Tri-level System Defense

Eitan Israeli
Kevin Wood kwood@nps.navy.mil
Operations Research Department
Naval Postgraduate School Oct. 2002

Purpose

- **Describe models for efficiently interdicting a general “system” belonging to an adversary, e.g., an economy, a segment of an economy, energy production, munitions production, a nuclear program**
- **Show how to extend those bi-level interdiction models to tri-level system defense (interdict the interdictor)**

Assumptions

- Our adversary wishes to maximize the output of his system
- We wish to minimize that output by “interdicting” parts of his system
- We have limited interdiction resources
- Interdiction: attack, sabotage, embargo, litigate, freeze funds
- Interdiction stops a single “activity” of the adversary (partial interdiction possible, too)

Minimize the max output of a system modeled as a linear program

$$\min_{\mathbf{x} \in X} \max_{\mathbf{y}} \ \mathbf{cy}$$

$$\text{s.t.} \quad A\mathbf{y} \le \mathbf{b}$$

$$0 \le y_k \le u_k (1 - x_k) \quad \forall k$$

$\mathbf{x} \equiv$ binary interdiction variables

$X \equiv$ interdiction resource constraints and binary restrictions

Tri-level System Defense

$$\max_{\mathbf{w}\in W} \min_{\mathbf{x}\in X} \max_{\mathbf{y}} \ \mathbf{cy}$$

$$\text{s.t.} \quad A\mathbf{y} \le \mathbf{b}$$

$$0 \le y_k \le u_k \left(1 - x_k (1 - w_k)\right) \quad \forall k$$

Let w_k = 1 if we harden activity k against attack
= 0 otherwise

W **are constraints on defense (hardening) resources, and the fact that the w_k are binary**

Tri-level system defense: Interdict the interdictor

$$\max_{\mathbf{w}\in W} \min_{\mathbf{x}\in X} \max_{\mathbf{y}} \ \mathbf{cy}$$

$$\text{s.t.} \quad A\mathbf{y} \le \mathbf{b}$$

$$0 \le y_k \le u_k \left(1 - x_k (1 - w_k)\right) \quad \forall k$$

$$\max_{\mathbf{w}\in W} \min_{\mathbf{x}\in X} \ f(\mathbf{x})$$

$$\text{s.t.} \quad x_k \le 1 - w_k \quad \forall k$$

Back to bi-level interdiction

$$\min_{\mathbf{x} \in X} \max_{\mathbf{y}} \ \mathbf{cy}$$

$$\text{s.t.} \ \ A\mathbf{y} \le \mathbf{b}$$

$$0 \le y_k \le u_k (1 - x_k) \quad \forall k$$

Can take the dual of the inner maximization and get a nonlinear "min min model" = "min model." Reformulate to get a linear model.

Let π_k'' be an upper bound on the value to the adversary of one unit of capacity for activity *k* (upper bnd on dual variables for capacity constraints)

Convex reformulation

$$\min_{\mathbf{x} \in X} \max_{\mathbf{y}} \ \mathbf{cy} - \sum_k \pi''_k x_k y_k$$

$$\text{s.t.} \ \ A\mathbf{y} \leq \mathbf{b}$$

$$\mathbf{0} \leq \mathbf{y} \leq \mathbf{u} \quad \forall k$$

Conversion to a Mixed Integer Program (MIP)

$$\min_{\mathbf{x},\boldsymbol{\alpha},\boldsymbol{\pi}} \quad \mathbf{b}^T\boldsymbol{\alpha} + \mathbf{u}^T\boldsymbol{\pi}$$

$$\text{s.t.} \quad A^T\boldsymbol{\alpha} + I\boldsymbol{\pi} + \Pi''\mathbf{x} \geq \mathbf{c}^T$$

$$\mathbf{x} \in X$$

Where Π'' is a diagonal matrix of those upper bounds π'' on the dual variables

That's nice

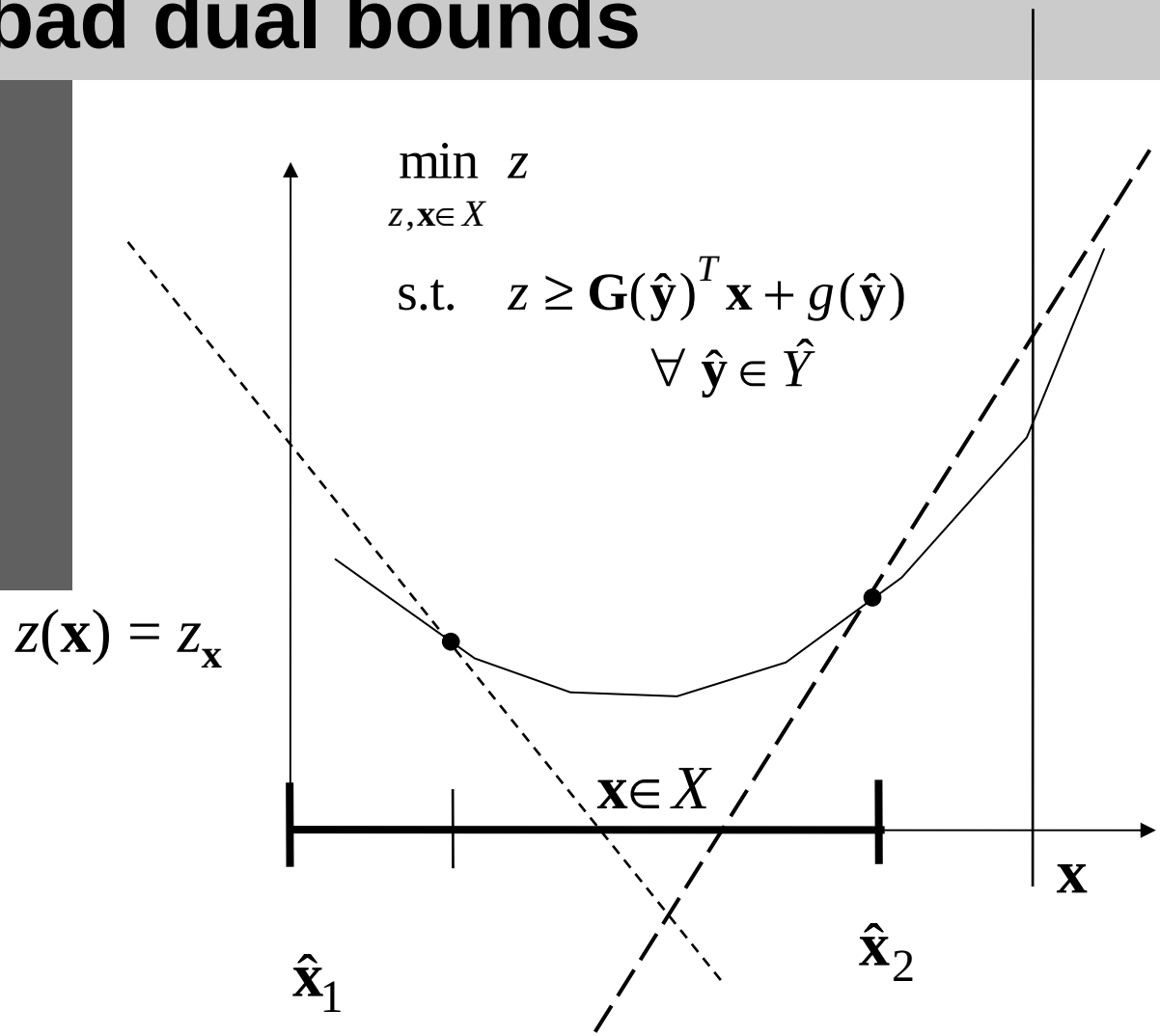
- Can we solve the problem?
- Theoretically, yes. In practice, maybe.
- Benders decomposition is a useful approach
- But, if those bounds are weak, it won't work very well: You can't just guess a huge number for the π_k''

Benders decomposition with bad dual bounds

$$\min_{z,\mathbf{x}\in X} \; z$$

$$\text{s.t.} \quad z \geq \mathbf{G}(\hat{\mathbf{y}})^T\mathbf{x} + g(\hat{\mathbf{y}})$$

$$\forall\ \hat{\mathbf{y}} \in \hat{Y}$$



Will still work…in theory

Better off with a “covering decomposition” when appropriate

- It will be OK if the total number of activities that the adversary can engage in at one time is small, at least in comparison to the total number of activities he might engage in
- For instance: The adversary wants to build 5 power plants and has 50 possible locations; the interdictor’s task is minimize power production by making some subset of the locations untenable

Covering Decomposition for the Power-Plant Example

Algorithm 2: Covering decomp for EPPs;

Output: An optimal interdiction plan;

Step 0: $x \leftarrow 0$;

Step 1: Given interdiction plan x, find the adversary's best set of plant locations $\hat{y}$ and obj value z_x; Save "siting plan" $\hat{y}$; (Note: IP)

If x is the best interd'n plan so far, save it;

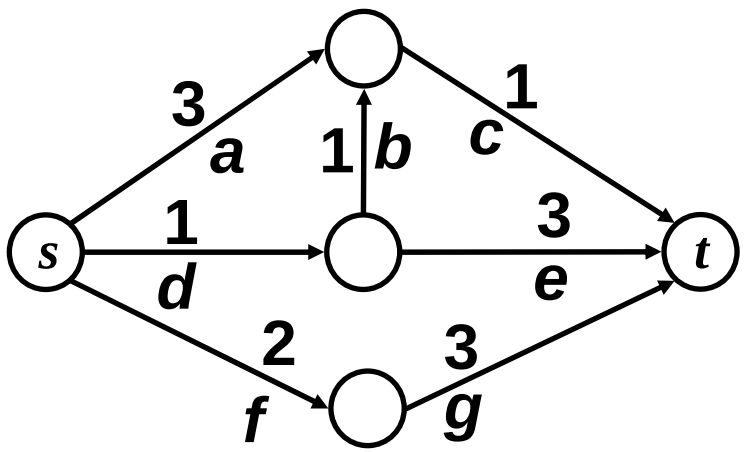
Step 2: Try to "cover" all the siting plans $\hat{y}$ seen so far with a new interdiction plan x.

If you can cover them, go to Step 1 with x;

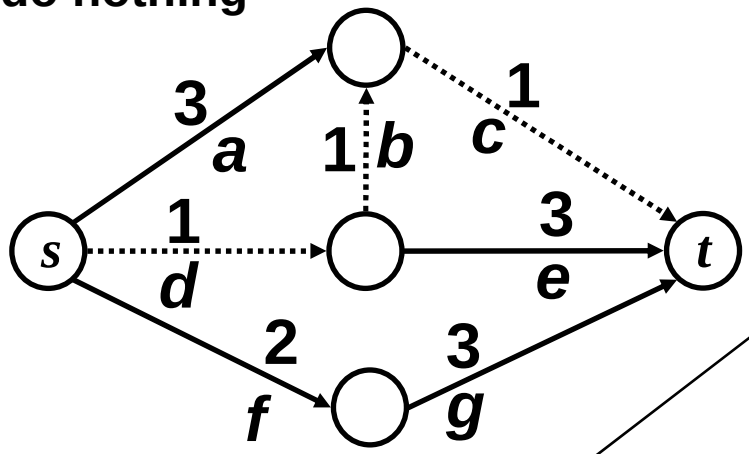
Step 3: Otherwise: Print the best interdiction plan found and Stop; /* best x is optimal */

Covering Decomp: Example

- **Shortest-path interdiction example (max-min, not min-max, but…)**
  - Each arc marked with label & length
  - Interdiction destroys arc completely
  - Can interdict only two arcs



First interdiction plan: x=0, do nothing

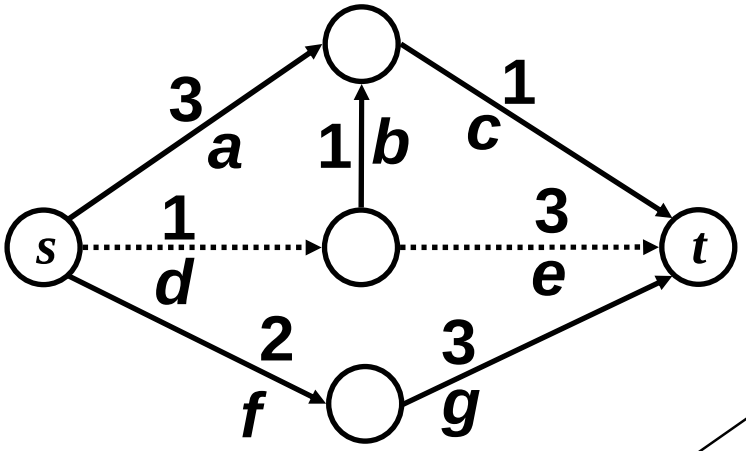


NU response: Take path *d*,*b*,*c*; length = 3

Yields this covering problem:

a	b	c	d	e	f	g
1	1	1				

2nd interdiction plan: interdict b and c

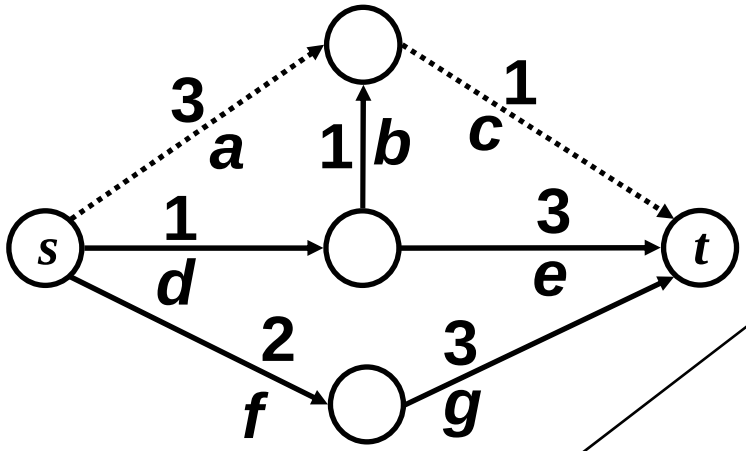


NU response: Take path *d*,*e*; length = 4

Yields this covering problem:

a	b	c	d	e	f	g
1	1	1				
			1		1	

3rd interdiction plan: interdict *d* and *e*

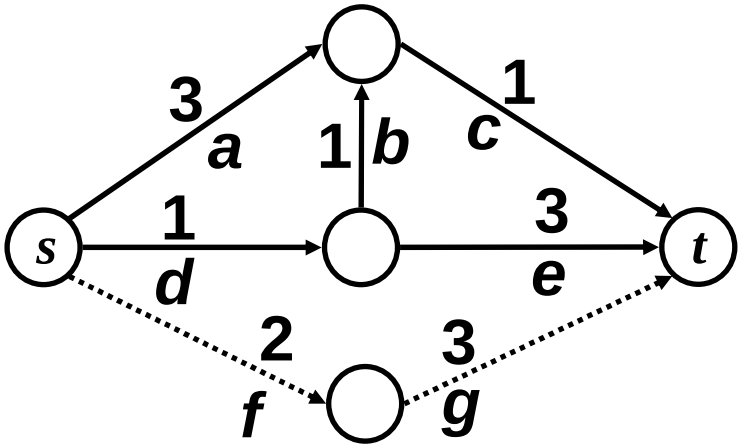


NU response: Take path *a*,*c*; length = 4

Yields this covering problem:

a	b	c	d	e	f	g
	1	1	1			
			1		1	
1		1				

4th interdiction plan: interdict *a* and *d*



NU response: Take path *f*,*g*; length = 5

Yields this covering problem:

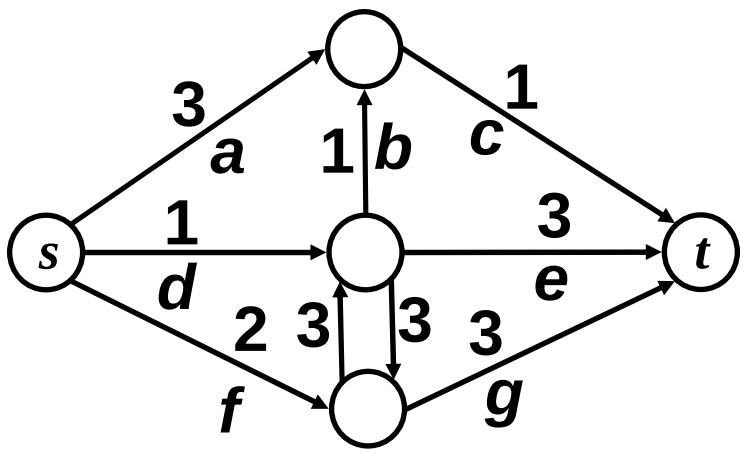
a	b	c	d	e	f	g
	1	1	1			
			1		1	
1		1				
					1	1

INFEASIBLE!

Last interdiction plan *a*, *d* was best

A note about the example

- It is a simple example and we looked at every path in *G*
- But the problem below has the same solution and we would not enumerate all paths: The method can be "efficient."



Tri-level System Defense

- Suppose the enemy has an interdiction model of our economy, or a regional power network in the US, or NMCI or ??
- How do we harden our systems against potential (interdiction) attack?
- Ad hoc solution: Study the results of interdiction models to look for critical components in our systems.
- Formal solution: Interdict the interdictor

Tri-level System Defense

$$\max_{\mathbf{w}\in W} \min_{\mathbf{x}\in X} \max_{\mathbf{y}} \ \mathbf{cy}$$

$$\text{s.t.} \quad A\mathbf{y} \le \mathbf{b}$$

$$0 \le y_k \le u_k \left(1 - x_k (1 - w_k)\right) \quad \forall k$$

Let w_k = *1* if we harden activity *k* against attack
= 0 otherwise

W **are constraints on defense (hardening) resources, and the fact that the w_k are binary**

System Defense: Interdict the interdictor

$$\max_{\mathbf{w}\in W} \min_{\mathbf{x}\in X} \max_{\mathbf{y}} \ \mathbf{cy}$$

$$\text{s.t.} \quad A\mathbf{y} \le \mathbf{b}$$

$$0 \le y_k \le u_k \left(1 - x_k (1 - w_k)\right) \quad \forall k$$

$$\max_{\mathbf{w}\in W} \min_{\mathbf{x}\in X} \ f(\mathbf{x})$$

$$\text{s.t.} \quad x_k \le 1 - w_k \quad \forall k$$

Covering decomposition for tri-level system defense

Algorithm 2: Interdicting the interdictor;

Output: An optimal defense plan;

Step 0: $w \leftarrow 0$;

Step 1: For defense plan w, find the adversary's best interdiction plan $\hat{x}$ and obj value *z*;

If w is the best defense plan so far, save it;

Step 2: Try to "cover" all the interdiction plans $\hat{x}$ seen so far with a new defense plan w.

If you can cover them, go to Step 1 with w;

Step 3: Otherwise: Print the best interdiction plan found and Stop; /* best w is optimal */

Computational results: 10×10 grid

RS6000, Model 595, about 400 1st-stage vars., defend/interdict 5 arcs

			Alg. NA-1				Alg.NA-2				
Prob.	h_0	r_0	T_1	S_1	N_1^D	N_1^t	T_2	S_2	N_2^D	N_i^t	P_i^D
1D	10	20	325	60	40	140	33	17	65	1210	155
2D	15	20	475	175	72	180	75	70	135	1670	305
3D	10	30	–	–	–	–	440	265	155	3375	535

T_i Running time in CPU seconds for Algorithm NA-*i*.

S_i Standard deviation in CPU seconds of T_i

N_i^D Number of iterations in the master defense problem in NA-*i*

N_i^t Total number of iterations in the master problem of the sub-problem (the interdiction problem) in NA-i.

P_i^D Number of interdiction plans (i.e., cuts) in master defense problem in NA-2.