An $O(1)$-Approximation for Minimum Spanning Tree Interdiction

Rico Zenklusen

ETH Zurich

What are interdiction problems?

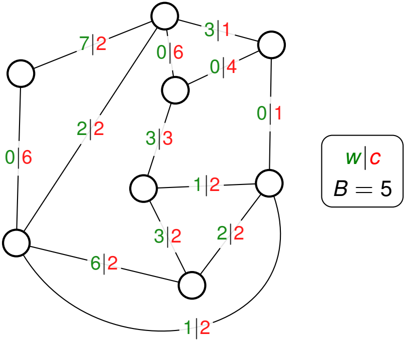
Key question motivating interdiction problems

How sensitive is a system wrt failure/destruction of some of its components?

Interdiction analysis is about worst-case failures.

- Learn how robust a system is.
- Identify weakest spots ⟨ protect system, / attack/inhibit undesirable system/process.

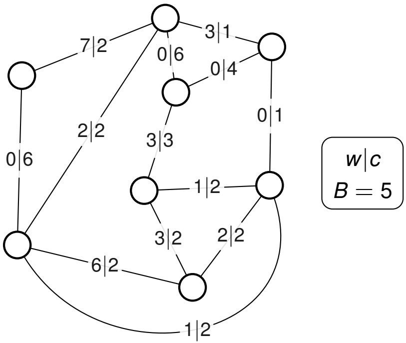
Minimum Spanning Tree (MST) Interdiction



Input

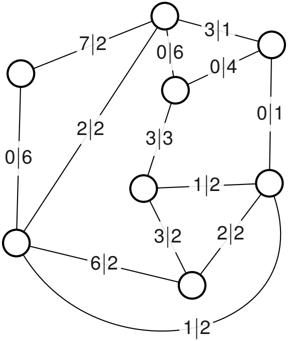
- $G = (V, E)$
- $w : E \to \mathbb{Z}_{\geq 0}$ (weights)
- $c : E \to \mathbb{Z}_{>0}$ (costs)
- $B \in \mathbb{Z}_{>0}$ (budget)

Minimum Spanning Tree (MST) Interdiction



Input

- $G = (V, E)$
- $w : E \to \mathbb{Z}_{\geq 0}$ (weights)
- $c : E \to \mathbb{Z}_{>0}$ (costs)
- $B \in \mathbb{Z}_{>0}$ (budget)



$w|c$
$B = 5$

Input

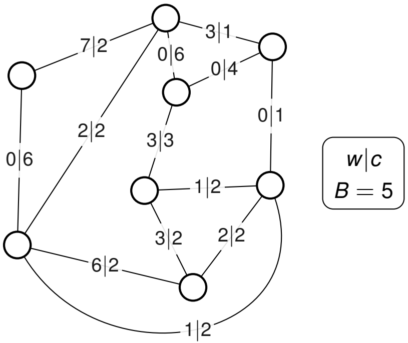
- $G = (V, E)$
- $w : E \to \mathbb{Z}_{\geq 0}$ (weights)
- $c : E \to \mathbb{Z}_{>0}$ (costs)
- $B \in \mathbb{Z}_{>0}$ (budget)

Task

Create graph with heavy MST by removing $R \subseteq E$ respecting budget.

$$\max\{w(\mathrm{MST}((V, E \setminus R))) \mid R \subseteq E, c(R) \leq B\}$$

Minimum Spanning Tree (MST) Interdiction



Input

- $G = (V, E)$
- $w : E \to \mathbb{Z}_{\geq 0}$ (weights)
- $c : E \to \mathbb{Z}_{>0}$ (costs)
- $B \in \mathbb{Z}_{>0}$ (budget)

Task

Create graph with heavy MST by removing $R \subseteq E$ respecting budget.

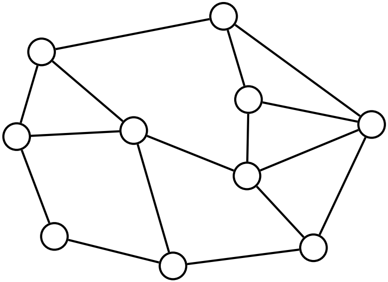
$$\max\{w(\mathrm{MST}((V, E \setminus R))) \mid R \subseteq E, c(R) \leq B\}$$

Assumption: Budget not large enough to disconnect graph.

Maximum Component Problem (MCP)

Break graph into max # of connected components by removing q edges.

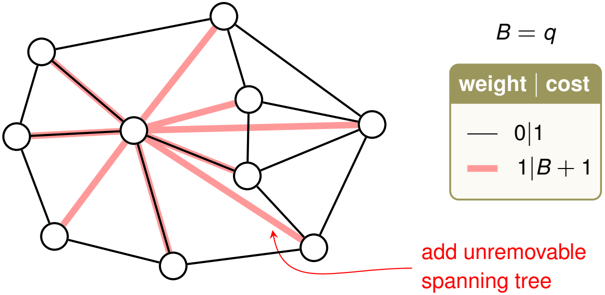
(k-cut problem where roles of objective and constraint are swapped.)



Maximum Component Problem (MCP)

Break graph into max # of connected components by removing q edges.

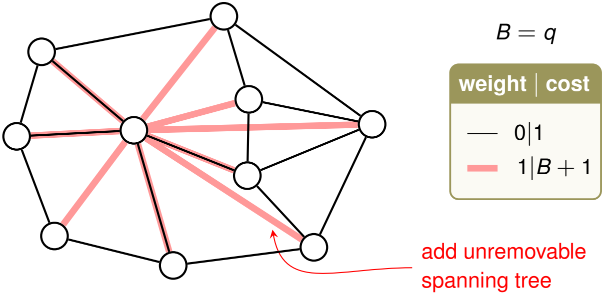
(k-cut problem where roles of objective and constraint are swapped.)



Maximum Component Problem (MCP)

Break graph into max # of connected components by removing q edges.

(k-cut problem where roles of objective and constraint are swapped.)



Generalization of MCP: Budgeted Graph Disconnection (BGD)

Break graph into max # of conn. comp. by removing edges of cost $\leq B$.

Best known approximation ratios

MST interdiction	$O(\log m)$	[Frederickson and Solis-Oba, 1996]
k most vital edges	$O(\log k)$	[Frederickson and Solis-Oba, 1996]

- m : # of edges

Previous results

Best known approximation ratios

MST interdiction	$O(\log m)$	[Frederickson and Solis-Oba, 1996]
k most vital edges	$O(\log k)$	[Frederickson and Solis-Oba, 1996]

- m : # of edges

Is there an $O(1)$-approximation for k most vital edges, or even MST interdiction?

Best known running times for most vital edge problem

Deterministic algo	$O(m \cdot \alpha(m, n))$	[Iwano and Katoh, 1993, Chazelle, 2000, Tarjan, 1979]
Randomized algo	$O(m)$	[Iwano and Katoh, 1993, Karger et al., 1995, Dixon et al., 1992]

- m : # of edges
- n : # of vertices
- $\alpha(\cdot, \cdot)$: inv. Ackermann function

Best known running times for most vital edge problem

Deterministic algo	$O(m \cdot \alpha(m,n))$	[Iwano and Katoh, 1993, Chazelle, 2000, Tarjan, 1979]
Randomized algo	$O(m)$	[Iwano and Katoh, 1993, Karger et al., 1995, Dixon et al., 1992]

- m : # of edges
- n : # of vertices
- $\alpha(\cdot,\cdot)$: inv. Ackermann function

Further related work on k most vital edges

- Exact exp. time algos (serial and parallel) [Liang and Shen, 1997, Liang, 2001, Bazgan et al., 2011]
- Parameterized complexity [Guo and Shrestha, 2014]

Our results

First $O(1)$-approximation for MST interdiction.

Some consequences

- $O(1)$-approx for metric TSP and TSP path interdiction.

"Remove edges of cost $\leq B$ to make it expensive to visit all vertices."

Follows from $\text{val}(\text{MST}) \leq \text{val}(\text{TSP (path)}) \leq 2\,\text{val}(\text{MST})$.

Some technical hurdles we overcome

- Lack of good relaxation.
- MST interdiction is equivalent to "multi-level" BGD.
  $\rightarrow$ Hard to control interactions between levels.

Some basic observations and notation

Assumption on weights (losing a factor of at most 2)

We assume weights are powers of 2 by rounding weights down.

$\Rightarrow E = E_{-1} \,\dot\cup\, E_0 \,\dot\cup\, E_1 \,\dot\cup\, \dots \,\dot\cup\, E_p.$

$$E_{-1} = \{e \in E \mid w(e) = 0\}$$
$$E_i = \{e \in E \mid w(e) = 2^i\} \text{ for } i \in [p]$$

Assumptions related to E_p

Can assume optimal interdiction set $R \subseteq E$ satisfies $R \cap E_p = \emptyset$.

(Since budget is not enough to disconnect G.)

Can assume $\exists$ interdiction set $R \subseteq E$ s.t. $\mathrm{MST}(V, E \setminus R) \cap E_p \neq \emptyset$.

(Otherwise, delete all edges E_p without changing problem.)

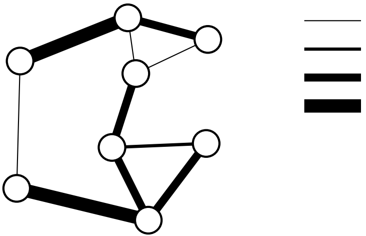
$\Rightarrow$ This implies that opt. value of MST interdiction $\geq 2^p$.

Interpretation as multi-level BGD

$$\text{val(MST)} = \sigma(E_{-1}) - 1 + \sum_{i=0}^{p-1} 2^i \cdot (\sigma(E_{\leq i}) - 1)$$

where for $A \subseteq E$:

$\sigma(A) :=$ # of conn. comp. in (V, A).



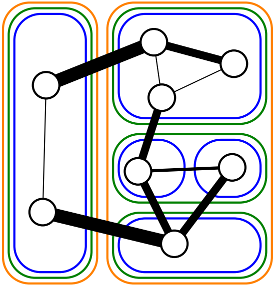
- weight = 0
- weight = 1
- weight = 2
- weight = 4

Interpretation as multi-level BGD

$$\text{val}(\text{MST}) = \sigma(E_{-1}) - 1 + \sum_{i=0}^{p-1} 2^i \cdot (\sigma(E_{\leq i}) - 1)$$

where for $A \subseteq E$:

$\sigma(A) :=$ # of conn. comp. in (V, A).



- weight = 0
- weight = 1
- weight = 2
- weight = 4
- comp. of E_{-1}
- comp. of $E_{\leq 0}$
- comp. of $E_{\leq 1}$

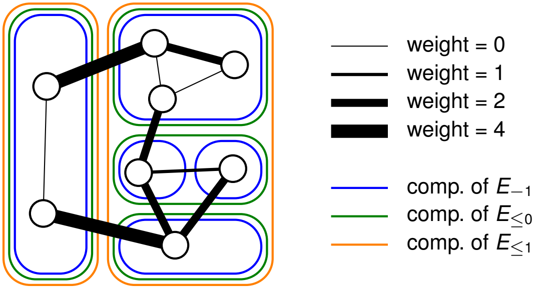
$$\begin{aligned}\text{val}(MST) = & \quad (5 - 1) \\ & +1 \cdot (4 - 1) \\ & +2 \cdot (2 - 1) \\ = & \; 9\end{aligned}$$

Interpretation as multi-level BGD

$$\text{val}(\text{MST}) = \sigma(E_{-1}) - 1 + \sum_{i=0}^{p-1} 2^i \cdot (\sigma(E_{\leq i}) - 1)$$

where for $A \subseteq E$:

$\sigma(A) :=$ # of conn. comp. in (V, A).



$$\begin{aligned} \text{val}(MST) = & \quad (5 - 1) \\ & + 1 \cdot (4 - 1) \\ & + 2 \cdot (2 - 1) \\ = & \; 9 \end{aligned}$$

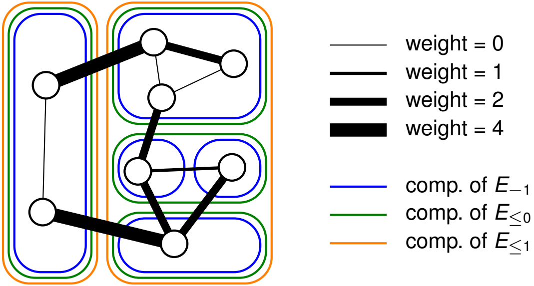
$$\text{val}(U) := \text{val}(MST(V, E \setminus U)) = \sigma(E_{-1} \setminus U) - 1 + \sum_{i=0}^{p-1} 2^i \cdot (\sigma(E_{\leq i} \setminus U) - 1)$$

Interpretation as multi-level BGD

$$\mathrm{val}(\mathrm{MST}) = \sigma(E_{-1}) - 1 + \sum_{i=0}^{p-1} 2^i \cdot (\sigma(E_{\leq i}) - 1)$$

where for $A \subseteq E$:

$\sigma(A) :=$ # of conn. comp. in (V, A).



$$\begin{aligned} \mathrm{val}(MST) = & \quad (5 - 1) \\ & +1 \cdot (4 - 1) \\ & +2 \cdot (2 - 1) \\ = 9 \end{aligned}$$

$$\mathrm{val}(U) := \mathrm{val}(MST(V, E \setminus U)) = \sigma(E_{-1} \setminus U) - 1 + \sum_{i=0}^{p-1} 2^i \cdot (\sigma(E_{\leq i} \setminus U) - 1)$$

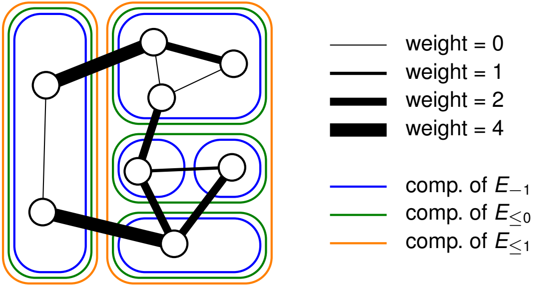
Goal: $\max\{\mathrm{val}(U) \mid U \subseteq E, c(U) \leq B\}$

Interpretation as multi-level BGD

$$\text{val}(\text{MST}) = \sigma(E_{-1}) - 1 + \sum_{i=0}^{p-1} 2^i \cdot (\sigma(E_{\leq i}) - 1)$$

where for $A \subseteq E$:

$\sigma(A) :=$ # of conn. comp. in (V, A).



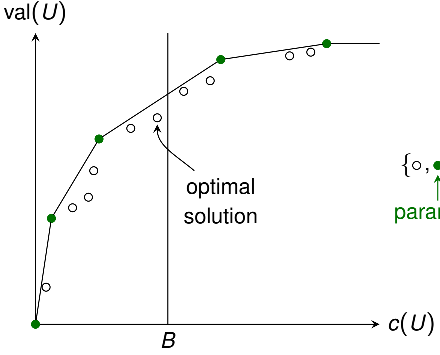
$$\begin{aligned} \text{val}(MST) = &\ (5 - 1) \\ &+1 \cdot (4 - 1) \\ &+2 \cdot (2 - 1) \\ = &\ 9 \end{aligned}$$

$$\text{val}(U) := \text{val}(MST(V, E \setminus U)) = \sigma(E_{-1} \setminus U) - 1 + \sum_{i=0}^{p-1} 2^i \cdot (\sigma(E_{\leq i} \setminus U) - 1)$$

Goal: $\max\{\text{val}(U) \mid U \subseteq E, c(U) \leq B\}$

$\max\{\sigma(E_{\leq i} \setminus U) \mid U \subseteq E, c(U) \leq B\}$ is BGD $\forall i$.

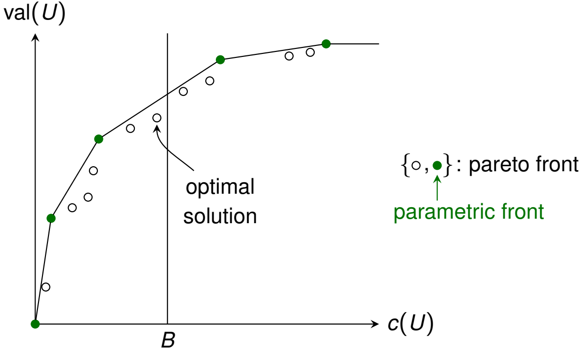
Relaxation through submodular minimization



$\{\circ, \bullet\}$: pareto front

parametric front

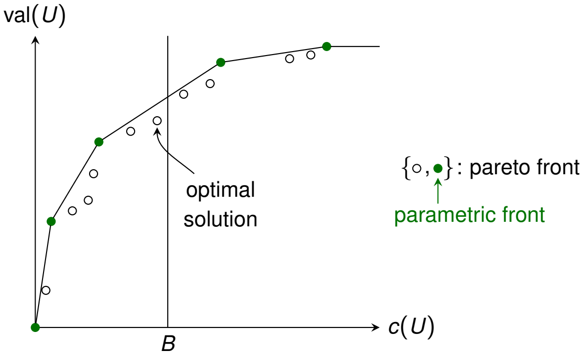
Relaxation through submodular minimization



$$\textcolor{red}{\mathrm{val}}(U) = \textcolor{orange}{\sigma}(E_{-1} \setminus U) - 1 + \sum_{i=0}^{p-1} 2^i \cdot (\textcolor{orange}{\sigma}(E_{\leq i} \setminus U) - 1)$$

supermodular (val) — supermodular (σ)

Relaxation through submodular minimization

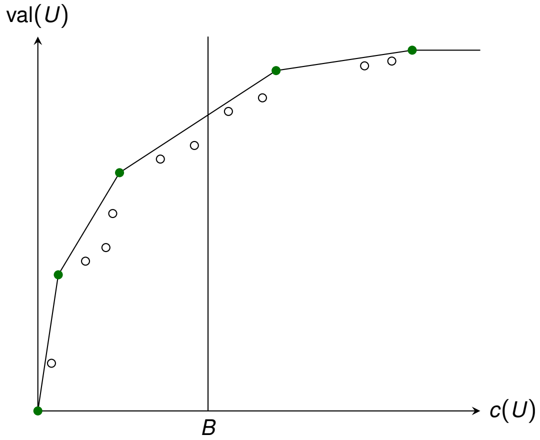


$$\text{val}(U) = \sigma(E_{-1} \setminus U) - 1 + \sum_{i=0}^{p-1} 2^i \cdot (\sigma(E_{\leq i} \setminus U) - 1)$$

supermodular (val) — supermodular (σ)

Can find parametric frontier efficiently by submodular function minimization (SFM):

Solve $\min\{\lambda \cdot c(U) - \text{val}(U) \mid U \subseteq E_{\leq p-1}\}$ for varying $\lambda \geq 0$.

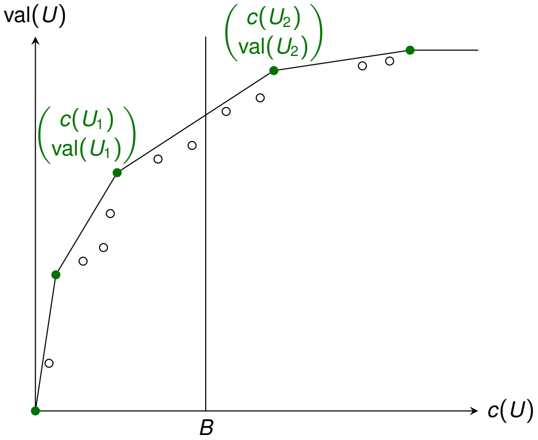


Theorem

For any $U \subseteq E$, $c(U) \geq B$, we can find $R \subseteq U$, $c(R) \leq B$ s.t.

$$\text{val}(R) \geq \frac{1}{6} \cdot B \cdot \frac{\text{val}(U)}{c(U)}.$$

High-level plan



Theorem

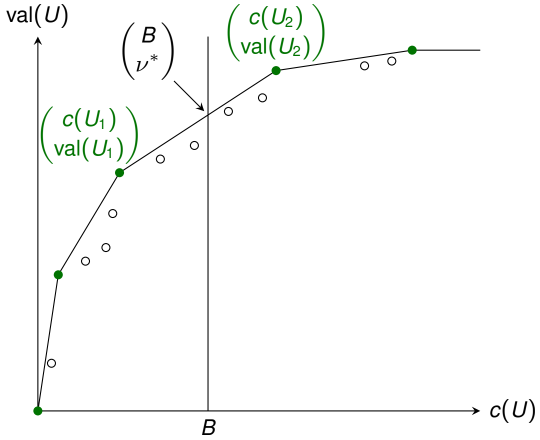
For any $U \subseteq E$, $c(U) \geq B$, we can find $R \subseteq U$, $c(R) \leq B$ s.t.

$$\text{val}(R) \geq \frac{1}{6} \cdot B \cdot \frac{\text{val}(U)}{c(U)}.$$

Algorithm

1. Compute U_1, U_2 with SFM.
2. Invoke theorem wrt $U_2 \to R_2$.
3. Return better of U_1, R_2.

High-level plan



Theorem

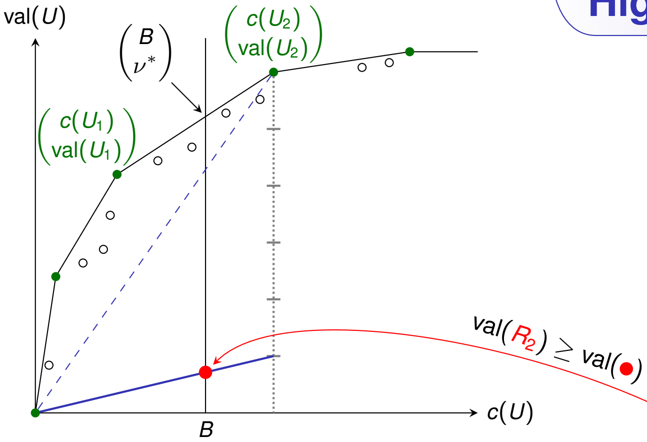
For any $U \subseteq E$, $c(U) \geq B$, we can find $R \subseteq U$, $c(R) \leq B$ s.t.

$$\text{val}(R) \geq \frac{1}{6} \cdot B \cdot \frac{\text{val}(U)}{c(U)}.$$

Algorithm

1. Compute U_1, U_2 with SFM.
2. Invoke theorem wrt $U_2 \to R_2$.
3. Return better of U_1, R_2.

High-level plan



Theorem

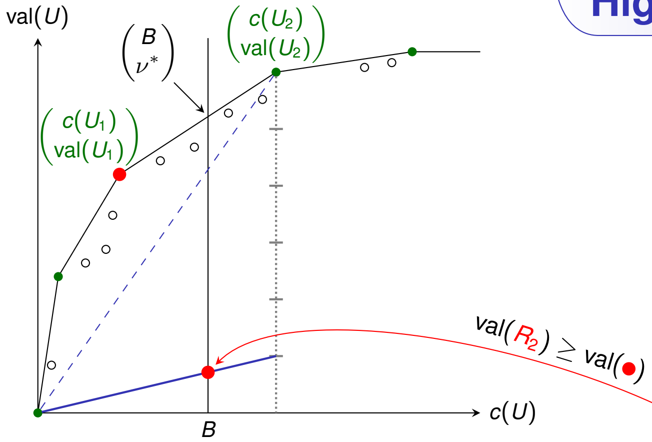
For any $U \subseteq E$, $c(U) \geq B$, we can find $R \subseteq U$, $c(R) \leq B$ s.t.

$$\text{val}(R) \geq \frac{1}{6} \cdot B \cdot \frac{\text{val}(U)}{c(U)}.$$

Algorithm

1. Compute U_1, U_2 with SFM.
2. Invoke theorem wrt $U_2 \rightarrow R_2$.
3. Return better of U_1, R_2.

High-level plan



Theorem

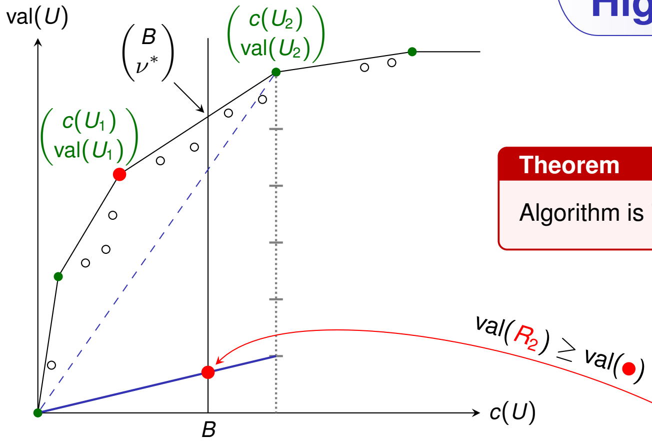
For any $U \subseteq E$, $c(U) \geq B$, we can find $R \subseteq U$, $c(R) \leq B$ s.t.

$$\text{val}(R) \geq \frac{1}{6} \cdot B \cdot \frac{\text{val}(U)}{c(U)}.$$

Algorithm

1. Compute U_1, U_2 with SFM.
2. Invoke theorem wrt $U_2 \to R_2$.
3. Return better of U_1, R_2.

High-level plan



Theorem

Algorithm is 7-approximation.

Theorem

For any $U \subseteq E$, $c(U) \geq B$, we can find $R \subseteq U$, $c(R) \leq B$ s.t.

$$\text{val}(R) \geq \frac{1}{6} \cdot B \cdot \frac{\text{val}(U)}{c(U)}.$$

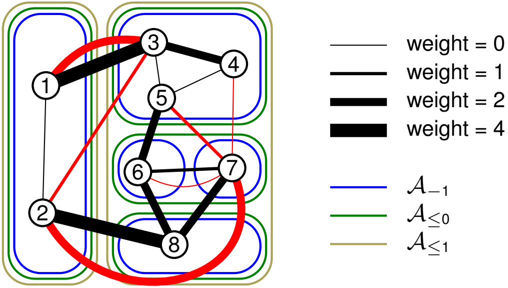
Algorithm

1. Compute U_1, U_2 with SFM.
2. Invoke theorem wrt $U_2 \rightarrow R_2$.
3. Return better of U_1, R_2.

Given: $U \subseteq E$, $c(U) > B$.
Goal: $R \subseteq U$, $c(R) \approx B$ s.t. $\frac{\text{val}(R)}{c(R)} \geq \frac{1}{6} \frac{\text{val}(U)}{c(U)}$.

Goal: exploit/preserve synergies of *U* across levels

$$\text{val}(U) = \sigma(E_{-1} \setminus U) - 1 + \sum_{i=0}^{p-1} 2^i \cdot (\sigma(E_{\leq i} \setminus U) - 1)$$



Looking at levels

Given: $U \subseteq E$, $c(U) > B$.
Goal: $R \subseteq U$, $c(R) \approx B$ s.t. $\frac{\text{val}(R)}{c(R)} \geq \frac{1}{6} \frac{\text{val}(U)}{c(U)}$.

Goal: exploit/preserve synergies of U across levels

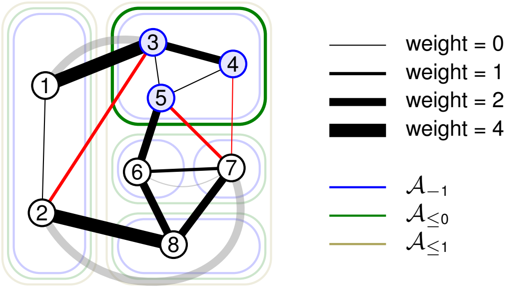
$$\text{val}(U) = \sigma(E_{-1} \setminus U) - 1 + \sum_{i=0}^{p-1} 2^i \cdot (\sigma(E_{\leq i} \setminus U) - 1)$$

$U_{\leq 0} := U \cap E_{\leq i}$

↓

$\delta(A) \cap U_{\leq 0}$

removing these edges creates comp. A in $\mathcal{A}_0$



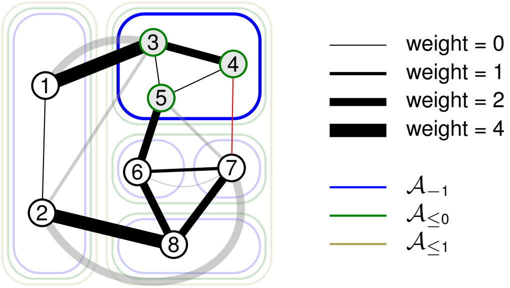
Given: $U \subseteq E$, $c(U) > B$.
Goal: $R \subseteq U$, $c(R) \approx B$ s.t. $\frac{\text{val}(R)}{c(R)} \geq \frac{1}{6} \frac{\text{val}(U)}{c(U)}$.

Goal: exploit/preserve synergies of U across levels

$$\text{val}(U) = \sigma(E_{-1} \setminus U) - 1 + \sum_{i=0}^{p-1} 2^i \cdot (\sigma(E_{\leq i} \setminus U) - 1)$$

$\delta(A) \cap U_{\leq -1}$

removing these edges creates comp. A in $\mathcal{A}_{-1}$



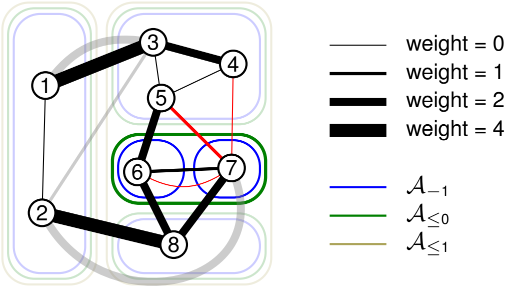
Given: $U \subseteq E$, $c(U) > B$.
Goal: $R \subseteq U$, $c(R) \approx B$ s.t. $\frac{\text{val}(R)}{c(R)} \geq \frac{1}{6} \frac{\text{val}(U)}{c(U)}$.

Goal: exploit/preserve synergies of *U* across levels

$$\text{val}(U) = \sigma(E_{-1} \setminus U) - 1 + \sum_{i=0}^{p-1} 2^i \cdot (\sigma(E_{\leq i} \setminus U) - 1)$$

$U_{\leq 0}(A)$

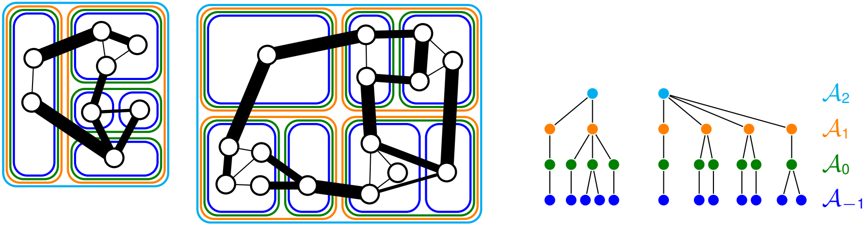
removing these edges creates comp. A in $\mathcal{A}_{-1}$ and all descendant comp.



$$U_{\leq i}(A) := \{e \in U_{\leq i} \mid e \text{ has at least one endpoint in } A\}$$

Picking disjoint comp. in hierarchy

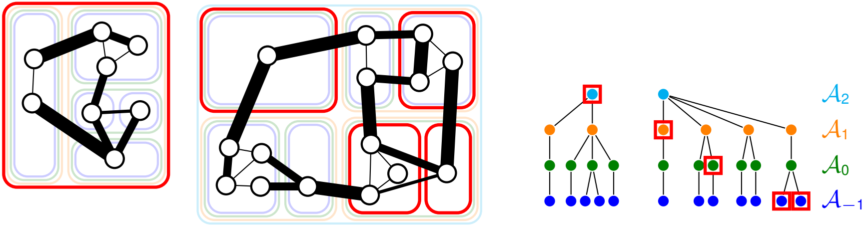
Will chose $(A_1, i_1), \ldots, (A_\beta, i_\beta)$ with $\{A_j \in \mathcal{A}_j\}_j$ disjoint and set $R = \cup_{j=1}^{\beta} U_{\leq i_j}(A_j)$.



- Example of $(V, E \setminus U)$ with edge weights $0, 1, 2, 4, 8$.
- Edges of weight 8 are not shown since wlog they span all vertices.

Picking disjoint comp. in hierarchy

Will chose $(A_1, i_1), \ldots, (A_\beta, i_\beta)$ with $\{A_j \in \mathcal{A}_j\}_j$ disjoint and set $R = \cup_{j=1}^{\beta} U_{\leq i_j}(A_j)$.

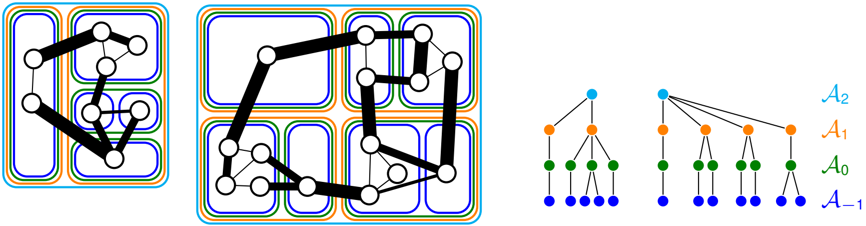


- Example of $(V, E \setminus U)$ with edge weights $0, 1, 2, 4, 8$.
- Edges of weight 8 are not shown since wlog they span all vertices.

Local measure of cost and impact

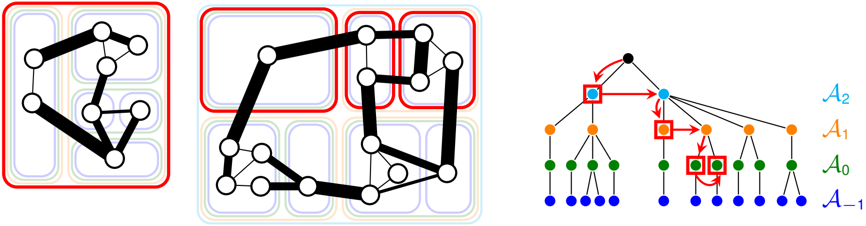
Auxiliary functions to measure things locally for $A \in \mathcal{A}_i$

$$\left.\begin{array}{r}\text{aux. cost: } \kappa_i(A) = \Theta\big(c(U_{\leq i}(A))\big) \\ \text{aux. impact: } g_i(A) = \Theta\left(\sum_{\ell=0}^{i} 2^\ell \cdot |\{D \in \mathcal{A}_\ell \mid D \subseteq A\}|\right)\end{array}\right\} \text{aux. eff.: } \rho_i(A) = \frac{g_i(A)}{\kappa_i(A)}.$$



Sketch of algorithm (traversing tree top down)

- Consider each child (A, i) of current node wrt decreasing auxiliary efficiency.
  - If $c(R \cup U_{\leq i}(A)) \leq B \longrightarrow R = R \cup U_{\leq i}(A)$.
  - Else: move down to first child of (A, i) and repeat.
- Algo stops when either: (i) leaf reached and not selected, or
  (ii) all children of some node are selected.



Ordering of siblings: Siblings further left have higher auxiliary efficencies.

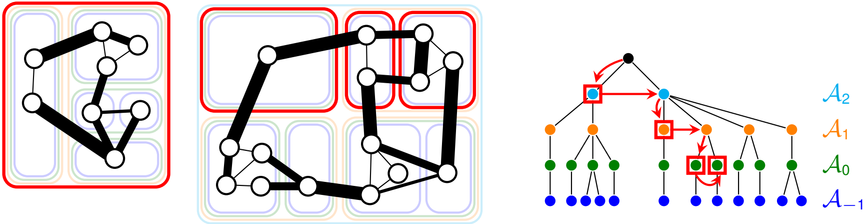
Some aspects of analysis

Average aux. efficiency of children is close to aux. eff. of parent.
→ Error term is gemetrically decreasing and of size $O(2^p)$.

To deal with error term we return best of following two:

(i) constructed set R;
(ii) trivial sol. of value 2^p (making sure one E_p edge is used).

→ Returned set has value: $\geq \frac{1}{6} \cdot B \cdot \frac{\mathrm{val}(U)}{c(U)}$



Summary

- 14-approximation for MST interdiction.
- 28-approximation for metric TSP interdiction.

Many interesting computational questions around interdiction problems are still open.

Some approximation/hardness results

Interdiction of ...	Best known approx. ratio	Hardness result
Shortest paths	-	APX-hardness [Khachiyan et. al, Th Comp Sys '08]
Maximum flow	- (2-pseudoapprox.) [Burch et al., thematic book on int. '03]	strongly NP-hard [Wood, Math & Comp Modeling, '93], [Phillips, STOC'93] "densest k-subgraph hard" [Chestnut & Z., arXiv 15]
Maximum flow on planar graphs	FPTAS [Phillips, STOC'93]	weakly NP-hard
MSTs	$O(\log m)$ [Frederickson & Solis-Oba, SODA'96] $O(1)$ [Z. FOCS'15]	strongly NP-hard [F. & S.-O., SODA'96]
Matchings and some packing problems	$O(1)$ [Dinitz & Gupta, IPCO'13]	strongly NP hard [Z. et al., Disc Math '10]
Connectivity	PTAS [Z., OR Letters '15]	weakly NP-hard
...		

Armon, A. and Zwick, U. (2006). Multicriteria global minimum cuts. *Algorithmica*, 46:15–26.

Bazgan, C., Toubaline, S., and Tuza, Z. (2011). The most vital nodes with respect to independent set and vertex cover. *Discrete Applied Mathematics*, 159(17):1933 – 1946.

Burch, C., Carr, R., Krumke, S., Marathe, M., Phillips, C., and Sundberg, E. (2003). A decomposition-based pseudoapproximation algorithm for network flow inhibition. In *Network Interdiction and Stochastic Integer Programming*, chapter 3, pages 51–68. Springer.

Chazelle, B. (2000). A minimum spanning tree algorithm with inverse-ackermann type complexity. *Journal of the ACM*, 47(6):1028–1047.

Dinitz, M. and Gupta, A. (2013). Packing interdiction and partial covering problems. In *Proceedings of 16th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 157–168.

Dixon, B., Rauch, M., and Tarjan, R. E. (1992). Verification and sensitivity analysis of minimum spanning trees in linear time. *SIAM Journal on Computing*, 21(6):1184–1192.

Frederickson, G. N. and Solis-Oba, R. (1996). Increasing the weight of minimum spanning trees. In *SODA '96: Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 539–546, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Guo, J. and Shrestha, Y. (2014). Parameterized complexity of edge interdiction problems. In *Proceedings of the 20th Computing and Combinatorics Conference (COCOON)*, pages 166–178.

Iwano, K. and Katoh, N. (1993). Efficient algorithms for finding the most vital edge of a minimum spanning tree. *Information Processing Letters*, 48(5):211 – 213.

References II

Karger, D. R., Klein, P. N., and Tarjan, R. E. (1995). A randomized linear-time algorithm to find minimum spanning trees. *Journal of the ACM*, 42(2):321–328.

Khachiyan, L., Boros, E., Borys, K., Elbassioni, K., Gurvich, V., Rudolf, G., and Zhao, J. (2008). On short paths interdiction problems: Total and node-wise limited interdiction. *Theoretical Computer Science*, 43(2):204–233.

Liang, W. (2001). Finding the k most vital edges with respect to minimum spanning trees for fixed k. *Discrete Applied Mathematics*, 113(2-3):319–327.

Liang, W. and Shen, X. (1997). Finding the k most vital edges in the minimum spanning tree problem. *Parallel Computing*, 23(13):1889–1907.

Tarjan, R. E. (1979). Applications of path compression on balanced trees. *Journal of the ACM*, 26(4):690–715.

Wood, R. K. (1993). Deterministic network interdiction. *Mathematical and Computer Modeling*, 17(2):1–18.

Zenklusen, R. (2010). Matching interdiction. *Discrete Applied Mathematics*, 158(15):1676–1690.

Zenklusen, R. (2014). Connectivity interdiction. *Operations Research Letters*, 42(6-7):450 – 454.

Zenklusen, R. (2015). An $O(1)$-approximation for minimum spanning tree interdiction. In *Proceedings of 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. To appear.