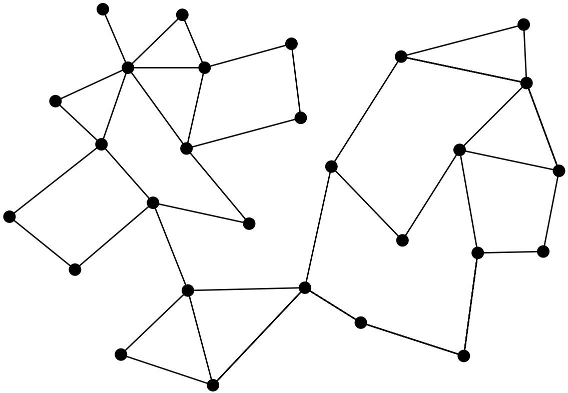
An Improved Analysis of Greedy for Online Steiner Forest

Étienne Bamas, Marina Drygala, Andreas Maggiori

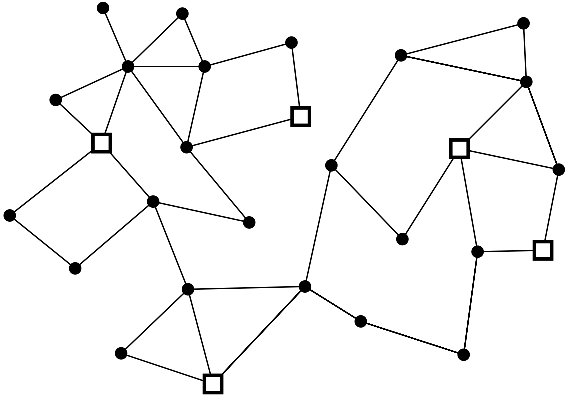
EPFL, Switzerland

ACM-SIAM Symposium on Discrete Algorithms
SIAM 2022

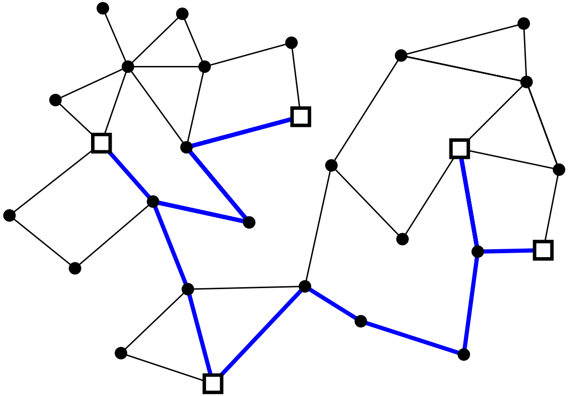
The Online Steiner Tree Problem



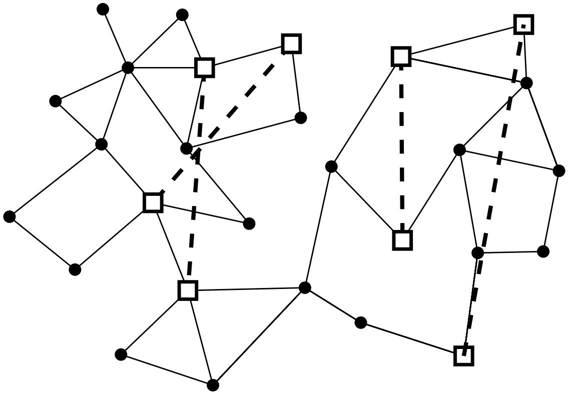
The Online Steiner Tree Problem



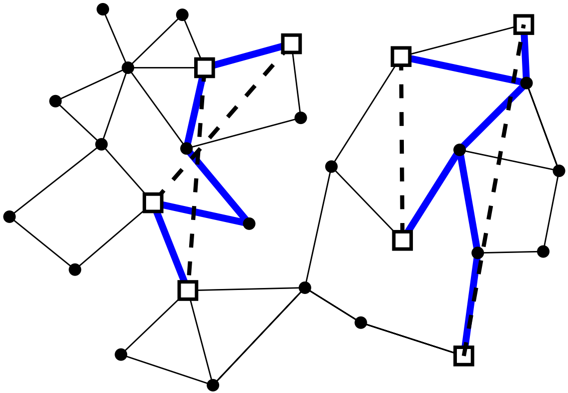
The Online Steiner Tree Problem



The Online Steiner Forest Problem



The Online Steiner Forest Problem



The Online Steiner Forest Problem

Online = you know the graph G in advance but the requests are revealed in an online fashion. The decision to buy an edge is **irrevocable**.

For Steiner **Tree**, one can assume the terminals form a single connected component.

The Online Steiner Forest Problem

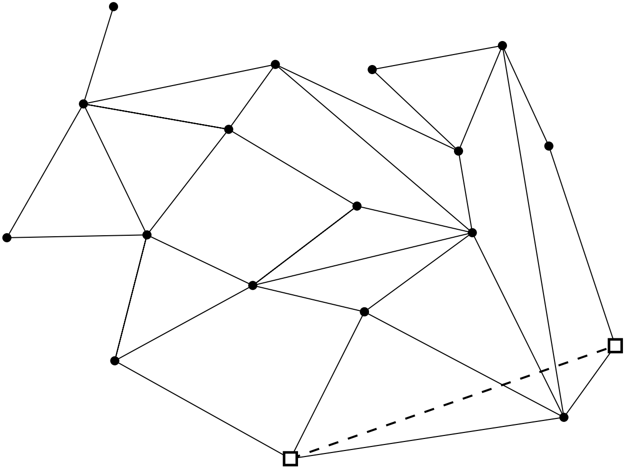
Online = you know the graph G in advance but the requests are revealed in an online fashion. The decision to buy an edge is **irrevocable**.

For Steiner **Tree**, one can assume the terminals form a single connected component.

What is the simplest online algorithm one can think of?

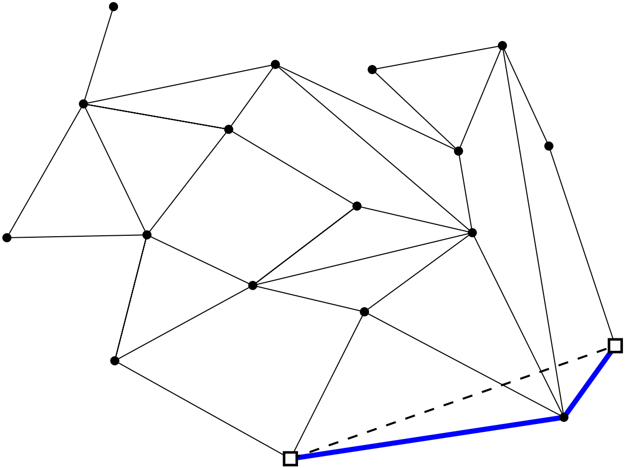
The Greedy algorithm

When a pair $\{s, t\}$ of terminals arrives, connect it by adding the cheapest edge set to the current solution.



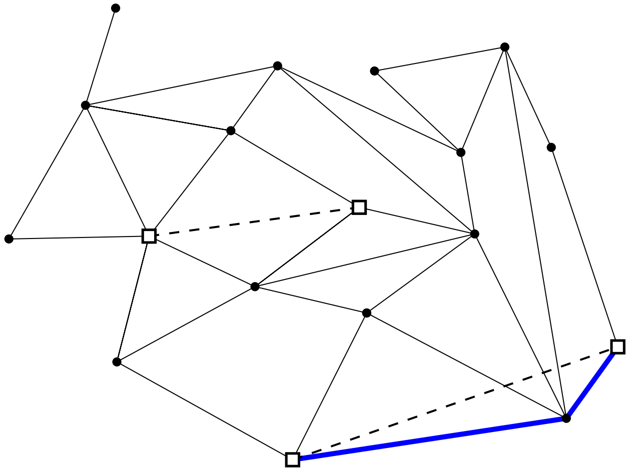
The Greedy algorithm

When a pair $\{s, t\}$ of terminals arrives, connect it by adding the cheapest edge set to the current solution.



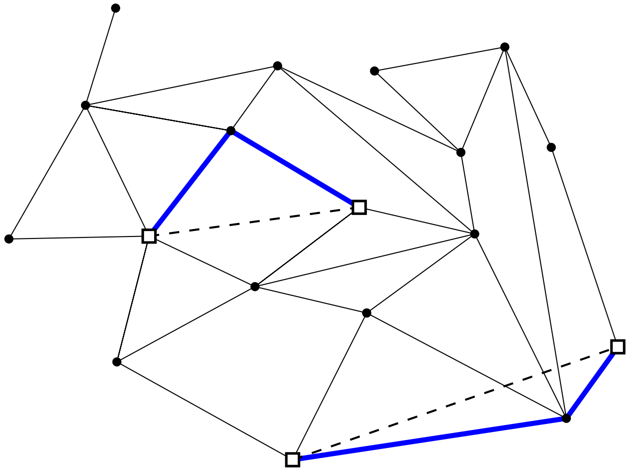
The Greedy algorithm

When a pair $\{s, t\}$ of terminals arrives, connect it by adding the cheapest edge set to the current solution.



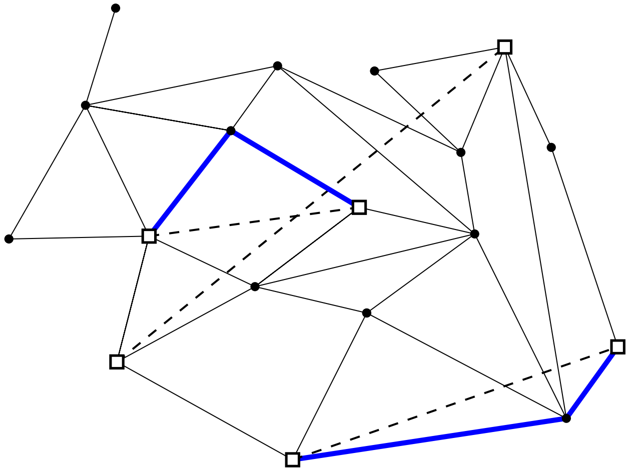
The Greedy algorithm

When a pair $\{s, t\}$ of terminals arrives, connect it by adding the cheapest edge set to the current solution.



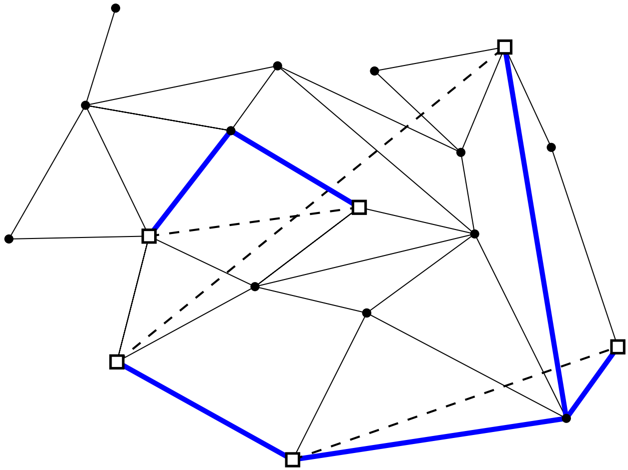
The Greedy algorithm

When a pair $\{s, t\}$ of terminals arrives, connect it by adding the cheapest edge set to the current solution.



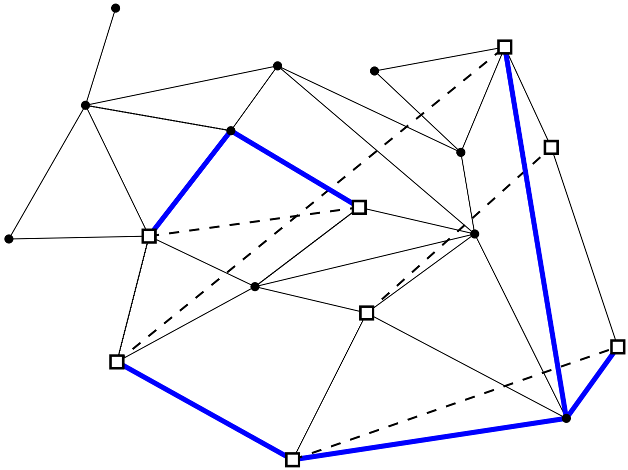
The Greedy algorithm

When a pair $\{s, t\}$ of terminals arrives, connect it by adding the cheapest edge set to the current solution.



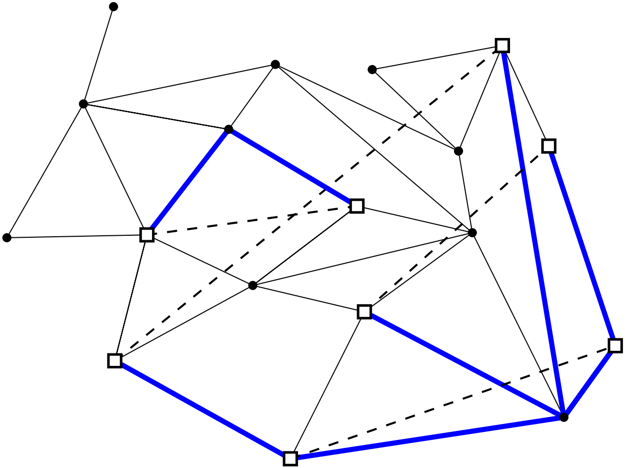
The Greedy algorithm

When a pair $\{s, t\}$ of terminals arrives, connect it by adding the cheapest edge set to the current solution.



The Greedy algorithm

When a pair $\{s, t\}$ of terminals arrives, connect it by adding the cheapest edge set to the current solution.



What do we know on Greedy?

	UB for Greedy	LB for any algorithm
Steiner Tree	$O(\log(k))$ [IW91]	$\Omega(\log(k))$ [IW91]
Steiner Forest	$O(\log^2(k))$ [AAB96]	$\Omega(\log(k))$ [IW91]

(k refers to the number of terminals/pairs of terminals).
[IM91]: Imase and Waxman, SIAM J. Discrete Math. 1991.
[AAB96]: Awerbuch, Azar, and Bartal, SODA 1996.

What do we know on Greedy?

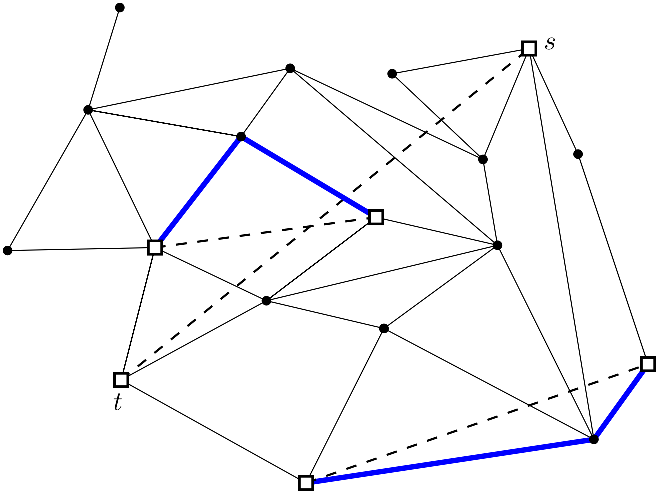
	UB for Greedy	LB for any algorithm
Steiner Tree	$O(\log(k))$ [IW91]	$\Omega(\log(k))$ [IW91]
Steiner Forest	$O(\log^2(k))$ [AAB96]	$\Omega(\log(k))$ [IW91]

(k refers to the number of terminals/pairs of terminals).
[IM91]: Imase and Waxman, SIAM J. Discrete Math. 1991.
[AAB96]: Awerbuch, Azar, and Bartal, SODA 1996.

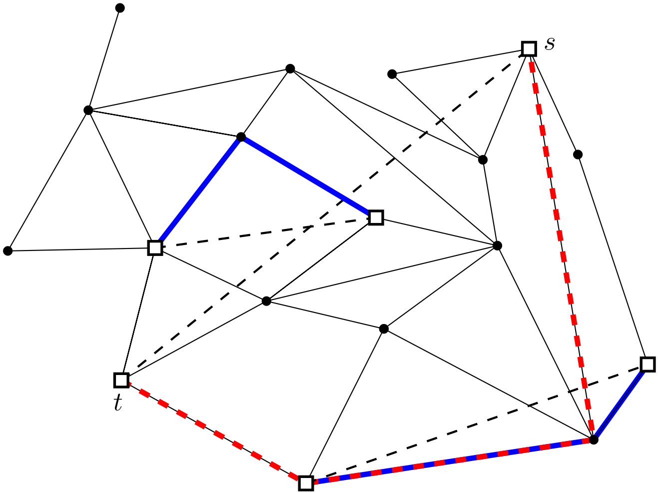
Conjecture (Awerbuch, Azar, and Bartal, SODA'96)

Greedy is $O(\log(k))$*-competitive for Online Steiner Forest.*

Our contributions

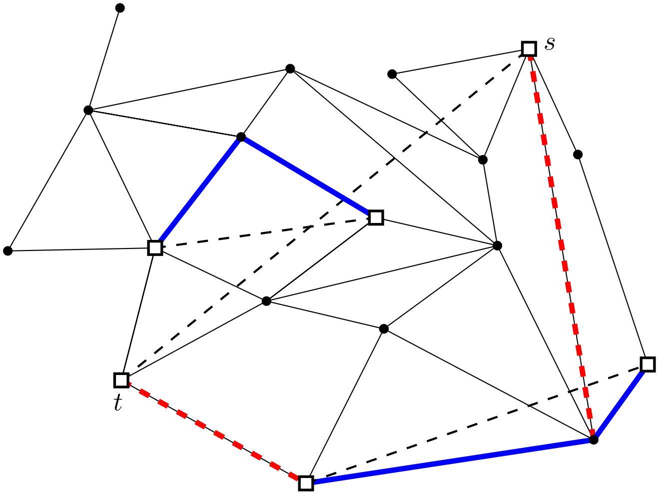


Our contributions



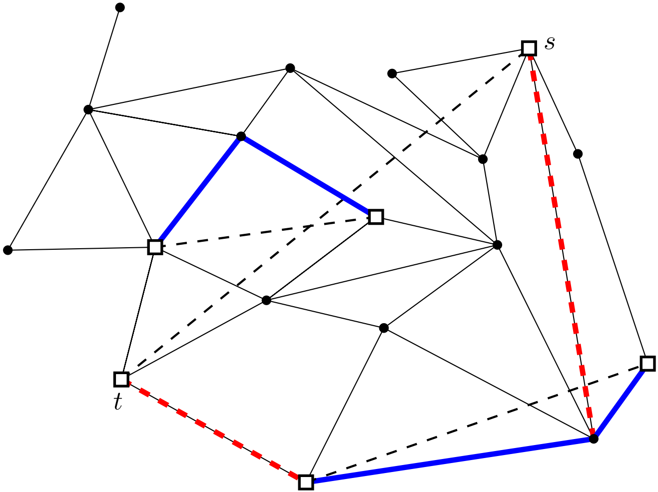
We have that $d_G(s, t) = 3$.

Our contributions



But Greedy pays only 2 to connect $p = \{s, t\}$!

Our contributions



Define the **contraction** by $\alpha(p) = d_G(s,t)/c_p = 3/2$.

Our contributions

Intuition: If the instance is hard for greedy, all pairs should have contraction 1.

Our contributions

Intuition: If the instance is hard for greedy, all pairs should have contraction 1.

Interestingly, all hard instances that were exhibited in the literature have contraction 1!

Our contributions – The main theorem

Theorem (Main theorem)

Fix $\alpha \geq 1$. If 1% of all pairs have contraction at most α, then Greedy is $O(\log(k) \cdot \max\{\log\log(k), \log(\alpha)\})$-competitive.

Our contributions – The main theorem

Theorem (Main theorem)

Fix $\alpha \geq 1$. *If* 1% *of all pairs have contraction at most* α, *then Greedy is* $O(\log(k) \cdot \max\{\log\log(k), \log(\alpha)\})$*-competitive.*

In an instance where all pairs have contraction 1, then greedy is $O(\log(k)\log\log(k))$-competitive.

Our contributions

Using our main theorem, we show.

Our contributions

Using our main theorem, we show.

Theorem

Greedy is $O(\log(k) \log \log(k))$*-competitive when comparing against the optimum* **tree** *solution.*

Our contributions

Using our main theorem, we show.

Theorem

Greedy is $O(\log(k) \log\log(k))$*-competitive when comparing against the optimum* **tree** *solution.*

Theorem (Offline Greedy)

Greedy is an $O(\log(k) \log\log(k))$*-approximation when the pairs* $\{s_i, t_i\}_{i \leq k}$ *are revealed in non-increasing value of* $d_G(s_i, t_i)$*.*

Our contributions

Using our main theorem, we show.

Theorem

Greedy is $O(\log(k) \log\log(k))$*-competitive when comparing against the optimum* **tree** *solution.*

Theorem (Offline Greedy)

Greedy is an $O(\log(k) \log\log(k))$*-approximation when the pairs* $\{s_i, t_i\}_{i \le k}$ *are revealed in non-increasing value of* $d_G(s_i, t_i)$.

Previous best upper and lower bounds for offline Greedy were also $O(\log^2(k))$ and $\Omega(\log(k))$.

Introduction to the proof techniques

The proof of Alon and Azar (SoCG'92) that Greedy is $O(\log(k))$-competitive for Steiner **Tree**:

Introduction to the proof techniques

The proof of Alon and Azar (SoCG'92) that Greedy is $O(\log(k))$-competitive for Steiner **Tree**:

1. By geometric grouping and rescaling, we can partition T into

$$T = \bigcup_{i=0}^{\log(k)} T^{(i)}$$

   where each $T^{(i)}$ is a **cost class**.

Introduction to the proof techniques

The proof of Alon and Azar (SoCG'92) that Greedy is $O(\log(k))$-competitive for Steiner **Tree**:

1. By geometric grouping and rescaling, we can partition T into

$$T = \bigcup_{i=0}^{\log(k)} T^{(i)}$$

   where each $T^{(i)}$ is a **cost class**.

2. Show that for each cost class $T^{(i)}$, the cost that Greedy pays for this cost class is $O(\mathrm{OPT})$.

Introduction to the proof techniques

The proof of Alon and Azar (SoCG'92) that Greedy is $O(\log(k))$-competitive for Steiner **Tree**:

1. By geometric grouping and rescaling, we can partition T into

$$T = \bigcup_{i=0}^{\log(k)} T^{(i)}$$

   where each $T^{(i)}$ is a **cost class**.

2. Show that for each cost class $T^{(i)}$, the cost that Greedy pays for this cost class is $O(\mathrm{OPT})$.

In total, Greedy pays $(\text{number of cost classes}) \cdot (\text{cost of one cost class})$ which is at most $O(\log(k))\mathrm{OPT}$.

Bounding the total cost of one cost class

Alon and Azar (SoCG'92) in a nutshell: build a dual solution!

Bounding the total cost of one cost class

Alon and Azar (SoCG'92) in a nutshell: build a dual solution!

Define c_i the cost of terminals in $T^{(i)}$.

Bounding the total cost of one cost class

Alon and Azar (SoCG'92) in a nutshell: build a dual solution!

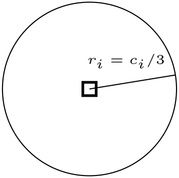
Define c_i the cost of terminals in $T^{(i)}$.

□

Bounding the total cost of one cost class

Alon and Azar (SoCG'92) in a nutshell: build a dual solution!

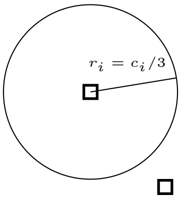
Define c_i the cost of terminals in $T^{(i)}$.



Bounding the total cost of one cost class

Alon and Azar (SoCG'92) in a nutshell: build a dual solution!

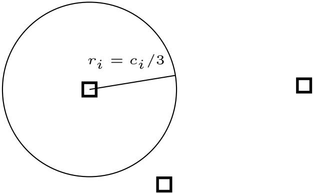
Define c_i the cost of terminals in $T^{(i)}$.



Bounding the total cost of one cost class

Alon and Azar (SoCG'92) in a nutshell: build a dual solution!

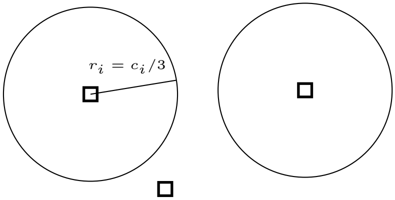
Define c_i the cost of terminals in $T^{(i)}$.



Bounding the total cost of one cost class

Alon and Azar (SoCG'92) in a nutshell: build a dual solution!

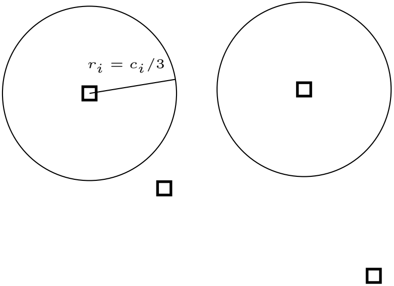
Define c_i the cost of terminals in $T^{(i)}$.



Bounding the total cost of one cost class

Alon and Azar (SoCG'92) in a nutshell: build a dual solution!

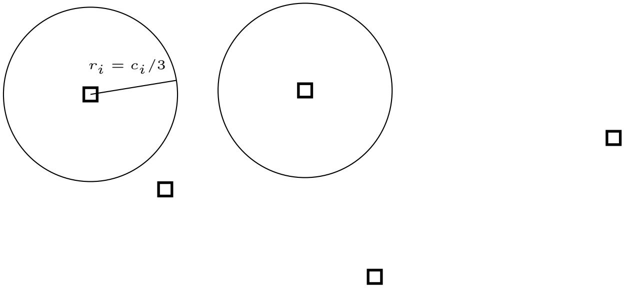
Define c_i the cost of terminals in $T^{(i)}$.



Bounding the total cost of one cost class

Alon and Azar (SoCG'92) in a nutshell: build a dual solution!

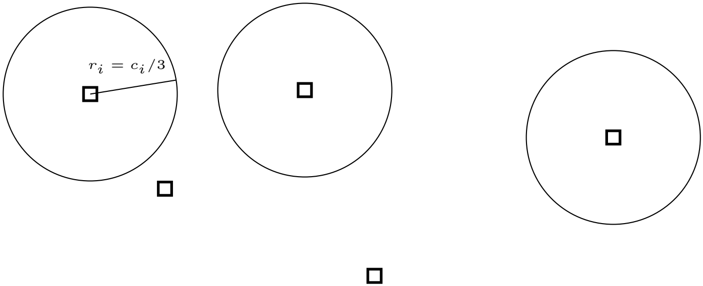
Define c_i the cost of terminals in $T^{(i)}$.



Bounding the total cost of one cost class

Alon and Azar (SoCG'92) in a nutshell: build a dual solution!

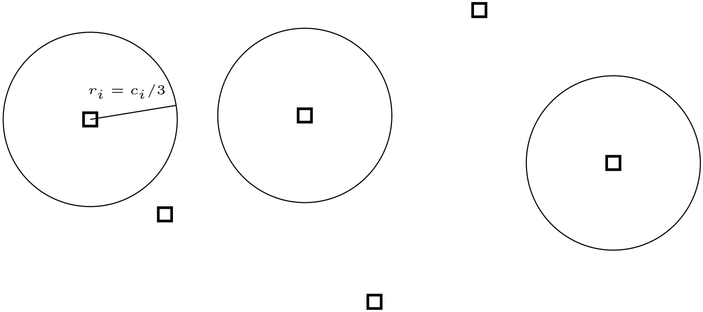
Define c_i the cost of terminals in $T^{(i)}$.



Bounding the total cost of one cost class

Alon and Azar (SoCG'92) in a nutshell: build a dual solution!

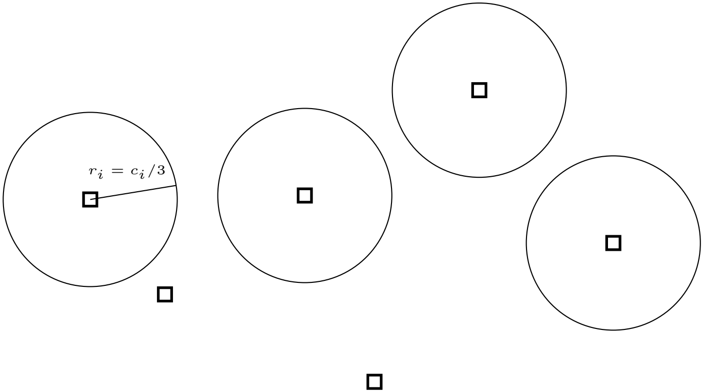
Define c_i the cost of terminals in $T^{(i)}$.



Bounding the total cost of one cost class

Alon and Azar (SoCG'92) in a nutshell: build a dual solution!

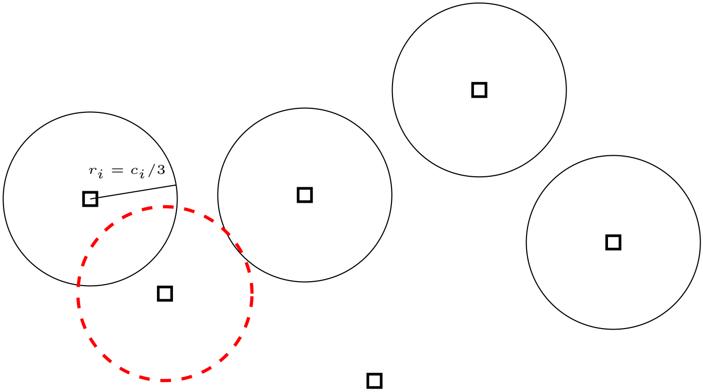
Define c_i the cost of terminals in $T^{(i)}$.



Bounding the total cost of one cost class

Alon and Azar (SoCG'92) in a nutshell: build a dual solution!

Define c_i the cost of terminals in $T^{(i)}$.



The proof in the case of Steiner Forest

The proof of Awerbuch, Azar, and Bartal (SODA'96) that Greedy is $O(\log^2(k))$-competitive for Steiner **Forest**:

The proof in the case of Steiner Forest

The proof of Awerbuch, Azar, and Bartal (SODA'96) that Greedy is $O(\log^2(k))$-competitive for Steiner **Forest**:

1. Partition P into
$$P = \bigcup_{i=0}^{\log(k)} P^{(i)}$$
where each $P^{(i)}$ is a **cost class**.

The proof in the case of Steiner Forest

The proof of Awerbuch, Azar, and Bartal (SODA'96) that Greedy is $O(\log^2(k))$-competitive for Steiner **Forest**:

1. Partition P into
$$P = \bigcup_{i=0}^{\log(k)} P^{(i)}$$
where each $P^{(i)}$ is a **cost class**.

2. Show that for each cost class $P^{(i)}$, the cost that Greedy pays for this cost class is $O(\log(k_i)) \cdot \mathrm{OPT}$, with $k_i = |P^{(i)}|$.

The proof in the case of Steiner Forest

"Show that for each cost class $P^{(i)}$, the cost that Greedy pays for this cost class is $O(\log(k_i)) \cdot \mathrm{OPT}$, with $k_i = |P^{(i)}|$."

Is this bound tight?

The proof in the case of Steiner Forest

"Show that for each cost class $P^{(i)}$, the cost that Greedy pays for this cost class is $O(\log(k_i)) \cdot \mathrm{OPT}$, with $k_i = |P^{(i)}|$."

Is this bound tight? **Yes!** 

The proof in the case of Steiner Forest

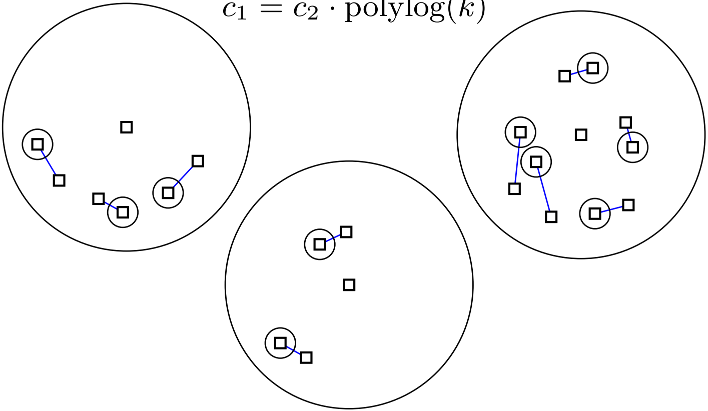
"Show that for each cost class $P^{(i)}$, the cost that Greedy pays for this cost class is $O(\log(k_i)) \cdot \mathrm{OPT}$, with $k_i = |P^{(i)}|$."

Is this bound tight? **Yes!** 

In those bad examples, **all** pairs have contraction exactly 1!

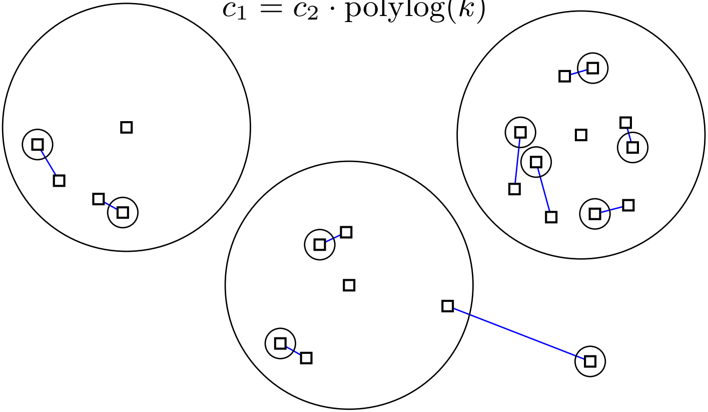
A better proof when the contraction is 1

$$c_1 = c_2 \cdot \text{polylog}(k)$$



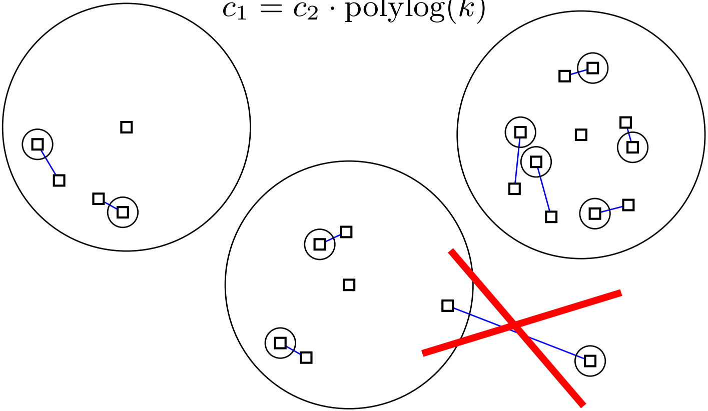
A better proof when the contraction is 1

$$c_1 = c_2 \cdot \text{polylog}(k)$$

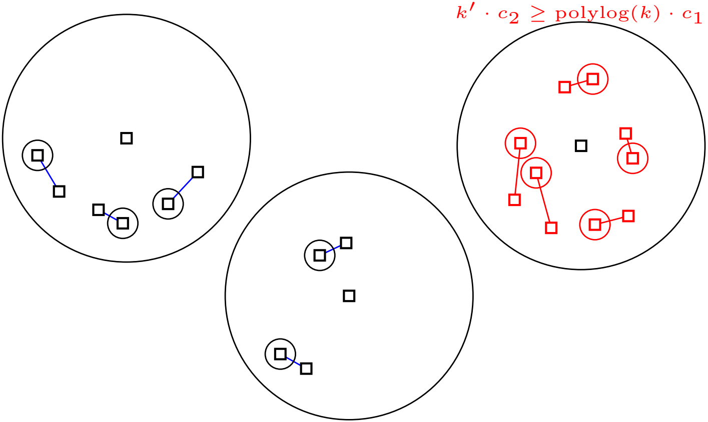


A better proof when the contraction is 1

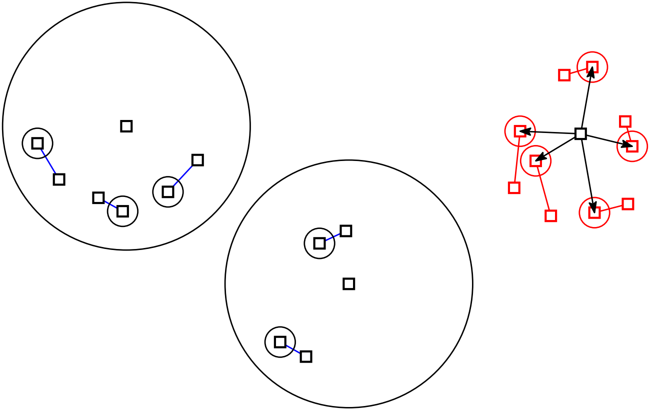
$$c_1 = c_2 \cdot \mathrm{polylog}(k)$$



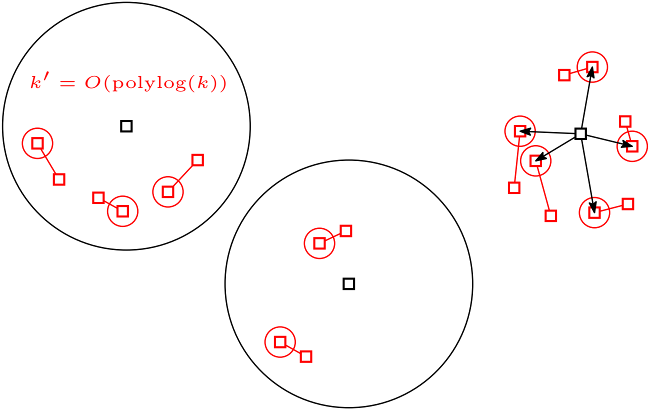
A better proof when the contraction is 1



A better proof when the contraction is 1



A better proof when the contraction is 1



A better proof when the contraction is 1

Theorem

If all pairs have contraction 1, then Greedy is $O(\log(k) \log\log(k))$*-competitive.*

Conclusion

- Can we show the $O(\log(k) \log\log(k))$ upper bound in general?
  Our belief: The hardest instances have low contraction.

Conclusion

- Can we show the $O(\log(k)\log\log(k))$ upper bound in general? **Our belief:** The hardest instances have low contraction.
- If you believe that Greedy is $\Omega(\log^{1+\epsilon}(k))$-competitive, then the lower bound should have almost all pairs with contraction $2^{\Omega(\log^{\epsilon}(k))}$! **It would require a totally different thinking!**

Conclusion

- Can we show the $O(\log(k) \log\log(k))$ upper bound in general? **Our belief:** The hardest instances have low contraction.
- If you believe that Greedy is $\Omega(\log^{1+\epsilon}(k))$-competitive, then the lower bound should have almost all pairs with contraction $2^{\Omega(\log^{\epsilon}(k))}$! **It would require a totally different thinking!**
- How to get rid of the $\log\log(k)$?