Coordination despite constrained communications: a satellite constellation case

Grégory Bonnet and Catherine Tessier*

ONERA DCSD/CD
2 avenue Edouard Belin BP 74025
31055 Toulouse Cedex 4

Abstract

In real-world applications where physical agents (such as robots) are used, agents often share information in order to build a common point of view or a common plan. These agents are generally constrained in their communication capabilities and must make decisions without consultation. Consequently, the agents' plans may change without the other agents being aware. In a multi-agent system composed of physical agents, these constraints have a strong influence on the organization and the coordination mechanisms. This paper deals with a satellite constellation, for which we propose a collaboration method based on an incremental coalition formation in order to optimize individual plans so as to satisfy collective objectives. This involves a communication protocol, a common knowledge notion, a definition of trust based on the agents' communication capabilities and two coordination mechanisms: (1) an incentive to join coalitions and (2) coalition minimization. Results on a simulated satellite constellation are presented and discussed.

Keywords

Coalition formation, Coordination and teamwork, Multi-agent systems, Observation satellites.

1 INTRODUCTION

In the multi-agent literature, most of the coordination mechanisms either based on norms [6], contracts [14] or organizations [3, 8] involve *software agents* or *social agents*. In such contexts communications are generally assumed to be unconstrained. As far as *physical agents* such as robots or satellites are concerned, physical and cost constraints have a major impact on communication and therefore on coordination. On the first hand an agent cannot always communicate with another agent or the communications are restricted to short time intervals; on the other hand an agent cannot always wait until the coordination process terminates before acting. Such constraints are present in space applications.

Let us consider satellite constellations i.e. 3 to 20 satellites placed in low orbit around the Earth to take pictures of the ground [4]. Observation requests are generated asynchronously with various priorities by ground stations or the satellites themselves. As each satellite is equipped with a single observation instrument with use constraints, too close requests cannot be realized

*We would like to thank Marie-Claire Charmeau (CNES) and Serge Rainjonneau (Thales Alenia Space) for their comments on this work.

by the same satellite. Likewise, each satellite is constrained in memory resources and can realize only a limited number of requests before downloading[1]. Finally, the orbits of the satellites cross around the poles: two (or more) satellites that meet in the polar areas can communicate *via* InterSatellite Links (ISL) without any ground intervention. So the satellites can communicate from time to time.

Centralized planning is not considered because (1) the aim of future space applications is to avoid using ground stations as much as possible (operating a ground station is expensive); (2) the asynchronous generation of new requests by each satellite prevents us from having a centralized view of the problem and therefore a centralized resolution.

Consequently the problem we focus on is a decentralized task allocation problem in a multi-agent system with new tasks arriving asynchronously and intermittent communications. Each satellite (each agent) builds and revises a task plan such that the number of tasks realized by the constellation is the highest possible, they are realized as soon as possible, the number of redundancies[2] is the lowest possible and the number of high priority tasks that are not realized is the lowest possible. Notice that these constraints are not necessarily compatible with each other. The communication problem was firstly addressed in [2]. In this paper the allocation problem is addressed with an online incremental dynamic organization mechanism in three steps:

1. agents plan individually ;
2. agents communicate in order to build a common knowledge ;
3. agents build and revise coalitions that influence their individual plans.

Our paper is organized as follows. In Section 2, we will first present the multiagent system modelling the satellite constellation. The communication protocol is described in Section 3 and a trust model is proposed in Section 4 in order to evaluate the transmitted pieces of information. In Section 5 we will present our collaboration model through a coalition formation mechanism. Simulation results are given in Section 6 before concluding.

2 THE AGENTS

2.1 The multi-agent system structure

The constellation is modelled as a multi-agent system where each satellite is represented by an agent:

Definition 1 (Constellation) *The* constellation $\mathcal{S}$ *is a triplet* $\langle \mathcal{A}, \mathbb{T}, Vicinity \rangle$ *with* $\mathcal{A} = \{a_1 \dots a_n\}$ *the set of n agents representing the n satellites,* $\mathbb{T} \subset \mathbb{N}^+$ *a set of dates defining a common clock and* $Vicinity : \mathcal{A} \times \mathbb{T} \mapsto 2^{\mathcal{A}}$ *a symmetric non transitive relation specifying for a given agent and a given date the set of agents with which it can communicate at that date (acquaintance model). Vicinity represents the temporal windows when the satellites meet; it is calculated from the satellite orbits, which are periodic.*

Definition 2 (Periodicity) *Let* $\mathcal{S}$ *be a constellation and* $\{p_1 \dots p_n\}$ *the set of the orbital cycle durations* $p_i \in \mathbb{T}$ *of agents* $a_i \in \mathbb{A}$*. The Vicinity period* $\mathring{p} \in \mathbb{T}$ *is the* lowest common multiple *of set* $\{p_1 \dots p_n\}$*.*

The constellation (agents, clock and Vicinity) is knowledge that all the agents hold in common.

[1]Downloading consists in transferring data to a ground station (i.e. the pictures).

[2]There is a redundancy when two different agents realize the same task whereas only one would have been sufficient.

2.2 Tasks

Each agent within the constellation knows some *tasks* to realize.

Definition 3 (Task) *A* task *t is an observation request associated with a priority*[3] $prio(t) \in \mathbb{N}^*$ *and with a boolean b_t that indicates whether t has been realized or not.*

The tasks may be constrained in two ways:

- **mutual exclusion**: it is an agent's constraint meaning that it cannot realize several tasks at the same time τ ;
- **composition** of n tasks: all the n tasks must be realized, it is useless to realize only a strict subset of them. Formally,

  Definition 4 (Compound task) *A* compound task *is a subset $\mathcal{T}$ of tasks such that* $(\exists t_i \in \mathcal{T}, t_i$ is realized$) \Rightarrow (\forall t_j \in \mathcal{T}, t_j \neq t_i$ must be realized$)$.

Moreover when a task is realized by an agent, it is redundant if it has already been realized by another agent:

Definition 5 (Redundancy) *Let a_i be an agent that realizes a task t at time $\tau \in \mathbb{T}$. There is a* redundancy *about t if and only if $\exists\, a_j \in \mathcal{A}$ and $\exists\, \tau' \in \mathbb{T}$ ($\tau' \leq \tau$) such that a_j has realized t at time τ'.*

2.3 Intentions

Each agent within the constellation knows some *intentions* about the tasks.

Definition 6 (Intention) *Let $I_t^{a_i}$ be the* intention *of agent a_i towards task t. $I_t^{a_i}$ is a modality of proposition (a_i **realizes** t) :*

- $\Box$ (commitment)*: a_i is committed to realize t*
- $\Diamond$ (proposal)*: a_i proposes to realize t*
- $\Box\neg$ *(*strong withdrawal*): a_i will not realize t*
- $\Diamond\neg$ *(*weak withdrawal*): a_i does not propose to realize t*

A realization date $rea(I_t^{a_i}) \in \mathbb{T} \cup \{\varnothing\}$ and a download date $tel(I_t^{a_i}) \in \mathbb{T} \cup \{\varnothing\}$ are associated with each intention.

The set of an agent's intentions corresponds to its current plan. Each commitment or proposal means that the associated task is planned. The tasks associated with withdrawals are not planned. We assume that each agent has an individual planner. Planning is a three-step process:

1. From the set of unrealized tasks known by a_i at time τ, a_i computes an optimal local plan under two criteria[4]:
   - maximize the number of planned tasks;

[3] In the space domain, the lower $prio(t)$, the more important task t.

[4] The individual planning process itself is beyond the scope of our work. The mono-agent planning problem may be addressed with many techniques such as constraint programming or HTN planning.

- minimize the number of unplanned high priority tasks.

2. The intentions of agent a_i about tasks t at time $(\tau - 1)$ constrain the planning process (step 1):
   - tasks associated with a commitment ($\Box$) are *always* planned;
   - tasks associated with a strong withdrawal ($\Box\neg$) are *never* planned.
3. Agent a_i's plan at time τ modifies its intentions as follows:
   - each new planned task generates a proposal ($\Diamond$);
   - each new unplanned task generates a weak withdrawal ($\Diamond\neg$).

We can notice that commitments ($\Box$) and strong withdrawals ($\Box\neg$) are not generated by the planning process. We will see in Section 5 that these intentions are generated by a collaboration process.

2.4 Private knowledge

Tasks and intentions an agent knows are captured by knowledge:

Definition 7 (Knowledge) *A* piece of knowledge $K^\tau_{a_i}$ *of agent a_i at time τ is a triplet* $\langle D_{K^\tau_{a_i}}, A_{K^\tau_{a_i}}, \tau_{K^\tau_{a_i}} \rangle$*:*

- $D_{K^\tau_{a_i}}$ *is a task t or an intention $I^{a_k}_t$ of a_k about t, $a_k \in \mathcal{A}$;*
- $A_{K^\tau_{a_i}} \subseteq \mathcal{A}$ *is the subset of agents knowing $K^\tau_{a_i}$;*
- $\tau_{K^\tau_{a_i}} \in \mathbb{T}$ *is the date when $D_{K^\tau_{a_i}}$ was created or updated;*

Let $\mathcal{K}^\tau_{a_i}$ be agent a_i's knowledge at time τ: $\mathcal{K}^\tau_{a_i}$ is the set of all the pieces of knowledge $K^\tau_{a_i}$.

From $\mathcal{K}^\tau_{a_i}$, we define $\mathcal{T}^\tau_{a_i} = \{t_1 \ldots t_m\}$ the set of tasks known by agent a_i at time τ ; and $\mathcal{I}^\tau_{a_i} = (I^{a_k}_t)$ the matrix of the intentions known by agent a_i at time τ. Each agent a_i has resources available to realize only a subset of $\mathcal{T}^\tau_{a_i}$.

3 COMMUNICATION

The agents have to reason on a common knowledge in terms of tasks and intentions. Consequently, a communication protocol is defined to allow an agent to know what the other agents know. Communication is based on Vicinity: when two agents meet they can communicate. Consequently the Vicinity structure influences the communication capabilities.

3.1 Definition

Two kinds of communications are defined:

Definition 8 (Communication) *Let $\mathcal{S}$ be a constellation and $a_i, a_j \in \mathcal{A}$:*

- *Agent a_i can* communicate directly *with agent a_j iff $\exists\ \tau$ within $\mathring{p}$ such that $a_j \in Vicinity(a_i, \tau)$;*
- *Agent a_i can* communicate indirectly *with agent a_j iff $\exists\ \{a_k \in \mathcal{A}, i \leq k < j\}$ and $\exists\ \{\tau_k$ within $\mathring{p}$, $i \leq k < j\}$ such that $a_{k+1} \in Vicinity(a_k, \tau_k)$.*

As Vicinity is symmetric but not transitive, direct communication is symmetric whereas indirect communication is oriented from an agent to another one. Each communication from a_i to a_j is associated with a couple $(\tau_i, \tau_j) \in \mathbb{T}^2$ with τ_i the emitting date of a_i and τ_j the receipt date of a_j. We will write: a_i communicates with a_j at (τ_i, τ_j). In case of a direct communication, $\tau_i = \tau_j$.

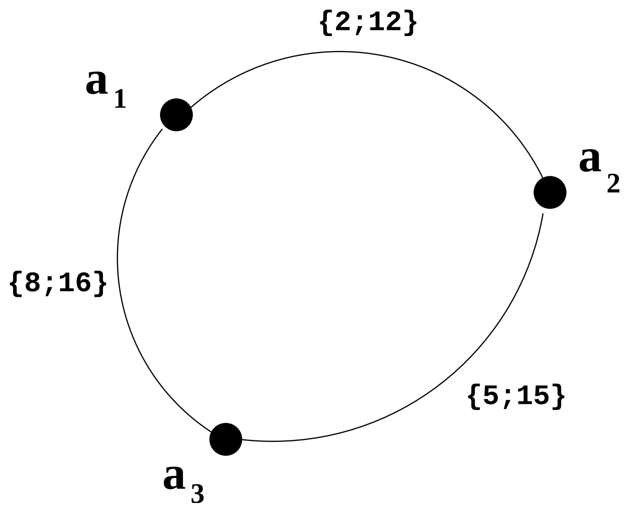

Figure 1: Vicinity graph for Example 1

3.2 Unfolding the Vicinity relation

In order to compute the next indirect communication between two agents, Vicinity is projected on a valued-directed-graph $\mathcal{V}$. Formally,

Definition 9 (Vicinity graph) *The Vicinity graph $\mathcal{V}$ is such that $\mathcal{V} = (\mathcal{A}, (a_i, a_j), v_{ij})$ where:*

- *$\mathcal{A}$ is the set of vertices of $\mathcal{V}$;*
- *the edge (a_i, a_j) exists iff $\exists\ \tau \in \mathbb{T}$ such that $a_j \in Vicinity(a_i, \tau)$;*
- *each edge is valued with the set $v_{ij} = \{\tau \in \mathbb{T} : a_j \in Vicinity(a_i, \tau)\}$.*

Let the following example illustrate this definition.

Example 1 *Let a_1, a_2, a_3 be three agents. Let us suppose that Vicinity is defined as follows on period $\mathring{p} = 20$:*

$$\begin{cases} Vicinity(a_1, 2) = \{a_2\} \\ Vicinity(a_2, 5) = \{a_3\} \\ Vicinity(a_3, 8) = \{a_1\} \\ Vicinity(a_1, 12) = \{a_2\} \\ Vicinity(a_2, 15) = \{a_3\} \\ Vicinity(a_3, 16) = \{a_1\} \end{cases}$$

The vicinity graph is shown on Figure 1.

Intuitively an indirect communication from agent a_i to agent a_j is a path from vertex a_i to vertex a_j. Thereby from this multi-valued graph, we unfold a single-valued graph with respect to the current date and compute the lowest weighted path between both vertices. This single-valued graph is built as it is explored. In order to do that, we propose a modified Dijkstra's algorithm where:

1. the current time τ_i is stored in vertex a_i (initial time plus the weight of the current path);
2. the weight of each edge (a_i, a_j) is computed as follows: $\min v_{ij} - \tau_i [\bmod\ \mathring{p}]$.

Example 2 *Let us resume Example 1 and compute from time* 1 *the next indirect communication from* a_1 *to* a_3*. The weight of edge* (a_1, a_2) *is* 1 *(*$\min(2-1[\bmod 20], 12-1[\bmod 20])$*). The weight of edge* (a_1, a_3) *is* 7*. Thereby, the current time for vertex* a_2 *and* a_3 *are respectively* 2 *and* 8*. A first solution* (a_1, a_3) *has been found: a communication at* $(2, 8)$*. Let us continue the exploration from vertex* a_2*. The weight of edge* (a_2, a_3) *is* 3 *(*$\min(5-2[\bmod 20], 15-2[\bmod 20])$*) and the current time for vertex* a_3 *is* 5*. A better solution has been found: an indirect communication at* $(2, 5)$*.*

3.3 An epidemic protocol

An epidemic protocol based on overhearing [10] has been proposed [2]. The agents use every communication opportunity even to communicate information that does not concern themselves:

1. each agent a_i considers its own knowledge changes;
2. a_i communicates the changes to $a_j \in \text{Vicinity}(a_i, \tau)$;
3. a_j updates its own knowledge thanks to the timestamp $\tau_{K^{\tau}_{a_i}}$.

It has been proved that, in a set of n agents where a single agent knows a piece of information, an epidemic protocol needs $O(\log n)$ communication rounds to completely propagate this information [12]. During a communication round, each agent executes a communication step that has a polynomial complexity in the number of agents and tasks [2].

The agents have to reason on a common knowledge in terms of tasks and intentions. Because of the communication delays, this common knowledge concerns only a subset of agents. Formally,

Definition 10 (Common knowledge) *At time* τ*, agent* a_i *knows that agent* a_j *knows the intention* $I_t^{a_i}$ *captured by* $K^{\tau}_{a_i}$ *iff:*

- $a_j \in A_{K^{\tau}_{a_i}}$ ***or***
- a_i *communicated with* a_j *at* (τ_i, τ_j) *such that* $\tau_{K^{\tau}_{a_i}} \leq \tau_i$*,* $\tau_j \leq \tau$*.*

4 THE TRUST MODEL

Our application can be viewed as an ad-hoc network. However trust literature on ad-hoc networks [11, 20, 23] focus on the reliability of a node in itself and the way to route reliable information. In our application, as agents are trustworthy, trust erosion does not come from the nodes themselves but from interactions between nodes.

4.1 Last confirmation

When two agents communicate at time τ, the agent that receives a given intention cannot be sure that this intention will be the same at time τ' ($\tau' > \tau$). Indeed as the environment is dynamic, an agent may receive new tasks or new intentions and modify its plan, i.e. its own intentions, accordingly. The more time between the generation of a given intention and the realization date, the less an agent can trust this intention. However a further confirmation transmitted by the agent that has generated this intention increases the associated trust again.

As we consider the agents honest and cooperative, an indirect communication (which is a testimony) is trustworthy in itself. Thereby an agent a_i considers that a given proposal generated by an agent a_j has been confirmed if a_j communicates (directly or not) with a_i without modifying its proposal. We define formally the last confirmation.

Definition 11 (Last confirmation) *Let a_i be an agent, $I_t^{a_j}$ a proposal of an agent a_j about a task t known by a_i. The* last confirmation *of proposal $I_t^{a_j}$ for a_i at time τ is:*

$$\tau^* = \max_{\substack{\tau_{K^{\tau}_{a_i}} < \tau_j \\ \tau_i < \tau}} \{\tau_j : a_j \textit{ communicates with } a_i \textit{ at } (\tau_j, \tau_i)\}$$

Example 3 *Let us resume Example 1. Let us suppose that, at time* 15*, a_3 computes the trust associated with an intention of agent a_1 generated at time* 7*. a_1 communicated directly with a_3 at time* 8 *then it communicated indirectly with a_3 at time $(12, 15)$ without modifying its proposal. Thereby the last confirmation is* 12 *and a_3 knows that a_1 kept its proposal between times* 7 *and* 12.

4.2 Trust

Intuitively, the trust associated with a proposal depends on the time between its last confirmation and its realization. As the agents do not have a model of the environment, they cannot predict the arrival of new tasks. However as time passes, an agent meets other agents and each meeting is an opportunity to receive new tasks and revise its intentions. Consequently an agent's trust about a given proposal is defined from the number of meetings between the last confirmation and the realization date. This number is based on Vicinity therefore each agent can compute its own trust in the others' proposals.

Definition 12 (Meetings) *Let a_i be an agent, $I_t^{a_j}$ a proposal known by a_i and τ the current date. Let τ^* be the last confirmation of $I_t^{a_j}$ for a_i at time τ. The number of agents $M_{\tau^*}^{a_i}(I_t^{a_j})$ agent a_j will meet between τ^* and $rea(I_t^{a_j})$ is given by:*

$$M_{\tau^*}^{a_i}(I_t^{a_j}) = |\bigcup_{\tau^* < \tau' < rea(I_t^{a_j})} Vicinity(a_j, \tau')|$$

Finally, an agent trusts or does not trust a given proposal:

Definition 13 (Trust) *Let a_i be an agent, $I_t^{a_j}$ a proposal known by a_i and τ the current date. Agent a_i* trusts *agent a_j about $I_t^{a_j}$ if and only if $M_{\tau^*}^{a_i}(I_t^{a_j}) = 0$.*

Example 4 *Let a_i be an agent that knows a proposal $I_t^{a_j}$ at time τ. Let us suppose that $M_{\tau^*}^{a_i}(I_t^{a_j})$ $= 5$. Agent a_i does not trust a_j about this proposal. Let us suppose that a_j keeps its proposal for long enough to confirm it twice. At each confirmation, a_i can compute $M_{\tau^*}^{a_i}(I_t^{a_j})$ again, e.g.* 3 *and* 1*. Each time, a_i trusts a_j more.*

We can notice that the trust criterion of Definition 13 is hard: an agent is not trusted if it meets at least another agent before realizing its proposal ($M_{\tau^*}^{a_i}(I_t^{a_k}) = 0$). This pessimistic assumption can be relaxed (e.g. $M_{\tau^*}^{a_i}(I_t^{a_k}) \leq 1$).

5 Collaboration via coalitions

5.1 Coalitions

A coalition is an agent organization with a short life cycle. It is formed in order to realize a given goal and is destroyed when the goal is achieved. Through a coalition, each agent tries to maximize its personal outcome. In the literature, the methods dedicated to coalition formation are based on the exploration of the lattice of the possible coalition structures [15, 22]. In order to find the optimal structure, the agents often have uncertain and (or) incomplete information on the other agents' costs and preferences: they need to use heuristics [9] or trust [16] to evaluate a coalition value.

Generally speaking, these methods have two limits.
On the one hand, they are often centralized, they assume that all tasks are known by all agents and they are performed off-line [5, 7, 13, 17] ; or they use an auctioneer (or other kinds of hierarchy) [1, 18] that centralizes the information and organizes the negotiations.
As far as communications are concerned, methods based on the system organization structure consider constrained communications: agents can communicate through a hierarchy [1, 18] or in a vicinity [19]. These constraints are associated with a communication cost [21]. However in a real dynamic environment, agents are not always able to exchange information and may have to decide alone. Moreover some tasks cannot wait for the complete computation of the coalition structure and must be realized quickly. Consequently these methods are very sensitive to the system dynamics.

Be that as it may, the coalition formation mechanisms are interesting for three reasons: (1) agents gather in order to realize a collective task; (2) the short life cycle of coalitions is adapted to dynamic environments; (3) agents search for efficient solutions under uncertain and (or) incomplete information. In our application, compound tasks require that some agents should realize some subsets of tasks jointly. However these joint realizations cannot be planned by the agents' individual planners as an agent does not plan for the others. In order to dynamically organize the agents, we will consider a decentralized coalition formation mechanism taking into account the features of our problem, i.e. cooperative agents and constrained communications. The mechanism is as follows:

1. Agents build maximal-size coalitions from their own knowledge;
2. Coalitions are refined as the agents meet to remove useless agents.

Coalitions are defined as follows:

Definition 14 (Coalition) *A* coalition *C is a triplet $\langle A, O, P \rangle$:*

- *$A \subseteq \mathcal{A}$ is a subset of agents that are the* members *of the coalition;*
- *O is the set of tasks that are the* goals *of the coalition;*
- *P is the set of tasks that are in the* power *of the coalition.*

A coalition C can be in different states:

- *C is* complete *iff $O \subseteq P$;*
- *C is* minimal *iff C is complete and A is minimal for inclusion ($\subseteq$).*

Coalitions are build and managed locally by each agent, given the knowledge it has about the other agents through communication. Indeed each agent uses the coalition notion to reason and adapt its own intentions to the others' intentions. Therefore coalitions are formed implicitly through intentions but are not explicitly built by the multi-agent system. Each agent:

1. computes the current coalition structure from its point of view;
2. checks whether it should join a coalition to increase its power;
3. checks whether it can withdraw from a coalition to minimize it ;
4. modifies its intentions accordingly.

5.2 Computation of the coalition structure

Each agent a_i generates the current coalition structure as follows:

1. a_i organizes the set of tasks $\mathcal{T}_{a_i}^{\tau}$ as a partition $\{\mathcal{T}_1 \ldots \mathcal{T}_h\}$ according to the compound tasks;

   Example 5 *Let $\mathcal{T}_{a_i}^{\tau}$ be $\{t_1, t_2, t_3, t_4, t_5\}$. Let us suppose that tasks t_1 and t_2 form a compound task as well as t_4 and t_5. Then $\mathcal{T}_{a_i}^{\tau}$ is organized as $\{\{t_1, t_2\}, \{t_3\}, \{t_4, t_5\}\}$.*

2. each $\mathcal{T}_i$ is the goal of a single potential coalition; as subsets $\mathcal{T}_i$ are disjoint[5], the number of potential coalitions generated by agent a_i is given by the number of compound tasks a_i knows;

3. from agent a_i's point of view, the potential coalition members for subset $\mathcal{T}_i$ are defined as: $\{a_k \in \mathcal{A} : \exists t \in \mathcal{T}_i \,/\, \exists I_t^{a_k} \in \mathcal{K}_{a_i}^{\tau} \text{ such that } I_t^{a_k} \in \{\Box, \Diamond\}\}$

   Example 6 *Let us resume Example 5. Let us consider t_3 and suppose that $I_{t_3}^{a_i} = \Diamond$ and $I_{t_3}^{a_k} = \Box$. a_i can build coalition $C = \langle \{a_i, a_k\}, \{t_3\}, \{t_3\} \rangle$. This coalition is complete but not minimal because $\{a_i, a_k\}$ is not minimal for inclusion. Notice that a_i plans t_3 even if it knows that a_k did too. Indeed the others' intentions are not taken into account in the planning step: they will be taken into account in the collaboration steps (steps 2, 3, 4 described in Section 5.1).*

4. then the power of each potential coalition is defined as: $P = \{t \in O | \exists a_i \in A : I_t^{a_i} \in \{\Box, \Diamond\}\}$

Notice that this framework defines the current coalition structure from the agent's point of view. It captures *covering* and *disjoint* coalitions: if an agent has many intentions, it can be a member of many coalitions; if it has a single intention (because it lacks resources or capabilities), it is a member of a single coalition.

A potential coalition may be minimal (thus complete), complete and not minimal or incomplete.

5.3 An incentive to join coalitions

An incomplete coalition means that at least one goal task is not within the coalition power. But the more tasks within the coalition power, the more important goal tasks become because a coalition must realize all its goal tasks. If the coalition remains incomplete, all its members waste resources.

When agent a_i computes the current coalition structure according to its knowledge, it can detect incomplete coalitions. As a_i is cooperative, it should be incited to modify its intentions and complete these coalitions when planning. In order to do that, we propose to increase the priorities of the goal tasks of the incomplete coalitions. In the remainder of the paper, we will note $prio(t)'$ the priority of task t a_i uses for its next planning step. Notice that $prio(t)'$ is a local priority only used by a_i. The initial priority $prio(t)$ of task t remains the same.

Protocol 1 (Join a coalition) *For each incomplete coalition $C = \langle A, O, P \rangle$, agent a_i computes: $\forall\, t \in O$, $prio(t)' \leftarrow \frac{prio(t)}{1+|P|}$.*

The agent is encouraged to join a coalition if and only if the goal of the coalition is to realize a compound task that is partially planned. This mechanism is *stable*, i.e. two successive incentive steps are consistent. For instance, an agent is not encouraged to give up a given task in order to realize another one, then *ceteris paribus* is not encouraged to give up the latter to realize the former.

Remark: as far as singletons $\{t\}$ are concerned,

[5]The compound tasks are assumed disjoint but notice that they can overlap without modifying the collaboration process.

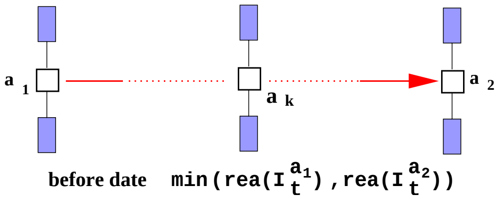

Figure 2: Soft conflict

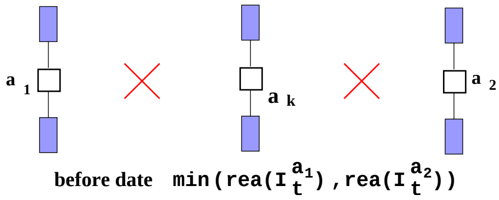

Figure 3: Hard conflict

- if t is not planned by a_i, it is because it does not satisfy the optimization criteria (Section 2.3); therefore a_i does not build any coalition concerning t and the priority of t remains the same;
- if t is planned, the coalition concerning t is complete and its priority remains the same.

Example 7 *Let us resume Example 5. Let us consider* $\{t_1, t_2\}$ *and suppose that* $I_{t_1}^{a_i} = \Diamond\neg$, $I_{t_2}^{a_i} = \Diamond\neg$, $I_{t_1}^{a_k} = \Diamond\neg$ *and* $I_{t_2}^{a_k} = \Box$. a_i *can build coalition* $C = \langle\{a_k\}, \{t_1, t_2\}, \{t_2\}\rangle$. *This coalition is incomplete. So* a_i *applies Protocol 1. As* a_k *is already a member of the coalition, the priorities of* t_1 *and* t_2 *are halved for* a_i. *Therefore at its next planning step,* a_i *is more likely to plan* t_1 *or* t_2 *instead of other tasks.*

5.4 Minimizing coalitions

A complete and non minimal coalition has the power to realize its goals with useless agents, i.e. agents that have redundant intentions. Within a coalition, an agent has to consider the agents that have planned the same tasks as it has, then to make a decision about modifying or not its own intentions. There is a conflict between two agents within a coalition if they have planned the same task(s). Formally:

Definition 15 (Conflict) *Let* a_i, a_j *be two agents and* C *a coalition* $< A, O, P >$ *such that* $\{a_i, a_j\} \subseteq A$. *There is a* conflict *between* a_i *and* a_j *iff* $\exists\, t \in P$ *such that* $I_t^{a_i} \in \{\Box, \Diamond\}$ *and* $I_t^{a_j} \in \{\Box, \Diamond\}$. *It is a* soft conflict *iff either* a_i *communicates with* a_j *at* (τ_i, τ_j) *such that* $\tau_{I_t^{a_i}} < \tau_i$ *and* $\tau_j < \min(rea(I_t^{a_i}), rea(I_t^{a_j}))$ *or* a_j *knows agent* a_i*'s intention about* t. *Else it is a* hard conflict.

Conflicts are illustrated on Figure 2 and Figure 3

Example 8 *Let us resume Example 6. The coalition is not minimal: there is a conflict about task* t_3 *between agents* a_i *and* a_k. *So* a_i *has to make a decision in order to withdraw* ($\Box\neg$)*, to keep its intention* ($\Diamond$) *or to commit* ($\Box$)*.*

In the remainder, given an agent a_i and a task t, we will denote A^* the set of agents with which it is in conflict about task t, $A^+ \subseteq A^*$ the set of agents in soft conflict and $A^- \subseteq A^*$ the set of agents in hard conflict.

Proposition 1 (Symmetry) *Let* a_i *be an agent and a task* t. $\forall\, a_j \in A^+$, *the conflict is* symmetric. $\forall\, a_j \in A^-$, *the conflict is* asymmetric.

Proof 1 *Let* a_i *be an agent and* A^* *the set of agents with which it is in conflict about task* t.

- $\forall\, a_j \in A^+$, a_i *knows* $I_t^{a_j}$. *Conversely either* a_j *knows* $I_t^{a_i}$, *or* $\exists\, \tau_b, \tau_e \in \mathbb{T}$ *such that* $\tau_{I_t^{a_i}} < \tau_b$ *and* $\tau_e < \min\,(rea(I_t^{a_i}), rea(I_t^{a_j}))$ *when* a_j *knows* $I_t^{a_i}$ *because it will receive the information. In both cases, the conflict is symmetric and it is a soft conflict.*

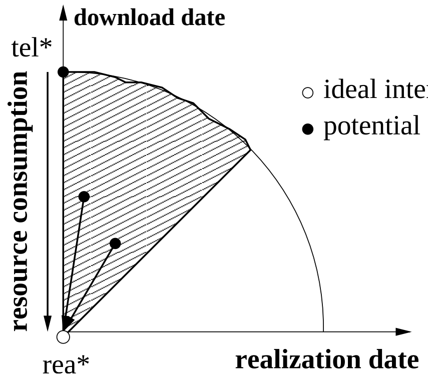

Figure 4: Graphical representation of the expertise criterion

- $\forall\, a_j \in A^-$, a_j *does not know* $I_t^{a_i}$ *and will not know it before the date* $\min\,(rea(I_t^{a_i}), rea(I_t^{a_j}))$. *So,* a_j *is not and will not be aware of the conflict ; it is a hard conflict.*

Both soft and hard conflicts are dealt with through protocols based on the agents' expertise for realizing the task.

As we are seeking to optimize the system reactivity, it is better that the agents realize the tasks as soon as possible and use the fewest resources possible[6]. Let us aggregate both criteria in a single expertise criterion. Formally:

Definition 16 (Expertise) *Let* $A^* \subseteq \mathcal{A}$ *be a set of agents in conflict about a task* t. *Let us note* $rea^* = \min_{a_i \in A^*} rea(I_t^{a_i})$ *the earliest realization date for task* t. *The* expert *agent for* t *is defined thanks to the following distance:*

$$a^* = \arg\min_{a_i \in A^*} ||(rea(I_t^{a_i}) - rea^*, tel(I_t^{a_i}) - rea^*)||$$

The resource consumption, i.e. how long the picture corresponding to t will remain in the memory of the satellite, is defined as a duration. The distance between a potential intention and an ideal intention (the earliest realization and download date) represents time criteria. The expert agent for t is the one that minimizes this distance.

Figure 4 is a representation of the expertise criterion for a task t in the plan $(rea(I_t^{a_i}), tel(I_t^{a_i}))$, $a_i \in A^*$. The origin rea^* is the earliest realization date for t and intention (rea^*, rea^*) is the ideal intention corresponding to an agent being able to realize t at time rea^* and download the corresponding picture immediately. tel^* is the latest download date for t, if t is realized at time rea^*. Obviously $tel(I_t^{a_i}) > rea(I_t^{a_i})$ therefore only the hatched part is meaningful.

In order to solve a conflict, three strategies are defined:

- a_i maintains its proposal ($\Diamond$) if it does not trust the other agents therefore maintaining redundancies to make sure that the task will be realized;

- a_i commits ($\Box$) if it is the expert agent therefore deciding on a part of the current coalition structure;

- a_i strongly withdraws ($\Box\neg$) if the expert agent is trusted thus minimizing the size of the coalition.

[6]Using fewer resources means keeping the pictures in the satellite memory for the shortest time possible, i.e. downloading them as soon as possible.

Protocol 2 (Hard conflict) *Let A^* be the set of the coalition members with which agent a_i is in conflict about task t such that $A^- \neq \emptyset$. a_i is aware of the conflict and applies:*

1. `if` $\min\limits_{a_k \in A^-} M_{\tau^*}^{a_i}(I_t^{a_k}) > 0$ `then` $I_t^{a_i} \leftarrow \Diamond$

2. `else` $I_t^{a_i} \leftarrow \Box\neg$

In case of a hard conflict, the agent who is aware of the conflict (1) maintains its proposal if it does not trust the agents within the conflict ; else (2) withdraws.

Protocol 3 (Soft conflict) *Let A^* be the set of the coalition members with which agent a_i is in conflict about task t such that $A^+ \neq \emptyset$. Let rea^* be $\min\limits_{a_i \in A^+} rea(I_t^{a_i})$:*

1. `if` $a_i = \arg\min\limits_{a_i \in A_+} ||(rea(I_t^{a_i}) - rea^*, tel(I_t^{a_i}) - rea^*)||$
   `then` $I_t^{a_i} \leftarrow \Box$

2. `else` *let a^* be the expert agent:*

   (a) `if` $M_{\tau^*}^{a_i}(I_t^{a^*}) > 0$ `then` $I_t^{a_i} \leftarrow \Diamond$
   (b) `else` $I_t^{a_i} \leftarrow \Box\neg$

For soft conflicts, each agent computes the expert agent. (1) If it is the expert agent, it commits. (2.a) If not, it maintains its proposal if it does not trust the expert. (2.b) If it trusts the expert, it withdraws.

6 SIMULATIONS AND RESULTS

Simulations have been conducted on three kinds of constellations:

- *isolated*: no communication;
- *informed*: agents communicate only about tasks and coordinate *a posteriori* by withdrawing already realized tasks from their plans;
- *coordinated*: agents communicate about tasks and intentions and coordinate *a priori* thanks to coalition formation.

6.1 Reference framework : static simulations

The reference experiments are based on a scenario with 3 agents and 100 tasks. It is a static scenario, meaning that the initial set of tasks is fixed and new tasks will not appear during the simulations. Two parameters are considered: the task density and the task composition rate. For each parameter value, we have launched 100 simulations and computed the average result.

Definition 17 (Density) *The task* density *represents how close to each other the tasks are. The closer the tasks, the more they are likely to be in mutual exclusion.*

(Figure 5) The results for informed and coordinated constellations are better than for isolated constellations. Although informed and coordinated constellations realize nearly the same number of tasks (with a slight advantage for coordinated constellations), coordination allows the number of minimal (i.e. optimal) coalitions to be increased drastically. However we can notice that for coordinated constellations, the difference between minimal and complete coalitions is not so

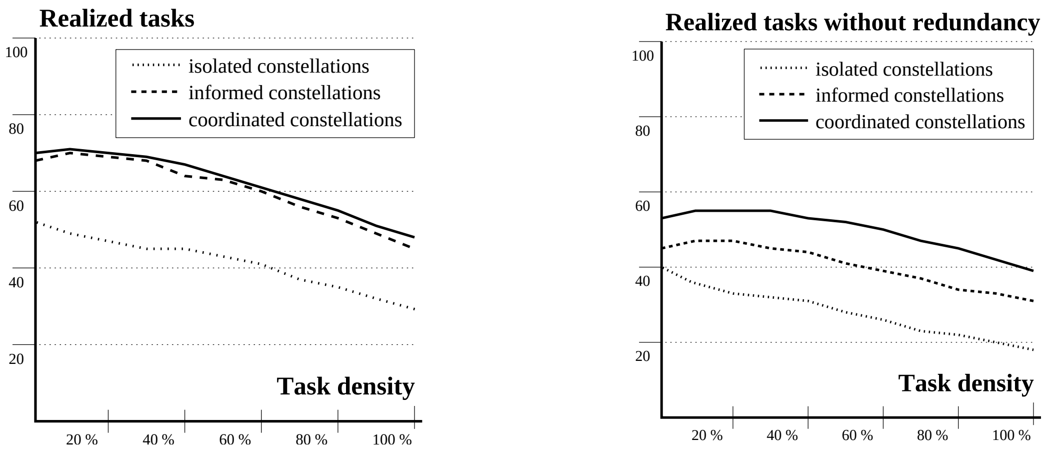

Figure 5: Realized tasks (with and without redundancy) under density constraint

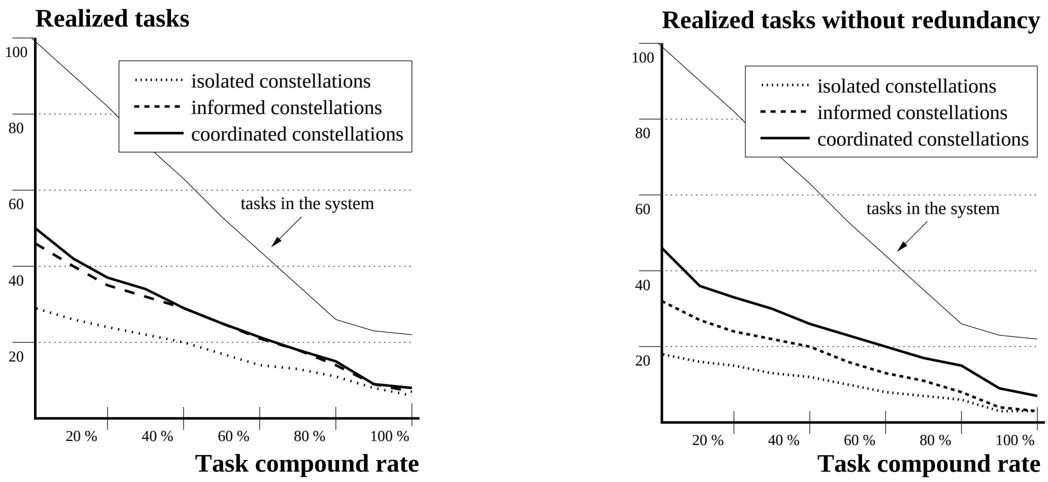

Figure 6: Realized tasks (with and without redundancy) under composition constraint

important: this comes from the fact that these experiments are within a static world, new tasks do not appear during the simulations: when resources are saved by an agent, they are not necessarily reallocated. In a dynamic world with new tasks and no bounded temporal horizon, resources will be reallocated.

Definition 18 (Compound rate) *The task* compound rate *represents the percentage of tasks that are in mutual exclusion with another task and that are jointly the goal of a potential coalition.*

(Figure 6) We can notice that increasing the compound rate decreases the number of potential coalitions, and consequently the maximal number of complete and minimal coalitions. This affects the informed and coordinated constellations more than the isolated ones: the relative loss of efficiency in terms of complete and minimal coalitions is higher. However, the absolute results for informed and coordinated constellations are better than for isolated ones.

6.2 Real-world framework : dynamic simulations

The first simulation round is based on a dynamic scenario with 3 agents. Every 6th hour, the ground stations send 40 new compound tasks (including at least 2 atomic tasks) to the agents. We have launched 25 simulations and computed the average result. Two metrics are considered:

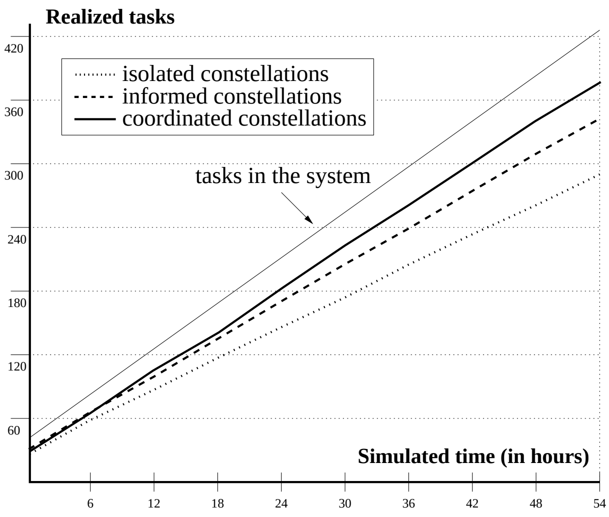

Figure 7: Tasks

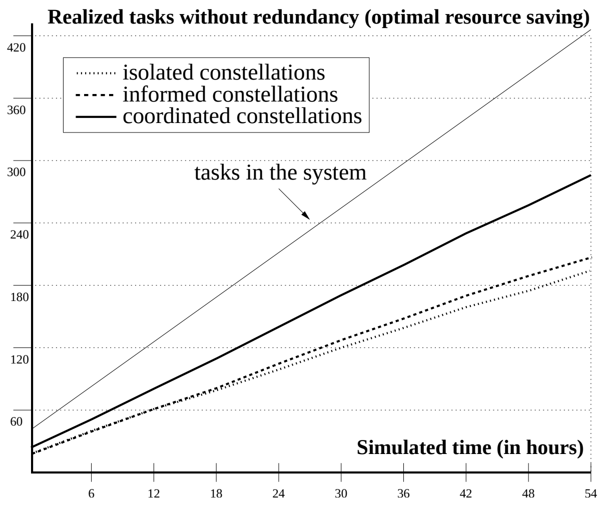

Figure 8: Tasks with no redundancy

the number of realized tasks (Figure 7) and the number of realized tasks without redundancy (Figure 8).

Informed and coordinated constellations outperform isolated ones. However we can notice that the benefits increase as time passes. Indeed incremental coordination allows coordinated constellations to realize more tasks than the other kinds of constellations. And as time passes the difference between informed and coordinated constellations increases: incremental coordination allows coordinated constellations to efficiently save and reallocate resources.

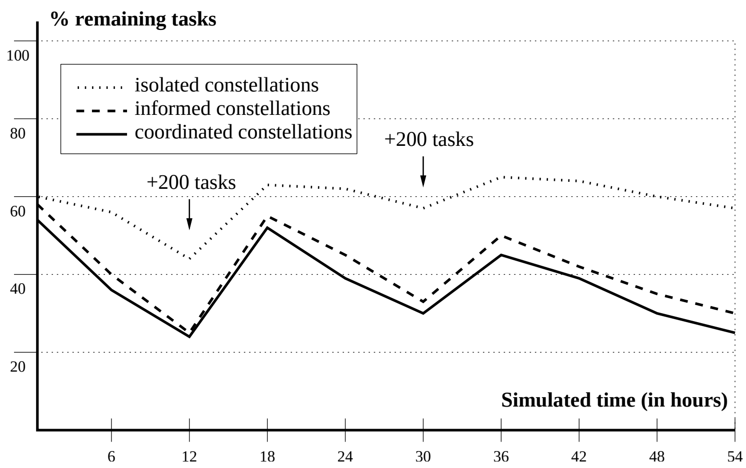

Figure 9: Twofold disturbances

The second simulation round is based on another dynamic scenario with 3 agents. The system is initialized with 150 atomic tasks and two sets of 200 new tasks are sent to the agents after 18 and 36 hours. We have launched 25 simulations and computed the average result. The metric is the percentage of remaining tasks in the system as time passes (Figure 9).

We can notice that isolated constellations are quickly overloaded. They cannot realize all the new tasks and the system load decreases slowly. With coordination, the new tasks are mostly realized and the system load decreases quickly. For coordinated constellations the benefits correspond to saved resources that are reallocated.

7 CONCLUSION

We have proposed a collaboration method for physical agents that communicate from time to time in a dynamic environment. This method has been applied to a constellation of satellites. A communication protocol has been proposed in order to build common knowledge (in terms of tasks and intentions) as the agents meet.

The collaboration process is an online incremental coalition formation that proceeds through a *planning - communication - collaboration* loop within each agent. Each agent builds an initial plan; from its knowledge, it builds the potential coalitions that can realize the tasks it knows; afterwards these coalitions are refined thanks both to an *incentive* mechanism and an *optimization* mechanism. The agents' communication capabilities on the one hand and conflict definitions on the other hand allow us to define protocols that refine the coalition structure dynamically and adapt it to new knowledge.

As new tasks may appear in the system, the agents may revise their plans, that is to say their intentions. However in order to coordinate, the agents must rely on the others' intentions: they must trust them. Thereby we propose a trust notion which is defined through the communications between agents. Each time an agent communicates, it may receive new information that modifies its intentions; on the other hand the more an agent communicates, the more it can confirm its intentions and the more trust may increase.

The experimental results are promising. The coalition formation mechanism allows the resource consumption to be minimized; then the saved resources are reallocated in a incremental way and the number of realized tasks is increased. Future work will deal with the possible failures of the agents and the consequences on the other agents' trusts. Furthermore, simulations involving a higher number of satellite agents (up to 20) will be performed to scale the approach.

References

[1] S. Abdallah and V. Lesser, 'Organization-based cooperative coalition formation', in *Proceedings of the IEEE IAT*, (2004).

[2] G. Bonnet and C. Tessier, 'Collaboration among a satellite swarm', in *Proceedings of the 6th AAMAS*, pp. 287–294, (2007).

[3] C.H. Brooks and E.H. Durfee, 'Congregation formation in multiagent systems', *JAAMAS*, **Vol. 7**, pp. 145–170, (2003).

[4] S. Damiani, G. Verfaillie, and M.-C. Charmeau, 'An Earth watching satellite constellation: How to manage a team of watching agents with limited communications', in *Proceedings of the 4th AAMAS*, pp. 455–462, (2005).

[5] A.R.V. Dung Dang, R.K. Dash, and N.R. Jennings, 'Overlapping coalition formation for efficient data fusion in multi-sensor networks', in *Proceedings of the 21st AAAI*, pp. 635–640, (2006).

[6] F. Dignum, 'Autonomous agents with norms', *Artificial Intelligence and Law*, **Vol. 7**, pp. 69–79, (1999).

[7] H. Goradia and J. Vidal, 'An equal excess negotiation algorithm for coalition formation', in *Proceedings of the 6th AAMAS*, pp. 1052–1054, (2007).

[8] B. Horling and V. Lesser, 'A survey of multi-agent organizational paradigms', *The Knowledge Engineering Review*, **Vol. 19**, pp. 281–316, (2004).

[9] S. Kraus, O. Shehory, and G. Taase, 'Coalition formation with uncertain heterogeneous information', in *Proceedings of the 2nd AAMAS*, (2003).

[10] F. Legras and C. Tessier, 'LOTTO: group formation by overhearing in large teams', in *Proceedings of 2nd AAMAS*, (2003).

[11] A.A. Pirzada and C. McDonald, 'Trust Establishment in Pure Ad-Hoc Networks', *Wireless Personal Communications*, **Vol. 37(1-2)**, pp. 139–168, (2006).

[12] B. Pittel, 'On spreading a rumor', *SIAM Journal of Applied Mathematics*, **Vol. 47**, pp. 213–223, (1987).

[13] S.D. Ramchurn V.D.D.T Rahwan and N.R. Jennings, 'Near-optimal anytime coalition structure generation', in *Proceedings of the 21st IJCAI*, pp. 2365–2371, (2007).

[14] T. Sandholm, 'Contract types for satisficing task allocation', in *Proceedings of the AAAI Spring Symposium: Satisficing Models*, pp. 23–25, (1998).

[15] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé, 'Coalition structure generation with worst case guarantees', *Artificial Intelligence*, **Vol. 111(1-2)**, pp. 209–238, (1999).

[16] I.G.S. Sen and S. Airiau, 'Expertise adn trust-based formation of effective coalitions: an evalutation on the ART testbed', in *Proceedings of the ALAMAS workshop at the 5th AAMAS*, (2006).

[17] O. Shehory and S. Kraus, 'Feasible formation of coalitions among autonomous agents in non-super-additive environments', *Computational Intelligence*, **Vol. 15(3)**, 218–251, (1999).

[18] M. Sims, C.V. Goldman, and V. Lesser, 'Self-organization through bottom-up coalition formation', in *Proceedings of the 2nd AAMAS*, (2003).

[19] L.-K. Soh and C. Tsatsoulis, 'Utility-based multiagent coalition formation with incomplete information and time constraints', in *Proceedings of the IEEE International Conference on SMC*, (2003).

[20] G. Theodorakopoulos and J.S. Baras, 'On Trust Models and Trust Evaluation Metrics for Ad-Hoc Networks', *IEEE Journal on Selected Areas in Communications*, **Vol. 24(2)**, pp. 318–328, (2006).

[21] F. Tohmé and T. Sandholm, 'Coalition formation processes with belief revision among bounded-rational self-interested agents', *Journal of Logic and Computation*, **Vol. 9(6)**, pp> 793–815, (1999).

[22] B Thanh-Tung, B. Frankovic, C. Sheahan, and I. Bundiska, 'Using agent coalitions for improving plan quality', *Intelligent Systems at the Service of Mankind*, **Vol. 2**, pp. 351–364, (2005).

[23] M. Virendra, M. Jadliwala, M. Chandrasekaran and S. Upadhyaya, 'Quantifying Trust in Mobile Ad-Hoc Networks', In *Proceedings of the IEEE International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, pp. 65–71, (2004).