Steiner Tree Problems with Side Constraints Using Constraint Programming

AAAI 2016

Diego de Uña, Graeme Gange, Peter Schachte and Peter J. Stuckey

February 14, 2016

University of Melbourne — CIS Department
1. Background

   Example of a Steiner Tree Problem

   The Steiner Tree Problem

   Constraint Programming

2. Steiner Tree Propagator

   Steiner Tree Propagator

   Some of the main ideas: Treeness

   Some of the main ideas: Lower bounds
Background
Example of a Steiner Tree Problem

- Given an electrical network, connect a given subset of the nodes with minimum cost.

Fig. 1: Base solution. All weights are 1 (solid) or 9 (dashed). Cost 10
Example of a Steiner Tree Problem

- Given an electrical network, connect a given subset of the nodes with minimum cost.
- Side constraint: for each pair of nodes there need to be 2 edge-disjoint alternative paths.

Fig. 1: Problem with side constraint between $b$ and $g$
Example of a Steiner Tree Problem

- Given an electrical network, connect a given subset of the nodes with minimum cost.
- Side constraint: for each pair of nodes there need to be 2 edge-disjoint alternative paths.

![Diagram of a Steiner Tree Problem](image)

Fig. 1: Solution (solid lines): all pairs of black nodes have at least two edge-disjoint alternative routes. Cost 17
The Steiner Tree Problem

Definition:
- Given a weighted graph $G$ and a set $M$ of mandatory nodes (aka terminals).
- Find a tree $T$ of minimum weight s.t. all the nodes in $M$ are in $T$.
- Proved NP-complete by Karp.
- Generalization of MST.

Useful for:
- Electrical networks (Agrawal et al., 1995).
- VLSI design (Hwang et al., 1992).
- Phylogenetics (Winter, 1987).
- Broadcasting (quality) (Karpinski et al., 2003).
- ... c.f. *The Steiner Tree Problem* (book, (Hwang et al., 1992)).
The STP is never pure

- State of the art in the *pure* STP achieved by (Polzin et al., 2001).
  - Reductions remove edges that are useless.

- In the real world: STP is never alone.
  Examples: “Find STP such that...
  - Maximum/minimum degree on the nodes.
  - Mandatory nodes must be leafs.
  - Capacity constraint.
  - Connectivity constraints.
  - Diameter constrained.
  - Geometric constraints.
  - … anything you can imagine.
  - and all combined!
Constraint Programming

- High-level programming paradigm:
  - Programmer becomes a *modeller*.
  - Uses a *solver* to solve his model.
- Versatile and reusable code.
- Fast, low memory usage.

Our choice:
- Modelling language: MiniZinc (Nethercote et al., 2007).
- Solver: Chuffed (Chu, 2011).

```plaintext
% Baking cakes
var 0..100: b; % nb banana cakes
var 0..100: c; % nb chocolate cakes
% flour
constraint 250*b + 200*c <= 4000;
% bananas
constraint 2*b <= 6;
% sugar
constraint 75*b + 150*c <= 2000;
% butter
constraint 100*b + 150*c <= 500;
% cocoa
constraint 75*c <= 500;
% maximize our profit
solve maximize 400*b + 450*c;
```

Fig. 2: Example of MiniZinc code
A CP solver works by interleaving search and propagation.

**Search**: try all possible values for all variables. ⇒ Too slow!

**Propagation**: filter the possible values of variables to not try too many options. ⇒ Reduce the search space

### Example of propagation

\[ a \in \{2, 3, 4\}, \ b \in \{1, 2, 3\} \text{ such that } a + 2b \leq 4 \land a \neq b \]

Search: try \( a = 2 \)

\[ \downarrow \text{Propagation: } a \neq b \Rightarrow b \in \{1, 2, 3\} \]

\[ \downarrow \text{Propagation: } a + 2b \leq 4 \Rightarrow b \in \{1, 3\} \]

- **Propagators** DO NOT solve the problem, just filter domains.
- **LCG**: Propagators can generate explanations/clauses.
Steiner Tree Propagator
Steiner Tree Propagator

Graphs in CP:

- Boolean variables for edges and nodes.
- If true \(\Rightarrow\) part of the solution

The global constraint

\[
\text{steiner\_tree}(\{n|n \in V\}, \{e|e \in E\}, \text{adj, ends, ws, w})
\]

- Ensures that the solution is a tree (checks failure and propagates).
- Ensures the solution has cost \(w\).
Some of the main ideas: Treeness

- **reachable**: run DFS to detect unreachable nodes. Explain: run DFS from unreachable node ⇒ find the “border”.

\[ \neg e_5 \land \neg e_6 \land \neg e_7 \land n \land o \Rightarrow \text{fail} \]

---

Fig. 3: Explaining reachability
Some of the main ideas: Treeness

- **articulations**: bridges and articulations must be in the solution.

Explain: During Tarjan’s algorithm record mandatory nodes. Run DFS’s to find forbidden edges.

\[ c \land h \Rightarrow d \]
\[ c \land h \land \neg e_4 \Rightarrow e_5 \]
\[ e \land c \Rightarrow e_6 \]
\[ b \land c \land \neg e_4 \Rightarrow e_3 \]

---

**Fig. 3**: Explaining articulations and bridges
Some of the main ideas: Treeness

- **cycle**: prevent and detect cycles by maintaining a UF. 
  - Explain: “fancy” UF remembers connections (no path compression) 
  - ⇒ can retrieve entire paths.

\[
e_4 \land e_5 \Rightarrow \neg e_2 \\
e_4 \land e_5 \land e_3 \Rightarrow \neg e_1
\]

Fig. 3: Cycle prevention
Some of the main ideas: Lower bounds

If the lower bound is higher than $\bar{w}$, we can stop the search here.

- Shortest path based LB ($SPLB$):
  1) Contract
  2) Connect mandatory pairs by shortest paths
  ↓ Explain: all mandatory edges + all forbidden edges that used to be in a shortest path

\[ LB(G') = \frac{1}{2} \sum_{u \in M'} spc_{G'}(u) \]
Some of the main ideas: Lower bounds

If the lower bound is higher than $\bar{w}$, we can stop the search here.

- **LP based lower bound ($L_{PLB}$):**
  Solve pure STP by LP, row generation (Aneja, 1980).
  ▼ Explain: any edge with non-zero reduced cost.

\[
\minimize \sum_{e \in E} ws[e] \times \text{bool}(e) \text{ such that:}
\]
\[
\forall W, \sum_{e \in \delta(W)} \text{bool}(e) \geq 1
\]

**Fig. 4:** Cut formulation of the STP
Some results

<table>
<thead>
<tr>
<th></th>
<th>Conflicts</th>
<th>Nodes</th>
<th>Propagations</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPLB</td>
<td>16713</td>
<td>144110</td>
<td>33533492</td>
<td>47.63 (2)</td>
</tr>
<tr>
<td>LPLB</td>
<td>11330</td>
<td>135492</td>
<td>18648420</td>
<td>37.61 (2)</td>
</tr>
<tr>
<td>LPLB (n.l)</td>
<td>20376</td>
<td>207503</td>
<td>27413586</td>
<td>43.19 (2)</td>
</tr>
<tr>
<td>Sp+LPLB</td>
<td>9550</td>
<td>122314</td>
<td>16627051</td>
<td>42.67 (2)</td>
</tr>
<tr>
<td>NOLB</td>
<td>15343</td>
<td>191803</td>
<td>27413586</td>
<td>50.96 (2)</td>
</tr>
<tr>
<td>NoProp</td>
<td>96642</td>
<td>466502</td>
<td>331019078</td>
<td>330.34(3)</td>
</tr>
<tr>
<td>Choco3</td>
<td>184248</td>
<td>184307</td>
<td>4937258941</td>
<td>14790.80 (10)</td>
</tr>
</tbody>
</table>

Table 1: Results for the GoSST.

<table>
<thead>
<tr>
<th></th>
<th>Conflicts</th>
<th>Nodes</th>
<th>Propagations</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPLB</td>
<td>33</td>
<td>60</td>
<td>730</td>
<td>1.37</td>
</tr>
<tr>
<td>LPLB</td>
<td>29</td>
<td>46</td>
<td>565</td>
<td>0.30</td>
</tr>
<tr>
<td>LPLB (n.l)</td>
<td>405</td>
<td>573</td>
<td>5570</td>
<td>2.29 (1*)</td>
</tr>
<tr>
<td>Sp+LPLB</td>
<td>26</td>
<td>44</td>
<td>524</td>
<td>0.40</td>
</tr>
<tr>
<td>NOLB</td>
<td>55</td>
<td>101</td>
<td>1354</td>
<td>1.62</td>
</tr>
<tr>
<td>NoProp</td>
<td>20644413</td>
<td>21174280</td>
<td>2637608115</td>
<td>5040.19 (8)</td>
</tr>
<tr>
<td>Choco3</td>
<td>398026</td>
<td>384652</td>
<td>628843</td>
<td>26.27 (3)</td>
</tr>
</tbody>
</table>

Table 2: Results for the TSTP.
Some results

<table>
<thead>
<tr>
<th></th>
<th>Conflicts</th>
<th>Nodes</th>
<th>Propagations</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPLB</td>
<td>16713</td>
<td>144110</td>
<td>33533492</td>
<td>47.63 (2)</td>
</tr>
<tr>
<td>LPLB</td>
<td>11330</td>
<td>135492</td>
<td>18648420</td>
<td><strong>37.61</strong> (2)</td>
</tr>
<tr>
<td>LPLB (n.l)</td>
<td>20376</td>
<td>207503</td>
<td>27413586</td>
<td>43.19 (2)</td>
</tr>
<tr>
<td>Sp+LPLB</td>
<td><strong>9550</strong></td>
<td><strong>122314</strong></td>
<td><strong>16627051</strong></td>
<td>42.67 (2)</td>
</tr>
<tr>
<td>NOLB</td>
<td>15343</td>
<td>191803</td>
<td>27413586</td>
<td>50.96 (2)</td>
</tr>
<tr>
<td>NoProp</td>
<td>96642</td>
<td>466502</td>
<td>331019078</td>
<td>330.34 (3)</td>
</tr>
<tr>
<td>Choco3</td>
<td>184248</td>
<td>184307</td>
<td>4937258941</td>
<td>14790.80 (10)</td>
</tr>
</tbody>
</table>

Table 1: Results for the GoSST.

<table>
<thead>
<tr>
<th></th>
<th>Conflicts</th>
<th>Nodes</th>
<th>Propagations</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>B05 LPLB</td>
<td>33</td>
<td>212</td>
<td>1554</td>
<td>0.07</td>
</tr>
<tr>
<td>B05 LPLB (n.l)</td>
<td>42535406</td>
<td>85070845</td>
<td>564478786</td>
<td>5 hours</td>
</tr>
</tbody>
</table>

Table 2: LCG can save a lot of search!
### Table 3: Results for the pure Steiner Tree problem.

<table>
<thead>
<tr>
<th>Method</th>
<th>Conflicts</th>
<th>Nodes</th>
<th>Propagations</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ES10FST</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPLB</td>
<td>12</td>
<td>15</td>
<td>123</td>
<td>0.01</td>
</tr>
<tr>
<td>LPLB</td>
<td>11</td>
<td>14</td>
<td>101</td>
<td>0.01</td>
</tr>
<tr>
<td>LPLB (n.l.)</td>
<td>12</td>
<td>22</td>
<td>112</td>
<td>0.01</td>
</tr>
<tr>
<td>Sp+LPLB</td>
<td>10</td>
<td><strong>13</strong></td>
<td><strong>99</strong></td>
<td><strong>0.01</strong></td>
</tr>
<tr>
<td>NOLB</td>
<td>13</td>
<td>138</td>
<td>134</td>
<td>0.01</td>
</tr>
<tr>
<td>NoProp</td>
<td>10686</td>
<td>1079</td>
<td>797998</td>
<td>1.75 (4)</td>
</tr>
<tr>
<td>Choco3</td>
<td>130</td>
<td>138</td>
<td>198</td>
<td><strong>0.01</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Conflicts</th>
<th>Nodes</th>
<th>Propagations</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ES20FST</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPLB</td>
<td>729</td>
<td>816</td>
<td>6383</td>
<td>1.35 (2)</td>
</tr>
<tr>
<td>LPLB</td>
<td>492</td>
<td>534</td>
<td>3723</td>
<td>0.49</td>
</tr>
<tr>
<td>LPLB (n.l.)</td>
<td>632</td>
<td>1080</td>
<td>4636</td>
<td><strong>0.47</strong></td>
</tr>
<tr>
<td>Sp+LPLB</td>
<td><strong>477</strong></td>
<td><strong>520</strong></td>
<td><strong>3626</strong></td>
<td>0.74</td>
</tr>
<tr>
<td>NOLB</td>
<td>746</td>
<td>831</td>
<td>7016</td>
<td>0.67 (2)</td>
</tr>
<tr>
<td>NoProp</td>
<td>604055</td>
<td>604200</td>
<td>43168067</td>
<td>1065 (11)</td>
</tr>
<tr>
<td>Choco3</td>
<td>45434</td>
<td>46175</td>
<td>67780</td>
<td>1.10 (3)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Conflicts</th>
<th>Nodes</th>
<th>Propagations</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>B</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPLB</td>
<td>55979</td>
<td>83120</td>
<td>531033</td>
<td>22.25 (2)</td>
</tr>
<tr>
<td>LPLB</td>
<td><strong>8097</strong></td>
<td><strong>9157</strong></td>
<td><strong>53731</strong></td>
<td><strong>5.17</strong></td>
</tr>
<tr>
<td>LPLB (n.l.)</td>
<td>18000</td>
<td>37096</td>
<td>110711</td>
<td>6.99 (1)</td>
</tr>
<tr>
<td>Sp+LPLB</td>
<td>8102</td>
<td>10372</td>
<td>68814</td>
<td>7.11</td>
</tr>
<tr>
<td>NOLB</td>
<td>114538</td>
<td>192343</td>
<td>1460071</td>
<td>17.80 (2)</td>
</tr>
<tr>
<td>NoProp</td>
<td>177364998</td>
<td>179890423</td>
<td>18203144792</td>
<td>18000 (11)</td>
</tr>
<tr>
<td>Choco3</td>
<td>80406257</td>
<td>80406288</td>
<td>152270771</td>
<td>3339.66 (5)</td>
</tr>
</tbody>
</table>
Conclusion

Contributions:

- First CP approach to the problem.
- Tree propagator with explanations.
- New fast lower bounding technique based on shortest paths, with explanations.
- Use of LP formulation as lower bound, added explanations.
Thank you for your attention.

Questions?
Proof of SPLB (just in case)

- In a Steiner Tree, there is a path such that contains 2 terminals and only one node of degree 3 or more.
- Removing such path from the tree yields a tree.
- Do:
  - Find 2 mandatory nodes such that they are connected by one such path.
  - Remove that path.
  - Repeat until no pairs are left.

The cost of those paths can only be ≤ to the cost of the tree.
- Shortest paths are even cheaper than those paths.
- Because we count some paths twice, we divide by 2.