Polyhedral Aspects of Nonconvex, Lower Semicontinuous Piecewise Linear Optimization

Juan Pablo Vielma[1] Ahmet B. Keha[2]
George L. Nemhauser[1]

[1]School of Industrial and Systems Engineering
Georgia Institute of Technology

[2]Department of Industrial Engineering
Arizona State University

INFORMS Annual Meeting, 2006 – Pittsburgh

Outline

The Problem

$$\min \sum_{j \in N} f_j(x_j)$$

$$s.t.$$

$$\sum_{j \in N} g_{ij} x_j \leq b_i \quad \forall i \in \{1, \ldots, m\}$$

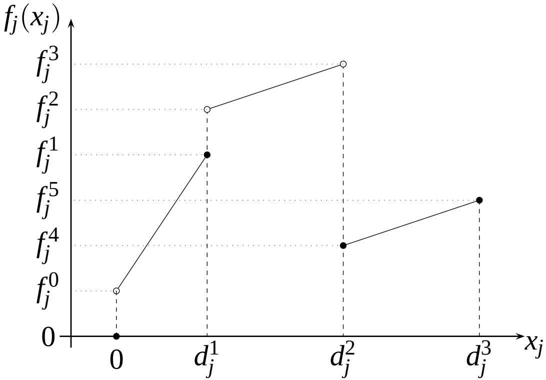
$$0 \leq x_j \leq u_j \quad \forall j \in N := \{1, \ldots, n\}$$

- $f_j(x_j) : [0, u_j] \to \mathbb{R}$ is lower semicontinuous, nonconvex and piecewise linear.
- Simplifying assumption $m = 1$.

History and Objectives

- History:
  - Problem is NP-hard and has many applications. Keha et. al (2004):
    - Network flow problems with non-convex objectives and fixed charges.
  - Branch-and-Cut algorithm without binary variables for the continuous case. Keha et. al (2004).
  - Extension of model to the non-lower semicontinuous case and new cuts. de Farias et. al (2005).
- Objective:
  - Extend cuts from Keha et. al (2004) to the lower semicontinuous case.
  - New cuts for fixed charge case.

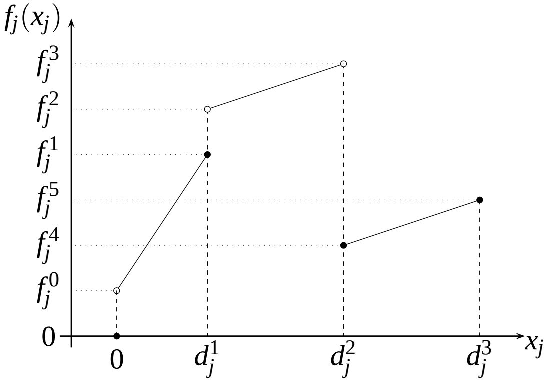
$f_j(x_j)$ is linear in (d_j^k, d_j^{k+1}) and lower semicontinuous



- $e_j^0 = 0$, $e_j^1 = d_j^1$, $e_j^2 = d_j^1$, $e_j^3 = d_j^2$, $e_j^4 = d_j^2$, $e_j^5 = d_j^3$

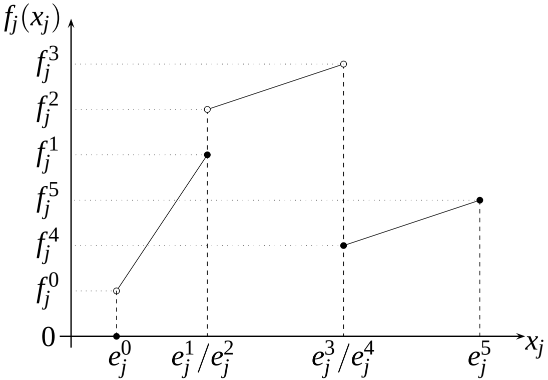
- $x_j = \sum_{k=0}^{5} e_j^k \lambda_j^k$ and $f_j(x_j) = \sum_{k=0}^{5} f_j^k \lambda_j^k$
- $\sum_{k=0}^{5} \lambda_j^k = 1$, $\lambda_0^k \geq 0$ and $(\lambda_j^k)_{k=0}^5$ is *SOS2*.
- Fixed Charge: $\underline{\lambda}_j^0 + \overline{\lambda}_j^0 = \lambda_j^0$, $\underline{\lambda}_j^0 \in \{0,1\}$, $\overline{\lambda}_j^0 \geq 0$ and $f_j(x_j) = f_j^0 \overline{\lambda}_j^0 + \sum_{k=1}^{5} f_j^k \lambda_j^k$.

$f_j(x_j)$ is linear in (d_j^k, d_j^{k+1}) and lower semicontinuous



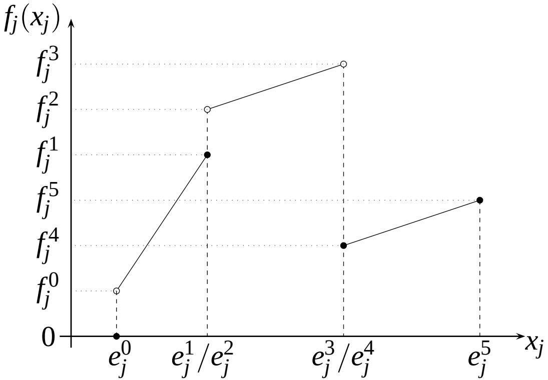
- $e_j^0 = 0$, $e_j^1 = d_j^1$, $e_j^2 = d_j^1$, $e_j^3 = d_j^2$, $e_j^4 = d_j^2$, $e_j^5 = d_j^3$
- $x_j = \sum_{k=0}^5 e_j^k \lambda_j^k$ and $f_j(x_j) = \sum_{k=0}^5 f_j^k \lambda_j^k$
- $\sum_{k=0}^5 \lambda_j^k = 1$, $\lambda_0^k \geq 0$ and $(\lambda_j^k)_{k=0}^5$ is *SOS2*.
- Fixed Charge: $\underline{\lambda}_j^0 + \overline{\lambda}_j^0 = \lambda_j^0$, $\underline{\lambda}_j^0 \in \{0,1\}$, $\overline{\lambda}_j^0 \geq 0$ and $f_j(x_j) = f_j^0 \overline{\lambda}_j^0 + \sum_{k=1}^5 f_j^k \lambda_j^k$.

$f_j(x_j)$ is linear in (d_j^k, d_j^{k+1}) and lower semicontinuous



- $e_j^0 = 0$, $e_j^1 = d_j^1$, $e_j^2 = d_j^1$, $e_j^3 = d_j^2$, $e_j^4 = d_j^2$, $e_j^5 = d_j^3$
- $x_j = \sum_{k=0}^5 e_j^k \lambda_j^k$ and $f_j(x_j) = \sum_{k=0}^5 f_j^k \lambda_j^k$
- $\sum_{k=0}^5 \lambda_j^k = 1$, $\lambda_0^k \geq 0$ and $(\lambda_j^k)_{k=0}^5$ is *SOS2*.
- Fixed Charge: $\underline{\lambda}_j^0 + \overline{\lambda}_j^0 = \lambda_j^0$, $\underline{\lambda}_j^0 \in \{0,1\}$, $\overline{\lambda}_j^0 \geq 0$ and $f_j(x_j) = f_j^0 \overline{\lambda}_j^0 + \sum_{k=1}^5 f_j^k \lambda_j^k$.

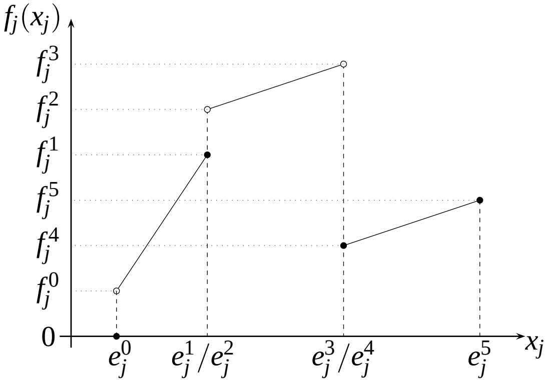
$f_j(x_j)$ is linear in (d_j^k, d_j^{k+1}) and lower semicontinuous



- $e_j^0 = 0$, $e_j^1 = d_j^1$, $e_j^2 = d_j^1$, $e_j^3 = d_j^2$, $e_j^4 = d_j^2$, $e_j^5 = d_j^3$

- $x_j = \sum_{k=0}^{5} e_j^k \lambda_j^k$ and $f_j(x_j) = \sum_{k=0}^{5} f_j^k \lambda_j^k$
- $\sum_{k=0}^{5} \lambda_j^k = 1$, $\lambda_0^k \geq 0$ and $(\lambda_j^k)_{k=0}^5$ is *SOS2*.
- Fixed Charge: $\underline{\lambda}_j^0 + \overline{\lambda}_j^0 = \lambda_j^0$, $\underline{\lambda}_j^0 \in \{0,1\}$, $\overline{\lambda}_j^0 \geq 0$ and $f_j(x_j) = f_j^0 \overline{\lambda}_j^0 + \sum_{k=1}^{5} f_j^k \lambda_j^k$.

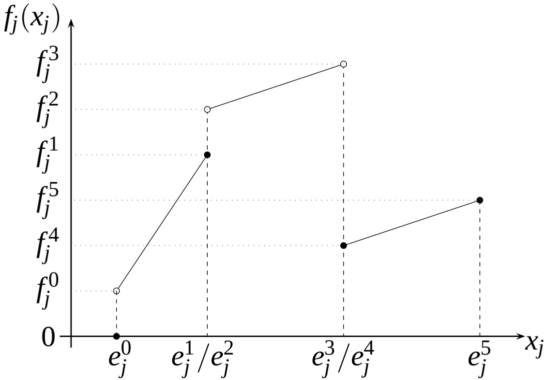
$f_j(x_j)$ is linear in (d_j^k, d_j^{k+1}) and lower semicontinuous



- $e_j^0 = 0$, $e_j^1 = d_j^1$, $e_j^2 = d_j^1$, $e_j^3 = d_j^2$, $e_j^4 = d_j^2$, $e_j^5 = d_j^3$

- $x_j = \sum_{k=0}^{5} e_j^k \lambda_j^k$ and $f_j(x_j) = \sum_{k=0}^{5} f_j^k \lambda_j^k$
- $\sum_{k=0}^{5} \lambda_j^k = 1$, $\lambda_0^k \geq 0$ and $(\lambda_j^k)_{k=0}^{5}$ is *SOS2*.
- Fixed Charge: $\underline{\lambda}_j^0 + \overline{\lambda}_j^0 = \lambda_j^0$, $\underline{\lambda}_j^0 \in \{0, 1\}$, $\overline{\lambda}_j^0 \geq 0$ and $f_j(x_j) = f_j^0 \overline{\lambda}_j^0 + \sum_{k=1}^{5} f_j^k \lambda_j^k$.

$f_j(x_j)$ is linear in (d_j^k, d_j^{k+1}) and lower semicontinuous



- $e_j^0 = 0$, $e_j^1 = d_j^1$, $e_j^2 = d_j^1$, $e_j^3 = d_j^2$, $e_j^4 = d_j^2$, $e_j^5 = d_j^3$

- $x_j = \sum_{k=0}^{5} e_j^k \lambda_j^k$ and $f_j(x_j) = \sum_{k=0}^{5} f_j^k \lambda_j^k$
- $\sum_{k=0}^{5} \lambda_j^k = 1$, $\lambda_0^k \geq 0$ and $(\lambda_j^k)_{k=0}^5$ is *SOS2*.
- Fixed Charge: $\underline{\lambda}_j^0 + \overline{\lambda}_j^0 = \lambda_j^0$, $\underline{\lambda}_j^0 \in \{0,1\}$, $\overline{\lambda}_j^0 \geq 0$ and $f_j(x_j) = f_j^0 \overline{\lambda}_j^0 + \sum_{k=1}^{5} f_j^k \lambda_j^k$.

The *SOS2* Based Model

$$\min \quad \sum_{j\in N} f_j^0 \overline{\lambda}_j^0 + \sum_{k=1}^{T} f_j^k \lambda_j^k$$

$s.t.$

$$\sum_{j\in N}\sum_{k=1}^{T} a_j^k \lambda_j^k \le b$$

$$\sum_{k=0}^{T} \lambda_j^k = 1,\ \lambda_j^k \ge 0,\ (\lambda_j^k)_{k=0}^{T} \text{ is } SOS2$$

$$\underline{\lambda}_j^0 + \overline{\lambda}_j^0 = \lambda_j^0,\ \underline{\lambda}_j^0 \in \{0,1\},\ \overline{\lambda}_j^0 \ge 0$$

- a_j defined appropriately from original linear constraint.
- Fixing $\underline{\lambda}_j^0 = 0$ removes fixed charge.
- Obs: $\underline{\lambda}_j^0 \in \{0,1\}$ is not artificial.

Because model is still *SOS2* cuts from Keha et. al (2004) are directly valid.

- Small care with $a_j^k \leq a_j^{k+1}$ instead of $a_j^k < a_j^{k+1}$.
- Lifted Convexity Constraints:
  - Obtained by lifting $\sum_{k=0}^{T} \lambda_j^k = 1$.
  - For $i \neq j \in N$:

$$\sum_{k=1}^{T} \lambda_j^k + \sum_{k=k_i-1}^{T} \alpha_i^k \lambda_i^k \leq 1$$

- Lifted Cover Constraints:
  - Extend the concept of a cover to *SOS2* continuous variables.
  - For $C \subseteq N$:

$$\sum_{j \in C} (\alpha_j \lambda_j^{k_j - 1} + \sum_{k=k_j}^{T} \lambda_j^k) \leq |C| - 1$$

Because model is still *SOS2* cuts from Keha et. al (2004) are directly valid.

- Small care with $a_j^k \leq a_j^{k+1}$ instead of $a_j^k < a_j^{k+1}$.
- Lifted Convexity Constraints:
  - Obtained by lifting $\sum_{k=0}^{T} \lambda_j^k = 1$.
  - For $i \neq j \in N$:

$$\sum_{k=1}^{T} \lambda_j^k + \sum_{k=k_i-1}^{T} \alpha_i^k \lambda_i^k \leq 1$$

- Lifted Cover Constraints:
  - Extend the concept of a cover to *SOS2* continuous variables.
  - For $C \subseteq N$:

$$\sum_{j \in C} (\alpha_j \lambda_j^{k_j-1} + \sum_{k=k_j}^{T} \lambda_j^k) \leq |C| - 1$$

Because model is still *SOS2* cuts from Keha et. al (2004) are directly valid.

- Small care with $a_j^k \leq a_j^{k+1}$ instead of $a_j^k < a_j^{k+1}$.
- Lifted Convexity Constraints:
  - Obtained by lifting $\sum_{k=0}^{T} \lambda_j^k = 1$.
  - For $i \neq j \in N$:

$$\sum_{k=1}^{T} \lambda_j^k + \sum_{k=k_i-1}^{T} \alpha_i^k \lambda_i^k \leq 1$$

- Lifted Cover Constraints:
  - Extend the concept of a cover to *SOS2* continuous variables.
  - For $C \subseteq N$:

$$\sum_{j \in C} (\alpha_j \lambda_j^{k_j-1} + \sum_{k=k_j}^{T} \lambda_j^k) \leq |C| - 1$$

Lifted Convexity Constraints can be Improved

- Aggregated Lifted Convexity Constraints:
  - Obtained by adding groups of convexity constraints and then lifting.
  - For $l \in N$ and $I \subseteq N \setminus \{l\}$:

$$\sum_{i \in I} \sum_{k=1}^{T} \lambda_i^k + \alpha_l^{k_l} \lambda_l^{k_l} \leq |I|$$

SOS2 Model can be Relaxed to Variable Upper Bound Model

$$\sum_{j \in N} \sum_{k=1}^{T} a_j^k \lambda_j^k \leq b$$

$$\sum_{k=0}^{T} \lambda_j^k = 1,\ \lambda_j^k \geq 0,\ (\lambda_j^k)_{k=0}^T \text{ is } SOS2$$

$$\underline{\lambda}_j^0 + \overline{\lambda}_j^0 = \lambda_j^0,\ \underline{\lambda}_j^0 \in \{0,1\},\ \overline{\lambda}_j^0 \geq 0$$

-
- $z_j = 1 - \underline{\lambda}_j^0$.

SOS2 Model can be Relaxed to Variable Upper Bound Model

$$\sum_{j\in N}\sum_{k=1}^{T} a_j^k \lambda_j^k \leq b$$

$$\sum_{k=0}^{T} \lambda_j^k = 1,\ \lambda_j^k \geq 0,\ (\lambda_j^k)_{k=0}^{T} \text{ is } SOS2$$

$$\underline{\lambda}_j^0 + \overline{\lambda}_j^0 = \lambda_j^0,\ \underline{\lambda}_j^0 \in \{0,1\},\ \overline{\lambda}_j^0 \geq 0$$

- $y_j = \sum_{k=1}^{T} a_j^k \lambda_j^k$
- $z_j = 1 - \underline{\lambda}_j^0$.

SOS2 Model can be Relaxed to Variable Upper Bound Model

$$\sum_{j \in N} y_j \leq b$$
$$y_j \leq a_j^T z_j$$
$$z_j \in \{0, 1\}$$

- $y_j = \sum_{k=1}^{T} a_j^k \lambda_j^k$
- $z_j = 1 - \underline{\lambda}_j^0$.
- Variable Upper Bound Relaxation.

SOS2 Model can be Relaxed to Variable Upper Bound Model

$$\sum_{j \in N} y_j \le b$$

$$y_j \le a_j^T z_j$$

$$z_j \in \{0, 1\}$$

- $y_j = \sum_{k=1}^{T} a_j^k \lambda_j^k$
- $z_j = 1 - \underline{\lambda}_j^0$.
- Variable Upper Bound Relaxation.

SOS2 Model can be Relaxed to Variable Upper Bound Model

$$\sum_{j \in N} y_j \leq b$$

$$y_j \leq a_j^{k_j} z_j$$

$$z_j \in \{0, 1\}$$

- $y_j = \sum_{k=1}^{k_j} a_j^k \lambda_j^k$, $k_j \in \{1, \ldots, T\}$
- $z_j = 1 - \underline{\lambda}_j^0$.
- Variable Upper Bound Relaxation.

Flow Cover Cuts can be Obtained from Variable Upper Bound Relaxation

- For $C \subseteq N$ and $k_j \geq 1$, $j \in C$ such that $\sum_{j \in C} a_j^{k_j} = b + \Delta$ with $\Delta > 0$ we get the *Fixed Charge Flow Cover Cut*:

$$\sum_{j \in C} \sum_{k=1}^{k_j - 1} a_j^k \lambda_j^k + \sum_{j \in C} a_j^{k_j} \sum_{k=k_j}^{T} \lambda_j^k + \sum_{j \in C} (a_j^{k_j} - \Delta)^+ \underline{\lambda}_j^0 \leq b$$

- Stronger cuts can be used (i.e. Lifted flow cover cuts).

Flow Cover Cuts can be Obtained from Variable Upper Bound Relaxation

- For $C \subseteq N$ and $k_j \geq 1$, $j \in C$ such that $\sum_{j \in C} a_j^{k_j} = b + \Delta$ with $\Delta > 0$ we get the *Fixed Charge Flow Cover Cut*:

$$\sum_{j \in C} \sum_{k=1}^{k_j - 1} a_j^k \lambda_j^k + \sum_{j \in C} a_j^{k_j} \sum_{k=k_j}^{T} \lambda_j^k + \sum_{j \in C} (a_j^{k_j} - \Delta)^+ \underline{\lambda}_j^0 \leq b$$

- Stronger cuts can be used (i.e. Lifted flow cover cuts).

SOS2 Model can be Relaxed to Binary Knapsack Model

$$\sum_{j\in N}\sum_{k=1}^{T} a_j^k \lambda_j^k \geq b$$

$$\sum_{k=0}^{T} \lambda_j^k = 1,\ \lambda_j^k \geq 0,\ (\lambda_j^k)_{k=0}^{T} \text{ is } SOS2$$

$$\underline{\lambda}_j^0 + \overline{\lambda}_j^0 = \lambda_j^0,\ \underline{\lambda}_j^0 \in \{0,1\},\ \overline{\lambda}_j^0 \geq 0$$

- Use binary variables $\underline{\lambda}_j^0$.
- Model can be improved by fixing $\lambda_i^k = 0,\ k \geq k_i + 1$.
- Cover cuts for this model need to be lifted.

SOS2 Model can be Relaxed to Binary Knapsack Model

$$\sum_{j \in N} a_j^T \underline{\lambda}_j^0 \leq \sum_{j \in N} a_j^T - b$$

$$\underline{\lambda}_j^0 \in \{0,1\} \qquad \forall j \in N$$

- Use binary variables $\underline{\lambda}_j^0$.
- Model can be improved by fixing $\lambda_i^k = 0,\ k \geq k_i + 1$.
- Cover cuts for this model need to be lifted.

SOS2 Model can be Relaxed to Binary Knapsack Model

$$\sum_{j \in N} a_j^T \underline{\lambda}_j^0 \leq \sum_{j \in N} a_j^T - b$$

$$\underline{\lambda}_j^0 \in \{0, 1\} \qquad \forall j \in N$$

- Use binary variables $\underline{\lambda}_j^0$.
- Model can be improved by fixing $\lambda_i^k = 0,\ k \geq k_i + 1$.
- Cover cuts for this model need to be lifted.

SOS2 Model can be Relaxed to Binary Knapsack Model

$$\sum_{j\in N\setminus C} a_j^{k_j}\underline{\lambda}_j^0 + \sum_{j\in C} a_j^T\underline{\lambda}_j^0 \le \sum_{j\in N\setminus C} a_j^{k_j} + \sum_{j\in C} a_j^T - b$$

$$\underline{\lambda}_j^0 \in \{0,1\} \qquad \forall j \in N$$

- Use binary variables $\underline{\lambda}_j^0$.
- Model can be improved by fixing $\lambda_i^k = 0,\ k \ge k_i + 1$.
- Cover cuts for this model need to be lifted.

Lifting Back Continuous Variables λ_i^k Yields Valid Cut

- For $C \subseteq N$ and $k_j \geq 1$, $j \in N \setminus C$ such that

$$\rho = b - \sum_{i \in N \setminus C} a_i^{k_i} > 0 \quad (1)$$

$$\sum_{i \in N \setminus C} a_i^{k_i} + a_j^T \geq b \quad \forall j \in C \quad (2)$$

$$\sum_{i \in N \setminus (C \cup \{j\})} a_i^{k_i} + a_j^{k_j+1} \geq b \quad \forall j \in N \setminus C \quad (3)$$

  we get *Fixed Charge Cover Cut*:

$$\sum_{j \in C} \underline{\lambda}_j^0 + \sum_{i \in N \setminus C} \left[\left(\frac{a_i^{k_i} - a_i^{k_i+1}}{\rho} \right) \lambda_i^{k_i+1} - \sum_{k=k_i+2}^{T} \lambda_i^k \right] \leq |C| - 1$$

- Not clear how to lift stronger cuts (i.e. Lifted cover cover cuts).

Lifting Back Continuous Variables λ_i^k Yields Valid Cut

- For $C \subseteq N$ and $k_j \geq 1, j \in N \setminus C$ such that

$$\rho = b - \sum_{i \in N \setminus C} a_i^{k_i} > 0 \tag{1}$$

$$\sum_{i \in N \setminus C} a_i^{k_i} + a_j^T \geq b \quad \forall j \in C \tag{2}$$

$$\sum_{i \in N \setminus (C \cup \{j\})} a_i^{k_i} + a_j^{k_j+1} \geq b \quad \forall j \in N \setminus C \tag{3}$$

we get *Fixed Charge Cover Cut*:

$$\sum_{j \in C} \lambda_j^0 + \sum_{i \in N \setminus C} \left[\left(\frac{a_i^{k_i} - a_i^{k_i+1}}{\rho} \right) \lambda_i^{k_i+1} - \sum_{k=k_i+2}^{T} \lambda_i^k \right] \leq |C| - 1$$

- Not clear how to lift stronger cuts (i.e. Lifted cover cover cuts).

Branch–and-Cut without Binary Variables isn't Always Practical

- Branch–and-Cut without binary variables implemented in Minto is faster than binary variable version (Keha et. al (2004))
- "Good" implementation of SOS2 requirements using variable branching:
  - *Disaggregated convex combination model.*
  - Sherali (2001), Croxton et. al. (2003).
- CPLEX's binary variables implementation more advanced that SOS2 implementation:
  - Branching (Pseudocosts, strong branching, etc.)
  - Heuristics (RINS, etc.)
  - Preprocessing.
  - Cuts for binary variables.
- Using binary variables is currently best "practical" implementation of SOS2.

Branch–and-Cut without Binary Variables isn't Always Practical

- Branch–and-Cut without binary variables implemented in Minto is faster than binary variable version (Keha et. al (2004))
- "Good" implementation of SOS2 requirements using variable branching:
  - *Disaggregated convex combination model*.
  - Sherali (2001), Croxton et. al. (2003).
- CPLEX's binary variables implementation more advanced that SOS2 implementation:
  - Branching (Pseudocosts, strong branching, etc.)
  - Heuristics (RINS, etc.)
  - Preprocessing.
  - Cuts for binary variables.
- Using binary variables is currently best "practical" implementation of SOS2.

Branch–and-Cut without Binary Variables isn't Always Practical

- Branch–and-Cut without binary variables implemented in Minto is faster than binary variable version (Keha et. al (2004))
- "Good" implementation of SOS2 requirements using variable branching:
  - *Disaggregated convex combination model*.
  - Sherali (2001), Croxton et. al. (2003).
- CPLEX's binary variables implementation more advanced that SOS2 implementation:
  - Branching (Pseudocosts, strong branching, etc.)
  - Heuristics (RINS, etc.)
  - Preprocessing.
  - Cuts for binary variables.
- Using binary variables is currently best "practical" implementation of SOS2.

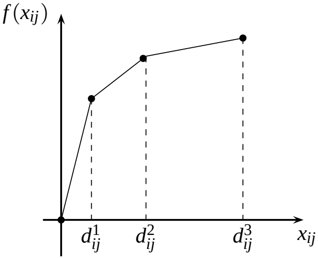
Branch–and-Cut without Binary Variables isn't Always Practical

- Branch–and-Cut without binary variables implemented in Minto is faster than binary variable version (Keha et. al (2004))
- "Good" implementation of SOS2 requirements using variable branching:
  - *Disaggregated convex combination model*.
  - Sherali (2001), Croxton et. al. (2003).
- CPLEX's binary variables implementation more advanced that SOS2 implementation:
  - Branching (Pseudocosts, strong branching, etc.)
  - Heuristics (RINS, etc.)
  - Preprocessing.
  - Cuts for binary variables.
- Using binary variables is currently best "practical" implementation of SOS2.

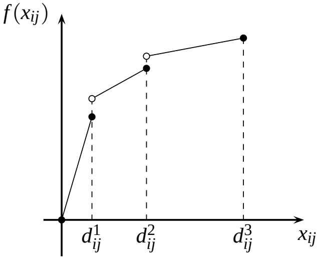
Test instances

- Transportation problems with various supply×demand nodes:
  - 10×10,12×18,15×15 and ,20×20.
  - 5 randomly generated instances for each size.
  - Minimization of 4 types of nonconvex separable piecewise linear function with 4 and 5 segments.
- Solved with CPLEX 9.0:
  - Using binary variables to implement SOS2.
  - Default CPLEX and Default CPLEX + SOS2 Cuts.

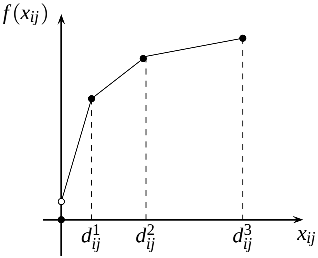
Types of Objective Functions



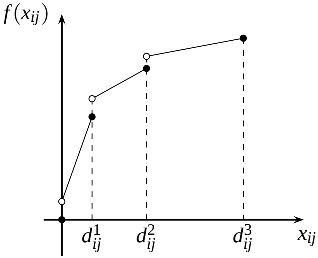
Continuous No Fixed Charge



Discontinuous No Fixed Charge



Continuous Fixed Charge



Discontinuous Fixed Charge

Total Decrease in # of Branch–and–Bound Nodes When Adding SOS2 Cuts

	Continuous	Discontinuous
Without Fixed Charge	91%	88%
With Fixed Charge	94%	94%

Total Decrease in Solve Time When Adding SOS2 Cuts

	Continuous	Discontinuous
Without Fixed Charge	21%	17 %
With Fixed Charge	18%	16%

Conclusions

- Cuts for continuous piecewise linear SOS2 models can be extended to the lower semicontinuous case.
- Lifted convexity constraints can be strengthened by aggregation.
- Cuts for fixed charge linear transportation problems can be extended to the piecewise linear case.
- Binary variables currently best way of implementing SOS2.
- Cuts for SOS2 improve performance of solves using CPLEX.