Analyzing Network Robustness via Interdiction Problems

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.

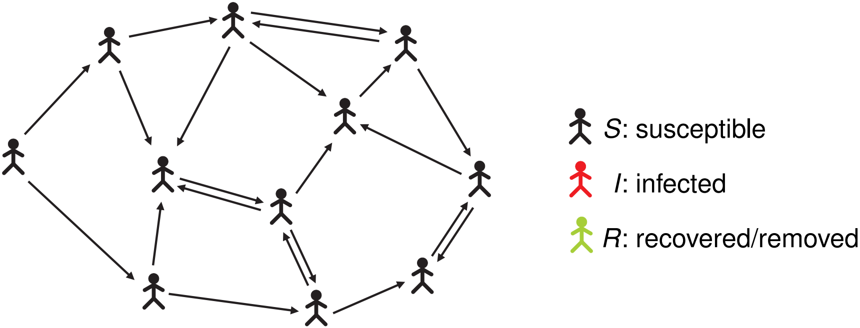
An early application: drug interdiction

One aspect of US "war on drugs"

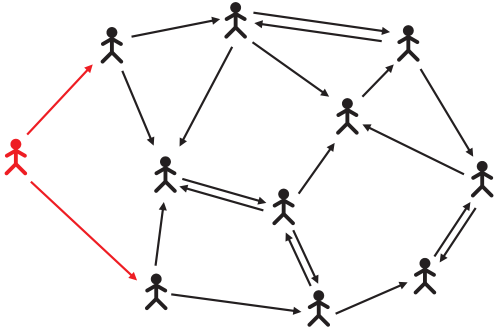
Assist South American countries to disrupt production and distribution process of drugs.

- How to achieve highest impact with limited resources?
- Good target: reduce flow of precursor chemicals.
- What rivers and roads should be checked to best interdict drug flow?

Vaccination: interdict spread of a disease



Vaccination: interdict spread of a disease

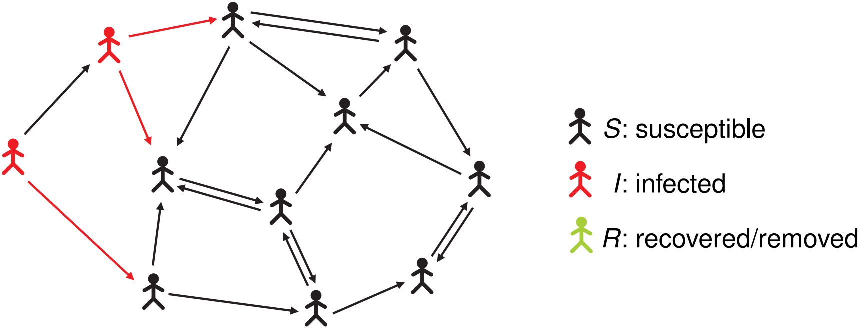


S: susceptible

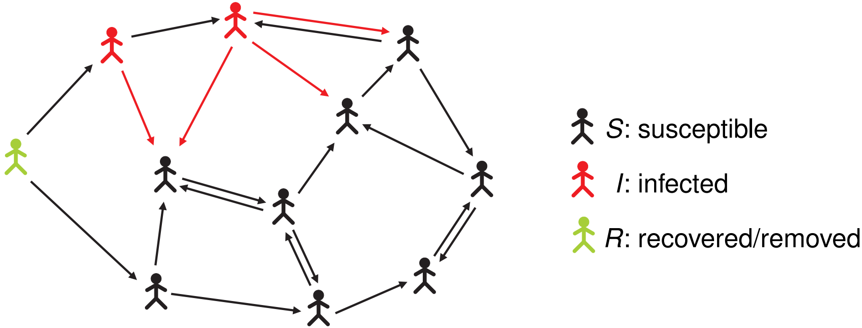
I: infected

R: recovered/removed

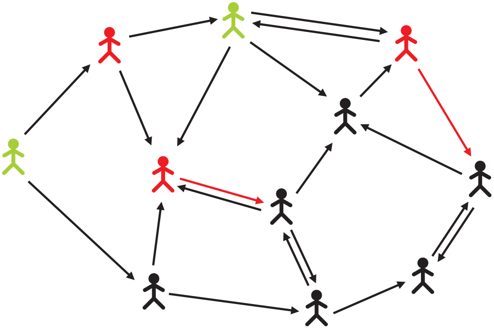
Vaccination: interdict spread of a disease



Vaccination: interdict spread of a disease



Vaccination: interdict spread of a disease

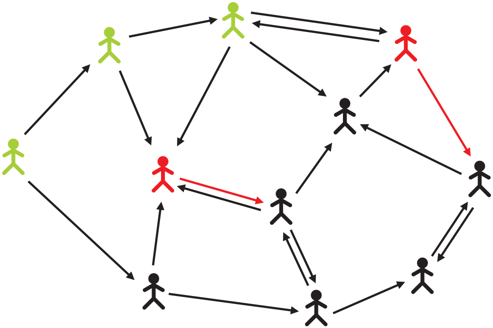


S: susceptible

I: infected

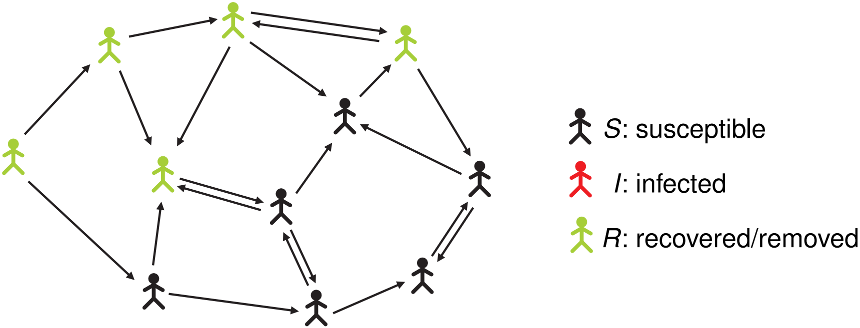
R: recovered/removed

Vaccination: interdict spread of a disease

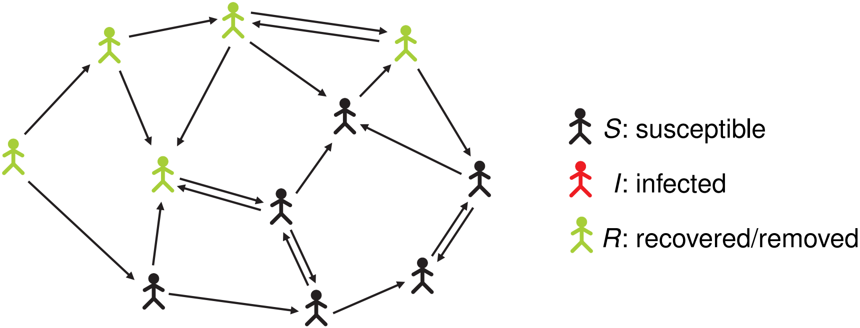


- *S*: susceptible
- *I*: infected
- *R*: recovered/removed

Vaccination: interdict spread of a disease



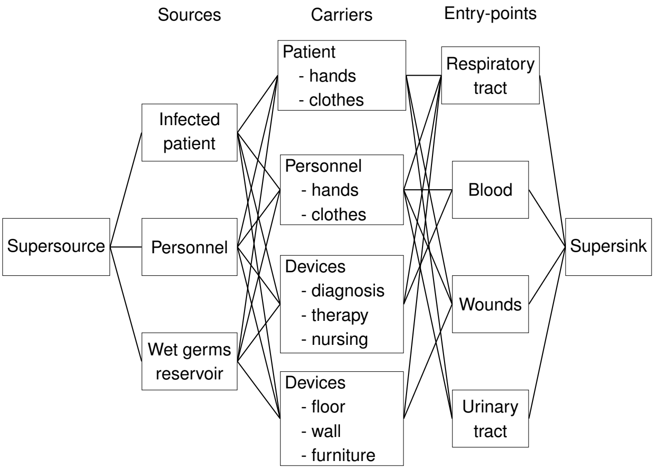
Vaccination: interdict spread of a disease



Who to vaccinate to best interdict the disease spreading process?

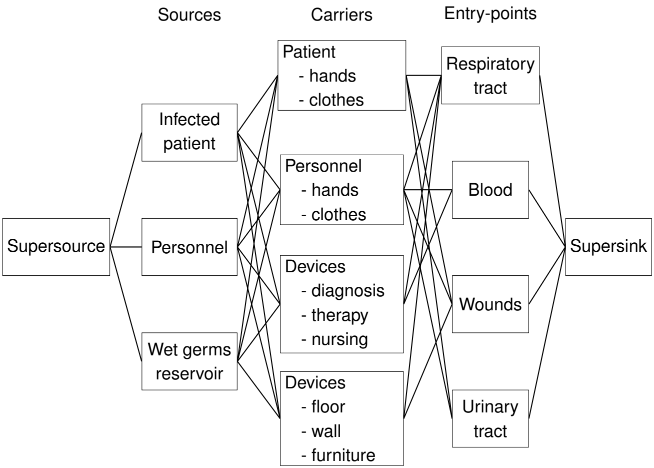
Inhibit dissimination of germs

source:Assimakopoulos [1987]



Inhibit dissimination of germs

source:Assimakopoulos [1987]



Most cost effective way to block germ dissimination?

Robustness of infrastructure

Protecting power grids against terrorists *Salmeron et al. [2004]*

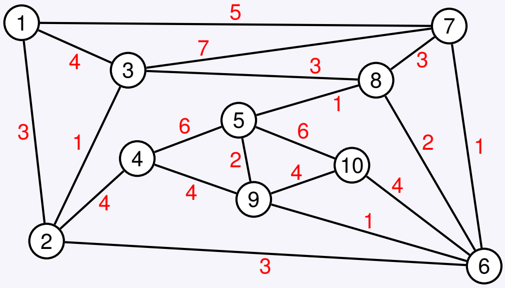
What is highest impact of potential terrorist attack with limited resouces?

Robustness of infrastructure

Protecting power grids against terrorists *Salmeron et al. [2004]*

What is highest impact of potential terrorist attack with limited resouces?

Robustness/redundancy of communication networks



- capacity

How many links/edges must fail to reduce bandwidth to < 3 for at least one pair of nodes?

Formalizing interdiction problems

e.g., max *s*-*t* flow on *G* for some fixed edge capacities

- $G = (V, E)$: network.
- $\nu(G)$: opt. value of maximization problem on G.
- $c : E \to \mathbb{Z}_{\geq 0}$: interdiction costs (cost to remove edges).
- $B \in \mathbb{Z}_{\geq 0}$: removal/attack budget.

$$c(R) := \sum_{e \in R} c(e)$$

$$\min\{\nu((V, E \setminus R)) \mid R \subseteq E, c(R) \leq B\}$$

Formalizing interdiction problems

- $G = (V, E)$: network.
- $\nu(G)$: opt. value of maximization problem on G.
  (e.g., max s-t flow on G for some fixed edge capacities)
- $c : E \to \mathbb{Z}_{\geq 0}$: interdiction costs (cost to remove edges).
- $B \in \mathbb{Z}_{\geq 0}$: removal/attack budget.

$$c(R) := \sum_{e \in R} c(e)$$

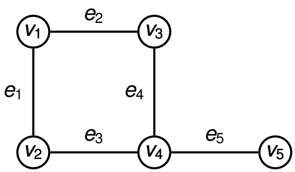
$$\min\{\nu((V, E \setminus R)) \mid R \subseteq E, c(R) \leq B\}$$

To simplify, we will sometimes assume unit removal costs.

$$\min\{\nu((V, E \setminus R)) \mid R \subseteq E, |R| \leq B\}$$

From *min max* to *min* via dualization

Consider max card. bipartite matching problem and its dual (min vertex cover).



$G = (V, E)$

$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

$$\begin{array}{lrcl} \max & \mathbf{1}^T x & & \\ & Ax & \leq & \mathbf{1} \\ & x & \geq & 0 \end{array} \qquad \begin{array}{lrcl} \min & \mathbf{1}^T y & & \\ & A^T y & \geq & \mathbf{1} \\ & y & \geq & 0 \end{array}$$

Task in unit-cost bibpartite matching interdiction

Remove B edges to minimize cardinality of max card. matching.

Dual problem: Remove B constraints in dual to minimize min card. vertex cover.

Interdiction in terms of a math. program

A primal and dual problem for bipartite matching interdiction:

$$\min_{\substack{r \in \{0,1\}^E \\ r(E) \le B}} \max \quad (\mathbf{1} - r)^T x \qquad \begin{aligned} Ax &\le \mathbf{1} \\ x &\ge 0 \end{aligned}$$

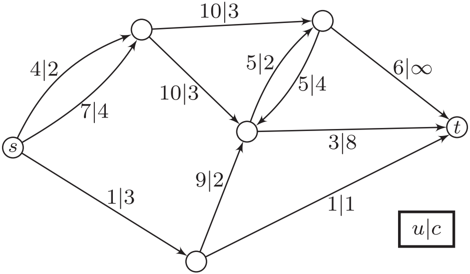
$$\begin{aligned} \min \quad & \mathbf{1}^T y \\ A^T y &\ge \mathbf{1} - r \\ \mathbf{1}^T r &\le B \\ y &\ge 0 \\ r &\in \{0,1\}^E \end{aligned}$$

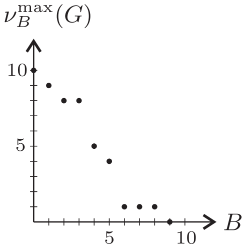
Remarks

- This formulation exploits down-monotonicity.
- Notice: Dual is TU except for constraint $1^T r \le B$.
- Resulting dual can often be interpreted as biobjective problem.

Example reduction to biobjective problem

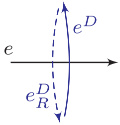
Consider maximum flow interdiction on an *s*-*t* planar graph:



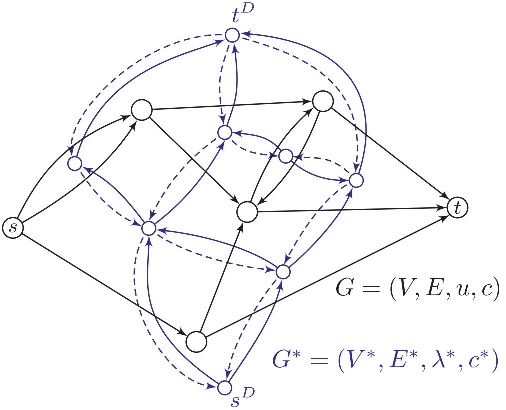


- *u*: arc capacities.
- *c*: interdiction costs.
- $\nu_B^{\max}(G)$: maximum flow value after best interdiction with budget *B*.

Dualizing using planar duality

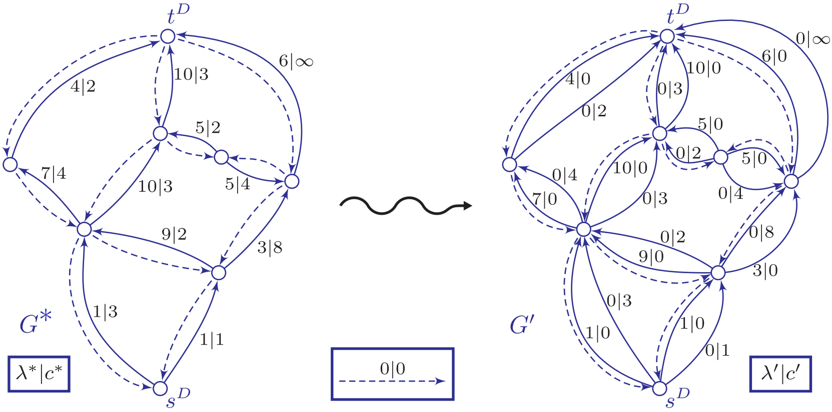


$$\lambda^*(e^D) = u(e)$$
$$c^*(e^D) = c(e)$$
$$\lambda^*(e_R^D) = 0$$
$$c^*(e_R^D) = 0$$



s-*t* cuts in primal (*G*) $\leftrightarrow$ s^D-t^D paths in planar dual (G^*).

To budgeted shortest path problem



$$\nu_B^{\max}(G) = \min\{\lambda'(P') \mid P' \text{ path from } s^D \text{ to } t^D \text{ in } G', c'(P') \leq B\}$$

- This is a budgeted shortest path problem (solvable in pseudopoly. time).

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]
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
...		

Interdiction problems have been studied under various aspects:

- Exact exponential-time algorithms.
- (Mixed-)Integer mathematical programming formulations.
- Fixed-parameter tracktability.
- ...

Part II

Connectivity Interdiction

Connectivity Interdiction

Given

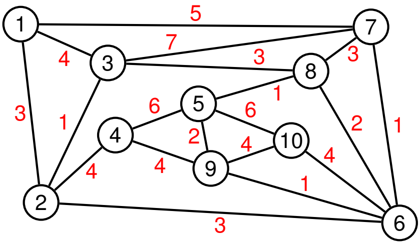
- $G = (V, E)$: undirected graph.
- $\mathrm{w} : E \to \mathbb{Z}_{\geq 0}$: edge weights (capacities).
- $B \in \mathbb{Z}_{\geq 0}$: removal/attack budget (# of edges we can remove).

$\nu(G)$: connectivity of $G =$

$$\max\{\theta \geq 0 \mid \text{between any } u, v \in V \text{ one can send } \theta \text{ units of flow}\}.$$

Goal

Minimize connectivity by removing B edges.



- w: capacities

On example of Connectivity Interdiction we present:

1. methods to approach interdiction problems,

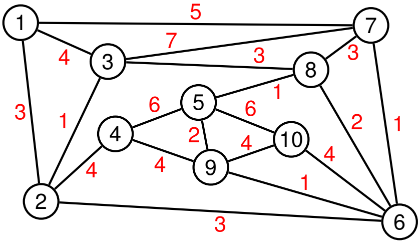
2. different viewpoints on interdiction
   $\longrightarrow$ enhance understanding of interdiction problems,

3. efficient algo to solve Connectivity Interdiction (with unit costs).

From min-max to simple min (via dualization)

Recap: max-flow min-cut theorem

For $s, t \in V$: value of max s-t flow = size of min s-t cut.



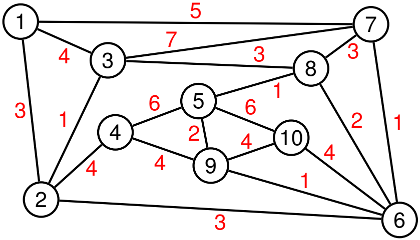
- w: capacities

From min-max to simple min (via dualization)

Recap: max-flow min-cut theorem

For $s, t \in V$: value of max s-t flow = size of min s-t cut.

$\Rightarrow$ connectivity of G: $\nu(G) =$ size of *global* min cut.



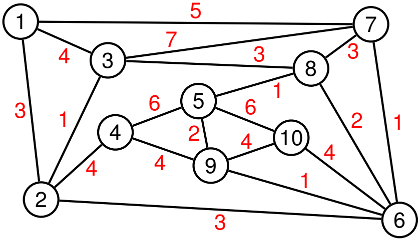
- w: capacities

From min-max to simple min (via dualization)

Recap: max-flow min-cut theorem

For $s, t \in V$: value of max s-t flow = size of min s-t cut.

$\Rightarrow$ connectivity of G: $\nu(G) =$ size of *global* min cut.



- w: capacities

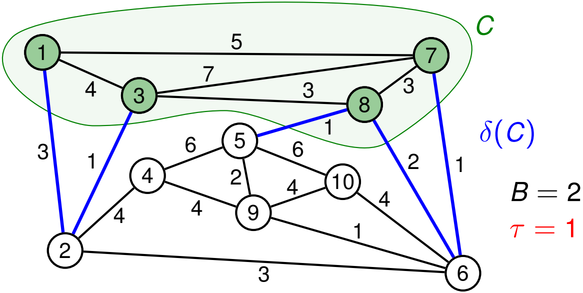
Goal of Connectivity Interdiction (rephrased)

Remove B edges to minimize size of global min cut.

Best strategy: attack single (target) cut

Different way to think about Connectivity Interdiction

Find best cut to attack, and decrease its value as much as possible.



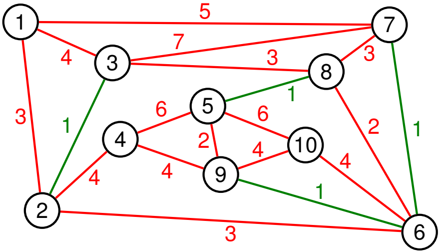
Best way to attack target cut

Once target cut $C \subseteq V$ is fixed ⟶ remove B highest cap. edges in $\delta(C)$.

$\Rightarrow \exists$ threshold weight τ s.t. all $e \in \delta(C)$ with $\mathrm{w}(e) > \tau$ will be removed.

Assume we knew optimal threshold weight τ

- Remaining task: find cut C to attack.
- For each edge $e \in E$, if e is part of the cut we attack, then
  - e get removed if $\mathrm{w}(e) > \tau$
  - e stays if $\mathrm{w}(e) \leq \tau$



$E_r = \{e \in E \mid \mathrm{w}(e) > \tau\}$

$E_s = \{e \in E \mid \mathrm{w}(e) \leq \tau\}$

$B = 2$

$\tau = 1$

Knowing τ, Connectivity Interdiction reduces to

$$\min\{\mathrm{w}(\delta(C) \cap E_s) \mid \text{ cut } C \text{ with } |\delta(C) \cap E_r| \leq B\}.$$

How to get the right τ?

Plan: simply try all "candidates" for τ

Consider all different weights (assume disjoint weights):

$$0 < \mathrm{w}(e_1) < \mathrm{w}(e_2) < \cdots < \mathrm{w}(e_m) = \text{max weight.}$$

1. For $\tau \in \{0, \mathrm{w}(e_1), \ldots, \mathrm{w}(e_m)\}$: Let
   - $E_s = \{e \in E \mid \mathrm{w}(e) \leq \tau\}$,
   - $E_r = \{e \in E \mid \mathrm{w}(e) > \tau\}$.

   will discuss later how to solve this

   Compute optimal solution C_τ for

   $$\min\{\mathrm{w}(\delta(C) \cap E_s) \mid \text{ cut } C \text{ with } |\delta(C) \cap E_r| \leq B\}. \qquad (1)$$

2. Attack best cut C_τ among all computed cuts.

How to get the right τ?

Plan: simply try all "candidates" for τ

Consider all different weights (assume disjoint weights):

$$0 < \mathrm{w}(e_1) < \mathrm{w}(e_2) < \cdots < \mathrm{w}(e_m) = \text{max weight.}$$

1. For $\tau \in \{0, \mathrm{w}(e_1), \ldots, \mathrm{w}(e_m)\}$: Let
   - $E_s = \{e \in E \mid \mathrm{w}(e) \leq \tau\}$,
   - $E_r = \{e \in E \mid \mathrm{w}(e) > \tau\}$.

   will discuss later how to solve this

   Compute optimal solution C_τ for

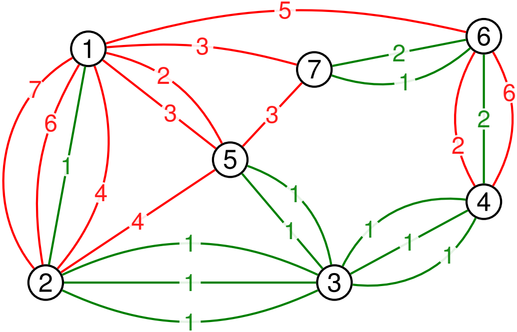
   $$\min\{\mathrm{w}(\delta(C) \cap E_s) \mid \text{ cut } C \text{ with } |\delta(C) \cap E_r| \leq B\}. \qquad (1)$$

2. Attack best cut C_τ among all computed cuts.

- Above procedure solves problem by returning best cut to interdict.
- Its runtime: $(m+1)\times$ (time to solve (1)).

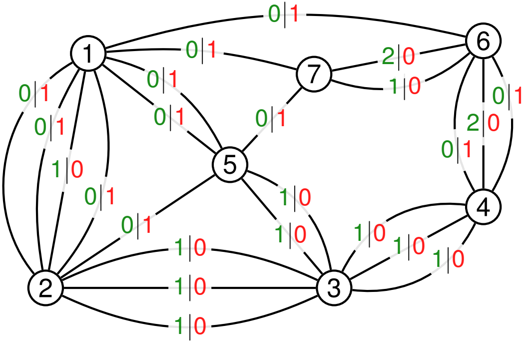
Transformation to budgeted cut problem

$$\min\{\mathrm{w}(\delta(C) \cap E_s) \mid \text{ cut } C \text{ with } |\delta(C) \cap E_r| \leq B\}.$$



Transformation to budgeted cut problem

$$\min\{\mathrm{w}(\delta(C) \cap E_s) \mid \text{ cut } C \text{ with } |\delta(C) \cap E_r| \leq B\}.$$



Introduce 2 lengths ℓ_1, ℓ_2 for each edge e:

$$\ell_1(e) = \begin{cases} \mathrm{w}(e) & \text{if } e \in E_s, \\ 0 & \text{if } e \in E_r. \end{cases}$$

$$\ell_2(e) = \begin{cases} 0 & \text{if } e \in E_s, \\ 1 & \text{if } e \in E_r. \end{cases}$$

Problem transforms into: $\min\{\ell_1(\delta(C)) \mid \text{cut } C \text{ with } \ell_2(\delta(C)) \leq B\}.$

↑ this is a budgeted cut problem

A closer look at budgeted cuts

Given

- Graph $G = (V, E)$,
- edge lengths $\ell_1, \ell_2 : E \to \mathbb{Z}_{\geq 0}$,
- budget $B \in \mathbb{Z}_{\geq 0}$.

Task (budgeted cut problem)

$$\min \left\{ \ell_1(\delta(C)) \;\middle|\; \begin{array}{l} \text{cut } C \text{ with} \\ \ell_2(\delta(C)) \leq B \end{array} \right\}$$

A closer look at budgeted cuts

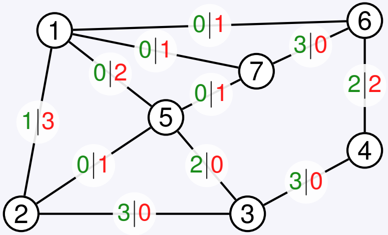
Given

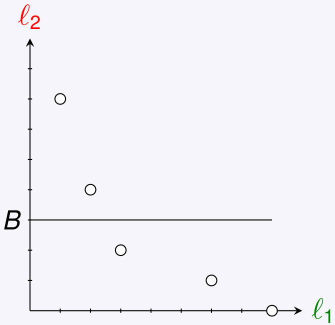
- Graph $G = (V, E)$,
- edge lengths $\ell_1, \ell_2 : E \to \mathbb{Z}_{\geq 0}$,
- budget $B \in \mathbb{Z}_{\geq 0}$.

Task (budgeted cut problem)

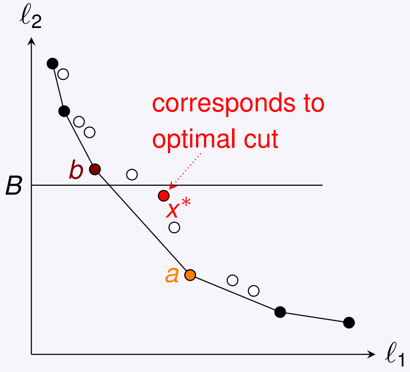
$$\min \left\{ \ell_1(\delta(C)) \;\middle|\; \begin{array}{l} \text{cut } C \text{ with} \\ \ell_2(\delta(C)) \leq B \end{array} \right\}$$

A biobjective viewpoint on budgeted cuts





How to solve the budgeted cut problem?



- Solid points are min-cuts for some *mixed length* $\lambda\ell_1 + (1-\lambda)\ell_2$.
- There is a mixed length ℓ^* for which both a and b are optimal.

ℓ^*-size of $x^* \leq 2 \times \min \ell^*$-size.

i.e., x^* is a 2-minimum cut wrt ℓ^*

Suffices to find all 2-min cuts wrt ℓ^* to solve budgeted cut problem.

Back to single objective problem.

Can be solved efficiently!

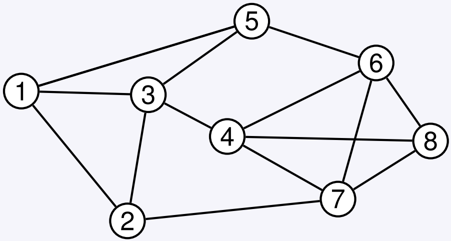
Let's start with easier prob.: How to find a min cut in a unit-weighted graph?

A randomized approach *Karger [1993]*

- Repeat following step until 2 (super-)vertices are left.

  Choose an edge uniformly at random and contract it.

- Return cut defined by remaining 2 vertices.

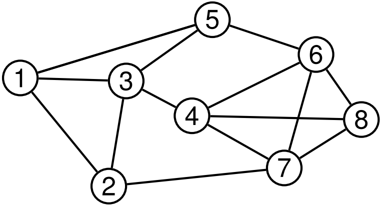


How likely is it to get some min cut that way

- Algo returns $C \Leftrightarrow$ no edge of $\delta(C)$ ever gets contracted.

Probab.: 1st contr. edge e_1 is in $\delta(C)$

- min cut size = $k \;\Rightarrow\;$ min degree $\geq k$ $\;\Rightarrow\; m \geq k \cdot n/2$.
- $\Pr[e_1 \in \delta(C)] = |\delta(C)|/m \leq 2/n$.



What about later edge contractions?

When contracting i-th edge e_i: only $n+1-i$ (super-)vertices left:

min deg. $\geq k \;\Rightarrow\; m \geq k \cdot (n+i-i)/2 \;\Rightarrow\; \Pr[e_i \in \delta(C)] \leq 2/(n+1-i)$.

$$\Pr[\text{algo returns } \delta(C)] = \prod_{i=1}^{n-2}(1 - \Pr[e_i \in \delta(C)]) \geq \prod_{i=1}^{n-2} \frac{n-1-i}{n+1-i}$$
$$= 2/n(n-1) \geq 1/n^2.$$

How to get 2-min cuts from that?

Summary of randomized procedure

A single run of algo returns any min cut C with probability $\geq 1/n^2$.

Implications

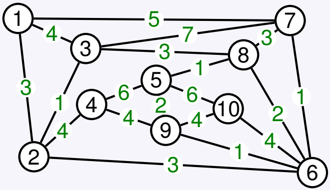
- There are at most n^2 min cuts.
- $n^2 \log n$ repetitions of algo finds all min cuts with prob. $\approx 1 - 1/n$.

Getting all 2-min cuts

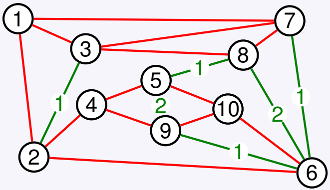
- Algorithm generalizes to general edge weights.
  (Contract edge with prob. proportional to its weight.)
- Analysis extends to 2-min cuts: any 2-min cut is returned with prob. $1/n^4$.
- $n^4 \log n$ repetitions of algo finds all 2-min cuts with prob. $\approx 1 - 1/n^4$.

Summary of approach for conn. interd.

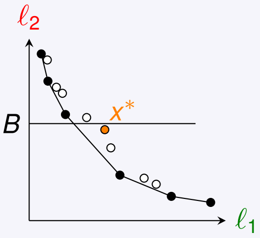
Dualize → minimize min cut



Guess removal threshold τ



Budgeted min cut problem



Karger's randomized algo

- Interdiction problems have many interesting applications.
- Dualization: min-max problem → min-min problem.
- Threshold guessing: transformation to biobjective problem.

Special case: Connectivity Interdiction with unit removal costs

Can be rephrased as budgeted cut problem, which is efficiently solvable.

References I

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

Assimakopoulos, N. (1987). A network interdiction model for hospital infection control. *Computers in Biology and Medicine*, 17(6):413–422.

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.

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.

Frederickson, G. N. and Solis-Oba, R. (1996). Increasing the weight of minimum spanning trees. In *Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 539–546, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Karger, D. R. (1993). Global min-cuts in RNC, and other ramifications of a simple min-out algorithm. In *Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 21–30.

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.

Phillips, C. A. (1993). The network inhibition problem. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC)*, pages 776–785.

Salmeron, J., Wood, K., and Baldick, R. (2004). Analysis of electric grid security under terrorist thread. *IEEE Transaction on Power Systems*, 19(2):905–912.

References II

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)*, pages 709–728.