A Local-Search Algorithm for Steiner Forest

M. Groß, A. Gupta, A. Kumar, J. Matuschke, D. Schmidt, J. Verschae
M. Schmidt

Highlights of Algorithms 2017, Berlin

June 10th 2017

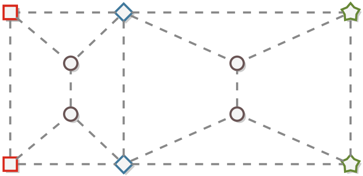
The Steiner Forest Problem

Input

Graph	$G = (V, E)$
Terminal pairs	$(s_1, \bar{s}_1), \ldots, (s_k, \bar{s}_k) \in V \times V$
Edge costs	$c : E \to \mathbb{R}^+$

Output

Minimum cost forest $F \subseteq E$ containing s_i-$\bar{s}_i$-path for all $i = 1, \ldots, k$



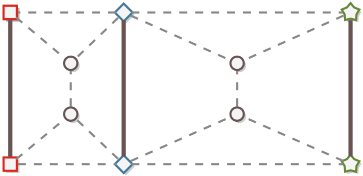
The Steiner Forest Problem

Input

Graph	$G = (V, E)$
Terminal pairs	$(s_1, \bar{s}_1), \ldots, (s_k, \bar{s}_k) \in V \times V$
Edge costs	$c : E \to \mathbb{R}^+$

Output

Minimum cost forest $F \subseteq E$ containing s_i-$\bar{s}_i$-path for all $i = 1, \ldots, k$



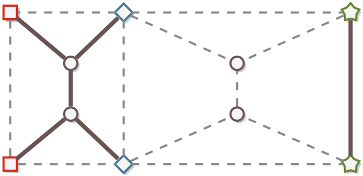
The Steiner Forest Problem

Input

Graph	$G = (V, E)$
Terminal pairs	$(s_1, \bar{s}_1), \ldots, (s_k, \bar{s}_k) \in V \times V$
Edge costs	$c : E \to \mathbb{R}^+$

Output

Minimum cost forest $F \subseteq E$ containing s_i-$\bar{s}_i$-path for all $i = 1, \ldots, k$



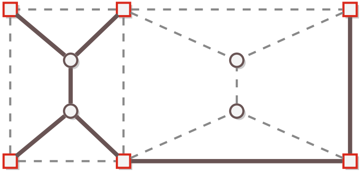
The Steiner Tree Problem

Input

Graph	$G = (V, E)$
Terminals	$s_1, \ldots, s_k \in V$
Edge costs	$c : E \to \mathbb{R}^+$

Output

Minimum cost tree $T \subseteq E$ containing all s_i



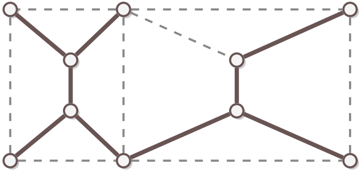
The Minimum Spanning Tree Problem

Input

Graph	$G = (V, E)$
Terminals	V
Edge costs	$c : E \to \mathbb{R}^+$

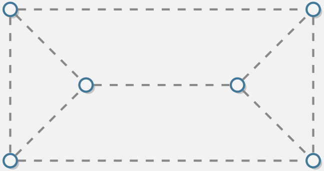
Output

Minimum cost tree $T \subseteq E$ containing all v



Local Search

Example: Local search for MST with euclidean distances

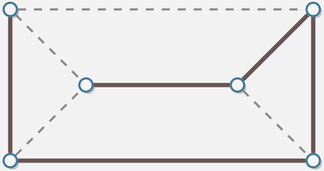


Local search algorithm.

1. Start from arbitrary feasible solution.
2. Reach next feasible solution by executing single edge swaps.
3. Iterate until no improving swap $\rightsquigarrow$ Local optimum reached.

Local Search

Example: Local search for MST with euclidean distances

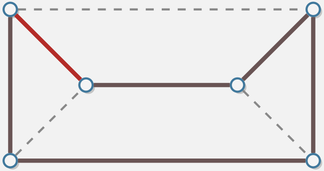


Local search algorithm.

1. Start from arbitrary feasible solution.
2. Reach next feasible solution by executing single edge swaps.
3. Iterate until no improving swap $\rightsquigarrow$ Local optimum reached.

Local Search

Example: Local search for MST with euclidean distances

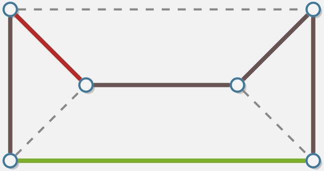


Local search algorithm.

1. Start from arbitrary feasible solution.
2. Reach next feasible solution by executing single edge swaps.
3. Iterate until no improving swap $\rightsquigarrow$ Local optimum reached.

Local Search

Example: Local search for MST with euclidean distances

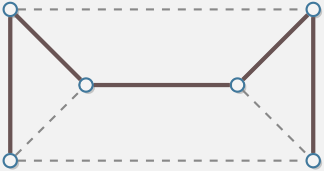


Local search algorithm.

1. Start from arbitrary feasible solution.
2. Reach next feasible solution by executing single edge swaps.
3. Iterate until no improving swap $\rightsquigarrow$ Local optimum reached.

Local Search

Example: Local search for MST with euclidean distances

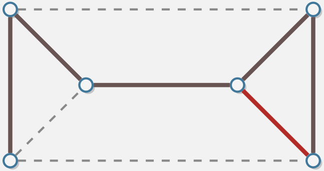


Local search algorithm.

1. Start from arbitrary feasible solution.
2. Reach next feasible solution by executing single edge swaps.
3. Iterate until no improving swap $\leadsto$ Local optimum reached.

Local Search

Example: Local search for MST with euclidean distances

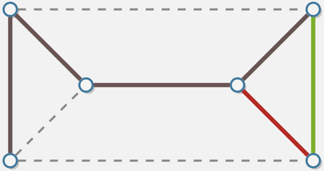


Local search algorithm.

1. Start from arbitrary feasible solution.
2. Reach next feasible solution by executing single edge swaps.
3. Iterate until no improving swap $\rightsquigarrow$ Local optimum reached.

Local Search

Example: Local search for MST with euclidean distances

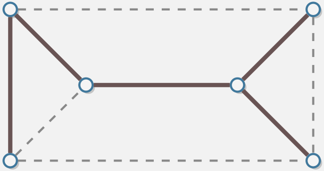


Local search algorithm.

1. Start from arbitrary feasible solution.
2. Reach next feasible solution by executing single edge swaps.
3. Iterate until no improving swap $\rightsquigarrow$ Local optimum reached.

Local Search

Example: Local search for MST with euclidean distances

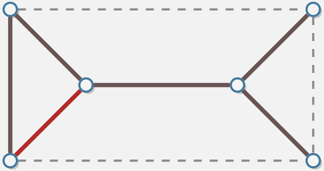


Local search algorithm.

1. Start from arbitrary feasible solution.
2. Reach next feasible solution by executing single edge swaps.
3. Iterate until no improving swap $\rightsquigarrow$ Local optimum reached.

Local Search

Example: Local search for MST with euclidean distances

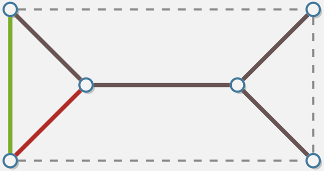


Local search algorithm.

1. Start from arbitrary feasible solution.
2. Reach next feasible solution by executing single edge swaps.
3. Iterate until no improving swap $\rightsquigarrow$ Local optimum reached.

Local Search

Example: Local search for MST with euclidean distances

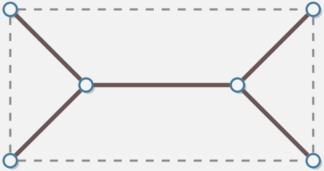


Local search algorithm.

1. Start from arbitrary feasible solution.
2. Reach next feasible solution by executing single edge swaps.
3. Iterate until no improving swap $\rightsquigarrow$ Local optimum reached.

Local Search

Example: Local search for MST with euclidean distances

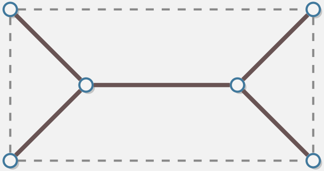


Local search algorithm.

1. Start from arbitrary feasible solution.
2. Reach next feasible solution by executing single edge swaps.
3. Iterate until no improving swap $\rightsquigarrow$ Local optimum reached.

Local Search

Example: Local search for MST with euclidean distances



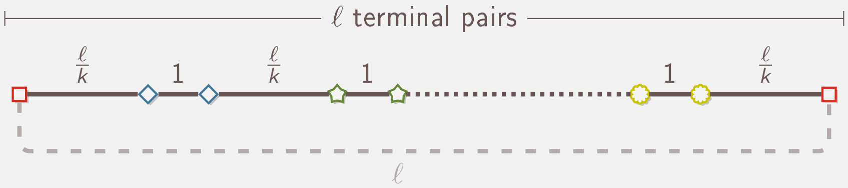
Local search algorithm.

1. Start from arbitrary feasible solution.
2. Reach next feasible solution by executing single edge swaps.
3. Iterate until no improving swap $\rightsquigarrow$ Local optimum reached.

- For MST, this is optimal!
- $\rightsquigarrow$ 2-approximation for Steiner Tree

Local Search for Steiner Forest: Simple moves don't work

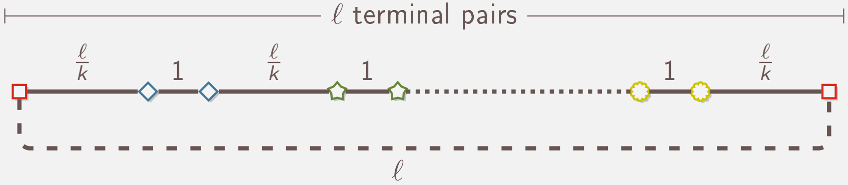
Choose ℓ and k with $\ell \gg k$



Need to remove more than $O(1)$ edges.

Local Search for Steiner Forest: Simple moves don't work

Choose ℓ and k with $\ell \gg k$

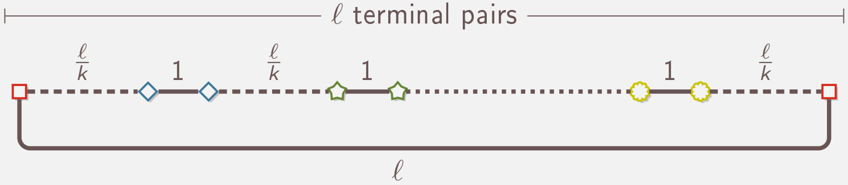


Need to remove more than $O(1)$ edges.

- Insert edge of cost ℓ
- Unless more than k edges are removed, no improving move.
- Local OPT $> \ell^2/k$ vs. global OPT 2ℓ.

Local Search for Steiner Forest: Simple moves don't work

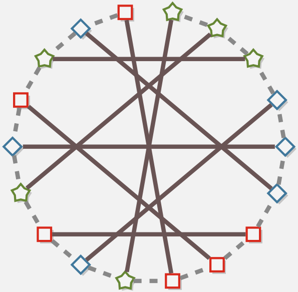
Choose ℓ and k with $\ell \gg k$



Need to remove more than $O(1)$ edges.

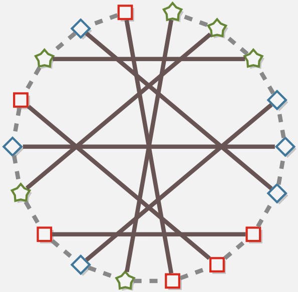
- Insert edge of cost ℓ
- Unless more than k edges are removed, no improving move.
- Local OPT $> \ell^2/k$ vs. global OPT 2ℓ.

Local Search for Steiner Forest: Simple moves don't work



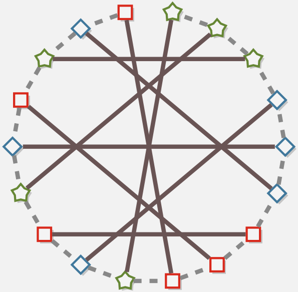
- Solid edges cost 4, dashed edges cost 1.

Local Search for Steiner Forest: Simple moves don't work



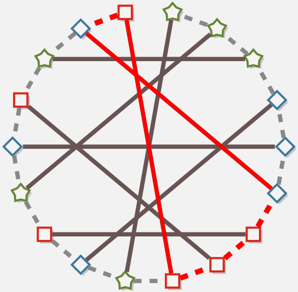
- Solid edges cost 4, dashed edges cost 1.
- Locally optimal for many simple swaps.

Local Search for Steiner Forest: Simple moves don't work



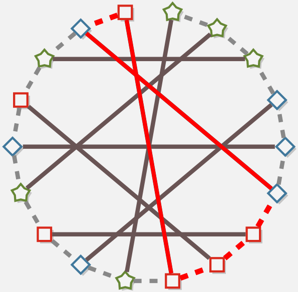
- Solid edges cost 4, dashed edges cost 1.
- Locally optimal for many simple swaps.
- More general version of this example: $\Omega(\log n)$ approximation factor.

Local Search for Steiner Forest: Simple moves don't work



- Solid edges cost 4, dashed edges cost 1.
- Locally optimal for many simple swaps.
- More general version of this example: $\Omega(\log n)$ approximation factor.
- Reason lies in high girth.

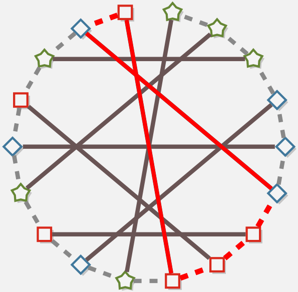
Local Search for Steiner Forest: Simple moves don't work



- Solid edges cost 4, dashed edges cost 1.
- Locally optimal for many simple swaps.
- More general version of this example: $\Omega(\log n)$ approximation factor.
- Reason lies in high girth.

Solution: Complex puzzle with a potential function and involved local steps.

Local Search for Steiner Forest: Simple moves don't work



- Solid edges cost 4, dashed edges cost 1.
- Locally optimal for many simple swaps.
- More general version of this example: $\Omega(\log n)$ approximation factor.
- Reason lies in high girth.

Solution: Complex puzzle with a potential function and involved local steps.

Theorem

There is a non-oblivious local search algorithm for the Steiner Forest Problem with a constant locality gap.