Hewlett Packard Enterprise

BINARY DECISION DIAGRAMS AND DISCRETE RELAXATIONS OF INTEGER PROGRAMMING PROBLEMS

Utz-Uwe Haus

MINOA ESR Days 2021-03-05

HPE HPC/AI EMEA Research Lab, Switzerland

OVERVIEW

1. Motivation: A stochastic combinatorial optimization problem that needs an manageable encoding of families of paths
2. Binary decision diagrams: Concept and some efficient constructions
3. Discrete Relaxations using BDDs

About ERL

HPE HPC/AI EMEA RESEARCH LAB

We're a European research group in HPE, working on the most challenging problems in high performance computing for the Exascale era and beyond.

Offices in Basel, Grenoble and Bristol.

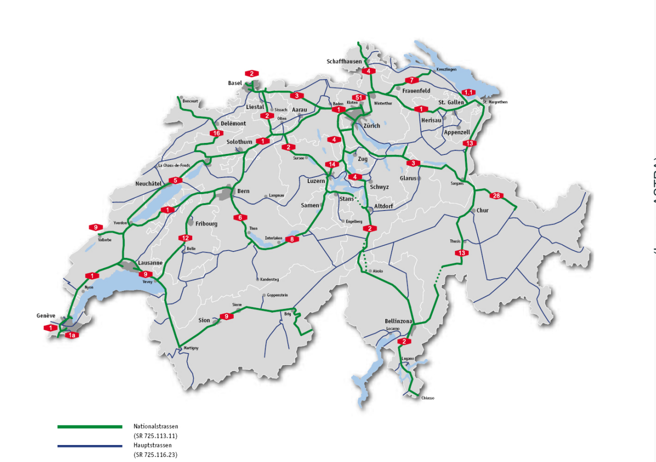
Part of the HPE HPC CTO office.

`https://hpe.com/emea_europe/en/compute/hpc/emea-research-lab.html`

Goals and Motivation

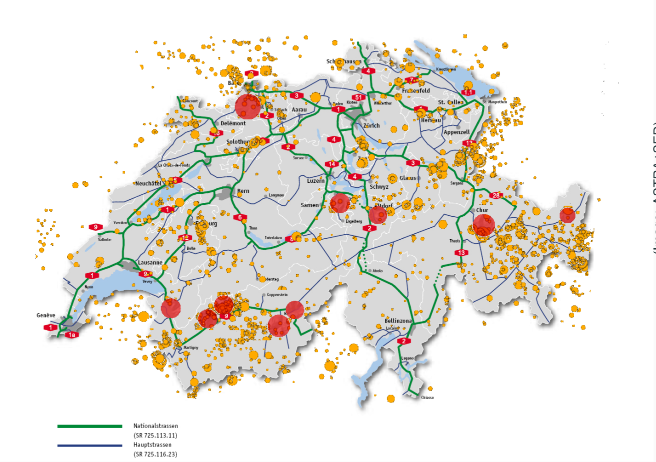
(H., Laumanns, Michini 2015)

INTRODUCTION: INFRASTRUCTURE RISK



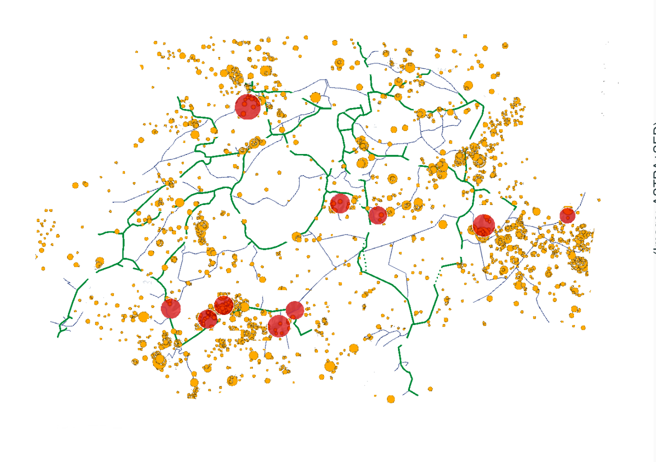
(Image: ASTRA)

INTRODUCTION: INFRASTRUCTURE RISK



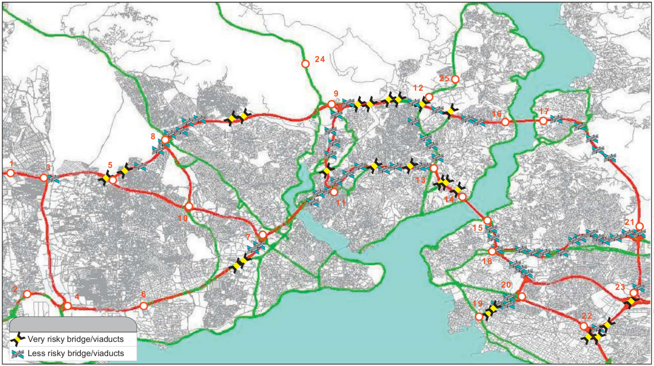
(Image: ASTRA+SED)

INTRODUCTION: INFRASTRUCTURE RISK



(Image: ASTRA+SED)

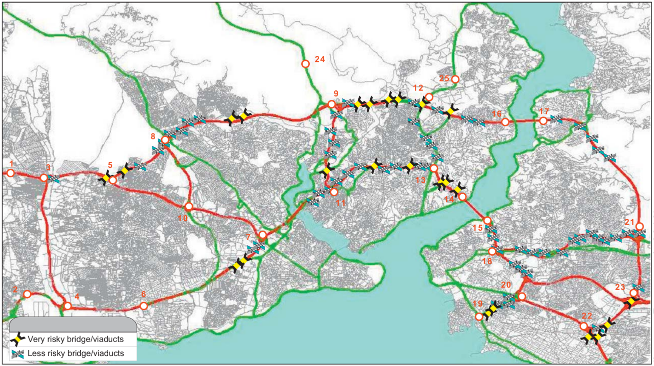
INTRODUCTION: INFRASTRUCTURE RISK



(Image: Peeta et al. 2010)

- Shortest paths between two locations
- Expected shortest path after quake
- Network stabilization? Network extension?

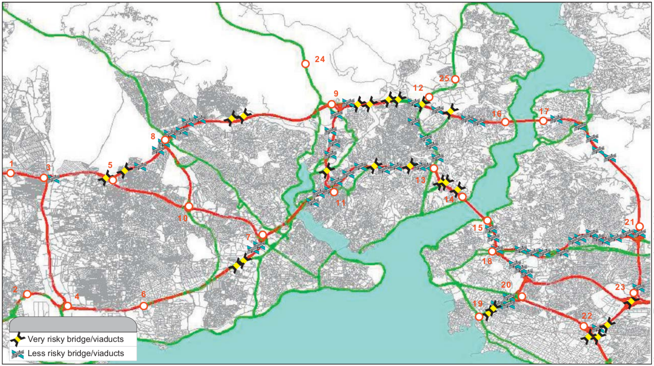
INTRODUCTION: INFRASTRUCTURE RISK



(Image: Peeta et al. 2010)

- Shortest paths between two locations
- Expected shortest path after quake
- Network stabilization? Network extension?

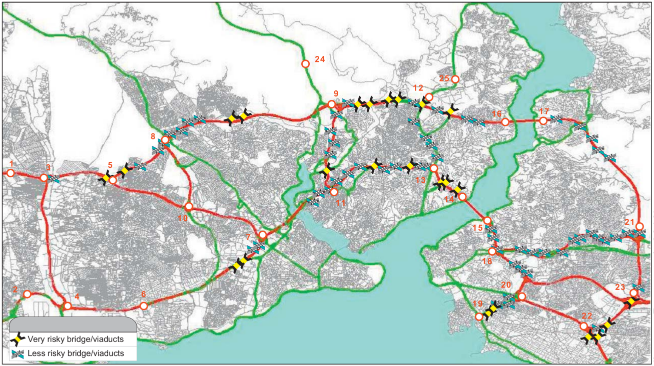
INTRODUCTION: INFRASTRUCTURE RISK



(Image: Peeta et al. 2010)

- Shortest paths between two locations
- Expected shortest path after quake
- Network stabilization? Network extension?

INTRODUCTION: INFRASTRUCTURE RISK



(Image: Peeta et al. 2010)

- Shortest paths between two locations
- Expected shortest path after quake
- Network stabilization? Network extension?

INTRODUCTION: INFRASTRUCTURE RISK

- Decision-dependent uncertainty in stochastic networks
  - Robustification of Infrastructure
  - Network interdiction problems
  - Project task networks: reducing risk on critical path
- Models
  - failure protection, cost-robustness
  - structural robustness
- modeling and solution approaches:
  - (worst-case-) robust optimization
  - models with failure tolerant feasibility

INTRODUCTION: INFRASTRUCTURE RISK

- Decision-dependent uncertainty in stochastic networks
  - Robustification of Infrastructure
  - Network interdiction problems
  - Project task networks: reducing risk on critical path
- Models
  - failure protection, cost-robustness
  - structural robustness
  - Pre-Disaster Investment Planning
- modeling and solution approaches:
  - (worst-case-) robust optimization
  - models with failure tolerant feasibility
  - (multistage) stochastic models

INTRODUCTION: INFRASTRUCTURE RISK

- Decision-dependent uncertainty in stochastic networks
  - Robustification of Infrastructure
  - Network interdiction problems
  - Project task networks: reducing risk on critical path
- Models
  - failure protection, cost-robustness
  - structural robustness
  - Pre-Disaster Investment Planning
- modeling and solution approaches:
  - (worst-case-) robust optimization
  - models with failure tolerant feasibility
  - (multistage) stochastic models

INTRODUCTION: INFRASTRUCTURE RISK

- Decision-dependent uncertainty in stochastic networks
  - Robustification of Infrastructure
  - Network interdiction problems
  - Project task networks: reducing risk on critical path
- Models
  - failure protection, cost-robustness
  - structural robustness
  - Pre-Disaster Investment Planning
- modeling and solution approaches:
  - (worst-case-) robust optimization
  - models with failure tolerant feasibility
  - (multistage) stochastic models

Endogenous robustness

NETWORKS WITH UNCERTAIN DATA

Setting

A network problem with uncertain cost, resources, or capacities: uncertainty region Δ

Approaches

- both worst-case ($\forall \delta \in \Delta$) and best-case ($\exists \delta \in \Delta$) are interesting
- $\forall$: reformulate and solve robust counterpart as LP/IP/...
- $\exists$: reformulate and solve as generalized LP(/IP/...)
- ... or separate robust/generalized (split-)cuts directly
- Caveat 1: reformulation often destroys combinatorial structure
- Caveat 2: even shortest path with 2 cost scenarios is (weakly) $\mathcal{NP}$-hard

NETWORKS WITH UNCERTAIN DATA

Setting
A network problem with uncertain cost, resources, or capacities: uncertainty region Δ

Approaches

- both worst-case ($\forall \delta \in \Delta$) and best-case ($\exists \delta \in \Delta$) are interesting
- $\forall$: reformulate and solve robust counterpart as LP/IP/...
- $\exists$: reformulate and solve as generalized LP(/IP/...)
- ... or separate robust/generalized (split-)cuts directly
- Caveat 1: reformulation often destroys combinatorial structure
- Caveat 2: even shortest path with 2 cost scenarios is (weakly) $\mathcal{NP}$-hard

NETWORKS WITH UNCERTAIN DATA

Setting
A network problem with uncertain cost, resources, or capacities: uncertainty region Δ

Approaches

- both worst-case ($\forall \delta \in \Delta$) and best-case ($\exists \delta \in \Delta$) are interesting
- $\forall$: reformulate and solve robust counterpart as LP/IP/...
- $\exists$: reformulate and solve as generalized LP(/IP/...)
- ... or separate robust/generalized (split-)cuts directly
- Caveat 1: reformulation often destroys combinatorial structure
- Caveat 2: even shortest path with 2 cost scenarios is (weakly) $\mathcal{NP}$-hard

NETWORKS WITH UNCERTAIN DATA

Setting
A network problem with uncertain cost, resources, or capacities: uncertainty region Δ

Approaches

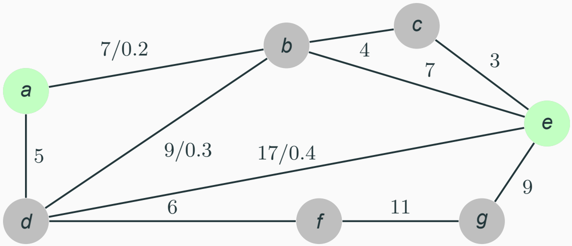
- both worst-case ($\forall \delta \in \Delta$) and best-case ($\exists \delta \in \Delta$) are interesting
- $\forall$: reformulate and solve robust counterpart as LP/IP/...
- $\exists$: reformulate and solve as generalized LP(/IP/...)
- ... or separate robust/generalized (split-)cuts directly
- Caveat 1: reformulation often destroys combinatorial structure
- Caveat 2: even shortest path with 2 cost scenarios is (weakly) $\mathcal{NP}$-hard

PRE-DISASTER INVESTMENT PLANNING

Example

Consider a graph $G = (V, E)$, edge lengths $(l_e)_{e \in E}$, and edge stability probabilities $(p_e)_{e \in E}$.

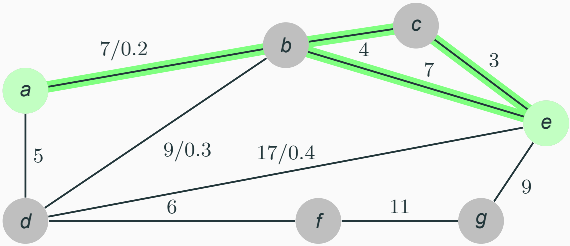
Scenarios: Sets of surviving edges after a disaster. Let $f_{\mathrm{SP}}(s, t, G_\xi)$ the shortest path length between nodes $s, t \in V$. Decisions $x_e = 1$ correspond to investments in edge stability to achieve $\tilde{p}_e = p_e + \delta_e$.



Example

Consider a graph $G = (V, E)$, edge lengths $(l_e)_{e \in E}$, and edge stability probabilities $(p_e)_{e \in E}$.

Scenarios: Sets of surviving edges after a disaster. Let $f_{\mathrm{SP}}(s, t, G_\xi)$ the shortest path length between nodes $s, t \in V$. Decisions $x_e = 1$ correspond to investments in edge stability to achieve $\tilde{p}_e = p_e + \delta_e$.



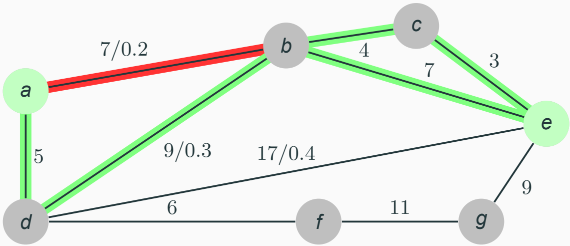
length: 14
2 paths

PRE-DISASTER INVESTMENT PLANNING

Example

Consider a graph $G = (V, E)$, edge lengths $(l_e)_{e \in E}$, and edge stability probabilities $(p_e)_{e \in E}$.

Scenarios: Sets of surviving edges after a disaster. Let $f_{\text{SP}}(s, t, G_\xi)$ the shortest path length between nodes $s, t \in V$. Decisions $x_e = 1$ correspond to investments in edge stability to achieve $\tilde{p}_e = p_e + \delta_e$.

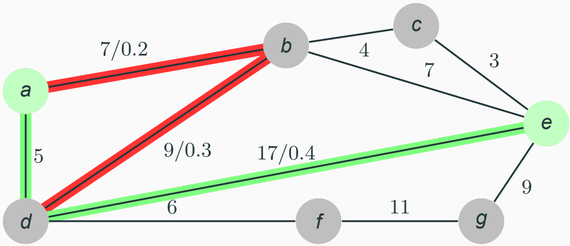


length: 21
2 paths
still
$(..., b, c, e)$ and
$(..., b, e)$

Example

Consider a graph $G = (V, E)$, edge lengths $(l_e)_{e \in E}$, and edge stability probabilities $(p_e)_{e \in E}$.

Scenarios: Sets of surviving edges after a disaster. Let $f_{\mathrm{SP}}(s, t, G_\xi)$ the shortest path length between nodes $s, t \in V$. Decisions $x_e = 1$ correspond to investments in edge stability to achieve $\tilde{p}_e = p_e + \delta_e$.



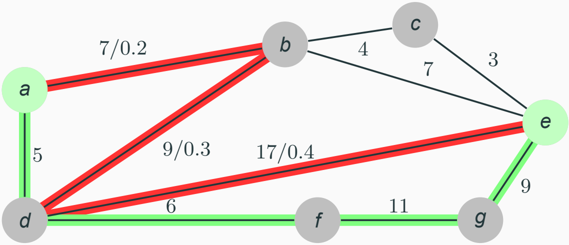
length: 22

1 path

Example

Consider a graph $G = (V, E)$, edge lengths $(l_e)_{e \in E}$, and edge stability probabilities $(p_e)_{e \in E}$.

Scenarios: Sets of surviving edges after a disaster. Let $f_{\text{SP}}(s, t, G_\xi)$ the shortest path length between nodes $s, t \in V$. Decisions $x_e = 1$ correspond to investments in edge stability to achieve $\tilde{p}_e = p_e + \delta_e$.



length: 31

1 path

PRE-DISASTER INVESTMENT PLANNING

Example

Consider a graph $G = (V, E)$, edge lengths $(l_e)_{e \in E}$, and edge stability probabilities $(p_e)_{e \in E}$.

Scenarios: Sets of surviving edges after a disaster. Let $f_{\text{SP}}(s, t, G_\xi)$ the shortest path length between nodes $s, t \in V$. Decisions $x_e = 1$ correspond to investments in edge stability to achieve $\tilde{p}_e = p_e + \delta_e$.

- 2^3 failure scenarios (different probabilities)
- paths and path lengths differ over scenario space
- Expected value of shortest path length?
- Investments to reduce this value?
- Can we reformulate a robust counterpart?
  - Caveat 1: reformulation often destroys combinatorial structure
  - Caveat 2: even shortest path with 2 cost scenarios is (weakly) $\mathcal{NP}$-hard

PRE-DISASTER INVESTMENT PLANNING

Example
Consider a graph $G = (V, E)$, edge lengths $(l_e)_{e \in E}$, and edge stability probabilities $(p_e)_{e \in E}$.
Scenarios: Sets of surviving edges after a disaster. Let $f_{\mathrm{SP}}(s, t, G_\xi)$ the shortest path length between nodes $s, t \in V$. Decisions $x_e = 1$ correspond to investments in edge stability to achieve $\tilde{p}_e = p_e + \delta_e$.

- 2^3 failure scenarios (different probabilities)
- paths and path lengths differ over scenario space
- Expected value of shortest path length?
- Investments to reduce this value?
- Can we reformulate a robust counterpart?
  - Caveat 1: reformulation often destroys combinatorial structure
  - Caveat 2: even shortest path with 2 cost scenarios is (weakly) $\mathcal{NP}$-hard

2-STAGE STOCHASTIC PROGRAM

Given is a weighted graph $G = (V, E, w)$ and investment costs c_e to harden/weaken each link, which increases/decreases its survival probability from some given q_e to r_e.

Decision structure

1. First stage decisions: link investments

$$x_e \in \{0, 1\} \text{ for all } e \in E$$

2. Second stage decisions: shortest/longest path, flow, ... with value

$$f(\xi)$$

in $G = (V, E \setminus \xi)$ under the realized scenario

$$\xi = (\xi_e)_{e \in E} \in \{0, 1\}^{|E|}$$

Problem: Minimize $\mathbb{E}_{\xi|x}[f(\xi)]$ subject to $c^\top x \leq B$ (B : given budget)

Difficulties:

- There are many $(2^{|E|})$ scenarios ξ.
- Scenario probabilities depend on the investment decisions x.

2-STAGE STOCHASTIC PROGRAM

Given is a weighted graph $G = (V, E, w)$ and investment costs c_e to harden/weaken each link, which increases/decreases its survival probability from some given q_e to r_e.

Decision structure

1. First stage decisions: link investments

$$x_e \in \{0, 1\} \text{ for all } e \in E$$

2. Second stage decisions: shortest/longest path, flow, ... with value

$$f(\xi)$$

in $G = (V, E \setminus \xi)$ under the realized scenario

$$\xi = (\xi_e)_{e \in E} \in \{0, 1\}^{|E|}$$

Problem: Minimize $\mathbb{E}_{\xi|x}[f(\xi)]$ subject to $c^\top x \leq B$ (B : given budget)

Difficulties:

- There are many $(2^{|E|})$ scenarios ξ.
- Scenario probabilities depend on the investment decisions x.

2-STAGE STOCHASTIC PROGRAM

Given is a weighted graph $G = (V, E, w)$ and investment costs c_e to harden/weaken each link, which increases/decreases its survival probability from some given q_e to r_e.

Decision structure

1. First stage decisions: link investments

$$x_e \in \{0, 1\} \text{ for all } e \in E$$

2. Second stage decisions: shortest/longest path, flow, ... with value

$$f(\xi)$$

in $G = (V, E \setminus \xi)$ under the realized scenario

$$\xi = (\xi_e)_{e \in E} \in \{0, 1\}^{|E|}$$

Problem: Minimize $\mathbb{E}_{\xi|x}[f(\xi)]$ subject to $c^\top x \leq B$ (B : given budget)

Difficulties:

- There are many ($2^{|E|}$) scenarios ξ.
- Scenario probabilities depend on the investment decisions x.

2-STAGE STOCHASTIC PROGRAM

Given is a weighted graph $G = (V, E, w)$ and investment costs c_e to harden/weaken each link, which increases/decreases its survival probability from some given q_e to r_e.

Decision structure

1. First stage decisions: link investments

$$x_e \in \{0, 1\} \text{ for all } e \in E$$

2. Second stage decisions: shortest/longest path, flow, ... with value

$$f(\xi)$$

in $G = (V, E \setminus \xi)$ under the realized scenario

$$\xi = (\xi_e)_{e \in E} \in \{0, 1\}^{|E|}$$

Problem: Minimize $\mathbb{E}_{\xi|x}[f(\xi)]$ subject to $c^\top x \leq B$ (B : given budget)

Difficulties:

- There are many ($2^{|E|}$) scenarios ξ.
- Scenario probabilities depend on the investment decisions x.

ENDOGENOUS UNCERTAINTY

Decision-dependent probabilities are a case of endogenous uncertainty, which is very difficult to handle in Stochastic Programming.

Expressions for the decision-dependent scenario probabilities

$$\overbrace{\mathbb{E}_{\xi|x}[f(\xi)]}^{\text{Expected shortest path length}} = \sum_{\tilde{\xi}} \overbrace{P\{\xi = \tilde{\xi} \mid x\}}^{\text{Probability for scenario}} \cdot \overbrace{f(\tilde{\xi})}^{\text{shortest path length}}$$

where the scenario probabilities are

$$P\{\xi = \tilde{\xi} \mid x\} = \prod_{e \in E} \left[\tilde{\xi}_e \overbrace{[(1 - x_e)q_e + x_e]}^{\text{survival prob}} + (1 - \tilde{\xi}_e) \overbrace{[(1 - x_e)(1 - q_e)]}^{\text{failure prob}} \right]$$

so polynomials in x of degree $|E|$.

How to deal with such non-linearities?

ENDOGENOUS UNCERTAINTY

Solution approaches

- scenario sampling/simulation
- Sample Average/Sample-Path: can yield statistically testable bounds
- partitioning or covering of the scenario space and exact reformulation

Example (cont.): Computation of 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 one can partition into sets with same f-value, whose probabilities can be computed. (Prestwich et al., '13/14)

Solution approaches

- scenario sampling/simulation
- Sample Average/Sample-Path: can yield statistically testable bounds
- partitioning or covering of the scenario space and exact reformulation

Example (cont.): Computation of 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 one can partition into sets with same f-value, whose probabilities can be computed. (Prestwich et al., '13/14)

ENDOGENOUS UNCERTAINTY

Solution approaches

- scenario sampling/simulation
- Sample Average/Sample-Path: can yield statistically testable bounds
- partitioning or covering of the scenario space and exact reformulation

Example (cont.): Computation of expected path length

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

Instead of enumerating $2^{|E|}$ scenarios one can partition into sets with same f-value, whose probabilities can be computed. (Prestwich et al., '13/14)

AGGREGABLE PROBLEMS

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 $\xi_1, \xi_2 \in 2^E$, and
- the probabilities of events $e \in E$ are independent.

We denote the image of f by

$$\mathcal{C}(f) = \{\alpha \,:\, \alpha = f(\xi), \xi \in 2^E\},$$

and the minimal scenarios for each critical value by

$$\mathcal{M}(f) = \{\mathcal{M}_\alpha(f) \,:\, \alpha \in \mathcal{C}(f)\},$$

where $\mathcal{M}_\alpha(f) = \{\xi \in 2^E \,:\, f(\xi) = \alpha, \forall \xi' \subset \xi : f(\xi') > f(\xi)\}$.

AGGREGABLE PROBLEMS

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 $\xi_1, \xi_2 \in 2^E$, and

- the probabilities of events $e \in E$ are independent.

We denote the image of f by

$$\mathcal{C}(f) = \{\alpha \, : \, \alpha = f(\xi), \xi \in 2^E\},$$

and the minimal scenarios for each critical value by

$$\mathcal{M}(f) = \{\mathcal{M}_\alpha(f) \, : \, \alpha \in \mathcal{C}(f)\},$$

where $\mathcal{M}_\alpha(f) = \{\xi \in 2^E \, : \, f(\xi) = \alpha, \forall \xi' \subset \xi : f(\xi') > f(\xi)\}$.

AGGREGABLE PROBLEMS

Relevant Examples
'tame': problems where $f(\xi)$ is computable in polynomial-time

- (multi-terminal-) shortest path
- number of edge-disjoint paths/*k*-connectivity
- longest paths in acyclic networks (critical path)
- maximal flows
- maximal/weight maximal matchings
- linear programs (with changing constraint set)

but also 'wild' ones

- clique number

Encoding Minimal Scenarios

MONOTONE BOOLEAN FUNCTIONS

Each $\mathcal{M}_\alpha(f)$ induces a monotone[1] Boolean function $\Phi_\alpha^{\leq}$ on the scenarios whose minimal true points are the elements of $\mathcal{M}_\alpha(f)$:

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

Encoding of $\Phi_\alpha^{\leq}$

- as DNF: using explicit list of $\mathcal{M}_\alpha(f)$
- as IP of covering-type: $p^\top x \geq 1 \ (\forall p \in \mathcal{M}_\alpha(f))$
- as BDD, constructed from explicit or implicit description of $\mathcal{M}_\alpha(f)$
- by using isomorphy of BDDs using the dual monotone Boolean function $\neg(\Phi_\alpha^{\leq}(\neg\xi))$

[1] A Boolean function f is (up-)**monotone** iff $\forall x \leq y : f(x) \leq f(y)$.

MONOTONE BOOLEAN FUNCTIONS

Each $\mathcal{M}_\alpha(f)$ induces a monotone[1] Boolean function $\Phi_\alpha^\leq$ on the scenarios whose minimal true points are the elements of $\mathcal{M}_\alpha(f)$:

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

Encoding of $\Phi_\alpha^\leq$

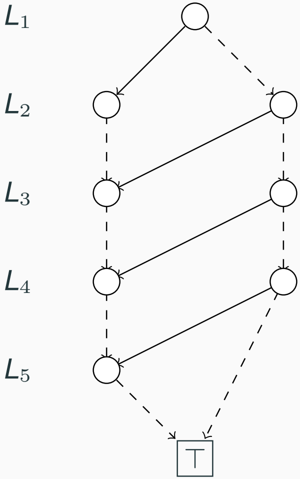
- as DNF: using explicit list of $\mathcal{M}_\alpha(f)$
- as IP of covering-type: $p^\top x \geq 1 \; (\forall p \in \mathcal{M}_\alpha(f))$
- as BDD, constructed from explicit or implicit description of $\mathcal{M}_\alpha(f)$
- by using isomorphy of BDDs using the dual monotone Boolean function $\neg(\Phi_\alpha^\leq(\neg\xi))$

[1] A Boolean function f is (up-)**monotone** iff $\forall x \leq y : f(x) \leq f(y)$.

BDDs

BINARY DECISION DIAGRAMS (BRYANT, 1986)

- layered (rooted) digraph
- arcs only between L_i and L_j with $j > i$
- every node has 1 or 2 out-arcs
- true-arcs —, false-arcs - -
- every path from root to $\top$ defines a feasible solution (or 'good' family)
- every path from root to inner node defines a partial solution
- no two sub-BDDs are isomorphic
- layer L_i has width $\omega_i = |L_i|$
- BDD-width $\omega = \max_i \omega_i$



Every logical formula can be represented in a BDD.

BDDS ENCODING MEMBERS OF AN INDEPENDENCE SYSTEMS

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, but not 0/1-knapsack)

The size of BDDs depends heavily on the ordering of the variables.

Aggregable Problems: Tools

FUNCTION IN CNF: 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:

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

FUNCTION IN CNF: 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:

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

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

FUNCTION IN CNF: 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:

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

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

FUNCTION IN CNF: 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 determines the same matrix minor

$\Rightarrow$ u and v can be merged!

Example:

$$\begin{array}{cccccccc}
x_1+ & & x_3+ & & & x_6 & \geq 1 \\
 & & & x_4+ & & x_6 & \geq 1 \\
 & x_2+ & & x_4+ & x_5 & & \geq 1 \\
x_1+ & x_2+ & x_3+ & & & & \geq 1 \\
 & & x_3+ & x_4+ & x_5 & & \geq 1 \\
\multicolumn{3}{l}{x \in \{0,1\}^n} & & & & &
\end{array}$$

FUNCTION IN CNF: 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:

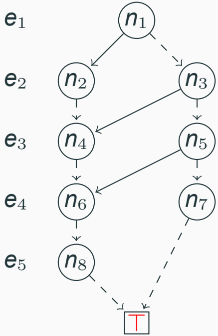
$$\begin{array}{lllllll} x_1+ & & x_3+ & & & x_6 & \geq 1 \\ & & & x_4+ & & x_6 & \geq 1 \\ & x_2+ & & x_4+ & x_5 & & \geq 1 \\ x_1+ & x_2+ & x_3+ & & & & \geq 1 \\ & & x_3+ & x_4+ & x_5 & & \geq 1 \\ x \in \{0,1\}^n & & & & & & \end{array}$$

(DNF of $\Phi_\alpha^{\leq}$ yields CNF of dual function for free: BDD only needs swap of arc types)

Pre-Disaster Investment Planning

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

Recursive definition of intermediate probabilities for 'scenarios with common suffix' in nodes in layer e^*. Survival probabilities p_e for every event.

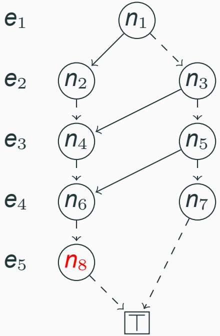


leaf $\top$:

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

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

Recursive definition of intermediate probabilities for 'scenarios with common suffix' in nodes in layer e^*. Survival probabilities p_e for every event.

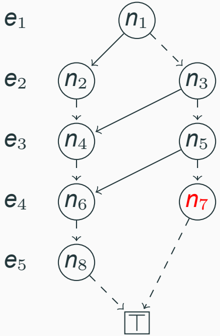


1-child node n_8:

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

COMPUTING $\mathrm{Prob}[\Phi^{\leq}_{\alpha} = 1]$: BDD TO LP

Recursive definition of intermediate probabilities for ‘scenarios with common suffix’ in nodes in layer e^*. Survival probabilities p_e for every event.

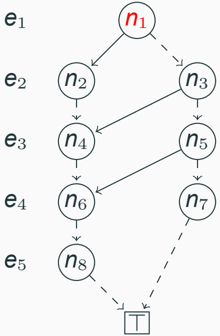


layer e_5 skipped below n_7:

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

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

Recursive definition of intermediate probabilities for ‘scenarios with common suffix’ in nodes in layer e^*. Survival probabilities p_e for every event.



otherwise, e.g. n_1:

$$\begin{aligned}&\text{Prob}[\Phi(\xi) = 1]\\ =&p_{e^*} p_{\text{True-child}} + (1 - p_{e^*}) p_{\text{False-child}}\end{aligned}$$

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

Recursive definition of intermediate probabilities for 'scenarios with common suffix' in nodes in layer e^*. Survival probabilities p_e for every event.

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

- leaf: $\mathrm{Prob}[\Phi(\xi) = 1] = 1$
- 1-child node (wlog: `True`-arc): $\mathrm{Prob}[\Phi(\xi) = 1] = p_{e^*} p_{\text{child}}$
- layers $2..(l-1)$ skipped (wlog: `True`-arc):
  $\mathrm{Prob}[\Phi(\xi) = 1] = p_{e^*} p_{\text{child}}$
- else: $\mathrm{Prob}[\Phi(\xi) = 1] = p_{e^*} p_{\text{True-child}} + (1 - p_{e^*}) p_{\text{False-child}}$

COMPUTING $\mathrm{Prob}[\Phi_\alpha^\leq = 1|X]$: BDD TO MIP

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

Define for every arc $(u, v) \in A$ of the BDD with label $\epsilon(u) = e$

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

and write the computations as linear inequalities, coupled with binaries x_e using big-M ($M = 1$).

Yields an (exact reformulation) MIP of size $4(\#\text{ BDD-nodes}) \times ((\#\text{ BDD-nodes}) + |A|)$.

Discrete relaxations

DISCRETE RELAXATIONS

Continuous Relaxations

- ignore integrality requirements
- ignore some constraints and/or use weaker valid constraints
- obtain a well-understood (continuous) problem (LP, convex, SDP, ...) with feasible region including original one

Yield valid bounds

Discrete Relaxations

- **preserve** integrality requirements
- ignore some constraints and/or use weaker valid constraints
- obtain a well-understood (discrete) problem class with feasible region including original one

Yield valid bounds
Yield candidates for feasible solutions or cuts

See primal reformulation techniques for inspirations.

DISCRETE RELAXATIONS

Continuous Relaxations

- ignore integrality requirements
- ignore some constraints and/or use weaker valid constraints
- obtain a well-understood (continuous) problem (LP, convex, SDP, ...) with feasible region including original one

Yield valid bounds

Discrete Relaxations

- **preserve** integrality requirements
- ignore some constraints and/or use weaker valid constraints
- obtain a well-understood (discrete) problem class with feasible region including original one

Yield valid bounds
Yield candidates for feasible solutions or cuts

See primal reformulation techniques for inspirations.

DISCRETE RELAXATIONS

Continuous Relaxations

- ignore integrality requirements
- ignore some constraints and/or use weaker valid constraints
- obtain a well-understood (continuous) problem (LP, convex, SDP, ...) with feasible region including original one

Yield valid bounds

Discrete Relaxations

- **preserve** integrality requirements
- ignore some constraints and/or use weaker valid constraints
- obtain a well-understood (discrete) problem class with feasible region including original one

Yield valid bounds
Yield candidates for feasible solutions or cuts

See primal reformulation techniques for inspirations.

BDDS AS DISCRETE RELAXATIONS

Sources of relaxations

- combinatorial inequalities: packing, covering, partitioning
- logical constraints: disjunctions, SOSs, among, ...
- knapsacks and packing 0/1 IPs (Behle, 2007)
- unions and intersections of BDDs

Using BDDs

- Maximize linear objective over BDD: shortest path in acyclic graph
- Find candidate for feasible solution: one root $\rightarrow \top$ path in BDD
  - check feasibility in original problem: harvest solution or create cut
  - BDD be amended to exclude the solution easily
- Identify don't-care variable: Find a layer in BDD that has no nodes

For non-binary cases there are MDDs too

BDDS AS DISCRETE RELAXATIONS

Sources of relaxations

- combinatorial inequalities: packing, covering, partitioning
- logical constraints: disjunctions, SOSs, among, ...
- knapsacks and packing 0/1 IPs (Behle, 2007)
- unions and intersections of BDDs

Using BDDs

- Maximize linear objective over BDD: shortest path in acyclic graph
- Find candidate for feasible solution: one root $\rightarrow \top$ path in BDD
  - check feasibility in original problem: harvest solution or create cut
  - BDD be amended to exclude the solution easily
- Identify don't-care variable: Find a layer in BDD that has no nodes

For non-binary cases there are MDDs too

BDDS AS DISCRETE RELAXATIONS

Sources of relaxations

- combinatorial inequalities: packing, covering, partitioning
- logical constraints: disjunctions, SOSs, among, ...
- knapsacks and packing 0/1 IPs (Behle, 2007)
- unions and intersections of BDDs

Using BDDs

- Maximize linear objective over BDD: shortest path in acyclic graph
- Find candidate for feasible solution: one root $\to \top$ path in BDD
  - check feasibility in original problem: harvest solution or create cut
  - BDD be amended to exclude the solution easily
- Identify don't-care variable: Find a layer in BDD that has no nodes

For non-binary cases there are MDDs too

Summary

SUMMARY

Ingredients

- non-dominated solution of monotone optimization problem
- BDD encoding of discrete sets
- sometimes: output-linear time construction of BDDs
- always: use of BDDs for computation

QUESTIONS?

uhaus@hpe.com