Some Network Interdiction Problems

Kevin Wood kwood@nps.navy.mil
Operations Research Department
Naval Postgraduate School

Purpose of this talk

- Describe some network interdiction problems of interest to DoD
- Give an overview of solution technologies

Generic network interdiction problem

- Using limited resources, attack an adversary’s network so as to minimize the functionality of that network (to the adversary).
- Networks: Road, pipeline, comm
- Functionality: Max flow, shortest path, convoy movement, path existence
- Attacks: Aerial sorties, cruise missiles, special ops, interception
- Can generalize: “system interdiction”

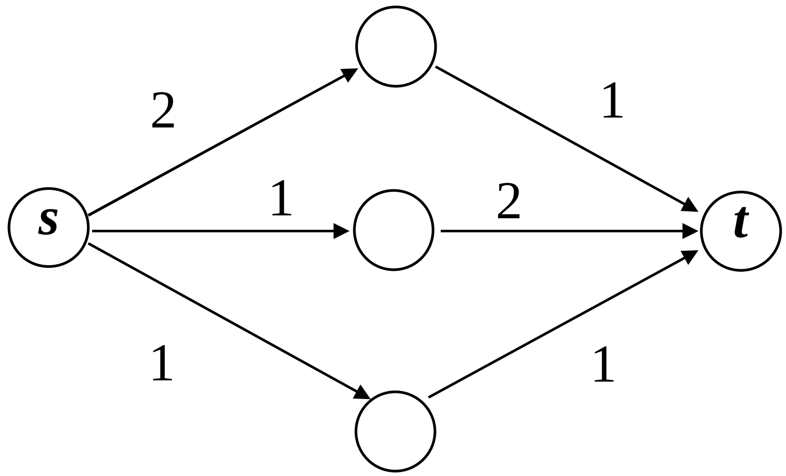
Simple network interdiction

- **Given network G=(*N*,*A*) and resource r_k required to "destroy" *k*=(*i,j*), find the minimum total resource required to cut all comm between nodes *s* and *t* .**
- **Simple solution via the max-flow min-cut theorem:**

  Set resources as arc capacities, find max flow and min capacity cut.

Max-flow, min-cut

(With multiple solutions)



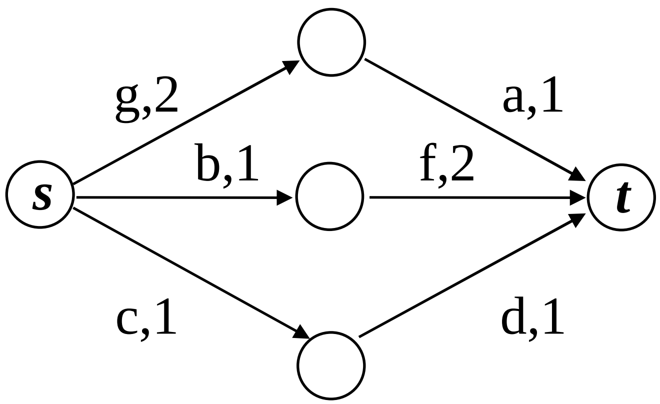
One min cut easy to find

- **But there may be secondary considerations, e.g., collateral damage, logical constraints**
- **So, find <u>all</u> <u>min</u> <u>cuts</u> and evaluate against other relevant criteria**
- **How to enumerate?**
  - Brute force: Enumerate <u>all</u> <u>cuts</u>
  - Some theoretical work in literature
  - Practical: Norm Curet, Applied Math, NSA

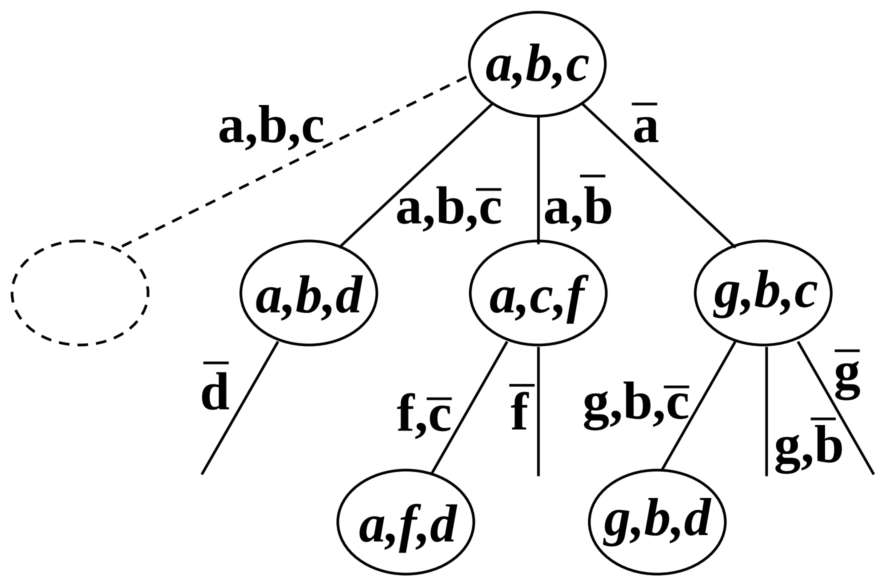
The next refinement

- **Allow near-optimal solutions, i.e., accept near-min cuts.**
- **Can still enumerate all cuts!**
- **Some graph theoretical work: Partition *G* via *G*e* and *G-e.***
- **New work described here**

Our “partitioning” scheme



Backtrack if +1 tolerance exceeded



An Algorithm

MAIN

Input: $G = (N, A)$, $\underline{u}$, ε, s, t

Output: All cuts with cap. $\leq (1 + \varepsilon)\, z_{\min}$

{

$(z_{\min}, A_C) \leftarrow$ Maxflow $(G, s, t, \underline{u})$;

$\mathit{fixed}_k \leftarrow$ false $\forall\, k \in A$ // No caps. fixed

Enumerate($z_{\min}$, $\underline{u}$, $\underline{\mathit{fixed}}$, G, s, t, ε);

}

Algorithm: recursive routine

Enumerate($z_{\min}$, $\underline{u}$, $\underline{fixed}$, G, s, t, ε)

{ (z', A_C) $\leftarrow$ Maxflow (G, s, t, $\underline{u}$);

If ($z' > (1 + \varepsilon)\times z_{\min}$) return;

If(A_C contains all fixed k) Print (z', A_C);

For (each $k \in A_C$ with $fixed_k$ == false){

$u_{save} \leftarrow u_k$; $u_k \leftarrow \infty$;

Enumerate($z_{\min}$, $\underline{u}$, $\underline{fixed}$, G, s, t, ε);

$u_k \leftarrow u_{save}$; $fixed_k \leftarrow$ true;

}

return;

}

What's really going on?

- **Attempting to enumerate near-optimal bfss to the max-flow dual LP**
- **Primal:**

$$\max_{\mathbf{y}} \; y_a$$

$$\text{s.t.} \sum_{k \in FS(i)} y_k - \sum_{k \in RS(i)} y_k = 0 \quad \forall i \in N$$

$$0 \le y_k \le u_k \qquad \forall k \in A$$

- **Dual:**

$$\min_{\theta,\pi} \sum_k u_k \theta_k$$

$$\pi_i - \pi_j + \theta_k \ge \begin{cases} 1 & \text{if } k = a \\ 0 & \forall k \ne A \end{cases}$$

$$\theta_k \ge 0 \qquad \forall k \in A$$

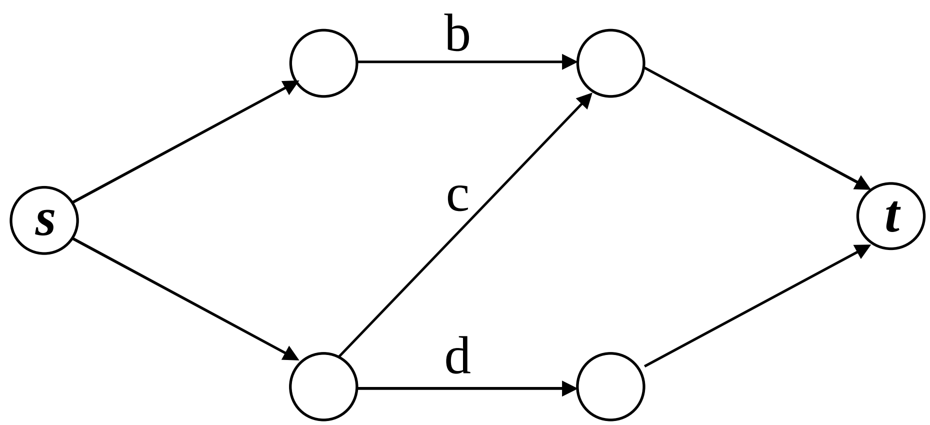
Learning from the dual LP

- **Dual LP**

$$\min_{\theta,\pi} \sum_k u_k \theta_k$$

$$\text{s.t. } \pi_i - \pi_j + \theta_k \geq \begin{cases} 1 & \text{if } k = a \\ 0 & \forall k \neq A \end{cases}$$

$$\theta_k \geq 0 \qquad \forall k \in A$$



Enhancements and results

- **Don't solve max flows from scratch**
- **Results (Java on a laptop)**
  - 2,500 nodes, 10,000 arcs, all caps. 1
  - All 98 min capacity cuts in 9 secs.
  - All 123 min_cap+5% cuts in 80 secs.
  - All 723 min_cap+10% cuts in 1.25 hrs.
  - Reoptimization critical to speed
  - But, a faster max-flow code wouldn't hurt
  - Fathoming rules need work
  - Java OK, but …

Change gears: Min max flow s.t. interdict'n resource constraints

$$\min_{\mathbf{x}\in X} \max_{\mathbf{y}} \; y_a$$

$$\text{s.t.} \sum_{k\in FS(i)} y_k - \sum_{k\in RS(i)} y_k = 0 \qquad \forall i \in N$$

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

x ≡ binary "interdiction variables"

X ≡ interdiction resource constraints and binary restrictions

Solving a min-max problem

- **Take the dual of the inner max**

$$\min_{\mathbf{x}\in X} \min_{\boldsymbol{\pi},\boldsymbol{\theta}} \sum_{k\in A} u_k (1-x_k)\theta_k$$

$$\pi_i - \pi_j + \theta_k \geq \begin{cases} 1 & \text{if } k = a \\ 0 & \forall k \in A - a \end{cases}$$

$$\theta_k \geq 0 \qquad \forall k \in A$$

- **Nonlinear! Need to reformulate**

Reformulation

- **"Primal"**

$$\min_{\mathbf{x}\in X} \max_{\mathbf{y}} \; y_a - \sum_{k\in A} x_k y_k$$

$$\text{s.t.} \sum_{k\in FS(i)} y_k - \sum_{k\in RS(i)} y_k = \begin{cases} 1 & \text{if } k = a \\ 0 & \forall k \in A - a \end{cases}$$

$$0 \le y_k \le u_k \qquad \forall k \in A$$

- **"Dual"**

$$\min_{\mathbf{x}\in X} \min_{\pi,\theta} \sum_{k\in A} u_k \theta_k$$

$$\text{s.t. } \pi_i - \pi_j + \theta_k + x_k \ge \begin{cases} 1 & \text{if } k = a \\ 0 & \forall k \in A - a \end{cases}$$

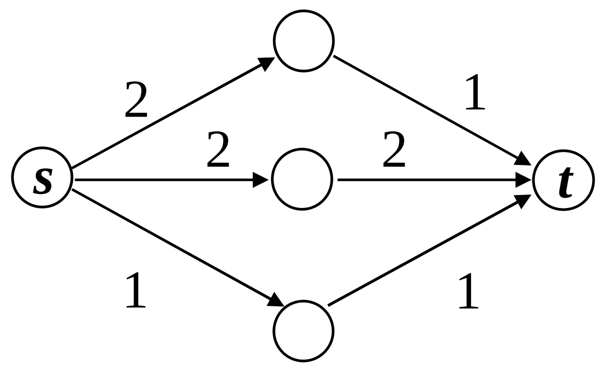
All vars. binary

Interpretation of new model

$$\min_{\mathbf{x}\in X} \min_{\pi,\theta} \sum_{k\in A} u_k \theta_k$$

$$\text{s.t. } \pi_i - \pi_j + \theta_k + x_k \geq \begin{cases} 1 & \text{if } k = a \\ 0 & \forall k \in A - a \end{cases}$$

All vars. binary



Shortest-path interdiction

Basic Model

$$\max_{\mathbf{x}\in X} \min_{\mathbf{y}} \sum_{k\in A} (c_k + x_k d_k) y_k$$

$$\text{s.t.} \sum_{k\in FS(i)} y_k - \sum_{k\in RS(i)} y_k = \begin{cases} 1 & \text{if } i\text{=s} \\ 0 & \text{if } i \in N - s - t \\ -1 & \text{if } i = t \end{cases}$$

$$y_k \ge 0 \qquad \forall k \in A$$

Converts to

$$\max_{\mathbf{x}\in X} \max_{\boldsymbol{\pi}} \pi_s - \pi_t$$

$$\text{s.t.} \quad \pi_i - \pi_j - d_k x_k \le c_k \quad \forall\, k \in A$$

Computational issues

- **If arcs are destroyed, is $d_k = \infty$ OK?**
- **Efficient solution procedures**
  - Not IP unless d_k are small
  - Decomposition uses fast sp subproblems
    - Benders decomposition
    - Set-covering decomposition (or hybrid)
- **Extensions**
  - Handling uncertainty
  - System interdiction

Interdiction under uncertainty

- **Uncertain success or data**
- **Approach: Extend Benders decomposition with cuts estimated by Monte Carlo sampling**
- **Another view of (say) the sp model**

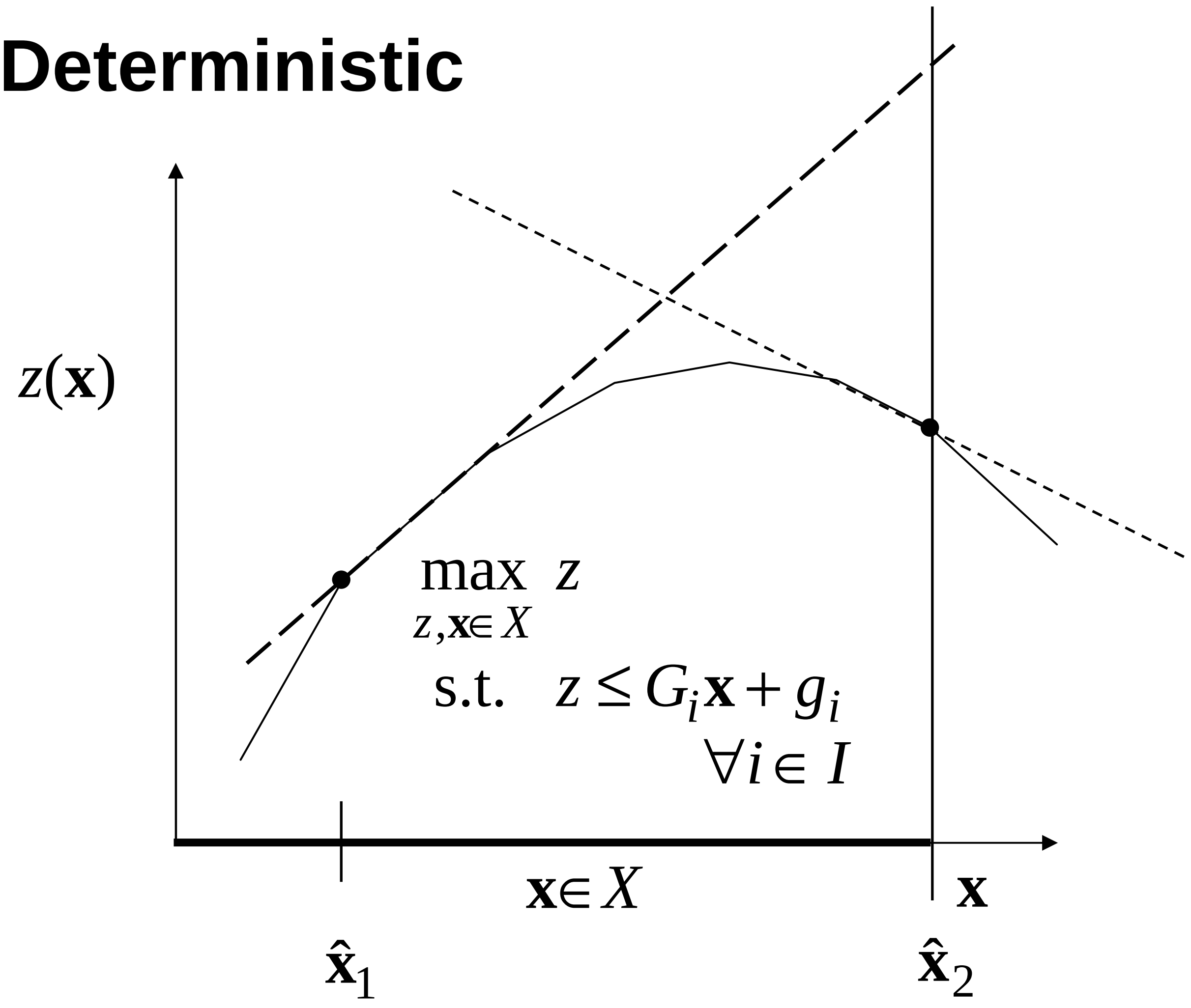
$$\max_{\mathbf{x} \in X} z(\mathbf{x})$$

$$\text{where} \quad z(\mathbf{x}) \equiv \min_{\mathbf{y}} \sum_{k \in A} (c_k + x_k d_k) y_k$$

$$\text{s.t. Flow bal. } \textsc{and } \; y_k \geq 0 \;\; \forall k$$

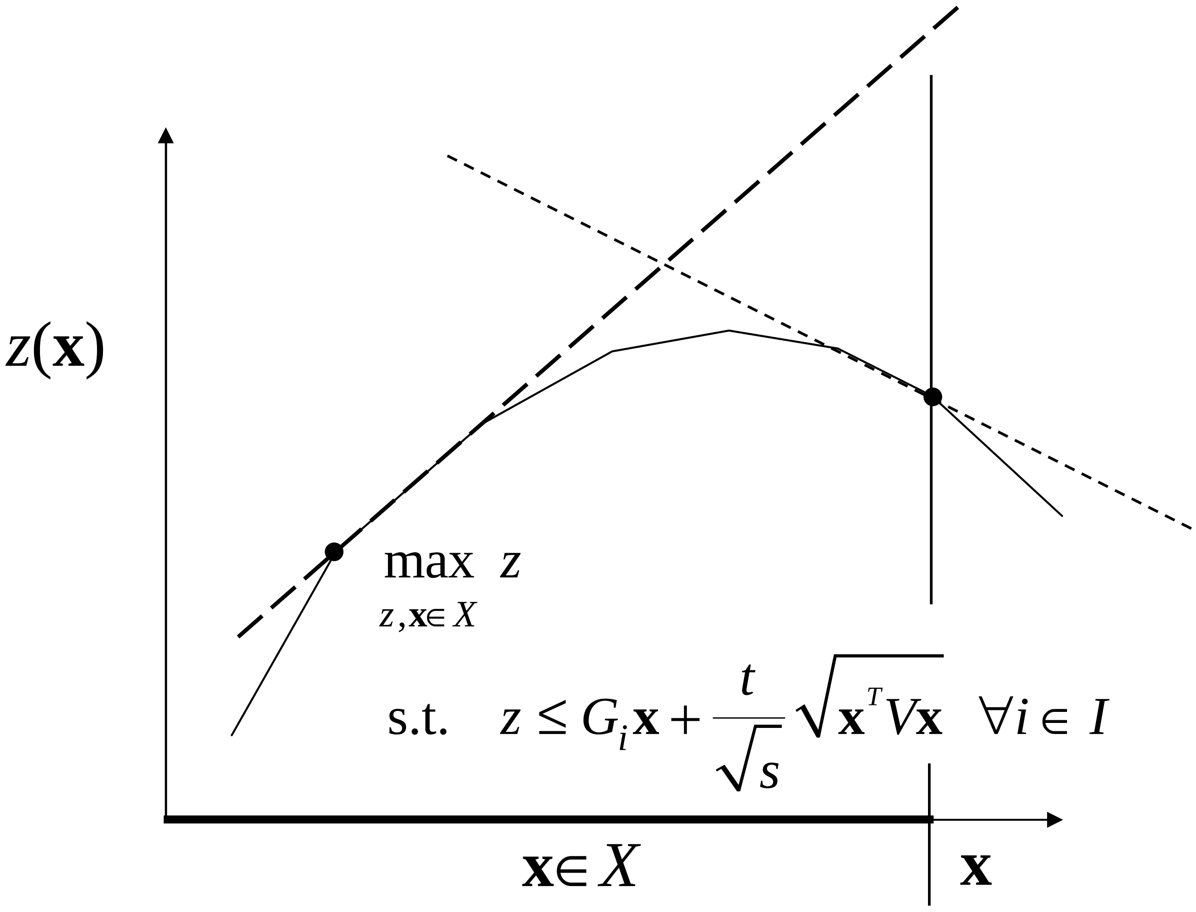
Pictorial view of Benders

Deterministic



Monte Carlo Benders (1)

Nonlinear cuts (g_i subsumed by G_i)

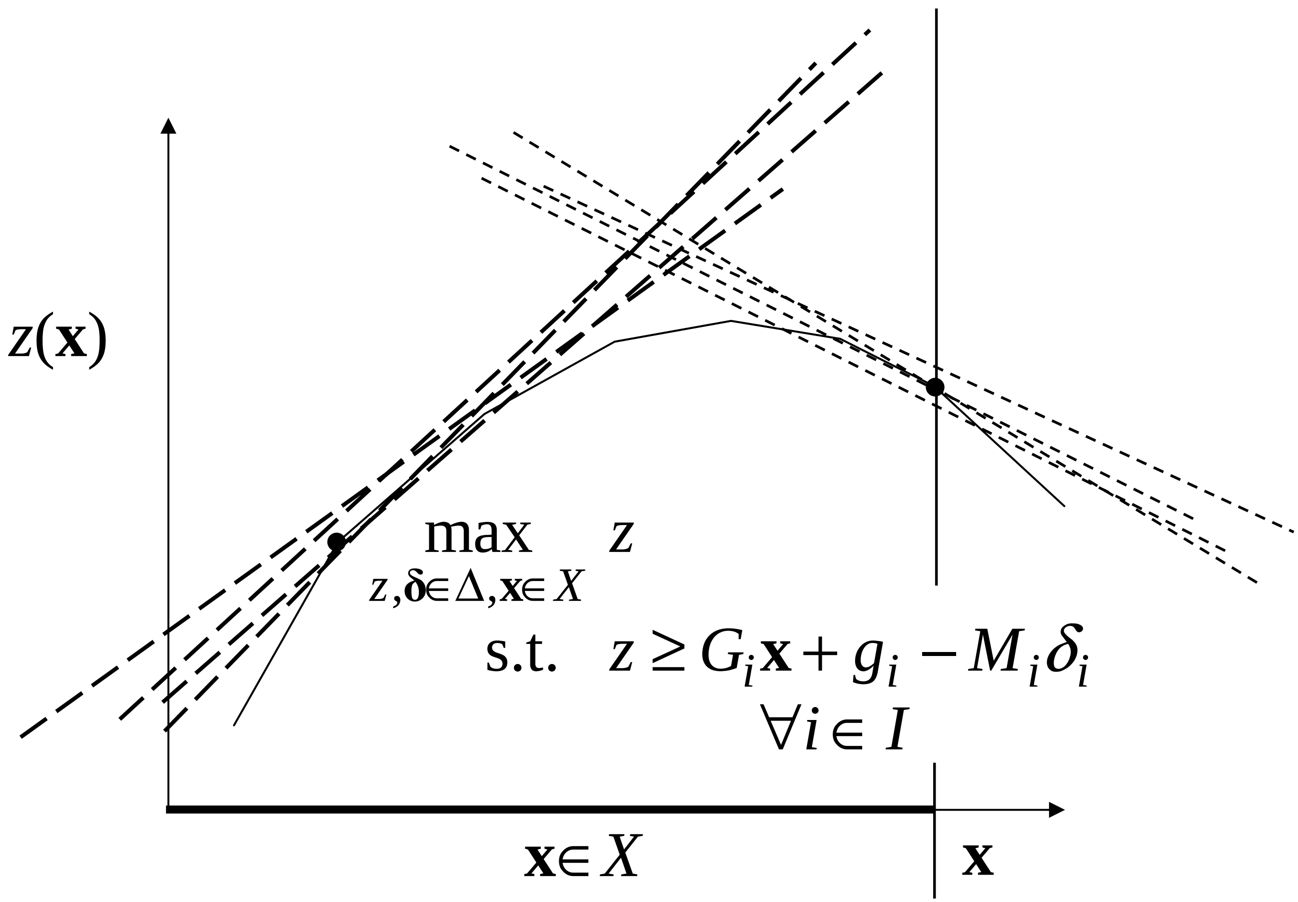


$$\max_{z,\mathbf{x} \in X} \; z$$

$$\text{s.t.} \quad z \le G_i \mathbf{x} + \frac{t}{\sqrt{s}} \sqrt{\mathbf{x}^T V \mathbf{x}} \quad \forall i \in I$$

Monte Carlo Benders (2)

Multiple sampled cuts, keep weakest



Contact info

- Kevin Wood OR/Wd
  Operations Research Dept.
  Naval Postgraduate School
  Monterey, CA 93943

- kwood@nps.navy.mil
  (831) 656-2523