Exploiting Tree Decomposition and Soft Local Consistency in Weighted CSP

Simon de Givry, Thomas Schiex, INRA
Gérard Verfaillie, ONERA
Toulouse, France
Radio Link Frequency Allocation Problem

Allocate frequencies to radio links such that the sum of interferences (soft binary distance constraints) is minimized.
Earth Observation Satellite Management Problem

Select images among a set of candidate images such that physical constraints (hard binary and ternary constraints) are satisfied and the sum of weights associated to selected images (soft unary constraints) is maximized.
Mendelian error detection in complex pedigree

Find a complete genotype of maximum a posteriori probability (MPE)

Conditional probability tables (soft unary, binary and ternary constraints)
Motivation of this work

Exploitation of the structure present in many real problems

temporal, spatial, causal ...
Framework: **weighted binary CSP**

- \((X, D, W)\)
  - \(X = \{x_1, \ldots, x_n\} \) \(n\) variables
  - \(D = \{D_1, \ldots, D_n\} \) \(n\) **finite domains** of maximum size \(d\)
  - \(W = \{W_1, \ldots, W_e\} \) \(e\) **cost functions**
    - \(W_{ij}, W_i, W_\emptyset\) with **scopes** \(\{x_i, x_j\}, \{x_i\}, \emptyset\)
    - \(W_{ij} : D_i \times D_j \rightarrow [0, k]\)
    - \(k\) is associated with completely **forbidden assignments**

- Find a **complete assignment** minimizing
  \[
  W_\emptyset + \sum_i W_i (a_i) + \sum_{ij} W_{ij} (a_i, a_j)
  \]

- **NP-hard** problem
Tree decomposition of a constraint graph

Tree of clusters
Tree decomposition of a constraint graph

The set of clusters covers the set of variables and the set of constraints

RLFAP
CELAR
SCEN-06
Tree decomposition of a constraint graph

Intersection between two connected clusters = Separator
Tree decomposition of a constraint graph

The assignment of a separator disconnects the problem into two independent subproblems
Tree decomposition of a constraint graph

Choice of a root cluster
rooted cluster tree
Exploitation of the tree decomposition

Dynamic programming
Cluster tree elimination
Time: $O(d^w)$, Space: $O(d^s)$

$w =$ largest cluster size
$s =$ largest separator size
Exploitation of the tree decomposition

Tree search

Pseudo-tree search (PTS)
Time: $O(d^h)$, Space: $O(nd)$

Constrained variable ordering
Graph-based backjumping

h = rooted cluster tree height

RLFAP
CELAR
SCEN-06
Exploitation of the tree decomposition

Backtrack Bounded by Tree Decomposition (BTD)
Time: $O(d^w)$, Space: $O(d^s)$

Constrained variable ordering
Graph-based backjumping
Graph-based learning

$w =$ largest cluster size
$s =$ largest separator size
Exploitation of the tree decomposition

\begin{align*}
\text{w} &= \text{largest cluster size} \\
\text{s} &= \text{largest separator size} \\
\text{k} &= \text{number of valuations}
\end{align*}

Tree search

Backtrack Bounded by Tree Decomposition (BTD+)

Time: $O(kd^w)$, Space: $O(d^s)$

Constrained variable ordering

Graph-based backjumping

Graph-based learning

Initial upper-bounds
Soft local consistency

- **Goal:** transforming a problem into an equivalent problem with a more explicit lower bound
- **Means:** enforcing a soft local consistency property by moving costs from binary constraints to unary constraints and to the zero-arity constraint $W_\emptyset$ (problem lower bound)

![Diagram](attachment:image.png)
Soft local consistency

\[ W_\emptyset = 0 \]

\[ k=2 \]

Cost projection from a binary to a unary constraint
Soft local consistency

\[ W_\emptyset = 0 \]

\[ k=2 \]

No idempotency property

Cost projection from a binary to a unary constraint
Soft local consistency

Two cost projections from two binary constraints to a unary constraint
Soft local consistency

Two cost projections
from two binary constraints
to a unary constraint
Soft local consistency

Cost projection from a unary to the zero-arity constraint

\[ W_\emptyset = 0 \]

\[ k=2 \]
Soft local consistency

\[ W_\emptyset = 1 \]

\[ k=2 \]

Cost projection from a unary to the zero-arity constraint
Soft local consistency

$W_\emptyset = 1$

$k=2$

Value removal
Soft local consistency

Value removal
Soft local consistency

\[ W_\emptyset = 1 \]

\[ k = 2 \]

Cost projection from a binary to a unary constraint
Soft local consistency

\[ W_{\emptyset} = 1 \]

\[ k=2 \]

Cost projection from a binary to a unary constraint
Soft local consistency

\[ W_\emptyset = 1 \]

\[ k=2 \]

Value removal
Soft local consistency

\[ W_{\emptyset} = 1 \]

\[ k = 2 \]

Value removal
Soft local consistency: various levels

- **NC***: Node Consistency
- **AC***: Arc Consistency
- **DAC***: Directed Arc Consistency
- **FDAC***: Full Directed Arc Consistency

\[ W_\emptyset = 1 \]

\[ k = 2 \]
Tree decomposition and soft local consistency

Two main difficulties:

- **Costs are moving between clusters and towards** $W_\emptyset$
  - Recorded subproblem lower bounds may be no longer valid
- **Value removals may affect any cluster**
  - No guarantee to improve the lower bound when revisiting the same subproblem
  - Loss in terms of theoretical complexity
Three approaches considered

- **Limited form of soft local consistency** (forward-checking)
  - FC-BTD (Time: $O(d^w)$, Space: $O(d^s)$)

- **Limited soft arc consistency**, with corrected recorded lower bounds and value removals limited to the current subproblem
  - **FDAC-BTD+** (Time: $O(kd^w)$, Space: $O(d^s)$)

- **Unlimited soft arc consistency**, without learning
  - **FDAC-PTS** (Time: $O(d^h)$, Space: $O(nd)$)
## Experimental results

### Radio Link Frequency Allocation Problem

The first time the whole SCEN-06 instance is solved by a search algorithm.

To be observed: small amount of memory required.

<table>
<thead>
<tr>
<th>RLFAP optimum</th>
<th>SUB$_1$</th>
<th>SUB$_4$</th>
<th>SCEN-06</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n, d, w, h$</td>
<td>2669</td>
<td>3230</td>
<td>3389</td>
</tr>
<tr>
<td>14, 44, 13, 14</td>
<td>22, 44, 19, 21</td>
<td>100, 44, 19, 67</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>time</th>
<th>#LB</th>
<th>time</th>
<th>#LB</th>
<th>time</th>
<th>#LB</th>
</tr>
</thead>
<tbody>
<tr>
<td>FC-BTD</td>
<td>1197</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>NC-BTD+</td>
<td>490</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>FDAC-BTD+</td>
<td>14</td>
<td>0</td>
<td>929</td>
<td>0</td>
<td>10,309</td>
<td>326</td>
</tr>
<tr>
<td>FDAC-PTS</td>
<td>14</td>
<td>n/a</td>
<td>851</td>
<td>n/a</td>
<td>-</td>
<td>n/a</td>
</tr>
<tr>
<td>MFDAC</td>
<td>14</td>
<td>n/a</td>
<td>984</td>
<td>n/a</td>
<td>-</td>
<td>n/a</td>
</tr>
</tbody>
</table>
Conclusion

- **FDAC-BTD+**: Cluster tree problem decomposition, Tree search, Graph-based backjumping and learning, (Limited) soft arc consistency enforcing, Initial upper-bounds

- **Cluster tree decomposition** and **soft local consistency** can be combined, but various technical options can be considered, and must be more widely experimented.
Random Binary Trees of Cliques

Parameters
- Clique size \((w+1) = 10\)
- Separator size \(s\)
- Clique tree height \(h' = 3\)
- Domain size \(d = 5\)

Constraint tightness

Tree decomposition based on Maximum Cardinality Search

Root selection minimizing tree-height \(h\)
Random Binary Trees of Cliques

CPU-time (sec.)

Small-size separator $s = 2$

Memory space (#lb)

$\text{dom/deg dynamic variable ordering. CPU-time to find the optimum and prove its optimality}$
Random Binary Trees of Cliques

CPU-time (sec.)

Medium-size separator $s = 5$

Memory space (#lb)

$n=40$ $d=5$ $c=32\%$ $s=5$ $w=9$ $h=20$

dom/deg dynamic variable ordering. CPU-time to find the optimum and prove its optimality.
Random Binary Trees of Cliques

CPU-time (sec.)

Large-size separator $s = 7$

Memory space (#lb)

dom/deg dynamic variable ordering. CPU-time to find the optimum and prove its optimality.