Minimum Weight Constrained Forest Problem

Xiaoyun Ji, John E. Mitchell
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY, USA
jix@rpi.edu, mitchj@rpi.edu

2005 Optimization Days
Montreal, Canada
May 09, 2005

Slide 1

Contents
- Minimum Weight Constrained Forest Problem (MWCF)
- Applications and Related Problems of MWCF
- IP Formulation for MWCF
- Polyhedral Theory for MWCF
- Branch and Cut Algorithm
- Computational Results
- Conclusions

Slide 2

Problem Definition
Minimum Weight Constrained Forest Problem (MWCF)
- Given an integer $S > 0$ and a complete graph $G = (V, E)$
  with weights $c_e$ on each edge $e$.
- Look for a spanning forest of graph $G$ with each tree in the
  forest spanning at least $S$ vertices, so as to minimize the
  weight of the spanning forest.

NP-hard when $S \geq 4$.
[Imielińska, Kalantari and Khachiyan, 1993]

Slide 3

An Example

Figure 1: One Solution for a problem of size $n = 50, S = 4$
Applications of clustering with minimum size requirements

- Micro-aggregation for Statistical Data
  Statistical data is released to the public in small groups so that the confidentiality of respondents are protected and the informational content of the data are preserved as much as possible.
- Political Districting
  Compact, contiguous, and balanced districts
- Sports Team Alignment
- Telecommunication Network Design

Related Graph Partition Problems

**Minimum weight balanced spanning forest problem**
[Ali and Huang, 1991]

- The number of trees are fixed and the number of vertices in each tree cannot differ by more than 1.
- Problem is set up as an IP. Lagrangian relaxation and heuristics are used to find lower bound and upper bounds.
- $n = 100$, edge density = 20%, gap = 3%, iterations = 700-1400

**k-tree partition**
[Buttmann-Beck and Hassin, 1998]

- To partition $V$ into $p$ subsets of given size, so as to minimize the spanning forest weight.
- Heuristic Algorithm is proposed.

**Min-max spanning forest problem**
[Yamada, Takahashi and Kataoka, 1997]

- The root of each tree is given.
- The objective is to minimize the maximum of the tree weights.
- A branch and bound method based on a combinatorial methods to establish lower bound.

**Clique partition with minimum size requirement**
[Ji and Mitchell, 2005]

- Each cluster has to have at least $S$ vertices, and the clique weight is used to measure the weight of each cluster.
- Problem is set up as an IP and solved by branch-and-cut.
- $n = 100$, gap = 3.77%, time = 176s
Integer Programming Formulation

Define a binary variable \( x_e \) with

\[
x_e = \begin{cases} 
1 & \text{if } e \text{ is in the spanning forest} \\
0 & \text{otherwise}
\end{cases}
\]

Notation

- number of vertices: \( n = |V| \)
- number of edges: \( \binom{n}{2} = \frac{n(n-1)}{2} \)
- \( k, r : n = kS + r, 0 \leq r \leq S - 1 \)

Our IP formulation for MWCF is

\[
\begin{align*}
\text{min} & \quad \sum_{e \in E} c_e x_e \\
\text{s.t.} & \quad \sum_{e \in E(Q)} x_e \leq |Q| - 1, \quad \forall Q \subseteq V \quad (1) \\
& \quad \sum_{e \in E(W)} x_e + \sum_{e \in \delta(W)} x_e \geq |W| - \left\lfloor \frac{|W|}{S} \right\rfloor, \quad \forall W \subseteq V \quad (2) \\
& \quad x_e \in \{0, 1\}
\end{align*}
\]

Where

- \( E(W) = \{(u, v) | u \in W, v \in W\} \)
- \( \delta(W) = \{(u, v) | u \in W, v \notin W\} \)

(1) Cycle Constraint.

(2) Flower Constraint.

Figure 2: Flower constraint violated on vertex set \{21, 23\}.

An Example for Flower Constraint

Initial Relaxation

\[
\begin{align*}
\text{min} & \quad \sum_{e \in E} c_e x_e \\
\text{s.t.} & \quad \sum_{e \in E(Q)} x_e \leq |Q| - 1 \quad (3) \\
& \quad \sum_{e \in E(V)} x_e \geq 1, \quad \forall v \in V \quad (4) \\
& \quad \sum_{e \in E(V)} x_e \geq |V| - \left\lfloor \frac{|V|}{S} \right\rfloor \quad (5) \\
& \quad 0 \leq x_e \leq 1 \quad (6)
\end{align*}
\]

Cycle Constraint – Constraint (3)

Flower Constraint – Constraints (4) and (5)
The MWCF Polytope

\[ F(G, S) := \text{conv}\{x \in \{0,1\}^n : x \text{ is the incidence vector of a spanning forest } F \text{ on } G, \text{each tree in the forest has at least } S \text{ nodes}\} \]

\[ F(G, S) \text{ is full dimensional. } \dim(F(G, S)) = n(n-1)/2. \]

Facet Defining Constraints

We have shown

- \[ x_e \geq 0 \] is a facet for \( F(G, S) \).
- Cycle constraint is a facet for \( F(G, S) \), where \(|V| \geq 2S\), if and only if \( Q = V \) or \(|Q| \leq n - S\).
- Flower constraint is a facet for \( F(G, S) \), where \( n = |V| = kS + r \) and \(|W| = pS + q\), if and only if \( W = V \) or \( q \geq 1 \) and \( p \leq k - 2\).

Branch and Cut Framework

1. Initialize: Set up initial relaxation.
2. Solve the current relaxation using CPLEX.
3. Use a primal heuristic to generate a feasible solution. Update the upper bound if necessary.
4. Call separation routine to add in violated constraints. If none is found then go to Step 5. Otherwise, add in constraints, and go to Step 2.
5. Start Branching using MINTO.

Separation Routine

1. Cycle inequalities:
   - can be identified in \( O(n^4) \) time [Padberg and Wolsey, 1983].
   - Because of the objective function, it is not often violated.
2. Flower inequalities:
   - Important: 2 more constraints can change the computation time for a problem of size \((n = 50, S = 4)\) from \((1241 \text{ nodes, 40 seconds})\) to \((3 \text{ nodes, 6 seconds})\).
   - Spend more effort to look for them.
Heuristic Algorithm for finding violated flower constraints

1. Consider a graph $K_n$ with the LP solution $x$ as edge weights.
2. Identify 2 types of vertices in this graph.
   - The vertices that are only connected with 2 fractional edges and they sum up to 1. (Eg, vertices 21, 23, etc.)
   - The vertices that are connected with one edge of value one and nothing else, at the same time, the other end point of this edge is connected to no other edges of value one. (Eg, vertices 24, 27, etc.).
3. Find the shortest path between every pair of vertices found above, and check if any path violates flower constraints.

Computational Results

Experiment Data

- Type I: Edge weight = Euclidean distance between data points
  - $x \sim U[1,100]$
  - $y \sim U[1,100]$
- Type II: Edge Weight ~ uniform distribution in $[1,100]$
- Type III: Micro-aggregation Data
  - $x \sim \exp(\lambda)$
  - $y \sim \exp(\lambda)$
- Type IV: Micro-aggregation Data
  - $u \sim \exp(\lambda)$, $v \sim U[0,1]$
  - $x = u$, $y = uv$

Experiment Setup

- Experiments on a Sun Ultra 10 workstation
- Algorithm coded in MINTO
- Runtime reported in seconds.
- Every set of experiments: $S=4$, $r=1,2,3$; $S=7$, $r=1,...,6$
- For problems not solved to optimality at the cutting plane stage, the branch and cut stage has a time limit of 500 seconds.

<table>
<thead>
<tr>
<th>$k = \ell + 2$</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>75</th>
<th>105</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heuristic Alg.</td>
<td>Gap</td>
<td>Time</td>
<td>Gap</td>
<td>Time</td>
<td>Gap</td>
<td>Time</td>
<td>Gap</td>
</tr>
<tr>
<td>Time</td>
<td>0.0216</td>
<td>0.0429</td>
<td>0.0729</td>
<td>0.1107</td>
<td>0.1537</td>
<td>0.2376</td>
<td>0.4592</td>
</tr>
<tr>
<td>Cutting Plane</td>
<td>Total Instances</td>
<td>Solved exactly</td>
<td>Gap</td>
<td>Time</td>
<td>Cycle</td>
<td>Flower</td>
<td>Flow</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>0.61%</td>
<td>1.15%</td>
<td>1.32%</td>
<td>1.90%</td>
<td>1.69%</td>
<td>3.64%</td>
<td>4.91%</td>
<td>4.04%</td>
</tr>
<tr>
<td>9.92</td>
<td>28.10</td>
<td>64.37</td>
<td>99.05</td>
<td>142.50</td>
<td>183.68</td>
<td>403.43</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>56</td>
<td>84</td>
<td>103</td>
<td>125</td>
<td>156</td>
<td>147</td>
<td></td>
</tr>
<tr>
<td>311</td>
<td>515</td>
<td>618</td>
<td>1013</td>
<td>1183</td>
<td>1120</td>
<td>1140</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>65</td>
<td>78</td>
<td>91</td>
<td>97</td>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>65</td>
<td>78</td>
<td>91</td>
<td>97</td>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>65</td>
<td>78</td>
<td>91</td>
<td>97</td>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>17.85</td>
<td>53.67</td>
<td>171.20</td>
<td>401.86</td>
<td>403.44</td>
<td>492.98</td>
<td>513.18</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>20</td>
<td>26</td>
<td>36</td>
<td>40</td>
<td>49</td>
<td>29</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 1: Branch-and-Cut Results on MWCF Type I Problems for $S = 4$
Table 2: Branch-and-Cut Results on MWCF Type I Problems for \( n = 121 - 123 \) and \( 5 = 7 - 40 \)

| \( k = |S| \) | 21-123 | 121-123 | 121-123 | 121-123 | 121-123 |
|---|---|---|---|---|---|
| \( k = |S| \) | 0 | 7 | 10 | 12 | 14 |

Table 3: Branch-and-Cut Results on MWCF Type II Problems for \( S = 4 \)

<table>
<thead>
<tr>
<th>( n \times )</th>
<th>21-123</th>
<th>121-123</th>
<th>121-123</th>
<th>121-123</th>
<th>121-123</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k = )</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
</tbody>
</table>

Table 4: Branch-and-Cut Results on MWCF Type III Problems for \( S = 4 \)

| \( k = \) | 20 | 30 | 40 | 50 | 60 |

Table 5: Branch-and-Cut Results on MWCF Type IV Problems for \( S = 4 \)

| \( k = \) | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |

Conclusion

- Studied a new class of graph partition problem: Graph Partition Problems with minimum size requirement
- Discussed MWCF polytope structure.
- Established facets for MWCF polytope.
- Solved MWCF successfully using branch-and-cut.