# Parameterized Approximations of Directed Steiner Networks

Martin Koreček
Supervisor: doc. Andreas Emil Feldmann, Dr.

2023

# Directed Steiner Network (DSN)

An instance of DSN consists of

- a directed graph $G = (V_G, E_G)$,

# Directed Steiner Network (DSN)

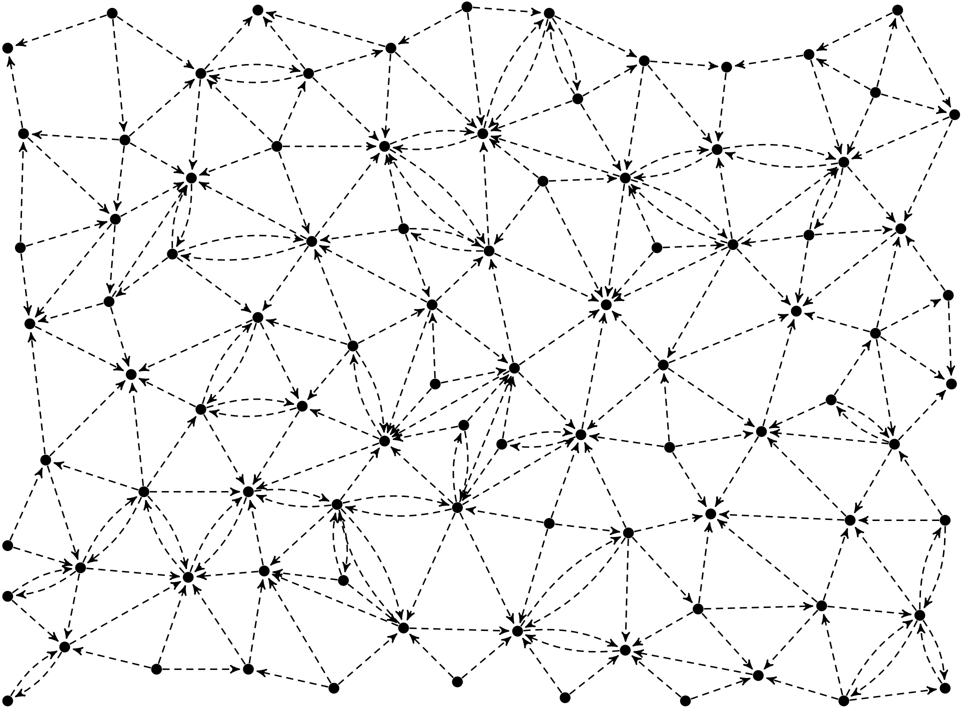
An instance of DSN consists of

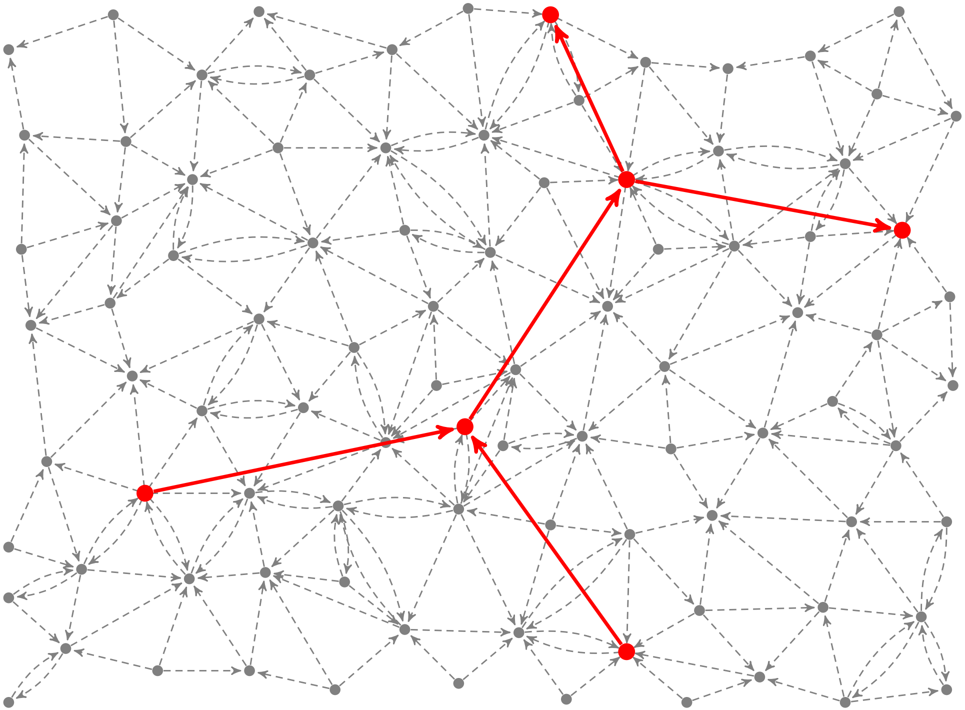
- a directed graph $G = (V_G, E_G)$,
- a non-negative edge-cost function $c : E_G \to \mathbb{R}^+$ and

# Directed Steiner Network (DSN)

An instance of DSN consists of

- a directed graph $G = (V_G, E_G)$,
- a non-negative edge-cost function $c : E_G \to \mathbb{R}^+$ and
- $k$ demand pairs $(s_1, t_1), \ldots, (s_k, t_k) \in V_G^2$, also called terminal pairs.



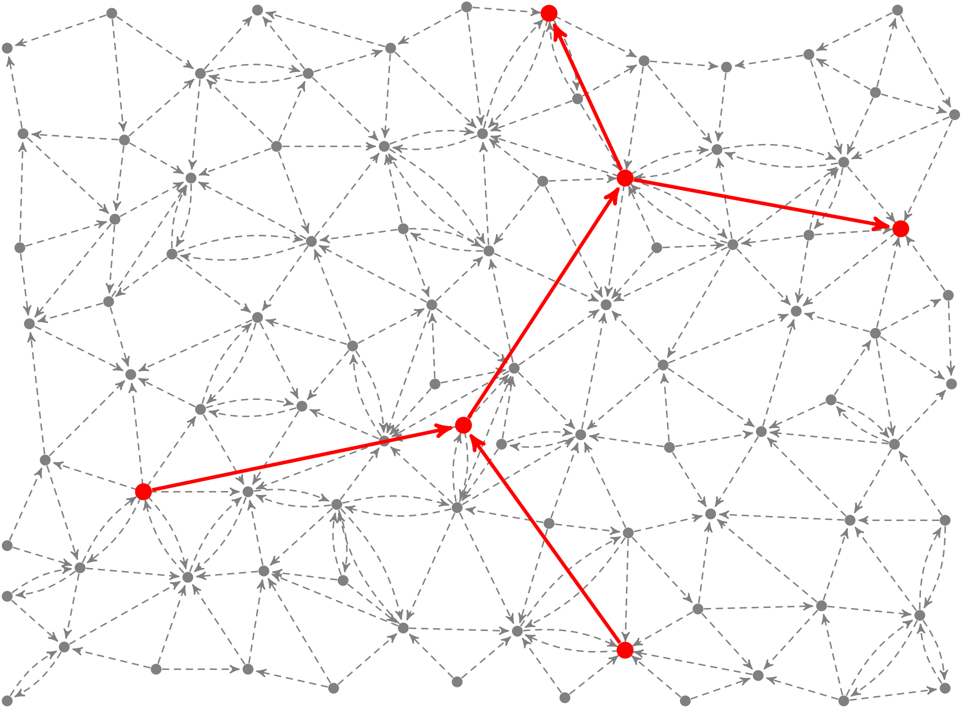


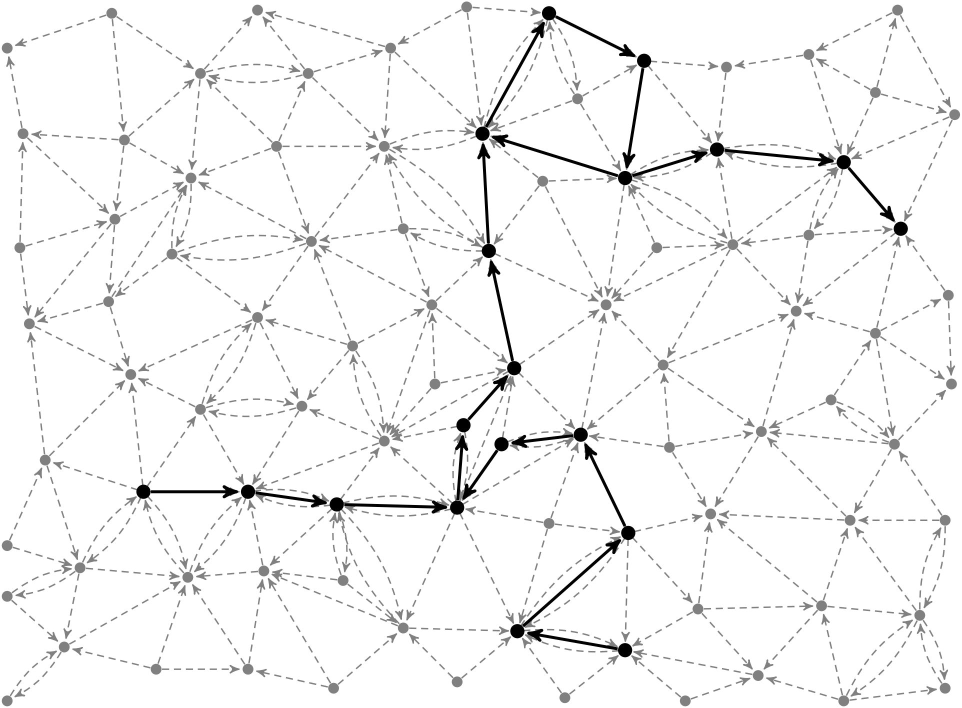
# Directed Steiner Network (DSN)

An instance of DSN consists of

- a directed graph $G = (V_G, E_G)$,
- a non-negative edge-cost function $c : E_G \to \mathbb{R}^+$ and
- $k$ demand pairs $(s_1, t_1), \ldots, (s_k, t_k) \in V_G^2$, also called terminal pairs.

The goal is to compute a minimum-cost subgraph of $G$ containing a directed $s_i \to t_i$ path for each $1 \leq i \leq k$.





# Directed Steiner Network (DSN)

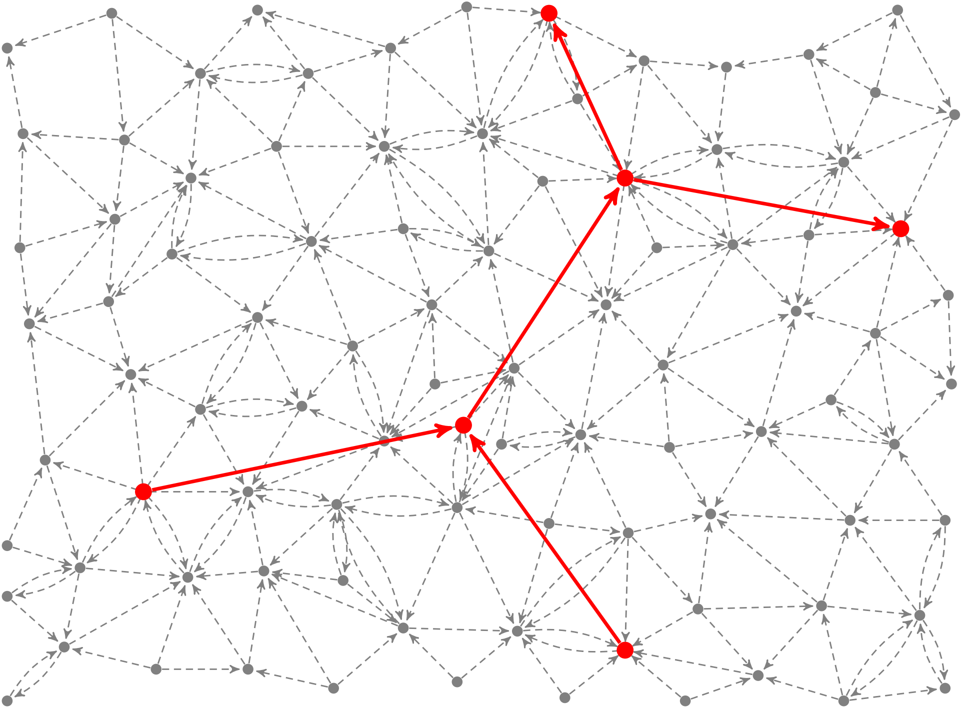
The demand pairs define a "pattern graph" $H = (V_H, E_H)$

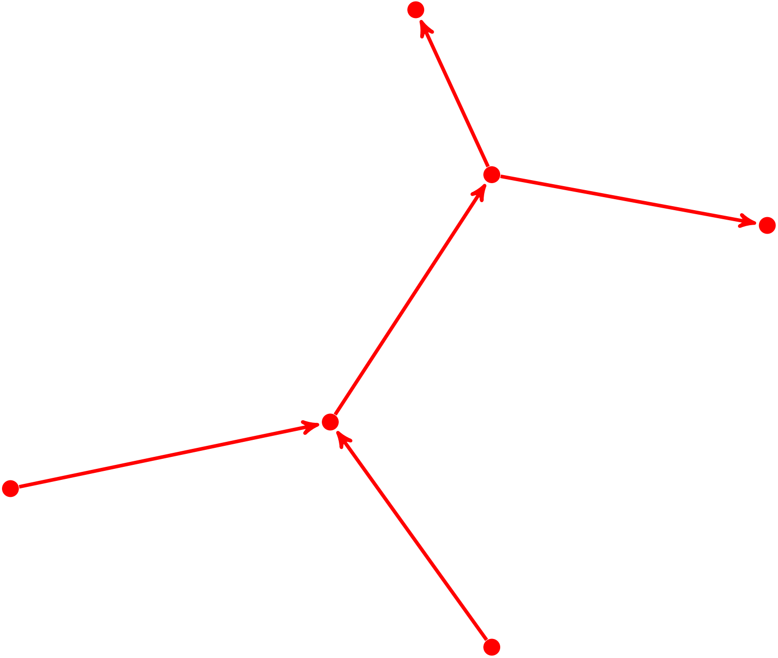
- $V_H = \{s_1, t_1, \ldots, s_k, t_k\}$,
- $E_H = \{s_1 t_1, \ldots, s_k t_k\}$.

# Directed Steiner Network (DSN)

The demand pairs define a “pattern graph” $H = (V_H, E_H)$

- $V_H = \{s_1, t_1, \ldots, s_k, t_k\}$,
- $E_H = \{s_1 t_1, \ldots, s_k t_k\}$.
- This yields an alternative form of the DSN instance, $\Delta = (G, H, c)$.





# Special cases

**Directed Steiner Tree (DST)**

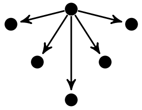
The output network is required to connect one “root node” to $k$ terminals.

Theorem (Karp, 1972)

*DST is NP-complete.*

Theorem (Dreyfus, Wagner, 1971)

*DST admits a* $3^k \cdot n^{\mathcal{O}(1)}$*-time algorithm.*



# Special cases

**Strongly Connected Steiner Subgraph (SCSS)**

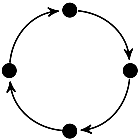
A strongly connected component with $k$ selected terminals is required.

Theorem (Guo et al., 2011)

*SCSS is $W[1]$-hard parameterized by $k$.*

Theorem (Halperin, Krauthgamer, 2003)

*There can be no polynomial time $\mathcal{O}(\log^{2-\epsilon} n)$-approximation algorithm for SCSS (for any $\epsilon > 0$), unless $NP \subseteq ZTIME(n^{polylog(n)})$.*

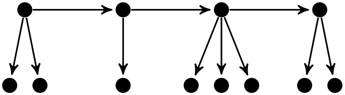


# $\mathcal{H}$-DSN

**Definition**

For a class $\mathcal{H}$ of directed graphs, the $\mathcal{H}$-DSN is the special case of DSN where the pattern graph $H$ is required to be in $\mathcal{H}$.

# $\mathcal{H}$-DSN

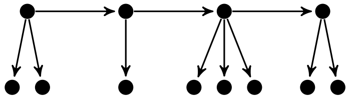


### Theorem (Feldmann, Marx, 2017)

*Given a class of pattern graphs $\mathcal{H}$ with $k$ edges,*

- *either there are constants $\lambda, \delta$ such that $\mathcal{H} \subseteq \mathcal{C}^*_{\lambda,\delta}$, in which case $\mathcal{H}$-DSN can be solved in FPT time $f(k) \cdot n^{\mathcal{O}(1)}$,*
- *or, if there are no such constants $\lambda$ and $\delta$, $\mathcal{H}$-DSN is $W[1]$-hard (for parameter $k$).*

# $\mathcal{H}$-DSN



Theorem (Feldmann, Marx, 2017)

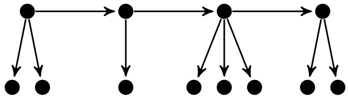
*Given a class of pattern graphs $\mathcal{H}$ with $k$ edges,*

- *either there are constants $\lambda, \delta$ such that $\mathcal{H} \subseteq \mathcal{C}^*_{\lambda,\delta}$, in which case $\mathcal{H}$-DSN can be solved in FPT time $f(k) \cdot n^{\mathcal{O}(1)}$,*
- *or, if there are no such constants $\lambda$ and $\delta$, $\mathcal{H}$-DSN is $W[1]$-hard (for parameter $k$).*

Theorem (This thesis)

*In the second case of the previous theorem (if there are no $\lambda$ and $\delta$ such that $\mathcal{H} \subseteq \mathcal{C}^*_{\lambda,\delta}$), there is no FPT $(\frac{4}{3} - \epsilon)$-approximation algorithm for $\mathcal{H}$-DSN, parameterized by $k$, unless Gap-ETH fails.*

# $\mathcal{H}$-DSN



## Theorem (Feldmann, Marx, 2017)

*Given a class of pattern graphs $\mathcal{H}$ with $k$ edges,*

- *either there are constants $\lambda, \delta$ such that $\mathcal{H} \subseteq \mathcal{C}^*_{\lambda,\delta}$, in which case $\mathcal{H}$-DSN can be solved in FPT time $f(k) \cdot n^{\mathcal{O}(1)}$,*
- *or, if there are no such constants $\lambda$ and $\delta$, $\mathcal{H}$-DSN is $W[1]$-hard (for parameter $k$).*

## Theorem (This thesis)

*In the second case of the previous theorem (if there are no $\lambda$ and $\delta$ such that $\mathcal{H} \subseteq \mathcal{C}^*_{\lambda,\delta}$), there is no FPT $(\frac{4}{3} - \epsilon)$-approximation algorithm for $\mathcal{H}$-DSN, parameterized by $k$, unless Gap-ETH fails.*

## Corollary

No parameterized approximation scheme for the W[1]-hard cases.

# Constant-factor approximation

Denote by Path the class of directed paths.

## Theorem (This thesis)

*There is a $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$-time 2-approximation algorithm for* Path-*DSN.*

*($k$ is the number of edges of the pattern graph)*

High-level idea: Guess what terminal vertices participate in the strongly connected components of an optimal solution.

# Constant-factor approximation

Denote by Path the class of directed paths.

## Theorem (This thesis)

*There is a $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$-time 2-approximation algorithm for* Path-*DSN.*

*($k$ is the number of edges of the pattern graph)*

High-level idea: Guess what terminal vertices participate in the strongly connected components of an optimal solution.

An approximation preserving reduction from SCSS shows this ratio is tight:

## Theorem (Chitnis et al., 2021)

*Assuming Gap-ETH, it is impossible to $(2-\epsilon)$-approximate SCSS for any $\epsilon > 0$ in FPT time, parameterized by $k$.*

# Constant-factor approximation

Let $p$-PATH be the class of directed graphs composed of (at most) $p$ paths.

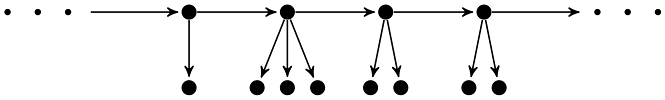
## Theorem (This thesis)

*There is a $k! \cdot 4^k \cdot n^{\mathcal{O}(p)}$-time 2-approximation algorithm for $p$-PATH-DSN.*
*($k$ being the number of edges of the pattern graph)*

Requires guessing more about the structure of an optimum solution.

# Constant-factor approximation

**Long caterpillars**



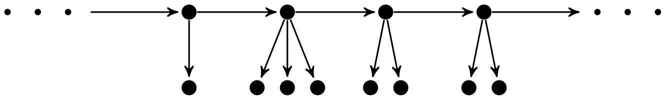
Theorem (This thesis)

*There is an* $8^k \cdot (k!)^2 \cdot n^{\mathcal{O}(1)}$*-time* 3*-approximation algorithm for the* $\textsc{Cat}$*-DSN problem.*

*(k being the number of edges of the pattern graph)*

# Constant-factor approximation

**Long caterpillars**



Theorem (This thesis)

*There is an* $8^k \cdot (k!)^2 \cdot n^{\mathcal{O}(1)}$*-time* 3*-approximation algorithm for the* Cat*-DSN problem.*

*(k being the number of edges of the pattern graph)*

Open question: Can we obtain a better approximation ratio?

# Combining the cases

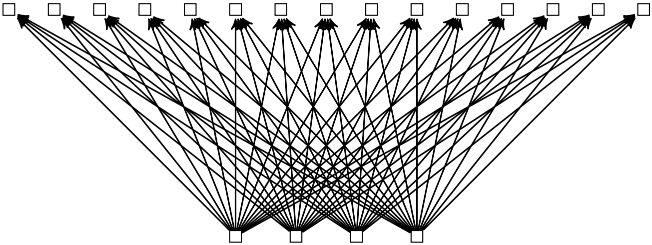
Let $(\chi, \pi)$-$\textsc{Cat}$ denote the class of graphs obtainable as unions of (at most) $\chi$ caterpillars and $\pi$ paths.

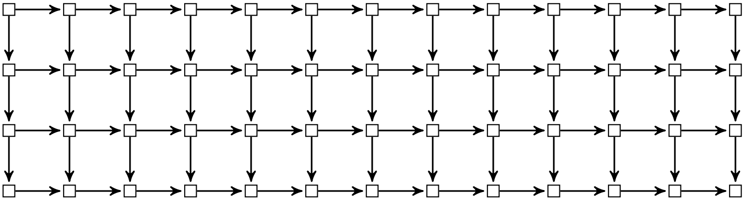
## Theorem (This thesis)

*There is an $f(k) \cdot n^{\mathcal{O}(\chi+\pi)}$-time $(2+\chi)$-approximation algorithm for $(\chi, \pi)$-$\textsc{Cat}$-DSN.*

# Further topics

**Significant unsolved cases**





*Thank you for listening.*