



**HAL**  
open science

## Apport des décompositions arborescentes dans les méthodes de recherche de type VNS

Mathieu Fontaine, Samir Loudni, Patrice Boizumault

► **To cite this version:**

Mathieu Fontaine, Samir Loudni, Patrice Boizumault. Apport des décompositions arborescentes dans les méthodes de recherche de type VNS. 12ème congrès de la société française de recherche opérationnelle et d'aide à la décision (ROADEF'11), 2011, Saint Etienne, France. pp.I: 89-90. hal-01017311

**HAL Id: hal-01017311**

**<https://hal.science/hal-01017311v1>**

Submitted on 2 Jul 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Apport des décompositions arborescentes dans les méthodes de recherche de type VNS

Mathieu Fontaine, Samir Loudni et Patrice Boizumault

GREYC (CNRS - UMR 6072); Université de Caen, France

Prénom.Nom@unicaen.fr

**Mots-clés** : *décomposition arborescente, recherche locale, VNS.*

Actuellement, la structure d'un problème est très peu prise en compte dans la résolution des problèmes de satisfaction et d'optimisation sous contraintes. Or, il existe de nombreux problèmes réels fortement structurés dont la décomposition arborescente peut s'avérer très profitable. Les travaux menés jusqu'à présent exploitent les décompositions arborescentes uniquement dans le cadre des méthodes de recherche complètes [1]. Dans ce papier, nous proposons un premier schéma de coopération permettant d'exploiter le graphe de clusters, résultant de la décomposition par triangulation du réseau de contraintes, afin de guider efficacement l'exploration des voisinages dans une recherche de type VNS (*Variable Neighborhood Search* [2]).

## 1 Définitions

**Définition 1 (Décomposition arborescente d'un graphe)** Soit  $G = (V, E)$  un graphe (connexe) de contraintes. Une décomposition arborescente de  $G$  est définie par un arbre  $\langle C, T \rangle$ . Chaque cluster  $C_i \in C$  est formé d'un sous-ensemble de sommets de  $V$ .  $T$  est un ensemble d'arêtes définissant un graphe acyclique connexe sur  $C$ . L'ensemble des clusters  $C$  doit couvrir tous les sommets de  $G$ .

**Définition 2 (Séparateur, degré et taux d'absorption d'un cluster)** Deux clusters  $C_i$  et  $C_j$  sont dits adjacents ssi ils partagent au moins une variable. Dans ce cas,  $S_{ij} = C_i \cap C_j$  est appelé ensemble séparateur. Le degré d'un cluster  $C_i$  est le nombre de clusters adjacents. Le taux d'absorption d'un cluster  $C_i$  par rapport à un cluster  $C_j$  est défini par le rapport  $|S_{ij}| / |C_i|$ . Enfin, le taux d'absorption d'un cluster est la moyenne des taux d'absorption avec ses voisins.

## 2 DECOMP/VNS : un premier schéma de coopération

Nous avons proposé et mis en œuvre un **premier schéma de coopération** permettant d'exploiter, dans une recherche de type VNS, le graphe de clusters résultant de la décomposition par triangulation<sup>1</sup> du graphe de contraintes. Ce schéma comporte deux étapes principales :

1. Effectuer une *décomposition sémantique* du graphe de contraintes ;
2. Associer à chaque cluster  $C_i$ , résultant de la décomposition, une *structure de voisinage*  $N_i$ . Puis, appliquer successivement une recherche de type VNDS [3] sur chacun des  $N_i$ .

L'idée sous-jacente est de favoriser les *petits mouvements* dans des voisinages *fortement liés*. La notion de cluster correspond à ce critère, de part sa taille (réduite par rapport à la taille du problème initial), et de part la forte corrélation des variables qu'il contient.

L'algorithme 1 part d'une solution initiale. A chaque étape,  $k$  variables sont sélectionnées dans la structure de voisinage  $N_i$ . Une affectation partielle  $\mathcal{A}$  est obtenue, à partir de la solution courante  $S$ , en désaffectant les  $k$  variables sélectionnées. Une solution voisine  $S'$  est alors reconstruite en utilisant

1. Pour la triangulation, nous avons utilisé l'heuristique MCS (Max Cardinality Search).

---

**Algorithm 1:** Principales étapes de DECOMP/VNS

---

```
begin
  Soit  $\langle C, T \rangle$  une décomposition arborescente du problème à traiter.
  Soient  $N_i, i = 1, \dots, i_{max} = |C|$  les structures de voisinage associées.
   $S \leftarrow \text{genInitSol}(), i \leftarrow 1$ 
  while  $(i \leq i_{max}) \wedge (\text{not TimeOut})$  do
     $k_{max} \leftarrow |N_i|, k \leftarrow k_{init}$ 
    while  $(k \leq k_{max}) \wedge (\text{not TimeOut})$  do
       $\mathcal{X}_{unaffected} \leftarrow \text{Hneighborhood}(N_i, S, k)$ 
       $\mathcal{A} \leftarrow S \setminus \{(x_i = a) \mid x_i \in \mathcal{X}_{unaffected}\}$ 
       $S' \leftarrow \text{Rebuild-LDS}(\mathcal{A}, \mathcal{X}_{unaffected}, S)$ 
      if  $f(S') < f(S)$  then
         $S \leftarrow S', k \leftarrow k_{init}$ 
      else
         $k \leftarrow k + 1$ 
    end while
     $i \leftarrow i + 1,$ 
end
```

---

une recherche arborescente partielle (LDS [4, 5]). Si  $S'$  est de meilleure qualité que  $S$ , alors  $S'$  devient la solution courante et  $k$  reprend sa valeur de départ. Sinon,  $k$  est incrémenté de 1 afin de s'échapper de ce minimum local. L'exploration du voisinage  $N_i$  prend fin lorsque le nombre maximal de variables à désinstancier est atteint. L'algorithme passe alors au voisinage suivant  $N_{i+1}$ . Il s'arrête dès que toutes les structures de voisinages ont été examinées ou que le *timeout* est atteint.

**Heuristiques expérimentées pour ordonner les structures de voisinages :**

- Taille des clusters : (i) *le plus grand* : on privilégie au début les plus grands clusters afin de réparer le maximum de conflits ; (ii) *le plus petit* : on commence par des mouvements très rapides mais avec une faible marge d'amélioration, pour finir sur des mouvements plus longs.
- Degré des clusters : (i) *le plus grand* : on commence par les clusters centraux ; (ii) *le plus petit* : on commence par les extrémités du graphe.
- Taux d'absorption des clusters : (i) *le moins absorbé*, (ii) *le plus absorbant*.

L'heuristique qui ordonne les clusters par taille décroissante fournit les meilleurs résultats. Mais, comme la décomposition arborescente n'est basée que sur des critères topologiques, elle ne permet pas d'identifier certaines sous-structures *intéressantes* du réseau de contraintes.

**Décomposition sémantique.** Nous avons choisi la *dureté des contraintes*<sup>2</sup> comme critère sémantique. La décomposition s'effectue alors en 2 temps : tout d'abord, extraire le sous-graphe  $G'$  constitué uniquement des contraintes ayant une dureté supérieure à un seuil fixé ; puis, effectuer la décomposition arborescente de  $G'$ . Les premières expérimentations menées sur des instances réelles du problème d'allocation de fréquences à des liens radio (RLFAP) [6] nous ont permis de valider la pertinence de notre approche et les résultats obtenus montrent l'importance d'allier prise en compte de la sémantique et méthodes de décomposition.

**Références**

- [1] S. de Givry, T. Schiex, and G. Verfaillie. Exploiting tree decomposition and soft local consistency in Weighted CSP. *AAAI'06*, pages 1-6, 2006.
- [2] N. Mladenovic and P. Hansen. Variable Neighborhood Search. *Computers & OR*, 24 :1097-1100, 1997.
- [3] P. Hansen, N. Mladenovic, and D. Perez-Britos. Variable Neighborhood Decomposition Search. *Journal of Heuristics*, 7(4) :335-350, 2001.
- [4] W. Harvey and M. Ginsberg. Limited Discrepancy Search. *IJCAI'95*, pages 607-614, 1995.
- [5] S. Loudni and P. Boizumault. Combining VNS with Constraint Programming for solving anytime optimization problems. *EJOR*, 191(3) :705-735, 2008.
- [6] B. Cabon, S. de Givry, L. Lobjois, T. Schiex, and J. Warners. Radiolink frequency assignment. *Constraints*, 4(1) :79-89, 1999.

---

2. Pourcentage de t-uples ne vérifiant pas la contrainte vs la taille du produit cartésien des domaines des variables.