Cutting Planes for Some Nonconvex Combinatorial Optimization Problems

Ismael Regis de Farias Jr.
Department of Industrial Engineering
Texas Tech

Summary

- Problem definition
- Solution strategy
- Multiple-choice
- Piecewise linear optimization
- Further research

Problem definition

Given $c_{1\times n}$, $A_{m\times n}$, and $b_{m\times 1}$, find $x_{n\times 1}$ that maximizes

$$cx$$

subject to

$$Ax \le b,$$

$$x \ge 0,$$

and complicating combinatorial constraints

Examples of combinatorial constraints

- Special ordered sets of type I:
  - at most one variable in the set can be positive, e.g. $\{x_1, x_2, x_3, x_4, x_5\}$ and $\{x_6, x_7, x_8\}$
  - LCP, capital budgeting, variable upper bound flows with GUBS
- Cardinality Constraints:
  - at most K variables can be positive
  - portfolio optimization, location-allocation

Examples of combinatorial constraints

- Semi-continuous variables:
  - $x = 0$ or $l \leq x \leq u$
  - portfolio optimization, synthesis of process networks
- Special ordered sets of type II:
  - at most 2 variables can be positive, and if two variables are positive they must be adjacent in the set, e.g. $\{x_1, x_2, x_3, x_4, x_5\}$ and $\{x_6, x_7, x_8\}$
  - piecewise linear optimization, scheduling

Solution strategy

- We will tackle these problems through *branch-and-cut*
- Our cutting plane strategy will follow the pioneering work of Crowder, Johnson, and Padberg (1985)
- Another ingredient is *lifting*

Branch-and-bound

It is an enumeration approach that uses:

- a divide-and-conquer strategy
- *linear programming relaxations* (LPR) to reduce the required amount of enumeration (or the number of *branch-and-bound nodes*) to prove optimality or infeasibility

Branch-and-cut

- Makes the use of LPR more efficient by sharpening them with *cutting planes*
- In many cases, improves over the number of branch-and-bound enumeration nodes and branch-and-bound computational time tremendously
- However, it needs to be used judiciously

Crowder, Johnson, and Padberg cutting plane strategy

- Introduced for 0-1 programming
- Consists of deriving cutting planes valid for *knapsack relaxations* of the problem, given by the individual constraints of $Ax \leq b$

Lifting

- Deriving cutting planes is a difficult task
- One alternative to deriving *strong cutting planes* is *lifting*, which consists in simplifying the problem first, obtaining a cutting plane for the simple problem, and then *incrementally* returning to the original problem

Multiple-choice

- If x and y are variables, *complementarity* means that at most one can be positive in a feasible solution, i.e. $x\,y = 0$
- If $x_1, \ldots, x_n$ are variables and satisfy complementarity pairwisely, we call the set $\{x_1, \ldots, x_n\}$ a *special ordered set of type* 1 (SOS1)

Multiple-choice

Traditionally it is modeled by introducing to each variable x_j a binary variable y_j and constraints:

- $x_j \leq u_j y_j$
- $\sum_{j \in S} y_j \leq 1$

Applications

Numerous, see e.g. Williams (1999)

- Approximation of nonlinear functions, e.g. bilinear (de Farias, Kozyreff, and Zhao, forthcoming)
- Auctions (Sandholm 2007)
- Capacity planning (Wolsey, 1990)
- Finance (Perold 1984)
- Lienar complementarity problem (Cottle et al., 1990)

Linear complementarity problem

They arise as:

$$A\,x + s = b$$

$$x, s \geq 0$$

$$\{x_i, s_i\} \text{ is SOS1 } \forall i$$

Problem definition

maximize $c\,x$

subject to

$$A\,x \leq b$$

$$x \geq 0,$$

$$\{x_{1j}, \ldots, x_{Tj}\} \text{ is SOS1 } \forall j$$

Previous and related work

- Jeroslow (1972)
- Ibaraki (1975)
- Mitchell et al. (2011)
- Richard and Tawarmalani (2011)
- de Farias et al. (2002)

Assumptions

- $T \geq 2$
- $A \geq 0,\ b > 0$
- $x_{ij} \in [0, 1]\ \forall ij$

Polyhedral strategy

We use as cutting planes inequalities valid for the *complementarity knapsack set P*, i.e. the convex hull of the set defined by

- $\Sigma_{ij} a_{ij} x_{ij} \leq b$
- $x_{ij} \in [0, 1] \; \forall ij$
- $\{x_{1j}, \ldots, x_{Tj}\}$ is SOS1 $\forall j$

So, from now on we focus on the set *P*.

Assumptions

- $b \geq a_{1j} > \cdots > a_{Tj} \geq 0$
- $\Sigma_j \, a_{1j} \geq b$

Basic results

Proposition The inequality $\Sigma_i \, x_{ij} \leq 1$ is valid $\forall j$. It is facet-defining iff $a_{Tj} < b$.

Basic results

Proposition The set P is full-dimensional.

Proposition Inequality $x_{ij} \geq 0$ is facet-defining $\forall ij$.

Proposition Inequality $\Sigma_{ij}\, a_{ij}\, x_{ij} \leq b$ is facet-defining iff $\Sigma_{j-k}\, a_{1j} + a_{Tk} \geq b\ \forall k$.

Cutting planes for P

- Two families of *lifted cover inequalities*
- They include the inequalities of de Farias et al. (2002) as special cases

Lifted cover inequalities

Consider the knapsack constraint:

$(6x_{11} + x_{21}) + (2x_{12} + x_{22}) + (4x_{13} + 3x_{23}) + (8x_{14} + 6x_{24} + x_{34}) + (9x_{15} + 4x_{25}) \leq 13$

The set $C = \{(1, 4), (1, 5)\}$ is a *cover* and

$8x_{14} + 9x_{15} \leq 13$

is a *cover inequality*, which can be lifted to $(8x_{14} + 6x_{24}) + 9x_{15} \leq 13$. Because of SOS1, it can

be lifted further to $(8x_{14} + 6x_{24} + 4x_{34}) + (9x_{15} + 5x_{25}) \leq 13$

Lifted cover inequalities

THEOREM 1. *Let C be a cover, and suppose that $j = 1\ \forall\, ij \in C$. Assume that C satisfies* (12). *Then*

$$\sum_{i \in M_C} a_{i1} x_{i1} + \sum_{i \in M_C} \sum_{j \in N_i - 1} \max\left\{a_{ij},\, b - \sum_{k \in M_C - i} a_{k1}\right\} x_{ij} \le b \tag{25}$$

is valid and facet-defining.

Lifted cover inequalities

THEOREM 2. *Let C be a cover that satisfies (12) with $j = n_i \; \forall\, i \in M_C - i'$, $j' < n_{i'}$, and $j'' = n_{i'}$, i.e. $a_{i'j'} + \sum_{i \in M_C - i'} a_{in_i} > b$ and $\sum_{i \in M_C} a_{in_i} < b$. Then,*

$$\sum_{j \in N_{i'}} \max\left\{a_{i'j}, b - \sum_{k \in M_C - i'} a_{kn_k}\right\} x_{i'j} + \sum_{i \in M_C - i'} a_{in_i} x_{in_i} + \sum_{i \in M_C - i'} \sum_{j \in N_i - n_i} a_{in_i} \max\left\{1, \frac{a_{ij}}{b - \sum_{k \in M_C - i} a_{kn_k}}\right\} x_{ij} \le b \tag{26}$$

is valid and facet defining.

Lifted cover inequalities

The knapsack again is:

$(6x_{11} + x_{21}) + (2x_{12} + x_{22}) + (4x_{13} + 3x_{23}) + (8x_{14} +$

$6x_{24} + x_{34}) + (9x_{15} + 4x_{25}) \leq 13$

The cover is $C = \{(2, 2), (2, 3), (3, 4), (1, 5)\}$ and

the cover inequality is $x_{22} + 3x_{23} + x_{34} + 9x_{15} \leq 13$,

which can be lifted to:

$(x_{12} + x_{22}) + (3x_{13} + 3x_{23}) + (8/5x_{14} + 6/5x_{24} +$

Computation

We tested randomly generated instances:

- very sparse: 2% of SOS1's
- relatively small: 60 rows and 100 SOS1's to 200 rows and 260 SOS1's (5 instances of each size), each SOS1 has 5 elements
- additionally, instances with very large SOS1's
- In some of the instances, the variables x_{ij} were continuous, in some they were binary, and in some they were general integers

Computation

- We limited CPU time to 1 hour
- We used the Texas Tech High Performance Computing Center running GUROBI 4 Callable Library

Summary of computational results

- The cutting planes were extremely useful for the instances with continuous and general integer variables. Here, the average computational time reduction was 82% and the average number of nodes reduction was 98%
- The results for the instances with binary variables were mixed, mostly disadvantageous. Here, the average computational time reduction was −32% and the average number of nodes reduction was 13%

Some computational results

# rows & # SOS	Time default	Time B&B
60 &100	1,853	3,600
60 & 120	2,520	3,600
60 & 140	525	2,830
60 & 160	1,720	3,600
80 & 100	2,329	3,600
80 & 120	789	3,600
80 & 140	2,890	3,600
80 & 160	1,250	2,820
# rows & # SOS	**Time default**	**Time B&B**

Some computational results

# rows & # SOS	Time default	B&B w/ cuts
60 &100	1,853	850
60 & 120	2,520	376
60 & 140	525	200
60 & 160	1,720	1,000
80 & 100	2,329	1,550
80 & 120	789	320
80 & 140	2,890	880
80 & 160	1,250	430

# rows & # SOS	Time default	B&B w/ cuts

Some computational results

# rows & # SOS	Time default	Default w/ cuts
60 &100	1,853	38
60 & 120	2,520	32
60 & 140	525	2
60 & 160	1,720	59
80 & 100	2,329	110
80 & 120	789	58
80 & 140	2,890	86
80 & 160	1,250	220
# rows & # SOS	**Time default**	**Default w/ cuts**

Modeling alternatives

- SOS1 approach (Beale and Tomlin 1970)
- "Usual" MIP (Dantzig 1961; we will call it MIP)
- LOG (Vielma and Nemhauser 2009 and Vielma Ahmed and Nemhauser 2010)

Computational results

- The cuts were helpful in all three formulations
- However, by far, MIP was the best, consistently, in all instances, even the ones with very large SOS1's
- Why?

Further research

- Bilinear programming
- Multiple-choice over multiple rows
- New inequalities for multiple-choice

Piecewise linear optimization

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

Problem definition

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

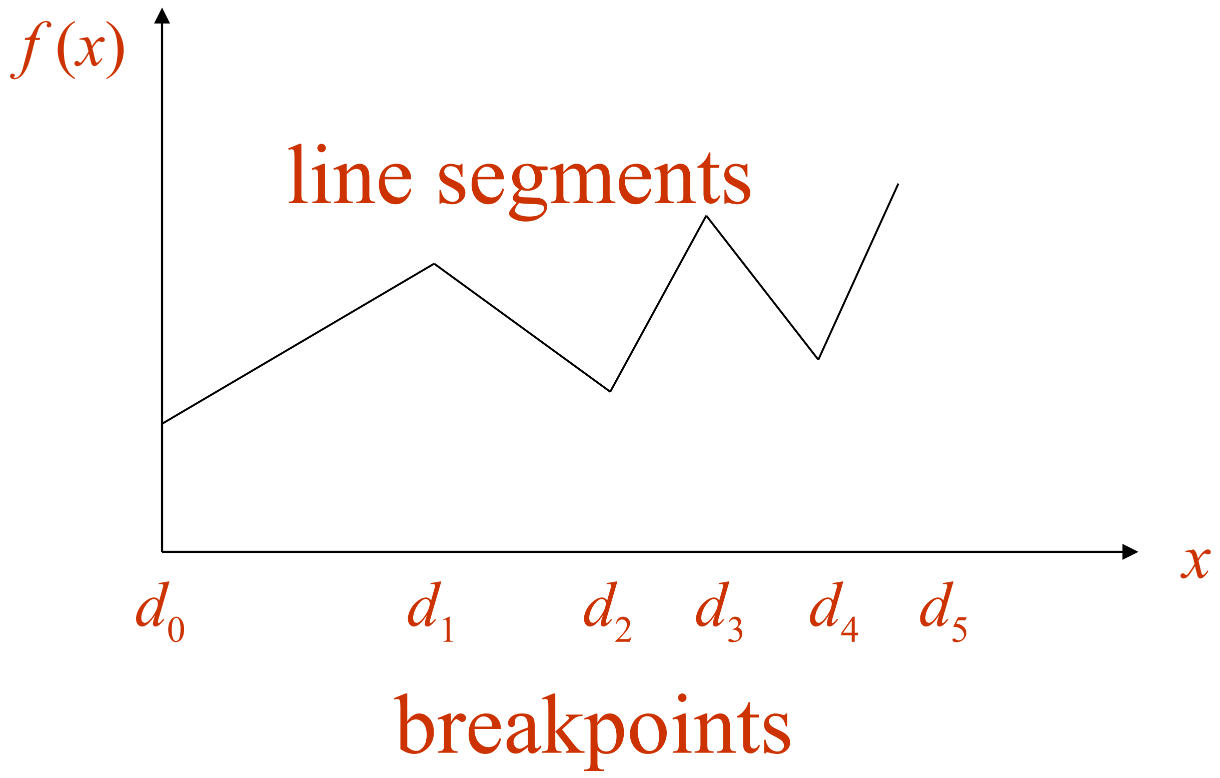
subject to

$$A\,x \leq b$$

$$x \geq 0,$$

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

Traditionally it is modeled by introducing to each line segment a binary variable y_j and constraints:

- $\lambda_1 \leq u_1 y_1$
- $\lambda_2 \leq u_2 (y_1 + y_2)$
- $\lambda_3 \leq u_3 (y_2 + y_3)$
- $\lambda_4 \leq u_4 (y_3 + y_4)$
- $\lambda_5 \leq u_5 y_4$
- $y_1 + \ldots + y_4 \leq 1$

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

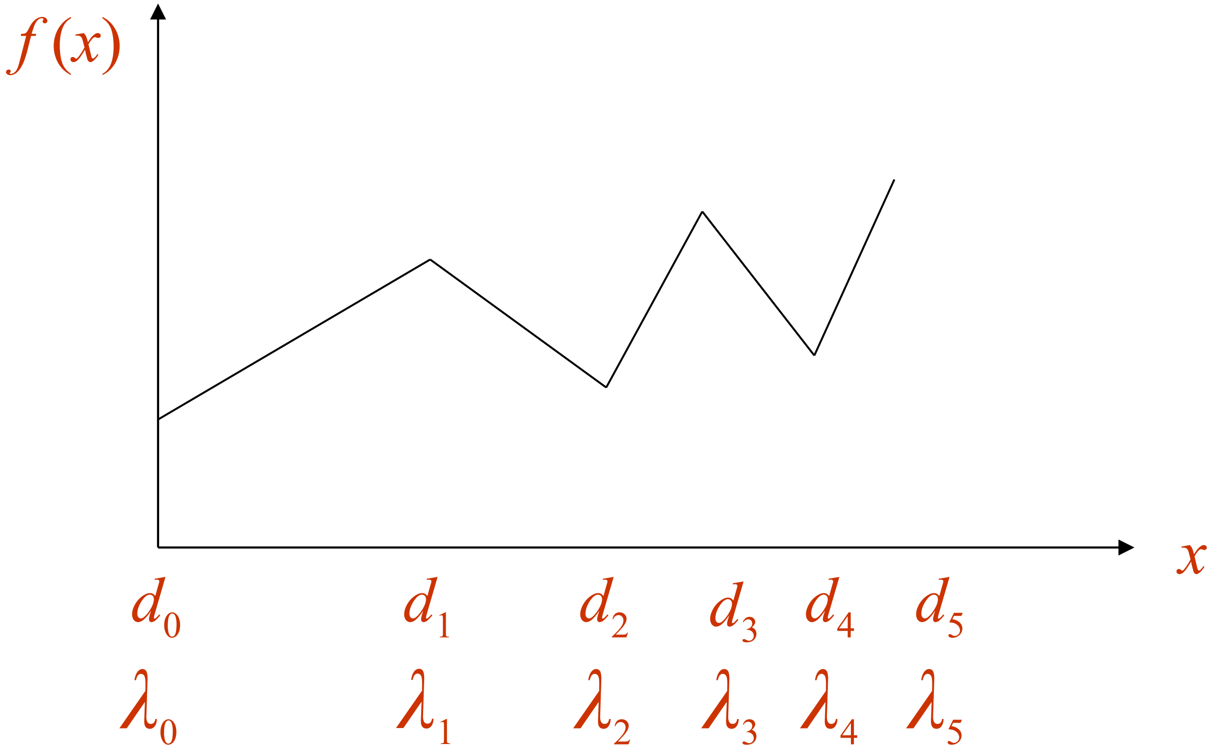
Let $x \in \{0\} \cup [1, 2]$.

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

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

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

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

$$\lambda \geq 0$$

$\{\lambda_1, \ldots, \lambda_T\}$ 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

- $S = \{ \lambda \in \Re \quad : \quad \lambda$ 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
# Nodes & part.	Time default	Time B&B

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
# Nodes & part.	**Time default**	**Time w/ cuts**

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
#Nodes & part.	Time default	Time B&B

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
#Nodes & part.	Time default	Time w/ cuts

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:

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

is valid for *P*, where:

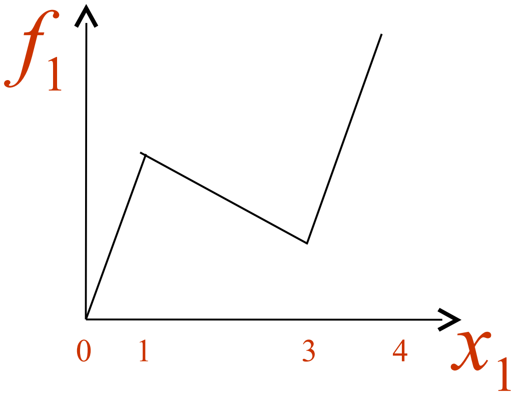
$$\left(\alpha_i^{k_i - 1}, \alpha_i^{k_i}\right) \in \left\{ (0,0), \left(\frac{a_j^s + a_i^{k_i - 1} - b'}{a_j^s}, \frac{a_j^s + a_i^{k_i} - b'}{a_j^s} \right) \right\} \ \forall i \in I \ \textit{with} \ k_i > 1 \ \textit{and} \ a_j^s + a_i^{k_i - 1} < b',$$

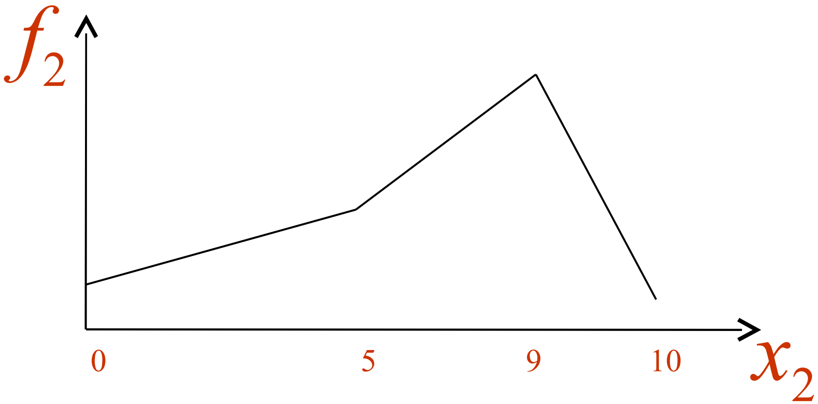
$$\left(\alpha_i^{k_i - 1}, a_i^{k_i}\right) = \left(0, \frac{a_j^s + a_i^{k_i} - b'}{a_j^s}\right) \ \forall i \in I \ \textit{with} \ k_i > 1 \ \textit{and} \ a_j^s + a_i^{k_i - 1} = b',$$

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

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

Example





$$2x_1 + x_2 \leq 10$$

$$(0 \cdot \lambda_1 + 2\lambda_1 + 6\lambda_1 + 8\lambda_1) + (0 \cdot \lambda_2 + 5\lambda_2 + 9\lambda_2 + 10\lambda_2) \leq 10$$

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:

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

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

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

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

Cover

Definition ($N = N$):

Let $2 \leq l_j \leq T \; \forall j \in N$ 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 $C^+ \subseteq N^+$, $C^- \subseteq N^-$, $2 \leq l_j \leq u \; \forall j \in C^+$, $1 \leq l_j \leq u \; \forall j \in C^-$, 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 ($N = N$):

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

and

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

Also in general, we have

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

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

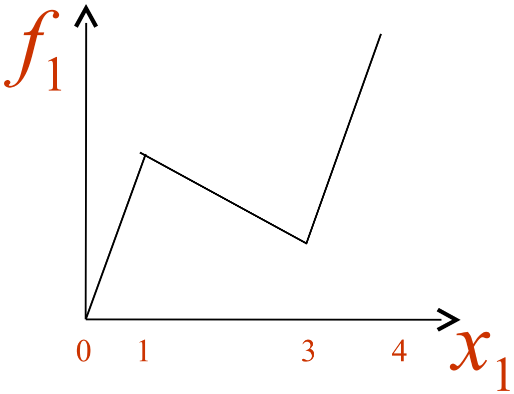
We hope that LOG breaks the symmetry of the network formulations. However, it could very well be that LOG's advantage is due to branching implementation.

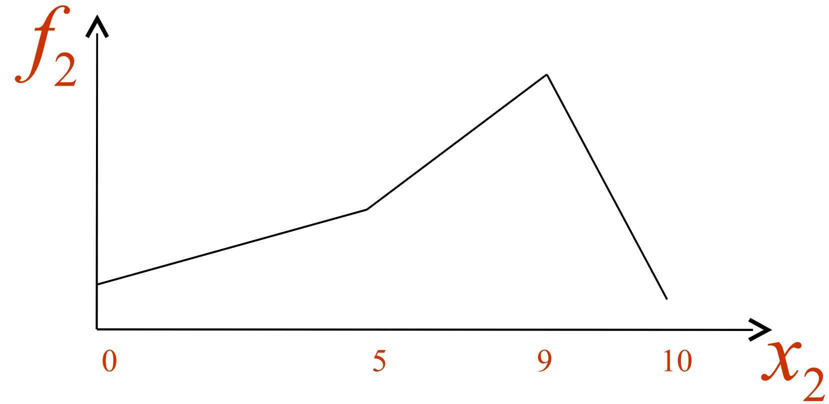
Formulation

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





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

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

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
Inst. size	**Default**	**Time SC cut**	**Time PL cut**

Further research

Piecewise linear optimization for:

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