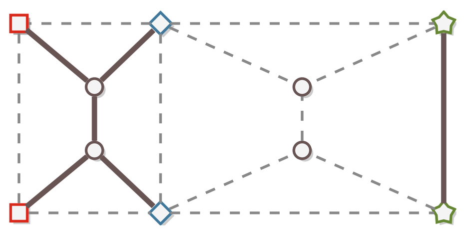
A Local-Search Algorithm for Steiner Forest

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

The Steiner Forest Problem



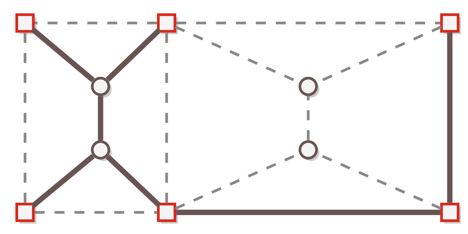
Input

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

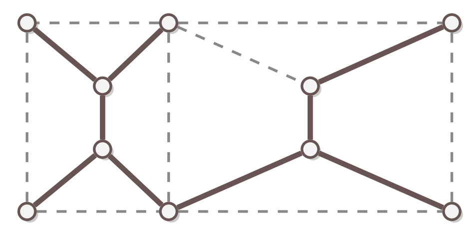
Minimum cost forest $F \subseteq E$ containing an s_i-$\bar{s}_i$-path for all $i = 1, \ldots, k$

The Steiner Tree Problem



No pairs, connect set of terminals by a tree.

The MST Problem



No terminals, connect all nodes in V.

Known algorithms for Steiner Forest:

- 2-approximable, [AKR95, GW95], LP-based
- Non-LP-based 96-approximation [GK2014], greedy (gluttonous) algorithm

Want: Local Search Algorithm for Steiner Forest (inspired by simple algorithm for MST!)
Some simplifying assumptions:

- can assume that c is metric
- ignore non-terminal nodes $\rightsquigarrow$ factor 2

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

Width of a (sub)tree T:

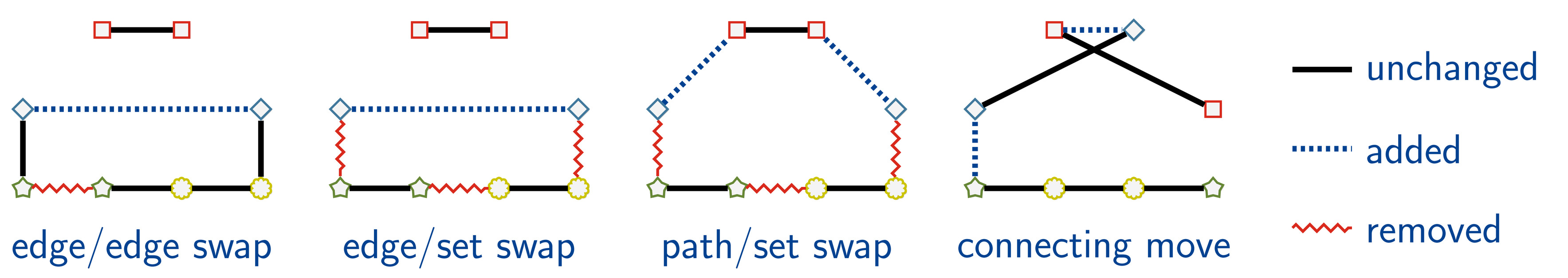
$$w(T) = \max_{\text{terminal pair } s_i, \bar{s}_i \text{ in } T} c(\{s_i, \bar{s}_i\})$$

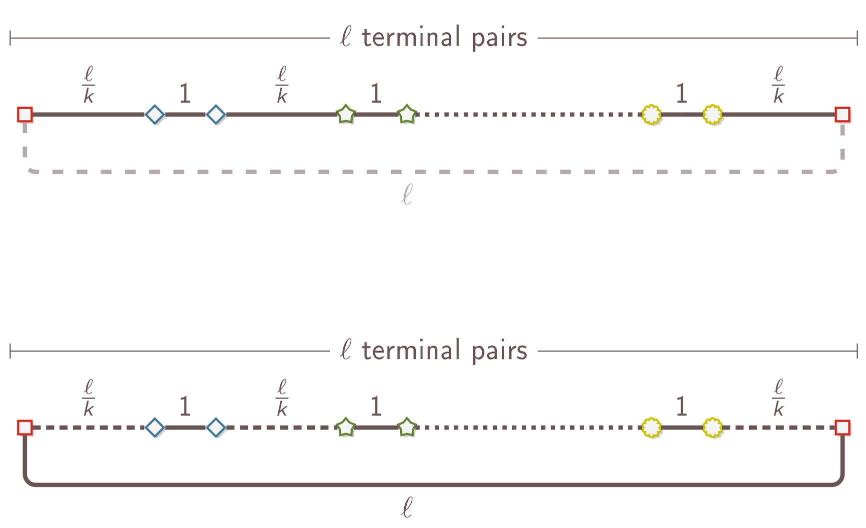
Our potential adds the width:

$$\phi(T) := w(T) + \sum_{e \in E(T)} c(e)$$

Forest: Add $\phi(T)$ for all subtrees T.

- Optimize $\phi \rightsquigarrow$ non-oblivious local search
- Observe: At most two times the connection cost
- After optimization, drop 'useless' edges (called inessential)

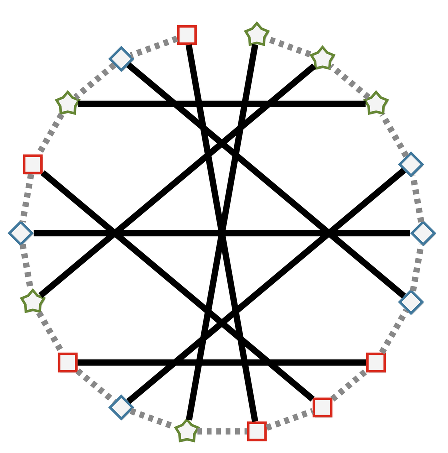




Assume that we only use single edge swaps or swaps that add/delete at most a constant number of edges.
Consider the example on the left:

- choose $k \in \omega(1)$ and $\ell \in \omega(k)$
- local OPT costs $\ell^2/k + \ell - 1 > \ell^2/k$
- OPT costs $\ell + \ell - 1 < 2\ell$
- factor $> \ell/(2k) \rightsquigarrow$ is in $\omega(1)$

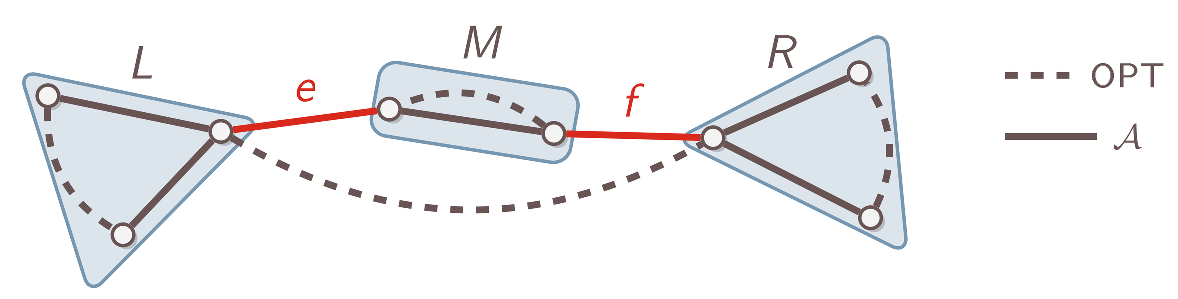
Assume that we only use path/set swaps.



Solid edges cost 4, dashed edges cost 1. No helpful path/set swap.

- $G = (V, E)$: a 3-regular graph with girth $g = c \log n$. (Such graphs exist, see [Biggs 1998].)
- OPT: A spanning tree T in G. Always feasible! All edges in $E(T)$ cost $1 \rightsquigarrow$ OPT costs $n - 1$.
- E': The non-tree edges $E \backslash E(T)$. These edges cost $g/4$.
- M: Any maximum matching in E'. Endpoints of each edge form a terminal pair.
- Observe: M is feasible. Degree bound ensures that $|M| \geq n/10$. Thus, M costs $\Omega(n \log n)$.

Two edges e and f are compatible wrt OPT if there are no OPT-edges between L and M and no OPT-edges between M and R.



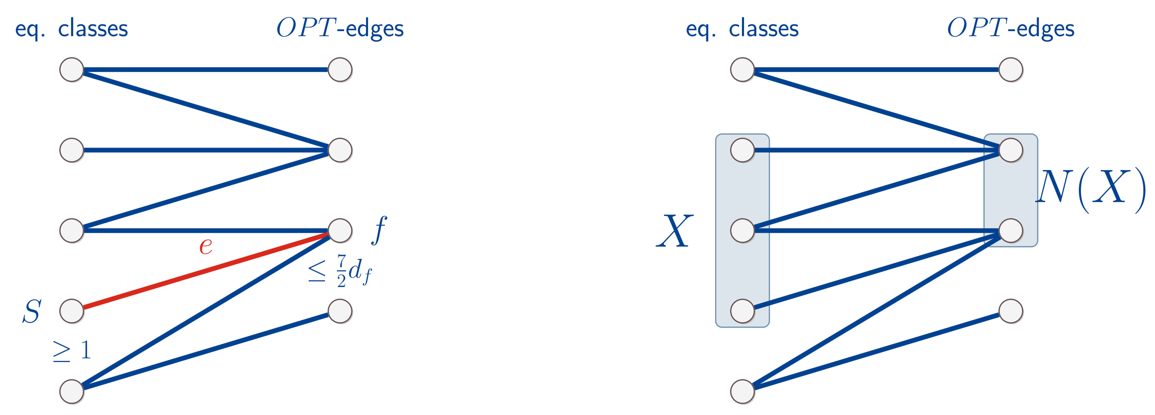
- compatibility is an equivalence relation
- equivalence classes lie on paths, behave 'like one edges'

Charging a locally opt tree $\mathcal{A}$ to OPT

- Find assignment of $\mathcal{A}$-edges to OPT-edges such that no OPT-edge is assigned more than $\frac{7}{2}$ its cost.

$$\sum_{e \in E[\mathcal{A}]} c(e) \leq \frac{7}{2} \sum_{e \in OPT} c(e)$$

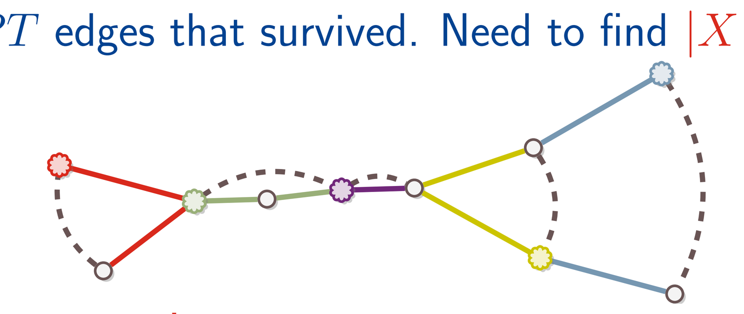
- Assignment: Capacitated perfect matching in a bipartite graph.



Left: node a_S for each equiv. class S
Right: node b_f for each $f \in$OPT

Show: $|X| \leq \frac{7}{2}|N(X)|$ for all X
Then: Hall's Theorem yields assignment

- Fix some set X. Contract all equivalence classes outside of X!
- Note: All OPT edges that survive are in $N(X)$.
- Count all OPT edges that survived. Need to find $|X|/c$ OPT edges.



- Concept of representatives
- Leaves always have an OPT-edge: Otherwise, inessential edge!
- Representatives of degree 2 have an OPT edge
- There are not too many nodes of higher degree!

These arguments give guarantee 4, but we can do slightly better.