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 to protect system, attack/inhibit undesirable system/process.
Minimum Spanning Tree (MST) Interdiction

Input

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

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

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

Assumption: Budget not large enough to disconnect graph.
Minimum Spanning Tree (MST) Interdiction

Input

- \( G = (V, E) \)
- \( w : E \rightarrow \mathbb{Z}_{\geq 0} \) (weights)
- \( c : E \rightarrow \mathbb{Z}_{>0} \) (costs)
- \( B \in \mathbb{Z}_{>0} \) (budget)
Minimum Spanning Tree (MST) Interdiction

Input

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

Task

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

$$\max\{ w(\text{MST}((V, E \setminus R))) \mid R \subseteq E, c(R) \leq B \}$$
Minimum Spanning Tree (MST) Interdiction

Input

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

Task

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

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

Assumption: Budget not large enough to disconnect graph.
Some special cases

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.)
Some special cases

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.)

$B = q$

<table>
<thead>
<tr>
<th>weight</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>$B + 1$</td>
</tr>
</tbody>
</table>

add unremovable spanning tree
Some special cases

Maximum Component Problem (MCP)

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

\((k\text{-}cut\ \text{problem \ where \ roles \ of \ objective \ and \ constraint \ are \ swapped.})\)

\[ B = q \]

<table>
<thead>
<tr>
<th>weight</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>( B + 1 )</td>
</tr>
</tbody>
</table>

Add unremovable spanning tree

Generalization of MCP: Budgeted Graph Disconnection (BGD)

Break graph into max # of conn. comp. by removing edges of \( \text{cost} \leq B \).
### Best known approximation ratios

<table>
<thead>
<tr>
<th></th>
<th>$O(\log m)$</th>
<th>[Frederickson and Solis-Oba, 1996]</th>
</tr>
</thead>
<tbody>
<tr>
<td>MST interdiction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k$ most vital edges</td>
<td>$O(\log k)$</td>
<td>[Frederickson and Solis-Oba, 1996]</td>
</tr>
</tbody>
</table>

$m$ : # of edges
### Best known approximation ratios

<table>
<thead>
<tr>
<th></th>
<th>$O(\log m)$</th>
<th>[Frederickson and Solis-Oba, 1996]</th>
</tr>
</thead>
<tbody>
<tr>
<td>MST interdiction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k$ most vital edges</td>
<td>$O(\log k)$</td>
<td>[Frederickson and Solis-Oba, 1996]</td>
</tr>
</tbody>
</table>

$m$ : # 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

<table>
<thead>
<tr>
<th></th>
<th>Time Complexity</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic algo</td>
<td>$O(m \cdot \alpha(m, n))$</td>
<td>[Iwano and Katoh, 1993, Chazelle, 2000, Tarjan, 1979]</td>
</tr>
<tr>
<td>Randomized algo</td>
<td>$O(m)$</td>
<td>[Iwano and Katoh, 1993, Karger et al., 1995, Dixon et al., 1992]</td>
</tr>
</tbody>
</table>

$m$: # of edges  
$n$: # of vertices  
$\alpha(\cdot, \cdot)$: inv. Ackermann function
## Best known running times for most vital edge problem

<table>
<thead>
<tr>
<th>Algo Type</th>
<th>Running Time</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic algo</td>
<td>$O(m \cdot \alpha(m, n))$</td>
<td>[Iwano and Katoh, 1993, Chazelle, 2000, Tarjan, 1979]</td>
</tr>
<tr>
<td>Randomized algo</td>
<td>$O(m)$</td>
<td>[Iwano and Katoh, 1993, Karger et al., 1995, Dixon et al., 1992]</td>
</tr>
</tbody>
</table>

- $m$: # 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.
  → 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.

\[ E = E_{-1} \cup E_0 \cup E_1 \cup \ldots \cup E_p. \]

\[ E_{-1} = \{ e \in E \mid w(e) = 0 \} \]

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

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. \( \text{MST}(V, E \setminus R) \cap E_p \neq \emptyset \).

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

\[ \Rightarrow \text{This implies that opt. value of MST interdiction } \geq 2^p. \]
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 \( A \subseteq E \):
\[
\sigma(A) := \# \text{ of conn. comp. in } (V, A).
\]
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 \( A \subseteq E \):

\[ \sigma(A) := \text{# of conn. comp. in } (V, A). \]

---

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

comp. of \( E_{-1} \)
comp. of \( E_{\leq 0} \)
comp. of \( E_{\leq 1} \)

\[ \text{val}(\text{MST}) = (5 - 1) + 1 \cdot (4 - 1) + 2 \cdot (2 - 1) = 9 \]
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 \( A \subseteq E \):

\[
\sigma(A) := \# \text{ of conn. comp. in } (V, A).
\]

\[
\text{val}(\text{MST}) = (5 - 1) + 1 \cdot (4 - 1) + 2 \cdot (2 - 1) = 9
\]

\[
\text{val}(U) := \text{val}(\text{MST}(V, E \setminus U)) = \sigma(E_{-1} \setminus U) - 1 + \sum_{i=0}^{p-1} 2^i \cdot (\sigma(E_{\leq i} \setminus U) - 1)
\]
Interpretation as multi-level BGD

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

where for \( A \subseteq E \):

\[
\sigma(A) := \# \text{ of conn. comp. in } (V, A).
\]

\[
\text{val}(\text{MST}) = (5 - 1) + 1 \cdot (4 - 1) + 2 \cdot (2 - 1) = 9
\]

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

Goal: \( \max \{ \text{val}(U) \mid U \subseteq E, c(U) \leq B \} \)
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 \( A \subseteq E \):
\[ \sigma(A) := \text{# of conn. comp. in } (V, A). \]

\[ \text{val}(\text{MST}) = (5 - 1) + 1 \cdot (4 - 1) + 2 \cdot (2 - 1) = 9 \]

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

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

\[ \max\{\sigma(E_{\leq i} \setminus U) \mid U \subseteq E, c(U) \leq B\} \text{ is BGD } \forall i. \]
Relaxation through submodular minimization

\[ \text{val}(U) = \sigma(E - 1 \cup U) - 1 + \sum_{i=0}^{p-1} 2^i \cdot (\sigma(E \leq i \cup U) - 1) \]

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

\[ \min \{ \lambda \cdot c(U) - \text{val}(U) | U \subseteq E \leq p - 1 \} \]

for varying \( \lambda \geq 0 \).

\{○, ●\}: pareto front

optimal solution

parametric front
Relaxation through submodular minimization

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

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

optimal solution

parametric front

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

\[
\min \left\{ \lambda \cdot \text{c}(U) - \text{val}(U) \mid U \subseteq E \leq p - 1 \right\}
\]

for varying \(\lambda \geq 0\).
Relaxation through submodular minimization

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

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

\[ \min \{ \lambda \cdot c(U) - \text{val}(U) \mid U \subseteq E_{\leq p-1} \} \]

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.

$$\text{val}(R) \geq \frac{1}{6} \cdot B \cdot \frac{\text{val}(U)}{c(U)}.$$
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 \( U_1, R_2 \).
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 \( U_1, R_2 \).
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 \( U_1, R_2. \)
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 $U_1, R_2$. 

val($U$) vs. $c(U)$ with $B$, $\nu^*$.
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 \( U_1, R_2 \).

Theorem
Algorithm is 7-approximation.
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

\[
\text{val}(U) = \sigma(E_{-1} \setminus U) - 1 + \sum_{i=0}^{p-1} 2^i \cdot (\sigma(E_{\leq i} \setminus U) - 1)
\]
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 \( A_0 \)

---

- \( \text{weight} = 0 \)
- \( \text{weight} = 1 \)
- \( \text{weight} = 2 \)
- \( \text{weight} = 4 \)

- \( A_{-1} \)
- \( A_{\leq 0} \)
- \( A_{\leq 1} \)
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

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

$\delta(A) \cap U_{\leq -1}$ removing these edges creates comp. $A$ in $A_{-1}$
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

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

\( U_{\leq 0}(A) \)
removing these edges
creates comp. \( A \) in \( A_{-1} \)
and all descendant comp.

\( U_{\leq i}(A) := \{e \in U_{\leq i} \mid e \text{ has at least one endpoint in } A\} \)
Will choose \((A_1, i_1), \ldots, (A_\beta, i_\beta)\) with \(\{A_j \in \mathcal{A}_j\}_j\) disjoint and set \(R = \bigcup_{j=1}^{\beta} U_{\leq i_j}(A_j)\).

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

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

<table>
<thead>
<tr>
<th>Function</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>aux. cost: $\kappa_i(A)$</td>
<td>$\Theta(c(U_{\leq i}(A)))$</td>
</tr>
<tr>
<td>aux. impact: $g_i(A)$</td>
<td>$\Theta\left(\sum_{\ell=0}^{i} 2^\ell \cdot</td>
</tr>
</tbody>
</table>

**Aux. eff.:** $\rho_i(A) = \frac{g_i(A)}{\kappa_i(A)}$. 

![Diagram showing relationships between $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_{-1}$]
Greedy selection strategy

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 efficiencies.
Average aux. efficiency of children is close to aux. eff. of parent.

\[ \text{Error term is geometrically decreasing and of size } O(2^p). \]

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).

\[ \text{Returned set has value: } \geq \frac{1}{6} \cdot B \cdot \frac{\text{val}(U)}{c(U)} \]
Many interesting computational questions around interdiction problems are still open.
## Some approximation/hardness results

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


