Integer Programming Formulations for the Steiner Forest Problem

Sarah Lewin François Margot Daniel Schmidt

12.01.2017

21st Combinatorial Optimization Workshop

Aussois

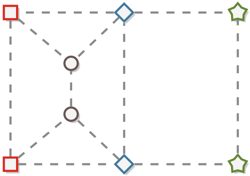
The Steiner Forest Problem

Input

Graph	$G = (V, E)$
Terminal sets	$T_1, \ldots, T_K \subseteq V$
Edge costs	$c : E \to \mathbb{R}_{\geq 0}$

Output

Minimum cost forest $F \subseteq E$ where for all $k = 1, \ldots, K$ and all $s, t \in T_k$, there is a s-t-path in F.



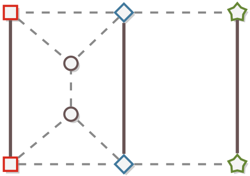
The Steiner Forest Problem

Input

Graph	$G = (V, E)$
Terminal sets	$T_1, \ldots, T_K \subseteq V$
Edge costs	$c : E \to \mathbb{R}_{\geq 0}$

Output

Minimum cost forest $F \subseteq E$ where for all $k = 1, \ldots, K$ and all $s, t \in T_k$, there is a s-t-path in F.



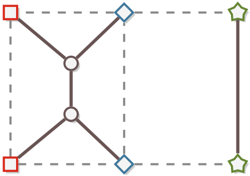
The Steiner Forest Problem

Input

Graph	$G = (V, E)$
Terminal sets	$T_1, \ldots, T_K \subseteq V$
Edge costs	$c : E \to \mathbb{R}_{\geq 0}$

Output

Minimum cost forest $F \subseteq E$ where for all $k = 1, \ldots, K$ and all $s, t \in T_k$, there is a s-t-path in F.



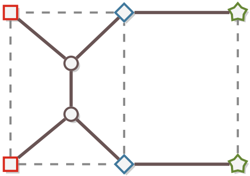
The Steiner Forest Problem

Input

Graph	$G = (V, E)$
Terminal sets	$T_1, \ldots, T_K \subseteq V$
Edge costs	$c : E \to \mathbb{R}_{\geq 0}$

Output

Minimum cost forest $F \subseteq E$ where for all $k = 1, \ldots, K$ and all $s, t \in T_k$, there is a s-t-path in F.



Standard MIP formulation(s)

Relevant Cut-Sets

A set $S \subseteq V$ is relevant if for some $k \in \{1, \dots, K\}$:

$$S \cap T_k \neq \emptyset \text{ and } (V \setminus S) \cap T_k \neq \emptyset$$

Cut-Set formulation

$$\begin{aligned} \min \sum_{\{i,j\} \in E} c_{ij} x_{ij} & & \\ \sum_{\{i,j\} \in \delta(S)} x_{ij} \geq 1 & \quad \text{for all} & \text{relevant } S \subseteq V \\ x_{ij} \in \{0, 1\} & \quad \text{for all} & \{i, j\} \in E \end{aligned}$$

Cut-Set formulation example

Eight instances, one hour on standard computer with 3GB of RAM:

no.	$\|V\|$	$\|E\|$	K	cpu [s]	bound[1]
1	10	20	3	< 1	84.97%
2	10	20	4	< 1	71.69%
3	10	30	3	< 1	72.80%
4	10	30	4	< 1	72.28%
5	10	45	3	< 1	67.82%
6	10	45	4	< 1	76.62%
7	15	50	4	< 1	72.47%
8	15	105	4	< 1	73.08%

[1] lp bound over best integer solution

Improvement: Könemann, Leonardi, Schäfer, van Zwam

Idea

- Define unique responsible terminal pair for each cut
- May choose to not select edge on some cut
- If so: Pay for direct connection of responsible terminal pair

Yields lower bound.

Example run...

Eight instances, one hour on standard computer with 3GB of RAM:

no.	$\|V\|$	$\|E\|$	K	cpu [s]	bound[2]
1	10	20	3	< 1	73.08%
2	10	20	4	< 1	67.32%
3	10	30	3	< 1	62.20%
4	10	30	4	< 1	66.58%
5	10	45	3	< 1	63.86%
6	10	45	4	< 1	67.53%
7	15	50	4	< 1	59.83%
8	15	105	4	< 1	63.68%

[2] lp bound over best integer solution

Magnanti, Raghavan 2005

Idea

Pick a root node ρ_k for each terminal set T_k. Then, any Steiner Forest can be directed such that in all connected components, all arcs point away from highest index root node

Can build flow based MIP-formulation

- Define independent flow for each pair of a root ρ_k and a terminal t (including roots)
- Certain combination of flows are not allowed to use the same edges
- Huge number of combinations, one constraint per combination

Magnanti, Raghavan 2005: Example run

Eight instances, one hour on standard computer with 3GB of RAM:

no.	$\|V\|$	$\|E\|$	K	cpu [s]	bound[3]
1	10	20	3	< 1	100.00%
2	10	20	4	< 1	100.00%
3	10	30	3	< 1	100.00%
4	10	30	4	< 1	100.00%
5	10	45	3	< 1	100.00%
6	10	45	4	< 1	100.00%
7	15	50	4	3600	72.87%
8	15	105	4	3600	75.21%

[3] lp bound over best integer solution

The Problem...

Quote from [MR2005]

"The number of constraints [...] is extremely large although still polynomial for fixed K. Consequently, to use this formulation to solve large scale problems, we might need to judiciously add a subset of the constraints [...] (adding others as necessary). This is a possible topic for future research."

Yet: No efficient separation algorithm known...

Another idea.

Steiner *Tree* formulation

[Lucena 91], [Goemans 94], [Margot, Prodon, Liebling 94]

$$
\begin{aligned}
\min \sum_{e \in E} c_{ij} x_{ij} & & \\
\sum_{i \in V} y_i - \sum_{\{i,j\} \in E} x_{ij} &= 1 & \\
\sum_{i \in S} y_i - \sum_{\{i,j\} \in E[S]} x_{ij} &\geq y_i & \text{for all } S \subseteq V, i \in V \\
x_{ij} &\in \{0,1\} & \text{for all } \{i,j\} \in E \\
y_i &\in \{0,1\} & \text{for all } i \in V \\
y_i &= 1 & \text{for all } i \in T
\end{aligned}
$$

Building a cycle free graph

First set of variables

x_{ij}	include edge $\{i, j\}$ in forest?	(binary)
y_i	connect node i to forest?	(binary)
R	number of connected components of forest	(integer)

First set of constraints

- No cycles [previous slide]:

$$\sum_{i \in S} y_i \quad - \sum_{\{i,j\} \in E[S]} x_{ij} \geq y_i \qquad \text{for all} \quad S \subseteq V, i \in V$$

- Compute number of connected components:

$$\sum_{i \in V} y_i \quad - \sum_{\{i,j\} \in E} x_{ij} = R$$

Ensuring proper connections

Second set of variables (all binary)

$w_{k\ell}$ Are T_k and T_ℓ in the same connected component?
z_{ik} Is node i assigned to the same connected component as T_k?

Second set of constraints

- Set $w_{k\ell}$ if edge between T_k and T_ℓ:

$$w_{k\ell} \geq x_{ij} - (1 - z_{ik}) - (1 - z_{j\ell}) \quad \forall \{i, j\} \in E, \ \forall k < l$$

- Set z_{ik} if edge between node i and node j from T_k.

$$z_{ik} \geq x_{ij} + z_{jk} - 1 \qquad \forall \{i, j\} \in E, \forall\ k$$

Setting active roots

Pick root ρ_k for each Terminal set T_k

Third set of variables

r_k Is root ρ_k active?

Third set of constraints

- Only highest index root in component can be active:

$$r_k \leq 1 - w_{k\ell} \qquad \text{for all } \; k < l$$

- There are exactly R active roots

$$\sum_{k=1}^{K} r_k = R$$

Wrapping things up...

Additional constraints

- 3-cycle constraints to make w transitive
- fix $y_i = 1$ for all terminal nodes
- fix $z_{ik} = 1$ for all k and all $i \in T_k$

Resulting total size

- $O(|E| + |V| \cdot K + K^2)$ variables
- $O(K^2 \cdot |E|)$ initial constraints
- Plus: $O(K^3)$ 3-cycle constraints (via efficient separation)
- Plus: $O(|V| \cdot 2^{|V|})$ subtour elimination constraints (via efficient separation)

Lower Bounds: Comparison on Small Instances

	#solved	avg. cpu [s]	avg. bound	#best
Cut-Set	85	< 1	71.76%	1
MR 2005	50	162.16	99.97%	50
KLSvZ 2008	85	< 1	65.94%	0
our formulation	85	< 1	93.20%	46

Benchmark II

- 85 instances
- 10 – 20 nodes
- 20 – 105 edges
- up to 6 terminal sets
- Standard computer, CPLEX, 3GB of RAM

Lower Bounds: Comparison on Large Instances

	#solved	avg. cpu [s]	avg. bound	#best
Cut-Set	8	< 1	65.57%	5
KLSvZ 2008	8	40	65.24%	3
our formulation	8	2	54.70%	3

Benchmark I

- 8 instances
- 300–400 edges
- up to 10 terminal sets
- Standard computer, CPLEX, 3GB of RAM

Conclusion

Some Observations

- Formulations theoretically incomparable / no clear domination
- Unclear when to use which formulation

Further research

- Strengthening the formulation?
- Are all constraints needed in [MR2005]?
- Separation algorithm for [MR2005]?
- Other lower bounds for Steiner Forest?

Conclusion

Some Observations

- Formulations theoretically incomparable / no clear domination
- Unclear when to use which formulation

Further research

- Strengthening the formulation?
- Are all constraints needed in [MR2005]?
- Separation algorithm for [MR2005]?
- Other lower bounds for Steiner Forest?

— The End —