Minorants de problèmes de minimisation de violation de contraintes : recherche de bonnes relaxations à l'aide de méthodes incomplètes

Simon de Givry
Thomson-CSF LCR, laboratoire A&TS
Domaine de Corbeville
91404 Orsay cedex
degivry@lcr.thomson-csf.com

Résumé

Les problèmes de minimisation de violation de contraintes de grande taille sont souvent traités par des méthodes incomplètes. Or, ces méthodes ne donnent aucune garantie sur la distance à l'optimum des solutions trouvées. Nous proposons de calculer des minorants pour évaluer la qualité du résultat d'une méthode incomplète. Notre calcul repose sur une relaxation des contraintes pour obtenir des sous-problèmes de petite taille. La recherche d'une bonne relaxation nous conduit à l'exprimer sous la forme d'un nouveau problème de minimisation que nous traitons par une méthode incomplète. Notre approche s'avère particulièrement efficace sur des problèmes réels d'allocation de fréquences à des liens radio.

1 Introduction

Les problèmes de minimisation de violation de contraintes sont une extension à l'optimisation des Problèmes de Satisfaction de Contraintes (CSP) [Mon74]. Ils considèrent les contraintes comme des préférences pouvant être satisfaites ou non. A chacune est associé un poids. La fonction objectif est l'agrégation des poids des contraintes insatisfaites. Il existe plusieurs formalisations de ces problèmes. Nous les distinguons suivant le type d'opérateur d'agrégation employé :

- Max-CSP [FW92] : opérateur additif et poids égaux à 1 ;
- VCSP [SFV95] : opérateur quelconque, imposant uniquement un ordre total sur les solutions (ex: opérateur logique ($\wedge$), possibiliste (max), lexicographique ou additif) ;
- SCSP [BMR95] : opérateur quelconque, imposant seulement un ordre partiel (ex: opérateur union ensembliste) ;
- PCSP [FW92] : opérateur quelconque ; notion de treillis de relaxations équipé d'une distance.

Ainsi, il devient possible de traiter les CSP sur-contraints, en acceptant de relaxer certaines contraintes. Par la suite, nous ne considérons que des **problèmes avec un opérateur additif**. Notons que ces problèmes de minimisation s'avèrent beaucoup plus difficiles à résoudre qu'en satisfaction [VdG97]. Deux familles de méthodes ont été développées pour résoudre ces problèmes : les méthodes complètes [FW92, Wal94, VLS96, LMSV98] et les méthodes incomplètes [MJPL92, HG95, Wal96, VT96, CVMB96]. En général, nous observons que :

- le temps de résolution des méthodes complètes est inconnu et souvent très grand ;
- la distance à l'optimum des solutions trouvées par les méthodes incomplètes est inconnue ;
- en temps limité, les méthodes incomplètes produisent généralement des solutions de meilleure qualité que les méthodes complètes.

Lorsque le temps de calcul est limité, quelles que soient les méthodes, elles ne donnent aucune garantie sur la qualité des résultats. **Nous proposons de calculer des minorants de l'optimum de problèmes de minimisation de violation de contraintes**. Ces minorants peuvent servir à estimer la qualité du résultat d'une méthode incomplète. Ils peuvent également être utilisés pour orienter les efforts de calcul dans un système ayant plusieurs problèmes à résoudre en temps limité [dG98].
Le calcul de minorants s'obtient en appliquant une des (ou combinant les) stratégies suivantes :

- simplification de la structure du problème consistant à relaxer des contraintes pour obtenir, soit des sous-problèmes de complexité polynomiale (ex: par une décomposition du graphe de contraintes en arbres à l'aide d'une méthode de recherche, soit d'un ensemble coupe-cycle [DP87], soit d'un arbre couvrant de poids maximum [Kru56]), soit des sous-problèmes de petite taille ; la somme des optima des sous-problèmes constitue un minorant ;
- simplification de l'objectif, soit en utilisant un opérateur d'agrégation plus simple [dGVS97], soit en redéfinissant l'objectif comme étant la recherche d'une relaxation minimale localement cohérente [dG98] ;
- modification de la procédure de résolution en effectuant, soit un parcours en largeur (ex: algorithme *Recursive Best First Branch and Bound* [Kor93]), soit un parcours bridé en profondeur d'abord (ex: algorithme *Iterative Deepening* [Kor85], algorithme *Iterative Deepening Russian Doll Search* [CdGV98]).

Dans tous les cas, une méthode complète est nécessaire pour obtenir des minorants. Soit la méthode est modifiée, soit le problème est simplifié de façon à produire un minorant plus rapidement qu'une méthode complète appliquée sur le problème complet.
Dans la section 2, nous donnons une représentation formelle des problèmes. Puis nous étudions deux approches pour rechercher des sous-problèmes de petite taille, avec moins de variables (section 3) ou moins de valeurs (section 4) :

- séparation des variables en composantes faiblement inter-connectées ; les contraintes d'interconnexion sont relaxées ; la somme des optima des composantes est un minorant du problème complet ;
- regroupement des valeurs quasi-interchangeables avec relaxation interne des contraintes par ajout de n-uplets autorisés ; l'optimum du problème relaxé, avec ses domaines réduits, est un minorant du problème complet.

Ces deux relaxations ne garantissent pas de trouver un bon minorant. Néanmoins, le fait de minimiser la relaxation doit y contribuer. Cela nous a conduit à définir chacune des relaxations comme un nouveau problème de minimisation. Nous employons une méthode incomplète pour chercher une bonne relaxation. De plus, nous vérifions que la relaxation est prometteuse en appliquant sur le problème relaxé une méthode incomplète : le majorant obtenu renseigne sur la limite supérieure atteignable par le minorant.
Enfin, dans la section 5, nous validons notre approche par des expérimentations sur des problèmes réels réputés difficiles d'allocation de fréquences à des liens radio.

2 Représentation des problèmes

Nous prenons un formalisme proche de celui des VCSP additifs [SFV97]. Un problème cible P est défini par le quadruplet (V, D, C, φ), tel que :

- V est une liste de N variables ;
- D est une liste de domaines finis ; $D(i)$ dénote le domaine de la $i$ème variable ;
- C est une liste de contraintes, **toutes relaxables** ; une contrainte c est définie par $V(c)$, une liste de variables sur lesquelles porte la contrainte et $R(c)$, une liste de n-uplets autorisés qui définit la contrainte en extension ;

– φ est une fonction qui associe à chaque contrainte un poids (valeur entière) ; soit A, une instanciation complète des variables ; soit $A(i)$, la valeur associée à la $i$ème variable ; soit $A(V(c))$, le n-uplet correspondant à l'affectation suivant A des variables reliées par c ; par extension, $\varphi(A) = \sum_{c \in C / A(V(c)) \notin R(c)} \varphi(c)$ représente le coût de l'instanciation A .

La fonction objectif de P, à minimiser, est la somme des poids des contraintes insatisfaites, c'est-à-dire $\varphi(A)$. Par la suite, nous supposons n'avoir à traiter que des **contraintes binaires**, bien que les relaxations étudiées puissent s'étendre pour le cas de contraintes n-aires.

3 Séparation des variables en composantes faiblement interconnectées

Nous exprimons la relaxation qui consiste à séparer les variables en composantes faiblement interconnectées sous la forme d'un problème de K-partition équilibrée du graphe des contraintes [1].

Définition 1 **K-partition équilibrée de graphe** *:*
étant donné un graphe non orienté $G = (S, A)$ composé de N sommets S et M arêtes A pondérées par un poids, la K-partition équilibrée vise à découper ce graphe en K sous-graphes distincts tels que la somme des poids des arêtes d'interconnexion entre les K sous-graphes soit minimum, et tels que l'écart entre les tailles des K sous-graphes soit minimum (le cas idéal correspondant à N/K sommets par sous-graphes).

Le paramètre K règle le nombre de composantes et donc la taille des sous-problèmes résultants. Le fait de minimiser la somme des poids des contraintes d'interconnexion devrait favoriser l'obtention de composantes fortement connectées, ayant chacune une somme élevée des poids de ses contraintes. Or, une analyse portant sur des VCSP générés aléatoirement montre que l'optimum de ces problèmes croît avec l'augmentation des paramètres de génération suivants : connectivité du graphe de contraintes, dureté des contraintes, ou encore, somme des poids des contraintes [dG98]. Ainsi une bonne K-partition des variables devrait produire une relaxation donnant un minorant élevé.
Nous représentons le problème de la K-partition dans le formalisme des problèmes de minimisation de violation de contraintes. La seule entorse à ce formalisme porte sur la fonction objectif qui devient bicritère. A cette différence près, nous pourrons utiliser la même méthode incomplète pour chercher un bon majorant du problème cible et pour chercher une bonne K-partition.
Soit le problème cible, noté $P = (V, D, C, \varphi)$. Alors, nous définissons le nouveau problème $P' = (V, D', C', \varphi')$, correspondant à une K-partition de P, tel que :

– $D'(i) = \{1, \ldots, K\} \quad \forall i \in [1, N]$; les domaines sont tous identiques ; la valeur d'une variable indique le numéro de la composante dans laquelle est rangée la variable ;

– $C' = \{c' \ / \ c \in C \wedge V(c') = V(c) \wedge R(c') = \{\{1, \ldots, 1\}, \ldots, \{K, \ldots, K\}\}$; les contraintes de P deviennent des contraintes d'égalité dans P' ;

– $\varphi'(c') = \varphi(c)$; les poids des contraintes ne changent pas [2].

La fonction objectif de P', à minimiser, est une somme, pondérée par un facteur h, d'un coût d'interconnexion, égal à la somme des poids des contraintes insatisfaites, et d'un coût de déséquilibre [VM89, Cab96] :

$$E(A) = \varphi'(A) \quad + \quad h \times \sum_{k=1}^{K} \left(\sum_{i=1}^{N} \sigma_{ik}(A) - \frac{N}{K} \right)^2 \quad , \quad \sigma_{ik}(A) = \begin{cases} 1 & \text{si } A(i) = k \\ 0 & \text{sinon} \end{cases}$$

Lors des expérimentations, nous avons utilisé comme méthode incomplète un algorithme de recuit simulé [KGV83, VM89, Cab96]. Brièvement décrit, cet algorithme génère une première instanciation complète A de façon aléatoire. Puis, il effectue une série de modifications locales. Chaque modification, qui correspond à passer de A à A' en changeant aléatoirement la valeur k d'une variable en k', est testée et retenue

1. Les variables constituent les sommets du graphe et les contraintes définissent les arêtes.
2. Notons que la K-partition ne prend pas en compte la dureté des contraintes qui peut être un facteur pour l'obtention d'un minorant élevé. Il faudrait redéfinir le poids d'une contrainte comme une agrégation bicritère de son poids initial et de sa dureté.

suivant une probabilité d'acceptation égale à 1 si $E(A') \leq E(A)$ et à $e^{-\frac{E(A')-E(A)}{T}}$ sinon. Un schéma de décroissance est appliqué à la température T : celle-ci est multipliée par un coefficient m $(0 < m < 1)$ toutes les l tentatives de modification locale. Notons que le calcul de la probabilité d'acceptation se simplifie un peu puisque :

$$E(A') - E(A) = \varphi'(A') - \varphi'(A) \quad + \quad 2 \times h \times (\sum_{i=1}^{N} \sigma_{ik'}(A) - \sum_{i=1}^{N} \sigma_{ik}(A) + 1)$$

4 Réduction des domaines par regroupement de valeurs quasi-substituables

Nous définissons une nouvelle relaxation, liée à une variable x, qui consiste en une réduction du domaine de x et en une modification des poids des contraintes reliées à x. La réduction consiste à regrouper les d valeurs du domaine de x en d' $(1 \leq d' < d)$ paquets de valeurs distinctes et à ne considérer ensuite qu'une seule valeur par paquet. La modification des poids nécessite une nouvelle représentation des problèmes de minimisation de violation de contraintes. La fonction φ ne retourne plus un seul poids pour une contrainte mais différents poids suivant les valeurs des variables reliées par la contrainte. Dans le cas d'une contrainte binaire c portant sur les variables x et y, nous notons $\varphi(c, a, b)$ le poids associé à l'affectation $(x = a, y = b)$. Par extension, le coût d'une instanciation complète A s'écrit $\varphi(A) = \sum_{c \in C} \varphi(c, A(V(c)))$. Cette nouvelle représentation suppose qu'il n'existe au plus qu'une seule contrainte reliant un même ensemble de variables[3]. L'association d'un poids différent par n-uplet existe aussi dans le cadre des CSP flous [DFP93].

Supposons que la relaxation appliquée à la variable x forme d' paquets de valeurs, notés $D_1, \ldots, D_{d'}$. Soit e_k $(1 \leq k \leq d')$, l'unique valeur du paquet D_k retenue après réduction du domaine de x. Dans ce cas, la modification des poids due à la relaxation équivaut à :[4]

$$\forall k \in [1, d'] \; \forall c \in C \;/\; V(c) = (x, y) \; \forall b \in D(y), \quad \varphi(c, e_k, b) \leftarrow \min_{a \in D_k} \varphi(c, a, b)$$

Ainsi, toute affectation complète du problème cible, avec $x = a$ $(a \in D_k, 1 \leq k \leq d')$, a un coût supérieur au coût de la même affectation, avec $x = e_k$, dans le problème relaxé. L'optimum du problème relaxé, avec ses domaines réduits, est bien un minorant du problème cible.

Il se peut que la valeur e_k soit telle que la relaxation liée à la réduction de D_k ne fasse aucune modification dans φ. Dans ce cas, la valeur e_k est dite voisinage-substituable aux valeurs de D_k [Fre91]. Si le même phénomène intervient pour tous les paquets du domaine de x, alors l'optimum du problème relaxé est également l'optimum du problème cible. De cette observation, nous en déduisons un critère d'optimisation pour rechercher une bonne relaxation. Nous exprimons le regroupement des valeurs du domaine de x en d' paquets sous la forme d'un nouveau problème de minimisation dont la fonction objectif est :[4]

$$\sum_{k=1}^{d'} \min_{e \in D_k} \left(\sum_{c \in C / V(c) = (x,y)} \; \sum_{b \in D(y)} \left(\varphi(c, e, b) - \min_{a \in D_k} \varphi(c, a, b) \right) \right)$$

Il s'agit donc de trouver une décomposition des valeurs du domaine de x en d' paquets, telle que la somme des coûts des paquets, c'est-à-dire la somme des modifications dans φ, soit minimale. Pour chaque paquet, la valeur e retenue après réduction du domaine est celle qui minimise la somme des modifications dans φ. Si cette valeur est voisinage-substituable aux autres, alors le coût du paquet est nul.

Notons que le nombre de partitions de d objets en d' classes, égal au nombre de Stirling de 2ième espèce [Ber71], devient vite très grand. Heureusement, notre problème de minimisation peut être résolu optimalement par un simple algorithme glouton. Il suffit de réduire le domaine de un en un, en appliquant à chaque fois la modification des poids, jusqu'à obtenir d' paquets. A chaque réduction, l'algorithme teste toutes les paires de deux valeurs et retient la meilleure.

---

3. Notons qu'il y a équivalence entre cette nouvelle représentation et celle décrite en section 2. Soit le problème $P = (V, D, C, \varphi)$ exprimé dans l'ancienne représentation et $P' = (V, D, C', \varphi')$, le même problème, exprimé dans la nouvelle représentation, alors $\forall c' \in C' \; \forall (a, b) \in D(V(c')), \quad \varphi'(c', a, b) = \sum_{c \in C / V(c) = V(c') \wedge (a,b) \notin R(c)} \varphi(c)$.

4. Pour des raisons de lisibilité, la formule est simplifiée : il faudrait distinguer le cas où $V(c) = (x, y)$, du cas où $V(c) = (y, x)$.

La démonstration que la solution est optimale s'effectue par récurrence. C'est vrai à l'étape d et $d-1$. Supposons que la solution est optimale à l'étape $d-i$. En appliquant la relaxation à cette étape, le domaine ne contient alors que $d-i$ valeurs. La relaxation, qu'elle soit faite à chaque étape ou uniquement à la fin, modifie φ de la même façon, du fait de la monotonie de l'opérateur min. La meilleure paire de valeurs conduit à une nouvelle solution optimale pour l'étape $d-i-1$ □
Soit g, le degré maximum d'une variable ; soit d, la taille du plus grand domaine ; soit $s = d - d'$, le nombre d'étapes de l'algorithme glouton ; alors sa complexité est en $O(s \times g \times d^3)$ appels à la fonction φ.

En pratique, il est nécessaire de réduire le domaine de toutes les variables pour obtenir un problème relaxé abordable par une méthode complète. Or, les relaxations associées à des variables reliées par des contraintes ne sont pas indépendantes. Ainsi, une relaxation effectuée d'abord sur x, puis sur y peut aboutir à des regroupements et des coûts de relaxation différents que si la relaxation avait été faite d'abord sur y puis sur x. C'est pourquoi nous utilisons un autre algorithme glouton, sans garantie d'optimalité. Nous réduisons progressivement le domaine de chaque variable en parcourant les variables de façon cyclique. A chaque étape, toutes les variables ont leurs domaines réduits de un. Il est préférable d'ordonner les variables par les moins contraintes en premier car ainsi les premières relaxations ont des coûts plus faibles et peuvent favoriser la relaxation des variables les plus contraintes. Par contre, une approche qui consisterait, à chaque étape de l'algorithme, à réduire uniquement la variable dont le coût de la relaxation est minimum n'aboutit pas à une bonne relaxation en final : cela a tendance à réduire seulement les variables les moins contraintes.

5 Expérimentations sur des problèmes réels du CELAR

Le problème, proposé par le Centre d'Electronique de l'Armement (CELAR), consiste à affecter des fréquences à un ensemble de liens de communication radio définis entre paires de sites, en évitant les interférences. Chaque lien de communication est représenté par une variable dont le domaine correspond à l'ensemble des fréquences disponibles (au maximum, 44 fréquences) pour le couple émetteur-récepteur. Entre deux liens de communication proches, notés F_1 et F_2, il y a une contrainte de non interférence du type $|F_1 - F_2| > cste$. La constante dépend de la proximité des deux liens et aussi de l'environnement géographique.
Parmi les onze instances, formant le benchmark RLFAP [CdGL+98], générées à partir d'un unique réseau réel du CELAR, nous ne considérons que les instances 6,7 et 8 qui sont insatisfiables et pour lesquelles aucun minorant n'a été fourni lors du projet européen CALMA [THL95b, THL95a]. Ces instances s'expriment sous la forme de problèmes de minimisation de violation de contraintes avec un opérateur d'agrégation additif. Toutes les contraintes sont relaxables, avec un poids choisi parmi quatre valeurs. Le tableau suivant indique pour chaque instance, son nombre de variables [5], son nombre de contraintes, ses quatre poids, la somme des poids de ses contraintes, le meilleur majorant connu et le plus mauvais majorant , sur 10 exécutions (avec le temps moyen), trouvé par la méthode *Guided Local Search* [VT98] :

CELAR	$\|V\|$	$\|C\|$	φ	$\sum_{c \in C} \varphi(c)$	*best ub*	*ub* **[VT98]**
6	100	1222	1,10,100,1000	255193	3389	3415 (15 min)
7	200	2665	1,100,10000,1000000	468527293	343592	354205 (18 min)
8	458	5286	1,2,3,4	12710	262	285 (16 min)

Pour l'instance 6, nous avons effectué une 6-partition, produisant 6 composantes contenant entre 15 et 21 variables chacune. L'algorithme de recuit simulé ($h = 100, m = 0.9, l = 50000$) (cf. section 3) a mis 80 secondes pour obtenir cette décomposition de coût d'interconnexion égal à 16725 et de coût de déséquilibre égal à 2933. Les composantes obtenues ne sont pas toutes connexes. Une décomposition en composantes connexes les sépare en 11 composantes. Le même algorithme de recuit simulé (sans le coût de déséquilibre) est appliqué sur les 11 composantes pour produire un bon majorant. Ce majorant permet d'accélérer l'algorithme complet *Russian Doll Search* [VLS96]. Le tableau suivant indique pour chaque composante connexe, son nombre de variables, son nombre de contraintes, un majorant (avec le temps mis par le recuit) et son optimum (avec le temps mis par l'algorithme *Russian Doll Search*, lorsque le majorant n'est pas nul) :

5. Il existe une astuce pour diminuer le nombre de variables des instances par deux [CdGL+98]. Nous avons appliqué cette réduction à chaque fois. Le nombre de variables, ainsi que le nombre de contraintes, tiennent compte de cette réduction. De plus, cette réduction entraine l'apparition de plusieurs contraintes de poids différents portant sur les mêmes paires de variable, ce qui est favorable à la relaxation décrite en section 4.

k'	$\|V\|$	$\|C\|$	*ub*	*optimum*
1	18	229	159 (37 sec)	159 (259 sec)
2	6	10	0 (0.1 sec)	0
3	14	300	2737 (280 sec)	2669 (27540 sec)
4	9	76	0 (4 sec)	0
5	21	188	0 (9 sec)	0
6	14	91	0 (2 sec)	0
7	13	78	110 (103 sec)	110 (0.2 sec)
8	1	0	0 (0 sec)	0
9	1	0	0 (0 sec)	0
10	2	4	0 (0.1 sec)	0
11	1	0	0 (0 sec)	0
Total =	100	976	3006 (435 sec)	2938 (27799 sec)

Ainsi, nous obtenons un minorant de l'instance 6 égal à 2938 en environ 8 heures de calcul sur un SUN UltraSparc 167Mhz [6]. Pour comparaison, l'algorithme *Iterative Deepening Russian Doll Search* produit un minorant égal à 2338 en 18 heures [CdGV98].
Pour tenter d'améliorer ce minorant, nous avons exécuté d'autres 6-partitions avec des facteurs de pondération plus faible (h variant entre 10 et 50), de manière à détecter des composantes plus intéressantes mais aussi plus dures à résoudre. Nous avons ainsi généré une nouvelle série de composantes, notés *CELAR6-SUBi* [CdGL+98], à partir de l'instance 6 du CELAR : $\forall i \geq 1$ *CELAR6-SUBi* $\subset$ *CELAR6-SUBi+1* , *CELAR6-SUB0* $\cap$ *CELAR6-SUB4* $= \emptyset$. Le tableau suivant résume ces composantes qui ont toutes été résolues complètement par l'algorithme *Russian Doll Search* (temps de calcul indiqué dans le tableau) :

CELAR6-SUB*i*	$\|V\|$	$\|C\|$	*optimum*	*temps*
CELAR6-SUB0	16	207	159	107 sec
CELAR6-SUB1	14	300	2669	7 h
CELAR6-SUB2	16	353	2746	5 h
CELAR6-SUB3	18	421	3079	$\simeq$ 18 jours [7]
CELAR6-SUB4	22	477	3230	$\simeq$ 26 jours [7]
CELAR6	100	1222	3389	$\simeq$ 26 jours [7]

Les composantes *CELAR6-SUB0* et *CELAR6-SUB4* n'ont aucune variable en commun et ne partagent aucune contrainte. La somme de leurs optima, $159 + 3230 = 3389$, est un minorant de l'instance 6. Or, le meilleur majorant connu sur ce problème vaut aussi 3389 [Kol96]. Notre décomposition a donc permis de prouver l'optimum de l'instance 6 du CELAR, qui était auparavant un problème ouvert [dGVS97].

Pour accélérer la production de minorants, nous avons appliqué l'algorithme glouton de réduction des domaines (cf. section 4) sur la composante *CELAR6-SUB1*. Il met huit minutes pour retirer vingt valeurs à chaque domaine. Le coût total des relaxations est de 1362339, avec un coût pour regrouper une paire de valeur variant entre 1676 et 11281. La figure 1 montre l'évolution des minorants et des temps pour les obtenir (sur un SUN Sparc 110Mhz) en fonction du nombre de réductions, noté s. Le temps croît de manière exponentielle lorsque s diminue. Les minorants diminuent fortement pour s supérieur à 24 (cette transition brutale est également observée en satisfaction [SM96]). La réduction de moitié des domaines ($s = 22$) génère un problème relaxé dont l'optimum, égal à 2556, obtenu en 20 minutes, est seulement à 4.2% de l'optimum de la composante initiale *CELAR6-SUB1* (obtenu en 15 heures sur la même station).

Pour l'instance 7, nous avons généré une composante contenant les mêmes variables que la composante *CELAR6-SUB4*. L'algorithme de recuit simulé trouve un majorant égal à 242840. Puis, nous avons appliqué une réduction de moitié des domaines ($s = 22$). L'algorithme *Russian Doll Search* termine en 34 minutes sur un SUN UltraSparc10 300Mhz, produisant un minorant égal à 202342.

Pour l'instance 8, nous avons effectué une 20-partition [8], aboutissant à 90 composantes connexes. L'algorithme de recuit simulé trouve un majorant égal à 233 en 62 minutes sur un SUN UltraSparc10 300Mhz. Une réduction de moitié des domaines des quatre (resp. deux) composantes les plus difficiles permet

6. Un nouvel algorithme complet [LMSV98] met seulement 20 minutes sur un PC PentiumII 266Mhz pour résoudre la composante 3.

7. Une implémentation parallèle de l'algorithme *Russian Doll Search* sur un réseau de 40 stations SUN a résolu les composantes *CELAR6-SUB3* et *CELAR6-SUB4* en un week-end chacune [dG98].

8. L'algorithme de recuit simulé ($h = 20, m = 0.9, l = 100000$) trouve une 20-partition de coût d'interconnexion égal à 2341 et de coût de déséquilibre égal à 98 en 252 secondes.

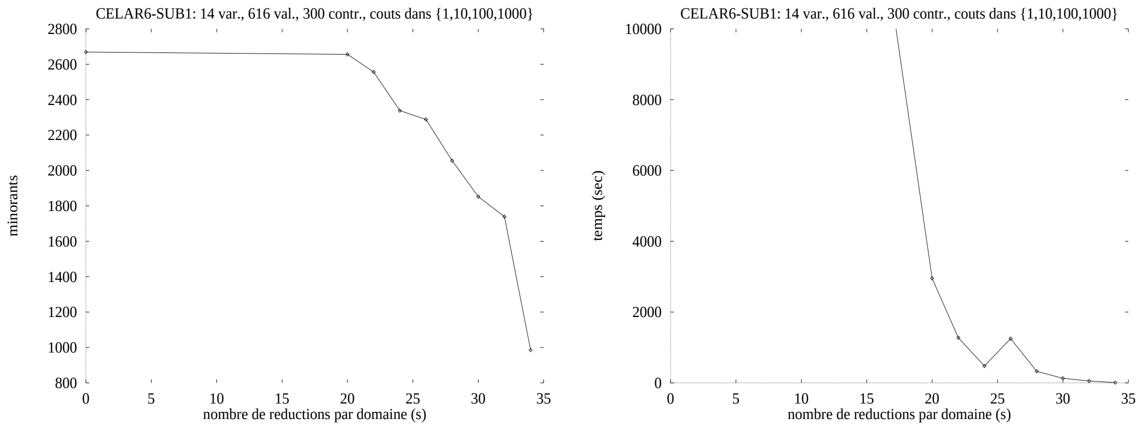

FIG. 1 – *Evolution des minorants et des temps de résolution des relaxations par réduction des domaines de la composante CELAR6-SUB1*

d'obtenir un minorant égal à 195 (resp. 216) en 15 (resp. 158) minutes.

Le tableau suivant résume les meilleurs minorants obtenus par notre approche sur les trois instances :

CELAR	$\|V\|$	*best ub*	*best lb*	*lb/ub*
6	100	3389	3389	100%
7	200	343592	202342	59%
8	458	262	216	82%

6 Conclusion

L'efficacité de notre approche sur les problèmes du CELAR semble due à la structure non uniforme de leurs graphes de contraintes (présence de petites composantes fortement connectées) et à la nature de leurs domaines (grands domaines contenant des valeurs numériques assez voisines). Notre approche reste très empirique. Il serait intéressant de l'intégrer dans une stratégie itérative qui construirait des problèmes de moins en moins relaxés (par des K-partitions avec K décroissant et des réductions de domaines avec un nombre d'étapes décroissant), de plus en plus difficiles à résoudre, fournissant des minorants de plus en plus élevés. La bonne combinaison des deux types de relaxation, séparation des variables et réduction des domaines, est également à étudier.

Je tiens à remercier Bertrand Cabon pour sa participation dans la mise au point des relaxations.

Références

[Ber71] C. Berge. *Principles of Combinatorics.* Academic Press, 1971.

[BMR95] Stefano Bistarelli, Ugo Montanari, and Francesca Rossi. Constraint solving over semirings. In *Proc. of IJCAI-95*, pages 624–630, Montréal, Canada, 1995.

[Cab96] Bertrand Cabon. *Problèmes d'optimisation combinatoire: évaluation de méthodes de la physique statistique.* PhD thesis, ENSAE, Toulouse, France, 1996.

[CdGL+98] B. Cabon, S. de Givry, L. Lobjois, T. Schiex, and J.P. Warners. Benchmark Problems: Radio Link Frequency Assignment. *Constraints*, 3(3):275–285, 1998.

[CdGV98] Bertrand Cabon, Simon de Givry, and Gérard Verfaillie. Anytime Lower Bounds for Constraint Optimization Problems. In *Proc. of CP-98*, pages 117–131, Pisa, Italy, October 26-30 1998.

[CVMB96] Bertran Cabon, Gérard Verfaillie, David Martinez, and Paul Bouret. Using Mean Field Methods for Boosting Backtrack Search in Constraint Satisfaction Problems. In *Proc. of ECAI-96*, pages 165–169, Budapest, Hungary, 1996.

[DFP93] D. Dubois, H. Fargier, and H. Prade. *Fuzzy Sets, Neural Networks and Soft Computing*, chapter Propagation and satisfaction of fuzzy constraints, pages 166–187. Kluwer Acad., 1993.

[dG98] Simon de Givry. *Algorithmes d'optimisation sous contraintes étudiés dans un cadre temps réel.* PhD thesis, École Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, France, 1998.

[dGVS97] Simon de Givry, Gérard Verfaillie, and Thomas Schiex. Bounding the Optimum of Constraint Optimization Problems. In *Proc. of CP-97*, pages 405–419, Schloss Hagenberg, Austria, October 29 - November 1 1997.

[DP87] Rina Dechter and Judea Pearl. The cycle-cuset method for improving search performance in AI. In *Proc. of the 3rd IEEE Conference on AI Applications*, pages 224–230, Orlando, FL, 1987.

[Fre91] Eugene C. Freuder. Eliminating Interchangeable Values In Constraint Satisfaction Problems. In *Proc. of AAAI-91*, pages 227–233, Anaheim, CA, 1991.

[FW92] Eugene C. Freuder and Richard J. Wallace. Partial constraint satisfaction. *Artificial Intelligence*, 58:21–70, 1992.

[HG95] William D. Harvey and Matthew L. Ginsberg. Limited discrepancy search. In *Proc. of IJCAI-95*, pages 607–613, Montréal, Canada, 1995.

[KGV83] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, May 1983.

[Kol96] Antoon Kolen. Results on infeasible instances of the rlfap. Personal communication, 1996.

[Kor85] Richard Korf. Depth-First Iterative Deepening: An Optimal Admissible Tree Search. *Artificial Intelligence*, 27:97–109, 1985.

[Kor93] Richard E. Korf. Linear-space Best-first Search. *Artificial Intelligence*, 62:41–78, 1993.

[Kru56] J.B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. In *Proc. of the American Mathematical Society*, pages 48–50, 1956.

[LMSV98] J. Larrosa, P. Meseguer, T. Schiex, and G. Verfaillie. Reversible DAC and Other Improvements for Solving Max-CSP. In *Proc. of AAAI-98*, pages 347–352, Madison, WI, 1998.

[MJPL92] Steve Minton, Mark D. Johnston, Andrew B. Philips, and Philip Laird. Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence*, 58:161–205, 1992.

[Mon74] U. Montanari. Networks of constraints: Fundamental properties and applications to picture processing. *Information Sciences*, 7:95–132, 1974.

[SFV95] Thomas Schiex, Hélène Fargier, and Gérard Verfaillie. Valued constraint satisfaction problems: Hard and easy problems. In *Proc. of IJCAI-95*, pages 631–637, Montréal, Canada, 1995.

[SFV97] T. Schiex, H. Fargier, and G. Verfaillie. Problèmes de satisfaction de contraintes valués. *Revue d'Intelligence Artificielle*, 11(3):339–373, 1997.

[SM96] R. Shrag and D. Miranker. Abstraction and the CSP Phase Transition Boundary. In *Proc. of the Fourth International Symposium on Artificial Intelligence and Mathematics*, pages 138–141, Fort Lauderdale, FL, USA, 1996.

[THL95a] S. Tiourine, C. Hurkens, and J. Lenstra. Radio Link Frequency Assignment Project Technical Annex T-2.2.1.B: Combinatorial Lower Bounds for the RLFAP. Technical report, Eindhoven University of Technology, 1995.

[THL95b] S. Tiourine, C. Hurkens, and J. Lenstra. A review of algorithmic approaches to frequency assignment problems. Technical Report 3507.00/W.E 2.5.4, Eindhoven University of Technology, 1995.

[VdG97] Gérard Verfaillie and Simon de Givry. Algorithmic problems and solutions in the valued constraint satisfaction problem framework. In *Proc. of the Fifth European Congress on Fuzzy and Intelligent Technologies (EUFIT-97)*, Aachen, Germany, 1997.

[VLS96] Gérard Verfaillie, Michel Lemaître, and Thomas Schiex. Russian Doll Search for Solving Constraint Optimization Problems. In *Proc. of AAAI-96*, pages 181–187, Portland, OR, 1996.

[VM89] D.E. Van den Bout and T.K. Miller. Graph Partitioning using Annealed Neural Networks. In *Proc. of IJCNN-89*, pages 521–528, Washington, 1989.

[VT96] C. Voudouris and E.P.K. Tsang. Partial constraint satisfaction problems and guided local search. In *Practical Application of Constraint Technology (PACT'96)*, pages 337–356, London, UK, April 1996.

[VT98] Christos Voudouris and E.P.K. Tsang. Solving the Radio Link Assignment Problem using Guided Local Search. In *NATO Symposium on Radio Length Frequency Assignment, Sharing and Conservation Systems (Aerospace)*, Aalborg, Denmark, 1998. http://cswww.essex.ac.uk/CSP/gls-papers.html.

[Wal94] R. Wallace. Directed Arc Consistency Preprocessing. In *Proc. of the ECAI-94 Workshop on Constraint Processing (LNCS 923)*, pages 121–137. Springer, 1994.

[Wal96] R. Wallace. Analysis of Heuristic Methods for Partial Constraint Satisfaction Problems. In *Proc. of CP-96*, pages 482–496, Cambridge, MA, USA, 1996.