



Recherche complète à voisinages variables guidée par la décomposition arborescente pour la minimisation d'énergie dans les modèles graphiques

Abdelkader Ouali, David Allouche, Simon de Givry, Samir Loudni, Yahia Lebbah, Francisco Ribeiro Eckhardt, Lakhdar Loukil

► To cite this version:

Abdelkader Ouali, David Allouche, Simon de Givry, Samir Loudni, Yahia Lebbah, et al.. Recherche complète à voisinages variables guidée par la décomposition arborescente pour la minimisation d'énergie dans les modèles graphiques. 9e Journées Francophones sur les Réseaux Bayésiens et les Modèles Graphiques Probabilistes 2018 (JFRB 2018), May 2018, Toulouse, France. 114 p., 10.3166/RIA.-1-7 . hal-02733676

HAL Id: hal-02733676

<https://hal.inrae.fr/hal-02733676>

Submitted on 2 Jun 2020

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.

Recherche complète à voisinages variables guidée par la décomposition arborescente pour la minimisation d'énergie dans les modèles graphiques

Résumé de l'article paru à la conférence Uncertainty in Artificial Intelligence (UAI-17), pages 550-559, Sydney, Australia, 2017

A. Ouali¹, D. Allouche², S. de Givry², S. Loudni¹, Y. Lebbah³,
F. Eckhardt², L. Loukil³

1. University of Caen Normandy,
CNRS, UMR 6072 GREYC, 14032 Caen,
France
samir.loudni@unicaen.fr, oualiaek@gmail.com

2. INRA
MIA Toulouse, UR-875, 31320 Castanet-Tolosan
France
david.allouche@inra.fr, simon.de-givry@inra.fr

3. University of Oran 1
Ahmed Ben Bella, Lab. LITIO, 31000 Oran,
Algeria
ylebbah@gmail.com, loukil_lakhdar@yahoo.fr

RÉSUMÉ. Les modèles graphiques probabilistes unifient la théorie des probabilités et les modèles graphiques via des variables aléatoires reliées entre elles par une distribution de loi jointe décomposable. L'objectif étudié ici est connu sous le nom de Maximum A Posteriori dans les champs aléatoires de Markov et de Most Probable Explanation dans les réseaux bayésiens, consiste à trouver une affectation globale de toutes les variables ayant une probabilité maximale ou de manière équivalente une énergie minimale. Les méthodes de résolution MAP/MPE sont qualifiées habituellement de complète ou incomplète, selon leur capacité à prouver l'optimalité

ou non. Alors que la plupart des méthodes complètes reposent sur la recherche arborescente, les méthodes incomplètes reposent, quant à elles, sur la recherche locale. Dans cet article, nous proposons une approche itérative potentiellement complète au-dessus d'une recherche à voisinages variables qui utilise la recherche arborescente partielle dans son exploration locale des voisinages. La méthode hybride qui en résulte a été évaluée sur un large éventail de benchmarks issus de l'analyse d'images, la reconnaissance de formes, l'allocation de ressources, la bio-informatique, la bio-physique... Les résultats montrent que comparée aux méthodes de recherche arborescente existantes, notre méthode offre un meilleur compromis entre le comportement anytime et la preuve d'optimalité. Enfin, l'expérimentation de notre version parallèle sur des instances difficiles issues de la conception de protéines a mis en évidence l'efficacité de la version parallèle pour trouver des bonnes solutions. Le solveur est librement téléchargeable sur <https://github.com/toulbar2/toulbar2>.

ABSTRACT. *Graphical models factorize a global probability distribution/energy function as the product/sum of local functions. A major inference task, known as MAP in Markov Random Fields and MPE in Bayesian Networks, is to find a global assignment of all the variables with maximum probability/minimum energy. A usual distinction on MAP/MPE solving methods is complete/incomplete, i.e. the ability to prove optimality or not. Most complete methods rely on tree search, while incomplete methods rely on local search. Among them, we study Variable Neighborhood Search (VNS) for graphical models. In this paper, we propose an iterative approach above VNS which uses (partial) tree search inside its local neighborhood exploration. The resulting hybrid method offers a better compromise between completeness and anytime behavior than existing tree search methods while still being competitive for proving optimality. We further propose a parallel version of our method improving its anytime behavior on difficult instances coming from a large graphical model benchmark. Last we experiment on the challenging minimum energy problem found in Computational Protein Design, showing the practical benefit of our parallel version. Solver is available at <https://github.com/toulbar2/toulbar2>.*

MOTS-CLÉS : *optimisation combinatoire, modèle graphique, recherche de l'explication la plus probable, recherche à voisinages variables, recherche arborescente partielle, réseau de fonctions de coûts.*

KEYWORDS: *combinatorial optimization, graphical model, most probable explanation, variable neighborhood search, limited discrepancy search, cost function network.*

1. Introduction

Les Modèles Graphiques Probabilistes (MGPs) (Koller, Friedman, 2009) unifient la théorie des probabilités et les modèles graphiques via des variables aléatoires reliées entre elles par une distribution de loi jointe décomposable. Formellement, un MGP est un triplet $(\mathcal{X}, \mathcal{D}, \mathcal{F})$ avec $\mathcal{X} = \{X_1, \dots, X_n\}$ un ensemble de n variables aléatoires discrètes, $\mathcal{D} = \{D_1, \dots, D_n\}$ un ensemble de domaines finis de valeurs, et $\mathcal{F}$, un ensemble de fonctions sur les variables à valeurs réelles positives (probabilités conditionnelles dans les réseaux bayésiens ou *potentiels* dans les champs de Markov). Une affectation de l'ensemble $\mathcal{X}$ est le tuple $x = (x_1, \dots, x_n)$, avec $x_i \in D_i$. L'ensemble de toutes les affectations possibles de $\mathcal{X}$ est noté $\Delta = \prod_{i=1}^n D_i$. On note $A = \{D'_1, \dots, D'_n\}$, avec $D'_i \subseteq D_i$, une affectation partielle représentant un sous-ensemble de Δ . Soit S un sous-ensemble de $V = \{1, \dots, n\}$, X_S , x_S et Δ_S dénotent respectivement un sous-ensemble de variables aléatoires $\{X_i, i \in S\}$, l'affectation $(x_i, i \in S)$ obtenue à partir de x , et l'ensemble des affectations possibles des variables de X_S . Étant donné S un ensemble de parties de V , l'ensemble $\mathcal{F} = \{f_S\}_{S \in \mathcal{S}}$ définit une factorisation de la distribution de probabilité jointe de $\mathbb{P}$ ssi:

$$\mathbb{P}(x) = \frac{1}{Z} \prod_{f_S \in \mathcal{F}} f_S(x_S) \quad (1)$$

où $Z = \sum_{x \in \Delta} \prod_{f_S \in \mathcal{F}} f_S(x_S)$ est la constante de normalisation. Le problème *Most Probable Explanation* (MPE) consiste à trouver une affectation $x \in \Delta$ de toutes les variables de $\mathcal{X}$ de telle sorte que la probabilité jointe $\mathbb{P}(x)$ soit maximale. Il s'agit d'un problème NP-difficile (Shimony, 1994). Le problème MPE se traduit directement en un *réseau de fonctions de coût* (Mesequer *et al.*, 2006) où l'on cherche à minimiser la somme de fonctions de coûts, appelées aussi *fonctions d'énergie* (Hurley *et al.*, 2016).

Les méthodes de résolution des MGPs sont qualifiées respectivement de méthodes complètes ou incomplètes selon leur capacité à prouver ou non l'optimalité de la solution trouvée. Elles font généralement appel soit à la recherche arborescente soit à la recherche locale. La combinaison de ces deux aspects a été étudiée dans peu de travaux. Parmi eux, les approches qui consistent à explorer les voisinages de la recherche locale par une recherche arborescente de manière systématique ou non systématique.

VNS/LDS+CP (Loudni, Boizumault, 2003) combine une métaheuristique Variable Neighbourhood Search (VNS) (Mladenović, Hansen, 1997) avec une recherche arborescente partielle de type LDS (Harvey, Ginsberg, 1995). Récemment, Fontaine *et al* (Fontaine *et al.*, 2013) ont proposé le premier cadre générique, appelé Decomposition Guided VNS (DGVNS), qui exploite une décomposition arborescente (Robertson, Seymour, 1986) au sein de VNS. Dans ce papier, nous proposons une variante itérative de DGVNS appelée Unified Decomposition Guided VNS (noté UDGVNS) capable de prouver l'optimalité des solutions trouvées si le temps de résolution n'est pas borné.

Algorithme 1 : Unified DGVNS algorithm.

```

Function UDGVNS ( $\ell_{min}, \ell_{max}, +\ell, k_{min}, k_{max}, +k, ub : In/Out, x : In/Out$ ) : boolean
  let  $(C_T, T)$  be a tree decomposition of  $(\mathcal{X}, \mathcal{D}, \mathcal{F})$ ;
   $opt \leftarrow \mathbf{true}$ ;
  1  LDSr ( $\infty, \mathcal{D}, ub, x, opt$ ) ; // initial solution
  2  if ( $ub = lb(\mathcal{D})$ ) then  $opt \leftarrow \mathbf{true}$ ;
      $c \leftarrow 1$ ; // current cluster index
      $r \leftarrow 0$ ; // number of iterations
      $\ell \leftarrow \ell_{min}$ ; // initial discrepancy limit
     while ( $\neg opt \wedge \ell \leq \ell_{max}$ ) do
        $i \leftarrow 0$ ; // number of failed neighborhoods
       3   $k \leftarrow k_{min}$ ; // initial neighborhood size
          while ( $\neg opt \wedge k \leq k_{max}$ ) do
             $A \leftarrow \text{getNeighborhood}(x, C_c, k)$ ;
             $ub' \leftarrow ub, opt \leftarrow \mathbf{true}$ ;
            4  LDSr ( $\ell, A, ub', x', opt$ ) ; // neighborhood search
            5  if ( $ub' = lb(\mathcal{D})$ ) then  $opt \leftarrow \mathbf{true}$ ;
            6  else if ( $A \neq \mathcal{D}$ ) then  $opt \leftarrow \mathbf{false}$ ;
            7  if ( $ub' < ub$ ) then
                $x \leftarrow x', ub \leftarrow ub'$ ; // new best solution
            8   $i \leftarrow 0, k \leftarrow k_{min}$ ;
            9   $r \leftarrow 0, \ell \leftarrow \ell_{min}$ ;
               else
                  $i \leftarrow i + 1$ ;
                 if ( $k < k_{max}$ ) then
                    $k \leftarrow \min(k_{max}, k_{min} + k \cdot i)$ ;
                 else  $k \leftarrow \infty$ ;
            10  $c \leftarrow 1 + c \bmod |C_T|$ ; // get next cluster
                $r \leftarrow r + 1$ ;
               if ( $\ell < \ell_{max}$ ) then
                  $\ell \leftarrow \min(\ell_{max}, \ell_{min} + \ell \cdot r)$ ;
               else  $\ell \leftarrow \infty$ ;
  11 return  $opt$ ;

```

2. DGVNS itératif

L'algorithme 1 décrit le pseudo-code d'UDGVNS. Il restaure l'exhaustivité de DGVNS en appliquant des appels successifs avec des valeurs croissantes d'écart à l'heuristique de choix de valeurs (*discrepancy*, noté ℓ) pour LDS¹, en contrôlant si la recherche arborescente est partielle grâce au drapeau opt et le fait que le voisinage actuel garde certaines variables assignées dans l'affectation partielle A (test à la ligne 6). Dans UDGVNS, l'optimalité peut être prouvée dans deux cas: (i) lorsque le voisinage actuel correspond à l'ensemble des variables du problème et que la valeur ℓ est supérieure ou égale au nombre maximal de branches droites ou (ii) en examinant les bornes au

1. Nous supposons un arbre de recherche binaire où à chaque nœud soit la variable sélectionnée est affectée à sa valeur préférée (branche de gauche) soit la valeur est supprimée du domaine (branche de droite). Chaque suppression correspond à une mauvaise décision prise par la recherche.

nœud racine ($ub = lb(\mathcal{D})$), cf. lignes 2 et 5). Dans ce cas, l’optimalité est prouvée implicitement, l’espace de recherche n’est pas exploré.

La solution initiale est obtenue à la ligne 1 par une version modifiée de LDS, notée LDS^r , qui stoppe après l’obtention de la première solution.

UDGVNS ajuste le compromis entre la preuve d’optimalité et le comportement *any-time* (i.e., l’évolution de la qualité des solutions trouvées au cours du temps) via deux paramètres: la limite de l’écart à l’heuristique (ℓ) et la taille du voisinage (k). Dès qu’une meilleure solution est trouvée par LDS^r dans le voisinage courant (ligne 4), nous arrêtons la recherche afin de réinitialiser les deux paramètres à leur valeur minimale, car il est plus rapide d’explorer des petits voisinages (lignes 8-9). Nous avons évalué trois stratégies de mise à jour de ℓ et k pour l’opérateur $+_{\ell/k}$: augmente d’une unité (`add1`), multiplie par deux (`mult2`), enfin selon une suite Luby (Luby *et al.*, 1993)² qui renvoie la valeur $a +_{\ell/k} b = \text{Luby}(a, b) = a \times \text{luby}(1+b)$ pour $\forall a, b \in N^*$.

L’opérateur $+_k$ contrôle le compromis entre intensification et diversification. Le but de la stratégie de Luby est d’intensifier l’effort de recherche sur les petits voisinages en augmentant exponentiellement le nombre de voisinages de petite taille versus ceux de grande taille, permettant ainsi de passer plus de temps sur les petits voisinages pour améliorer localement la solution actuelle, favorisant ainsi l’intensification. La stratégie `mult2` réduit le nombre d’explorations de voisinage pour une valeur ℓ donnée, afin d’essayer plus rapidement des valeurs de ℓ plus grandes. Si le problème peut être résolu par une recherche complète dans les délais impartis, cela permet d’avoir une preuve d’optimalité plus tôt.

L’opérateur $+_\ell$ contrôle l’équilibre entre recherche incomplète et complète. L’utilisation d’une stratégie de croissance rapide accentue l’exhaustivité alors qu’une croissance lente devrait favoriser les comportements anytime. Nous avons noté qu’il est plus efficace de couvrir toutes les variables par l’union des voisinages explorés afin de ne pas manquer certaines variables importantes.

Pour exploiter cette idée, nous avons testé une quatrième stratégie, qui exploite la décomposition arborescente du problème, notée (C_T, T) avec $T = (I, A)$ un arbre de sommets I et d’arêtes A , et $C_T = \{C_i \mid i \in I\}$ un ensemble de parties de $\mathcal{X}$ appelées clusters. Elle consiste à modifier k via un incrément lent (de +1) au début jusqu’à $k = \max_{i \in I}(|C_i|) + |C_T| - 1$ puis *saute* directement à $k = k_{max}$. Afin de garantir que k augmente lentement jusqu’à ce que le plus grand groupe de variables ait été totalement exploré au moins une fois lors d’une recherche dans un voisinage. Ensuite, quand $k = k_{max} = |\mathcal{X}|$, UDGVS effectue une nouvelle recherche de solution via LDS^r sur l’ensemble du problème. S’il ne parvient pas à trouver une meilleure solution, un écart ℓ plus grand est pris et UDGVS poursuit son processus d’intensification en repartant d’un voisinage de petite taille avec un nombre de variables égale à k_{min} (ligne 3).

2. Rappelons que $\text{luby}(i) = \{1, 1, 2, 1, 1, 2, 4, 1, 1, 2, 1, 1, 2, 4, 8, \dots\}$.

3. Expérimentations

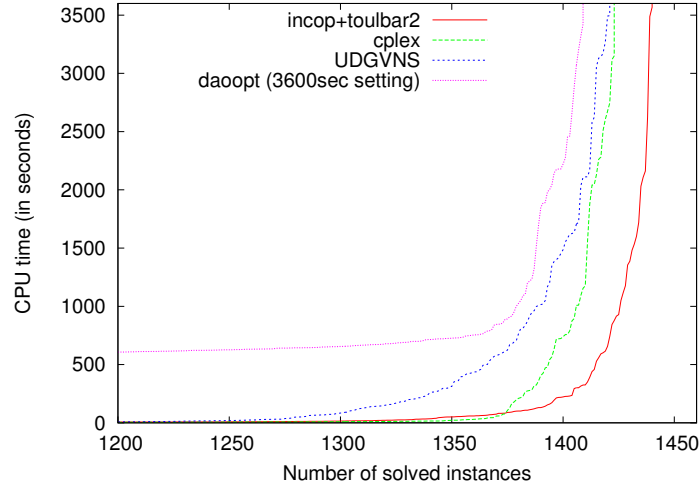


Figure 1. Nombre d'instances résolues à l'optimum en fonction du temps de calcul

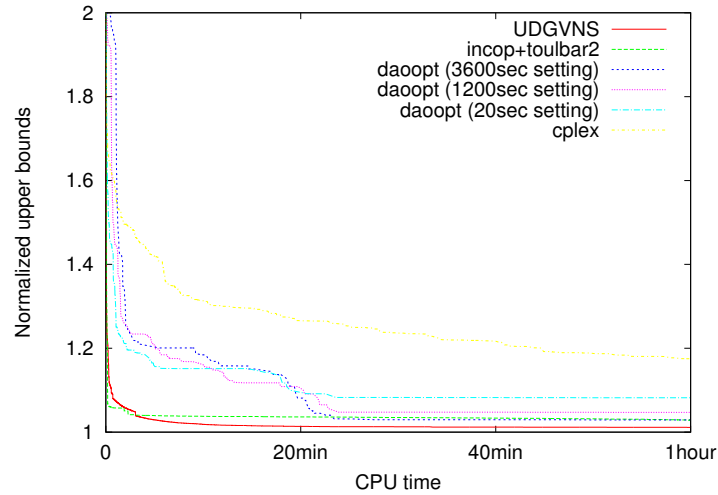


Figure 2. Qualité des solutions au cours du temps pour 114 instances difficiles

UDGVNS a été évalué sur 1669 instances non-triviales issues de compétitions sur les modèles graphiques (Hurley *et al.*, 2016). Elles recouvrent divers domaines d'applications (Probabilistic Inference Challenge, Computer Vision and Pattern Recognition, OpenGM2 benchmark, Cost Function Library). À notre connaissance, ce travail

est la première tentative de restauration de la complétude sur une méthode de recherche locale de type VNS. En pratique, nous avons fixé $k_{min} = 4$, $k_{max} = |\mathcal{X}|$, $+_k = \text{add1} / \text{jump}$, $\ell_{min} = 1$, $\ell_{max} = |\mathcal{X}| \times \max_{i=1}^n (|D_i| - 1)$ et $+_\ell = \text{mult} 2$. Les résultats, présentés en Figures 1 et 2, montrent que l'approche offre un bon compromis entre comportement anytime et preuve d'optimalité, comparé à des méthodes de l'état de l'art dont IBM ILOG CPLEX 12.7.0.0 (paramètres EPAGAP, EPGAP, EPINT mis à zéro pour éviter un arrêt prématuré), DAOOPT³ (incluant une recherche locale GLS⁺ suivie d'une recherche arborescente exploitant la décomposition arborescente) et INCOP+TOULBAR2⁴ (ancienne version 0.9.8 avec les paramètres *-i -dee -hbfs* effectuant une recherche locale *IDWalk* suivie par une recherche arborescente hybride meilleure-en-premier *hbfs* (Hurley *et al.*, 2016)). L'article original à UAI-17 présente également une version parallèle d'UDGVNS.

Bibliographie

- Fontaine M., Loudni S., Boizumault P. (2013). Exploiting tree decomposition for guiding neighborhoods exploration for VNS. *RAIRO OR*, vol. 47, n° 2, p. 91-123.
- Harvey W., Ginsberg M. (1995). Limited discrepancy search. In *Proc. of ijcai*, p. 607-615.
- Hurley B., O'Sullivan B., Allouche D., Katsirelos G., Schiex T., Zyticki M. *et al.* (2016). Multi-Language Evaluation of Exact Solvers in Graphical Model Discrete Optimization. *Constraints*, vol. 21, n° 3, p. 413-434.
- Koller D., Friedman N. (2009). *Probabilistic graphical models: principles and techniques*. The MIT Press.
- Loudni S., Boizumault P. (2003). Solving constraint optimization problems in anytime contexts. In *Proc. of ijcai*, p. 251-256.
- Luby M., Sinclair A., Zuckerman D. (1993). Optimal speedup of Las Vegas algorithms. In *Proc. of tcs*, p. 128-133.
- Meseguer P., Rossi F., Schiex T. (2006). Soft constraints processing. In *Handbook of constraint programming*, chap. 9. Elsevier.
- Mladenović N., Hansen P. (1997, novembre). Variable Neighborhood Search. *Comput. Oper. Res.*, vol. 24, n° 11, p. 1097-1100.
- Robertson N., Seymour P. D. (1986). Graph minors. ii. algorithmic aspects of tree-width. *Journal of Algorithms*, vol. 7, n° 3, p. 309-322.
- Shimony S. (1994). Finding MAPs for belief networks is NP-hard. *Artificial Intelligence*, vol. 68, p. 399-410.

3. graphmod.ics.uci.edu/group/DAOOPT-UA112

4. www.inra.fr/mia/T/toulbar2