Résolution de Max-CSP par Recherche Locale guidée par la Relaxation Lagrangienne

Mohand Ou Idir Khemmoudj, Fayçal Djerourou, Hachemi Bannaceur

Résolution de Max-CSP par Recherche Locale guidée par la Relaxation Lagrangienne*

Mohand Ou Idir Khemmoudj **Fayçal Djerourou** **Hachemi Bennaceur**

LIPN-CNRS UMR 7030 Av J-B. Clément
93430 Villetaneuse
France

{MohandOuIdir.Khemmoudj,Faycal.Djerourou,Hachemi.Bennaceur}@lipn.univ-paris13.fr

Résumé

Nous proposons dans cet article une nouvelle méthode de recherche locale pour la résolution de Max-CSP. Elle combine l'heuristique Minconflict et la relaxation Lagrangienne. La relaxation Lagrangienne guide la recherche vers les régions prometteuses de l'espace tandis que l'heuristique Minconflict explore les voisinages de ces dernières. Les expérimentations préliminaires de notre technique ont donné de bons résultats sur des instances aléatoires de Max-CSP.

Abstract

In this paper, we propose a new local search technique combining Minconflict heuristic and Lagrangean relaxation technique to solve Max-CSP. Lagrangean relaxation guides the search towards the most promising regions of the search space, whereas the Minconflict heuristic is used to explore the neighborhoods of these regions. First experiments of our algorithm have provided good results on random Max-CSP instances.

1 Introduction

Durant ces dernières années, différentes méthodes approchées (incomplètes) ont été proposées pour la résolution de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire. Ces méthodes sont intéressantes dans le cas où l'optimalité n'est pas primordiale ou si la taille du problème est telle qu'il n'est pas possible de le résoudre par une méthode exacte (complète). Elles permettent de fournir des solutions de bonne qualité en un laps de temps raisonnable.

*Ce travail est en partie soutenu par Electricité De France (EDF).

Dans ce papier, nous proposons tout d'abord une nouvelle formulation mathématique pour les Max-CSP binaires. Si cette dernière peut être exploitée dans le cadre d'une méthode exacte, le but de notre travail est de présenter une nouvelle méthode de résolution incomplète que nous avons labellisé **LRGLS** (LRGLS pour Lagrangean Relaxation Guided Local Search). Elle combine une méthode de recherche locale avec la relaxation Lagrangienne [8]. La recherche locale utilisée est une adaptation de la méthode Minconflict [13].

Le principe de LRGLS consiste à résoudre une séquence de problèmes en nombres entiers, IP (IP pour integer programming). Chaque IP définit un espace de recherche (un voisinage) autour d'un optimum local. La résolution de IP est effectuée de manière approchée. Pour cela, un problème plus facile à résoudre (LR) lui est associé. LR est obtenu par la relaxation Lagrangienne d'une partie des contraintes de IP.

La résolution de LR est effectuée par une méthode de sous-gradient [3] combinée avec la méthode de recherche locale Minconflict. A chaque itération du sous-gradient, la méthode de recherche locale Minconflict est appliquée autour de la solution obtenue. La particularité de la méthode Minconflict que nous utilisons est qu'elle exploite les coûts réduits obtenus grâce aux multiplicateurs de Lagrange.

À la fin de l'exécution du sous-gradient et dans le but de diversifier la recherche, un nouveau IP, IP' est formulé. Il est obtenu par la perturbation de certains coefficients de IP. Après un certain nombre de perturbations, la méthode est soit arrêtée, soit réinitialisée avec la dernière solution obtenue.

La méthode que nous proposons peut être appliquée aux CSP valués (VCSP). Cependant et par soucis de clarté, nous nous limitons aux Max-CSP dont le but

consiste à trouver une solution satisfaisant le maximum de contraintes possible[1].

Les expérimentations préliminaires de LRGLS ont donné de bons résultats sur des instances aléatoires de Max-CSP.

La suite de cet article est organisée comme suit : à la section 2, nous présentons le formalisme CSP ainsi que quelques méthodes de résolution approchées pour les Max-CSP. La section 3 est consacrée à la présentation de la formulation mathématique proposée pour les Max-CSP. Dans la section 4, nous présentons la méthode LRGLS. Ensuite, nous présentons les expérimentations réalisées dans la section 5. Enfin, nous concluons à la section 6.

2 Préliminaires

Dans cette section, nous présentons le formalisme CSP ainsi que quelques méthodes de résolution approchées pour les Max-CSP.

2.1 Le problème de satisfaction de contraintes

Le problème de satisfaction de contraintes (CSP pour Constraints Satisfaction Problem) consiste à affecter un ensemble de valeurs à un ensemble de variables assujetties à un ensemble de contraintes. Formellement, un CSP est la donnée d'un quadruplet (X,D,C,R) où :

- X est un ensemble de variables $\{X_1, X_2, \ldots, X_n\}$;
- D est un ensemble de domaines $\{D_1, D_2, \ldots, D_n\}$ où chaque D_i représente les d_i valeurs possibles pour X_i ;
- C est un ensemble de m contraintes où chaque contrainte C_i porte sur un sous-ensemble de variables $C_i = \{X_{i1}, \ldots, X_{in_i}\}$;
- R est un ensemble de m relations où chaque R_i définit l'ensemble des combinaisons de valeurs satisfaisant C_i : R_i est un sous-ensemble du produit cartésien $D_{i1} \times D_{i2} \times \cdots \times D_{in_i}$.

Dans ce travail, nous nous somme intéressés aux problèmes de satisfaction de contraintes binaires. Une contrainte C_{ij} met en relation deux variables X_i et X_j et elle est définie par la relation R_{ij}. Le prédicat $R_{ij}(k,l)$ est vrai si et seulement si le couple (v_k, v_l) est dans R_{ij}. Pour les problèmes auxquels nous nous intéressons, nous considérons que : $(v_k, v_l) \in R_{ij}$ si et seulement si $(v_l, v_k) \in R_{ji}$ (relations symétriques).

Une solution d'un CSP est une instanciation (affectation) satisfaisant toutes les contraintes. Dans de nombreuses situations, le CSP peut être sur-contraint et n'admettre ainsi aucune solution. On peut alors être intéressé par une solution qui minimise le nombre de contraintes violées. Dans la littérature CSP, ce problème est référencé Max-CSP, problème de satisfaction de contraintes maximal.

[1] Pour les Max-CSP, toutes les contraintes ont la même importance (elles ont toutes le même poids).

2.2 Méthodes de résolution approchées pour les Max-CSP

De nombreux travaux ont été consacrés en intelligence artificielle à la résolution du problème Max-CSP par des méthodes approchées. L'heuristique MinConflict [13] est parmi les premières à être proposée. Partant d'une configuration initiale, son principe consiste à l'améliorer de pas en pas en choisissant à chaque étape une variable en conflit, c'est-à-dire une variable impliquée dans une contrainte violée, et l'instancier par la valeur qui minimise le nombre de conflits. Elle est une méthode de réparation n'acceptant aucune dégradation et ne faisant appel à aucune technique lui permettant de ressortir des optima locaux.

Dans le but de permettre de ressortir des optima locaux, la variante *random-walk* [17] de Minconflict a été proposée. Son principe est basé sur l'introduction de mouvements aléatoires : à chaque itération, on effectue un mouvement aléatoire avec une probabilité fixée p et un mouvement de descente avec une probabilité 1 - p.

Dans la littérature, d'autres méthodes approchées plus puissantes, souvent appelées métaheuristiques, ont été proposées. Ces dernières peuvent être scindées en deux groupes : les méthodes évolutives et les méthodes de recherche locale.

Les méthodes évolutives manipulent toute une population de solutions qu'elles font évoluer vers une population ne contenant que de bonnes solutions. Pour cela, elles utilisent des techniques qui s'inspirent de la théorie de l'évolution des êtres vivants. Ces techniques consistent à faire subir à la population des opérations tel le croisement et la mutation. Parmi les méthodes évolutives on peut citer les algorithmes génétiques.

Les algorithmes évolutifs ont été appliqués à de nombreux problèmes de référence et pratiques. Voir [2] pour une liste d'applications.

Quant aux méthodes de recherche locale, elles manipulent des solutions complètes durant la recherche qu'elles essaient d'améliorer par des déplacements élémentaires. Elles font toutes appel à des techniques leur permettant de ressortir des optima locaux.

Parmi les méthodes de recherche locale, on peut citer la recherche Tabou [9] et la recherche locale guidée [16].

Dans [16], les auteurs utilisent la méthode de recherche locale guidée (GLS). Afin de quitter les optima locaux, ils modifient la fonction objectif (somme des contraintes violées) en lui rajoutant une somme

de pénalités, initialement nulle. Chacune des pénalités est associée à une contrainte du Max-CSP. Une pénalité est modifiée (incrémentée de 1) chaque fois que la contrainte qui lui est associée est violée par un optimum local. Dans [4, 7], une recherche tabou a été appliquée avec succès sur plusieurs problèmes réels formalisés en Max-CSP.

D'autres méthodes approchées ont été utilisées pour la résolution du Max-CSP. Citons simplement la méthode de recherche locale avec saut aléatoire GRAPSP [14] et la méthode ACO (pour Ant Colony Optimization) [15].

Dans [17], plusieurs méthodes approchées sont comparées sur des instances aléatoires de Max-CSP binaires. La méthode Minconflict avec l'option *random walk* obtient de bons résultats. Cette méthode est cependant nettement dominiée sur ce type d'instances par un algorithme tabou de base [6].

Pour une introduction plus poussée aux métaheuristiques, Le lecteur peut se référer à [5].

3 Modélisation

Dans la littérature, plusieurs modélisations mathématiques ont été proposées pour les problèmes de satisfaction de contraintes. L'intérêt de ces modélisations est qu'elles ouvrent la voie à des méthodes hybrides issues des deux domaines (techniques mathématiques et CSP).

Dans [1], une modélisation sous forme d'un problème de minimisation sous contraintes d'une fonction quadratique a été proposée et exploitée pour définir le concept de consistance d'arc pondérée (Weighted Arc Consistency (WAC)).

Koster [11] propose un modèle linéaire à variables bivalentes 0-1 pour les CSP valués. Il l'a utilisé pour résoudre de manière efficace certaines instances du problème d'allocation de fréquences.

Nous proposons ici une nouvelle modélisation du Max-CSP sous forme d'un programme linéaire à variables bivalentes. Il utilise un nombre de contraintes réduit par rapport au modèle proposé dans [11].

Nous rappelons tout d'abord la modélisation que nous avons proposé pour les CSP [10], puis nous montrons comment elle peut être généralisée pour représenter les Max-CSP.

Nous commençons la modélisation par l'introduction, pour chaque variable X_i, de d_i variables binaires x_i^k, $k = 1, ..., d_i$ (d_i étant la taille du domaine D_i), telles que :

$$x_i^k = \begin{cases} 1 & \text{si } X_i = v_k \quad (v_k \in D_i) \\ 0 & \text{sinon.} \end{cases}$$

Dans la suite de ce papier, nous noterons par x le vecteur à composantes x_i^k : $x = \left(x_i^k\right)_{i=\overline{1,n}}^{k=\overline{1,d_i}}$.

Pour exprimer le fait que toutes les variables doivent être instanciées par une unique valeur, on écrit :

$$\sum_{v_k \in D_i} x_i^k = 1 \tag{1}$$

Pour exprimer toutes les paires de valeurs incompatibles, nous désignerons par a_i^k ($v_k \in D_i$), le nombre de variables X_j ($j \neq i$) contenant dans leur domaine au moins une valeur incompatible avec la valeur v_k de la variable X_i, $a_i^k = |\{j \neq i / \exists v_l \in D_j \wedge \neg R_{ij}(k,l)\}|$.

On écrit alors :

$$a_i^k x_i^k + \sum_{j \neq i} \sum_{v_l \in D_j, \neg R_{ij}(k,l)} x_j^l \leq a_i^k \tag{2}$$

Cette contrainte signifie que si la variable X_i est instanciée par la valeur v_k, alors aucune variable X_j ($j \neq i$) ne doit être instanciée par une valeur v_l telle que $\neg R_{ij}(k,l)$. Nous désignerons par a, le vecteur à composantes a_i^k, $a = (a_i^k)_{i=\overline{1,n}}^{k=\overline{1,d_i}}$ et par $\psi_i^k(x)$, le terme de gauche de l'inégalité (2),

$$\psi_i^k(x) = a_i^k x_i^k + \sum_{j \neq i} \sum_{v_l \in D_j, \neg R_{ij}(k,l)} x_j^l.$$

En associant à tous les couples (X_i, v_k) une inégalité de type (2), toutes les paires de valeurs incompatibles seront exprimées.

Proposition 1. *L'inégalité (2) est violée si et seulement si $x_i^k = 1$ et s'il existe au moins une variable binaire x_j^l, ($j \neq i$) telle que $\neg R_{ij}(k,l)$ et $x_j^l = 1$. Si $x_i^k = 1$ ($X_i = v_k$),*

$$\eta_i^k = \psi_i^k(x) - a_i^k = \sum_{j \neq i} \sum_{v_l \in D_j, \neg R_{ij}(k,l)} x_j^l$$

est le nombre de contraintes violées parmi celles qui lient la variable X_i à une autre variable.

Démonstration. Soit $I = (v^1, v^2, \ldots, v^n)$ une instanciation du CSP. Le nombre de contraintes violées parmi celles qui lient la variable X_i à une autre variable, $\{C_{ij} \in C : j \neq i\}$ est donné par :

$$\begin{aligned} \sum_{j \neq i, \neg R_{ij}(v^i, v^j)} 1 &= \sum_{j \neq i, \neg R_{ij}(v^i, v^j)} \sum_{v_l \in D_j} x_j^l \\ &= \sum_{j \neq i} \sum_{v_l \in D_j, \neg R_{ij}(v^i, l)} x_j^l \end{aligned}$$

□

À ce stade de la modélisation, nous pouvons affirmer que le CSP est équivalent au problème à variables bivalentes suivant [10] :

$$\begin{cases} \psi_i^k(x) \leq a_i^k & i = \overline{1,n},\ k = \overline{1,d_i} \\ x \in S \end{cases}$$

tel que S est l'ensemble des solutions du système :

$$\begin{cases} \sum\limits_{v_k \in D_i} x_i^k = 1 & i = \overline{1,n} \\ x_i^k \in \{0,1\} & i = \overline{1,n},\ v_k \in D_i \end{cases}$$

Remarque 1. *Le nombre d'inégalités de type (2) nécessaires à la modélisation est borné par nd, où n est le nombre de variables du CSP et d la taille du plus grand domaine.*

Remarque 2. *Dans la formulation ci-dessus, toutes les contraintes du CSP sont doublement exprimées. Par exemple, si on a $\neg R_{ij}(k,l)$, alors l'incohérence de l'instanciation partielle $(X_i = v_k, X_j = v_l)$ sera à la fois exprimée par la contrainte de type (2) associée à la valeur v_k de la variable X_i et par celle associée à la valeur v_l de la variable X_j. Cette redondance peut être évitée par l'ordonnancement des variables du CSP.*

La modélisation peut être généralisée pour représenter les Max-CSP comme suit :

1. On remplace l'inégalité (2) par la suivante :

$$a_i^k\, x_i^k + \sum_{j \neq i} \sum_{v_l \in D_j, \neg R_{ij}(k,l)} x_j^l \quad \leq a_i^k + \eta_i^k \qquad (3)$$

   où η_i^k est une variable positive ;

2. On associe au problème la fonction de coût suivante : $\frac{1}{2} \sum\limits_i \sum\limits_{v_k \in D_i} \eta_i^k$ à minimiser. Le facteur $\frac{1}{2}$ traduit le fait que les contraintes sont doublement représentées (voir la remarque 2).

Considérons maintenant le problème d'optimisation suivant :

$$(IP) \begin{cases} \min \frac{1}{2} \sum\limits_i \sum\limits_{v_k \in D_i} \eta_i^k \\ s.t. \quad \psi_i^k(x) \leq a_i^k + \eta_i^k \quad i = \overline{1,n},\ k = \overline{1,d_i} \\ \qquad (x,\eta) \in S \end{cases}$$

tel que :
- $\psi_i^k(x) \leq a_i^k + \eta_i^k$ est une simple représentation de l'inégalité de type (3) ;
- η est le vecteur à composantes η_i^k, $\eta = \left(\eta_i^k\right)_{i=\overline{1,n}}^{k=\overline{1,d_i}}$;
- S est l'ensemble des solutions du système :

$$\begin{cases} \sum\limits_{v_k \in D_i} x_i^k = 1 & i = \overline{1,n} \\ 0 \leq \eta_i^k \leq a_i^k & i = \overline{1,n},\ v_k \in D_i \\ x_i^k \in \{0,1\} & i = \overline{1,n},\ v_k \in D_i \end{cases}$$

Pour simplifier la présentation, nous désignerons le système (IP) précédent par :

$$(IP) \begin{cases} \min \frac{1}{2} \sum\limits_i \sum\limits_{v_k \in D_i} \eta_i^k \\ s.t. \quad Ax \leq a + \eta \\ \qquad (x,\eta) \in S \end{cases}$$

où $Ax \leq a + \eta$ est le système d'inégalités :

$$\psi_i^k(x) \leq a_i^k + \eta_i^k : i = \overline{1,n},\ k = \overline{1,d_i}.$$

Théorème 1. *Le Max-CSP est équivalent au problème à variables entières (IP).*

Démonstration. Il suffit de montrer qu'à une solution optimale du Max-CSP correspond une solution du système (IP) du même coût et inversement. Soit $I = (v^1, v^2, \ldots, v^n)$, une solution optimale du Max-CSP, $V(I)$ son coût (le nombre de contraintes violées) et considérons le couple de vecteurs $(\bar{x}, \bar{\eta})$ tel que :
- $\bar{x} = \left(\bar{x}_i^k\right)_{i=\overline{1,n}}^{k=\overline{1,d_i}}$: $\bar{x}_i^{v^i} = 1$ et $\bar{x}_i^k = 0, \forall v_k \in D_i - \{v^i\}$;
- $\bar{\eta} = \left(\bar{\eta}_i^k\right)_{i=\overline{1,n}}^{k=\overline{1,d_i}}$: $\bar{\eta}_i^{v^i} = \psi_i^{v^i}(x) - a_i^{v^i}$ et $\bar{\eta}_i^k = 0, \forall v_k \in D_i - \{v^i\}$;

Par construction, le couple $(\bar{x}, \bar{\eta})$ est une solution du système (IP). Comparons maintenant son coût $V(\bar{x}, \bar{\eta})$ avec le coût $V(I)$ de la solution I du Max-CSP. Nous avons :

$$\begin{aligned} V(\bar{x}, \bar{\eta}) &= \tfrac{1}{2} \sum_i \sum_{v_k \in D_i} \eta_i^k \\ &= \tfrac{1}{2} \sum_i \eta_i^{v^i} \end{aligned}$$

Puisque $\eta_i^{v^i}$ est le nombre de contraintes violées parmi celles qui lient la variable X_i à une autre variable (proposition 1) et puisque chaque contrainte est doublement exprimée dans le système (IP), on peut déduire que $\frac{1}{2} \sum\limits_i \eta_i^{v^i} = V(I)$.

Inversement, soit $(\bar{x}, \bar{\eta})$ une solution optimale du système (IP), $V(IP)$ son coût et considérons la solution $I = (v^1, v^2, \ldots, v^n)$ du Max-CSP telle que $\bar{x}_i^{v^i} = 1$; elle est solution optimale. Autrement, le Max-CSP posséderait une solution I' de coût $V(I') < V(I)$ et dans ce cas, le système (IP) posséderait lui aussi une solution de coût $V(I')$ (premier sens de la démonstration). Or, par hypothèse, $V(IP)$ est la valeur optimale de (IP) et par conséquent, $V(I)$ est la valeur optimale du Max-CSP. □

Grâce à cette modélisation, des techniques connues dans le domaine de la recherche opérationnelle et en particulier des techniques de relaxation peuvent être exploitées dans le cadre des Max-CSP.

4 Recherche locale guidée par la relaxation lagrangienne

L'avantage majeur du modèle proposé réside en le petit nombre de ses contraintes. Cependant, même si les contraintes de type (3) qui le constituent sont assez denses (elles contiennent au moins $a_i^k + 1$ variables binaires), elles ne sont pas toujours actives. En effet, si $x_i^k = 0$, alors la contrainte de type (3) associée à la valeur v_k de la variable X_i devient inactive. Ceci constitue un inconvénient pour la résolution. Pour avoir plus de contraintes actives, une solution possible est de revoir la façon de calculer les coefficients a_i^k. Dans le cadre des CSP, on peut envisager par exemple de les remplacer par des meilleures bornes supérieures des termes $\psi_i^k(x) - a_i^k(x)$.

La méthode de recherche locale guidée par la relaxation lagrangienne (LRGLS) tente de fournir de bons coefficients $\tilde{a}_i^k$ qui vont permettre de résoudre efficacement le problème. Cependant, le programme qu'on obtient ne modélise pas exactement le problème d'origine : certaines contraintes du type (3) seront renforcées[2]. Le calcul des coefficients $\tilde{a}_i^k$ suppose toujours que nous disposons d'une solution x du problème (configuration courante[3]). Ils sont tout d'abord initialisés par la procédure **initProblem**.

initProblem

$$\tilde{a}_i^k = \max(\psi_i^k(x) - a_i^k.x_i^k, 1), i = \overline{1,n}, k = \overline{1,d_i}.$$

En remplaçant dans le modèle (IP) le vecteur a des coefficients a_i^k par celui $(\tilde{a})$ obtenu par la procédure **initProblem()**, on obtient un modèle dont l'ensemble des solutions est le même que celui du modèle d'origine[4]. Toutefois, les coûts des solutions ne sont pas les mêmes dans les deux modèles. En effet, si on a par exemple dans le modèle d'origine la contrainte $2 \cdot x_1^1 + x_2^1 + x_3^1 \leq 2 + \eta_1^1$ et si $\tilde{a}_1^1$ est donné égale à 1 par la procédure **initProblem()**, on aura dans le nouveau modèle la contrainte $x_1^1 + x_2^1 + x_3^1 \leq 1 + \eta_1^1$ qui ne sera une "vrai" contrainte du problème que si l'on a $\neg R_{23}(1,1)$. Elle aura pour effet d'augmenter d'une unité le coût de certaines solutions. Ces solutions sont celles qui affectent la valeur 0 à la variable x_1^1. En effet, si $x_1^1 = 1$, le coût de la satisfaction des deux contraintes (la contrainte d'origine et la contrainte renforcée) est le même.

[2]Cela n'est pas permis dans une méthode exacte à moins que toutes les contraintes résultantes ne constituent des coupes valides.

[3]La configuration de départ est obtenue par l'heuristique Minconflict classique.

[4]Toutes les instanciations sont des solutions du Max-CSP. Elles ne diffèrent que par leur coûts.

Soit $(\tilde{IP})$ le modèle obtenu :

$$(\tilde{IP}) \begin{cases} \min \frac{1}{2} \sum_{i=1}^{n} \sum_{v_k \in D_i} \eta_i^k \\ s.t. \quad \tilde{A}x \leq \tilde{a} + \eta \\ \quad\quad (x, \eta) \in S \end{cases}$$

Remarque 3. *Le coût de la solution x utilisée par la procédure* **initProblem()** *pour le calcul des coefficients $\tilde{a}_i^k$ est le même dans les deux modèles, (IP) et $(\tilde{IP})$. En revanche, les solutions qui diffèrent trop de x peuvent avoir un coût beaucoup plus important ; il y a alors moins de chances qu'elles soient considérées. On a donc défini une sorte de voisinage autour de la solution x par un programme en nombres entiers.*

La résolution de $(\tilde{IP})$ peut être effectuée de manière exacte. Toutefois, cela risque d'être coûteux en temps. LRGLS résout donc une relaxation par une méthode de sous-gradient combinée avec la méthode de recherche locale Minconflict adaptée. La relaxation considérée consiste à dualiser les contraintes $\tilde{A}x \leq \tilde{a} + \eta$ de $(\tilde{IP})$. Cette relaxation est intéressante dans le sens où elle permet de séparer le problème en n sous-problèmes indépendants très faciles à résoudre, chacun étant relatif à une variable du Max-CSP.

En notant par u le vecteur des multiplicateurs de Lagrange associés aux contraintes dualisées, le problème lagrangien qu'on obtient est le suivant :

$$(LR(u)) \begin{cases} \min \sum_{i} \sum_{v_k \in D_i} (\eta_i^k + c_i^k(u).x_i^k - u_i^k.a_i^k) \\ s.t. (x, \eta) \in S \end{cases}$$

où $c(u) = A \cdot u$ est le vecteur des coûts réduits associés aux variables binaires $x_i^k, i = \overline{1,n}, k = \overline{1,d_i}$.

Le but de l'algorithme du sous-gradient est d'approximer de manière itérative la solution du problème dual lagrangien (D) : $max_u V(LR(u))$, où $V(LR(u))$ est la valeur optimale du problème lagrangien $LR(u)$.

À chaque itération de l'algorithme du sous-gradient, on dispose d'un couple de vecteurs (u, x), où x est la solution de $(LR(u))$. On dispose aussi du vecteur $c(u)$ des coûts réduits. Au vecteur x correspond une solution X du Max-CSP qui est considérée comme une configuration de départ par la méthode **Minconflict**.

La différence entre l'heuristique Minconflict classique et celle que nous utilisons est que cette dernière, en plus des mouvements qui améliorent la solution, effectue des mouvements qui affectent aux variables de nouvelles valeurs dont les coût réduits sont inférieurs à ceux des valeurs prises actuellement. Ceci à conditions que ces mouvements ne détériorent pas la qualité de la solution.

L'optimum local obtenu par la procédure **Minconflict** est utilisé par la méthode du sous-gradient pour

Minconflict
- **repeat**
  - stop = true;
  - $\forall i = \overline{1,n}$;
    - $v_i = X_i$;
    - $\forall k = \overline{1,d_i}, k \neq v_i$
      - **if** ((**conflicts**$(X_i,k) <$ **conflicts**(X_i,v_i)) **or** (**conflicts**$(X_i,k) =$ **conflicts**(X_i,v_i) **and** $c_i^k < c_i^{v_i}$)) $\{v_i = k, stop = false\}$
  - **until** stop = true.

fournir une nouvelle configuration X et un nouveau vecteur de coûts réduits. Ce processus est répété K fois, K étant un paramètre à régler. À la fin de ces K itérations et dans le but de diversifier la recherche, le modèle ($\tilde{IP}$) est bruité par la procédure noiseProblem.

noiseProblem
- $\forall i = \overline{1,n}, k = \overline{1,d_i}$
  - **if** $\psi_i^k(x) > \tilde{a}_i^k$ **and** $\tilde{a}_i^k < a_i^k$ **then** $\tilde{a}_i^k = \tilde{a}_i^k + 1$;
    else if $\psi_i^k(x) < \tilde{a}_i^k$ **then** $\tilde{a}_i^k = \tilde{a}_i^k - 1$;

Le but de la procédure **noiseProblem** est de diriger la recherche vers d'autres régions de l'espace de recherche. Cette procédure est exécutée $N1$ fois avant que le modèle ne soit initialisé de nouveau par la procédure **initProblem**. La réinitialisation est effectuée du fait qu'après les $N1$ exécutions de la procédure **noiseProblem**, les coefficients $\tilde{a}_i^k$ peuvent devenir très proches des coefficients du modèle d'origine : les contraintes peuvent alors redevenir peu actives. On note $N2$, le nombre de réinitialisations à effectuer avant d'arrêter LRGLS et de retourner la meilleure configuration rencontrée; $N2$ est lui aussi un paramètre à régler.

5 Expérimentations

Cette section présente l'évaluation préliminaire de LRGLS. Les tests ont été effectués sur six classes de Max-CSP aléatoires définies dans [12]. Elles sont données par le tableau suivant :

$1.\langle 10,10,1,p_2\rangle$	$2.\langle 15,5,1,p_2\rangle$
$3.\langle 15,10,50/105,p_2\rangle$	$4.\langle 20,5,100/190,p_2\rangle$
$5.\langle 25,10,37/300,p_2\rangle$	$6.\langle 40,5,55/780,p_2\rangle$

TAB. 1 – Les différentes classes du Max-CSP utilisées dans les expérimentations

Chaque classe du tableau 1 est représentée par un quadruplet $\langle n,d,p_1,p_2\rangle$ tel que n est le nombre de variables du Max-CSP, d la taille du plus grand domaine et p_1 la densité du graphe de contraintes (nombre de contraintes sur le nombre maximum possible de contraintes). Le paramètre p_2 est variable. Il est égal au nombre de couples de valeurs interdits par chaque contrainte.

Les graphes de contraintes des Max-CSP appartenant aux classes (1) et (2) sont complets. La densité des Max-CSP appartenant aux classes (3) et (4) est moyenne, tandis que celle des Max-CSP des classes (5) et (6) est faible.

Pour chaque classe, et pour chaque valeur du paramètre p_2, 50 instances de Max-CSP sont générées et résolues à la fois par LRGLS et par la méthode de résolution PFC-MRDAC[12]. Cette dernière étant une méthode exacte, elle permet de mesurer la qualité des solutions produites par LRGLS. Les paramètres $N1, N2, K$ de LRGLS sont respectivement fixés à 10, 20 et 2.

Les résultats obtenus sont donnés par les figures 1 et 2 (voir l'annexe). La première colonne de ces tableaux représente le paramètre p_2 ; la deuxième et la troisième colonnes donnent la somme, sur les 50 instances, des nombres de contraintes violées respectivement par LRGLS et PFC-MRDAC.

Les résultats obtenus montrent que LRGLS a résolu à l'optimalité la quasi totalité des Max-CSP suffisamment contraints (voir annexe, figure 1). Pour les Max-CSP peu contraints, LRGLS perd un peu de son efficacité (voir annexe, figure 2). Cependant, le pourcentage des problèmes résolus à l'optimalité est supérieur à 85 % et le plus grand écart à l'optimalité est de 3.

6 Conclusion et perspectives

Dans ce papier, nous avons présenté une nouvelle formulation mathématique pour les Max-CSP. Grâce à cette formulation, une nouvelle méthode approchée (**LRGLS**) a été proposée. Elle combine la méthode de recherche locale Minconflict et la relaxation lagrangienne pour guider la recherche vers des optima locaux. Pour quitter ces optima locaux, elle emploie une méthode de bruitage des données.

Les tests préliminaires effectués sur plusieurs instances de Max-CSP ont montré l'efficacité de LRGLS. Ces mêmes tests montrent que LRGLS est plus adaptée aux Max-CSP suffisamment contraints qu'aux Max-CSP peu contraints. Pour ces derniers, les paramètres de LRGLS doivent être encore ajustés.

La modélisation proposée et la méthode LRGLS peuvent être adaptées aux CSP valués (VCSP).

Nos perspectives sont les suivantes :
- Évaluer LRGLS sur des instances de Max-CSP de grande taille et sur des problèmes réels.

– Ajuster de manière plus précise les paramètres de LRGLS pour résoudre plus efficacement les Max-CSP peu contraints ;

Références

[1] Affane, M.S., Bennaceur, H. : A Weighted Arc Consistency Technique for MAX-CSP. ECAI (1998) 209-213.

[2] Bäck, T., Hoffmeister, F., Schwefel H.P. (Eds.) : Applications of evolutionary algorithms. Report of the system Analysis Research Group (Sys), Univ. of Dortmund, 1993.

[3] Camerini, P.M., Fratta, L., Maffioli, F. : On improving relaxation methods by modified gradient techniques. Mathematical programming Study **3** (1975) 26-34.

[4] Dorne, R., Hao, J.K., Galinier, P. : Tabu search for frequency assignment in mobile radio networks. Journal of Heuristics, **(4-1)** (1998) 47-62.

[5] Hao, J.K. : Metaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes. Revue d'Intelligence Artificielle, 13(2) : (1999) 283-324.

[6] Galinier, P., Hao, J. : Tabu search for maximal constraint satisfaction problems. LNCS 1330, Springer-Verlag, (1997) 196-208.

[7] Galinier, P., Hao, J. : A General Approach for Constraint Solving by Local Search, Proceedings of the Second International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR'00), Paderborn, Germany, (2000).

[8] Geoffrion, A.M. : The Lagrangean Relaxation for Integer Programming. Mathematical Programming. **2** (1974) 82-114.

[9] Glover, F., Laguna, M. : Tabu search. Modern heuristic techniques for combinatorial problems. (1993) 70-150.

[10] Khemmoudj, M.O.I., Bennaceur, H., Nagih, A. : Combining Arc-Consistency and Dual Lagrangean Relaxation for Filtering CSPs. CPAIOR 2005, R.Barták and M. Milano (Eds.), LNCS 3524, Springer-Verlag, (2005) 258-272.

[11] Koster, A.M.C. : Frequency Assignment Problem, Models and Algorithms. Proefschrift Universiteit Maastricht, (1999).

[12] Larrosa, J., Meseguer, P., Schiex, T., Verfaillie, G. : Reversible DAC and Other Improvements for Solving Max-CSP". AAAI/IAAI", (1998) 347-352.

[13] Minton, S., Johnston, M.D., Andre, B.P., Laird, P. : Minimizing Conflicts : A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems. Artificial Intelligence. **(58)** (1992) 161-205.

[14] Silva, J., Sakallah, K. : GRASP – A New Search Algorithm for Satisfiability, Technical report, (1996).

[15] Solnon, C. : Ants can solve constraint satisfaction problems. IEEE transactions on Evolutionary computation **6** (2002) 347-357.

[16] Voudouris,C., Tsang, E. : Partial Constraint Satisfaction Problems and Guided Local Search. Proceedings of the Second International Conference on the Practical Application of Constraint Technology (PACT-96). London, (1996) 337-356.

[17] Wallace, R. J. : Analysis of Heuristic Methods for Partial Constraint Satisfaction Problems, Principles and Practice of Constraint Programming, (1996) 482-496.

7 Annexe

classe 1	$\sum$ # de contraintes violées	
# couples interdits	LRGLS	PFC-MPRDAC
80	857	857
81	887	887
82	928	928
83	957	957
84	1003	1003
85	1029	1029
86	1080	1080
87	1110	1110
88	1149	1149
89	1182	1182
90	1231	1231
91	1290	1290
92	1341	1341
93	1378	1378
94	1443	1443
95	1510	1510
96	1566	1566
97	1656	1656
98	1734	1734
99	1850	1850

classe 3	$\sum$ # de contraintes violées	
# couples interdits	LRGLS	PFC-MPRDAC
80	769	768
81	815	814
82	845	845
83	885	884
84	931	931
85	981	981
86	1012	1011
87	1062	1060
88	1106	1105
89	1152	1152
90	1211	1211
91	1263	1263
92	1340	1339
93	1392	1392
94	1466	1464
95	1536	1536
96	1619	1619
97	1704	1704
98	1818	1817
99	1961	1959

classe 2	$\sum$ # de contraintes violées	
# couples interdits	LRGLS	PFC-MPRDAC
10	673	673
11	817	817
12	1006	1006
13	1199	1199
14	1401	1401
15	1618	1618
16	1829	1829
17	2049	2049
18	2309	2309
19	2585	2585
20	2838	2838
21	3142	3142
22	3450	3450
23	3801	3801
24	4268	4268

classe 4	$\sum$ # de contraintes violées	
# couples interdits	LRGLS	PFC-MPRDAC
10	424	424
11	567	565
12	722	722
13	900	900
14	1087	1087
15	1266	1265
16	1465	1465
17	1718	1718
18	1945	1944
19	2194	2194
20	2451	2450
21	2746	2746
22	3083	3083
23	3430	3430
24	3895	3895

FIG. 1 – Résultats de LRGLS et PFC-MPRDAC sur les classes 1-4

classe 5	$\sum$ # de contraintes violées		classe 6	$\sum$ # de contraintes violées	
# couples interdits	LRGLS	PFC-MPRDAC	# couples interdits	LRGLS	PFC-MPRDAC
85	301	283	10	0	0
86	342	319	11	0	0
87	368	349	12	0	0
88	408	387	13	2	2
89	451	437	14	26	9
90	480	472	15	73	55
91	521	511	16	133	117
92	579	570	17	220	199
93	617	607	18	324	305
94	686	678	19	431	416
95	763	749	20	578	570
96	842	831	21	760	750
97	930	925	22	981	972
98	1039	1035	23	1256	1252
99	1199	1186	24	1583	1574

FIG. 2 – Résultats de LRGLS et PFC-MPRDAC sur les classes 5-6