Parameterized Approximation Schemes for Steiner Trees with Small Number of Steiner Vertices

Pavel Dvořák[1], Andreas Emil Feldmann[1], Dušan Knop[1,2], Tomáš Masařík[1], Tomáš Toufar[1], Pavel Veselý[1]

[1]Charles University, Prague, Czech Republic
[2]University of Bergen, Bergen, Norway

Steiner Tree

Let $G = (V, E)$ be a graph and $R \subseteq V$ be a set of *terminals* (vertices in $V \setminus R$ are called *Steiner* vertices). A *Steiner tree* $T = (W, F)$ is a tree such that $R \subseteq W \subseteq V$ and $F \subseteq E$.

Problem Steiner Tree

Instance: Graph $G = (V, E)$, $R \subseteq V$, edge weights $w : E \to \mathbb{R}^+_0$.
Goal: Find a minimum Steiner tree according to w.

Parameters

- $|R|$ – number of terminals – subject of previous works.
- p – number of Steiner vertices in the optimal solution – studied in this work.

Previous Results

Parameterized Complexity

- FPT algorithm for the parameter $|R|$ – time $(2+\gamma)^{|R|} \cdot poly(n)$ for any fixed $\gamma > 0$ [Mölle et al. '06].
- $(1+\varepsilon)$-approximate kernel of size $|R|^k$ for $k = 2^{\mathcal{O}(1/\varepsilon)}$ [Lokshtanov et al. '16].
- W[2]-hard for the parameter p [folklore].

Approximation

- 96/95-approximation is NP-hard [Chlebík and Chlebíková '02].
- 1.39-approximation algorithm [Byrka et al. '13].

Our Results

1. Efficient parameterized approximation scheme – it returns $(1+\varepsilon)$-approximation in time $f(p, \varepsilon) \cdot poly(n)$.
2. Polynomial size approximate kernelization scheme.

	Unweighted		Weighted	
Undirected	✓	✓	✓	✓
Directed	✓	✗*	✗**	✗**

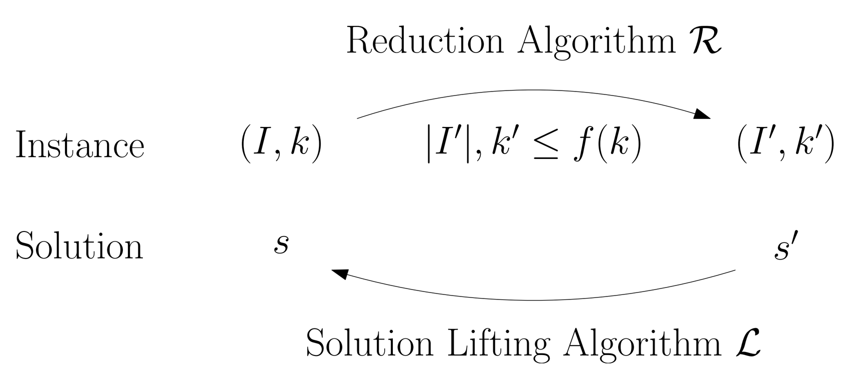
* Unless NP ⊆ coNP/poly.
** Unless FPT= W[2].

Main Idea

Reduce the number of terminal under some bound $f(p, \varepsilon)$ and use the algorithms or kernels for the parameter $|R|$.

α-approximate kernel

- An α-approximate kernel is a pair of algorithms $\mathcal{R}$ and $\mathcal{L}$ [Lokshtanov et al. '16]:



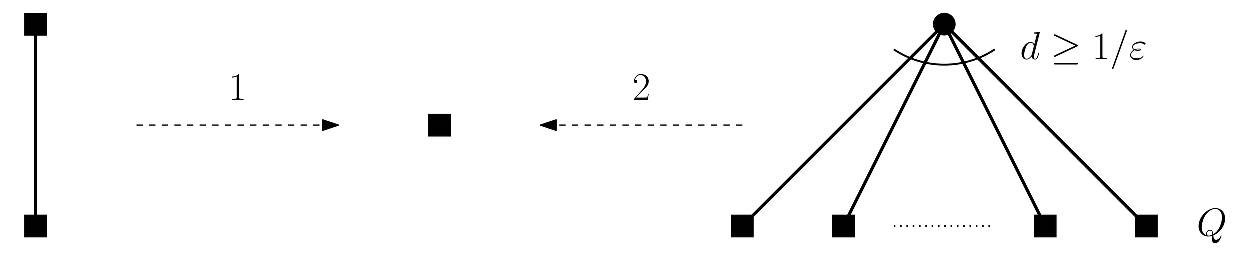
- If s' is a β-approximation of (I', k'), then s is an $(\alpha \cdot \beta)$-approximation of (I, k).
- If f is a polynomial, then $(\mathcal{R}, \mathcal{L})$ is a polynomial size α-approximate kernel.
- Polynomial size approximate kernelization scheme – for every $\varepsilon > 0$ there exists a polynomial size $(1+\varepsilon)$-approximate kernel.
  – The exponent in the polynomial can depend on ε.

Funding: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 616787. This work was partially supported by the projects SVV–2017–260452, 17-09142S of GAČR, Czech Science Foundation GAČR (grant #17-10090Y), GAUK 1514217.

arXiv version:

Unweighted Cases

Two Reduction Rules for Undirected Graphs



Reduction Rule 1: We can assume any such edge is in the optimal solution.

Reduction Rule 2: The optimal solution uses at least d edges to connect terminals in Q. Our solution uses at most $d+1$ edges.

$$\frac{\mathsf{ALG}}{\mathsf{OPT}} = \frac{d+1}{d} = 1 + \frac{1}{d} \le 1 + \varepsilon.$$

- If we cannot apply any of these two rules to a graph G, then there are at most p/ε terminals in G.
- Reduction rules for the directed case is similar.

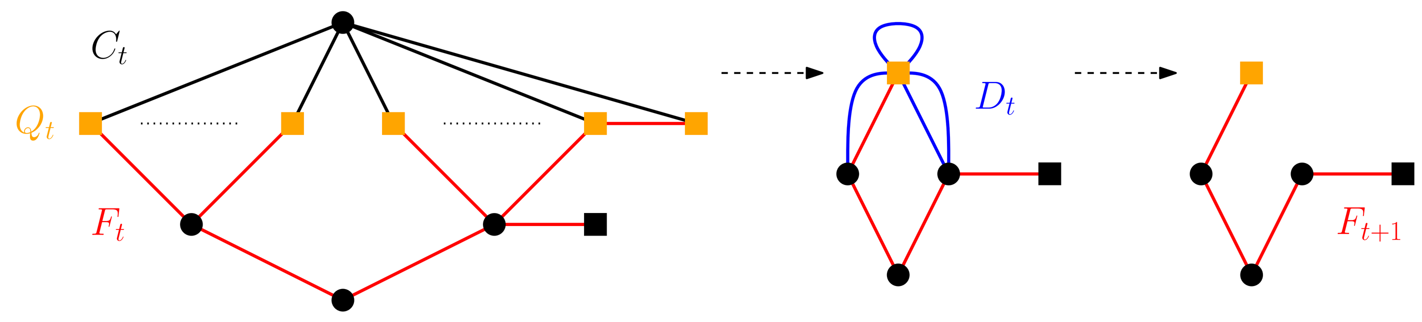
Weighted Undirected Case

Reduction Rule

1. Let C be a subgraph of G, such that C is a star and Q are terminals in C.
2. A *ratio* of C is $r(C) = \frac{w(C)}{|Q|-1}$ where $w(C)$ is a weight of all edges in C.
3. We find a star of the smallest ratio and contract it.

We apply the reduction rule until there is less than $\tau = f_1(p, \varepsilon)$ terminals.

- $G_0 = G, G_1, G_2, \dots$
  – G_{t+1} arises from G_t by contracting a star C_t of the smallest ratio.
- Q_t – terminals in C_t.
- F_0 – the optimum of G_0.
- F_{t+1} – a Steiner tree of G_{t+1} which arises from F_t when we identify all terminals in Q_t into a new terminal.
- $D_t = E(F_{t+1}) \setminus E(F_t)$ – deleted edges in F_t to get F_{t+1}.



Definition. A contraction of star C_t is *good* if $w(C_t) \le (1+\varepsilon)w(D_t)$, otherwise C_t is *bad*.

How to Handle Bad Contractions

1. If $|Q_t| \ge \lambda = f_2(p, \varepsilon)$ then the contraction of C_t is good.
2. Let $L_i = \big((1+\delta)^i, (1+\delta)^{i+1}\big]$, for $\delta : (1+\delta)^2 = (1+\varepsilon)$.
   - i-contraction: contraction of a star C if $r(C) \in L_i$.
3. For all i the number of bad i-contraction is smaller than $\kappa = f_3(p, \varepsilon)$.
4. Let j be the largest number such that some j-contraction occurs during the algorithm.
5. Let W_B be a weight of all bad contracted stars, then
   - $W_B \le \kappa \cdot \lambda \cdot \frac{(1+\delta)^{j+2}}{\delta}$.
   - $w(F_0) \ge (1+\delta)^j \cdot (\tau - 2p)$.
   - For $\tau = \mathcal{O}(p^2/\varepsilon^4)$ large enough, $W_B \le \varepsilon \cdot w(F_0)$.

$(1+2\varepsilon)$-approximation

1. We run the exact algorithm for the parameter $|R|$ on the reduced graph G_t.
   - Time $(2+\gamma)^{\mathcal{O}(p^2/\varepsilon^4)} \cdot poly(n)$.
   - We get Steiner tree F'_t of the graph G_t.
2. Add all edges from the contracted stars – Steiner tree F of the original graph G.
3. Weight of all bad contracted stars is at most $\varepsilon \cdot \mathsf{OPT}$.
4. Weight of F'_t and all good contracted stars is at most $(1+\varepsilon) \cdot \mathsf{OPT}$.

$$w(F) \le (1+2\varepsilon) \cdot \mathsf{OPT}$$

If we run a β-approximation on the reduced graph G_t we get $(1+\varepsilon) \cdot \beta$-approximation.

An algorithm based on the star contractions performs quite well in practice.

LaTeX TikZposter