Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute for Operations Research
Department of Mathematics

Robust Network Planning Problems

Utz-Uwe Haus[1] Carla Michini[2] Marco Laumanns[3]

Supported by EU-FP7-ITNs 289581 ‘NPlast’ and 316647 ‘MINO’

[1]ETHZ [2]U Wisconsin, Madison [3]IBM Research (Switzerland)

CWM^{3}EO, Budapest, Hungary, September 25/26, 2014

Goal and motivation

- Robustifying Infrastructure
  - survivability, cost robustness
  - structural robustness
  - pre-disaster investment
- Different modeling and solution approaches
  - (worst-case-) robust optimization
  - fault-tolerant-feasibility models
  - multistage stochastic models

Goal and motivation

- Robustifying Infrastructure
  - survivability, cost robustness
  - structural robustness
  - pre-disaster investment
- Different modeling and solution approaches
  - (worst-case-) robust optimization
  - fault-tolerant-feasibility models
  - multistage stochastic models

Networks with uncertain data: Robust Optimization

Setting

A network problem with uncertain cost, supply/demand or capacity data: uncertainty set Δ

Solution approaches

- Worst-case ($\forall \delta \in \Delta$) or best-case ($\exists \delta \in \Delta$) setting are interesting
- $\forall$: Reformulate and solve robust counterpart
- $\exists$: Reformulate as Generalized LP(/IP/...)
- ... or separate robust/generalized (split-)cuts directly
- Caveat 1: Reformulation often loses combinatorial structure
- Caveat 2: Shortest path with 2 cost scenarios is (weakly) $\mathcal{NP}$-hard

Networks with uncertain data: Robust Optimization

Setting

A network problem with uncertain cost, supply/demand or capacity data: uncertainty set Δ

Solution approaches

- Worst-case ($\forall \delta \in \Delta$) or best-case ($\exists \delta \in \Delta$) setting are interesting
- $\forall$: Reformulate and solve robust counterpart
- $\exists$: Reformulate as Generalized LP(/IP/...)
- ... or separate robust/generalized (split-)cuts directly
- Caveat 1: Reformulation often loses combinatorial structure
- Caveat 2: Shortest path with 2 cost scenarios is (weakly) $\mathcal{NP}$-hard

Structural Robustness – Fault-tolerant Feasibility

Deal with failure of resources, have certain cost structure
Requires: up-monotonicity of feasible sets

Structural Robust Counterpart

Given nominal instance $P = (A, S, w)$ (A: ground set, S: feasible solutions, w: cost function) and failure scenarios $\Omega = \{F_1, \dots, F_k\}$ with $F_i \subseteq A$ find X^* attaining

$$\min_{X \subseteq A : \forall i : X \setminus F_i \in S} w(X).$$

Idea: Accept (potentially more expensive) solutions that remain feasible in every scenario

Structural Robustness: Survey

Examples for $P = (A, S, w)$

Shortest Path (SP), Bipartite Matching (BM), Spanning Tree (ST), Matroid Linear Optimization (MLO), Sparsest k-Spanner (SkS)

Scenario encodings

- explicit $\Omega = \{F_1, \ldots, F_k\}$: ERCC$(P, k)$
- implicit:
  - uniform cardinality constrained $\Omega = \{F \subseteq A : |F| \leq k\}$: IRCC$(P, k)$
  - uniform cardinality constrained in subset $U \subseteq A$ ('unguarded' elements) $\Omega = \{F \subseteq U : |F| \leq k\}$: SIRCC$(P, U, k)$

for details: see Adjiashvili et al, MathProg A 2014

Structural Robustness: Some results

ERCC(P,k) hardness

Assuming $\mathcal{NP} \not\subset \mathcal{DTIME}(n^{\log\log n})$ there is no polynomial $c \log k$ approximation for ERCC(P) for any $c < 1$. (P=SP,BM,ST,MLO,SkS)

ERCC(MLO,k) approximability

There is a polynomial $\mathcal{O}(\log \mathrm{rk}(M) + \log k)$ approximatation algorithm for ERCC(MLO,k) (M a matroid).

ERCC(SP,k,$|F_j| \leq 2$) approximability

There is a a constant-factor approximation for ERCC(SP,k,$|F_j| \leq 2$), and for $|F_j| = 1$ the algorithm is exact.

(Adjiashvili et al, MathProg A 2014)

Structural Robustness: Some results

ERCC(P,k) hardness

Assuming $\mathcal{NP} \not\subset \mathcal{DTIME}(n^{\log\log n})$ there is no polynomial $c \log k$ approximation for ERCC(P) for any $c < 1$. (P=SP,BM,ST,MLO,SkS)

ERCC(MLO,k) approximability

There is a polynomial $\mathcal{O}(\log \mathrm{rk}(M) + \log k)$ approximatation algorithm for ERCC(MLO,k) (M a matroid).

ERCC(SP,k,$|F_j| \leq 2$) approximability

There is a a constant-factor approximation for ERCC(SP,k,$|F_j| \leq 2$), and for $|F_j| = 1$ the algorithm is exact.

(Adjiashvili et al, MathProg A 2014)

Structural Robustness: Some results

ERCC(P,k) hardness

Assuming $\mathcal{NP} \not\subset \mathcal{DTIME}(n^{\log\log n})$ there is no polynomial $c \log k$ approximation for ERCC(P) for any $c < 1$. (P=SP,BM,ST,MLO,SkS)

ERCC(MLO,k) approximability

There is a polynomial $\mathcal{O}(\log \mathrm{rk}(M) + \log k)$ approximatation algorithm for ERCC(MLO,k) (M a matroid).

ERCC(SP,k,$|F_j| \leq 2$) approximability

There is a a constant-factor approximation for ERCC(SP,k,$|F_j| \leq 2$), and for $|F_j| = 1$ the algorithm is exact.

(Adjiashvili et al, MathProg A 2014)

Pre-Disaster Investment – Stochastic Models

2-stage stochastic problem

$$\min \mathbb{E}_{\xi|x}[f(\xi)]$$
$$Cx \leq d$$
$$x = (x_e)_{e\in E} \in \{0,1\}^{|E|}, \xi \quad = (\xi_e)_{e\in E} \in \{0,1\}^{|E|}$$

where computing $f(\xi)$ means solving an optimization problem in scenario realization ξ after decisions x (ξ_e indep. rand. var.).

Example

Consider a graph $G = (V, E)$, edge lengths $(l_e)_{e\in E}$, and edge survivability probabilities $(p_e)_{e\in E}$. Scenarios correspond to sets of surviving edges after a disaster. $f_{\text{SP}}(s, t, G_\xi)$ is the shortest path length between two designated nodes $s, t \in V$. Decisions are whether to strengthen edge e, i.e. improve resilience, to $p_e + \delta_e$.

Pre-Disaster Investment – Stochastic Models

2-stage stochastic problem

$$\min \mathbb{E}_{\xi|x}[f(\xi)]$$
$$Cx \leq d$$
$$x = (x_e)_{e\in E} \in \{0,1\}^{|E|},\ \xi \quad = (\xi_e)_{e\in E} \in \{0,1\}^{|E|}$$

where computing $f(\xi)$ means solving an optimization problem in scenario realization ξ after decisions x (ξ_e indep. rand. var.).

Example

Consider a graph $G = (V, E)$, edge lengths $(l_e)_{e\in E}$, and edge survivability probabilities $(p_e)_{e\in E}$. Scenarios correspond to sets of surviving edges after a disaster. $f_{\mathrm{SP}}(s, t, G_\xi)$ is the shortest path length between two designated nodes $s, t \in V$. Decisions are whether to strengthen edge e, i.e. improve resilience, to $p_e + \delta_e$.

Solution approaches

- sampling scenarios/simulation
- sample average/sample-path: may yield statistically testable bounds
- partition or cover scenario space and do exact reformulation

Example (cont.): Computing expected path length

$$\min \sum_{\xi \in 2^E} \left(\prod_{e \in \xi} p_e \prod_{e \notin \xi} (1 - p_e) \right) f_{\mathrm{SP}}(s, t, G_\xi = (V, \xi))$$

Instead of enumerating $2^{|E|}$ scenarios the scenarios can be partitioned into sets with same f-value whose probabilities can be computed. (Prestwich et al., '13)

Solution approaches

- sampling scenarios/simulation
- sample average/sample-path: may yield statistically testable bounds
- partition or cover scenario space and do exact reformulation

Example (cont.): Computing expected path length

$$\min \sum_{\xi \in 2^E} \left(\prod_{e \in \xi} p_e \prod_{e \notin \xi} (1 - p_e) \right) f_{\mathrm{SP}}(s, t, G_\xi = (V, \xi))$$

Instead of enumerating $2^{|E|}$ scenarios the scenarios can be partitioned into sets with same f-value whose probabilities can be computed. (Prestwich et al., '13)

Solution approaches

- sampling scenarios/simulation
- sample average/sample-path: may yield statistically testable bounds
- partition or cover scenario space and do exact reformulation

Example (cont.): Computing expected path length

$$\min \sum_{\xi \in 2^E} \left(\prod_{e \in \xi} p_e \prod_{e \notin \xi} (1 - p_e) \right) f_{\mathrm{SP}}(s, t, G_\xi = (V, \xi))$$

Instead of enumerating $2^{|E|}$ scenarios the scenarios can be partitioned into sets with same f-value whose probabilities can be computed. (Prestwich et al., '13)

A 2-stage stochastic optimization problem is called aggregable if

- f is order-reversing (or order-preserving) wrt. taking subsets of scenarios,
$$\xi_1 \subseteq \xi_2 \Rightarrow f(\xi_1) \geq f(\xi_2) \tag{1}$$
for all $\xi_1, \xi_2 \in 2^E$,
- the probabilities of events $e \in E$ are independent.

We denote the range of f by $\mathcal{C}(f) = \{\alpha : \alpha = f(\xi), \xi \in 2^E\}$, and the minimal survivable scenarios for each critical value by $\mathcal{M}(f) = \{\mathcal{M}_\alpha(f) : \alpha \in \mathcal{C}(f)\}$ with $\mathcal{M}_\alpha(f) = \{\xi \in 2^E : f(\xi) = \alpha, \forall \xi' \subset \xi : f(\xi') > f(\xi)\}$.

A 2-stage stochastic optimization problem is called aggregable if

- f is order-reversing (or order-preserving) wrt. taking subsets of scenarios,
$$\xi_1 \subseteq \xi_2 \Rightarrow f(\xi_1) \geq f(\xi_2) \tag{1}$$
for all $\xi_1, \xi_2 \in 2^E$,
- the probabilities of events $e \in E$ are independent.

We denote the range of f by $\mathcal{C}(f) = \{\alpha \ : \ \alpha = f(\xi), \xi \in 2^E\}$, and the minimal survivable scenarios for each critical value by
$\mathcal{M}(f) = \{\mathcal{M}_\alpha(f) \ : \ \alpha \in \mathcal{C}(f)\}$ with
$\mathcal{M}_\alpha(f) = \{\xi \in 2^E \ : \ f(\xi) = \alpha, \forall \xi' \subset \xi : f(\xi') > f(\xi)\}$.

Examples

'friendly': where computing $f(\xi)$ is polynomial time

- (multi-terminal-) shortest path
- number of edge-disjoint paths/k-connectivity
- longest path in acyclic networks
- maximal flow
- maximal/max weighted matching
- LP (with vanishing constraints)

but also

- clique number

Each $\mathcal{M}_\alpha(f)$ induces a monotone Boolean function $\Phi_\alpha^\leq$ on the scenarios whose minimal true points are the members of $\mathcal{M}_\alpha(f)$ by

$$\Phi_\alpha^\leq(\xi) = 1 \text{ if and only if } f(\xi) \leq \alpha.$$

Encoding $\Phi_\alpha^\leq$

- as DNF: using explicit list $\mathcal{M}_\alpha(f)$
- as IP of covering type: $p^\top x \geq 1 (\forall p \in \mathcal{M}_\alpha(f))$
- as binary decision diagram (BDD), built from explicit or implicit $\mathcal{M}_\alpha(f)$
- using the fact that it is isomorphic to the BDD of the dual monotone Boolean function $\neg(\Phi_\alpha^\leq(\neg\xi))$

Each $\mathcal{M}_\alpha(f)$ induces a monotone Boolean function $\Phi_\alpha^\leq$ on the scenarios whose minimal true points are the members of $\mathcal{M}_\alpha(f)$ by

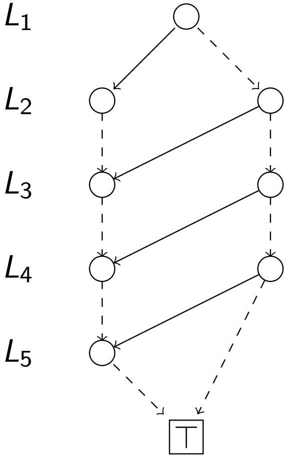
$$\Phi_\alpha^\leq(\xi) = 1 \text{ if and only if } f(\xi) \leq \alpha.$$

Encoding $\Phi_\alpha^\leq$

- as DNF: using explicit list $\mathcal{M}_\alpha(f)$
- as IP of covering type: $p^\top x \geq 1 (\forall p \in \mathcal{M}_\alpha(f))$
- as binary decision diagram (BDD), built from explicit or implicit $\mathcal{M}_\alpha(f)$
- using the fact that it is isomorphic to the BDD of the dual monotone Boolean function $\neg(\Phi_\alpha^\leq(\neg\xi))$

(Reduced Ordered) Binary Decision Diagrams (Bryant, 1986)

- Layered (rooted) digraph
- arcs only from L_i to L_j s.t. $j > i$
- each node has at most two outgoing arcs
- true-arcs are plain, false-arcs are dotted
- each path from the root to $\top$ defines a feasible solution (or a 'nice' family)
- each path from the root to some node defines a partial feasible solution
- each node is root of a unique sub-BDD encoding all completions
- Layer L_i has width $\omega_i = |L_i|$
- BDD width $\omega = \max_i \omega_i$



Boolean function in CNF: A covering problem

Let $A \in \{0,1\}^{m \times n}$.

$$\begin{aligned} Ax &\geq 1 \\ x &\in \{0,1\}^n \end{aligned} \qquad \text{(SC)}$$

TOP-DOWN BDD COMPILATION:
Let $u, v \in L_4$ with paths $(1,0,0)$ and $(0,0,1)$

Example:

$$\begin{array}{cccccccc} x_1+ & & x_3+ & & & x_6 & \geq 1 \\ & & & x_4+ & & x_6 & \geq 1 \\ & x_2+ & & x_4+ & x_5 & & \geq 1 \\ x_1+ & x_2+ & x_3+ & & & & \geq 1 \\ & & x_3+ & x_4+ & x_5 & & \geq 1 \\ x \in & \{0,1\}^n & & & & & \end{array}$$

Boolean function in CNF: A covering problem

Let $A \in \{0,1\}^{m \times n}$.

$$\begin{aligned} Ax &\geq 1 \\ x &\in \{0,1\}^n \end{aligned} \qquad \text{(SC)}$$

TOP-DOWN BDD COMPILATION:

Let $u, v \in L_4$ with paths $(1,0,0)$ and $(0,0,1)$

Example:

$$\begin{array}{cccccccc}
1 & 0 & 0 & & & & & \\
x_1+ & & x_3+ & & & x_6 & \geq & 1 \\
 & & & x_4+ & & x_6 & \geq & 1 \\
 & x_2+ & & x_4+ & x_5 & & \geq & 1 \\
x_1+ & x_2+ & x_3+ & & & & \geq & 1 \\
 & & x_3+ & x_4+ & x_5 & & \geq & 1
\end{array}$$

$x \in \{0,1\}^n$

Boolean function in CNF: A covering problem

Let $A \in \{0,1\}^{m \times n}$.

$$\begin{array}{l} Ax \geq 1 \\ x \in \{0,1\}^n \end{array} \qquad \text{(SC)}$$

TOP-DOWN BDD COMPILATION:
Let $u, v \in L_4$ with paths $(1,0,0)$ and $(0,0,1)$

Example:

$$\begin{array}{ccccccccc}
0 & & 0 & & 1 & & & & \\
x_1 + & & & & x_3 + & & & x_6 & \geq 1 \\
 & & & & & x_4 + & & x_6 & \geq 1 \\
 & & x_2 + & & & x_4 + & x_5 & & \geq 1 \\
x_1 + & & x_2 + & & x_3 + & & & & \geq 1 \\
 & & & & x_3 + & x_4 + & x_5 & & \geq 1 \\
x \in & & \{0,1\}^n & & & & & &
\end{array}$$

Boolean function in CNF: A covering problem

Let $A \in \{0,1\}^{m \times n}$.

$$\begin{aligned} Ax &\geq 1 \\ x &\in \{0,1\}^n \end{aligned} \qquad \text{(SC)}$$

TOP-DOWN BDD COMPILATION:
Let $u, v \in L_4$ with paths $(1,0,0)$ and $(0,0,1)$

Completions at u and v determined by same matrix minor

$\Rightarrow$ u and v can be merged!

Example:

$$\begin{array}{ccccccl} x_1 + & & x_3 + & & & x_6 & \geq 1 \\ & & & x_4 + & & x_6 & \geq 1 \\ & x_2 + & & x_4 + & x_5 & & \geq 1 \\ x_1 + & x_2 + & x_3 + & & & & \geq 1 \\ & & x_3 + & x_4 + & x_5 & & \geq 1 \\ x \in & \{0,1\}^n & & & & & \end{array}$$

Boolean function in CNF: A covering problem

Let $A \in \{0,1\}^{m \times n}$.

$$\begin{aligned} Ax &\geq 1 \\ x &\in \{0,1\}^n \end{aligned} \qquad \text{(SC)}$$

TOP-DOWN BDD COMPILATION:

Let $u, v \in L_4$ with paths $(1,0,0)$ and $(0,0,1)$

Example:

$$\begin{array}{cccccccc}
x_1 + & & x_3 + & & & x_6 & \geq 1 \\
& & & x_4 + & & x_6 & \geq 1 \\
& x_2 + & & x_4 + & x_5 & & \geq 1 \\
x_1 + & x_2 + & x_3 + & & & & \geq 1 \\
& & x_3 + & x_4 + & x_5 & & \geq 1 \\
x \in & \{0,1\}^n & & & & &
\end{array}$$

(having DNF of $\Phi_{\alpha}^{\leq}$ gives us CNF of its dual for free; resulting BDD only needs arc label flipping)

BDDs encoding the members of an Independence System/Circuit System

Top-down compilation rule for BDDs encoding the members of $\mathcal{I}$.

Key ingredient: an oracle to decide if two minors of the circuit system of $\mathcal{I}$ are equivalent.

Examples: stable sets, packing, matching, covering, knapsack.

If an efficient oracle is available, the procedure yields an output-linear time algorithm for BDD compilation (e.g.: stable sets, packing, covering, or graphic matroid, but not 0/1-knapsack)

The size of BDDs depends heavily on the ordering of the variables. Is it possible to bound the maximum width of a BDD with respect to some ordering?

BDDs encoding the members of an Independence System/Circuit System

Top-down compilation rule for BDDs encoding the members of $\mathcal{I}$.

Key ingredient: an oracle to decide if two minors of the circuit system of $\mathcal{I}$ are equivalent.

Examples: stable sets, packing, matching, covering, knapsack.

If an efficient oracle is available, the procedure yields an output-linear time algorithm for BDD compilation (e.g.: stable sets, packing, covering, or graphic matroid, but not 0/1-knapsack)

The size of BDDs depends heavily on the ordering of the variables. Is it possible to bound the maximum width of a BDD with respect to some ordering?

BDDs encoding the members of an Independence System/Circuit System

Top-down compilation rule for BDDs encoding the members of $\mathcal{I}$.

Key ingredient: an oracle to decide if two minors of the circuit system of $\mathcal{I}$ are equivalent.

Examples: stable sets, packing, matching, covering, knapsack.

If an efficient oracle is available, the procedure yields an output-linear time algorithm for BDD compilation (e.g.: stable sets, packing, covering, or graphic matroid, but not 0/1-knapsack)

The size of BDDs depends heavily on the ordering of the variables. Is it possible to bound the maximum width of a BDD with respect to some ordering?

Graphic matroid

Enumerating all spanning forests of a graph

- Independence system: spanning forests of G
- Circuit system: simple cycles of G

Circuit system minor equivalence check

Needs to check whether two minors of G have the same simple cycles

- Need to check graph 2-isomorphism, but only for two minors of G under edge deletion/contraction of a common initial segment of the edge order: linear time!
  - Choose basis for minors (necessarily same for both minors)
  - compute basis representation
  - compare coefficients (feature of binary matroids)

Bandwidth and BDD width

Let $\mathcal{C}$ be a clutter and let A be the matrix whose rows are the incidence vectors of the members of $\mathcal{C}$.

The bandwidth $b(A)$ of A is the largest distance between any two ones in a row.

Bandwidth and BDD width

Let $\mathcal{C}$ be a clutter and let A be the matrix whose rows are the incidence vectors of the members of $\mathcal{C}$.

The bandwidth $b(A)$ of A is the largest distance between any two ones in a row.

Bandwidth and BDD width

Let $\mathcal{C}$ be a clutter and let A be the matrix whose rows are the incidence vectors of the members of $\mathcal{C}$.

The bandwidth $b(A)$ of A is the largest distance between any two ones in a row.

Example:

$$A = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

$$b(A) = 4$$

Bandwidth and BDD width

Let $\mathcal{C}$ be a clutter and let A be the matrix whose rows are the incidence vectors of the members of $\mathcal{C}$.

The bandwidth $b(A)$ of A is the largest distance between any two ones in a row.

Example:

$$A = \begin{array}{c} \begin{array}{ccccccccc} & & & & & L_j & & & \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

$$b(A) = 4$$

Bandwidth and BDD width

Let $\mathcal{C}$ be a clutter and let A be the matrix whose rows are the incidence vectors of the members of $\mathcal{C}$.

The bandwidth $b(A)$ of A is the largest distance between any two ones in a row.

For the BDD B associated to $\mathcal{C}$ (in the variable ordering given by the constraint matrix), it holds that $\omega(B) \leq 2^{b(A)-1}$.

Example:

$$\begin{array}{cc} & \begin{array}{ccccccccc} & & & & & L_j & & & \end{array} \\ A = & \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

$$b(A) = 4$$

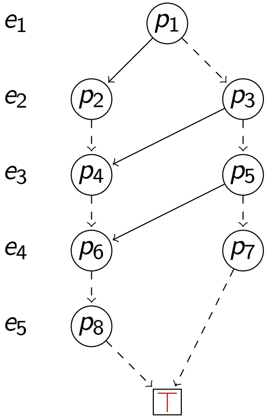
$$\omega(B) \leq 2^3$$

Proof:

- Consider a node $u \in L_j$ in layer j of the BDD encoding the transversals of the clutter. Since the clutter is nonempty every row of A has at least one nonzero.
- If the i-th row of A has all nonzeros before j it is deleted in $M(u)$ (otherwise the empty row in the minor would make it infeasible, so u would not be a node in the BDD.)
- By the bandwidth limit, if $A_{ih} \neq 0$ for $h \geq j$, among the entries preceding h only those in $\{A_{i,h-(k-1)}, \dots, A_{i,h-1}\}$ can be nonzero.
- Since there are at most 2^{k-1} ways of differently assigning values to the $k-1$ variables directly preceding x_j, we can construct at most 2^{k-1} different deletion/contraction minors at layer L_j, limiting the BDD-width to 2^{k-1}.

Computing $\text{Prob}[\Phi_{\alpha}^{\leq} = 1]$: BDD to LP

Recursive definition of intermediate probabilities for 'scenarios sharing suffix' at node in layer e^*. Survival probabilities p_e for each event.

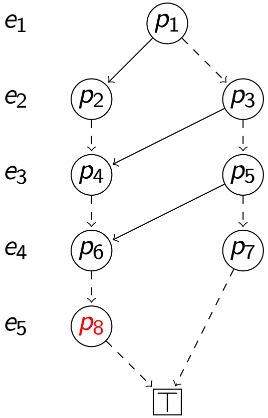


leaf $\top$:

$$\text{Prob}[\Phi(\xi) = 1] = 1$$

Computing $\text{Prob}[\Phi_\alpha^\leq = 1]$: BDD to LP

Recursive definition of intermediate probabilities for 'scenarios sharing suffix' at node in layer e^*. Survival probabilities p_e for each event.

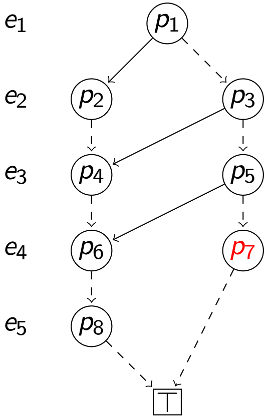


1-child node p_8:

$$\text{Prob}[\Phi(\xi) = 1] = (1 - p_{e_5})p_{\text{child}}$$

Computing $\text{Prob}[\Phi_{\alpha}^{\leq} = 1]$: BDD to LP

Recursive definition of intermediate probabilities for 'scenarios sharing suffix' at node in layer e^*. Survival probabilities p_e for each event.

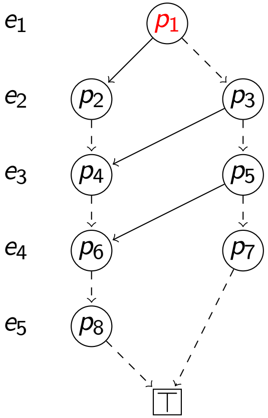


layer e_5 skipped below p_7:

$$
\begin{aligned}
&\text{Prob}[\Phi(\xi) = 1] \\
=&(1 - p_{e_4}) \\
&\cdot \left(\prod_{i=5}^{5} (p_{e_i} + (1 - p_{e_i})) \right) \cdot \\
&\cdot\, p_{\text{child}} \\
=&p_{e_4} p_{\text{child}}
\end{aligned}
$$

Computing $\mathrm{Prob}[\Phi_{\alpha}^{\leq} = 1]$: BDD to LP

Recursive definition of intermediate probabilities for 'scenarios sharing suffix' at node in layer e^*. Survival probabilities p_e for each event.



otherwise, e.g., p_1:

$$\mathrm{Prob}[\Phi(\xi) = 1]$$
$$= p_{e^*}\, p_{\text{TRUE-child}} + (1 - p_{e^*})\, p_{\text{FALSE-child}}$$

Computing Prob$[\Phi_\alpha^{\leq} = 1]$: BDD to LP

Recursive definition of intermediate probabilities for 'scenarios sharing suffix' at node in layer e^*. Survival probabilities p_e for each event.

Linear equations with $\mathcal{O}(\omega(BDD) \cdot |E|)$ auxilliary variables:

- leaf:

$$\text{Prob}[\Phi(\xi) = 1] = 1$$

- 1-child node (wlog: `True`-edge):

$$\text{Prob}[\Phi(\xi) = 1] = p_{e^*} p_{\text{child}}$$

- layers 2..$(l-1)$ skipped (wlog: `True`-edge):

$$\text{Prob}[\Phi(\xi) = 1] = p_{e^*} p_{\text{child}}$$

- otherwise:

$$\text{Prob}[\Phi(\xi) = 1] = p_{e^*} p_{\text{TRUE-child}} + (1 - p_{e^*}) p_{\text{FALSE-child}}$$

Consider binary decisions x_e such that

$$p_e(x) = \begin{cases} p_e & \text{if } x_e = 0, \\ p_e + \Delta_e & \text{if } x_e = 1 \end{cases}$$

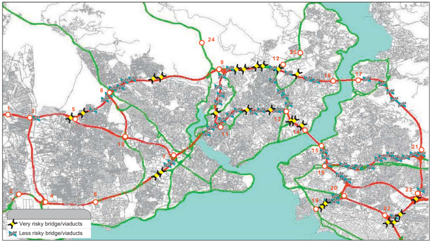
(where $\Delta_e \in [-p_e, 1 - p_e]$).

Then we can define for each arc $(u, v) \in A$ with label $\epsilon(u) = e$ of the BDD

$$p_{(u,v)}(x) = \begin{cases} p_e(x) & \text{if } (u, v) \in A, \epsilon(u) = e, l((u, v)) = 1 \\ (1 - p_e(x)) & \text{if } (u, v) \in A, \epsilon(u) = e, l((u, v)) = 0, \end{cases}$$

and write the computations as linear constraints coupled to the binaries x_e by big-M ($M = 1$).

Yields a MIP of size $4(\#\text{ BDD nodes}) \times ((\#\text{ BDD nodes}) + |E|)$.



Pre-disaster investment decisions for Istanbul road network from PEETA ET AL. '10 (30 decision variables)
(total construction time $< 1s$)

O-D-pair	cutoff dist	#bundles	MIP size (in PRESTWICH '13)	# BDDs	MIP size (using BDD bundles)
14–20	31	39		4	237×89
14–7	31	29		6	333×113
12–18	28	56		4	237×89
9–7	19	26		4	164×71
4–8	35	73		6	421×135
$\sum$		223	14174×6221	24	1466×454

No cutoff: BDD construction $< 1s$

O-D-pair	#bundles	MIP size (in PRESTWICH '13)	# BDDs	MIP size (using BDD bundles)
14–20	378		14	2609×682
14–7	712		30	13097×3304
12–18	233		8	997×1026
9–7	266		8	1137×314
4–8	305		12	2301×605
$\sum$	1894	123682×56851	72	20137×5064
MIP solve				36s

Generic application framework

… for 2-stage stochastic problems with scenario-monotone objective

Construct BDDs $B_\alpha^\leq$ for all $\alpha \in [L, U]$

Can be achieved by

- enumerating α-solutions, or
- enumerating α-cutsets, or
- circuit system oracle and equivalence check

BDD size may be bounded due to structural properties, e.g., bandwidth, treewidth, …

Build MIP

- linear in BDD size
- has network-flow flavor
- can accomodate an IP constraining the decisions

Generic application framework

... for 2-stage stochastic problems with scenario-monotone objective

Construct BDDs $B_{\alpha}^{\leq}$ for all $\alpha \in [L, U]$

Can be achieved by

- enumerating α-solutions, or
- enumerating α-cutsets, or
- circuit system oracle and equivalence check

BDD size may be bounded due to structural properties, e.g., bandwidth, treewidth, ...

Build MIP

- linear in BDD size
- has network-flow flavor
- can accomodate an IP constraining the decisions

Generic application framework

... for 2-stage stochastic problems with scenario-monotone objective

Construct BDDs $B_\alpha^\leq$ for all $\alpha \in [L, U]$

Can be achieved by

- enumerating α-solutions, or
- enumerating α-cutsets, or
- circuit system oracle and equivalence check

BDD size may be bounded due to structural properties, e.g., bandwidth, treewidth, ...

Build MIP

- linear in BDD size
- has network-flow flavor
- can accomodate an IP constraining the decisions

Applications

Expected shortest path with failing edges

… was the running example (interdiction problems also fit)

Expected assignment value with failing edges

crew/task assignments, assignment costs, probabilities of assignment failing (e.g., to complete task before deadline), decisions to invest in training crew members

Expected network flow with edge failures

network flow problem, edge capacities, probabilities of discrete edge capacity changes, decisions to influence edge capacities

Expected maximum clique size with edge failures

maximum clique problem, probabilities of edges failing, decisions to suppress or strengthen edges

Applications

Expected shortest path with failing edges

… was the running example (interdiction problems also fit)

Expected assignment value with failing edges

crew/task assignments, assignment costs, probabilities of assignment failing (e.g., to complete task before deadline), decisions to invest in training crew members

Expected network flow with edge failures

network flow problem, edge capacities, probabilities of discrete edge capacity changes, decisions to influence edge capacities

Expected maximum clique size with edge failures

maximum clique problem, probabilities of edges failing, decisions to suppress or strengthen edges

Applications

Expected shortest path with failing edges

... was the running example (interdiction problems also fit)

Expected assignment value with failing edges

crew/task assignments, assignment costs, probabilities of assignment failing (e.g., to complete task before deadline), decisions to invest in training crew members

Expected network flow with edge failures

network flow problem, edge capacities, probabilities of discrete edge capacity changes, decisions to influence edge capacities

Expected maximum clique size with edge failures

maximum clique problem, probabilities of edges failing, decisions to suppress or strengthen edges

Applications

Expected shortest path with failing edges

... was the running example (interdiction problems also fit)

Expected assignment value with failing edges

crew/task assignments, assignment costs, probabilities of assignment failing (e.g., to complete task before deadline), decisions to invest in training crew members

Expected network flow with edge failures

network flow problem, edge capacities, probabilities of discrete edge capacity changes, decisions to influence edge capacities

Expected maximum clique size with edge failures

maximum clique problem, probabilities of edges failing, decisions to suppress or strengthen edges