

Résolution itérative du MAX-CSP

Jun HU¹, Alexandre CAMINADA¹

Université de Technologie de Belfort-Montbéliard
{jun.hu,alexandre.caminada}@utbm.fr

Mots-Clés : *Recherche tabou, Max-CSP, sous graphe critique*

1 Introduction

Nous proposons dans cet article une nouvelle procédure pour la résolution du Max-CSP. Les expérimentations préliminaires de notre technique ont donné de bons résultats sur les instances CELAR [1]. Différentes méthodes approchées ont été proposées pour résoudre de problème satisfaction de contraintes. Ici nous proposons une méthode de résolution itérative qui utilise une procédure de recherche tabou. Nous illustrons notre approche en utilisant les instances CELAR.

2 Notre approche

La nouvelle procédure est basée sur une idée simple : réaliser de manière itérative, l'identification et l'optimisation des parties difficiles du problème puis étendre à l'ensemble de l'instance. Cette procédure est basée sur la recherche des sous graphes critiques proposée par Desrosiers et al[2].

L'algorithme est détaillé en Algorithme 1. Une recherche tabou est utilisée comme méthode de filtrage, qui sélectionne le sous ensemble de contraintes dures à satisfaire. Ensuite, la méthode de recherche tabou s'effectue uniquement sur l'ensemble des contraintes dures, et minimise le nombre de contraintes violées. Finalement, la procédure étend sa résolution à l'ensemble du problème à partir des éléments de la solution partielle. La procédure est itérée jusqu'à ce que la solution partielle soit plus coûteuse que la meilleure solution globale trouvée.

3 Expérimentations préliminaires

Ici, nous présentons les résultats préliminaires de notre approche. Les instances de problèmes traitées ont été obtenues à partir des instances CELAR auxquelles certaines fréquences ont été supprimées. Le tableau 1 montre les résultats obtenus, où la colonne *tabou* indique le nombre de contraintes violées trouvée par la recherche tabou, *optimiseur* indique le nombre de contraintes violées trouvée par notre algorithme et *Itération* est le nombre totale d'itérations réalisé par la procédure.

Références

- [1] CELAR, <http://www.inra.fr/mia/T/schiex/Doc/CELAR.shtml>.
- [2] C. Desrosiers and P. Galinier and A. Hertz. Efficient algorithms for finding critical subgraphs. *Discrete Appl. Math.*, 156:244–266,2008

Algorithm 1: *Optimiseur*

entrées: Ensemble de contraintes C
sorties : Solution S

```

1  $S \leftarrow \text{MinConflict}(f, C, \emptyset), H \leftarrow \emptyset;$ 
2  $\text{Cout}^* \leftarrow f(S), \text{Cout} \leftarrow \text{Cout}^* + \epsilon;$ 
3 pour  $\text{Cout} > \text{Cout}^*$  faire
4    $H \leftarrow \text{Violée}(f, S, C) \cup H;$ 
5    $H \leftarrow \text{Saturer}(C, H);$ 
6    $S_H \leftarrow \text{MinConflict}(f, H, \emptyset);$ 
7   si  $f(S_H) < \text{Cout}^*$  alors
8      $S \leftarrow \text{MinConflict}(f, C, S_H);$ 
9      $\text{Cout} \leftarrow f(S);$ 
10    si  $\text{Cout} < \text{Cout}^*$  alors
11       $\text{Cout}^* \leftarrow \text{Cout};$ 
12       $\text{Cout} \leftarrow \text{Cout} + \epsilon;$ 
13    fin
14  sinon
15    retourner  $S;$ 
16  fin
17 fin
18 retourner  $S;$ 

```

CELAR	Nb Vars	Nb Ctrs	Max Freq	tabou	Optimiseur	Itération
scen01	916	5548	666	32	15	7
scen02	200	1235	380	11	4	5
scen03	400	2760	380	67	66	4
scen06	200	1322	792	136	120	5
scen07	400	2865	792	333	315	4
scen08	916	5744	792	730	701	5
scen11	680	4103	666	29	20	10

TAB. 1 – Résultats