort utile qui consiste pour le pingouin, lorsqu'il est attaqué par une ou plusieurs abeilles et qu'il se trouve proche de l'arête d'un mur de glace, à contourner le cube le plus proche pour s'en servir comme projectile.

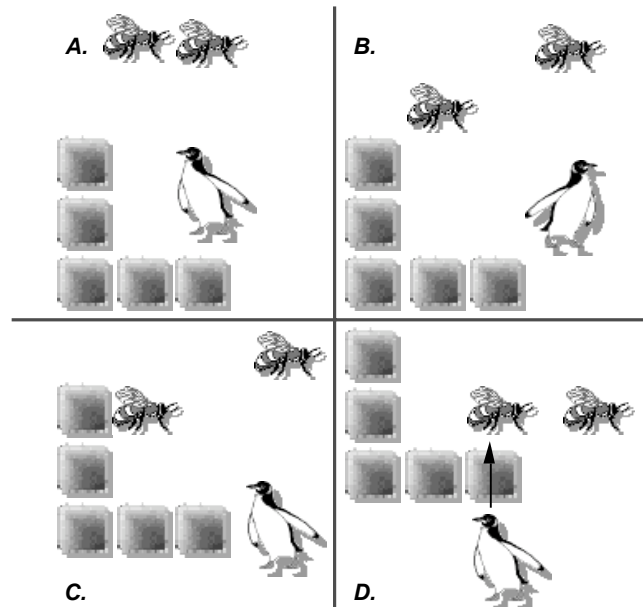


Figure IV.8 - Exemple d'un invariant comportemental du pingouin, qui consiste à contourner les cubes de glace pour en faire des projectiles lorsqu'il se trouve attaqué par des abeilles.

En déchiffrant ce comportement sans rien connaître de sa programmation, il est aisé d'en extraire une intentionnalité certainement partagée par tous les joueurs, qui est de fuir la menace tout en cherchant à contre-attaquer le plus vite possible avant d'être enfermé dans l'essaim des abeilles. En d'autres termes, ce comportement est signifiant pour les personnes regardant le pingouin jouer, d'autant plus qu'il apparaît relativement souvent dans le cours d'une partie, et toujours à bon escient. Cependant, il n'a pas été fourni au pingouin en tant que tel, et nous aurions été d'ailleurs bien en peine de le lui fournir, car il semble à première vue requérir un effort de perception assez important, notamment en terme de reconnaissance de situation. Si maintenant nous analysons cette suite d'actions à la lumière des comportements fournis au pingouin, nous pouvons nous apercevoir qu'en fait il émerge des interactions des deux heuristiques élémentaires contenues dans la méthode `findPlaceForFleeing`: utilisée dans l'activité de fuite. La première est d'essayer, autant que possible, de fuir sur une place ne contenant pas de cube de glace (la même heuristique est d'ailleurs utilisée dans `findPlaceForSatisfaction`, afin dans les deux cas, de ne pas trop bouleverser l'environnement), la seconde est de se déplacer sur une case contenant un cube si et seulement si elle possède le stimulus `#flee` le plus fort. L'interaction et l'enchaînement de ces deux comportements de fuite, qui sont absolument indépendants, donne alors l'impression que le pingouin cherche, et trouve, le cube de glace qui lui permettra de contre-attaquer. Et si le fait de pousser ce cube est bien programmé, toute la séquence d'actions permettant au pingouin de se positionner correctement pour le faire ne l'est pas. Ce qui émerge ici, c'est donc une séquence comportementale assez complexe, séquence comportementale dont le contrôle est directement fonction de la situation dans laquelle est plongé le pingouin. Elle débute par une attaque d'abeilles, et se poursuit par la conjonction de cette attaque et de la présence proche d'un cube de glace susceptible d'être poussé. Ce qui est remarquable, c'est qu'il n'y a nulle part reconnaissance ni représentation, sous quelque forme que ce soit, de la situation par le pingouin, non plus qu'il n'y a de recherche de sa part du cube de glace à pousser. L'ensemble est donc extrêmement robuste vis-à-vis des perturbations éventuelles pouvant survenir dans

l'environnement, et flexible vis-à-vis des modifications de détail pouvant exister entre deux situations proches. Que cela signifie-t-il ? Tout simplement que tout un chacun, y compris nous-mêmes, prêtons sans doute une trop grande intentionnalité, en fait, *notre* intentionnalité, à la plupart des comportements que nous sommes amenés à observer. L'avantage indéniable d'une approche minimaliste comme celle que nous préconisons est de montrer que cette intentionnalité dans le comportement n'existe peut-être, dans bien des cas, que dans notre esprit (voir Chapitre I). Vouloir à tout prix l'inclure dans le système, que ce soit au niveau de la perception ou du comportement, n'est sans doute pas la première des priorités si nous souhaitons obtenir un système réellement adaptatif. Si nous paraphrasons [Simon 1991], nous pouvons donc conclure que *le pingouin considéré comme un système comportemental est un être relativement simple. L'apparente complexité des évolutions de son comportement est pour une grande part le reflet de la complexité de l'environnement dans lequel il se trouve.*

X.4.3. Résultats Expérimentaux et Analyse

Agre et Chapman n'ont jamais publié de compte-rendu d'expérimentation concernant leur système, et selon [Jacopin, comm. pers.], il n'en existe pas. Nous publions donc nos résultats sans pouvoir les confronter aux leurs, ce qui est dommage, mais il sont assez parlants en eux-mêmes pour que nous puissions en tirer quelques conclusions non dénuées d'intérêts. Le Tableau IV.1 montre ainsi, pour une même configuration de l'environnement, les proportions respectives d'échecs et de succès de notre agent confronté à un nombre croissant (de 1 à 6) d'abeilles, toutes dotées d'une probabilité de 0.3 de se déplacer aléatoirement. Il indique également la longueur moyenne de la solution trouvée, à comparer à la longueur optimale calculée pour une partie sans abeilles, qui est de 65 déplacements. Ce que nous pouvons constater, c'est que le pourcentage de succès reste élevé dans toutes les expériences, même s'il est logiquement plus élevé pour les parties contenant peu d'abeilles. Le pingouin ne gagne donc pas tout le temps, mais il arrive en moyenne à gagner deux fois sur trois, ce qui est un bon score, surtout dans les parties comportant plus de quatre abeilles. Un autre point intéressant est que la dégradation de ses performances en présence d'un nombre croissant d'abeilles est plutôt faible, alors même que la dynamique de son environnement devient de plus en plus importante. Nous pouvons d'ailleurs avoir un aperçu de cette dynamique en analysant l'évolution de la longueur moyenne de la résolution qui montre bien que le nombre de déplacements nécessaires à la résolution du problème augmente en fonction du nombre d'abeilles. Là encore, cependant, la dégradation n'est pas catastrophique, dans le sens où à aucun moment n'apparaît une brusque augmentation de ce nombre de mouvements. Le comportement du pingouin n'est donc pas optimal, du moins intuitivement, mais il semble par contre particulièrement bien s'adapter à la complexité croissante de son environnement.

Nombre d'abeilles	Succés	Echecs	Nuls	Longueur Moyenne
1 abeille	70	10	0	90
	87,50%	12,50%	0,00%	
2 abeilles	76	16	8	115
	76,00%	16,00%	8,00%	
3 abeilles	62	30	8	158
	62,00%	30,00%	8,00%	
4 abeilles	49	24	7	172
	61,25%	30,00%	8,75%	
5 abeilles	64	24	12	190
	64,00%	24,00%	12,00%	
6 abeilles	60	32	8	214
	60,00%	32,00%	8,00%	

Tableau IV.1 - Evolution du pourcentage d'échecs et de succès du pingouin en fonction du nombre d'abeilles. Les nuls représentent les parties où le pingouin n'a pu ramasser tous les diamants, mais sans avoir été tué.

X.5. Conclusion

Cette application, au demeurant fort simple, nous a permis de mettre en évidence trois points essentiels. La décomposition du problème a ici été effectuée en deux temps. Une première décomposition en termes de buts à atteindre, ramasser les diamants et rester en vie, perceptibles par l'agent pingouin sous la forme de températures locales matérialisées par des stimuli d'intensité élevés. Une seconde décomposition en termes de comportements téléonomiques d'attraction/répulsion à l'égard de ces perturbations locales, qui forme le modèle de comportement du système d'éco-résolution. Si cette méthodologie n'est pas encore très bien formalisée, elle n'en a pas moins déjà été utilisée avec succès pour d'autres problèmes proches de PENGU (PacMan [Drogoul, Ferber et Jacopin 1991] ou le TileWorld [Drogoul en prép.]). Un agent équipé de ces comportements élémentaires peut suffire à résoudre une tâche complexe, mais à la condition que son comportement soit adapté à l'environnement dans lequel il évolue, c'est-à-dire qu'il sache en retirer le maximum d'informations sans en avoir de représentation (voir [Brooks 1990]).

Mais sommes-nous bien sûr que le pingouin n'a pas de représentation du monde dans lequel il évolue ? En fait, il est fondamental de comprendre qu'il ne possède pas de représentation *symbolique* de son environnement, mais qu'il exploite une représentation *analogique* fournie par les gradients de stimuli. Ceci est donc semblable à ce que propose [Steels 1990], et le fait que ces gradients fassent partie de l'environnement plutôt que d'un système de représentation interne à l'agent n'a que peu d'importance: l'environnement n'est-il pas son meilleur modèle? [Brooks 1991]. Enfin, de même que pour les organisations d'agents qualifiées d'intentionnelles mais dont l'apparition ne nécessite aucune intentionnalité de la part des agents (la chaîne, par exemple), nous pouvons obtenir des séquences comportementales qui seraient très certainement qualifiées d'intentionnelles par un observateur neutre, mais dont l'apparition ne résulte que de l'interaction entre des composants comportementaux de plus bas niveau et l'environnement dans lequel est plongé l'agent.

Il faut bien voir cependant que l'exemple de PENGU n'est pas un bon exemple de résolution *collective* de problèmes. Même s'il est possible d'extrapoler pour inclure dans les agents résolvant le problème les abeilles et les cubes de glace, et de voir ainsi dans sa résolution l'aboutissement d'une dynamique à la fois intentionnelle (due au comportement téléonomique du pingouin) et non-intentionnelle (due aux comportements des abeilles), le comportement permettant d'atteindre la solution n'est implémenté que chez le pingouin. Il nous reste donc à voir si l'emploi de l'éco-résolution, qui offre un modèle de comportement plus épuré que l'EthoModélisation, nous permet de retrouver des propriétés collectives telles que celles que nous avons étudiées dans la troisième Partie. Pour bien mettre en évidence l'apparition de ces propriétés, et surtout leur intérêt en tant qu'alternative aux méthodes plus classiques de résolution de problèmes, nous allons nous attaquer à un problème qui, à première vue, apparaît comme très différent de tous ceux que nous avons abordés jusqu'à présent, puisqu'il n'est pas "naturellement" distribué. Il sera pour nous l'occasion de démontrer qu'une résolution distribuée ne s'applique pas qu'à des problèmes distribués, et qu'elle peut servir dans ce cas à réduire de manière substantielle la complexité et le coût des solutions obtenues.



CHAPITRE XI

UN PROBLEME CLASSIQUE: LE TAQUIN

XI.1. Présentation du problème

XI.1.1. Le Taquin

Le Taquin est un problème célèbre en Intelligence Artificielle, et bien sûr en Résolution de Problèmes, car extrêmement simple à modéliser en termes d'états et de transitions entre états, mais très difficile à résoudre du fait du nombre d'états possibles (voir l'introduction de [Ratner et Warmuth 1986]). C'est un jeu de patience qui consiste en un damier carré de dimension $n \times n$ contenant $n^2 - 1$ palets et un emplacement vide, appelé le "blanc". La seule opération autorisée consiste à déplacer un palet adjacent au blanc sur le blanc, la tâche globale étant de réarranger les palets pour obtenir une configuration globale souhaitée à partir d'une configuration initiale quelconque.

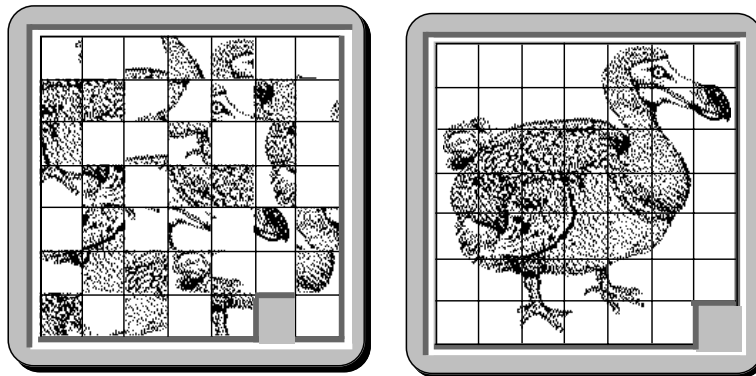


Figure IV.9 - Exemple de Taquin.

XI.1.2. Les méthodes de résolution

Ainsi que le fait remarquer R. Korf dans [Korf 1990a], la planification classique et la recherche heuristique n'apparaissent pas capables, même sur des machines extrêmement puissantes, de résoudre des instances du problème de largeur supérieure à 10 (c'est-à-dire un taquin contenant 99 palets), la raison la plus évidente étant la taille de l'espace d'états généré par le planificateur. Depuis plus de vingt ans, donc, la plupart des travaux se sont concentrés sur des taquins de largeur 4 (15 palets), taquins pour lesquels la solution optimale a pu être trouvée. Mais l'impossibilité pratique de trouver la solution optimale du taquin de taille 5, et le fait que, pour des tailles plus grandes, aucune solution n'a été publiée, ont prouvé l'incapacité partielle de ces techniques à appréhender des problèmes générant un espace d'état important. L'algorithme de recherche le plus employé a sans doute été A* [Hart, Nilsson et Raphael 1968] qui, si on lui fournit une fonction heuristique⁸⁵ correcte, est théoriquement capable de

⁸⁵ Cette fonction heuristique est employée pour évaluer le mérite d'une situation et ainsi pouvoir calculer la distance entre ce mérite et le mérite de la situation finale. L'heuristique la plus employée dans le cas du taquin est la distance de Manhattan généralisée, qui consiste à additionner les distances respectives qui séparent chaque palet de sa position dans la situation finale, situation qui se caractérise par une distance égale à zéro.

trouver la solution optimale pour n'importe quelle taille de taquin. En pratique, cependant, le temps et l'espace requis croissent exponentiellement avec la taille du problème, ce qui fait qu'il n'est guère applicable qu'à des taquins de largeur restreinte (3 pour A*, 4 pour sa variante IDA*, présentée dans [Korf 1985]). Devant cette difficulté, les techniques présentées dans [Korf 1990a; Korf 1990b] ont sacrifié la recherche de la solution optimale au profit d'une limite de la profondeur de recherche. Les deux algorithmes développés, nommés respectivement Real-Time-A* (RTA*) et Learning-Real-Time-A* (LRTA*) ont certes permis de résoudre le taquin de largeur 5 en un temps raisonnable (c'est-à-dire inférieur à celui mis par un joueur humain), mais font apparaître de sévères limitations pour des tailles plus importantes. Premièrement, la profondeur de recherche nécessaire à l'obtention de solutions correctes (*i.e.*, proches de la solution optimale) semble également croître exponentiellement avec la taille du taquin (on passe ainsi de 92 états explorés en moyenne pour un taquin de taille 3 à 2622 états pour un taquin de taille 4). Deuxièmement, les heuristiques employées semblent devoir être adaptées à chaque taille de taquin, car les effets secondaires qu'elles génèrent (voir [Korf 1990c]), surtout pour les taquins de grande taille, augmentent la longueur des solutions, alors qu'elles sont prévues pour faire l'inverse. Tout ceci apparaît d'autant plus paradoxal que la plupart des articles publiés sur la question insistent sur le fait que les techniques employées par l'homme pour résoudre ce problème ne diffèrent pas d'une taille de taquin à l'autre. Il semblerait donc que, si une technique "intelligente" devait être trouvée, elle pourrait tout à fait s'affranchir du facteur d'échelle.

La première constatation qui peut être faite à propos de la recherche heuristique afin d'expliquer cette incapacité est qu'elle adopte une position laissant une place marginale à l'action. L'action, au sens large, c'est-à-dire, pour le taquin, le mouvement d'un palet, n'est considérée que comme le produit d'une recherche plus ou moins longue dans un espace d'états. Or, nous estimons qu'adopter ce point de vue, c'est méconnaître l'aspect fondamentalement organisateur de ce qui pourrait être appelé une "activité de résolution" comme celle qui est à l'oeuvre dans PENGU. Comme le reconnaît R. Korf dans [Korf 1990c], bon nombre d'actions mises en oeuvre par l'homme pour résoudre des problèmes comme le taquin ne sont pas exactement planifiées, mais apparaissent plutôt comme faisant partie d'un "tropisme" vers la solution, la plupart des mouvements réduisant effectivement la distance entre la nouvelle situation qu'ils génèrent et la situation finale, sans pour autant avoir été séquencés par un plan préconçu. C'est cette constatation qui nous a conduit à nous intéresser au taquin. Notre approche va donc consister à réexaminer ce problème à la lumière de la résolution collective de problèmes, en portant notre attention sur, successivement, la méthodologie de décomposition du problème en une population d'agents, l'obtention de structures stables au sein de cette population et de structures d'organisation et de coordination dynamiques permettant de nous affranchir de toute recherche heuristique pour résoudre le problème.

XI.2. Programmation du problème

XI.2.1. Décomposition du problème en agents

La méthode la plus puissante que nous connaissions pour trouver comment résoudre un problème difficile consiste à découvrir un moyen permettant de le diviser en plusieurs problèmes plus simples, dont chacun peut être résolu indépendamment ([Minsky 1988], p. 128)

Nous présentons dans cette section une approche formelle de la décomposition de problèmes comme le taquin en une population d'agents. En effet, si les problèmes précédents étaient aisément décomposables de manière intuitive du fait de leur distribution naturelle, le taquin est habituellement appréhendé sous une forme centralisée. L'approche présentée ici n'est cependant pas la seule possible,

mais nous pensons qu'elle est aisément applicable à d'autres problèmes tout aussi centralisés, car nous avons pris soin de nous appuyer sur une démarche classique en résolution de problèmes, démarche qui consiste à diviser le but global en un ensemble de sous-buts afin de les résoudre indépendamment. Nous allons donc décomposer le but global en un ensemble comprenant les plus petits sous-buts indépendants pouvant être décrits, qu'ils soient considérés comme sérialisables ou non⁸⁶. Si nous appelons **B** le but global, nous obtenons donc:

$$\underline{B = \{ b_1, \dots, b_n \}} \quad (1)$$

où **b**₁, ..., **b**_n sont tous les sous-buts indépendants devant être atteints pour que **B** soit satisfait. Dans le cas du taquin, ces sous-buts sont logiquement de positionner chaque palet sur la case qu'il occupera dans la situation finale. Nous avons donc:

$$\underline{B_{\text{taquin}} = \{ \text{positionner}(p_1, c_1), \dots, \text{positionner}(p_n, c_n) \}} \quad (2)$$

où les **p**_i et **c**_i représentent respectivement les palets et leur case-but. Cette première décomposition nous fournit un ensemble de couples (**a**_i, **but**(**a**_i)), où **a**_i est un composant du problème et **but**(**a**_i) le but (un autre composant) qu'il doit atteindre, c'est-à-dire avec lequel il doit être dans une relation particulière. La satisfaction de chacun des sous-buts est décrite dans notre formalisme grâce à une fonction booléenne appliquée à chacun de ces couples:

$$\underline{\forall b_i \forall a_i / \text{but}(a_i) = b_i, \text{satisfait}(b_i) = f(a_i, \text{but}(a_i)) \rightarrow \{\text{vrai}, \text{faux}\}} \quad (3)$$

Le sous-but **b**_i sera donc considéré comme satisfait ssi **f**(**a**_i, **but**(**a**_i)) retourne vrai. Dans le cas du taquin, nous avons par exemple:

$$\underline{\text{satisfait}(\text{positionner}(p_1, c_1)) = \text{sur}(p_1, c_1)} \quad (4)$$

Si nous changeons notre point de vue pour nous concentrer sur les composants du problème (les agents), dire que **b**_i est satisfait revient à dire que l'agent **a**_i a atteint son but. Nous pouvons donc décrire **a**_i de la manière suivante:

$$\underline{\forall a_i, a_i = \langle \text{but}(a_i), \text{état}(a_i), \text{comportement}(a_i) \rangle} \quad (5)$$

où **but**(**a**_i) représente la même chose qu'en (3), **état**(**a**_i) est l'état courant de l'agent, et **comportement**(**a**_i) un ensemble d'actions qu'il met en œuvre pour atteindre son but à partir de cet état. L'état est défini de manière à autoriser la définition suivante:

$$\underline{\forall b_i \forall a_i / \text{satisfait}(b_i) \Leftrightarrow \text{état}(a_i) = \text{but}(a_i)} \quad (6)$$

⁸⁶ On appelle sous-buts sérialisables tous ceux dont l'obtention ne nécessite pas de défaire un sous-but précédent. Ils peuvent alors être effectués en séquence, puis oubliés.

Si nous appliquons ces définitions au taquin, nous pouvons donc décrire les agents palets de la manière suivante:

$$p_i = \{ c_i, c_k, \text{comportement}(p_i) \} \quad (7)$$

$$\text{satisfait}(\text{positionner}(p_i, c_j)) \Leftrightarrow c_k = c_i \quad (8)$$

où c_i est le but de p_i , c_k est la case où il se trouve actuellement et $\text{comportement}(p_i)$ un ensemble d'actions lui permettant de glisser depuis cette case jusqu'à son but.

Une fois cette décomposition effectuée, la satisfaction d'un sous-but dépend maintenant entièrement de la capacité de son agent à atteindre son but. Nous ne décomposons donc pas le problème en un ensemble de sous-buts de manière à optimiser une recherche globale, comme le propose [Korf 1990c], mais nous distribuons également le système de résolution lui-même en un ensemble de processus indépendants qui constitueront les comportement des agents. Nous passons donc d'une description "orientée-état" à une description orientée agents du problème.

Dans le même esprit, la description de l'état initial et de l'état final d'un problème P va se réaliser par l'intermédiaire des agents qui le composent. Son état initial Init_P est donné par la liste de leurs états initiaux et de leurs comportements, son état final Final_P par la liste de leurs buts respectifs, comme indiqué ci-dessous:

$$P = \{ a_1, \dots, a_n \} \quad (9)$$

$$\text{Init}_P = \{ (\text{état}(a_1), \text{comportement}(a_1)), \dots, (\text{état}(a_n), \text{comportement}(a_n)) \} \quad (10)$$

$$\text{Final}_P = \{ \text{but}(a_1), \dots, \text{but}(a_n) \} \quad (11)$$

Le problème est donc considéré comme résolu quand tous les agents ont atteint leur but, ce qui signifie effectivement, si l'on se réfère à la définition (6), que tous les sous-buts ont été satisfaits. En instanciant cette définition pour le problème du taquin, nous obtenons donc la décomposition suivante:

$$\text{Taquin} = \{ p_1, \dots, p_n, c_1, \dots, c_k \} \quad (12)$$

$$\text{Init}_{\text{Taquin}} = \{ (c_\alpha, \text{comportement}(p_1)), \dots, (c_\lambda, \text{comportement}(p_n)) \} \quad (13)$$

$$\text{Final}_{\text{Taquin}} = \{ \text{but}(p_1) = c_i, \dots, \text{but}(p_n) = c_j \} \quad (14)$$

La population d'agents du Taquin comprend donc des agents palets et des agents cases, la structure formée par ces derniers servant d'environnement aux palets.

XI.2.2. Le noyau d'EcoTalk

Le modèle de comportement des agents que nous allons utiliser dans le Taquin du est toujours l'éco-résolution, mais dans sa version originelle, identique à celle présentée par Jacques Ferber [Ferber 1989], et reprogrammée en Smalltalk-80 comme extension au langage d'acteurs Actalk, sous le nom ECO-TALK [Drogoul 1990]. Nous n'utiliserons donc pas l'EthoModélisation dans cette partie, et ce pour deux raisons. La première est purement historique et tient au fait que les premières versions du Taquin ont été programmées avant que le noyau d'EthoModélisation ne soit écrit, et que nous avons fait évoluer le programme à partir de ces versions, en y greffant certaines parties logicielles venant de l'EthoModélisation quand le besoin s'en est fait sentir.

La seconde est que le noyau de base de l'éco-résolution (et les modifications qui y ont été apportées, voir [Bura et al. 1991; Ferber et Drogoul 1992]) se prête facilement à un travail de formalisation du fait de l'extrême compacité de son code et de la transparence qui en résulte. Nous avons besoin de cette qualité pour démontrer la complétude de notre approche (voir Section X1.4.1.). La différence entre la version que nous allons donc utiliser pour le Taquin et la version que nous avons présenté au Chapitre IX réside dans les deux aspects suivants:

- Le comportement n'est plus déclenché que par envois de messages en provenance soit de l'utilisateur, soit d'autres agents. Il n'y a donc pas de déclenchement possible des comportements de fuite ou de satisfaction par propagation de stimuli dans l'environnement. Nous verrons que les mécanismes de propagation ne seront plus utilisés que comme représentation analogique de l'environnement.
- Les agents sont dotés de deux variables d'instances, ou plutôt de deux listes d'accointances, qui se nomment `goal` et `dependencies`. La variable `goal` représente explicitement le but de l'agent, c'est-à-dire un autre agent du système avec lequel il doit être dans une relation particulière, appelée relation de satisfaction, pour être satisfait, et `dependencies` une liste d'agents dont la satisfaction dépend de l'agent considéré. Cette liste est construite dynamiquement, en partant du postulat que si un agent **A** a pour but un agent **B**, alors **A** devra attendre que **B** soit satisfait pour se satisfaire, ce qui signifie qu'il *dépendra* de lui. **A** fera donc partie des *dépendances* de **B**. Ces dépendances seront averties par **B** qu'elles pourront se satisfaire lorsque lui-même se sera satisfait.

Le noyau d'EcoTalk est ainsi constitué de la classe `EcoAgent`, dont la définition est la suivante:

```
ActorBehavior subclass: #EcoAgent
  instanceVariableNames: 'goal dependencies '
  category: 'Eco-Resolution'
```

Les méthodes qui constituent le noyau sont communes à tous les agents et font appel, pour certaines, à des méthodes dépendantes du domaine de résolution (équivalentes aux primitives que nous avons dû coder pour PENG1, elles sont indiquées en italique). Du fait de leur petit nombre, nous pouvons les détailler ici⁸⁷:

```
doGoal: aGoal replyTo: cont
  goal := aGoal.
  goal addDependency: self replyTo: cont
```

La méthode `doGoal:replyTo:` est la méthode permettant d'activer l'agent. Elle initialise son but, puis fait appel à `addDependency:replyTo:`

```
addDependency: anAgent replyTo: cont
  self satisfied
    ifTrue: [anAgent trySatisfactionReplyTo: cont]
    ifFalse: [dependencies add: anAgent]
```

⁸⁷ La programmation du noyau s'est effectuée en utilisant des continuations explicites, d'où le suffixe `replyTo:` rajouté à la plupart des méthodes, qui permet de passer cette continuation en argument.

Cette méthode permet de calculer la liste des dépendances d'un agent. Si cet agent est déjà satisfait, il demande simplement à l'agent qui s'est passé en argument d'essayer de se satisfaire. S'il ne l'est pas, il le rajoute parmi ses dépendances.

```
trySatisfactionReplyTo: cont
  self satisfactionAgressionReplyTo:
    [self doSatisfactionReplyTo:
      [self informReplyTo: cont]]
```

La méthode permettant aux agents d'essayer de se satisfaire fait successivement appel à deux méthodes dépendantes du domaine d'application. (1) *satisfaction AgressionReplyTo:* est appelée pour le cas où l'agent ne peut immédiatement se satisfaire (il doit faire fuir d'autres agents avant), et (2) *doSatisfactionReplyTo:* doit implémenter l'action finale de satisfaction.

Après quoi, l'agent n'a plus qu'à informer ses dépendances qu'il est satisfait par un appel à *informReplyTo:*

```
informReplyTo: cont
  dependencies do:
    [:each | each satisfied
      ifTrue: [each informReplyTo: cont]
      ifFalse: [each trySatisfactionReplyTo: cont]]
```

Cette méthode permet à un agent satisfait de prévenir ses dépendances qu'elles peuvent maintenant se satisfaire, en leur envoyant le message *try SatisfactionReplyTo:* si elles ne le sont pas déjà.

```
flee: aConstraint replyTo: cont
  | place |
  place := self findPlaceForFleeing: constraint.
  self fleeAgression: place replyTo: [self doFlee: place replyTo:
  cont]
```

La méthode *flee:replyTo:* est la seule méthode du noyau qui ne fasse pas partie de la boucle de satisfaction. Elle est déclenchée lorsque l'agent reçoit le message correspondant d'un autre agent pour lequel il constitue une gêne. Elle prend en argument une contrainte (qui doit être un agent, par exemple l'emplacement de l'agent attaquant) et fait appel à trois méthodes dépendantes du domaine de résolution. (1) *findPlaceForFleeing:* permet à l'agent de trouver une place où fuir, (2) *fleeAgression:replyTo:* lui permet de faire éventuellement fuir à son tour l'agent se trouvant sur cette place, et (3) *doFlee:replyTo:* d'effectuer l'action réelle de fuite.

Il est à noter, et ceci sera important par la suite, que les contraintes ainsi passées en argument ne se propagent pas d'agression en agression (on peut le voir aisément à la lecture de la méthode) et ne sont donc valables que pour une seule interaction.

XI.2.3. Ordonnancer les satisfactions des agents

La différence essentielle entre le Taquin et les problèmes que nous avons abordés précédemment réside dans son aspect très contraint, et, pour tout dire, peu dynamique. Ceci provient essentiellement

du fait qu'un palet seulement peut se déplacer à la fois sur le blanc, ce qui empêche d'avoir plusieurs palets qui se satisfont simultanément. La solution retenue dans un premier temps a donc été de générer une liste arbitraire aléatoire qui ordonnerait les palets en se basant sur le système de dépendances.

Cette solution est décrite dans [Drogoul 1990]. Les palets qui y sont donc activés dans un ordre aléatoire, cherchent à se satisfaire, puis passent la main aux palets qui dépendent d'eux. Tous les palets satisfaits qui se trouvent déplacés par l'un des palets suivants sont de nouveau replacés dans la liste et ceci jusqu'à ce que la liste soit vide.

Pour plusieurs raisons, dont le désir que nous avons de prouver la complétude de notre méthode de résolution, cette solution a été assez vite écartée. En effet, tout palet satisfait pouvait être amené à fuir s'il était agressé par l'un des palets suivants dans la liste et nous n'avions aucun moyen, autre qu'empirique (car cette solution fonctionne parfaitement), de prouver la convergence du système vers l'état final.

La solution que nous présentons ici va donc consister à ordonner les tentatives de satisfaction des palets de manière "intelligente", avant même le début de la résolution. Il s'agit donc, et nous ne le cachons pas, d'une heuristique globale qui contrevient donc à certains des principes que nous énonçons depuis le début de cette section. Mais il semble que ce soit le prix à payer pour pouvoir comparer en toute rigueur nos résultats avec ceux obtenus en recherche heuristique.

L'idée de base de cette solution est de faire en sorte que la satisfaction d'un palet gêne le moins possible la recherche de satisfaction des palets suivants. En examinant les différentes configurations possibles, il est aisé de se rendre compte que ce sont les palets dont la position finale est en bordure du taquin qui devront être choisis en premier, afin de laisser le maximum d'espace aux suivants.

De plus, il est préférable, pour des raisons évidentes, de choisir une ligne ou une colonne qui pourra être complétée, ce qui élimine celles contenant le blanc dans la situation finale. Il ne nous reste donc plus comme choix que la ligne ou la colonne la plus éloignée du blanc dans sa position finale. En appliquant ce raisonnement récursivement à l'ensemble du taquin, on obtient alors le chaînage des cases suivant (si la position finale du blanc est en bas à droite):

Chaînage obtenu d'après la configuration finale

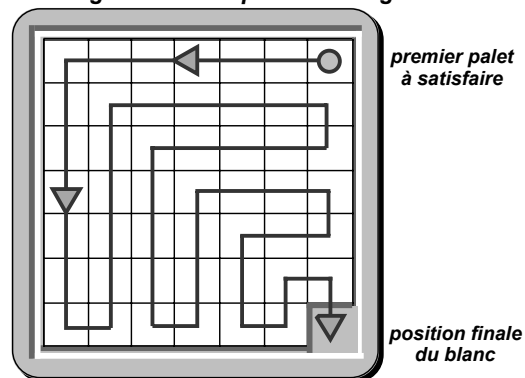


Figure IV.10 - Chaînage des satisfactions. En fonction de la configuration finale désirée, le premier palet à se satisfaire sera celui devant se rendre dans le coin supérieur droit, suivi par son voisin de gauche, etc. Le chaînage se termine logiquement par la position du blanc.

Ce chaînage est calculé en appliquant au taquin, avant le lancement de la résolution, un programme écrit en éco-résolution capable de le construire pour n'importe quelle configuration finale. Ce programme, que nous ne décrivons pas ici (voir [Drogoul et Dubreuil 1990]), fournit à chacun des agents (palets et cases) une relation de dépendance correspondant à l'ordre choisi. Ces relations de dépendance sont créées en fournissant un but fictif aux cases. En effet, si nous posons que $c_i > c_j$ signifie que c_i est placée juste avant c_j dans la liste de chaînage,

$$\text{si } c_i > c_j \text{ et } \text{but}(p_i) = c_i \text{ alors } \text{but}(c_j) = p_i \quad (17)$$

$$\text{ce qui signifie: } \text{dependance}(c_i) = p_i \text{ et } \text{dependance}(p_i) = c_j \quad (18)$$

Donc, quand la palet p_i se satisfera, il ira informer sa dépendance c_j qu'elle pourra se satisfaire. Comme les places sont toujours considérés comme satisfaites, c_j demandera à son tour à sa dépendance de se satisfaire, et ainsi de suite jusqu'à ce que la dernière case soit atteinte (case dont le but est le dernier palet à satisfaire) qui, elle, n'a pas de dépendances.

Un effet de bord intéressant de ce chaînage (déjà souligné par [Korf 1990c]) est qu'il permet de réduire progressivement la taille du taquin pendant la résolution. En effet, une fois que la première colonne et la première ligne ont été résolues, le système se retrouve dans un état pseudo-initial avec un taquin dont la largeur est décrétementée de 1 par rapport à la largeur initiale. De plus, les agents composant cette ligne et cette colonne ne seront plus déplacés par les agent suivants, ce qui réduira les mouvements nécessaires au déplacement d'un palet se satisfaisant.

XI.2.4. Le comportement des agents Palets

XI.2.4.1. LES ACTIONS DEPENDANTES DU DOMAINE

Les agents palets sont instance de la classe `EcoPalet`, sous-classe directe d'`EcoAgent`. C'est dans cette classe que vont être programmées les actions dépendantes du domaine (ici, le taquin) permettant à ces agents d'atteindre leur but. Sa définition est la suivante:

```
EcoAgent subclass: #EcoPalet
instanceVariableNames: 'patch '
category: 'Eco-Taquin'
```

où `patch` représente la place sur laquelle se trouve le palet. Les méthodes codant les actions des palets sont décrites ici partiellement (toutes les méthodes d'accès aux variables, de même que celles comme `moveOn:replyTo:` qui ressemblent à l'identique à des primitives du noyau d'EthoModélisation sont ici omises):

```
satisfied
    ^(patch == goal)

canBeSatisfied
    ^(patch adjacents includes: goal) & (goal isEmpty)

canFleeOn: aPatch
    ^(aPatch isEmpty)
```

Ces trois méthodes permettent de vérifier l'état de l'agent. Il est satisfait si sa place actuelle est égale à son but, il peut immédiatement se satisfaire si son but est adjacent à sa place et s'il est vide (*i.e.* si aucun autre palet n'est posé dessus), et il peut fuir sans agression sur une place si elle est également vide.

```
doSatisfactionReplyTo: cont
  self satisfied ifTrue: [^cont reply].
  self moveOn: goal replyTo: cont
```

L'action de satisfaction consiste simplement à se déplacer sur le but de l'agent s'il n'y est pas déjà.

```
satisfactionAgressionReplyTo: cont
| constraint aPatch |
(self canBeSatisfied | self satisfied) ifTrue: [^cont reply].
aPatch := self findPlaceForSatisfaction.
constraint := self findConstraintForSatisfaction.
self aggress: aPatch with: constraint replyTo:
  [self moveOn: aPatch replyTo:
   [self satisfactionAgressionReplyTo: cont]]
```

Si l'agent ne peut immédiatement se satisfaire (*i.e.* il est séparé de son but par plusieurs places), il va rechercher une place le rapprochant de son but et agresser cette place (faire fuir le palet qui s'y trouve) afin de s'y poser. Il rappelle ensuite récursivement cette méthode, jusqu'à ce qu'il puisse se satisfaire, auquel cas il passe la main à la méthode précédente.

```
doFlee: aPatch replyTo: cont
  self moveOn: aPatch replyTo: cont
```

L'action de fuite sur une place (déjà choisie dans le noyau) consiste simplement à se déplacer sur cette place passée en paramètre.

```
fleeAgression: aPatch replyTo: cont
(self canFleeOn: aPatch) ifTrue: [^cont reply].
self aggress: aPatch with: patch replyTo: [self fleeAgression:
place replyTo: cont]
```

Si l'agent ne peut pas immédiatement fuir sur une place, il l'agresse en lui passant comme contrainte la place sur laquelle il se trouve (le palet qui l'occupe devra fuir ailleurs). Cette méthode s'appelle récursivement jusqu'à ce que la place soit libre.

```
aggress: aPatch with: aConstraint replyTo: aCont
  aPatch isEmpty ifTrue: [^aCont reply].
  aPatch agent flee: aConstraint replyTo: aCont
```

Enfin, la méthode d'agression d'une case vérifie tout d'abord si la place n'est pas vide, puis demande à l'agent qui s'y trouve de fuir en respectant la contrainte passée en argument, par un appel direct à la méthode `flee:replyTo:` du noyau. Le comportement d'un agent n'est donc pas bien compliqué. Il cherchera à se satisfaire tant qu'il ne le sera pas, en poussant les palets qui le gênent dans son entreprise. Et tout palet agressé cherchera simplement à laisser le chemin libre à celui qui le pousse, en agressant au besoin d'autres agents qui le gênent, lui, pour fuir. Il nous reste simplement à voir de quelle manière les palets choisissent les places leur permettant soit de se satisfaire, soit de fuir, et, comme nous le verrons dans les parties suivantes, il s'agit du point le plus délicat.

XI.2.4.2. LE CHOIX DU DEPLACEMENT

Ce choix est effectué, si nous regardons les comportements des agents, dans les méthodes `findPlaceForSatisfaction`, `findPlaceForFleeing`: et `findConstraintForSatisfaction` (cette dernière permettant de contraindre la fuite des agents agressés de manière appropriée à la situation).

En recherche heuristique, comme nous l'avons déjà évoqué, la manière la plus usuelle de mesurer la distance séparant tout état du taquin de l'état final consiste à calculer la distance de Manhattan de chacun des palets à son but, et d'en faire la somme. Ceci fournit une bonne estimation du désordre de la configuration courante par rapport à la configuration recherchée (où chacune des distances doit être égale à zéro). Or, d'un point de vue local, ou, si l'on préfère, du point de vue du palet, la distance de Manhattan entre sa position et toute autre position sur la taquin lui fournira la mesure exacte du nombre de mouvements qu'il aura à effectuer pour la rejoindre. S'il souhaite donc se rapprocher d'une place particulière, il devra choisir, parmi les places adjacentes à celle qu'il occupe, celle dont la distance de Manhattan à cette place est la plus réduite.

Ce qui nous intéresse de plus est que le calcul de cette distance peut s'effectuer par simple propagation d'un gradient, équivalent aux gradients que nous avons utilisé dans les applications précédentes, entre les deux places concernées, la distance étant égale à la différence de potentiel existant entre ces deux places (*i.e.* la différence d'intensité entre les stimuli présents sur chacune d'entre elles).

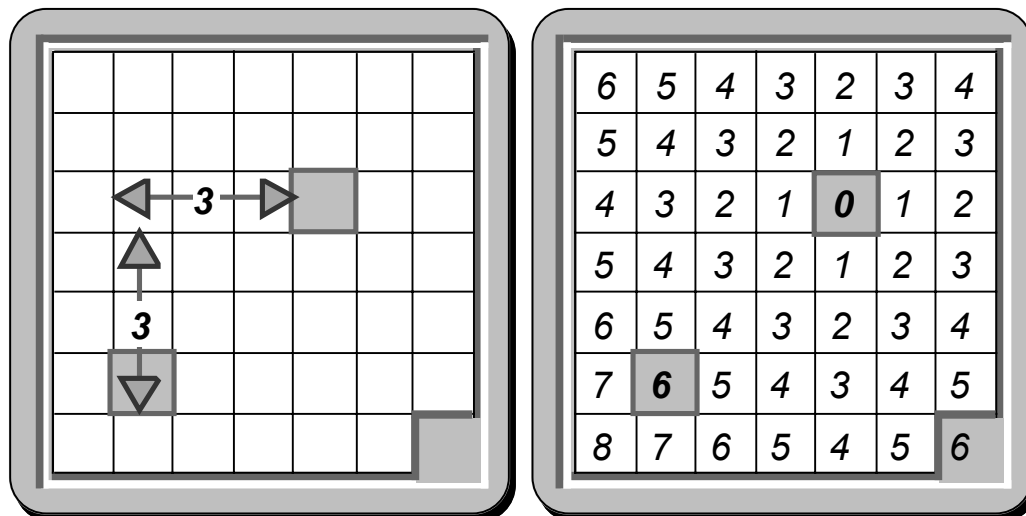


Figure IV.11 - A gauche le calcul de la distance de Manhattan entre deux positions. A droite, l'émission d'un gradient incrémenté de 1 à chaque propagation à partir de la case supérieure. La différence de potentiel entre les deux positions est bien égale à la distance de Manhattan entre elles.

Si un palet souhaite par exemple se rapprocher de son but, il lui suffira donc d'aller interroger chacune des places adjacentes pour connaître l'intensité de leur stimulus, et d'élire celle qui possède la plus faible différence de potentiel d'avec sa case but.

Le choix des mouvements effectués par un palet se fera dans l'application sur la base de deux informations: sa distance à son but, et sa distance au blanc. La première distance sera calculée à chaque fois que le palet devient actif (qu'il doive se satisfaire ou fuir, il enverra à son but l'ordre de propager un

stimulus `#goal`), et la seconde maintenue dans l'environnement par l'intermédiaire de la propagation du stimulus `#blank` par la case ne contenant pas de palet. Les dépropagations sont bien sûr automatiquement effectuées à chaque changement. Nous pouvons donc maintenant décrire les méthodes servant à choisir la place où un palet va se déplacer:

- `findPlaceForSatisfaction` va choisir la place adjacente la plus proche du blanc parmi celles qui sont les plus proches du but de l'agent (dans le cas où plusieurs places sont équidistantes de son but).
- `findPlaceForFleeing`: va choisir le but de l'agent s'il est adjacent, et sinon la place adjacente la plus proche de ce but parmi celles qui sont les plus proches du blanc (dans le cas où plusieurs places sont équidistantes du blanc).
- Enfin, `findConstraintForSatisfaction` renverra le but de l'agent si l'agent n'en est pas adjacent, et sinon la place actuelle du palet précédent (dans l'ordre de la liste de chaînage).

Nous pouvons voir que les calculs de distance sont utilisés de manière différente par les deux méthodes de recherche de place. Dans son comportement de satisfaction, un palet va chercher à minimiser la distance qui le sépare de son but, puis à minimiser le nombre d'agressions nécessaires à ce mouvement (en choisissant la case la plus proche du blanc). Dans son comportement de fuite, la distance au blanc devient prédominante, bien que l'agent soit toujours attiré par son but, ceci afin de minimiser ses déplacements futurs lorsqu'il devra à son tour se satisfaire.

La méthode permettant de fournir une contrainte lors de l'agression d'un agent par un autre qui cherche à se satisfaire est quant à elle fondée sur le simple bon sens. Le palet se satisfaisant va demander au palet agressé de ne pas se déplacer ni sur son but (puisque c'est là qu'il souhaite aller), ni sur la place du palet précédent (dont on fait l'hypothèse qu'il s'est déjà satisfait et qu'il ne faut donc pas le déranger). A première vue, le comportement des agents est donc complet. Ils sont pourvus de toutes les actions nécessaires pour mettre en oeuvre leurs comportements de satisfaction et de fuite, ainsi que des méthodes de recherche locales leur permettant de rejoindre leurs buts. Mais il reste cependant un problème à régler, qui est celui des boucles dans les agressions multiples.

XI.2.4.3. COMMENT SUPPRIMER LES BOUCLES ?

Il y a de nombreuses situations favorisant l'apparition de boucles d'agression dans le système tel que nous l'avons décrit. Par exemple, si deux palets adjacents devant fuir sont chacun posés sur le but de l'autre, ils s'agresseront indéfiniment. Plus généralement, nous pouvons établir que si un palet déjà en fuite (et qui attend que la place choisie pour fuir se libère) se trouve de nouveau agressé dans la continuation de l'agression qu'il a effectuée, tous les palets faisant partie de la boucle ainsi générée s'agresseront indéfiniment. Ils n'auront en effet pas de raison objective de faire, lors de la seconde agression, un choix différent de celui qu'ils ont effectué la première fois, et continueront donc à agresser le palet qu'ils agressaient précédemment. Il nous faut donc trouver un moyen de fournir aux palets une information locale leur permettant d'éviter l'apparition de boucles. Dans la plupart des systèmes réactifs que nous avons présentés jusqu'ici, nous avons toujours insisté sur le double processus de structuration intervenant entre la population d'agents et l'environnement. L'environnement est en partie façonné par les agents, et la transformation qui en résulte va orienter leur comportement de manière adéquate. L'action d'un agent dans le monde va donc se répercuter d'une manière ou d'une autre sur le comportement des autres agents, via l'environnement.

Si nous revenons au taquin, nous constatons que les agents palets utilisent l'environnement comme guide (par l'intermédiaire des méthodes décrites dans la section précédente), en se basant sur la manière dont les gradients partant de leur but et du blanc y sont propagés. Or, ce guide n'est jamais affecté par le comportement des autres palets, ni par leur prise de décision. Et c'est pour cette raison que des boucles peuvent apparaître. Parce qu'un agent aura choisi une place qui lui apparaît comme plus proche du blanc, alors même que le chemin vers le blanc contient des palets déjà en train de fuir. Parce qu'un agent choisira de se déplacer sur son but, alors que l'agression dont il est victime provient précisément du palet se trouvant sur son but.

Il faut donc que le comportement, ou que l'état d'un agent, ait un effet sur l'environnement, effet qui modifie le guide fourni aux agents suivants de façon à modifier leur choix au moment de décider de la place où fuir, en faisant en sorte que le déplacement vers des palets déjà en train de fuir devienne moins attractif. Or, le guide fourni aux agents est composé de deux gradients. Si nous voulons que les agents qui les perçoivent évitent les palets déjà en fuite, il suffit de considérer les cases de ces palets comme des obstacles temporaires. De cette manière, le gradient contournera tous les palets déjà en fuite, et l'agent qui le suit aveuglément en fera de même. De plus, le comportement qui leur est actuellement fourni n'aura besoin que d'être très légèrement modifié, la seule modification devant intervenir dans les méthodes de propagation des cases. La Figure IV.12 fournit deux exemples où le fait de tenir compte des palets en fuite permet d'éviter l'apparition de boucles.

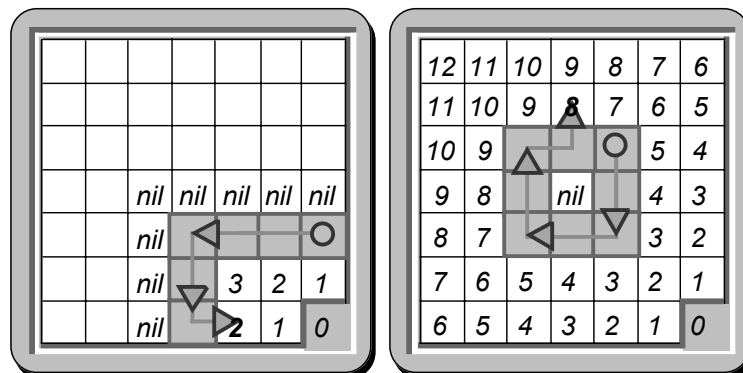


Figure IV.12 - Deux exemples d'évitement de boucles lors d'agressions en chaîne. La figure de gauche montre que le dernier palet devant fuir est obligé de choisir la case le rapprochant du blanc, dans la mesure où le gradient n'arrive plus sur les cases au-delà. La figure de droite montre que le palet devant fuir se trouve également obligé de choisir une case différente du centre du carré d'agression, et différente des cases des palets en fuite (ces deux choix conduiraient à une boucle), parce que le gradient ne peut les atteindre.

Concrètement, la transcription de cette fonctionnalité dans le comportement du système s'effectue en rajoutant le champ booléen `locked` aux places. Les palets devant fuir ou se satisfaire mettront ce champ à vrai au début des méthodes `fleeAgression:replyTo:` et `satisfactionAgression:replyTo:`, et il sera remis à faux lors de leur déplacement, soit dans la méthode `moveOn:replyTo:`. De la même manière, pour empêcher que les palets satisfaits ne soient déplacés par les palets suivants, ils bloqueront leur case but dans la méthode `doSatisfaction:replyTo:`. Enfin, s'il arrive qu'un palet ne puisse trouver une case où se déplacer, c'est-à-dire si la conjonction de la contrainte qui lui est fournie et du blocage des places ne lui laisse aucune place disponible, il peut envoyer le message `unlockAll` à sa place, ce qui aura pour effet de débloquer toutes les places du taquin qui sont bloquées (par simple propagation du message entre les places). Ce message est envoyé dans la méthode `findPlaceForFleeing:`, qui recalcule alors un nouveau gradient après que toutes les cases sont débloquentées. L'interprétation visuelle la plus parlante de ce mécanisme de blocage peut-être faite en terme de représentation topographique de l'environnement. Si nous assimilons le gradient

en provenance du blanc à une indication de la pente, le palet suivra tout simplement la loi de la gravité (voir Figure IV.13), les places bloquées devenant alors des pics infranchissables.

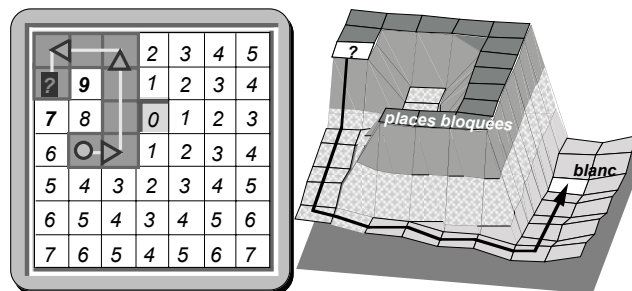


Figure IV.13 - La topographie du taquin vue par le palet devant fuir (marqué d'un point d'interrogation) Il s'agit d'une interprétation du gradient en terme de déclivité. La flèche noire indique le chemin le plus court entre sa place et le blanc.

La présentation de ce mécanisme clôt la section consacrée aux comportements des agents. Nous allons maintenant montrer comment la société que forment ces agents arrive à résoudre le problème, avant de nous attacher à démontrer la complétude du système de résolution qu'elle constitue.

XI.3. Illustration

XI.3.1 Résolution d'un Taquin 3x3

Afin de constater *de visu* que le système fonctionne, voici, Figure IV.14, la trace graphique d'une résolution. Pour des raisons de place, seuls les mouvements des palets cherchant à se satisfaire sont ici décomposés.

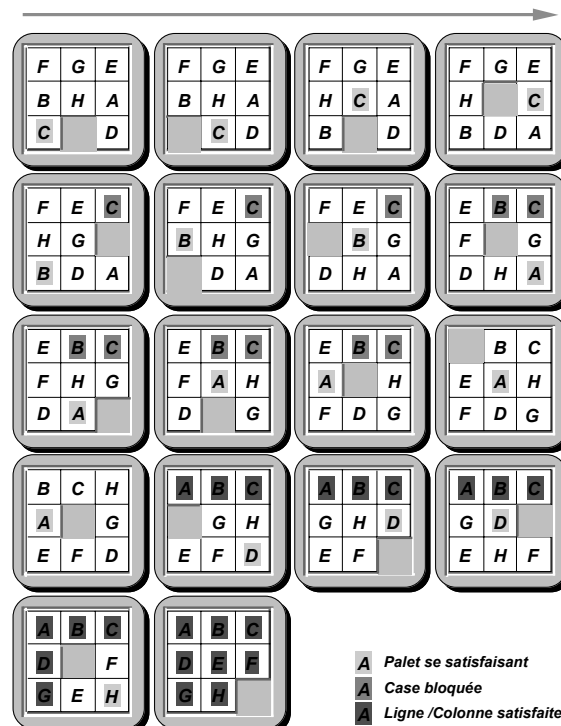


Figure IV.14 - Exemple de résolution d'un taquin de largeur 3. La figure se lit de gauche à droite et de bas en haut.

XI.3.2 Résolution des coins du Taquin

On pourrait avoir l'impression, à la lecture des sections précédentes, que rien ne peut réellement émerger du système qui n'ait été prévu explicitement par le concepteur. L'ordre de satisfaction est fixé, et les heuristiques locales suffisamment complètes pour ne rien laisser au hasard. Il faut cependant comprendre que faire se déplacer un seul palet vers son but est à la portée de n'importe quel système de résolution, qu'il planifie ou pas son action. Et c'est précisément ce que nous avons décrit jusqu'à présent. La difficulté du taquin ne provient donc pas de la résolution individuelle des sous-buts (ou, pour nous, de la satisfaction de chaque agent pris individuellement), mais du fait que ces sous-buts ne sont pas sérialisables. Il n'est donc pas possible de les traiter en séquence, l'un après l'autre, sans remettre en cause les buts précédemment résolus⁸⁸. Ce qui s'exprime en terme de recherche ou de planification dans les systèmes classiques, va s'exprimer dans notre système sous la forme d'une coordination nécessaire entre les agents déjà satisfaits et les agents devant se satisfaire dont l'action risque de remettre en cause les satisfactions précédentes.

L'exemple le plus connu est celui des coins du taquin. Quand tous les palets composant une ligne (ou une colonne), sauf le dernier, sont satisfaits, il est pratiquement impossible (sauf cas exceptionnel) de le satisfaire sans avoir à déplacer les palets précédents. Aucun comportement particulier n'a été fourni aux palets pour le résoudre et, cependant, dans toutes les résolutions, la même tactique est employée, tactique qui s'adapte de plus de manière très souple aux configurations parfois fort différentes auxquelles elle a à faire face (voir l'exemple décrit Figure IV.15).

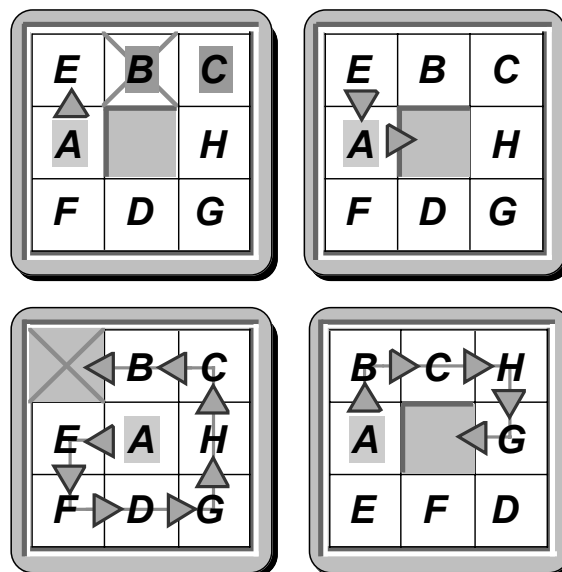


Figure IV.15 - Résolution du problème des coins du taquin. B et C étant satisfaits, A agresse E pour pouvoir se satisfaire à son tour. La contrainte passée en argument est la place de B. E ne trouve aucune place où fuir et donc, après avoir débloqué toutes les cases, agresse A en retour. A fuit sur le blanc. Etant toujours insatisfait, il agresse de nouveau E, mais la contrainte passée est cette fois le blanc, c'est-à-dire le but de A. E agresse alors F, qui agresse D, etc. jusqu'à ce que B fuie sur le blanc et qu'A puisse se déplacer. A agresse B (pas de contrainte cette fois, puisque B n'est plus satisfait), qui agresse C, qui, choisissant de se déplacer sur son but, agresse H, qui, enfin, agresse G qui fuit sur le blanc. Quand tous les mouvements sont effectués, la ligne est résolue.

⁸⁸ Les sous-buts consistant à déplacer chaque palet sur son but ne sont pas sérialisables. Par contre, ces sous-buts peuvent se combiner en sous-buts plus généraux, comme établir une colonne ou une ligne entière, qui, eux, sont sérialisables (voir [Korf 1990a]).

Cette tactique est le fruit de l'interaction entre les différents palets et constitue à ce titre un comportement émergent d'autant plus intéressant qu'il est parfaitement réutilisable dans d'autres contextes de résolution (par exemple, par un joueur humain) sous la forme d'un macro-opérateur [Korf 1990a], du fait, sans doute, de l'univers très contraint et très peu dynamique qu'est le taquin. Nous pouvons même le formuler symboliquement, à l'aide des règles informelles suivantes:

- Placer correctement tous les palets composant une ligne ou une colonne sauf le dernier (que nous appellerons p_n), ce qui est relativement simple à faire.
- Amener p_n sur la case adjacente à son but sans déplacer les précédents, ce qui est aussi très simple.
- Faire glisser ensemble p_n et le palet (p_i) se trouvant sur son but vers le blanc (qui doit normalement être adjacent à p_n).
- Puis, déplacer p_i sur la case qui n'est ni le but de p_n , ni celle où est posé p_n . Ce mouvement doit normalement déplacer p_{n-1} sur le but de p_n .
- Il ne reste plus ensuite qu'à faire glisser p_n vers son but, en repoussant p_{n-1} (et éventuellement p_{n-2} et les palets suivants) sur sa case.

Et nous pouvons certifier, pour l'avoir vérifié de nombreuses fois à la main, que cette heuristique est parfaitement applicable à tous les cas de figure que l'on peut rencontrer lors de la résolution d'un taquin. Elle est certes sous-optimale, mais sa puissance réside dans sa simplicité de mise en oeuvre. Elle permet en outre d'appliquer une décomposition en terme de sous-buts sérialisables, même s'ils ne sont pas identiques.

Nous avons donc l'exemple, ici, d'une structure de coordination d'actions entre agents qui apparaît d'elle-même quand elle est nécessaire pour résoudre un problème particulier. Cette structure n'est générée que par des interactions de compétition entre ces agents, interactions contraintes par un environnement qu'ils contribuent à façonner. Nous nous trouvons donc dans un cas de figure très proche de celui de la chaîne (cf. Chapitre VIII).

XI.4. Complétude et Complexité de la Résolution

XI.4.1. Complétude

La complétude de notre méthode de résolution pour le taquin, quelle que soit la taille choisie, va être prouvée par récurrence sur cette taille. Le chaînage des satisfactions est le même que celui présenté Section XI.2.3, et les lignes et colonnes sont numérotées de 1 à n , en partant de la position finale du blanc. Les premiers palets devant se satisfaire seront donc ceux qui constituent la ligne n .

Théorème 1: Les $n-1$ premiers palets de la ligne n peuvent être correctement placés.

Preuve: Le premier palet P_1 peut évidemment se satisfaire et bloquer sa place sans déranger aucun palet précédemment satisfait. Si nous admettons donc que P_i ($1 < i < n-1$) se satisfait sans déplacer les $i-1$ précédents palets, P_{i+1} peut-il en faire de même? Un palet satisfait P_j ne peut être déplacé que si et seulement si un palet en fuite P_f l'agresse après avoir débloqué sa place. En effet, un palet qui cherche à se satisfaire n'attaquera jamais une place bloquée pour la simple raison que le gradient en provenance de son but la contournera. Si donc un palet en fuite pouvait agresser P_j , cela signifierait que P_j obstrue le chemin vers le blanc et qu'il n'existe pas de chemin alternatif (pas d'autres places atteignables par le

gradient en provenance du blanc). Or, si l'on se fie à la manière dont les palets prennent la décision d'attaquer, P_f n'a pu être agressive que *parce qu'il existait un chemin libre entre sa place et le blanc*, sans quoi un autre palet aurait été choisi. Il est donc impossible que P_f agresse P_j , quel que soit j compris entre 1 et $i-1$. Nous avons ainsi montré que, si i ($1 < i < n-1$) palets composant la ligne n étaient satisfaits, le $(i+1)$ ème pouvait se satisfaire sans les déplacer. Nous avons également montré que P_1 pouvait se satisfaire. La preuve du Théorème 1 par simple récurrence sur la valeur de i est donc évidente.

Théorème 2: Quand le n ème palet de la ligne n (P_n) tente de se satisfaire, il peut détruire certaines des satisfactions précédentes, mais les remet en place en atteignant son but.

Preuve: En arrivant près de son but, P_n peut se trouver dans trois situations distinctes:

- Son but est vide et il en est adjacent. Il peut donc se déplacer dessus sans remettre en cause les satisfactions précédentes.
- Son but est adjacent à la place qu'il occupe mais n'est pas libre. C'est le cas décrit Section XI.3.2. Certaines satisfactions sont détruites, mais, comme il est facile de le voir, le jeu des contraintes passées en argument des agressions fait que P_{n-1} fuira obligatoirement sur le but de P_n (éventuellement suivi par d'autres palets satisfaits). Quand celui-ci l'attaquera pour se satisfaire, P_{n-1} fuira sur son propre but (et les autres palets également, puisque la fuite sur le but est prioritaire), et les satisfactions précédentes seront donc rétablies.
- Son but est vide, mais il y a un palet P_k entre lui et P_n . Ceci correspond tout simplement à l'une des étapes de la situation précédente.

Nous sommes donc capables de placer correctement P_n quand les $n-1$ premiers palets de la ligne n ont été satisfaits, ce qui signifie que le système peut entièrement résoudre une ligne. Le Théorème 2 est donc prouvé. La même démonstration peut bien entendu être appliquée à la colonne n sans être modifiée. Comme nous l'avons déjà dit Section XI.2.3, une fois que la ligne et la colonne périphériques (par rapport à la position finale du blanc) ont été résolues, le reste du taquin peut être résolu sans les déranger. Puisque nous avons prouvé que nous savions toujours résoudre la ligne et la colonne périphériques, le système peut donc réduire la résolution d'un taquin de taille $n \times n$ à celle d'un taquin de taille $(n-1) \times (n-1)$, quel que soit $n \geq 2$. Le taquin de taille 1×1 est quant à lui aisément résolvable, puisqu'il n'est composé que d'une place contenant le blanc. Cette place étant, comme toutes les places, toujours satisfaite, la résolution s'arrête dès l'envoi du message `trySatisfactionReplyTo`: à la place. Ainsi, en conclusion, si nous appelons T_n un taquin de taille $n \times n$ et T_{n-1} le taquin de taille $(n-1) \times (n-1)$ obtenu après avoir retiré la ligne et la colonne périphériques de T_n , nous avons:

$$\left[\begin{array}{l} \forall n > 1, \text{résolu}(T_n) \Leftrightarrow \text{résolu}(T_{n-1}) \\ \text{résolu}(T_1) \end{array} \right. \Rightarrow \forall n, \text{résolu}(T_n) \quad (19)$$

Ce qui constitue bien la preuve de la complétude de notre méthode de résolution. Une des conséquences de cette preuve est que le comportement des agents n'a pas à être adapté à la taille du taquin dans lequel ils agissent, puisque ce comportement permet de déduire la complétude quelque soit n .

XI.4.2. Complexité

Les trois sections suivantes vont nous permettre de calculer la complexité de notre méthode de résolution en fonction de la taille du taquin, c'est-à-dire du nombre N de palets en faisant partie (à distinguer du nombre n représentant la largeur du taquin, la relation entre les deux étant: $N = n^2 - 1$). Nous nous attacherons tout d'abord à calculer la complexité spatiale $SP(N)$, qui, si elle est cruciale pour les méthodes de recherche, apparaît pour nous comme peu significative. Nous bornerons ensuite la longueur de la solution $LG(N)$ qui représente le nombre de mouvements à effectuer pour résoudre le Taquin, puis le temps d'exécution (la complexité algorithmique), qui sera noté $ET(N)$.

XI.4.2.1. COMPLEXITE SPATIALE

Notre méthode de résolution est ainsi faite qu'elle ne conserve aucun état du Taquin en mémoire, mis à part l'état courant. L'espace requis est donc une fonction linéaire du nombre d'agents, lui-même fonction linéaire de la taille du taquin. Un taquin de taille N fédère en effet N palets et $N + 1$ places. La complexité spatiale est donc:

$$SP(N) \sim O(N) \tag{20}$$

Comme nous l'avons indiqué Section XI.2.3., nous ordonnons les satisfactions des palets de manière à successivement résoudre des lignes et des colonnes entières. La longueur de la solution ainsi que la complexité algorithmique vont donc pouvoir être calculées récursivement. En effet, la complexité nécessaire à la résolution d'un taquin de largeur n , que nous appellerons $C(n)$ sera égale à la complexité nécessaire à la résolution de la ligne et de la colonne n , que nous appellerons $LC(n)$, plus celle nécessaire à la résolution du taquin de largeur $n-1$ résultant, c'est-à-dire $C(n-1)$. $C(n)$ pourra donc être calculé par la formule:

Comme $C(n) = LC(n) + C(n-1) = LC(n) + LC(n-1) + C(n-2) = \dots = LC(n) + \dots + LC(1)$.
 Nous obtenons: $C(n) = \sum_{i=1}^n LC(i)$ (21)

XI.4.2.2. LONGUEUR DE LA SOLUTION

Dans cette section, $LCM(n)$ représente donc le nombre de mouvements requis pour placer correctement la ligne n et la colonne n du taquin. Dans un taquin de taille $n \times n$, l'éloignement maximal d'un palet à son but est $2n-1$ places. Le premier mouvement que ce palet effectuera pour se satisfaire générera donc au maximum $2n - 1$ mouvements, c'est-à-dire son propre mouvement plus le mouvement de fuite de $2n - 2$ palets (ce qui correspond au cas où le blanc est positionné sur son but).

Après ce mouvement, le blanc sera forcément adjacent au palet se satisfaisant, puisque celui-ci se sera déplacé. Parcourir les $2n - 2$ places qui le séparent de son but nécessitera donc au maximum 5 mouvements par déplacement (voir par exemple la Figure IV.15). Enfin, nous pouvons borner le nombre de mouvements requis pour effectuer le dernier mouvement du dernier palet de la ligne (ou de la colonne) par $2n^2 - 2$, ce qui correspond au pire des cas, c'est-à-dire quand tous les palets du taquin doivent se déplacer deux fois chacun, une fois pour fuir, une fois pour se repositionner.

Le nombre de palets constituant cette ligne et cette colonne qui ne sont pas positionnés dans un coin est égal à $2n-1-2$, soit $2n-3$. Et, d'après ce que nous venons de dire, le nombre de mouvements que chacun requiert pour se satisfaire est $(2n-1)+ 5 (2n-2)$, alors que celui requis par les coins est $(2n-1)+ 5 (2n-2) + 2n^2 - 2$. Placer correctement l'ensemble de ces palets va donc nécessiter:

$$\begin{aligned} \text{LCM}(n) &= (2n-3) (2n - 1 + 5 (2n - 2)) + 2 ((2n-1)+ 5 (2n-2) + 2n^2 - 2) = 26n^2 - 34n + 7 \\ \text{d}'où nous déduisons immédiatement que } d^\circ(\text{LCM}(n)) &= 2 \end{aligned} \quad (22)$$

Comme $LG(n) = \sum_{i=1}^n LCM(i)$, nous en déduisons également que $d^\circ(LG(n)) = 3$.

La complexité du Taquin est usuellement exprimée en fonction du nombre de palets N (la Taquin s'appelle en effet N-Puzzle en anglais). Comme $N = n^2 - 1$, nous obtenons donc que la longueur de la solution, au pire, est:

$$LG(N) \sim O(N\sqrt{N}) \quad (23)$$

Le point important est que cette complexité est égale à la complexité de la longueur de la solution optimale, calculée de manière empirique par [Korf 1990c].

XI.4.2.3. COMPLEXITE ALGORITHMIQUE

Nous allons tout d'abord calculer $LCT(n)$, qui représente le temps d'exécution requis pour placer correctement la ligne n et la colonne n . Pour ce faire, nous décomposons chaque mouvement en ses quatre composantes: calculer la propagation à partir du blanc, puis à partir du but, prendre une décision et se déplacer physiquement (*i.e.* mettre à jour ses accointances). Le temps d'exécution des deux premières composantes est au pire $K_1 * (n^2 - 1)$, où K_1 représente le temps constant que prend le comportement de propagation de chaque place, et $n^2 - 1$ le nombre maximal de places qui l'effectuent, dans le cas où la propagation a lieu sur l'ensemble du taquin. La prise de décision, ainsi que le déplacement physique ne dépendant que des mêmes connaissances locales, quelle que soit la situation du palet par ailleurs, ils requièrent un temps constant K_2 . En conséquence, le temps consommé par un mouvement, qu'il soit de fuite ou de satisfaction est: $TM(n) = 2K_1*(n^2- 1) + K_2$. Puisque nous connaissons le nombre de mouvements que nécessite la résolution de la ligne n et de la colonne n , nous pouvons donc en déduire son temps d'exécution par la relation suivante:

$$\begin{aligned} LCT(n) &= LCM(n) * TM(n) \\ \text{ce qui implique } d^\circ(LCT(n)) &= d^\circ(LCM(n)) * d^\circ(TM(n)) = 4 \end{aligned} \quad (24)$$

Comme le temps d'exécution global $ET(n) = \sum_{i=1}^n LCT(i)$, nous pouvons ainsi en déduire que $d^\circ(ET(n)) = 5$.

En ramenant ceci à N , nous obtenons donc:

$$ET(N) \sim O(N^2\sqrt{N}) \quad (25)$$

Ce qui constitue une complexité algorithmique tout à fait honorable.

XI.5. Résultats expérimentaux

Le premier point fondamental que nous pouvons souligner à propos des résultats expérimentaux est qu'il a été possible, grâce à notre méthode de résolution, de résoudre des taquins de taille jusque-là jamais atteinte par les méthodes plus classiques. La plupart des résultats présentés ici sont donc inédits, puisqu'ils partent du taquin de largeur 3 pour arriver jusqu'à des taquins de largeur 30, alors que la largeur maximale atteinte par l'algorithme LRTA* [Korf 1990b] est de 10 (mais aucun résultat expérimental n'est fourni). Le Tableau IV.2 présente ces résultats par ordre croissant de largeur, en indiquant la moyenne du temps et de la longueur de résolution obtenus sur 100 instances aléatoires de même largeur. Le temps est indiqué en secondes, le système étant programmé sous Smalltalk 80 4.1, sur une station de travail Sun Sparc 10.

Configuration		Solution		
Largeur	Palets	Temps moyen	Longueur Moyenne	Longueur par Palet
<i>n</i>	<i>N</i>	<i>Secondes</i>	<i>Nombre de Mouvements</i>	
3	8	0,2218	48,90	6,11
4	15	0,7547	142,10	9,47
5	24	2,4812	283,00	11,79
6	35	4,5146	482,00	13,77
7	48	9,6559	746,30	15,55
8	63	17,027	1 109,20	17,61
9	80	23,1532	1 455,20	18,19
10	99	35,4682	1 947,70	19,67
11	120	52,7616	2 483,10	20,69
12	143	65,7846	2 889,10	20,20
13	168	86,7198	3 361,90	20,01
14	195	106,9209	4 199,70	21,54
15	224	116,4785	4 881,50	21,79
16	255	165,8702	5 581,90	21,89
17	288	204,4582	6 267,10	21,76
18	323	230,687	6 947,60	21,51
19	360	271,5735	7 797,20	21,66
20	399	292,0156	9 004,70	22,57
21	440	340,8768	9 882,00	22,46
22	483	415,1151	11 093,70	22,97
23	528	477,3549	12 404,00	23,49
24	575	544,9122	13 340,10	23,20
25	624	625,5805	15 181,20	24,33
26	675	655,0309	16 514,00	24,47
27	728	782,0069	18 496,80	25,41
28	783	897,6368	19 999,40	25,54
29	840	1036,3034	21 229,20	25,27
30	899	1179,58	23 009,00	25,59

Tableau IV.2 - Résultats expérimentaux de la résolution de 2800 taquins répartis à raison de 100 par largeur.

La seconde constatation que nous pouvons faire est que la résolution du taquin n'est pas optimale, puisque la longueur des solutions des taquins de largeur 3 et 4 (48,90 et 142,10 mouvements en moyenne) sont entre deux et trois fois plus importantes que la longueur de la solution optimale qui a pu être calculée par A* (respectivement 22 et 53 mouvements). Il va sans dire que nous ne cherchions pas cette optimalité (voir Chapitre VI). Les résultats sont néanmoins encourageants puisqu'ils se situent dans la fourchette de RTA* et LRTA* [Korf 1990b]. Pour les taquins de largeur plus importante, pour lesquels n'existent à l'heure actuelle aucune technique permettant de trouver la solution optimale, nous ne pouvons nous en remettre qu'à notre intuition et au fait que nous les résolvons en un temps raisonnable (largement inférieur à celui requis par un être humain), alors que le langage utilisé est

interprété (et non compilé) et que les Sparc 10 ne sont pas des super-calculateurs. Nous subodorons ainsi que nous restons dans un facteur multiplicatif compris entre deux et trois par rapport à la solution optimale. Mais ceci est bien sûr purement spéculatif.

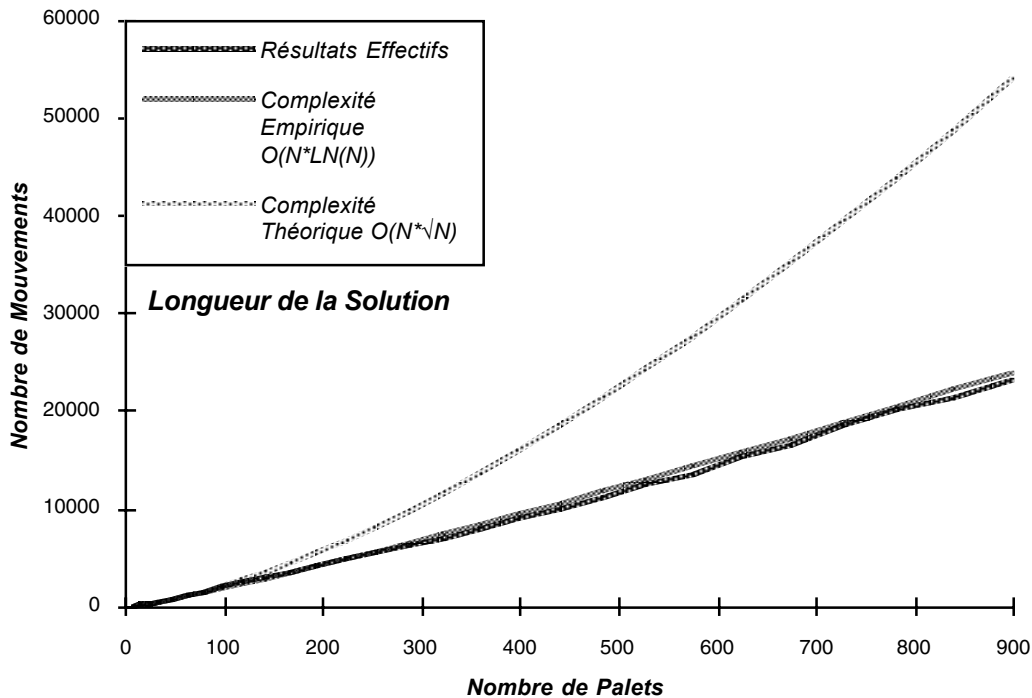


Figure IV.16 - Evolution de la longueur des solutions pour les taquins de largeur 3 à 30. Comparaison avec la complexité théorique.

La troisième constatation est que les résultats obtenus, aussi bien du point de vue de la longueur de la solution que de la complexité algorithmique, sont de toute évidence bien meilleurs que ceux déductibles du calcul théorique des complexités. La première raison, relativement évidente, est que les complexités théoriques reflètent uniquement les cas les plus défavorables. La seconde, plus directement liée à nos heuristiques, est que le comportement de fuite fourni aux palets, qui leur fait d'abord choisir leur but s'il est adjacent, et sinon les en fait se rapprocher, réduit progressivement le désordre du taquin pendant la résolution, alors même que les palets en fuite ne se satisfont pas. Ainsi, quand la première ligne et la première colonne ont été complétées, le taquin qui reste à résoudre est beaucoup plus ordonné qu'une instance aléatoire de la même largeur, toutes les fuites générées par les palets qui se sont satisfaits ayant en effet réduit la distance moyenne des palets non-satisfaits à leur but. Malheureusement, il nous a été difficile, sinon impossible, de prendre en compte cette propriété dans le calcul de la complexité théorique.

La troisième et dernière raison, est que la propagation des gradients s'effectue rarement sur l'ensemble des places du taquin, puisqu'elle est arrêtée par les cases bloquées. Dans la plupart des cas, sa complexité est donc bien inférieure à n^2 .

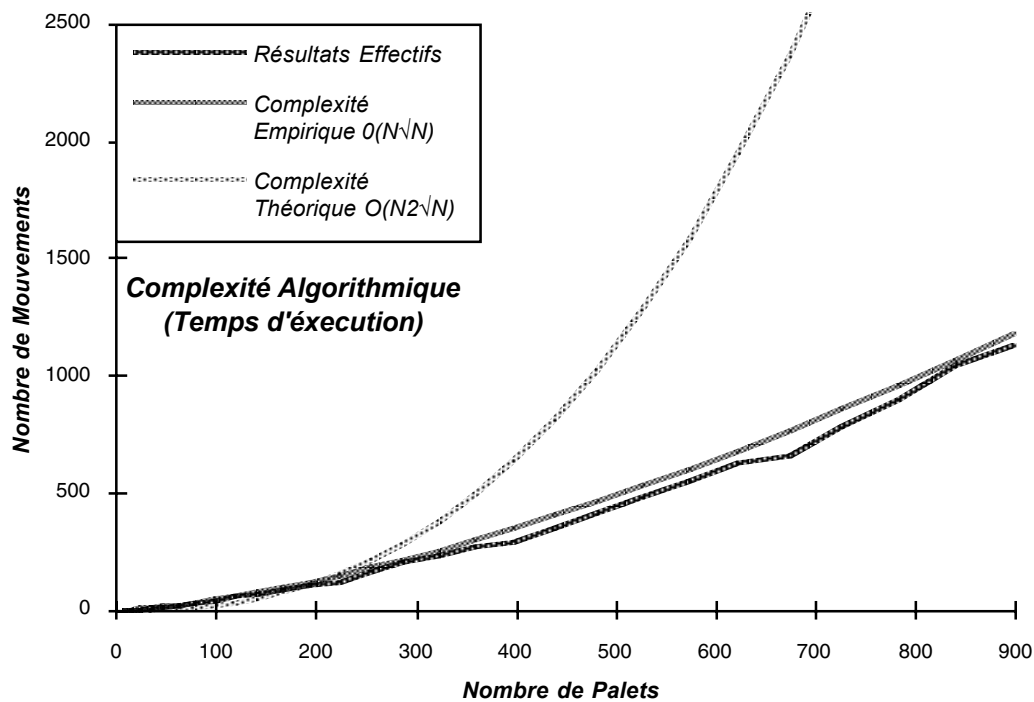


Figure IV.17 - Evolution du temps d'exécution requis pour des taquins de largeur 3 à 30. Comparaison avec la complexité algorithmique théorique.

Comme nous pouvons le constater sur la Figure IV.16, la courbe qui représente les résultats obtenus pour la longueur de la solution diverge en effet assez rapidement de la courbe représentant la complexité théorique. Nous pouvons en déduire une complexité empirique proche de $O(N \cdot \log(N))$. Ce résultat doit bien sûr être validé par de plus amples expériences, puisque nous n'avons ici que 28 points, et qu'il nous en faudrait sans doute le double pour pouvoir calculer plus exactement cette complexité empirique. Il est en effet tout à fait possible que le rapport entre la complexité théorique et la longueur de nos solutions soit constant à partir d'un certain rang. Nous obtenons des résultats similaires en ce qui concerne la complexité algorithmique, dont l'évolution est montrée Figure IV.17. Contre une complexité attendue en $O(N^2 \cdot \sqrt{N})$, les résultats empiriques laissent entrevoir la possibilité d'une complexité en $O(N \cdot \sqrt{N})$. Pour les mêmes raisons que précédemment, cependant, il nous faudra valider ce résultat par de plus amples expériences.

XI.6. Conclusion

La programmation du Taquin montre qu'une décomposition du problème en agents réactifs, dotés d'heuristiques locales plutôt que globales, permet d'obtenir une résolution qui n'est pas optimale, mais dont la longueur et la complexité algorithmique sont dans un rapport constant avec la longueur et la complexité de la solution optimale. C'est là un résultat intéressant, dans la mesure où il permet d'envisager le développement de cette technique à d'autres problèmes fortement combinatoires, comme le travail qui est présenté dans [Ghedira et Verfaillie 1991], qui combine une décomposition en agents réactifs et des techniques de recuit simulé distribué pour optimiser la résolution de problèmes d'ordonnement.

Cependant, la difficulté d'application de méthodes décentralisées à ce type de problèmes provient généralement de l'absence de méthodologie permettant d'effectuer une décomposition correcte du problème en agents. En réalité, cette absence n'est sensible qu'au niveau des comportements. La méthodologie que nous proposons pour décomposer le Taquin en se basant sur la notion de sous-buts (sérialisables ou non), et de réification (ou devrait-on dire "agentification" ?) de ces sous-buts sous la forme d'agents est en effet, dans l'esprit tout au moins, proche de nombreuses méthodes de conception orientée-objet. Une décomposition objets du Taquin aurait donc de fortes chances d'être similaire à celle que nous proposons.

Les comportements sont sans doute plus difficiles à déterminer, parce qu'ils apparaissent comme moins "intuitifs". Cependant, il ne faut pas négliger dans cette difficulté le poids culturel de la vision centralisée qui provient en droite lignée de la recherche heuristique. L'approche réactive, comme en son temps l'approche orientée-objets, nécessite un complet renversement de perspective de la part du concepteur, qui doit abandonner les notions de contrôle, de mémoire et d'état globaux au profit de celles de coordinations, perceptions et structures locales. Se mettre à la place d'un palet pour déterminer son comportement local est à la portée de tout concepteur, mais encore faut-il en avoir l'idée; et c'est cette démarche qui semble la plus difficile à effectuer.

Un des intérêts fondamentaux de l'application d'un système multi-agents réactifs à un problème comme le Taquin réside dans la capacité de cette population à s'adapter à de nombreuses configurations différentes de la configuration classique. Ceci a été particulièrement bien montré par [Dubreuil 1990] , qui a étudié des versions "multi-blancs", ainsi que des versions rectangulaires du Taquin, et qui a montré que les comportements des agents n'avaient pas besoin d'être modifiés pour que la population d'agents puisse résoudre le problème. La difficulté que constitue la conception des comportements individuels est donc à notre avis largement contrebalancée par l'importante généricité de ces comportements.



CONCLUSION DE LA QUATRIEME PARTIE

QUELLES LIMITES ? ILLUSTRATION PAR LE JEU D'ECHECS

1. De un à deux Joueurs

Les limites que peut rencontrer l'approche réactive de résolution distribuée de problèmes, que ce soit celle présentée dans la Troisième Partie ou l'Eco-Résolution, ne sont pas aisées à découvrir. Une part importante du travail de recherche consiste en effet à l'heure actuelle en travaux empiriques qui, s'ils sont couronnés de succès, permettent de mettre en valeur l'intérêt de la réactivité, mais qui, en cas d'échec, ne peuvent tout au plus qu'invalider l'instanciation particulière dont ils sont issus et non la démarche elle-même. Nous avons de plus montré qu'il était possible d'appréhender sous une forme réactive aussi bien des problèmes naturellement distribués (les robots fourrageurs) que des problèmes habituellement traités de manière centralisée (le Taquin). Les limitations éventuelles ne doivent donc pas être cherchées du côté de la difficulté de distribution du problème à résoudre, qui reste une notion éminemment subjective, et de plus battue en brèche par l'extension rapide des méthodologies de décomposition orientée-objet.

De l'analogie que nous faisons en considérant un problème comme un éco-système dans lequel évoluent des populations d'agents, nous pouvons cependant retenir une notion importante. Le succès d'une population végétale ou animale (donc, la résolution du problème que constitue sa survie) n'est vraisemblablement possible dans un éco-système donné auquel elle est censée pouvoir s'adapter que si (1) elle n'entre pas en compétition directe avec une autre espèce, si (2) cette compétition s'effectue à armes égales, ou enfin si (3) l'inégalité des armes est en sa faveur. Qu'entendons nous par "armes" ? De manière intuitive, c'est l'ensemble des moyens et des informations mis à la disposition de la population pour assurer son succès. Ou, plus précisément, c'est l'ensemble des informations qu'elle est capable de collecter dans l'environnement, couplé à l'ensemble des comportements qu'elle est capable d'effectuer en réponse à ces informations. Si nous prenons par exemple le cas des fourmis et des termites (exposé dans [Wilson 1992]), nous pouvons nous apercevoir que le fait qu'aucune de ces deux populations d'insectes sociaux n'ait réellement gagné dans la compétition immémoriale qui les oppose (en tout cas dans les biotopes qui leur sont à toutes deux accessibles) est explicable par l'identité à la fois des stratégies utilisées, des ressources consommées par ces stratégies et des informations nécessaires à leur mise en oeuvre. Les deux espèces usent en effet toutes deux d'une collecte d'informations et d'une coordination décentralisés, et la stratégie résultante est le fruit d'un ensemble de tactiques individuelles. Aucune des deux n'est donc en mesure de manipuler d'informations ni de générer de stratégies globales.

Les populations d'agents que nous avons utilisées pour résoudre certains problèmes dans les applications précédentes n'étaient confrontés à aucune compétition. Seul PENGUIN devait faire face à celle des abeilles, mais il est aisé de voir que les armes des abeilles sont bien moins efficaces que celles du

pingouin, essentiellement en raison du caractère semi-aléatoire de leur comportement offensif⁸⁹. Ni les fourmis, ni les robots et encore moins les palets du Taquin ne devaient lutter contre autre chose que leurs propres difficultés d'adaptation à l'environnement qui leur était fourni. Ces problèmes étaient donc assimilables à des jeux à un joueur (la population d'agents étant manipulée par ce joueur virtuel).

Il est probable qu'une des limitations des systèmes réactifs réside dans leur confrontation avec des systèmes cognitifs (au sens large: humains ou agents) aptes à réagir aussi rapidement qu'eux et de manière plus rationnelle dans des jeux à deux joueurs. Il existe cependant des parades, et l'une d'entre elles, utilisée par exemple par les fourmis dans la lutte qui les oppose à l'homme, ou à toute autre espèce animale plus cognitive (voir par exemple [Chauvin 1979]), est celle de la prolifération. Le manque de stratégie coordonnée est alors compensé par un dynamisme démographique bien plus important, ce qui permet d'assurer un équilibre entre pertes et gains, et explique que les fourmis (et les termites) fassent toujours partie de notre environnement. Mais, dans un système artificiel dont les règles ne sont pas forcément aussi souples, cette stratégie n'est pas toujours applicable. La deuxième limitation, dans le cas des jeux à deux joueurs, réside dans l'aspect plus ou moins aléatoire des contraintes environnementales et des comportements de l'adversaire. La faiblesse des systèmes rationnels qui planifient leurs actions réside justement dans l'importance des processus aléatoires pouvant les obliger à réviser leurs plans. Et il ne fait pas de doute que l'adaptativité que leur confère leur mode de fonctionnement assure aux agents réactifs une plus grande souplesse face à ces mêmes phénomènes. Moins les perturbations qu'ils subissent sont aléatoires, moins ils seront efficaces et à même de s'adapter. Il y donc fort à parier qu'une population d'agents réactifs soumise à des perturbations organisées et non capable de les contre-balancer par un accroissement de son effectif ne pourra correctement s'adapter à son environnement, quelque soit par ailleurs le système responsable de ces perturbations.

2. Un Joueur d'Echecs Réactif

2.1 Introduction

On en prendra comme exemple la programmation réactive d'un système joueur d'échecs que nous avons réalisée sous l'acronyme MARCH (Multi-Agent Reactive CHess). Ce système en est à ses premiers pas, et c'est pourquoi nous ne le présentons que dans cette conclusion (voir également [Drogoul 1993], article dans lequel nous replaçons MARCH dans la perspective des différentes applications présentées dans ce travail, de MANTA à PENGU). Le jeu d'échecs est l'exemple parfait d'application dans laquelle il semble absolument nécessaire d'utiliser une stratégie globale pour pouvoir gagner. Et, à part quelques opinions divergentes (comme celle de [Selfridge 1956], citée dans [Brooks 1991]), cette option a été majoritairement suivie par l'ensemble des recherches menées sur la programmation de joueurs d'échecs artificiels.

L'intérêt que représente la programmation d'un jeu d'échecs réactif est indéniable. Le fait de savoir bien jouer à ce jeu est à ce point confondu avec la marque certaine d'une intelligence qu'il est très tentant de voir s'il n'est pas possible de générer un tel comportement par de simples coordinations réactives, en un mot, de générer une stratégie efficace à partir de comportements tactiques individuels. Bien entendu, l'objectif de MARCH, du moins dans la version que nous avons élaborée, n'est pas de devenir un grand-maître, mais au moins de savoir jouer décemment et de pouvoir réagir correctement aux multiples situations qui peuvent se présenter sur l'échiquier. Un des objectifs principaux de cette implémentation

⁸⁹ C'est également le cas des prédateurs dans l'application proies-prédateurs que nous n'avons pas décrite ici, et dont les résultats expérimentaux se trouvent dans [Drogoul en prép.].

était en outre de programmer le système le plus simple possible, ce que nous avons fait en gardant à l'esprit qu'il ne s'agit que d'une implémentation particulière et que d'autres solutions peuvent tout à fait être envisagées.

2.2 Présentation de March

Nous ne ferons ici qu'une présentation succincte du programme, à la fois pour des raisons de place et parce que ce n'est pas le but de cette section. De la même manière que pour les autres applications présentées, nous considérons dans MARCH le jeu d'échecs comme un éco-système particulier. L'environnement est constituée de l'ensemble des cases, et les agents actifs sont les pièces. Chaque agent possède son propre comportement, en partie assigné par les règles du jeu (par exemple, la manière et les possibilités de se déplacer sur l'échiquier), et en partie dicté par la perception qu'il a de son environnement. Cette perception est bien entendu limitée aux cases sur lesquelles il est susceptible de se déplacer.

Le principe de cette implémentation est basé (ce qui n'est pas en soi une nouveauté) sur la notion d'*influence* globale comme produit d'un ensemble d'influences locales. Comment est calculée cette influence ? Chaque pièce possède une valeur d'agression et une valeur matérielle. La valeur d'agression représente l'intérêt qu'aurait cette pièce à attaquer, indépendamment de toute situation particulière. Elle est ainsi élevée pour les pions, et faible pour la reine et le roi. La valeur matérielle est identique à celle utilisée par les méthodes d'évaluation des programmes classiques (nous nous sommes basés sur celles proposées par le programme GNU Chess, dont les sources sont dans le domaine public), et représente le dommage que causerait la perte de cette pièce à son propre camp. Elle est donc très faible pour les pions, forte pour la reine, et très forte pour le roi.

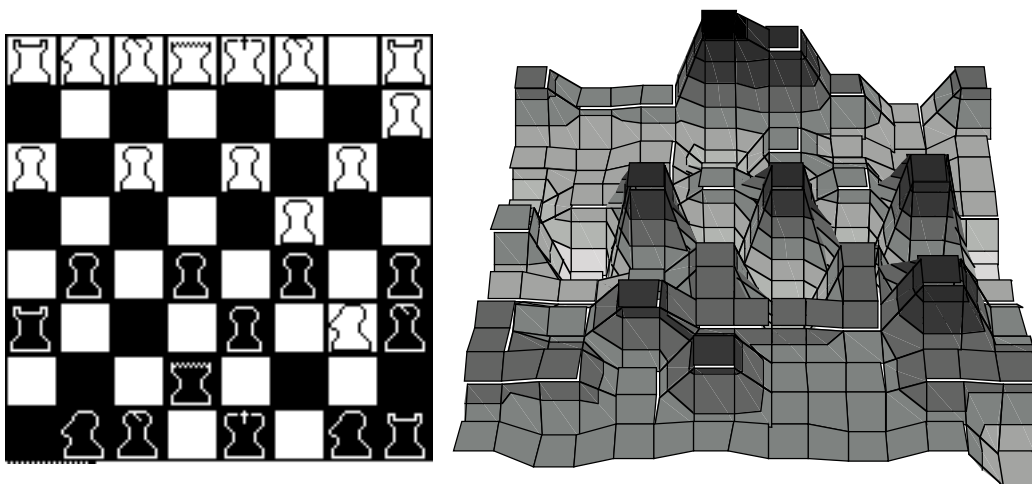


Figure IV.18 - Une situation de l'échiquier et sa perception par le joueur noir. Les niveaux de gris symbolisent l'influence du camp noir sur une case. Une case très foncée devient ainsi très attractive pour les pièces noires.

C'est à partir de ces deux valeurs que va se calculer l'influence globale respective des deux camps en présence. Chaque case est pourvue de deux champs (`#black` et `#white`) qui vont représenter l'influence que possèdent les pièces noires et les pièces blanches sur elle. Toutes les pièces vont propager leur valeur d'agression sur les cases qu'elles peuvent menacer, et cette valeur va être additionnée au champ correspondant à leur couleur. Elles vont ensuite soustraire leur valeur matérielle

de la case où elles se trouvent. De cette manière, case par case, le système construit une topographie globale de l'échiquier en fonction des influences respectives du camp noir et du camp blanc (voir Figure IV.18). Et la perception par les pièces de l'influence exercée sur les cases potentielles où elles peuvent se déplacer leur permet d'orienter leur décision de mouvement. Toute pièce sera ainsi encline à se déplacer vers les cases contenant la plus grande différence entre l'influence de son camp et l'influence du camp adverse.

Bien entendu, chaque pièce effectue son choix en fonction de la situation et de son type. A tour de rôle, elles vont donc attribuer une note aux cases sur lesquelles elles peuvent se déplacer, note qui tient compte de la différence évoquée, et de certaines particularités qui leur sont propres. Ainsi, par exemple, le roi attribue systématiquement une mauvaise note aux cases menacées par une pièce du camp adverse⁹⁰.

Comme les règles des échecs exigent qu'une et une seule pièce de chaque camp se déplace à tour de rôle, le choix de la pièce active est effectué de manière centralisée. La pièce choisie est celle dont la différence entre la note qu'elle attribue à sa position courante et la note maximale qu'elle a attribuée aux cases sur lesquelles elle peut se déplacer est la plus grande. En cas d'égalité entre pièces, la pièce se trouvant sur la case créditée de la note la plus faible est sélectionnée.

2.3 Analyse et Discussion

Malgré son extrême simplicité et son orientation purement tactique, ce système arrive à *jouer* aux échecs. Il le fait certes d'une manière plus que moyenne, et les premiers résultats expérimentaux sont là pour le prouver (voir Tableau IV.3). Mais l'obtention de 57 victoires (sur un total de 200 parties) face à un joueur humain plutôt mauvais est un signe encourageant pour la suite de ce projet.

	Victoires	Défaites	Nuls	Total
Joueur (très) moyen	57	83	60	200
GNU Chess	0	47	3	50

Tableau IV.3 - Résultats expérimentaux de la version actuelle de MARCH. Le joueur (très) moyen en question est bien entendu nous-même.

En observant le système jouer, nous pouvons nous rendre compte de ses deux défauts les plus importants. Le premier est son incapacité totale à effectuer une ouverture correcte, ce qui l'handicape grandement dans la suite de la partie. Le second est sa propension à effectuer l'échange systématique de pièces dès qu'il en a la possibilité. Ces deux défauts sont assez facilement interprétables au regard de ce qui a été décrit de son fonctionnement. Les seules informations globales que peuvent interpréter les pièces sont en effet celles concernant les influences réciproques de chaque camp sur l'échiquier. Or, les situations d'ouverture sont pour elles des situations totalement opaques, dans la mesure où aucune information concernant le camp adverse n'est disponible. Les choix de déplacement sont donc le plus souvent pris de manière aléatoire, ce qui est rarement la meilleure manière de commencer une partie. Il est à cet égard éclairant de constater l'amélioration du comportement du système dans des situations de défense, quand la quantité d'information est beaucoup plus importante. Deuxièmement, aucune

⁹⁰ La plupart des détails d'implémentation sont répertoriés dans [Drogoul 1993]. Nous ne nous étendons donc pas sur le calcul individuel des notes, qui fait également intervenir les menaces potentielles à l'encontre d'une pièce adverse et la protection des pièces amies, ainsi que quelques autres facteurs heuristiques.

pièce ne s'insérant dans une stratégie pré-établie, elles n'ont pas conscience de leur importance dans la suite de la partie et n'hésitent donc pas à s'échanger systématiquement avec les pièces adverses.

Si nous nous référons aux résultats obtenus contre le programme GNU Chess, il est ainsi possible de voir qu'un système comme MARCH qui repose sur des prises de décision individuelles à court-terme dépendant de perceptions locales n'a pratiquement aucune chance face à un système prenant des décisions à partir de prédictions à long-terme portant sur la globalité de l'échiquier. Les perturbations subies par le camp réactif ne sont en effet absolument pas aléatoires, mais au contraire parfaitement organisées, avec pour effet pour effet de le désorganiser complètement. Ceci est évidemment rendu possible par la capacité qu'a le système adverse d'effectuer des prédictions, ce qui n'est pas toujours le cas. Il serait ainsi intéressant d'étudier le comportement de MARCH au cours de parties d'échecs comportant des évènements aléatoires (déplacement, disparition, apparition de pièces, etc...)⁹¹.

Ainsi que nous le disions en ouverture de cette section, il n'est pas possible de tirer des leçons d'ordre général d'une implémentation particulière. Beaucoup de travail reste à accomplir avant de pouvoir, donc, estimer si nous avons atteint les limites des systèmes réactifs de résolution de problèmes avec MARCH. Si tel était le cas, cependant, que pourrions nous en déduire ? En fait, peu de choses. Les échecs, tout comme la plupart des jeux de stratégie à deux joueurs qui ne laissent aucune place au hasard (*i.e.* le jeu de Go, le jeu de Dames, etc.) modélisent des situations purement intellectuelles. Aucune bataille ne s'est jamais déroulée avec la précision d'une partie d'échecs et la majorité des problèmes réels que nous souhaitons résoudre par le biais de l'Intelligence Artificielle comportent tant de paramètres qu'il serait sans doute illusoire de vouloir tous les maîtriser⁹². Face à l'incertitude qui en découle, l'usage de la réactivité permet d'employer des stratégies distribuées, donc robustes, et adaptatives, donc peu sensibles à cette incertitude, plutôt que centralisées, donc fragiles, et prédictives, donc très sensibles à cette incertitude. Nous faisons ici l'hypothèse, et l'avenir seul dira si nous nous sommes trompés, que ces qualités feront la force des systèmes artificiels de demain, qui devront appréhender des problèmes du monde réel. Les fourmis ne jouent pas aux échecs, mais cela ne les a pas empêché de conquérir le monde. Les systèmes réactifs ont certainement encore un long chemin à parcourir avant de pouvoir être réellement *utilisés*, mais il ne fait pas de doute pour nous qu'ils le parcoureront.



⁹¹ Ce test serait équivalent à celui du *misachieving baby* du Monde des Cubes, dans lequel un bébé malfaisant déplace aléatoirement des cubes au cours d'une résolution (voir, par exemple, [Bura et al. 1992]).

⁹² Il suffit, pour s'en persuader, de constater l'incapacité des planificateurs à modéliser l'ensemble des états possibles des Taquins de taille importante, alors même que ce problème ne comporte aucune sorte d'incertitude.

Conclusion Générale

Bilan et Perspectives de l'Approche Réactive

"Aussi, (...), Joe avait-il choisi une petite fourmilière et s'était-il fixé la mission de jouer le dieu des fourmis, de changer le cours de leur destinée. (...) Et cela avait réussi. Les fourmis avaient progressé. Elles avaient fabriqué de minuscules chariots, fait fondre les minerais. [Et] ce qu'elles pouvaient faire d'autres, ce qu'elles apprenaient, au fond de leurs tunnels, nul ne pouvait le savoir. Un jour, il avait (...) labouré la fourmilière d'un coup de talon, puis il s'en était allé, sans plus se soucier de ce qu'il advenait des fourmis.

Mais les fourmis, elles, s'en souciaient."

Clifford D. Simak *Demain les chiens*

Le travail présenté dans la cadre de ce mémoire de thèse n'est pas un aboutissement. Il représente au contraire une ouverture vers un certain nombre de *possibles* dont nous espérons qu'ils constitueront une part non négligeable de la recherche de demain en Vie Artificielle et en Intelligence Artificielle. La clé de cette ouverture passe sans doute par un très important travail expérimental, et, de ce point de vue, l'Intelligence Artificielle Distribuée a tout à gagner à des collaborations interdisciplinaires comme celle que nous avons pu mener sur le projet MANTA, avec des sociologues, des éthologues, des écologistes, des biologistes, des économistes, des systémiciens, bref, avec tous ceux pour qui le mot *organisation* a un sens. S'il ne fallait retenir qu'une chose de cet ouvrage, c'est qu'il nous reste encore beaucoup de choses à apprendre du vivant avant de pouvoir nous lancer dans l'aventure du vivant-artificiel.

Ce qui a été réalisé...

Au terme de ces trois années de recherche, le bilan provisoire que nous pouvons tirer de notre travail est cependant plutôt positif. Nous avons ainsi progressé de manière tout à fait significative à la fois dans le domaine de la simulation multi-agents et dans le domaine de la résolution distribuée de problèmes. Bien que l'interaction entre les deux reste encore très forte dans notre manière de concevoir les systèmes réactifs, chacun de ces domaines a d'une certaine manière gagné son autonomie. On se reportera aux conclusions des Deuxième et Quatrième Parties pour constater que le bilan en termes d'avancées empiriques et conceptuelles est loin d'être négligeable. Sans avoir pour autant eu la possibilité de formaliser notre approche de manière forte, il est également certain qu'un certain nombre de préceptes méthodologiques ont été établis. Un modèle général de comportement a ainsi été introduit pour la conception du noyau d'EthoModélisation, et nous avons vu que ce modèle permet de programmer des agents aux comportements fort différents. Un ensemble de conseils permettant d'orienter le concepteur de systèmes de résolution distribuée de problèmes a également été présenté. Et enfin, des techniques, comme, par exemple, la propagation de signaux dans l'environnement formant à la fois un moyen de communication et un moyen d'orientation ont été généralisées à des applications *a priori* différentes, où elles constituent une alternative intéressante à l'envoi de messages. Ce qui a été fait peut donc se résumer par: l'introduction de techniques basées sur l'émergence et l'auto-organisation fonctionnelle de groupes d'agents dans des domaines habituellement peu sensibles à ces concepts, à savoir la simulation et la résolution de problèmes, et la démonstration, dans cette optique, des possibilités offertes par la modélisation de phénomènes auto-organisés existants en terme de retombées technologiques et conceptuelles.

... Ce qui n'a pas été réalisé...

Nous serions tenté de dire: à peu près tout ! Ce qui serait sans doute un petit peu injuste. Mais il est vrai que tant de choses restent à faire et que tant de concepts nous font encore défaut qu'il est probable que nous n'ayons abordé dans ce travail qu'une parcelle de la partie visible de l'iceberg réactif. En premier lieu, manque une véritable méthodologie de conception de systèmes multi-agents réactifs. Si nous souhaitons que cette approche se généralise, il faudra sans nul doute qu'elle devienne accessible au plus grand nombre, et nous ne pourrons alors éviter cet effort. Bien entendu, cette méthodologie devra

reposer sur les bases d'une formalisation complète des liens micro-macro, et donc de la notion d'émergence, dont l'auto-organisation n'est que l'une des facettes. Aucune formalisation n'a été proposée ici, en partie pour des raisons qui nous sont propres et qui concernent la primauté donnée à une approche *empirique* comme préalable à toute avancée théorique et en partie parce que les travaux menés actuellement de par le monde sur la question de l'émergence de fonctionnalités ne nous offrent aucune base conceptuelle claire. Ce manque de formalisme est clairement l'un des défauts les plus graves de ce travail, et nous en sommes pleinement conscients. Nous restons cependant persuadés qu'il serait illusoire de formaliser dans le vide, c'est-à-dire de ne pas accompagner ou même précéder toute formalisation d'un effort d'expérimentation. Nous pensons avoir fourni des outils et des idées propres à servir de base aux futures travaux qui seront réalisées dans cette optique.

... Ce qui devra être réalisé un jour...

Outre cet aspect formel, que nous aborderons sans doute dans un futur proche, nous souhaitons poursuivre encore quelques temps notre travail de recherche sur les bases expérimentales que nous avons fixées ici. Les deux grands axes en seront bien entendu la simulation multi-agents et la résolution distribuée de problèmes, mais il n'est pas impossible d'envisager dès à présent de transposer un certain nombre de nos résultats dans des applications réelles, ce qui assurerait une crédibilité certaine à l'approche que nous défendons. Les perspectives sont multiples: dans le domaine militaire, où la robotique réactive a sans doute de beaux jours devant elle (*e.g.* surveillance d'entrepôts, infiltration du camp adverse par "saupoudrage" de robots, systèmes armés autonomes, etc.), dans le domaine de la construction, où nous pourrions nous inspirer des comportements de construction et d'ordonnancement utilisés par certains insectes sociaux, dans celui des télécommunications, où commence à poindre la notion d'agent (transporteur de messages, questionneur de bases de données distantes) et où le problème d'adaptation est crucial vu la diversité des réseaux existants à l'heure actuelle, et dans bien d'autres encore. Une autre perspective est bien sûr celle offerte par la simulation multi-agents, qui devrait, en conjonction avec la simulation orientée-objets, investir d'ici peu de temps un grand nombre de domaines naturellement distribués dont les techniques classiques de simulation ne suffisent plus à rendre compte de la dynamique. Il en est ainsi de l'économie (voir Conclusion de la Deuxième Partie), de la biologie bien sûr, où les agents réactifs sont particulièrement bien positionnés, et de l'écologie. D'importantes avancées sont certainement à attendre du côté de ces trois domaines dans les années à venir.

... Et Ce qui ne sera pas réalisé avant longtemps

Enfin, le dernier point qui ne pourra manquer d'être abordé, et qui rejoint ainsi en boucle la dialectique présentée au Chapitre I, est celui du passage du réactif au cognitif, qui est à notre avis un point essentiel en ce qu'il permettra d'une part de considérer de manière unie le problème des organisations, et qu'il permettra sans doute d'explicitier une partie du problème plus vaste qui est le passage du vivant à l'intelligent. Nous avons en effet dans ce travail conçu et observé des systèmes qui pourraient être sans beaucoup de difficultés qualifiés de "globalement intelligents" par des observateurs indépendants. Si nous voulons progresser, il nous faut maintenant d'une certaine manière *réifier* ce que nous avons obtenu, cette "intelligence" qui nous semble encore un peu insaisissable. Quoi de plus naturel en I.A.D. que de vouloir la réifier sous la forme d'agents, et obtenir ainsi des agents cognitifs à partir d'agents réactifs ?

Ce passage est envisageable de plusieurs manières. La première est de progressivement complexifier des agents réactifs pour les doter de capacités cognitives de plus haut niveau, un peu à la manière de ce que nous avons fait pour les robots **Dockers**. Il s'agit là d'une démarche incrémentale qui tend à conserver les propriétés réactives des agents tout en leur donnant des possibilités cognitives d'apprentissage et de contrôle. Cette démarche est actuellement majoritaire parmi les chercheurs travaillant sur le sujet. La seconde manière consiste à obtenir des individus cognitifs comme composition d'agents réactifs, comme si nous réifiions une colonie de fourmis pour en faire un agent à part entière. Il s'agit là d'une démarche uniquement basée sur l'émergence de fonctionnalités, qui, si elle n'est pas majoritaire, commence cependant à se frayer un chemin dans les recherches actuelles. Il existe enfin une troisième démarche médiane, sorte de compromis entre les deux précédentes, basée sur l'émergence de comportements complexes à partir de comportements réactifs en interaction (voir Chapitre X). C'est une démarche incrémentale au sens où les comportements émergents sont réifiés et intégrés au comportement de l'agent, et émergente au sens où c'est l'auto-organisation des comportements entre eux qui permet l'apparition de nouveaux comportements. Aucune de ces trois démarches n'a encore permis d'effectuer le saut conceptuel entre réactif et cognitif, et il est fort probable que ce saut n'est pas envisageable dans l'immédiat.

A la différence des fourmis de Clifford Simak, nos agents ne se soucient pas encore que nous éteignons la machine qui leur sert d'environnement. Leur font encore défaut une véritable autonomie par rapport à leur créateur, et sans doute une capacité de raisonnement suffisante. Et, en définitive, la question à laquelle nous sommes confrontés est bien toujours la même que celle qui habitait les débuts de la Cybernétique ou de l'Intelligence Artificielle: *serons nous un jour capable de créer de toutes pièces des artefacts logiciels ou matériels autonomes et intelligents ?*

Nous souhaitons, par l'écriture de cet ouvrage, avoir progressé, et avoir permis à d'autres chercheurs de progresser ne serait-ce que de quelques pas dans cette direction. Nous souhaitons également que le point de vue défendue dans cette thèse, à savoir l'intelligence comme produit de l'interaction entre agents élémentaires et comme propriété émergente du groupe qu'ils forment, amènent de nombreux lecteurs à explorer plus avant dans cette voie, très certainement riche en surprises et découvertes de toutes sortes. C'était là toute l'ambition de ce travail, et nous espérons qu'il aura été perçu comme tel.



Bibliographie

- [Agre et Chapman 1987] AGRE P. E. ET CHAPMAN D., *Pengi: an Implementation of a Theory of Activity*, Actes de AAAI'87, pp. 268-272, 1987.
- [Agre et Chapman 1990] AGRE P. E. ET CHAPMAN D., "What are Plans for ?", *Journal of Robotics and Autonomous Systems*, vol. 6, n° 17-34, 1990.
- [Anzieu et Martin 1969] ANZIEU D. ET MARTIN J. Y., *La Dynamique des Groupes Restreints*, Presses Universitaires de France, Paris, 1969.
- [Arkin 1988] ARKIN R. C., "Cooperation without Communication: Multi-Agent Based Robot Navigation", Mobile Robot Laboratory - Georgia Institute of Technology, 1988
- [Arkin et Hobbs 1993] ARKIN R. C. ET HOBBS J. D., "Dimensions of Communications and Social Organization in Multi-Agent Robotic Systems", in *From Animals to Animats II*, MIT Press, Cambridge, pp. 486-493, 1993.
- [Armitage 1987] ARMITAGE K. B., "Social dynamics of mammals : reproductive success, kinship and individual fitness", *TREE*, n° 2, pp. 279-284, 1987.
- [Ashby 1962] ASHBY W. R., "Principles of the Self-Organizing System", in *Principles of Self Organization*, Pergamon Press, New York, pp. 255-278, 1962.
- [Assad et Packard 1992] ASSAD A. M. ET PACKARD N. H., "Emergent Colonization in an Artificial Ecology", in *Towards a Practice of Autonomous Systems*, MIT Press, Cambridge, pp. 143-152, 1992.
- [Atlan 1972] ATLAN H., *L'organisation Biologique et la Théorie de l'Information*, Hermann, Paris, 1972.
- [Atlan 1979] ATLAN H., *Entre le Cristal et la Fumée, Essai sur l'Organisation du Vivant*, Editions du Seuil, Paris, 1979.
- [Atlan et Meyer 1991] ATLAN L. ET MEYER J.-A., "Genetic Programming Applied to Neural Network Design", Rapport Technique Ecole Normale Supérieure - Groupe de BioInformatique, n° 91-1, 1991.
- [Beer 1990] BEER R. D., *Intelligence as Adaptive Behavior*, Academic Press, Londres, 1990.
- [Belic et al. 1986] BELIC M. R., SKARKA V., DENEUBOURG J. L. ET LAX M., "Mathematical Model of Honeycomb Construction", *Journal of Mathematical Biology*, n° 24, pp. 437-449, 1986.
- [Beni et Hackwood 1992] BENI G. ET HACKWOOD S., "The Maximum Entropy Principle and Sensing in Swarm Intelligence", in *Toward a Practice of Autonomous Systems*, MIT Press, Cambridge, pp. 153-160, 1992.
- [Benz 1993] BENZ T., "The Microscopic Traffic Simulator AS (Autobahn Simulator)", in *Modelling and Simulation: Proceedings of ESM'93*, Simulation Councils, 486-489, 1993.
- [Bond et Gasser 1988] *Readings In Distributed Artificial Intelligence*, sous la direction de BOND A. H. ET GASSER L., Morgan Kaufman, 1988.
- [Bonnet, Ghiglione et Richard 1989] *Perception, Action, Langage - Traité de Psychologie Cognitive*, sous la direction de BONNET C., GHIGLIONE R. ET RICHARD J.-F., Dunod, Paris, 1989.
- [Boudon 1992] *Traité de Sociologie*, sous la direction de BOUDON R., Presses Universitaires de France, Paris, 1992.
- [Bouron 1991] BOURON T., "COMMAS: A Communication Model for Multi-Agents System", in *Modelling and Simulation: Proceedings of ESM'91*, Simulation Councils, Copenhague, pp. 220-225, 1991.
- [Bouron 1992] BOURON T., *Structures de Communication et d'Organisation pour la Coopération dans un Univers Multi-Agents*, Thèse de Doctorat, Université Paris VI, 1992.
- [Bousquet et al. 1992] BOUSQUET F., CAMBIER C., MULLON C. ET QUENSIERE J., *Simulating Fishermen Society*, Actes de Simulating Societies Symposium, Guildford (Angleterre), pp. 63-79, 1992.
- [Briot 1988] BRIOT J. P., "From Objects to Actors, Study of a Limited Symbiosis in Smalltalk-80", Rapport Technique LITP, n° 88-58 RXF, 1988.
- [Brooks 1983] BROOKS R., "Solving the Find-Path Problem by Good Representation of Free Space", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 13, n° 3, pp. 190-197, 1983.
- [Brooks 1987] BROOKS R., "Planning is just a Way of Avoiding Figuring Out What to Do Next", Working Paper MIT, n° 303, 1987.
- [Brooks 1990a] BROOKS R., "Elephants Don't Play Chess", *Journal of Robotics and Autonomous Systems*, vol. 6, n° 1,2, pp. 3-15, 1990a.
- [Brooks 1990b] BROOKS R., "Integrated Systems Based on Behaviors", *SIGART Bulletin*, vol. 2, n° 4, pp. 46-50, 1990b.
- [Brooks 1990c] BROOKS R., "A Robot that Walks: Emergent Behavior from a Carefully Evolved Network", in *A.I. at MIT - Expanding Frontiers*, MIT Press, Cambridge, pp. 29-39, 1990c.

- [Brooks 1991] BROOKS R., *Intelligence without Reason*, Actes de IJCAI'91, Sydney (Australie), Morgan-Kaufmann, pp. 569-595, 1991.
- [Brooks et Connell 1986] BROOKS R. ET CONNELL J. H., "Asynchronous distributed control system for a mobile robot", *SPIE*, vol. 727 (Mobile Robots), pp. 77-83, 1986.
- [Bura et al. 1991] BURA S., DROGOUL A., FERBER J. ET JACOPIN E., *Eco-Résolution: un Modèle de Résolution par Interactions*, Actes de RFIA, Lyon, 1991.
- [Bura et al. 1993] BURA S., GUERIN-PACE F., MATHIAN H., PUMAIN D. ET SANDERS L., *Multi-Agents Systems and the Dynamics of a Settlement System*, Actes de Simulating Societies Symposium, Sienna (Italie), 1993.
- [Carbiener 1991] CARBIENER R., "L'Écologie, Science de l'Économie de la Nature", in *Systèmes Naturels, Systèmes Artificiels*, Champ Vallon, Seyssel, pp. 69-96, 1991.
- [Castelfranchi 1993] CASTELFRANCHI C., *Towards a Theory of Emergence*, Actes de Simulating Societies Symposium, Panel on "Emergence", Sienna (Italie), 1993.
- [Castelfranchi et Conte 1992] CASTELFRANCHI C. ET CONTE R., *Mind is not Enough: Precognitive Bases of Social Interaction*, Actes de Simulating Societies Symposium, University of Surrey, Guildford (Angleterre), pp. 93-110, 1992.
- [Chapman 1990] CHAPMAN D., *Vision, Instruction and Action*, PhD Thesis, MIT AI Laboratory, 1990.
- [Chauvin 1982] CHAUVIN R., *Les Sociétés Animales*, Presses Universitaires de France, Paris, 1982.
- [Chauvin 1989] CHAUVIN R., *Des Animaux et des Hommes*, Seghers, Paris, 1989.
- [Chauvin et Callais-Hamonno 1972] CHAUVIN R. ET CALLAIS-HAMONNO F., "Simulation sur Ordinateur de la Construction du Dôme et du Ramassage des Brindilles chez les Fourmis", *Comptes-Rendus de l'Académie des Sciences*, n° 275, pp. 1275-1278, 1972.
- [Collinot 1988] COLLINOT A., *Le Problème du Contrôle dans un Système Flexible d'Ordonnement*, Thèse de Doctorat, Université Paris 6, 1988.
- [Collinot et Hayes-Roth 1991] COLLINOT A. ET HAYES-ROTH B., "Real-Time Performance of Intelligent Autonomous Agents", in *Decentralized A.I. 3*, Elsevier North Holland, Amsterdam, pp. 341-356, 1991.
- [Collins et Jefferson 1991] COLLINS R. J. ET JEFFERSON D. R., "Representation for Artificial Organisms", in *From Animals to Animals*, MIT Press, Cambridge, pp. 382-390, 1991.
- [Coloni, Dorigo et Maniezzo 1992a] COLONI A., DORIGO M. ET MANIEZZO V., "Distributed Optimization by Ant Colonies", in *Toward a Practice of Autonomous Systems*, MIT Press, Cambridge, pp. 134-142, 1992a.
- [Coloni, Dorigo et Maniezzo 1992b] COLONI A., DORIGO M. ET MANIEZZO V., *An Investigation of some Properties of an "Ant Algorithm"*, Actes de Parallel Problem Solving from Nature (PPSN'92), Bruxelles, 1992b.
- [Connah 1991] CONNAH D., "Why We Need a New Approach to the Design of Agents", *ALISBQ*, n°76, Spring 1991, pp. 13-15, 1991.
- [Connell 1990] CONNELL J. H., *Minimalist Mobile Robotics*, Academic Press, Londres, 1990.
- [Corbara 1991] CORBARA B., *L'organisation Sociale et sa Genèse chez la Fourmi Ectatomma ruidum Roger*, Thèse de troisième cycle, Paris XIII - Villetaneuse, 1991.
- [Corbara et al. 1991] CORBARA B., DENEUBOURG J. L., FRESNEAU D., GOSS S., LACHAUD J. P. ET PHAM-NGOC A., *Simulation de la Genèse d'une Division du Travail au Sein d'une Société de Fourmis Ponérines: un Modèle d'Auto-Organisation*, Actes de Colloque Insectes Sociaux 7, pp. 205-206, 1991.
- [Corbara et al. 1993] CORBARA B., DROGOUL A., FRESNEAU D. ET LALANDE S., "Simulating the Sociogenesis Process in Ant Colonies with MANTA", in *Towards a Practice of Autonomous Systems II*, MIT Press, Cambridge, 1993.
- [Corbara, Fresneau et Lachaud 1986] CORBARA B., FRESNEAU D. ET LACHAUD J.-P., *Données Préliminaires sur les Premiers Stades du Développement de la Société chez Ectatomma ruidum*, Actes du III^e Congrès National des Sociétés Savantes, Poitiers, pp. 203-212, 1986.
- [Corbara, Fresneau et Lachaud 1987] CORBARA B., FRESNEAU D. ET LACHAUD J.-P., "Niveaux de Description et Organisation Sociale chez les Fourmis", *Bulletin SFECA*, n° 2, pp. 31-36, 1987.
- [Corbara, Fresneau et Lachaud 1988] CORBARA B., FRESNEAU D. ET LACHAUD J. P., "La Flexibilité de l'Emploi chez les Fourmis", *La Recherche*, n° 195, pp. 116-117, 1988.
- [Corbara et al. 1986] CORBARA B., FRESNEAU D., LACHAUD J. P., LECLERC Y. ET GOODALL G., "An Automated Photographic Technique for Behavioural Investigations of Social Insects", *Behavioural Processes*, vol. 13, pp. 237-249, 1986.
- [Corsi 1990] *La Fabrique de la Pensée*, sous la direction de CORSI P., Electa, Milan, 1990.
- [Crozier et Friedberg 1977] CROZIER M. ET FRIEDBERG E., *L'acteur et le Système*, Editions du Seuil, Paris, 1977.
- [Czternasty 1993] CZTERNASTY N., "Impact de l'Introduction d'un Facteur de Dominance entre les Reines d'une Société Artificielle de Fourmis. Etude sur l'Utilisation de MANTA en Ethologie", Rapport de Stage du DEA I.A.R.F.A., Université Paris VI, 1993.
- [De Rosnay 1975] DE ROSNAY J., *Le Macroscopie: vers une Vision Globale*, Editions du Seuil, Paris, 1975.
- [Deffuant 1992] DEFFUANT G., *Réseaux Connexionnistes Auto-Constructifs*, Thèse de Troisième Cycle, Ecole des Hautes Etudes en Sciences Sociales et Université Paris VI, 1992.

- [Delaye et Ferber 1992] DELAYE C. ET FERBER J., *Morphological and Behavioral Adaptation of Robots Using Genetic Algorithms*, Actes de Artificial Life III, Santa-Fe, 1992.
- [Demazeau 1991] DEMAZEAU Y., *Coordination Patterns in Multi-Agent Worlds; Application to Computer Vision and Robotics*, Actes de IEE Colloquium on Intelligent Agents, Savoy Place, 1991.
- [Deneubourg 1977] DENEUBOURG J.-L., "Application de l'Ordre par Fluctuations à la Description de Certaines Etapes de la Construction du Nid chez les Termites", *Insectes Sociaux*, n° 24, pp. 117-130, 1977.
- [Deneubourg, Theraulaz et Beckers 1992] DENEUBOURG J.-L., THERAULAZ G. ET BECKERS R., "Swarm-Made Architectures", in *Towards a Practice of Autonomous Systems*, MIT Press, Cambridge, pp. 123-133, 1992.
- [Deneubourg et al. 1986] DENEUBOURG J. L., ARON S., GOSS S., PASTEELS J. M. ET DUERINCK G., "Random Behaviour, Amplification Processes and Number of Participants: How they Contribute to the Foraging Properties of Ants", *Physica*, vol. 22D, pp. 176-186, 1986.
- [Deneubourg et Goss 1989] DENEUBOURG J. L. ET GOSS S., "Collective Patterns and Decision-Making", *Ecology Ethology and Evolution*, vol. 1, pp. 295-311, 1989.
- [Deneubourg et al. 1987] DENEUBOURG J. L., GOSS S., PASTEELS J., FRESNEAU D. ET LACHAUD J. P., "Self-Organization Mechanisms in Ants Societies (II): Learning in Foraging and Division of Labour", in *From Individual to Collective Behaviour in Social Insects*, Birkhäuser Verlag, Basel, pp. 177-196, 1987.
- [Deneubourg et al. 1991] DENEUBOURG J. L., GOSS S., FRANKS N., SENDOVA-FRANKS A., DETRAIN C. ET CHRETIEN L., "The Dynamics of Collective Sorting, Robot-Like Ants and Ant-Like Robots", in *From Animals to Animats*, MIT Press, Cambridge, pp. 356-366, 1991.
- [Deneubourg, Pasteels et Verhaeghe 1984] DENEUBOURG J. L., PASTEELS J. ET VERHAEGHE J.-C., "Quand l'Erreur Alimente l'Imagination d'une Société: le Cas des Fourmis", *Nouvelles de la Science et des Technologies*, vol. 2, n° 1-2, pp. 47-52, 1984.
- [Doran et al. 1990] DORAN J., CARVAJAL H., CHOO Y. J. ET LI Y., "The MCS Multi-Agent Tesbed, Developments and Experiments", in *Cooperating Knowledge Based Systems*, Springer Verlag, 1990.
- [Doran, Palmer et Gilbert 1992] DORAN J., PALMER M. ET GILBERT N., *The EOS Project: Modelling Upper Palaeolithic Social Change*, Actes de Simulating Societies Symposium, University of Surrey, Guildford (Angleterre), pp. 31-47, 1992.
- [Drogoul 1990] DROGOUL A., "Un Eco-Planificateur en Smalltalk-80", Mémoire de Stage du DEA I.A.R.F.A, Université Paris VI, 1990.
- [Drogoul 1992] DROGOUL A., "From Animals to Animats: Notes de Lecture et Discussion", *Bulletin de l'AFIA*, n° 8, pp. 34-38, 1992.
- [Drogoul 1993] DROGOUL A., *When Ants Play Chess (or Can Strategies Emerge from Tactical Behaviors ?)*, Actes de MAAMAW'93, Lausanne, 1993. (A Paraître).
- [Drogoul en prép.] DROGOUL A., "Using Reactive Agents in Two Classical DAI Problems: the Pursuit Game and the Tileworld", en préparation.
- [Drogoul, Corbara et Fresneau 1993] DROGOUL A., CORBARA B. ET FRESNEAU D., *MANTA: New Experimental Results on the Emergence of (Artificial) Ant Societies*, Actes de Simulating Societies, Sienna, 1993.
- [Drogoul et Dubreuil 1990] DROGOUL A. ET DUBREUIL C., "Taquin et Eco-Résolution", Rapport Technique LAFORIA, n° 30/90, 1990.
- [Drogoul et Dubreuil 1991] DROGOUL A. ET DUBREUIL C., "Eco-Problem-Solving Model: Results of the N-Puzzle", in *Decentralized A.I. 3*, Elsevier North Holland, Amsterdam, pp. 283-295, 1991.
- [Drogoul et Dubreuil 1993] DROGOUL A. ET DUBREUIL C., *A Distributed Approach to N-Puzzle Solving*, Actes du Distributed Artificial Intelligence Workshop, 1993.
- [Drogoul et Ferber 1993] DROGOUL A. ET FERBER J., "From Tom-Thumb to the Dockers: Some Experiments with Foraging Robots", in *From Animals to Animats II*, MIT Press, Cambridge, pp. 451-459, 1993.
- [Drogoul et al. 1992] DROGOUL A., FERBER J., CORBARA B. ET FRESNEAU D., "A Behavioral Simulation Model for the Study of Emergent Social Structures", in *Towards a Practice of Autonomous Systems*, MIT Press, Cambridge, pp. 161-170, 1992.
- [Drogoul, Ferber et Jacopin 1991] DROGOUL A., FERBER J. ET JACOPIN E., "Pengi: Applying Eco-Problem-Solving for Behavior Modelling in an Abstract Eco-System", in *Modelling and Simulation: Proceedings of ESM'91*, Simulation Councils, Copenhagen, pp. 337-342, 1991.
- [Drogoul, Ferber et Lalande 1993] DROGOUL A., FERBER J. ET LALANDE S., "Du Petit Poucet aux Dockers: une Etude Comparative d'Architectures de Robots Fourrageurs", Rapport Technique du LAFORIA, n° 93/13, 1993.
- [Dumouchel et Dupuy 1983] *Colloque de Cerisy: l'Auto-Organisation, de la Physique au Politique*, sous la direction de DUMOUCHEL P. ET DUPUY J. P., Editions du Seuil, Paris, 1983.
- [Durfee 1989] DURFEE E., *Coordination of Distributed Problem Solvers*, Kluwer Academic Publishers, Amsterdam, 1989.
- [Entwistle 1991] ENTWISTLE B., "Micro-Macro Theoretical Linkages in Social Demography", in *Macro-Micro Linkages in Sociology*, Sage, Newbury Park (Californie), 1991.
- [Erceau et Ferber 1991] ERCEAU J. ET FERBER J., "L'Intelligence Artificielle Distribuée", *La Recherche*, n° 233, pp. 750-758, 1991.

- [Ferber 1989] FERBER J., *Objets et agents: une étude des structures de représentation et de communications en Intelligence Artificielle*, Thèse d'Etat, Université Paris VI, 1989.
- [Ferber 1990a] FERBER J., *Conception et Programmation par Objets*, Hermès, Paris, 1990a.
- [Ferber 1990b] FERBER J., "Eco Problem Solving: How to Solve a Problem by Interactions", Rapport Technique LAFORIA, n° 5/90, 1990b.
- [Ferber en prép.] FERBER J., *Organisations Artificielles: une Introduction à l'Intelligence Artificielle Distribuée et aux Systèmes Multi-Agents*, en préparation.
- [Ferber et Drogoul 1992] FERBER J. ET DROGOUL A., "Using Reactive Multi-Agent Systems in Simulation and Problem Solving", in *Distributed Artificial Intelligence: Theory and Praxis*, ECSC-EEC-EAEC, Bruxelles et Luxembourg, pp. 53-80, 1992.
- [Ferber et Jacopin 1990] FERBER J. ET JACOPIN E., "The Framework of Eco Problem Solving", in *Decentralized A.I.*, Elsevier North-Holland, Amsterdam, pp. 103-114, 1990.
- [Ferguson 1991] FERGUSON I. A., "Toward an Architecture for Adaptive, Rational, Mobile Agents", in *Decentralized A.I. 3*, Elsevier North Holland, Amsterdam, pp. 249-261, 1991.
- [Flynn et Brooks 1988] FLYNN A. M. ET BROOKS R. A., "MIT Mobile Robots: What's Next ?", *IEEE*, pp. 611-617, 1988.
- [Fogelman 1985] FOGELMAN F., *Cerveau et Machines: des Architectures pour Demain ?*, Actes du Forum Cognitiva Cesta, Paris, 1985.
- [Fresneau, Corbara et Lachaud 1989] FRESNEAU D., CORBARA B. ET LACHAUD J. P., "Organisation Sociale et Structuration Spatiale autour du Couvain chez *Pachycondyla apicalis* (Formicidae, Ponerinae)", *Actes Coll. Ins. Soc.*, n° 5, pp. 83-92, 1989.
- [Fresneau et Dupuy 1988] FRESNEAU D. ET DUPUY P., "A Study of Polyethism in a Ponerine Ant: *Neoponera Apicalis* (Hymenoptera, Formicidae)", *Animal Behaviour*, vol. 36, pp. 1389-1399, 1988.
- [Friedberg 1993] FRIEDBERG E., *Le Pouvoir et la Règle*, Editions du Seuil, Paris, 1993.
- [Ghedira et Verfaillie 1991] GHEDIRA K. ET VERFAILLIE G., "Approche Multi-Agents d'un Problème de Satisfaction de Contraintes: le Problème d'Affectation", Rapport Final du CERT - DERI, n° 1/3412-00/DERI, 1991.
- [Giroux 1993] GIROUX S., *Agents et Systèmes, une Nécessaire Unité*, Thèse de Doctorat, Université de Montréal, 1993.
- [Glance et Huberman 1992] GLANCE N. S. ET HUBERMAN B. A., "The Outbreak of Cooperation", (A Paraître dans *Journal of Mathematical Sociology*), 1992
- [Glandsdorff et Prigogine 1971] GLANSDORFF P. ET PRIGOGINE I., *Structure, Stabilité et Fluctuation*, Masson, Paris, 1971.
- [Gleick 1989] GLEICK J., *La Théorie du Chaos*, Flammarion, Paris, 1989.
- [Goss et al. 1990] GOSS S., BECKERS R., DENEUBOURG J. L., ARON S. ET PASTEELS J. M., "How Trail Laying and Trail Following Can Solve Foraging Problems For Ant Colonies", *Behavioural Mechanisms of Food Selection*, NATO ASI Series, n° G20, pp. 661-678, 1990.
- [Goss et Deneubourg 1992] GOSS S. ET DENEUBOURG J.-L., "Harvesting By a Group of Robots", in *Towards a Practice of Autonomous Systems*, MIT Press, Cambridge, pp. 195-204, 1992.
- [Graham et Wavish 1991] GRAHAM M. ET WAVISH P., *Simulating and Implementing Agents and Multiple Agent Systems*, Actes de European Simulation MultiConference, Copenhague, SCS, pp. 226-231, 1991.
- [Grassé 1959] GRASSE P.-P., "La Reconstruction du Nid et les Coordinations Interindividuelles. La Théorie de la Stigmergie", *Insectes Sociaux*, n° 6, pp. 41-84, 1959.
- [Guillot 1986] GUILLOT A., "Revue Générale des Méthodes d'Etude des Séquences Comportementales", *Etudes et Analyses Comportementales*, vol. 2, n° 3, pp. 89-106, 1986.
- [Hadet 1993] HADET A. D., *De la Synthébiologie à l'Intelligemence*, Presses Universelles de France, Paris, 1993.
- [Hamilton 1964] HAMILTON W. D., "The Genetical Evolution of Social Behaviour", *Journal of Theoretical Biology*, n° 7, pp. 1-52, 1964.
- [Hart, Nilsson et Raphael 1968] HART P. E., NILSSON N. J. ET RAPHAEL B., "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", *IEEE Trans. Syst. Sci. Cybern.*, n° 4, 1968.
- [Hatcher, Tofts et Francks 1992] HATCHER M., TOFTS C. ET FRANCKS N. R., "Synchronization and Mutual Exclusion within Ant Nests", in *Biology and Evolution of Social Insects*, Leuven University Press, Leuven, pp. 351-357, 1992.
- [Hofstadter 1985] HOFSTADTER D., *Gödel, Escher, Bach: les Brins d'une Guirlande Eternelle*, InterEditions, Paris, 1985.
- [Hogeweg 1988] HOGEWEG P., "MIRROR beyond MIRROR, Puddles of Life", in *Artificial Life*, Addison-Wesley, London, pp. 298-316, 1988.
- [Hogeweg et Hesper 1979] HOGEWEG P. ET HESPER B., "Heterarchical, Selfstructuring Simulation Systems: Concepts and Applications in Biology", in *Methodology in Systems Modelling and Simulation*, North-Holland, pp. 221-232, 1979.
- [Hogeweg et Hesper 1981] HOGEWEG P. ET HESPER B., "Two Predators and One Prey in a Patchy Environment: an Application of MICMAC Modelling", *Journal of Theoretical Biology*, n° 93, pp. 411-432, 1981.
- [Hogeweg et Hesper 1983] HOGEWEG P. ET HESPER B., "The Ontogeny of the Interaction Structure in Bumble Bee Colonies: a MIRROR Model", *Behavioral Ecology and Sociobiology*, n° 12, pp. 271-283, 1983.
- [Hogeweg et Hesper 1985] HOGEWEG P. ET HESPER B., "SocioInformatic Processes: MIRROR Modeling Methodology", *Journal of Theoretical Biology*, n° 113, pp. 311-330, 1985.

- [Hogeweg et Hesper 1991] HOGEWEG P. ET HESPER B., "Evolution as Pattern Processing: TODO as Substrate for Evolution", in *From Animals to Animats*, MIT Press, Cambridge, pp. 492-497, 1991.
- [Holland et Snaith 1991] HOLLAND O. ET SNAITH M., "Interrupt-Driven Architectures for Behavioural Robots: a Neural Implementation", *Artificial Life Technologies - Stroud* (Angleterre), 1991
- [Hölldobler et Wilson 1990] HÖLLDOBLER B. ET WILSON E. O., *The Ants*, Springer-Verlag, Berlin, 1990.
- [Hubbermann et Hogg 1988] HUBBERMANN B. A. ET HOGG T., "The Behavior of Computational Ecologies", in *The Ecology of Computation*, North-Holland, Amsterdam, 1988.
- [Huberman et Hogg 1987] HUBERMAN B. ET HOGG T., "Phase Transitions in Artificial Intelligence Systems", *Artificial Intelligence*, vol. 33, n° 2, pp. 155-171, 1987.
- [Jacopin 1991] JACOPIN E., "Des Fourmis à l'Attaque de Mars", *La Recherche*, n° 234, pp. 970-971, 1991.
- [Jacopin 1992] JACOPIN E., "La Modularité dans l'Action: les Enseignements d'une Implantation", Rapport Technique LAFORIA, n° 15/92, 1992.
- [Jaisson 1993] JAISSON P., *La Fourmi et le Sociobiologiste*, Editions Odile Jacob, Paris, 1993.
- [Julia 1992] JULIA D., *Dictionnaire de la Philosophie*, Larousse, Paris, 1992.
- [Kagawa 1992] KAGAWA Y., "A Behavior-Based Approach to Coordinate Multiple Mobile Robots", soumis à IEEE Conference on Robotics and Automation, 1992
- [Kiss 1991] KISS G., "Variable Coupling of Agents to their Environment: Combining Situated and Symbolic Automata", in *Decentralized A.I. 3*, Elsevier North Holland, Amsterdam, pp. 231-247, 1991.
- [Kolasa et Pickett 1991] *Ecological Heterogeneity*, sous la direction de KOLASA J. ET PICKETT S. T., Springer-Verlag, 1991.
- [Korf 1985] KORF R. E., "Depth-First Iterative-Deepening: an Optimal Admissible Tree Search", *AI Journal*, n°27, 1985.
- [Korf 1990a] KORF R. E., "Real-Time Heuristic Search", *AI Journal*, vol. 42, n° 1990a.
- [Korf 1990b] KORF R. E., "Real-Time Heuristic Search: New Results", 1990b
- [Korf 1990c] KORF R. E., *Real-Time Search for Dynamic Planning*, Actes de AAAI Symposium on Planning in Uncertain, Unpredictable or Changing Environments, Stanford, 1990c.
- [Korf 1991] KORF R. E., *A Simple Solution to Pursuit Games*, Actes du Distributed Artificial Intelligence Workshop, Boston, pp. 183-194, 1991.
- [Krebs 1978] KREBS J. R., "Optimal Foraging: Decision Rules for Predators", in *Behavioural Ecology : an Evolutionary Approach*, Blackwell Scientific Publications Ltd, Oxford, pp. 23-63, 1978.
- [Krebs et Davies 1978] *Behavioural Ecology*, sous la direction de KREBS J. R. ET DAVIES N. B., Blackwell Scientific Publications Ltd, Oxford, 1978.
- [Kuhn, Müller et Müller 1993] KUHN N., MÜLLER J. P. ET MÜLLER J., "Simulating Cooperative Transportation Companies", in *Modelling and Simulation: Proceedings of ESM'93*, Simulation Councils, pp. 333-337, 1993.
- [Lachaud et Fresneau 1987] LACHAUD J. P. ET FRESNEAU D., "Social Regulation in Ponerine Ants", in *From Individual to Collective Behaviour in Social Insects*, Birkhäuser Verlag, Basel, pp. 197-217, 1987.
- [Lalande 1992] LALANDE S., "Etude de l'Emergence de Structures Sociales dans une Société de Fourmis Simulée à travers le Suivi et la Réalisation d'une Sociogénèse", Mémoire de Projet du DEA I.A.R.F.A, Université Paris VI, 1992.
- [Langton 1988] *Artificial Life*, sous la direction de LANGTON C., Addison-Wesley, London, 1988.
- [Langton et al. 1990] *Artificial Life II*, sous la direction de LANGTON C., TAYLOR C., FARMER J. D. ET RASMUSSEN S., Addison-Wesley, London, 1990.
- [Lapierre 1992] LAPIERRE J.-W., *L'Analyse de Systèmes; L'application aux Sciences Sociales*, Syros, Paris, 1992.
- [Le Fur 1993] LE FUR J., *Praticabilité de l'Approche Système pour la Modélisation d'un Système d'Exploitation Halieutique*, Actes du Forum Halieumetrics, Rennes, 1993.
- [Le Moigne 1991] LE MOIGNE J.-L., "«...Voir la Nature avec les Yeux de l'Art»", in *Systèmes Naturels, Systèmes Artificiels*, Champ Vallon, Seyssel, pp. 123-139, 1991.
- [Lenoir 1979] LENOIR A., "Le Comportement Alimentaire et la Division du Travail chez la Fourmi *Lasius Niger*", *Bulletin Biologique de la France et de la Belgique*, vol. 113, n° 2-3, pp. 79-314, 1979.
- [Lenoir et Mardon 1978] LENOIR A. ET MARDON J.-C., "Note sur l'Application de l'Analyse des Correspondances à la Division du Travail chez les Fourmis", *Comptes Rendus de l'Académie des Sciences*, vol. 287, n° 5, pp. 555-558, 1978.
- [Lestel, Grison et Drogoul 1993] LESTEL D., GRISON B. ET DROGOUL A., "Les Agents Réactifs et le Vivant dans une Perspective d'Evolution Coopérative", 1993 (à paraître dans la revue *Intellectica*)
- [Levy 1965] LEVY A., *Psychologie Sociale, Textes Fondamentaux*, Dunod, Paris, 1965.
- [Lindauer 1986] LINDAUER M., "Communication and Orientation in Honeybees", *Monit. Zool. Ital. (N.S.)*, vol. 20, pp. 371-379, 1986.
- [Loeb 1900] LOEB J., *Comparative Physiology of the Brain and Comparative Psychology*, Putnam, New York, 1900.
- [Lomborg 1992] LOMBORG B., "An Evolution of Cooperation", Rapport Technique de l'Université de Aarhus - Institute of Political Science, 1992.
- [Lorenz 1984] LORENZ K., *Les Fondements de l'Ethologie*, Flammarion, Paris, 1984.

- [Mac Lennan 1991] MAC LENNAN B., "Synthetic Ethology: An Approach to the Study of Communication", in *Artificial Life II*, Addison-Wesley, London, pp. 631-657, 1991.
- [Maes 1990] MAES P., "Situating Agents Can Have Goals", *Journal of Robotics and Autonomous Systems*, vol. 6, n° 1-2, pp. 49-70, 1990.
- [Maes 1991a] MAES P., *Adaptive Action Selection*, Actes de Cognitive Science Conference, 1991a.
- [Maes 1991b] MAES P., *The Agent Network Architecture (ANA)*, Actes de AAAI Spring Symposium on Integrated Intelligent Architectures, 1991b.
- [Maes 1991c] MAES P., "A Bottom-Up Mechanism for Behavior Selection in an Artificial Creature", in *From Animals to Animats*, MIT Press, Cambridge, pp. 239-246, 1991c.
- [Marx et Engels 1968] MARX K. ET ENGELS F., *L'idéologie Allemande*, Editions Sociales, Paris, 1968.
- [Mataric 1993] MATARIC M. M., "Designing Emergent Behaviors: From Local Interactions to Collective Intelligence", in *From Animals to Animats II*, MIT Press, Cambridge, pp. 432-441, 1993.
- [May 1976] MAY R., "Simple Mathematical Models with Very Complicated Dynamics", *Nature*, vol. 261, pp. 459-467, 1976.
- [McFarland 1990] MCFARLAND D., *Dictionnaire du Comportement Animal*, Robert Laffont, Paris, 1990.
- [Meyer 1980] MEYER J. A., "La Théorie des Jeux à Deux Joueurs et ses Applications Ecologiques", Rapport Technique Ecole Normale Supérieure, n° 1, 1980.
- [Minsky 1988] MINSKY M., *La Société de l'Esprit*, InterEditions, Paris, 1988.
- [Miriad 1992] EQUIPE MIRIAD, *Approcher la Notion de Collectif*, Actes de la Journée Multi-Agents du PRC-IA, Nancy, 1992. (Article Collectif de l'équipe MIRIAD, LAFORIA, Université Paris VI).
- [Monod 1970] MONOD J., *Le Hasard et la Nécessité*, Editions du Seuil, Paris, 1970.
- [Morgan 1894] MORGAN C. L., *An Introduction to Comparative Psychology*, Scott, Londres, 1894.
- [Morin 1977] MORIN E., *La Méthode, Tome I: la Nature de la Nature*, Editions du Seuil, Paris, 1977.
- [Mosekilde 1991] *Modelling and Simulation: Proceedings of ESM'91*, sous la direction de MOSEKILDE E., Simulation Councils, Copenhagen, 1991.
- [Moysen et Manderick 1988] MOYSON F. ET MANDERICK B., "The Collective Behavior of Ants: an Example of Self-Organization in Massive Parallelism", Rapport du VUB AI Lab, n° 88-7, 1988.
- [Newell et Simon 1976] NEWELL A. ET SIMON H. A., "Computer Science as Empirical Inquiry: Symbols and Search", *Communications of the ACM*, vol. 19, n° 3, pp. 113-126, 1976.
- [Novels et Hackwell 1991] NOVELS M. D. ET HACKWELL G. B., *Simulation: Decision with Vision*, Actes de ESM'91, Copenhagen, Simulation Councils, pp. 425-430, 1991.
- [Nowak et Latané 1992] NOWAK A. ET LATANE B., *Simulating the Emergence of Social Order from Individual Behavior*, Actes de Simulating Societies Symposium, Université du Surrey, Guildford (Angleterre), pp. 139-141, 1992.
- [Papert 1981] PAPER S., *Mindstorms: Children, Computers, and Powerful Ideas*, Basic Books, New York, 1981.
- [Passera 1984] PASSERA L., *L'Organisation Sociale des Fourmis*, Privat, Toulouse, 1984.
- [Pavé 1993] *Modelling and Simulation: Proceedings of ESM'93*, sous la direction de PAVE A., Simulation Councils, 1993.
- [Payton 1990] PAYTON D. W., "Internalized Plans: a Representation for Action Resources", *Journal of Robotics and Autonomous Systems*, vol. 6, n° 1-2, pp. 89-103, 1990.
- [Pêcheux 1990] PECHEUX M. G., *Le Développement des Rapports des Enfants à l'Espace*, Nathan, Paris, 1990.
- [Perez 1989] PEREZ J.-C., *De Nouvelles Voies vers l'Intelligence Artificielle*, Masson, Paris, 1989.
- [Perrot 1990] PERROT J.-F., "Programmation par Objets", Rapport d'Activité Scientifique 1988-1990 du LAFORIA, pp. 65-72, n°1990.
- [Piaget 1967] PIAGET J., *Biologie et Connaissance*, Gallimard, Paris, 1967.
- [Pomian 1993] POMIAN J., *L'Intelligence Artificielle*, Presses Pocket, Paris, 1993.
- [Prigogine 1982] PRIGOGINE I., *Physique, Temps et Devenir*, Masson, Paris, 1982.
- [Prigogine et Stengers 1985] PRIGOGINE I. ET STENGERS I., *Order out of Chaos: Man's New Dialog with Nature*, Fontana Paperbacks, Londres, 1985.
- [Rabaud 1937] RABAUD E., *Phénomène Social et Sociétés Animales*, Bibliothèque de Philosophie Contemporaine. Librairie Felix Arcan, Paris, 1937.
- [Ratner et Warmuth 1986] RATNER D. ET WARMUTH M., *Finding a Shortest Solution for the NxN Extension of the 15-Puzzle is Intractable*, Actes de AAAI'86, Philadelphie, 1986.
- [Reeve et Gamboa 1987] REEVE H. K. ET GAMBOA G. J., "Queen Regulation of Worker Foraging in Paper Wasps: a Social Feedback Control System", *Behaviour*, vol. 102, pp. 147-167, 1987.
- [Resnick 1992] RESNICK M., *Beyond the Centralized Mindset*, Actes de WOFCAI'92, Paris, pp. 369-396, 1992.
- [Ribeiro, Barthès et Oliveira 1993] RIBEIRO F., BARTHES J.-P. ET OLIVEIRA E., "Dynamic Selection of Action Sequences", in *From Animals to Animats II*, MIT Press, Cambridge, pp. 189-195, 1993.
- [Ricard 1970] *Atlas de Biologie*, sous la direction de RICARD M., Stock, Paris, 1970.
- [Richard 1975] RICHARD G., *Les Comportements Instinctifs*, Presses Universitaires de France, Paris, 1975.

- [Richard 1980] RICHARD G., "Taxies", in *Encyclopædia Universalis*, Paris, pp. 756-760, 1980.
- [Rieu 1991] RIEU A.-M., "De la structure au Système dans la Théorie Sociale", in *Systèmes Naturels, Systèmes Artificiels*, Champ Vallon, Seyssel, pp. 212-225, 1991.
- [Roitblat 1982] ROITBLAT H. L., "The Meaning of Representation in Animal Memory", *The Behavioral and Brain Sciences*, vol. 5, pp. 353-406, 1982.
- [Schnepf 1991] SCHNEPF U., "Robot Ethology: a Proposal for the Research into Intelligent Systems", in *From Animals to Animats*, MIT Press, Cambridge, pp. 465-474, 1991.
- [Seel 1991] SEEL N., "Perspectives for Distributed Artificial Intelligence", *AISBQ*, n° 76, Spring 1991, pp. 11-12, 1991.
- [Seger 1993] SEGER J., "Cooperation and Conflict in Social Insects", in *Behavioural Ecology III*, Blackwell Scientific Publications Ltd, Oxford, pp. 339-373, 1993.
- [Sekine, Nakanishi et Ukita 1991] SEKINE M., NAKANISHI H. ET UKITA M., "A Shallow-Sea Ecological Model Using an Object-Oriented Programming Language", *Ecological Modelling*, vol. 57, pp. 221-236, 1991.
- [Selfridge 1956] SELFRIDGE O., "Pattern Recognition and Learning", in *Proceedings of the Third London Symposium on Information Theory*, Academic Press, New York, pp. 345-353, 1956.
- [Shamos et Hoey 1975] SHAMOS M. I. ET HOEY D., "Closest Point Problems", in *Proceedings of the 16th IEEE Symposium on Foundations of Computer Science*, 151-162, 1975.
- [Sichman, Demazeau et Boissier 1992] SICHTMAN J. S., DEMAZEAU Y. ET BOISSIER O., *When Can Knowledge-Based Systems Be Called Agents ?*, Actes de IX Brazilian Symposium on Artificial Intelligence, Rio de Janeiro, 1992.
- [Simon 1991] SIMON H., *Sciences des Systèmes, Science de l'Artificiel*, Dunod, Paris, 1991.
- [Skinner 1938] SKINNER B. F., *The Behavior of Organisms*, Appleton-Century-Crofts, New York, 1938.
- [Steels 1989] STEELS L., "Cooperation between Distributed Agents through Self-Organisation", in *Decentralized A.I.*, Elsevier North-Holland, Amsterdam, 1989.
- [Steels 1990] STEELS L., "Exploiting Analogical Representations", *Journal of Robotics and Autonomous Systems*, vol. 6, n° 1-2, pp. 71-88, 1990.
- [Steels 1991] STEELS L., "Towards a Theory of Emergent Functionality", in *From Animals to Animats*, MIT Press, Cambridge, pp. 451-461, 1991.
- [Stephens et Merx 1990] STEPHENS L. M. ET MERX M. B., *The Effect of Agent Control Strategy on the Performance of a DAI Pursuit Problem*, Actes de DAI Workshop, Bandera, Texas, 1990.
- [Suchman 1987] SUCHMAN L. A., *Plans and Situated Actions*, Cambridge University Press, Cambridge, 1987.
- [Tabourdeau et Ferber 1993] TABOURDEAU D. ET FERBER J., "Haematopoiesis Simulation with Agents", in *Modelling and Simulation: Proceedings of ESM'93*, Simulation Councils, pp. 109-113, 1993.
- [Theraulaz 1991] THERAULAZ G., *Morphogenèse et Auto-Organisation des Comportements dans les Colonies de Guêpes Polistes Dominilus (Christ)*, Thèse de Troisième Cycle, Université d'Aix-Marseille I, 1991.
- [Theraulaz et al. 1991] THERAULAZ G., GOSS S., GERVET J. ET DENEUBOURG J. L., "Task Differentiation in Polistes Wasp Colonies: a Model for Self-Organizing Groups of Robots", in *From Animals to Animats*, MIT Press, Cambridge, pp. 346-354, 1991.
- [Thierry 1990] THIERRY B., "Feedback Loop between Kinship and Dominance: the Macaque Model", *J. Theor. Biol.*, vol. 145, pp. 511-521, 1990.
- [Thierry 1991] THIERRY B., "Etre ce que l'on Naît", 1991
- [Thom 1983] THOM R., *Paraboles et Catastrophes*, Flammarion, Paris, 1983.
- [Tinbergen 1971] TINBERGEN N., *L'Etude de l'Instinct*, Editions Payot, Paris, 1971.
- [Tinbergen 1979] TINBERGEN N., *La Vie Sociale des Animaux*, Editions Payot, Paris, 1979.
- [Toates et Jensen 1991] TOATES F. ET JENSEN P., "Ethological and Psychological Models of Motivation: Towards a Synthesis", in *From Animals to Animats*, MIT Press, Cambridge, pp. 194-205, 1991.
- [Travers 1988] TRAVERS M., "Animal Construction Kits", in *Artificial Life*, Addison-Wesley, London, pp. 421-442, 1988.
- [Tytrel et Mayhew 1991] TYRRELL T. ET MAYHEW J. E. W., "Computer Simulation of an Animal Environment", in *From Animals to Animats*, MIT Press, Cambridge, pp. 263-272, 1991.
- [Varela 1983] VARELA F., "L'Auto-Organisation: de l'Apparence au Mécanisme", in *Colloque de Cerisy: L'Auto-Organisation, de la Physique au Politique*, Editions du Seuil, Paris, pp. 147-164, 1983.
- [Varela 1989] VARELA F., *Commaître. Les Sciences Cognitives, Tendances et Perspectives*, Editions du Seuil, Paris, 1989.
- [Vauclair 1992] VAUCLAIR J., *L'intelligence de l'Animal*, Editions du Seuil, Paris, 1992.
- [Viaud 1951] VIAUD G., *Les Tropismes*, Que Sais-Je?, Presses Universitaires de France, Paris, 1951.
- [Villa 1992] VILLA F., "New Computer Architectures as Tools For Ecological Thought", *TREE*, vol. 7, n° 6, pp. 179-183, 1992.
- [Volterra 1926] VOLTERRA V., "Variations and Fluctuation of the Number of Animal Species Living Together", *Animal Ecology*, vol. 1926.
- [Von Foerster 1960] VON FOERSTER H., "On Self-Organizing Systems and their Environments", in *Self-Organizing Systems*, Pergamon Press, New York, pp. 31-50, 1960.
- [Von Neumann 1966] VON NEUMANN J., *Theory of Self-Reproducing Automata*, University of Illinois Press, Chicago, 1966.

- [Waldrop 1990] WALDROP M., "Fast, Cheap and Out of Control", *Science*, vol. 248, n° 959-961, 1990.
- [Watson 1919] WATSON J. B., *Psychology from the Standpoint of a Behaviorist*, Lippincott, Philadelphie, 1919.
- [Wavish 1992] WAVISH P., "Exploiting Emergent Behaviour in Multi-Agent Systems", in *Decentralized A.I. 3*, Elsevier North-Holland, Amsterdam, pp. 297-310, 1992.
- [Wavish et Connah 1990] WAVISH P. R. ET CONNAH D. M., "Representing Multi-Agent Worlds in ABLE", Rapport Technique du Philips Research Laboratory (Redhill, Angleterre), n° 2964, 1990.
- [Weber 1964] WEBER M., *L'Ethique Protestante et l'Esprit du Capitalisme*, Plon, Paris, 1964. (publié en 1920 sous le titre *Die Protestantische Ethik und der Geist des Kapitalismus*).
- [Wheeler 1911] WHEELER W. M., "The Ant Colony as an Organism", *J. Morphol.*, vol. 22, pp. 307-325, 1911.
- [Wiener 1948] WIENER N., *Cybernetics, or Control and Communication in the Animal and the Machine*, J. Wiley & Sons, New York, 1948.
- [Wilson 1985] WILSON E. O., "The Sociogenesis of Insect Societies", *Science*, n° 228, pp. 1489-1495, 1985.
- [Wilson 1990] WILSON E. O., *Success and Dominance in Ecosystems: The Case of Social Insects*, Ecology Institute, Oldendorf, 1990.
- [Wilson 1992] WILSON E. O., "Social Insects as Dominant Organisms", in *Biology and Evolution of Social Insects*, Leuven University Press, Leuven, pp. 1-10, 1992.
- [Winston 1984] WINSTON P. H., *Artificial Intelligence*, Addison-Wesley, Reading (Etats-Unis), 1984.
- [Wolfe 1989] WOLFE A., "Whose Keeper ? Social Science and Moral Obligation", 1989
- [Zeghal 1993] ZEGHAL K., "Champs de Forces Symétriques: un Modèle de Coordination d'Actions Réactive Appliqué au Trafic Aérien", Rapport Technique LAFORIA, n° 93/14, 1993.
- [Zeigler 1978] ZEIGLER B. P., *Theoretical Ecological Systems*, Academic Press, New York, 1978.