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

D. Allouche

A. Ouali, D. Allouche, S. de Givry, S. Loudni, F. Eckhardt, Y. Lebbah, L. loukil
UAI 2017 conf paper (sydney)
Outline

- Graphical model :

- Unified Decomposition Guided Variable Neighborhood Search  
  - (UDGVNS : complete method)

- EXPERIMENTAL RESULTS

- CONCLUSION
Graphical Model

- Influence Diagram
- Markov Random Field
- Bayesian Network

Shapiro, Haralick, IEEE PAMI 81

\[ X = \{ X_1, \ldots, X_n \} \]
\[ X_i \in D_i, |D_i| \leq d \]
\[ F = \{ f_1, \ldots, f_e \} \]

Energy:
- Costs \( \in \{0, \ldots, +\infty\} \)
- finite or infinite integer

Minimize \( \sum_{F} f_i(X) \)

NP-hard
Markov Random Field

\[ P(x) = \frac{1}{Z} \prod_{f_S \in \mathcal{F}} f_S(x_S) \]

\[ Z = \sum_{x \in \Delta} \prod_{f_S \in \mathcal{F}} \]

Optimization \( \rightarrow \) Markov Random Field \( \rightarrow \) (MAP = Maximum a posteriori)

Cost Function Network equivalent after a \( - \) log transform.

For Bayesian Network \( \rightarrow \) (MPE = Most Probable Explanation)
DFBB (*Depth-First Branch & Bound*)

- **Dynamic variable ordering**
- **Dynamic value ordering**

**Search node** = Assignment + Propagation

- **Lower Bound** = best energy estimation in the current sub-tree
- **Upper Bound** = Energy of best solution known so far

If LB >= Ub

- **Hard Pb**
- **Optimal solution**

Dynamic value ordering
Limited Discrepancy Search (Ginsberg 95)

- Small example with 3 variables and 2 values per domain
Limited Discrepancy Search

- Small example with 3 variables and 2 values per domain
Limited Discrepancy Search (Ginsberg 95)
Limited Discrepancy Search (Ginsberg 95)

$l=3 \Rightarrow$ optimality proof

$$l_{max} = |\mathcal{I}| \cdot (D_{max} - 1) : \text{ in this case } l_{max} = 3 \cdot (2 - 1) = 3$$

Full exploration

in the worst case:

LDS Complexity: $(k.l)^{l+1}$
Variable Neighborhood Search (Hansen 97)

Neighborhood = Combinatorial sub space associated with Selected

LDS SEARCH with given discrepancy
UDGVNS : Exploration of both \( k \) and \( l \) dimensions
Step 0: Initial solution

Greedy assignment
NEW SOLUTION WITH BETTER E → RESTART

Lds

l=0

l=1

l=2

l_{\text{max}}

k_{\text{init}}=4

k=5

k=...

k_{\text{max}}

New E_{\text{best}}

DSF
Completeness restoration

In the worst case \( l \geq \text{max number of right branches} \)
\[
\begin{align*}
\text{IFF } k &= k_{\text{max}} = \text{problem size} \\
\end{align*}
\]

IFF \( \text{ub}=\text{lb}(\text{problem}) \) can be before \( k_{\text{max}} \)

In practice can be before \( l_{\text{max}} \) (due to the pruning in DFBB)
Resolution Strategy:
 conditioned by \((l, k)\) Operator:

- **Linear**: \(k^{++}, l^{++}\)
- **Mult \* 2**: \(l^*2, k^*2\)
- **Luby serie**: \((Luby \ & \ al \ 1993)\)
  - \(S=(1, 2, 1,1,2,4, 1,1,2,1,1,2,4,8, \ldots)\)
- **k JUMP** (Goal: solve completely the biggest cluster)
Cluster visit in a topological order:
k jump heuristic

If $k < W + |C| - 1$ then $k^{++}$
else
$k = |X| = k_{\text{max}}$
EXPERIMENTAL PROTOCOL

1 hour CPU time limit

Benchmark description:

Optimality proof: experiment on 1669 instances

- Probabilistic Inference Challenge (PIC 2011), UAI competitions
- CVPR instances (Computer Vision and Pattern Recognition)
- Cost Function Library

Anytime curves: Evolution of best energy over time

- subset of 114 difficult instances
  (unsolved after 1h search runtime with hbfs and BTD)
Solvers:

IBM ILOG CPLEX release 12.7.0.0 (MIP solver)
- parameter : Default
- precision threshold: EPAGAP, EPGAP, EPINT
  - (set to zero in order to enforce complete search)

DAOOPT: (Dechter & AL)
- GLS+ (Guided local search)
- AND/OR search based tree decomposition

INCOP + TOULBAR2 0.98 (previous release)
- IDWalk (local search in preprocessing)
- Dead End Elimination
- Hybrid Best First Search

Toulbar2 1.0: UDGVNS
- Parameters: \( k_{\text{min}} = 4 \); \( k_{\text{max}} = |X| \);
  \( \text{lds}_{\text{max}} = |X| \times (|D_{\text{max}}| - 1) \);
  \( D_{\text{max}} = \text{Max Domain size} \)
- With best Operators:
  - \( k \rightarrow k++ / \text{jump} \)
  - \( \text{LDS} \rightarrow \text{mult*2} \)

Lib DAI rel 0.3.2 (UAI 2010 settings):
- Message passing algorithm with decimation
Cactus plot

The cactus plot for optimality has been realised on 1669 instances.

UDGVNS with k++ jump and lds mult*2
Anytime behavior

UDGVNS with $k^{++}$ jump & $l \leftarrow \text{mult} \times 2$

Boosts noticeably the performance of UDGVNS
Parallelization

Optimality checking

Cluster 1

Cluster 2

Cluster i

Neighborhood

Neigh(1,k,i)

Neigh(2,k,i)

Tree decomposition & Neighborhoods

Master Process

Sends

Leaf Processes

Worker Process 1

Worker Process 2

One step of intensified shaking

Solution
Anytime Zoom with Parallele release

cluster of 96 Opteron 6174 at 2.2 GHz & 256 GB RAM

- UPDGVNS (30 cores)
- UPDGVNS (10 cores)
- UDGVNS
- incop+toulbar2
- daoopt (1200sec setting)

Normalized upper bounds

Wall-clock real time

INRA
Science & Impact
Conclusion

- **UDGVNS restores completeness**
  (with *various* LDS and neighborhood *evolution strategies*)

- **UDGVNS the best balance between anytime behaviour and optimality proof**
  (empirical results with *k jump and lds = mult*2)

- **Parallel version improves its anytime behavior**
  (better search space covering due to multiple selection of neighborhood with same k value)
**Acknowledgement**

- Ouali$^{2,3}$,
- S. de Givry$^1$, F. Eckhardt$^1$
- S. Loudni$^3$,
- Y. Lebbah, L. Loukil$^2$

1 MIAT UR 875, INRA, F-31320 Castanet Tolosan, France
2 Lab. LITIO University of Oran 31000 Oran, Algeria
3 CNRS, UMR 6072 GREYC, University of Caen Normandy, France

**Computational support:**
- HPC HAYTHAM of university of Oran 1
- Scientific and Technical Information center IBNBADIS
- Genotoul bioinformatic platform (Toulouse / occitanie)
Practical interest of graphical model

Image processing: (example from openGM2)
- Segmentation
- Form recognition

In-Painting

- Chinese Characters recognition
- Color Segmentation

Protein modeling:
- Computational Protein Design
- force fields tuning

\[ \text{hamiltonian} \rightarrow \text{objectif function } \leftrightarrow \text{Energie} \rightarrow \text{Probability} = e^{-\beta \cdot E} \]
merci
• [Larrosa et al., 2005] Existential arc consistency: getting closer to full arc consistency in weighted CSPs. IJCAI
• [Larrosa et al., 2016] Limited discrepancy AND/OR search and its application to optimization tasks in graphical models. IJCAI
• [Harvey and Ginsberg, 1995] W Harvey and M Ginsberg. Limited discrepancy search. IJCAI
• [Ouali et al., 2015] Replicated Parallel Strategies for Decomposition Guided VNS. ENDM.
• [Fontaine et al., 2013] Exploiting tree decomposition for guiding neighborhoods exploration for VNS. RAIRO OR
• [Hurley et al., 2016] MultiLanguage Evaluation of Exact Solvers in Graphical Model Discrete Optimization. Constraints
INS multi stat
Variable Neighborhood Search

- 1997: Variable Neighborhood Search (Mladenov & Hansen 1997) (local search)

- 2003: VNS/LDS + CP (Loudni & Boizunault, 2003) ⇒ (local search based on constraint programming)

- 2013: DGVNS (Fontaine & al 2013 RAIRO OR) ⇒ Decomposable Guided VNS tree decomposition (local search)

**UDGVNS**: Unified Decomposable Guided VNS (complete method)
- Enhancing proof optimality in UDGVNS by exploiting story search in LDS of previous iterations.

- Automatic tuning of the best parameter settings:
  - per instance / family of instances
    - based on dynamic metric during search
    - based on prior knowledge information
    - Tree decomposition impact

- Benchmarking on proteins design problem (and others putatives applications ;)

Perspectives
Tree decomposition
Anytime measure → convergence speed

Normalized energie:

\[ E^*(t) = \frac{E(t_0) - E(t_i)}{E_{\text{best}}} \]

\[ \bar{E}(t) = \frac{1}{N} \sum_{i=1}^{N} E^*(t) \]

\[ E^*(t) = \text{normalized energy} \]
\[ E(t_0) = \text{initial upper bound energy} \]
\[ E_{\text{best}} = \text{Energy of Best known solution} \]

Cactus plot → optimality

\[ \Sigma_0^t \#\text{solved} = f(t) \]

Link with ranked probability skill score (RPSS) ??
Algorithm 1: Unified DGVNS algorithm.

Function UDGVNS (ℓ_{min}, ℓ_{max}, +ℓ, k_{min}, k_{max}, +k, ub : In/Out, x : In/Out) : boolean

let (C_T, T) be a tree decomposition of (X, D, F);

opt ← true;
1 LDS^r (∞, D, ub, x, opt) ; // initial solution
2 if (ub = 1b(D)) then opt ← true;
3 c ← 1 ; // current cluster index
4 r ← 0 ; // number of iterations
5 ℓ ← ℓ_{min} ; // initial discrepancy limit
6 while (¬opt ∧ ℓ ≤ ℓ_{max}) do
7     i ← 0 ; // number of failed neighborhoods
8     k ← k_{min} ; // initial neighborhood size
9     while (¬opt ∧ k ≤ k_{max}) do
10        A ← getNeighborhood(x, C_c, k);
11        ub' ← ub, opt ← true ;
12        LDS^r (ℓ, A, ub', x', opt) ; // neighborhood search
13        if (ub' = 1b(D)) then opt ← true;
14        else if (A ≠ D) then opt ← false;
15        if (ub' < ub) then
16            x ← x', ub ← ub'; // new best solution
17            i ← 0, k ← k_{min} ;
18            r ← 0, ℓ ← ℓ_{min} ;
19        else
20            i ← i + 1 ;
21            if (k < k_{max}) then
22                k ← min(k_{max}, k_{min} + k i);
23            else k ← ∞;
24            c ← 1 + c mod |C_T| ; // get next cluster
25            r ← r + 1 ;
26            if (ℓ < ℓ_{max}) then
27                ℓ ← min(ℓ_{max}, ℓ_{min} + ℓ r);
28            else ℓ ← ∞;
29        end while
30     end while
31 end while
32 return opt ;
Instances contain from 130 up to 282 variables with maximum domain size from 383 to 438, and between 1706 and 6208 cost/energy functions. The tree width ranges from 21 to 68 (i.e. from 0.16 to 0.34 for a normalized tree width). Instances selected on the basis of 3D proteins critters (i.e. normalized Gyration radius)
## CPD results

<table>
<thead>
<tr>
<th>Instance</th>
<th>ncl</th>
<th>(1) Succ.</th>
<th>Time (s)</th>
<th>(2) Succ.</th>
<th>Time (s)</th>
<th>(3) Succ.</th>
<th>Time (s)</th>
<th>ΔE</th>
<th>Time (s)</th>
<th>ΔE</th>
<th>(1/2)</th>
<th>(1/3)</th>
<th>(2/3)</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>5db1</td>
<td>87</td>
<td>10/10</td>
<td>2761±67</td>
<td></td>
<td>10/10</td>
<td>963±53</td>
<td>10/10</td>
<td>0.27</td>
<td>783</td>
<td>0</td>
<td>2.86</td>
<td>18.53</td>
<td>6.46</td>
<td></td>
</tr>
<tr>
<td>5jdd</td>
<td>168</td>
<td>0/10</td>
<td>TO</td>
<td>10/10</td>
<td>3248±162</td>
<td>10/10</td>
<td>646±72</td>
<td>4.08</td>
<td>20662</td>
<td>0.04</td>
<td>-</td>
<td>-</td>
<td>5.02</td>
<td></td>
</tr>
<tr>
<td>3r8q</td>
<td>157</td>
<td>0/10</td>
<td>TO</td>
<td>10/10</td>
<td>3478±82</td>
<td>10/10</td>
<td>397±47</td>
<td>4.19</td>
<td>12762</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>8.76</td>
<td></td>
</tr>
<tr>
<td>4bxc</td>
<td>108</td>
<td>10/10</td>
<td>1,213±32</td>
<td>10/10</td>
<td>1352±40</td>
<td>10/10</td>
<td>216±24</td>
<td>0.26</td>
<td>2966</td>
<td>0</td>
<td>0.89</td>
<td>5.61</td>
<td>6.26</td>
<td></td>
</tr>
<tr>
<td>1f00</td>
<td>177</td>
<td>0/10</td>
<td>TO</td>
<td>10/10</td>
<td>2575±29</td>
<td>10/10</td>
<td>542±30</td>
<td>4.38</td>
<td>9749</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>4.75</td>
<td></td>
</tr>
<tr>
<td>2x8x</td>
<td>131</td>
<td>10/10</td>
<td>3312±111</td>
<td>4/10</td>
<td>2801±1071</td>
<td>10/10</td>
<td>521±58</td>
<td>4.16</td>
<td>69213</td>
<td>3.61</td>
<td>1.18</td>
<td>6.35</td>
<td>5.37</td>
<td></td>
</tr>
<tr>
<td>1xaw</td>
<td>66</td>
<td>0/10</td>
<td>TO</td>
<td>10/10</td>
<td>581±38</td>
<td>10/10</td>
<td>260±57</td>
<td>2.73</td>
<td>2804</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>2.23</td>
<td></td>
</tr>
<tr>
<td>5e10</td>
<td>74</td>
<td>10/10</td>
<td>1667±86</td>
<td>10/10</td>
<td>1412±21</td>
<td>10/10</td>
<td>132±24</td>
<td>0.26</td>
<td>1171</td>
<td>0</td>
<td>1.18</td>
<td>12.62</td>
<td>10.7</td>
<td></td>
</tr>
<tr>
<td>Idvo</td>
<td>82</td>
<td>10/10</td>
<td>940±22</td>
<td>10/10</td>
<td>940±22</td>
<td>10/10</td>
<td>197±25</td>
<td>2.90</td>
<td>34142</td>
<td>0.18</td>
<td>-</td>
<td>-</td>
<td>4.77</td>
<td>4.77</td>
</tr>
<tr>
<td>1lyq</td>
<td>67</td>
<td>10/10</td>
<td>2235±211</td>
<td>10/10</td>
<td>1304±40</td>
<td>10/10</td>
<td>280±26</td>
<td>1.67</td>
<td>17063</td>
<td>0.31</td>
<td>1.71</td>
<td>7.98</td>
<td>4.65</td>
<td></td>
</tr>
<tr>
<td>2af5</td>
<td>140</td>
<td>0/10</td>
<td>TO</td>
<td>10/10</td>
<td>2659±75</td>
<td>10/10</td>
<td>894±247</td>
<td>4.37</td>
<td>86029</td>
<td>0.60</td>
<td>-</td>
<td>-</td>
<td>2.97</td>
<td></td>
</tr>
<tr>
<td>Ing2</td>
<td>86</td>
<td>10/10</td>
<td>1065±69</td>
<td>10/10</td>
<td>547±18</td>
<td>10/10</td>
<td>260±26</td>
<td>1.14</td>
<td>38730</td>
<td>5.93</td>
<td>1.94</td>
<td>4.09</td>
<td>2.10</td>
<td></td>
</tr>
<tr>
<td>3sz7</td>
<td>79</td>
<td>0/10</td>
<td>TO</td>
<td>10/10</td>
<td>2952±276</td>
<td>10/10</td>
<td>320±50</td>
<td>3.11</td>
<td>82625</td>
<td>0.54</td>
<td>-</td>
<td>-</td>
<td>9.22</td>
<td></td>
</tr>
<tr>
<td>2gee</td>
<td>110</td>
<td>10/10</td>
<td>1,647±8</td>
<td>10/10</td>
<td>1276±19</td>
<td>10/10</td>
<td>286±38</td>
<td>1.68</td>
<td>5021</td>
<td>0</td>
<td>1.29</td>
<td>5.75</td>
<td>4.46</td>
<td></td>
</tr>
<tr>
<td>5e0z</td>
<td>73</td>
<td>10/10</td>
<td>621±15</td>
<td>10/10</td>
<td>995±18</td>
<td>10/10</td>
<td>105±8</td>
<td>0.16</td>
<td>999</td>
<td>0</td>
<td>0.62</td>
<td>5.91</td>
<td>9.47</td>
<td></td>
</tr>
<tr>
<td>1ly7</td>
<td>87</td>
<td>10/10</td>
<td>2148±9</td>
<td>10/10</td>
<td>945±24</td>
<td>10/10</td>
<td>457±85</td>
<td>2.91</td>
<td>83816</td>
<td>3.20</td>
<td>2.27</td>
<td>4.70</td>
<td>2.06</td>
<td></td>
</tr>
<tr>
<td>3lf9</td>
<td>72</td>
<td>10/10</td>
<td>1636±31</td>
<td>10/10</td>
<td>894±41.17</td>
<td>10/10</td>
<td>216±21</td>
<td>2.41</td>
<td>2667</td>
<td>0</td>
<td>1.82</td>
<td>7.57</td>
<td>4.13</td>
<td></td>
</tr>
<tr>
<td>3c3v</td>
<td>91</td>
<td>0/10</td>
<td>TO</td>
<td>10/10</td>
<td>2327±621</td>
<td>10/10</td>
<td>263±38</td>
<td>2.57</td>
<td>81574</td>
<td>0.15</td>
<td>-</td>
<td>-</td>
<td>8.84</td>
<td></td>
</tr>
<tr>
<td>1is1</td>
<td>107</td>
<td>10/10</td>
<td>3178±51</td>
<td>10/10</td>
<td>2274±421</td>
<td>10/10</td>
<td>329±48</td>
<td>3.45</td>
<td>63832</td>
<td>0.42</td>
<td>1.39</td>
<td>9.65</td>
<td>6.91</td>
<td></td>
</tr>
<tr>
<td>7eqz</td>
<td>89</td>
<td>10/10</td>
<td>1849±49</td>
<td>10/10</td>
<td>697±6</td>
<td>10/10</td>
<td>225±16</td>
<td>2.20</td>
<td>12768</td>
<td>0</td>
<td>2.65</td>
<td>8.21</td>
<td>3.097</td>
<td></td>
</tr>
<tr>
<td>4uos</td>
<td>118</td>
<td>10/10</td>
<td>2304±988</td>
<td>10/10</td>
<td>2109±62</td>
<td>10/10</td>
<td>469±79</td>
<td>5.20</td>
<td>58589</td>
<td>17.86</td>
<td>1.09</td>
<td>4.91</td>
<td>4.5</td>
<td></td>
</tr>
</tbody>
</table>

TO: TimeOut  
(1): VNS/LDS+CP (k++, l = 3)  
(2): UDGVNS (k++, l = 3)  
(3): UPDVNS (np, 1, k++, l = 3)  
(4): FixBB  
(5): Toulbar2
Anytime curve by operator
Cactus plot by operator

- DFBB
- LDS
- UDGVNS (k*2, lds*2)
- UDGVNS (k luby, lds*2)
- UDGVNS (k++/jump, lds*2)
- UDGVNS (k++, lds*2)
- UDGVNS (k++, lds luby)
- UDGVNS (k++, lds++)
- DGVNS (k++, lds = 3)

CPU time (in seconds)

Number of solved instances
Branch-and-Bound Search

- Upper Bound $UB$
- Lower Bound $LB(n)$
  
  $LB(n) = g(n) + h(n)$

- Prune if $LB(n) \geq UB$

$g(n)$ cost of the search path to $n$

$h(n)$ estimates the optimal cost below $n$

Depth first BB Tree
(Lawler & Wood66)
Parallelism enhances the best anytime Upper bound profile.
Parallelism enhances the best anytime Upper bound profil
UDGVNS provides the best anytime Upper bound profil
Depth-First Branch and bound

The Search Space algorithm: branch and bound (dffbb, btd)

Objective function:
\[ f(X) = \min_x \sum_{i=1}^n f_i(X) \]

Arc-costs are calculated based on cost components
The Search Space algorithm: branch and bound (dfffbb, btd)

Objective function:

\[ f(X) = \min_X \sum_{i=1}^{9} f_i(X) \]

Arc-costs are calculated based on cost components.
The Search Space algorithm: branch and bound (dffbb, btd)

Objective function:

\[ f(X) = \min_x \sum_{i=1}^n f_i(X) \]

Arc-costs are calculated based on cost components
Graphical Model

- $n$ variables
  - finite domains
    \[ X = \{ X_1, \ldots, X_n \} \]
    \[ x_i \in D_i, |D_i| \leq d \]
- $e$ local/global functions
  - scope, function with costs
    \[ F = \{ f_1, \ldots, f_e \} \]
- Costs $\in \{0, \ldots, +\infty\}$ finite or infinite integer

Minimize \[ \sum_{F} f_i(X) \] NP-hard

- Decision Diagramme
- Markov random field
- Belief propagation
- Bayesian Network
Random variables $X$ with discrete domains joint normalized probability distribution $p(X)$ defined as a product of positive real-valued functions:

- Optimal solution $\Rightarrow$ Most Probable Explanation (MPE) (with evidence or not)