Branch-and-Cut for Piecewise Linear Optimization and Extensions

Ismael Regis de Farias JR.

Department of Industrial Engineering

Texas Tech University

Jointly with Ming Zhao (SAS), Rajat Gupta (TTU) and Ernée Kozyreff (TTU)

Summary

- Problem definition
- Modeling alternatives
- Valid inequalities
- Intersection with semi-continuous constraint
- Computational results
- Further research

Problem definition

maximize $f_1(x_1) + \cdots + f_n(x_n)$

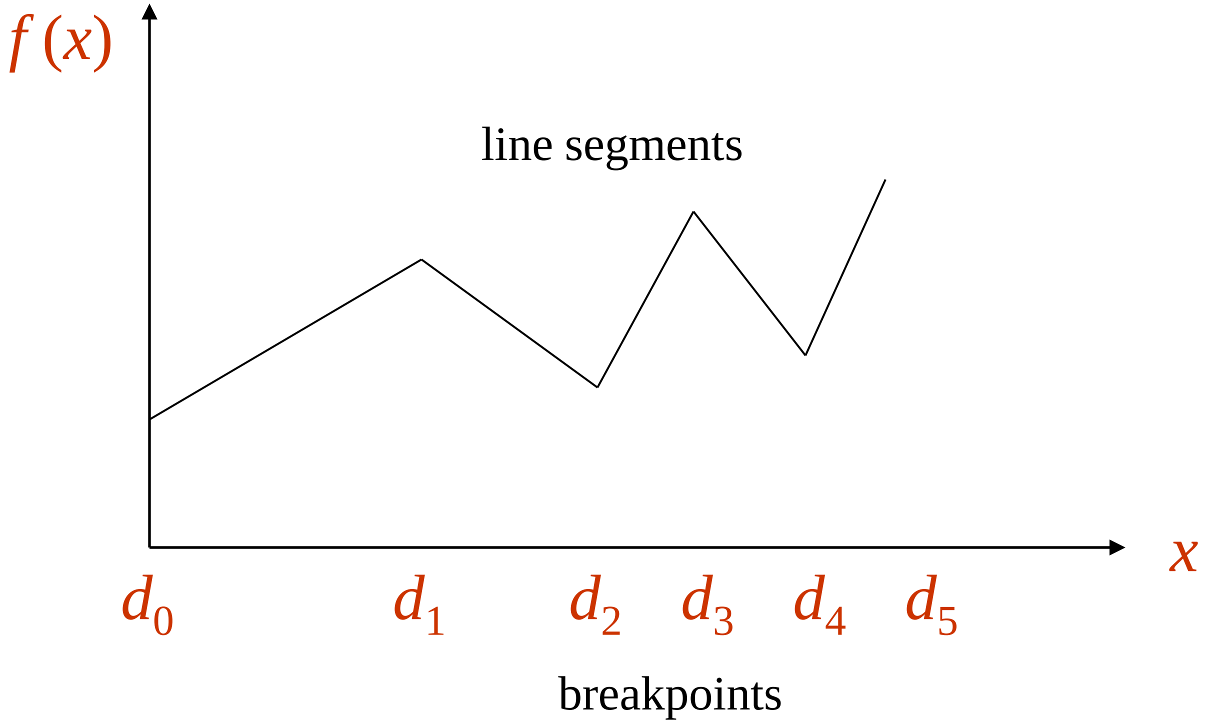
subject to

$$A\,x \le b$$

$$x \ge 0,$$

where $f_j(x_j)$ is a continuous piecewise linear function $\forall j$. We assume that some of the f_j's are nonconcave.

Problem definition



Applications

- Approximation of nonlinear functions (Bazaraa, Sherali, Shetty 2006)
- Network optimization with economies of scale (Ahuja, Magnanti, Orlin 1993)
- Auctions (Sandholm 2007)
- Gas network optimization (Martin, Möller, Moritz 2006)
- Portfolio optimization (Perold 1984)

SOS2

The set $\{\lambda_1, \ldots, \lambda_T\}$ is SOS2 if:

- at most 2 variables are allowed to be nonzero, and
- if 2 variables are nonzero, they must be adjacent in the set.

SOS2

Note that SOS2 is more general than it seems. For example, it can be used to enforce:

- multiple-choice
- semi-continuous
- general integer

constraints

Enforcing semi-continuous

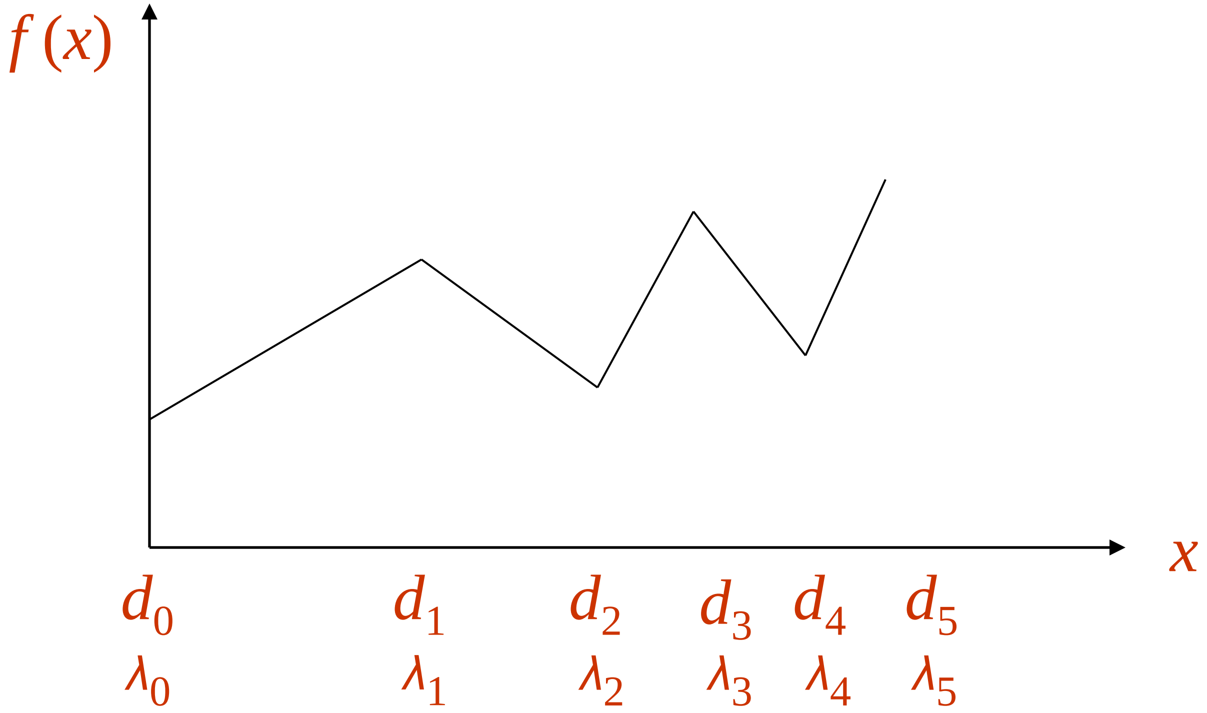
Let $x \in \{0\} \cup [1, 2]$.

1. Build the SOS2 $\{\lambda_0, \lambda, \lambda_1, \lambda_2\}$
2. Substitute $x = 0 \cdot \lambda_0 + ½ \cdot \lambda + 1 \cdot \lambda_1 + 2 \cdot \lambda_2$
3. Fix $\lambda = 0$

Goal

To turn a MILP solver into a general MINLP and NLP solver. In particular, NLP with structure.

Basic model



$f(x)$
x
d_0 d_1 d_2 d_3 d_4 d_5
λ_0 λ_1 λ_2 λ_3 λ_4 λ_5

Basic model

$$x = \sum_{k \in K} d_k \lambda_k \quad (K = \{1, \ldots, T\})$$

$$f(x) = \sum_{k \in K} f(d_k)\, \lambda_k$$

$$\sum_{k \in K} \lambda_k = 1$$

$$\lambda \geq 0$$

$$\{\lambda_1, \ldots, \lambda_T\} \text{ is SOS2}$$

We give new cutting planes for piecewise linear optimization implied by the following underlying knapsack set:

Underlying knapsack set

$$\sum_{j \in N^+} \sum_{k \in K} a_j^k \lambda_j^k - \sum_{j \in N^-} \sum_{k \in K} a_j^k \lambda_j^k \leq b \tag{6}$$

$$\sum_{k=1}^{T} \lambda_j^k \leq 1 \ \forall j \in N \tag{7}$$

$$\lambda_j^k \geq 0 \ \forall j \in N, \ \forall k \in \{0, \ldots, T\} \tag{8}$$

$$\{\lambda_j^0, \ldots, \lambda_j^T\} \text{ satisfies } SOS2' \ \forall j \in N. \tag{9}$$

Underlying knapsack set

- $S = \{ \lambda \in \Re^{nT} : \ \lambda \text{ satisfies } (6) - (9) \}$
- $P = \text{conv}\,(S)$
- We refer to (7) as convexity constraints
- $N = N^{+} \cup N^{-}$
- $a_{j1} > \cdots > a_{jT} > 0$

Approaches to the basic model

- Incremental cost (Markowitz and Manne 1957)
- Convex combination (Dantzig 1961; equivalent to incremental cost, see Keha, de Farias, Nemhauser 2004; we will call it MIP)
- Special ordered set of type 2 (SOS2, Beale and Tomlin 1970)
- Log (Vielma and Nemhauser 2009 and Vielma Ahmed and Nemhauser 2010)

Cutting planes

- Convexity constraint cutting planes
- Cover inequality cutting planes

But… do we really need such generic cuts? Aren't they (and more) already present in CPLEX, or GUROBI, or Xpress, or your favorite solver?

Computation

- We tested transportation and transshipment optimization problems with concave objective function
- Instances generated as in Keha at al. (2006)
- We used Texas Tech High Performance Computing Center nodes running GUROBI 3 Callable Library
- We limited CPU time to 1 hour for transportation and 2 hours for transshipment

Characteristics of the problem

- The transportation instances varied in size from 25 supply, 50 demand nodes and 7 breakpoints to 100 supply, 400 demand nodes and 22 breakpoints
- The transshipment instances varied in size from 15 to 100 nodes, and 7 to 22 breakpoints
- Integrality gap is extremely small

Do we need new (generic) cutting planes? (Transshipment tests)

# Nodes & part.	Time default	Time B&B
30 & 6	1,853	1,074
30 & 10	3,286	2,844
30 & 15	3,325	3,142
40 & 10	5,089	5,383
50 & 6	7,200	7,077
60 & 4	6,685	7,200
70 & 3	5,771	7,200
70 & 5	7,200	7,200
80 & 5	7,200	7,200

Do we need new (generic) cutting planes? (Transshipment tests)

# Nodes & part.	Time default	Time w/ cuts
30 & 6	1,853	81
30 & 10	3,286	119
30 & 15	3,325	299
40 & 10	5,089	524
50 & 6	7,200	871
60 & 4	6,685	707
70 & 3	5,771	289
70 & 5	7,200	3,132
80 & 5	7,200	4,948

Do we need new (generic) cutting planes? (Transportation tests)

#Nodes & part.	Time default	Time B&B
25 × 50 & 5	936	1,286
25 × 100 & 5	971	1,452
25 × 200 & 5	2,578	3,290
25 × 300 & 5	3,600	3,600
25 × 400 & 5	3,600	3,200
50 × 100 & 5	171	282
50 × 200 & 5	272	232
50 × 300 & 5	617	630
50 × 400 & 5	1,754	2,230

Do we need new (generic) cutting planes? (Transportation tests)

#Nodes & part.	Time default	Time w/ cuts
25 × 50 & 5	936	18
25 × 100 & 5	971	34
25 × 200 & 5	2,578	101
25 × 300 & 5	3,600	103
25 × 400 & 5	3,600	479
50 × 100 & 5	171	37
50 × 200 & 5	272	43
50 × 300 & 5	617	99
50 × 400 & 5	1,754	139

Cutting planes: previous work (Keha, de Farias, Nemhauser 2006)

Two families of valid inequalities:

- lifted convexity constraint
- lifted cover inequality

Computation:

- performed with MINTO
- cutting planes tremendously effective, in SOS2 and MIP
- clear best option is SOS2 branching with the three cuts

Cutting planes: new contribution

- New families of inequalities
- The inequalities of Keha et al. are special cases of the new inequalities
- Extension to intersecting with semi-continuous variables
- New computational analysis

Lifted convexity constraints 1

Let $N_1^- \subseteq N^-$ *and* $b' = b + \sum_{i \in N_1^-} a_i^{m_i}$, *where* $m_i \in K$ *for* $i \in N_1^-$.

$I = \left\{ i \in N^+ - \{j\} : a_j^s + a_i^T > b' \right\}$, and $k_i = \min \left\{ k \in K : a_j^s + a_i^k > b' \right\} \ \forall i \in I$.

The inequality:

$$\frac{1}{a_j^s} \sum_{k=1}^{s-1} a_j^k \lambda_j^k + \sum_{k=s}^{T} \lambda_j^k + \sum_{i \in I} \sum_{k=\max\{1, k_i - 1\}}^{T} \alpha_i^k \lambda_i^k - \sum_{i \in N_1^-} \sum_{k=m_i+1}^{T} \beta_i^k \lambda_i^k - \sum_{i \in N^- - N_1^-} \sum_{k \in K} \frac{a_i^k}{a_j^s} \lambda_i^k \le 1$$

is valid for *P*, where:

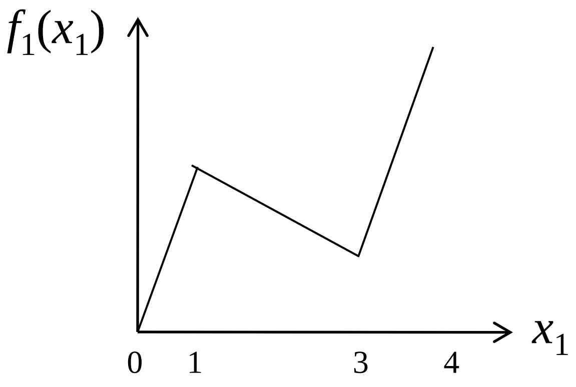
$$(\alpha_i^{k_i-1}, a_i^{k_i}) \in \left\{ (0,0), \left(\frac{a_j^s + a_i^{k_i-1} - b'}{a_j^s}, \frac{a_j^s + a_i^{k_i} - b'}{a_j^s} \right) \right\} \ \forall i \in I \ \textit{with} \ k_i > 1 \ \textit{and} \ a_j^s + a_i^{k_i-1} < b',$$

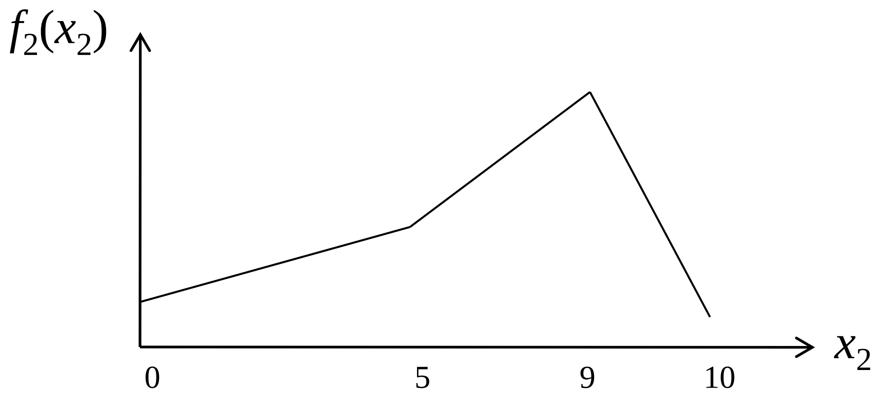
$$(\alpha_i^{k_i-1}, a_i^{k_i}) = \left(0, \frac{a_j^s + a_i^{k_i} - b'}{a_j^s} \right) \ \forall i \in I \ \textit{with} \ k_i > 1 \ \textit{and} \ a_j^s + a_i^{k_i-1} = b',$$

$$\alpha_i^{k_i} = 0 \ \forall i \in I \ \textit{with} \ k_i = 1,$$

$$\alpha_i^k = \frac{a_j^s + a_i^k - b'}{a_j^s} \ \forall i \in I \ \textit{with} \ k > k_i, \ \textit{and} \ \beta_i^k = \frac{a_i^k - a_i^{m_i}}{a_j^s}.$$

Example





$$2\,x_1 + x_2 \leq 10$$

$$(0 \cdot \lambda_1{}^0 + 2\,\lambda_1{}^1 + 6\,\lambda_1{}^2 + 8\,\lambda_1{}^3) + (0 \cdot \lambda_2{}^0 + 5\,\lambda_2{}^1 + 9\,\lambda_2{}^2 + 10\,\lambda_2{}^3) \leq 10$$

Example

The point:

$$\lambda_{12} = 5/6,\ \lambda_{21} = 1,\ \lambda_{ij} = 0 \text{ otherwise}$$

is an extreme point of the LP relaxation that is cut off by:

$$-3\,\lambda_{11} + \lambda_{12} + 3\,\lambda_{13} + 5\,\lambda_{21} + 5\,\lambda_{22} + 5\,\lambda_{23} \leq 5$$

Lifted convexity constraints 2

Suppose there exists $L \subseteq I$ such that $a_l^{k_l - 1} + a_j^s \geq b'$ and $k_l > 1 \; \forall l \in L$.

The inequality:

$$\sum_{k=1}^{s-1} \gamma_j^k \lambda_j^k + \sum_{k=s}^{T} \lambda_j^k + \sum_{i \in L} \sum_{k=k_i}^{T} \alpha_i^k \lambda_i^k - \sum_{i \in N_1^-} \sum_{k=m_i+1}^{T} \beta_i^k \lambda_i^k - \sum_{i \in N^- - N_1^-} \sum_{k \in K} \gamma_i^k \lambda_i^k \leq 1$$

is valid for *P*, where:

$$\alpha_i^k = \frac{a_i^k}{b' - a_L} \; \forall i \in I \;\; with \; k \geq k_i,$$

$$\beta_i^k = \frac{a_i^k - a_i^{m_i}}{b' - a_L} \; \forall i \in N_1^- \;\; with \; k > m_i$$

$$\gamma_i^k = \frac{a_i^k}{b' - a_L} \; \forall i \in N^- \cup \{j\} - N_1^- \;\; with \; k \in K$$

$$a_L = \min\{a_l^{k_l - 1} : \forall l \in L\}.$$

Cover

Definition ($N = N^+$):

Let $2 \le l_j \le T \ \forall j \in N$ and $C \subseteq N$ be such that

$$\sum_{j \in C} a_j^{l_j} = b + \rho$$

where $\rho > 0$. The set C is a cover.

Definition :

Let $C^+ \subseteq N^+$, $C^- \subseteq N^-$, $2 \le l_j \le u \ \forall j \in C^+$, $1 \le l_j \le u \ \forall j \in C^-$, and $C = C^+ \cup C^-$. If

$$\sum_{j \in C+} a_j^{l_j} - \sum_{j \in C-} a_j^{l_j} = b + \rho$$

with $\rho > 0$, C is a generalized cover.

Cover inequality

Let $u_j^k = a_j^k - a_j^{k-1} \; \forall j \in N,\, k \in K$

Let C be a cover and C_1, C_2 is a partition of C, such that

$$C_1 \subseteq \left\{ j \in C : l_j \geq 3 \text{ and } a_j^{l_j - 1} > b - \sum_{i \in C - \{j\}} a_i^{l_i} \right\}.$$

The cover inequality

$$\sum_{j \in C_1} (\alpha_j \lambda_j^{l_j - 2} + \beta_j \lambda_j^{l_j - 1} + \lambda_j^{l_j}) + \sum_{j \in C_2} (\gamma_j \lambda_j^{l_j - 1} + \lambda_j^{l_j}) \leq |C| - 1$$

is valid, where

$$\gamma_j = \min \left\{ 0, \frac{\rho - u_j^{l_j}}{\rho} \right\}, \;\; \beta_j = \frac{\rho - u_j^{l_j}}{\rho} \text{ and } \alpha_j = \min \left\{ 0, \frac{\rho - u_j^{l_j} - u_j^{l_j - 1}}{\rho} \right\}.$$

Lifted cover inequality

We have inequalities ($N = N^+$):

$$\sum_{j\in C}(\gamma_j\lambda_j^{l_j-1}+\sum_{k=l_j}^{T}\lambda_j^k)+\sum_{j\in N-C}\sum_{k=u_j}^{T}\lambda_j^k \le |C|-1 \qquad \text{(LCI1)}$$

and

$$\sum_{j\in C_1}(\alpha_j\lambda_j^{l_j-2}+\beta_j\lambda_j^{l_j-1}+\sum_{j=l_j}^{T}\lambda_j^{l_j})+\sum_{j\in C_2}(\gamma_j\lambda_j^{l_j-1}+\sum_{j=l_j}^{T}\lambda_j^{l_j})\le |C|-1 \qquad \text{(LCI2)}$$

Also in general, we have

$$\sum_{j\in C_1}(\alpha_j\lambda_j^{l_j-2}+\beta_j\lambda_j^{l_j-1}+\sum_{j=l_j}^{T}\lambda_j^{l_j})+\sum_{j\in C_2}(\gamma_j\lambda_j^{l_j-1}+\sum_{j=l_j}^{T}\lambda_j^{l_j})-\sum_{j\in N^-}(\tau_j\lambda_j^{l_j+1}+\sum_{j=l_j+2}^{T}\lambda_j^{k})\le \left|C^+\right|-1 \qquad \text{(GLCI)}$$

with $\tau_j = \max\{1, \frac{u_j^{l_j+1}}{\rho}\}.$ and $l_j = 0 \;\forall j \in N^- - C^-$

Summary of cutting planes results

- Regardless of the formulation (MIP, LOG, or SOS) the vast majority of the instances of either transportation or transshipment could not be solved by GUROBI in default setting
- Virtually all instances are solved through proven optimality with the cuts
- For the instances GUROBI could solve without our cuts, the average reduction in computational time is of 92% and in nodes 98%

Summary of cutting planes results

- For very large SOS2's (40 elements or above), the cuts were not efficient
- However, they were very efficient for the Vielma-Nemhauser instances

Formulation

- In the clear majority of cases SOS was better than MIP
- In some cases SOS and MIP were the best
- But in the vast majority of cases LOG was the best. Why? Is this due to MIP cutting planes? Preprocessing? Primal heuristic? Or just branching implementation?

Formulation

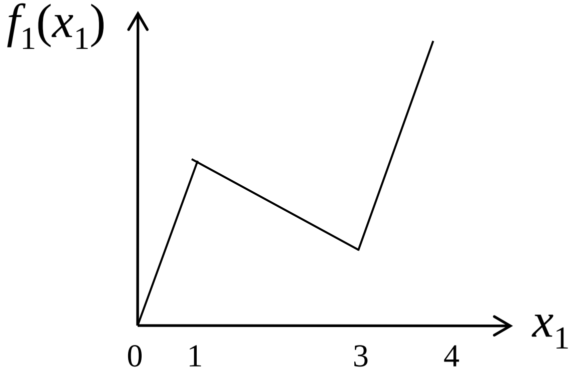
We hope that LOG breaks the symmetry of the network formulations. However, it could very well be that LOG's advantage is due to branching implementation.

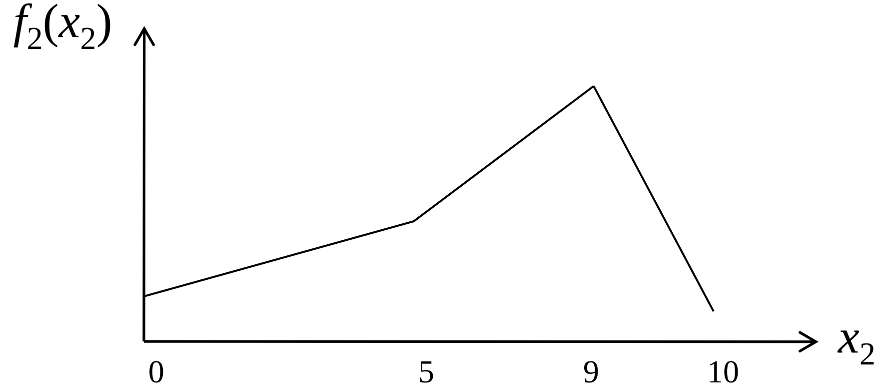
Formulation

We note that, regarding the Vielma-Nemhauser instances:

- B&B and default gave virtually the same results
- while with CPLEX LOG was considerably superior to SOS2, with GUROBI they were virtually the same
- with the PLO cuts, LOG and SOS2 were virtually the same

PLO with semi-continuous constraints





$$2\,x_1 + x_2 \leq 10$$

$$(2\,\lambda_1^1 + 6\,\lambda_1^2 + 8\,\lambda_1^3) + (5\,\lambda_2^1 + 9\,\lambda_2^2 + 10\,\lambda_2^3) \leq 10$$

Suppose now that $x_1 \in [0, 1] \cup [3, 4]$

Semi-continuous constraints

- The point $\lambda_{11} = 1/4, \lambda_{12} = 3/4, \lambda_{21} = 1, \lambda_{ij} = 0$ otherwise is an extreme point of P that does not satisfy the semi-continuous constraint
- We then add an artificial breakpoint with variable λ between λ_{11} and λ_{12}, with coefficient, say 3
- We obtain the lifted convexity constraint:

$$-2\,\lambda + \lambda_{12} + 3\,\lambda_{13} + 5\,\lambda_{21} + 5\,\lambda_{22} + 5\,\lambda_{23} \leq 5$$

- We fix $\lambda = 0$, and the resulting inequality

$$\lambda_{12} + 3\,\lambda_{13} + 5\,\lambda_{21} + 5\,\lambda_{22} + 5\,\lambda_{23} \leq 5$$

cuts off the point

Semi-continuous computation

- We tested transportation with the constraint $x \in \{0\} \cup [d_1, d_T]$ for all variables x
- The semi-continuous constraints made the problem considerably harder. We were able to solve only small instances, even with the piecewise linear cuts

Semi-continuous computation

Inst. size	Default	Time SC cut	Time PL cut
10×20 & 10	9	4	5
5×20 & 5	1,518	42	32
5×20 & 10	62	10	18
7×14 & 5	143	45	27
7×14 & 10	158	52	79
8×16 & 5	7,200	2,098	1,770
8×16 & 10	7,200	821	1,486
10×20 & 5	7,200	5,809	5,877
10×20 & 10	7,200	5,872	6,207

Further research

Piecewise linear optimization for:

- MINLP
- NLP
- NLP with structure, e.g. cardinality constraint
- MILP