The Complexity Landscape of Fixed-Parameter Directed Steiner Network Problems

Dániel Marx

Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI) Budapest, Hungary

(Joint work with Andreas Feldmann)

SIWAG 2016
Polignano a Mare
September 26, 2016
Steiner Tree

Given an edge-weighted graph $G$ and set $T \subseteq V(G)$ of terminals, find a minimum-weight tree in $G$ containing every vertex of $T$. 
Steiner Tree

Some known results:

- NP-hard
- Easy 2-approximation: use a minimum spanning tree.
- 1.386-approximation [Byrka et al. 2013].
- $3^k \cdot n^{O(1)}$ time algorithm for $k$ terminals using dynamic programming (i.e., fixed-parameter tractable parameterized by the number of terminals)
- Can be improved to $2^k \cdot n^{O(1)}$ time using fast subset convolution [Björklund et al. 2006].
**Steiner Forest**

Given an edge-weighted graph \( G \) and a list \((s_1, t_1), \ldots, (s_k, t_k)\) of pairs of terminals, find a minimum-weight forest in \( G \) that connects \( s_i \) and \( t_i \) for every \( 1 \leq i \leq k \).

Fixed-parameter tractable parameterized by \( k \): Guess a partition of the \( 2k \) terminals \( (k^O(k) = 2^O(k \log k)) \) possibilities) and solve a **Steiner Tree** for each class of the partition.
Variants of **Steiner Tree**

**Steiner Tree**

Connect all the terminals

**Steiner Forest**

Create connections satisfying every request
Variants of **Steiner Tree**

**Steiner Tree**
- Connect all the terminals

**Steiner Forest**
- Create connections satisfying every request

**Steiner Tree**
- Make every terminal reachable from the root

**Strongly Connected Steiner Subgraph (SCSS)**
- Make all the terminals reachable from each other

**Directed Steiner Network (DSN)**
- Create connections satisfying every request
Directed Steiner vs. SCSS

The DP for Steiner Tree generalizes to the directed version:

Directed Steiner Tree with $k$ terminals can be solved in time $2^k \cdot n^{O(1)}$. 
**Directed Steiner vs. SCSS**

The DP for **Steiner Tree** generalizes to the directed version:

**Directed Steiner Tree** with $k$ terminals can be solved in time $2^k \cdot n^{O(1)}$.

SCSS seems to be much harder:

**Theorem** [Feldman and Ruhl 2006]

**Strongly Connected Steiner Subgraph** with $k$ terminals can be solved in time $n^{O(k)}$.

**Theorem** [Chitnis, Hajiaghayi, and M. 2014]

Assuming ETH, **Strongly Connected Steiner Subgraph** is W[1]-hard and has no $f(k)n^{o(k/\log k)}$ time algorithm for any function $f$. 
**Directed Steiner Network**

**Theorem** [Feldman and Ruhl 2006]

Directed Steiner Network with $k$ requests can be solved in time $n^{O(k)}$.

**Corollary:** Strongly Connected Steiner Subgraph with $k$ terminals can be solved in time $n^{O(k)}$.

Proof is based on a “pebble game”: $O(k)$ pebbles need to reach their destinations using certain allowed moves, tracing the solution.
A new combinatorial result:

**Theorem [Feldmann and M. 2016]**

[The underlying undirected graph of] every minimum cost solution of *Directed Steiner Network* with $k$ requests has cutwidth and treewidth $O(k)$.  

A new algorithmic result:

**Theorem [Feldmann and M. 2016]**

If a *Directed Steiner Network* instance with $k$ requests has a minimum cost solution with treewidth $w$ of the underlying undirected graph, then it can be solved in time $f(k, w) \cdot n^{O(w)}$.

**Corollary:** A new proof that DSN and SCSS can be solved in time $f(k) n^{O(k)}$. 
Directed Steiner Network

A new combinatorial result:

**Theorem [Feldmann and M. 2016]**

[The underlying undirected graph of] every minimum cost solution of Directed Steiner Network with $k$ requests has cutwidth and treewidth $O(k)$.

A new algorithmic result:

**Theorem [Feldmann and M. 2016]**

If a Directed Steiner Network instance with $k$ requests has a minimum cost solution with treewidth $w$ [of the underlying undirected graph], then it can be solved in time $f(k, w) \cdot n^{O(w)}$.

**Corollary:** A new proof that DSN and SCSS can be solved in time $f(k)n^{O(k)}$. 

Special cases of Directed Steiner Network

Directed Steiner Tree and Strongly Connected Steiner Subgraph are both restrictions of Directed Steiner Network to certain type of patterns:

Goal: characterize the patterns that give rise to FPT/W[1]-hard problems.
Patterns for Directed Steiner Network

Question:
What is the complexity of Directed Steiner Network for this pattern?
Patterns for Directed Steiner Network

Question:
What is the complexity of Directed Steiner Network for this pattern?

Answer:
Directed Steiner Network has an $n^{O(k)}$ algorithm for $k$ requests, so it is polynomial-time solvable for every fixed pattern.
Patterns for Directed Steiner Network

Goal: For every class of $\mathcal{H}$ of directed patterns, characterize the complexity of Directed Steiner Network when restricted to demand patterns from $\mathcal{H}$.

Example:
- If $\mathcal{H}$ is the class of all directed in-stars (or out-stars), then $\mathcal{H}$-DSN is FPT.
- If $\mathcal{H}$ is the class of all directed cycles, then $\mathcal{H}$-DSN is W[1]-hard.
Patterns for **Directed Steiner Network**

**Goal:** For every class of $\mathcal{H}$ of directed patterns, characterize the complexity of **Directed Steiner Network** when restricted to demand patterns from $\mathcal{H}$.

**Example:**
- If $\mathcal{H}$ is the class of all directed in-stars (or out-stars), then $\mathcal{H}$-DSN is FPT.
- If $\mathcal{H}$ is the class of all directed cycles, then $\mathcal{H}$-DSN is W[1]-hard.

**Main result:**

**Theorem [Feldmann and M. 2016]**

For any class $\mathcal{H}$ of directed patterns,
- if $\mathcal{H}$ has combinatorial property X, then $\mathcal{H}$-DSN and $\mathcal{H}$-DSN is W[1]-hard otherwise.
FPT special cases

What classes $\mathcal{H}$ give FPT cases of $\mathcal{H}$-DSN?

We know that out-stars are FPT.
What classes $\mathcal{H}$ give FPT cases of $\mathcal{H}$-DSN?

This is also FPT: minimal solutions have bounded treewidth.
FPT special cases

What classes $\mathcal{H}$ give FPT cases of $\mathcal{H}$-DSN?

This is also FPT: minimal solutions have bounded treewidth.
FPT special cases

What classes $\mathcal{H}$ give FPT cases of $\mathcal{H}$-DSN?

$C_\lambda$: in- or out-caterpillar of length $\lambda$.

Lemma

If the pattern is in $C_\lambda$, then every minimal solution has treewidth $O(\lambda^2)$. 
FPT special cases

What classes $\mathcal{H}$ give FPT cases of $\mathcal{H}$-DSN?

What about this pattern?
FPT special cases

What classes $\mathcal{H}$ give FPT cases of $\mathcal{H}$-DSN?

Lemma

If the pattern is transitivey equivalent to a member of $\mathcal{C}_\lambda$, then every minimal solution has treewidth $O(\lambda^2)$. 
FPT special cases

What classes $\mathcal{H}$ give FPT cases of $\mathcal{H}$-DSN?

$C_{\lambda,\delta}$: in- or out-caterpillar of length $\lambda$ with $\delta$ additional edges.

**Lemma**

If the pattern is transitively equivalent to a member of $C_{\lambda,\delta}$, then every minimal solution has treewidth $O((1 + \lambda)(\lambda + \delta))$. 
FPT special cases

Theorem

If every $H \in \mathcal{H}$ is **transitively equivalent** to a member of $\mathcal{C}_{\lambda,\delta}$ for some constants $\lambda, \delta \geq 0$, then $\mathcal{H}$-DSN is FPT.

Does this cover all the FPT cases?
Theorem

If every $H \in \mathcal{H}$ is transitivity equivalent to a member of $\mathcal{C}_{\lambda,\delta}$ for some constants $\lambda, \delta \geq 0$, then $\mathcal{H}$-DSN is FPT.

Does this cover all the FPT cases?  

(Yes)
W[1]-hard special cases

We show that the following classes $\mathcal{H}$ make $\mathcal{H}$-DSN W[1]-hard:

- cycles (SCSS)
- out-diamonds
- in-diamonds
- flawed out-diamonds
- flawed in-diamonds
Identifying terminals

If $H'$ is obtained from $H$ by identifying terminals, then the problem cannot be harder for $H'$ than for $H$:

$H$ $\Rightarrow$ $H'$

$G$ $\Rightarrow$ $G'$

$\Rightarrow$ We can assume that $\mathcal{H}$ is closed under identifying terminals.
The following combinatorial result connects the algorithmic and the hardness results:

**Theorem**

Let $\mathcal{H}$ be a class of patterns closed under identifying terminals and transitive equivalence. Then exactly one of the following holds:

1. There are constants $\lambda, \delta$ such that every $H \in \mathcal{H}$ is transitively equivalent to a member of $C_{\lambda,\delta}$

2. $\mathcal{H}$ contains either
   - all directed cycles,
   - all in-diamonds,
   - all out-diamonds,
   - all flawed in-diamonds, or
   - all flawed out-diamonds.
Our main result:

Theorem [Feldmann and M. 2016]

Let $\mathcal{H}$ be a class of patterns.

1. If there are constants $\lambda, \delta$ such that every $H \in \mathcal{H}$ is transitively equivalent to a member of $C_{\lambda, \delta}$, then $\mathcal{H}$-DSN is FPT,

2. and it is W[1]-hard otherwise.