The Nested Event Tree Model with Application to Combating Terrorism

Brian J. Lunday[1] Dr. Hanif D. Sherali[1] Dr. Theodore S. Glickman[2]

[1]Grado Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University

[2]School of Business, The George Washington University

October 11, 2009

Acknowledgements: Dr. Nick Sahinidis of the Σ-Optimization Group at the UIUC
Omar Nelson Bradley Officer Research Fellowship in Mathematics

Outline

- Motivation
- Model presentation
- Algorithmic development
  - Reformulation
  - Relaxation-based algorithms
  - Approximation-based solution techniques
- Comparative performance
  - Testing environment
  - Direct comparison of algorithms
  - Enhancements to superlative method
- Conclusions and recommendations

Motivation

- Goal: minimize the overall risk associated with the contemplation of an attack by a terrorist organization

- Accepted methodology for measuring risk

  Risk ≡ Threat * Vulnerability * Consequence

  s.t. Threat ≡ Capability * Intent

  → "Risk = CIVC"

- Issue: coordination of countermeasure and consequence-related resources
  - Government and private agency involvement
  - Levels of government: national, state, local, tribal, territorial
  - Lack of central coordinating agency with authority to direct actions

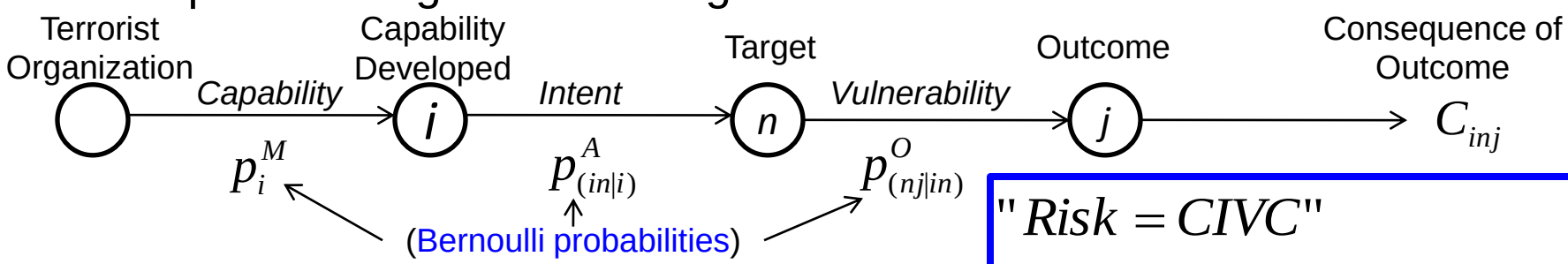
- Our problem: develop effective model to coordinate resources to minimize overall risk, along with efficient algorithm(s) to solve it to global optimality

Risk Determination

- Example capabilities, targets, and outcomes

Capability Types (i)	Target Types (n)	Attack Outcomes (j)
Assassination	Military base	Serious injuries
Dirty bombs	Political leaders	Fatalities
Kidnapping	High population density areas	Property damages
Chemical attack	Water treatment plant	Water supply contamination
Biological agents	Nuclear power plant	Infectious disease propagation
Nuclear weapons	Economic center	Economic Impact

- Example of a single terrorist organization-to-outcome risk calculation



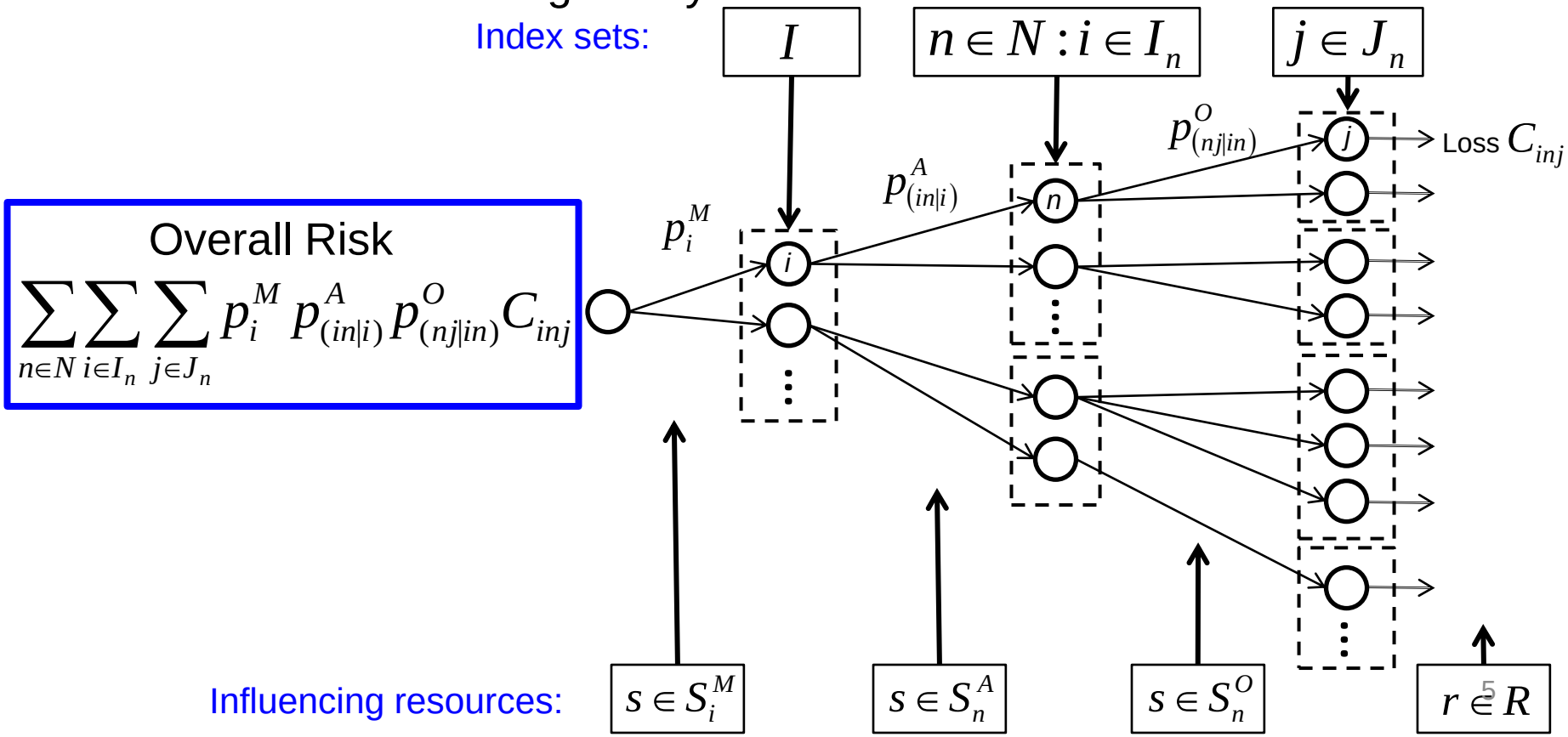
$$\text{"}Risk = CIVC\text{"}$$

$$Risk_{inj} = p_i^M \, p_{(in|i)}^A \, p_{(nj|in)}^O \, C_{inj}$$

- Model requirements
  - Account for all capabilities, targets, outcomes, and consequences within a single framework
  - Relate resource application to reduction of probabilities and consequences

Modeling Framework

- Nested event tree to account for all capabilities, targets, outcomes, and consequences within a single framework
  - A capability may affect more than one target
  - An attack on a target may result in more than one outcome



Textual Model Formulation

Minimize: Overall Risk

Subject to: Resource availabilities

Overall budget

Probability-resource relationships

Consequence-resource relationships

(Diminishing marginal returns desired)

Any additional bounds on probabilities (Often policy-driven)

Nonnegative resource application

*Results in a nonconvex NLP

Probability-Resource and Consequence-Resource Relationships

- Logit model for countermeasure resource to probability relationships

$$\ln\left[\frac{p_k^L}{1-p_k^L}\right] = \alpha^0 - \sum_{s\in S}\alpha_s x_s, \quad \forall \quad L\in\{M,A,O\}, k\in K_L$$

(Nonnegative constants)

  - Diminishing marginal returns below $p_k^L = 0.5$
  - Given limited resources, all probabilities are finitely bounded

$$p_k^{Ll} \le p_k^L \le p_k^{Lu}, \quad \forall \quad L,k\in K_L$$

- Exponential model for consequence management resource to consequence relationships
  - Resource vector y applied to consequence vector C

$$C_{inj} = \delta_j^{0in} e^{-h_j^{in}\sum_{r\in R}\delta_{rj}^{in} y_{rj}}, \quad \forall \quad i,n,j$$

(Nonnegative constants)

Reformulation of Complicating Constraints to Problem NETO(Ω)

1. Reformulate nonlinearities as logarithmic functions

(Resulting nonlinear constraints are univariate and monotonic)

$$\ln\left[\frac{p_k^L}{1-p_k^L}\right] = \alpha^0 - \sum_{s\in S}\alpha_s x_s \Rightarrow \quad \begin{aligned} q_{1k}^L &= \ln\left(p_k^L\right) \\ q_{2k}^L &= \ln\left(1-p_k^L\right) \end{aligned} \quad , \quad \forall \quad L,\left(k\in K_L\right)$$

$$q_{1k}^L - q_{2k}^L = \alpha^0 - \sum_{s\in S}\alpha_s x_s$$

2. Transform the costs

$$\theta_{inj} = \ln\left(C_{inj}\right) = \ln\delta_j^{0in} - h_j^{in}\sum_{r\in R}\delta_{rj}^{in} y_{rj}, \forall \quad i,n,j$$

3. Convexify the objective function

$$p_i^M p_{(in|i)}^A p_{(nj|in)}^O C_{inj} = e^{z_{inj}}, \forall \quad i,n,j \rightarrow \sum_{n\in N}\sum_{i\in I_n}\sum_{j\in J_n} e^{z_{inj}}$$

4. Take the natural logarithm of both sides and equivalently state

$$z_{inj} = q_{1i}^M + q_{1(in|i)}^A + q_{1(nj|in)}^O + \theta_{inj}, \quad \forall \quad i,n,j$$

5. Define the bounded hyperrectangle (Ω) on probabilities

$$\Omega = \left\{p_k^{Ll} \le p_k^L \le p_k^{Lu}, \quad \forall \quad L,\left(k\in K_L\right)\right\}$$

Algorithms Developed

	Relaxation/approximation to logarithmic functions in		Relaxation or Approximations Result in:
	(q_1,p)- and (q_2,p)-space	(q_1,q_2)-space	
• Polyhedral envelope construction on bounded logarithmic functions	A1	A1S	Convex NLPs
• Polyhedral envelope construction on bounded logarithmic functions • Relaxation of the objective function to linear supports	A2	A2S	LPs
• Relaxation of the objective function to linear supports • Piecewise linear inner approximations to logarithmic functions	A3	A3S	MIPs with convex feasible regions

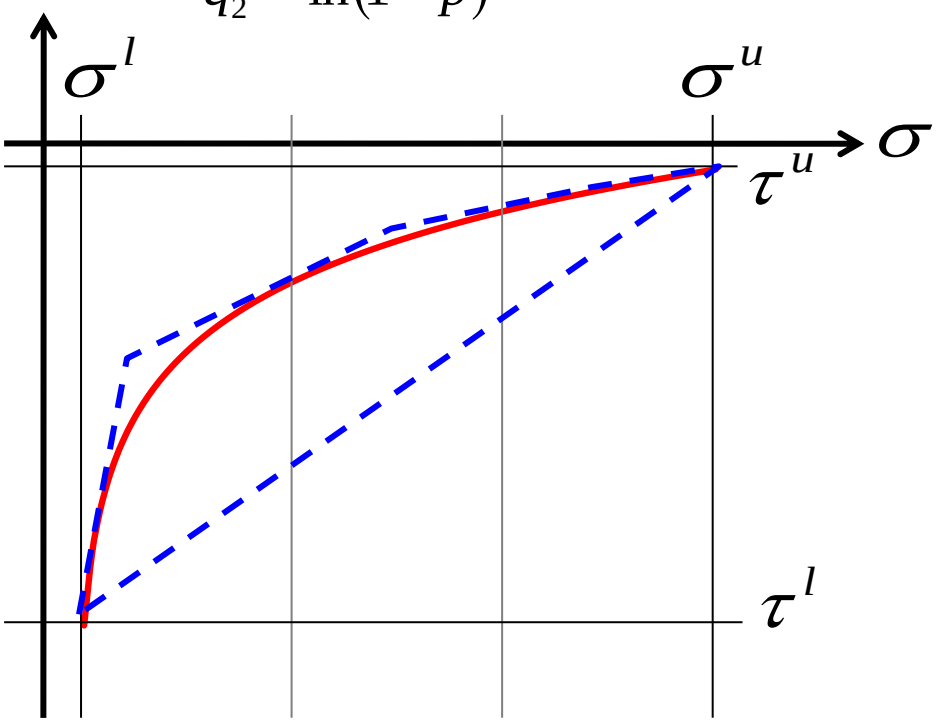
Algorithms A1, A1S, A2, A2S: Polyhedral Envelope Construction

- Construct a bounding, polyhedral envelope based on the hyperrectangle with upper bounding supports

For Algorithms A1 and A2

(q_1,p)- and (q_2,p)-space

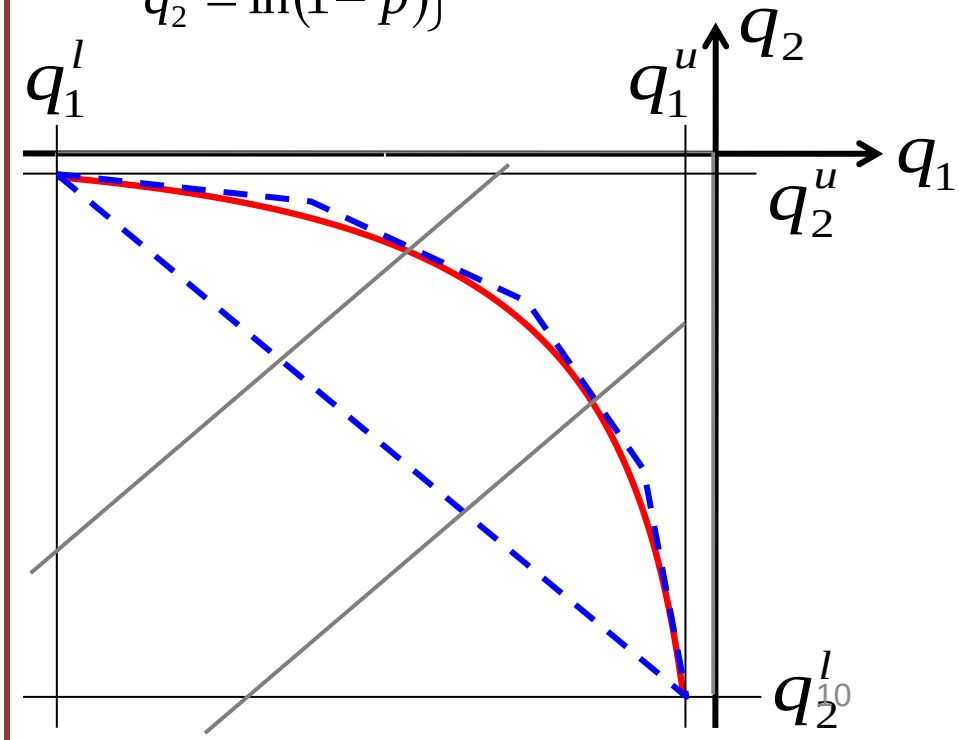
$$\left.\begin{aligned} q_1 &= \ln(p), \\ q_2 &= \ln(1-p) \end{aligned}\right. \Rightarrow \tau = \ln(\sigma)$$



For Algorithms A1S and A2S

(q_1,q_2)-space

$$\left.\begin{aligned} q_1 &= \ln(p) \\ q_2 &= \ln(1-p) \end{aligned}\right\} \Rightarrow q_2 = \ln\left(1-e^{q_1}\right)$$



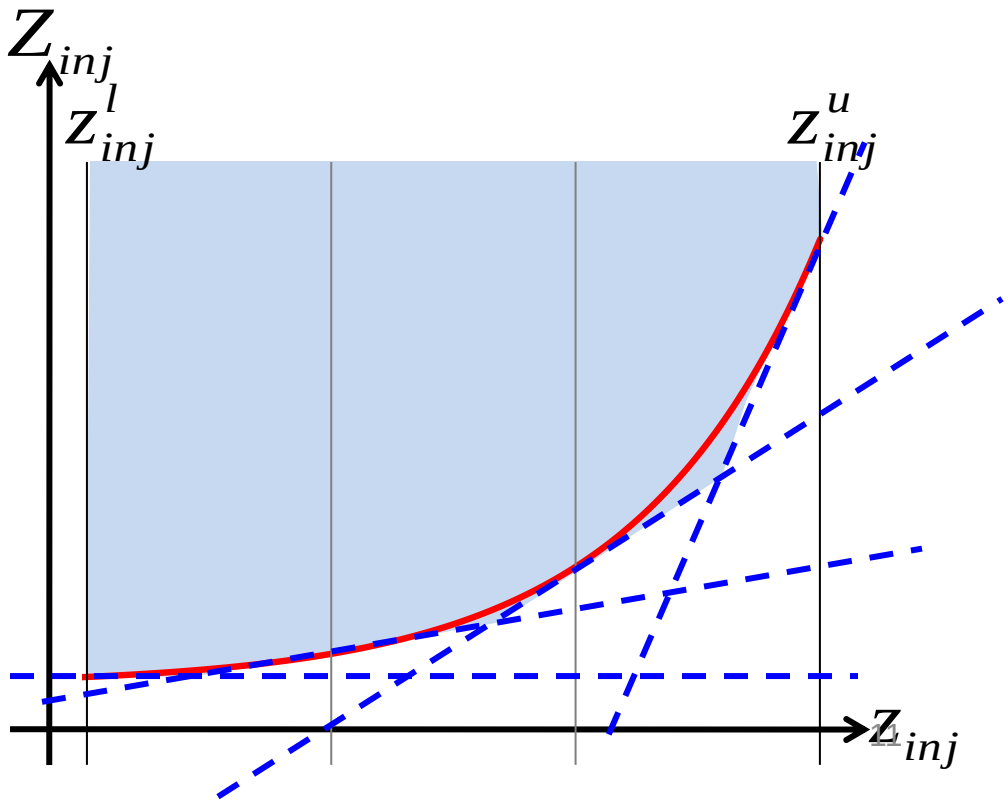
Algorithms A2, A2S, A3, A3S: Linear Bounding of NETO(Ω) Objective Function

<u>For Algorithms A2, A2S, A3 and A3S</u>

1. Rewrite the objective function as

$$\min \sum_{n \in N} \sum_{i \in I_n} \sum_{j \in J_n} Z_{inj} : Z_{inj} = e^{z_{inj}}, \quad \forall \quad i, n, j$$

2. Relax NETO(Ω) by replacing the exponential relationship with equispaced, lower bounding tangential supports



Properties of A1, A1S, A2, A2S Relaxations

1. The optimal objective function value for a relaxation is a lower bound on Problem NETO(Ω).

2. Given optimal resource allocations for a relaxation, can recover NETO(Ω) feasible solution with probabilities and consequences computed via the logit and exponential models.

3. If an optimal solution to the relaxation satisfies
$$q_1 = \ln(p), q_2 = \ln(1-p), \forall\, i, n, j$$
then it is also optimal to NETO(Ω).

4. Given an optimal solution to a relaxation, if each of the *p*-variables equals one of its bounds on the hyperrectangle Ω, then Property #3 applies.

Algorithmic A1, A1S, A2, A2S: Relaxation-based Branch and Bound

<u>Initialization</u>

1. Solve relaxation to determine LB_0
2. Calculate corresponding NETO solution (ζ^0) with objective value UB_0, setting it as incumbent solution with $\upsilon^*=UB_0$
3. Terminate if $LB_0 \geq \upsilon^*(1-\varepsilon)$ Otherwise, determine (L, $k \in K_L$) via the branching rule

<u>Node Selection and Branching</u>

1. Select node *a* for branching based on the least lower bound of active nodes
2. Create two subnodes, bifurcating the hyperplane Ω_a for each according to the partitioning rule

<u>Bounding Step</u> For nodes $h=a+1$, $a+2$, do the following:

1. Solve relaxation based on sub-hyperplane Ω_h to determine LB_h
2. If $LB_h < \upsilon^*(1-\varepsilon)$, determine (L_h, $k \in K_{Lh}$) via the branching rule. Compute corresponding ζ^h and UB_h.
3. If $UB_h < \upsilon^*$, then update incumbent solution

<u>Fathoming Step:</u> Fathom nodes for which If $LB_a \geq \upsilon^*(1-\varepsilon)$

Are all nodes fathomed?

Y → Accept incumbent solution as ε-optimal

N → Node Selection and Branching

Branch-and-Bound Variants and Enhancements

- Branching Rules: select the variable index for branching at node *a* by the maximum of a measure of deviation from the relaxed logarithmic constraint, noting that $L_a \in \{M, A, O\}, \Delta_a \in K_{L_a}$:
  1. Absolute deviation (ordinal axis)
  2. Relative deviation (ordinal axis)
  3. Relative deviation (abscissal axis)

- Partitioning Rules: on branching, split the interval $[p_{\Delta_a}^{L_a l}, p_{\Delta_a}^{L_a u}]$ at:
  1. Incumbent value, $p_{\Delta_a}^{L_a}$
  2. Arithmetic mean
  3. Geometric mean

- Range reduction at each node, for a maximum of 100 nodes, so long as the parent node exhibited improvement in at least one p-variable bound by at least 0.1%

Fundamental Algorithmic Properties

- **Proposition (Algorithm A1)**. The proposed branch-and-bound algorithm A1, with ε=0, either terminates finitely with the incumbent solution for the convex relaxation being optimal to Problem NETO, or else an infinite sequence of stages is generated such that, along any infinite branch of the branch-and-bound tree, any accumulation point of the (x,y,p,C)-variable part of the convex programming relaxation solutions generated for the corresponding node subproblems solves Problem NETO.

- **Corollary (Algorithm A1).** For ε>0, the proposed algorithm A1 applied to the convex relaxation will converge to an ε-optimal solution for Problem NETO within a finite number of iterations.

- **Proposition (Algorithm A1S)**. The algorithm is not impacted by the number of upper bounding, tangential supports in (q_1,q_2)-space, as the optimal solution to the relaxation will lie on the lower bounding support.

- **Proposition (Algorithm A2)**. For ε>0 and a sufficient number of linear bounding supports for the objective function in the linear relaxation to ensure its induced error is less than ε, then the proposed algorithm A2 applied to the linear relaxation will converge to an ε-optimal solution for Problem NETO within a finite number of iterations.

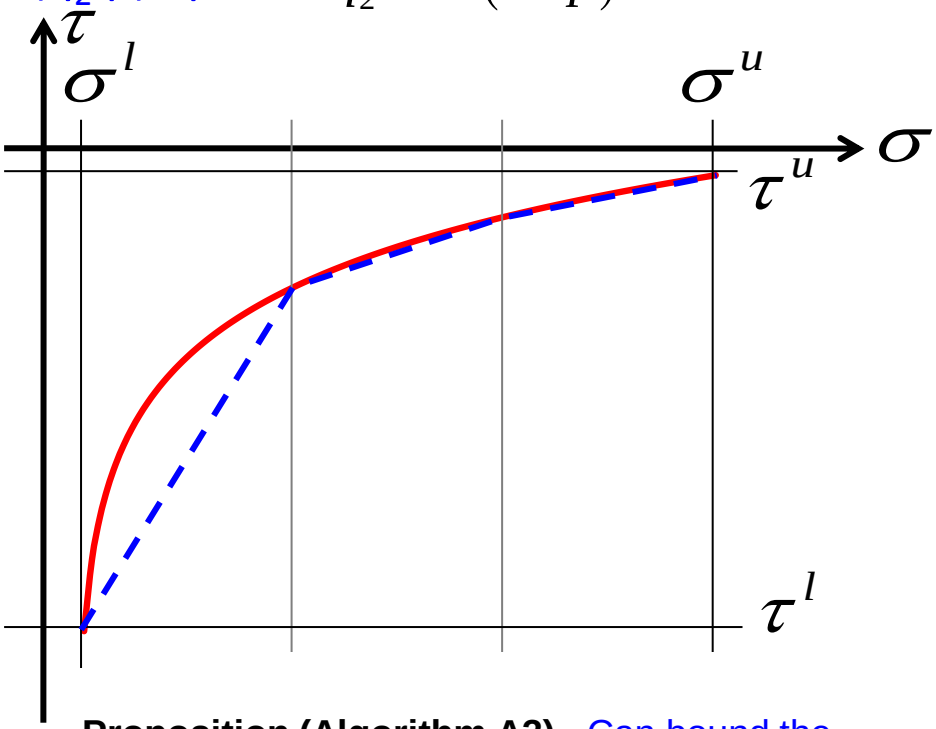
Approximations for A3 and A3S

- Piecewise inner linear approximations, with objective linearly bounded as per Algorithm A2 and A2S

Algorithm A3

(q_1,p)- and (q_2,p)-space

$$q_1 = \ln(p), \quad q_2 = \ln(1-p) \Rightarrow \tau = \ln(\sigma)$$

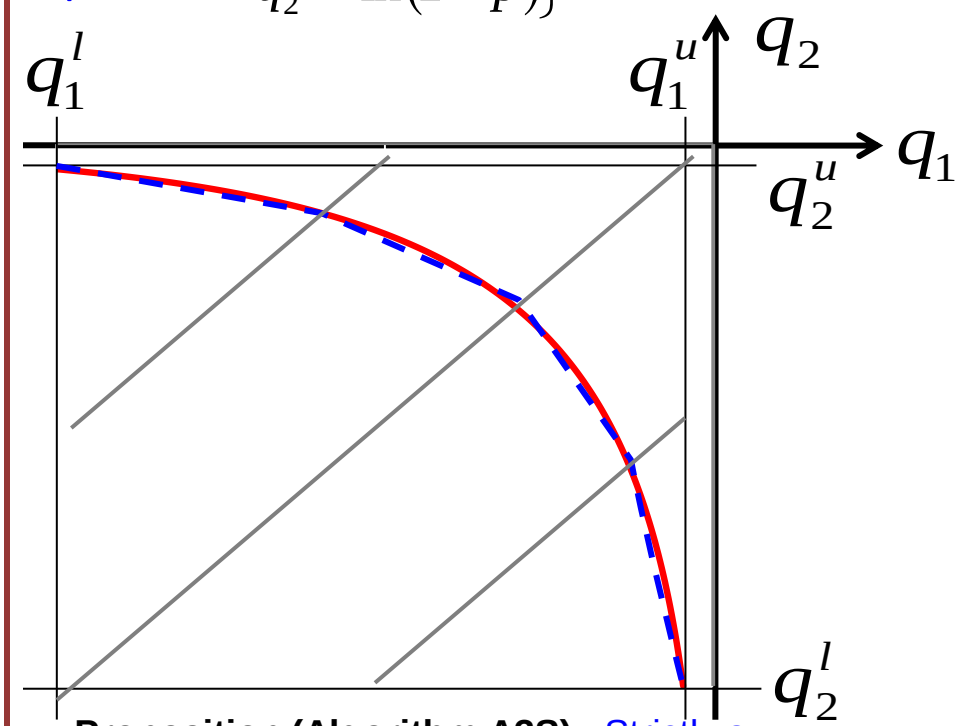


Proposition (Algorithm A3). Can bound the maximum error based on the equidistant width between breakpoints on the ordinal axis.

Algorithm A3S

(q_1,q_2)-space

$$\left.\begin{array}{l} q_1 = \ln(p) \\ q_2 = \ln(1-p) \end{array}\right\} \Rightarrow q_2 = \ln\left(1-e^{q_1}\right)$$



Proposition (Algorithm A3S). Strictly a relaxation to NETO(Ω), thereby providing a lower bound (and an optimality gap).

Testing Environment and Initial Computational Results

- Environment
  - Algorithm A1/A1S using C++ and SNOPT 7.2
  - Algorithm A2/A2S, A3/A3S using C++ and ILOG CPLEX 11.1
  - Compared to BARON 8.1.5 using CPLEX for LP subproblems and SNOPT for NLP subproblems
  - Run on Intel 2.40 GHz Xeon Processor and 1.5 GB of RAM
  - All tested over ten randomly generated instances

- Results that Prescribe Parametric Settings
  - Apply 4 upper bounding tangential supports in (p,q_1)- and (p,q_2)-space for Algorithms A1 and A2
  - Employ 32 lower bounding tangential supports for the objective function in Algorithms A2, A2S, A3, A3S
  - Branching rule: relative deviation (ordinal axis) from the logarithmic identity
  - Partitioning rule: arithmetic mean
  - Invoke range reduction

Summary of Computational Comparisons with BARON

- Termination criteria:

  ε=0.01 1001 nodes for A1, A1S, A2, A2S 1800 CPU seconds

- Commercial Solver Limitations
  - Average performances only reported for Instances 1-8,10, as BARON terminated at 1884.3 seconds on Instance 9 without identifying a feasible solution.

Strategy	A1	A1S	A2	A2S	A3	A3S	BARON
ε-optimal (out of 10)	9	9	8	9	9	10	4
Relative gap (%):	0.62	0.62	0.71	0.70	0.69	0.35	4.29
Nodes explored:	2.78	2.78	224.33	114.56	--	--	70.78
Effort (CPU sec.):	188.87	196.93	221.82	121.00	173.38	35.00	1144.94

- Conclusions
  - All of the proposed algorithms tested significantly outperformed BARON with respect to optimality gap attained and computational effort.
  - Only Algorithm A3S attained the specified optimality gap for all instances
  - Algorithm A3S versus BARON
    - 3.95% reduction in average optimality gaps attained
    - 96.94% reduction in computational effort

Enhancements to Algorithm A3S

- Increasing the number of segments in the piecewise linear approximation in (q_1,q_2)-space improves the optimality gap attained, though at a faster than linear increase in computational times for the six largest test instances
  - Decreasing segments did not attain ε-optimal solutions for all instances
  - Increasing segments ultimately exceeded implementation time constraints

Number of segments	2	4	8	16
Relative gap (%):	0.58	0.37	0.32	*
Effort (CPU sec.):	22.786	35.009	163.39	*

- Increasing the number of linear supports to the objective function improves the optimality gap attained, with computational effort increasing less than five-fold on all instances, for an eightfold increase in the number of supports

Number of supports	32	64	128	256
Relative gap (%):	0.37	0.14	0.088	0.072
Effort (CPU sec.):	35.009	52.339	81.900	157.33

Conclusions and Recommendations

- Nested event tree optimization modeling framework comprehensively addresses capability, intent, vulnerability, and consequence issues in combating terrorism.
  - Useful for allocation decisions and scenario analysis
- Alternative reformulations along with specialized global optimization algorithms to solve the resulting challenging nonconvex programming problems are effective and efficient, compared to BARON commercial software.
- Algorithmic performance can be further improved with alteration of certain parameters
- Model can be further modified to:
  - Represent multiple terrorist organizations
  - Account for partially aligned interests between nations applying resources
  - Consider alternative probability-resource relationships