Mesure de qualité des processus de recherche locale stochastique

Mériéma BELAIDOUNI
LGI2P, EMA-EERIE,
Parc Georges BESSE
30000 Nîmes
email: `meriema@site-eerie.ema.fr`

Jin-Kao HAO
LERIA, Université d'Angers,
2, bd. Lavoisier
49045 Angers Cedex 01
email: `Jin-Kao.Hao@univ-angers.fr`

Résumé

Nous proposons une mesure de la qualité des processus de recherche locale stochastique Métropolis sur les instances aléatoires du Problème de Satisfaction Maximale de Contraintes (MAX-CSP). La mesure se base sur la distribution des coûts des processus. Nous l'utilisons pour comparer des relations de voisinage et des instances ou plus généralement des composants de paysages.

Mots clés : Difficulté des Paysages, Recherche Locale Stochastique, MAX-CSP, Métropolis, Distribution des coûts.

1 Introduction

Le concept de paysage a été introduit par Wrigth en 1932 [28]. Depuis, la notion de paysage est devenue l'un des principaux outils d'étude des heuristiques telles que les algorithmes génétiques et la recherche locale [20, 22, 27]. Un paysage est défini par le triplet (S, f, v) où S est l'ensemble des configurations, f la fonction de coût $f : S \to R$ et v une relation de voisinage $v : S \to P(S)$. Deux paysages sont différents si au moins un des trois composants est différent. Etant donnés un paysage et le processus de recherche locale stochastique $\phi_{T=t}$ de vecteur de paramètres T fixé à t, on cherche à caractériser la qualité du processus $\phi_{T=t}$ sur le paysage. Cette question est utile si :

- pour un processus donné, on veut comparer l'efficacité des composants de paysage (relation de voisinage, instance, etc),
- pour un paysage donné, on veut comparer des heuristiques ou des vecteurs de

paramètres T d'une même heuristique (longueur de la liste Tabu, température et palier du Recuit Simulé, par exemple).

L'appréciation de la difficulté d'un paysage pour une heuristique de recherche locale stochastique est habituellement basée sur la comparaison du temps nécessaire pour atteindre une solution optimale [17, 18]. Cette appréciation de la difficulté découle des algorithmes exactes. En effet, ceux-ci cherchent une solution optimale (contrainte) en un minimum de temps (objectif).

Une autre façon d'envisager la mesure de la difficulté consiste à inverser le problème : on fixe le temps d'exécution (contrainte) et on veut trouver la meilleure solution possible (objectif). Ce cadre de mesure de la difficulté peut se révéler intéressant car (a) il correspond davantage au fonctionnement des algorithmes de recherche locale et (b) il permet d'exploiter les valeurs de coûts générées par le processus de recherche locale stochastique [3, 5, 11, 13, 14, 15, 16, 20, 22, 27].

Nous proposons de mesurer la qualité d'un processus de recherche locale c'est à dire son aptitude à évoluer dans le paysage et à atteindre une solution optimale. Cette mesure est basée sur la qualité et la fréquence des configurations générées, autrement dit la distribution des coûts. Nous introduisons une définition pour comparer la difficulté de deux paysages d'un même processus de recherche locale en comparant leurs distributions des coûts.

Rappelons que la distribution des coûts a déjà été utilisée pour l'étude de l'espace des configurations des problèmes combinatoires [2, 6, 24]. Elle a été également utilisée par Schreiber pour comparer les heuristiques locales sur les paysages du problème de bipartition de graphes [23].

Le papier est organisé comme suit : Nous présentons en section 2 le problème MAX-CSP qui est utilisé pour nos expérimentations. Nous introduisons en section 3 la distribution des coûts et la comparaison de la difficulté de deux paysages pour un processus donné. Nous présentons en section 4 une méthode pour mesurer la distribution des coûts de la Méthode Métropolis. En section 5, nous appliquons cette notion de difficulté pour comparer l'efficacité des relations de voisinages et classer la difficulté des instances. Nous concluons et présentons nos perspectives en dernière section.

2 MAX-CSP

Le problème de Satisfaction de Contraintes Maximale (MAX-CSP) est un problème général issu de l'intelligence artificielle. Il permet de modéliser des problèmes tels que le MAX-SAT et la coloration de graphes ainsi que de nombreuses applications réelles telles que l'allocation, la planification et l'ordonnancement.

2.1 Définition du problème

Le problème de Satisfaction de Contraintes Maximale peut-être défini à l'aide de la notion de *réseau de contraintes*. Un réseau de contraintes *CN* est un triplet $< V, D, C >$ où :

- $V = \{V_1, ..., V_n\}$ est un ensemble fini de variables ;
- $D = \{D_1, D_2, ..., D_n\}$ est un ensemble fini de domaines de valeurs associés aux variables ;

- $C = \{C_1, ..., C_n\}$ est un ensemble fini de contraintes binaires. Chaque contrainte est un sous ensemble du produit cartésien des domaines de quelques variables, spécifiant les valeurs interdites.

Étant donné un réseau de contraintes $CN < V, D, C >$, le problème de Satisfaction de Contraintes Maximale consiste à trouver une affectation de valeurs de D aux variables de V telle que le nombre de contraintes satisfaites soit maximal [7]. En pratique, au lieu de maximiser le nombre de contraintes satisfaites, on minimise le nombre de contraintes non-satisfaites, ce qui est strictement équivalent. Par la suite, nous adoptons cette version de minimisation du problème. Ainsi, pour un réseau de contraintes $CN < V, D, C >$, l'ensemble des configurations est l'ensemble des affectations de valeurs aux variables de S. Le coût $f(s)$ de la configuration $s \in S$ correspond simplement au nombre de contraintes violées par s.

A partir d'un réseau de contraintes, d'autres problèmes peuvent être définis. Par exemple, le problème de satisfaction de contraintes (CSP) [19] consiste à trouver une ou plusieurs affectations vérifiant simultanément toutes les contraintes. Il est également possible d'associer un poids à chaque contrainte pour définir le MAX-CSP pondéré (WMAX-CSP pour Weighted MAX-CSP). Ce dernier problème cherche des affectations qui maximisent le poids total des contraintes vérifiées (ou qui minimisent le poids total des contraintes violées). Un MAX-CSP est donc un WMAX-CSP dont les contraintes sont toutes de poids 1. Dans ce travail, nous nous intéressons au problème MAX-CSP, l'approche pouvant également s'appliquer au problème WMAX-CSP.

2.2 Quelques résultats sur l'espace des configurations du MAX-CSP

Dans une étude précédente [4] nous avons développé une étude statistique du problème MAX-CSP. L'un des résultats principaux concerne la distribution des coûts. Nous avons observé que la distribution des coûts peut-être approximée par une loi gaussienne, par conséquent deux mesures peuvent être appliquées sur l'espace des configurations du MAX-CSP :

1. l'espérance des coûts ;
2. l'écart type.

Dans ce papier, nous complétons nos travaux précédants en proposant de nouveaux résultats sur la mesure de qualité des processus de recherche locale stochastique (Métropolis en particulier) sur les paysages MAX-CSP aléatoires. Nous verrons que la normalité des configurations s'étend au processus Métropolis et nous appliquerons ce résultat.

3 Comparaison de paysages

Dans toute la suite les paysages seront issus du problème MAX-CSP dans sa version de minimisation. Considérons deux paysages A et B et le processus de recherche locale $\phi_{T=t}$ avec son vecteur de paramètres T fixé à t (noté $\phi_{T=t}$). Notre but est de comparer la difficulté de A et de B pour le processus $\phi_{T=t}$. La comparaison que nous proposons est basée sur la capacité, en termes de probabilité, du processus $\phi_{T=t}$ à parcourir la distance d'un point repère, ici la moyenne des coûts, à l'optimum des coûts de chaque paysage. Etant donnés deux paysages A et B, A est dit plus facile que B pour le processus $\phi_{T=t}$

si à partir d'un nombre suffisant de mouvements (voir section 4.2) le processus $\phi_{T=t}$ génère de meilleurs résultats sur le paysage A que sur le paysage B. Pour un problème de minimisation, ceci peut arriver dans deux cas :

- $\phi_{T=t}$ génère de meilleurs (plus faibles) valeurs de coût sur le paysage A que sur le paysage B ;
- $\phi_{T=t}$ génère les mêmes valeurs de coût sur les paysages A et B mais plus souvent sur A que sur B.

La définition que nous proposons ci-dessous est basée sur ces deux cas :

3.1 Comparaison de deux paysages pour un processus donné

Définition : Soient deux paysages A et B issus de la même instance (S, f). Le paysage A est dit plus facile que B pour le processus de recherche locale $\phi_{T=t}$ (noté $A \prec_{\phi_{T=t}} B$) s'il existe une valeur de coût f_l telle que pour tout coût inférieur à f_l les résultats de A sont meilleurs que les résultats de B :

$$\begin{array}{ll} \exists f_l, & P_{A,\phi_{T=t}}(F \leq f) > P_{B,\phi_{T=t}}(F \leq f) \\ \forall f < f_l, & P_{A,\phi_{T=t}}(F \leq f) \geq P_{B,\phi_{T=t}}(F \leq f) \end{array} \quad (1)$$

où F est une variable aléatoire discrète définie sur l'ensemble des valeurs de coûts, f et f_l sont des valeurs de coût. $P_{A,\phi_{T=t}}(X \leq x)$ (respectivement $P_{B,\phi_{T=t}}(X \leq x)$) est la cumulative de la distribution des coûts générés par le processus $\phi_{T=t}$ appliqué au paysage A (respectivement B).

3.2 Propriété

La condition suivante, qui utilise uniquement la densité des coûts, est suffisante pour prouver que le paysage A est plus facile que le paysage B pour le processus de recherche locale $\phi_{T=t}$:

$$\begin{array}{ll} \exists f_l, & P_{A,\phi_{T=t}}(F = f_l) > P_{B,\phi_{T=t}}(F = f_l) \\ et & \\ \forall f < f_l, & P_{A,\phi_{T=t}}(F = f) \geq P_{B,\phi_{T=t}}(F = f) \end{array} \quad (2)$$

où F est une variable aléatoire discrète définie sur l'ensemble des valeurs de coûts, f et f_l sont deux valeurs de coût. $P_{A,\phi_{T=t}}(X = x)$ (respectivement $P_{B,\phi_{T=t}}(X = x)$) est la fonction densité des coûts générés par le processus $\phi_{T=t}$ appliqué au paysage A (respectivement B). Les concepts introduits par les équations (1) et (2) sont illustrés en Figure 1.

3.3 Généralisation

A présent considérons le cas ou les paysages sont issus de différentes instances. La propriété (2) peut être généralisée comme suit : Soient A et B deux paysages issus de deux instances (S_A, f_A) et (S_B, f_B) d'un problème de minimisation. A est dit plus facile que B pour le processus $\phi_{T=t}$ si

$$\begin{array}{ll} \exists Cst \in \mathcal{R}, & P_{A,\phi_{T=t}}(F = [\frac{E_A(F)-f_A^*}{Cst})] > P_{B,\phi_{T=t}}(F = [\frac{E_B(F)-f_B^*}{Cst}]) \\ et & \\ \forall Cst' > Cst, & P_{A,\phi_{T=t}}(F = [\frac{E_A(F)-f_A^*)}{Cst'}] \geq P_{B,\phi_{T=t}}(F = [\frac{E_B(F)-f_B^*}{Cst'}]) \end{array} \quad (3)$$

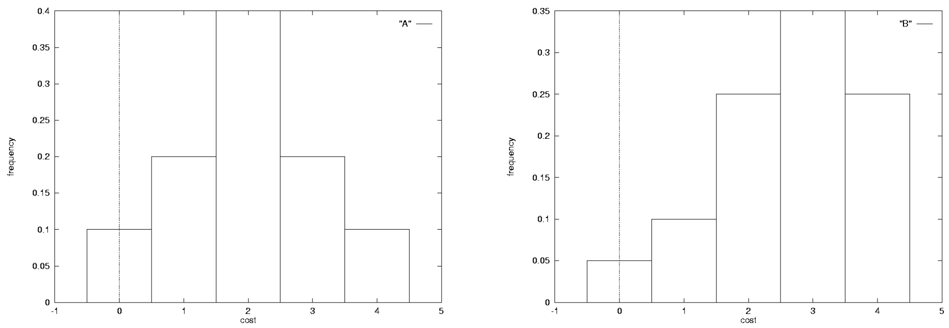


FIG. 1 – Coût en abscisses et fréquences des coûts générés par le processus $\phi_{T=t}$ en ordonnées. Pour un problème de minimisation, le paysage A (gauche) est dit plus facile que le paysage B (droite) pour le processus $\phi_{T=t}(A \prec_{\phi_{T=t}} B)$. Le paysage le plus facile est celui qui a la plus grande chance de réaliser les plus petits coûts ($f \leq 2$). La définition (1) peut être appliquée pour $f_l = 2$.

où $E_A(F)$ et f_A^* (respectivement $E_B(F)$ et f_B^*) sont les espérances et l'optimum des coûts de A (respectivement B) et [.] dénote la partie entière. $P_{A,\phi_{T=t}}(X = x)$ (respectivement $P_{B,\phi_{T=t}}(X = x)$) est la fonction densité des coûts générés par le processus $\phi_{T=t}$ appliqué au paysage A (respectivement B). Notons que l'équation (2) peut être déduite de l'équation (3) en posant $f_l = \frac{E_A(F)-f_A^*}{Cst}$, $E_A(F) = E_B(F)$ et $f_A^* = f_B^*$. Notons également que la procédure de normalisation n'est pas unique. On peut par exemple normaliser par rapport au nombre total de contraintes.

4 Estimation de la distribution des coûts

Etant données les définitions précédentes de la difficulté d'un paysage pour un processus de recherche locale stochastique, il est nécessaire de connaître les distributions des coûts $P_{A,\phi_{T=t}}$ et $P_{B,\phi_{T=t}}$ pour comparer deux paysages A et B. Dans cette section, nous décrivons une technique qui permet d'estimer ces distributions pour le processus Métropolis.

4.1 Estimation de la distribution des coûts d'un paysage

Etant donné le processus de recherche locale ϕ et un paysage A il est possible d'estimer la distribution des coûts de A en utilisant la procédure suivante :

- exécuter r fois le processus $\phi_{T=t}$ durant N mouvements (un mouvement correspond à l'acceptation d'un voisin par le processus, voir description de Métropolis)
- collecter les configurations acceptées pour obtenir l'échantillon E de taille r.N
- compter la fréquence de chaque valeur de coût dans l'échantillon E

A la fin de ce procédé, nous obtenons une estimation de $P_{A,\phi_{T=t}}$.

Cette technique d'estimation peut être utilisée pour tout processus de recherche local tel que Tabu, et la descente. Dans ce papier nous l'utilisons pour l'algorithme de Métropolis que nous décrivons dans la section suivante.

4.2 Algorithme de Métropolis

La méthode de Métropolis a été introduite en 1953 par N. Métropolis *et al.* [21]. Elle est à la base du Recuit Simulé [16] et de la méthode d'approximation de la densité d'états [2, 4]. Le processus de Métropolis se déroule comme suit : on initialise le processus avec une configuration aléatoire s de l'espace de recherche S d'un problème de minimisation, la configuration suivante s' est également prise au hasard parmi les voisins de s et est acceptée avec une probabilité p définie par :

$$p = \begin{cases} 1 & si \quad \triangle f < 0 \\ e^{-\triangle f/T} & si \quad \triangle f \geq 0 \end{cases} \tag{4}$$

où $\triangle f = f(s') - f(s)$.

Algorithme de Métropolis

Data : S ensemble des configurations, f fonction objectif, v relation de voisinage, T température, N taille de l'échantillon, c nombre de valeurs possibles ou nombre de catégories pour f.

```
begin
    choisir une configuration initiale aléatoire s dans S ;
    for i = 0 to i < c do
        Effectif[i] ⟵ 0 ;
    while N > 0 do
        choisir une configuration s' aléatoirement dans v(s), l'ensemble des voisins s ;
        △ ⟵ (f(s') − f(s)) ;
        rd ⟵ choisir un nombre aléatoire entre 0 et 1 ;
        if △f < 0 or rd < e^(−△f/T) then
            s ⟵ s' ;
            N ⟵ N − 1 ;
            Effectif[f(s)] ⟵ Effectif[f(s)] + 1 ;
end;
```

Result : $Effectif$, nombre de configurations par catégorie.

5 Expérimentations et Résultats

Dans cette section, nous appliquons la définition de comparaison des paysages (section 3) et la technique d'estimation de la distribution des coûts (section 4) pour comparer les paysages pour l'algorithme de Métropolis. Afin de maintenir l'étude à un niveau très général, nous choisissons d'étudier les instances aléatoires du problème MAX-CSP (section 2). Le but est de fixer un processus (Métropolis dans notre cas) et de comparer des composants de paysages. Nous examinons deux applications particulières (i) nous évaluons l'efficacité de deux relations de voisinages et (ii) nous classons les instances selon leur difficulté.

5.1 Génération des instances

Les jeux que nous avons utilisés pour mener nos études correspondent à des réseaux de contraintes binaires générés de manière aléatoire. Chaque instance est tirée aléatoirement d'une classe de réseaux qui est définie par la donnée des 4 paramètres $< n, d, p_1, p_2 >$. n représente le nombre de variables du réseau, d le nombre de valeurs dans chacun des domaines, p_1 la densité, *i.e.* le pourcentage de contraintes appartenant effectivement

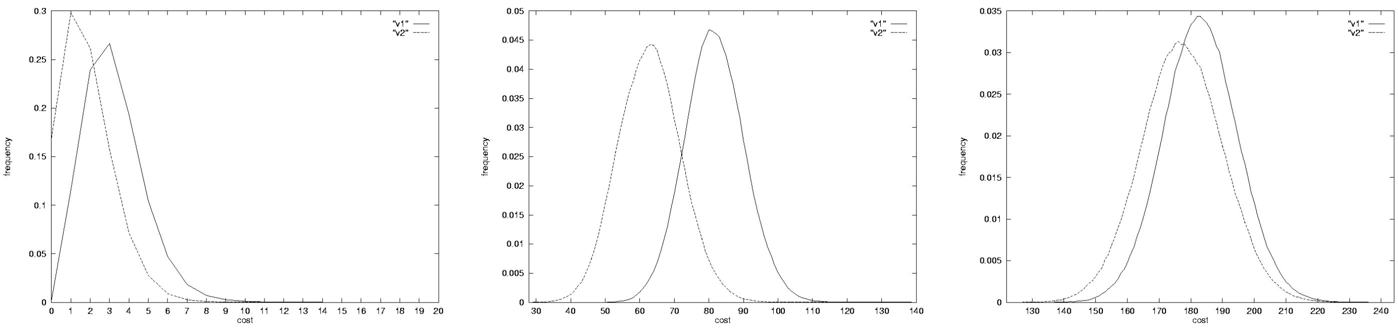


FIG. 2 – Coût en abscisses et fréquence en ordonnées - Comparaison de deux relations de voisinage v_1 et ${}_v2$ pour Métropolis à température $T = 0.2$; 1 et 55 (de gauche à droite) sur les paysages $A = (S, f, v_2)$ et $B = (S, f, v_1)$: v_2 est plus efficace que v_1.

au réseau parmi les $n(n-1)/2$ contraintes possibles, enfin p_2 représente la dureté, *i.e.* le pourcentage de couples interdits sur chacune des contraintes parmi les d^2 possibles. Pour obtenir des instances différentes d'une classe donnée, il suffit de choisir des germes différents.

Un réseau de contraintes ainsi généré peut être sous-contraint ou sur-contraint. Une phase de transition concernant la satisfiabilité apparaît quand le réseau est contraint de manière critique [12] (notion de seuil). Les réseaux sous-contraints ont tendance à être facilement satisfiables ($f = 0$) et ne sont donc pas intéressants pour l'optimisation. Les réseaux sur-contraints sont en général non-satisfiables ($f^* > 0$). Les réseaux au seuil peuvent être satisfiables ou peuvent ne pas l'être. Ces réseaux sont en général difficiles à résoudre du point de vue de la satisfaction.

Ces différentes régions sont caractérisées par un facteur de *seuil* κ, qui est défini dans [9] par :

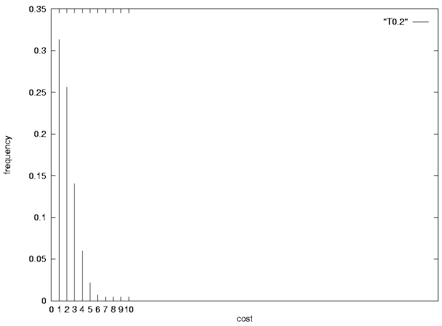
$$\kappa = \frac{n-1}{2} p_1 \log_m\left(\frac{1}{1-p_2}\right) \tag{5}$$

$\kappa = 1$ sépare les réseaux sous et sur-contraints. Les réseaux proches du seuil vérifient $\kappa \approx 1$.

5.2 Protocole d'expérimentation

Pour comparer deux paysages le protocole d'expérimentation est le suivant :

1. choisir deux paysages à comparer ;
2. exécuter Métropolis à température $T = t$ pour N (longueur du processus) suffisamment grand ;
3. associer à chaque valeur de coût sa fréquence dans l'échantillon E pour obtenir la distribution des fréquences ;
4. vérifier la densité en utilisant un test statistique approprié (khi ^{2} dans le cas de la normale). Si nécessaire augmenter la précision de la densité en augmentant la taille de l'échantillon. (Cette densité doit être indépendante de la graine du générateur de nombre aléatoire et de la configuration initiale) ;
5. comparer les deux distributions des coûts en utilisant la définition de la section 3.



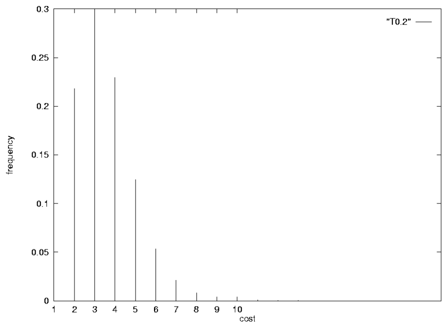


FIG. 3 – Coût en abscisses et fréquence en ordonnées - Distributions des coûts de Métropolis à température $T = 0.2$ sur le paysage A (gauche) et le paysage B (droite).

5.3 Comparaison des relations de voisinages

Dans cette section, nous nous interéssons à la comparaison de paysages définis par (S, f, v_1) et (S, f, v_2). Cette comparaison permet de savoir quelle relation de voisinage est la plus efficace. Nous choisissons de comparer les deux relations suivantes :

- relation de voisinage v_1 : deux configurations sont voisines si elles ne diffèrent que d'une seule variable ;
- relation de voisinage v_2 : deux configurations sont voisines si elles ne diffèrent que d'une variable en conflit.

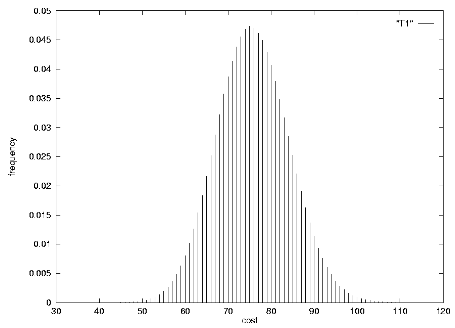
Il est facile de voir que la relation de voisinage v_1 est définie par un voisinage plus grand que v_2, puisque $v_2 \subset v_1$. Intuitivement, on peut croire que la relation de voisinage v_1 étant plus grande, elle donnera de meilleurs résulats. Cependant, les expérimentations montrent que v_2 est en fait plus efficace que v_1 [8, 10]. La comparaison des paysages confirme ce résultat.

Définissons deux paysages $A(I, v_2)$ et $B(I, v_1)$ où I est une instance aléatoire MAX-CSP, I=$<$ 100.10.15%.25%.3 $>$. Nous appliquons la technique d'estimation (section 5.2) à A et B avec l'algorithme Métropolis. Trois processus de Métropolis à température $T = 0.2$; $T = 1$ et $T = 55$ sont réalisés. La Figure 2 donne les distributions des coûts correspondantes. Pour chaque température la distribution des coûts du paysage A est toujours à gauche de celle du paysage B. Le paysage A est donc dit plus facile que le paysage B. Nous concluons que le relation de voisinage v_2 utilisée par A est plus efficace que la relation de voisinage v_1 utilisée par B pour Métropolis sur une instance de la classe $< 100, 10, 15\%, 25\% >$.

Des expérimentations similaires réalisées sur des instances MAX-CSP aléatoires avec des températures de Métropolis différentes ont aboutit au même résultat, ce qui nous permet de conclure que v_2 est plus efficace que v_1 pour Métropolis.

5.4 Comparaison d'instances de même moyenne et de même coût optimal

Nous nous intéressons, maintenant à la comparaison de la difficulté des instances pour Métropolis. Etant données deux instances de même moyenne et de même coût optimal I_1 et I_2, $I_1 =< 100, 10, 25\%, 15\% >$, $I_2 =< 100, 10, 15\%, 25\% >$ $(E_{I_1}(F) = E_{I_2}(F) =$



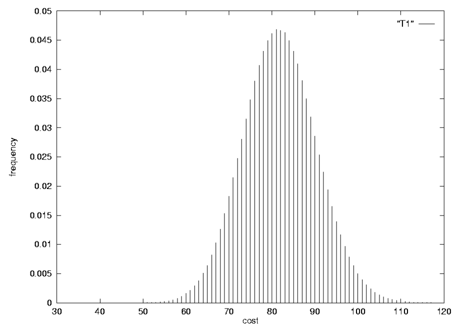


FIG. 4 – Coût en abscisses et fréquence en ordonnées - Distributions des coûts de Métropolis à température $T = 1$ sur le paysage A (gauche) et le paysage B (droite).

$\frac{n.(n-1)}{2}.p_1.p_2 = 186$ [26] et $f^*_{I_1} = f^*_{I_2} = 0$).

Nous appliquons la procédure sur les deux paysages $A = (I_1, v1)$ et $B = (I_2, v1)$, avec Métropolis à température $T = 0.2$. La distribution est stable pour $r = 20$ et $N =$ 1.200.000. Les résultats sont donnés dans la Figure 3. Nous observons que la distribution de A est plus à gauche que la distribution de B. Donc, selon la formule (2) appliquée pour f_l, nous avons :

$$P_A(f_l) > P_B(f_l)$$

et pour $f < 2$:

$$P_A(f_l) \geq P_B(f_l).$$

Nous concluons que le paysage A est plus facile que le paysage B pour Métropolis [1]. Afin de vérifier que cette classification ne dépend pas de la température de Métropolis, nous réalisons l'expérience précédente sur les températures $T = 1$ et $T = 55$. Les résultats sont donnés par les Figures 4 et 5.

De la Figure 4, nous observons que le paysage A est plus à gauche que le paysage B. Selon la définition (Section 3.1) appliquée pour $f_l = 70$ nous constatons que la conclusion "A est plus facile que B" pour Métropolis à température $T = 0.2$ est également vraie pour Métropolis à $T = 1$. Le même conclusion est valide pour $T = 55$, même si la différence entre les deux courbes n'est plus aussi importante (voir Figure 5).

Les expériences présentées ci-dessus ainsi que d'autres expériences similaires à différentes températures, nous permettent de conclure que le paysage A est plus facile que le paysage B pour tout processus Métropolis.

5.5 Comparaison d'instances de moyennes différentes et de même coût optimal

Enfin, nous comparons les paysages dans un contexte encore plus général. Nous utilisons des instances dont les moyennes diffèrent mais pas les solutions optimales ($f^* = 0$). Nous choisissons d'étudier 8 instances de différentes classes (voir colonne 1 de la Table 1). La valeur de κ est donnée à titre indicatif. Les résultats de cette comparaison sont présentés en Table 1 et Figure 6 (annexe).

[1]Les meilleurs résultats obtenus avec une algorithme de descente sont 15 pour I_1 et 18 pour I_2.

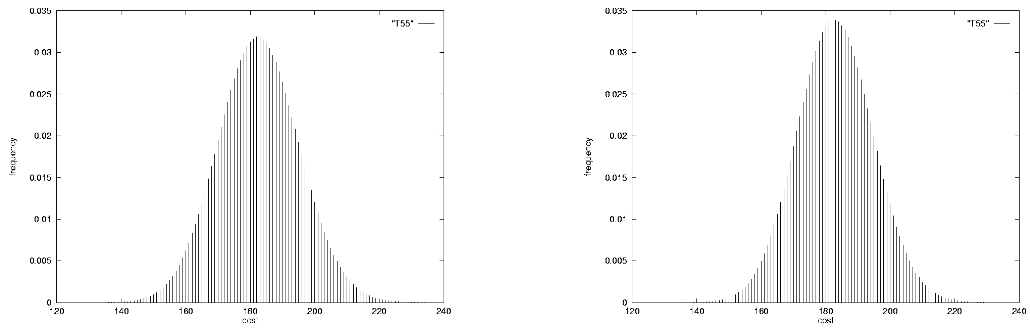


FIG. 5 – Coût en abscisses et fréquence en ordonnées - Distributions des coûts de Métropolis à température $T = 55$ sur le paysage A (gauche) et le paysage B (droite).

Dans ce qui suit, nous notons I au lieu de (I, v_1), de même nous notons $\prec$ au lieu de $\prec_{\phi_{T=t}}$. Afin de classer les paysages selon leur difficulté nous appliquons l'équation (3). Pour le paysage C et D nous appliquons l'équation (3) avec $Cst = 30$, nous obtenons $D \prec C$. Nous procédons de la même façon avec les autres paysages : les constantes que nous utilisons sont données dans la Table 2 (elles ne sont pas uniques). Il s'en suit par application de l'équation (3) la classification suivante : $E \equiv D \prec C \prec H \prec F \prec K \prec G \prec L$.

	Instances	κ	$E(F)$	$P(\frac{E(F)-f^*}{20})$	$P(\frac{E(F)-f^*}{25})$	$P(\frac{E(F)-f^*}{30})$	$P(\frac{E(F)-f^*}{35})$	$P(\frac{E(F)-f^*}{40})$
(C)	$< 100, 10, 10\%, 25\% >$	0.62	124	0.16	0.16	0.14	0.11	0.095
(D)	$< 100, 10, 30\%, 10\% >$	0.68	149	0.125	0.155	0.15	0.13	0.11
(E)	$< 100, 10, 20\%, 15\% >$	0.70	149	0.125	0.155	0.15	0.13	0.11
(F)	$< 100, 10, 15\%, 25\% >$	0.93	186	0.13	0.1	0.07	0.04	0.03
(G)	$< 100, 15, 10\%, 40\% >$	0.93	198	0.095	0.055	0.025	0.015	0.01
(H)	$< 100, 15, 30\%, 15\% >$	0.89	223	0.11	0.13	0.115	0.085	0.07
(K)	$< 200, 20, 18\%, 14\% >$	0.9	501	0.06	0.084	0.048	0.028	0.018
(L)	$< 500, 30, 04\%, 25\% >$	0.19	1247	0.046	0.025	0.003	0.001	8.10^{-6}

TAB. 1 – Distributions des coûts de Métropolis à température $T = 0.3$ sur différents paysages.

paysage 1	D	C	H	F	K	G
paysage 2	C	H	F	K	G	L
Cst	30	20	25	20	25	20

TAB. 2 – Constante (Cst) utilisée dans l'équation (3) pour établir la relation "paysage-1 $\prec$ paysage-2" pour Métropolis à $T = 0.3$.

6 Conclusion et perspectives

Dans ce papier nous utilisons la distribution des coûts pour mesurer la qualité d'un processus de recherche locale stochastique sur un paysage donné. En fixant le processus étudié et en variant les paysages, nous appliquons notre mesure pour comparer l'efficacité des relations de voisinage et la difficulté des instances.

Nous avons développé une technique d'approximation de la distribution des coûts sur un paysage donné. Les expérimentations ont été réalisées avec Métropolis sur des instances aléatoires d'un problème général en optimisation combinatoire : le MAX-CSP. L'approche proposée peut également être utilisée pour (i) évaluer l'impact de différentes formulations d'un problème sur la difficulté de la résolution pour un algorithme donné (ii) expliquer le comportement de l'heuristique de recherche (par exemple, l'effet de la configuration initiale) (iii) régler les paramètres de l'algorithme (iv) évaluer l'impact de l'ajout ou de la simplification de variables et contraintes (v) évaluer l'impact de la fonction de coût. Seules les applications (i) à (iv) ne nécessitent pas de normalisation. L'étude proposée peut être étendue de différentes manières. D'abord, il est possible de l'appliquer pour étudier les paysages structurés, ou d'autres problèmes d'optimisation combinatoire. Des expériences réalisées sur le problème de coloration semblent donner des résultats encourageants. Il est également possible de l'appliquer à d'autres heuristiques de recherche locale. Nos expériences actuelles concernent Tabu. Les processus Tabu semblent avoir un comportement équivalent aux processus Métropolis.

Remerciements : Nous remercions les relecteurs pour leurs remarques intéressantes.

Références

[1] Aarts E., Korst J., Simulated Annealing and Boltzmann Machines, *John Wiley and Sons*, 1989.

[2] Asselmeyer T., Rosé H., Ebeling W., Smoothing Representation of Fitness Landscape - The Genetype - Phenotype Map of Evolution, *Biol. Systems*, 1995.

[3] Belaidouni M., Hao J. K., Landscapes of the Maximal Constraint Satisfaction Problem, *to appear in Lecture Notes in Computer Science*, 2000.

[4] Belaidouni M., Hao J. K., Search Space Analysis of the Maximal Constraint Satisfaction Problem (in French), *5th French Workshop on Practical Solving of NP Complete Problems*, pp. 58-71, 1999.

[5] Boese K. D., Cost Versus Distance In the Travelling Salesman Problem, *Technical Report TR-950018, UCLA Computer Science Department*, May 1995.

[6] Bresina J., Drummond M. and Swanson K., Expected Solution Quality, *Proc. of 14th IJCAI, Morgan Kauffman Publishers*, 1995.

[7] Freuder E. C., Walace R. J., Partial Constraint Satisfaction, *Artificial Intelligence*, vol. 58(1-3), pp. 21-70, 1992.

[8] Galinier P., Hao J. K, Tabu Search for Maximal Constraint Satisfaction Problems, *Principles and Practice of Constraint Programming - CP97, LNCS 1330*, pp. 196-208, 1997.

[9] Gent I. P., MacIntyre E., Prosser P.,Walsh T., Scaling Effects in the CSP Phase Transition, *Proc. of CP95*, pp. 70-87, 1995.

[10] Hao J. K., Pannier J., Simulated Annealing and Tabu Search for Constraint Solving, *Artificial Intelligence and Mathematics IV*, 1998.

[11] Hertz A., Jaumard B., De Aragao M. P., Local Optima Topology for the k-coloring Problem, *Discrete Applied Mathematics*, vol. 49, pp. 257-280, 1994.

[12] Huberman, B. A., Hogg, T., Wiliams C. P., Special Issue on Phase Transition and Complexity, *Artificial Intelligence*, vol. 82, pp. 1-2, 1996.

[13] Jones T., Forrest S., Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms, *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 184-192, 1995.

[14] Naudts B., Kallel L., Some Facts About So Called GA-Hardness Measures, *Technical Report 379, CMAP, Ecole Polytechnique*, March 1998, submitted.

[15] Kauffman S. A., Adaptation on Rugged Fitness Landscapes, *Lectures in the Sciences of Complexity*, SFI Studies in the Sciences of Complexity, D. Stein (Ed.), Addison-Wesley Longman, pp. 527-618, 1989.

[16] Kirkpatrick S., Toulouse G., Configuration Space Analysis of Travelling Salesman Problem, *Physics*, vol.46, pp. 1277-1292, 1985.

[17] Knuth D., Efficiency of Backtrack Programs, *Mathematics of computation*, vol.29(129), pp. 121-136, 1975.

[18] Lobjois L., Résolution de problèmes d'optimisation combinatoire en temps contraint : vers une adaptation automatique des méthodes de résolution, *http ://www.cert.fr/francais/deri/lobjois/these.html*

[19] Macworth A. K., Constraint Satisfaction, *S.C. Shapiro (ed). Encyclopedia on Artificial Intelligence, John Wiley and Sons, NY*, 1987.

[20] Manderick B., de Weger B., Spiessens., The Genetic Algorithm and the Structure of the Fitness Landscape, *PPSN III*, 1994.

[21] Metropolis N., Rosenbluth A. W., Rosenbluth, M. N. and Teller, A. H., Equation of State Calculations by Fast Computing Machines, *The Journal of Chemical Physics*, vol. 21, pp. 1087-1092, 1953.

[22] Reeves C. R., Landscapes, Operators and Heuristic Search, *Annals of Operations Research*, 1999.

[23] Schreiber G. R., Martin O. C., Cut Size Statistics of Graph Bisection Heuristics, , *submitted to SIAM Journal on Optimization*, http ://ipnweb.in2p3.fr/, 1998.

[24] Stadler P. F., The Landscape of the Travelling Salesman Problem, *Physics Letters*,vol. 161,pp. 337-344, 1992.

[25] Tsang E., Foundations of Constraint Satisfaction, *Academic Press*, 1993.

[26] Verfaillie G., Lobjois L., Random CSP : Estimation of optimum (in French), *Research Note*, 23 Mars 1998.

[27] Weinberger E., Correlated and Uncorrelated Fitness Landscapes and How to Tell the Difference, *Biol. Cybern.*, vol. 63, pp. 325-336, 1990.

[28] Wright S., The Roles of Mutation, Inbreeding, and Selection in Evolution, *Proc. of the 6th Congress on Genetics*, vol. 1, pp. 356, 1932.

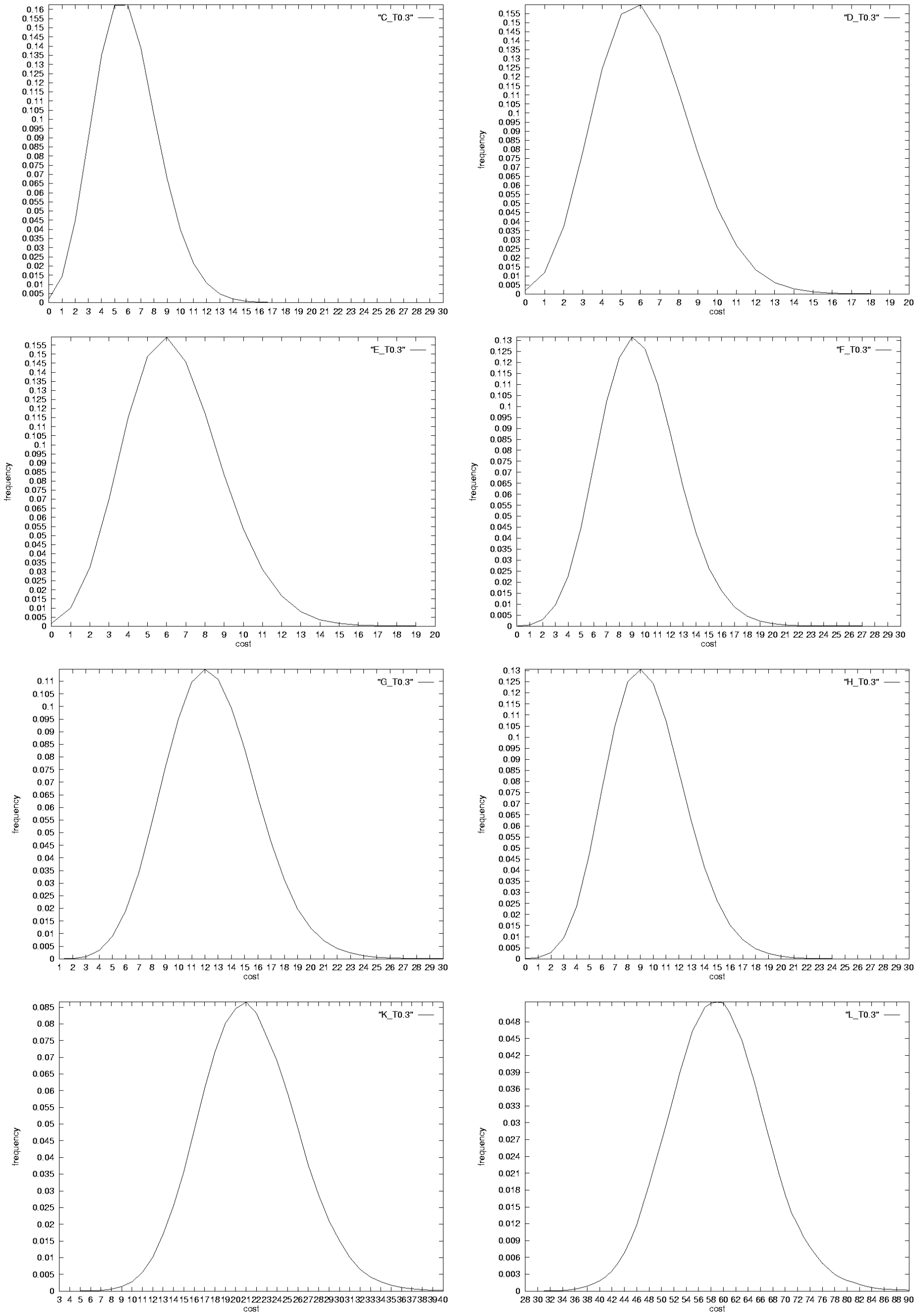


FIG. 6 – Coût en abscisses et fréquence en ordonnées - Comparaison des paysages avec Métropolis ($T = 0.3$) : classification des instances.