Extended Formulations, Lagrangian Relaxation, & Column Generation: tackling large scale applications

François Vanderbeck

University of Bordeaux

INRIA Bordeaux-Sud-Ouest

part 1: Defining Extended Formulations

Outline

1. Extented Formulations
   - Formulation
   - Extended Formulation
   - Reformulation
   - Decomposition & Reformulation
2. Examples
   - Steiner Tree Problem
   - Traveling Salesman Problem
   - Capacitated Network Design
3. How to build them
   - Variable Splitting
   - DP based reformulation
   - LP separation
   - Union of Polyhedra
   - Reduced coefficient & basis
4. Interests of Reformulations

Outline

1. Extented Formulations
   - Formulation
   - Extended Formulation
   - Reformulation
   - Decomposition & Reformulation
2. Examples
   - Steiner Tree Problem
   - Traveling Salesman Problem
   - Capacitated Network Design
3. How to build them
   - Variable Splitting
   - DP based reformulation
   - LP separation
   - Union of Polyhedra
   - Reduced coefficient & basis
4. Interests of Reformulations

Outline

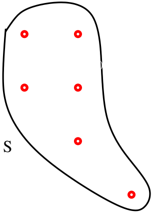
1. Extented Formulations
   - Formulation
   - Extended Formulation
   - Reformulation
   - Decomposition & Reformulation
2. Examples
   - Steiner Tree Problem
   - Traveling Salesman Problem
   - Capacitated Network Design
3. How to build them
   - Variable Splitting
   - DP based reformulation
   - LP separation
   - Union of Polyhedra
   - Reduced coefficient & basis
4. Interests of Reformulations

Formulation

Combinatorial Optimization Problem

$$(CO) \equiv \min\{c(s) : \; s \in \mathbf{S}\}$$

where $\mathbf{S}$ is the “discrete” set of feasible solutions.



Combinatorial Optimization Problem

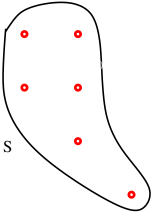
$$(CO) \equiv \min\{c(s) : \ s \in \mathbf{S}\}$$

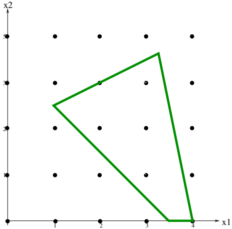
where $\mathbf{S}$ is the “discrete” set of feasible solutions.

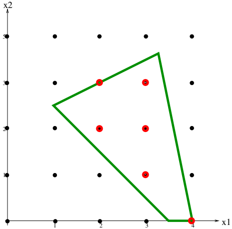
Formulation

A **polyhedron** $\mathbf{P} = \{x \in \mathbb{R}^n : Ax \geq a\}$ is a formulation for (CO) iff

$$\min\{c(s) : \ s \in \mathbf{S}\} \quad \equiv \quad \min\{cx : x \in \mathbf{P_I} = \mathbf{P} \cap \mathbb{N}^{\mathbf{n}}\}.$$



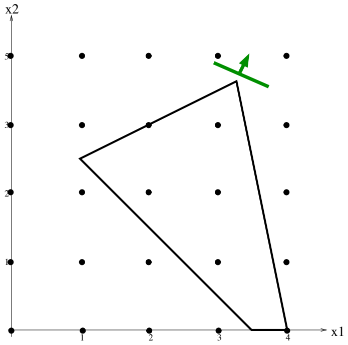




Integer Program

$$(IP) \qquad \min\{cx : x \in X\}$$

where $X = P \cap \mathbb{Z}^n$ with $P = \{x \in \mathbb{R}^n_+ : Ax \geq a\}$.



MIP - formulation & Generic Solver

Integer Program

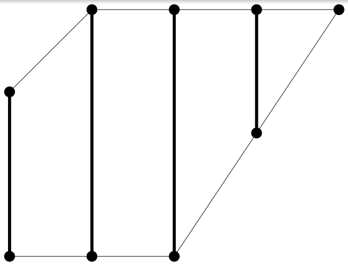
$$(IP) \qquad \min\{cx : x \in X\}$$

where $X = P \cap \mathbb{Z}^n$ with $P = \{x \in \mathbb{R}^n_+ : Ax \geq a\}$.

Mixed Integer Program

$$(MIP) \qquad \min\{cx + hy : (x,y) \in X\}$$

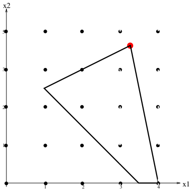
where $X = P \cap (\mathbb{Z}^n \times \mathbb{R}^p)$ with $P = \{(x,y) \in \mathbb{R}^n_+ \times \mathbb{R}^p_+ : Gx + Hy \geq b\}$.

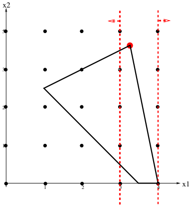


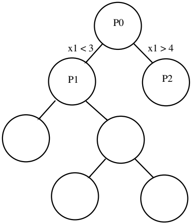
Integer Program

$$(IP) \qquad \min\{cx : x \in X\}$$

where $X = P \cap \mathbb{Z}^n$ with $P = \{x \in \mathbb{R}^n_+ : Ax \geq a\}$.





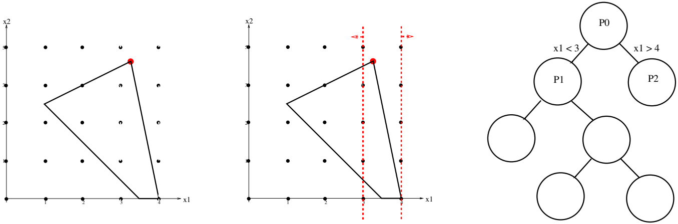


MIP - formulation & Generic Solver

Integer Program

$$(IP) \qquad \min\{cx : x \in X\}$$

where $X = P \cap \mathbb{Z}^n$ with $P = \{x \in \mathbb{R}^n_+ : Ax \geq a\}$.

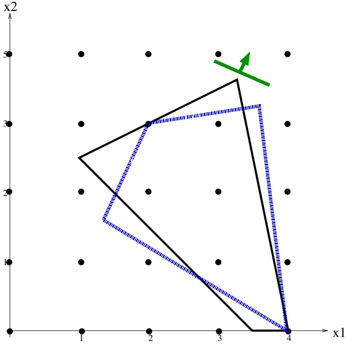


- **MIP solvers** are **efficient** but **"fail"** beyond a certain size.
- They barely exploit **"problem structure"**.
- The **"quality"** of the formulation is key for the solver.

A formulation is typically not unique

P and P' can be **alternative formulations** for (CO) if

$$(CO) \equiv \min\{cx : x \in P \cap \mathbb{N}^n\} \equiv \min\{c'x' : x' \in P' \cap \mathbb{N}^{n'}\}$$

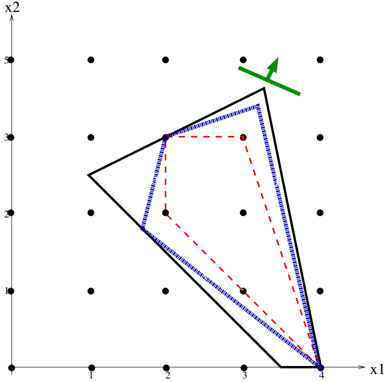


warning: can expressed in different variable-spaces.

Stronger formulation (in the same space)

Formulation $P' \subseteq \mathbb{R}^n$ is a **stronger** than $P \subseteq \mathbb{R}^n$ if $P' \subset P$. Then,

$$\min\{cx' : x' \in P'\} \geq \min\{cx : x \in P\}$$

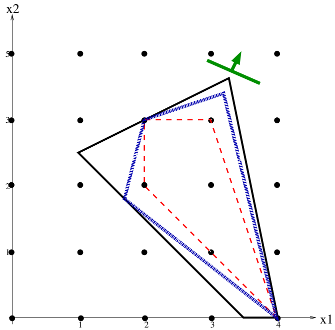


Dual Bound quality + considerations of **Size** + **Symmetry** issues

Ideal Formulation

The Convex hull of an IP set, P_I

$\text{conv}(P_I)$ is the smallest closed convex set containing P_I.



$\text{conv}(P_I)$ is an ideal polyhedron / formulation

If P_I is defined by rational data, $\text{conv}(P_I)$ is a polyhedron.

Outline

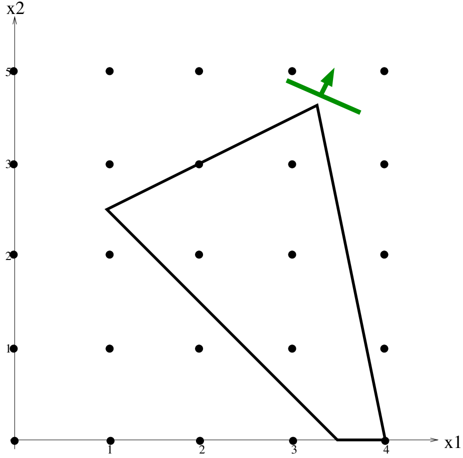
1 Extented Formulations
- Formulation
- Extended Formulation
- Reformulation
- Decomposition & Reformulation

2 Examples
- Steiner Tree Problem
- Traveling Salesman Problem
- Capacitated Network Design

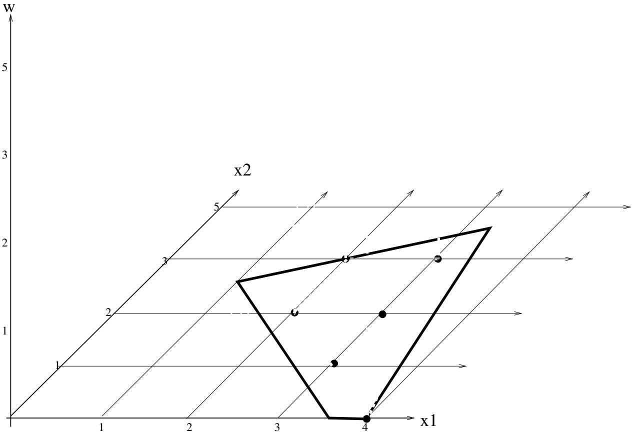
3 How to build them
- Variable Splitting
- DP based reformulation
- LP separation
- Union of Polyhedra
- Reduced coefficient & basis

4 Interests of Reformulations

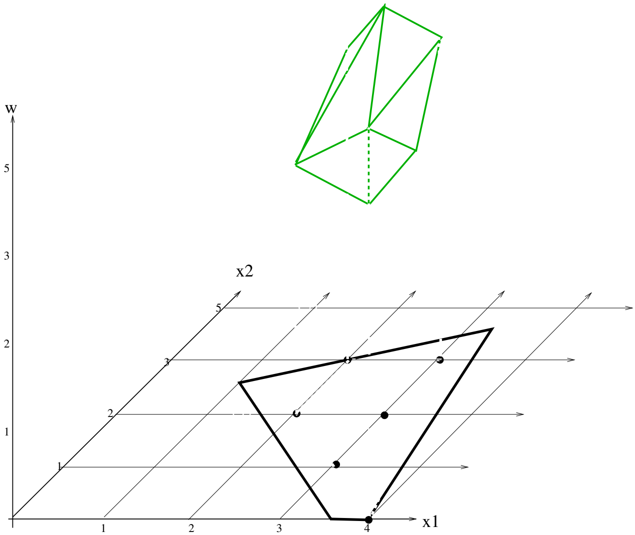
Given an initial **compact formulation**:



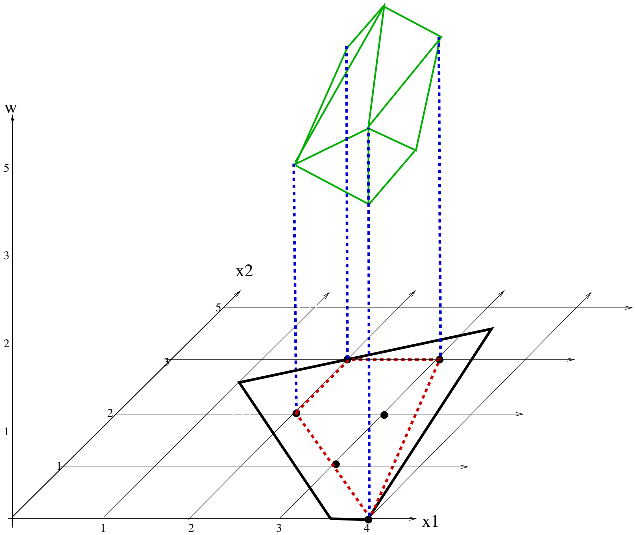
Extended Formulation



Extended Formulation



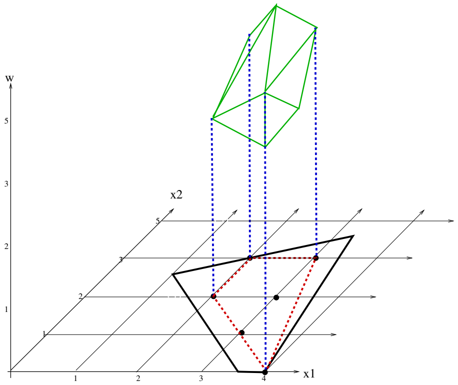
Extended Formulation



The Projection

of $Q = \{(x, w) \in \mathbb{R}^{n+e} : Gx + Hw \geq d\}$ on the x-space is:

$$\text{proj}_x(Q) := \{x \in \mathbb{R}^n : \exists\ w \in \mathbb{R}^e \text{ such that } (x, w) \in Q\}.$$



Projection

The Projection

of $Q = \{(x, w) \in \mathbb{R}^{n+e} : Gx + Hw \geq d\}$ on the x-space is:

$$\text{proj}_x(Q) := \{x \in \mathbb{R}^n : \exists\ w \in \mathbb{R}^e \text{ such that } (x, w) \in Q\}.$$

Farka's Lemma

Given $\tilde{x}$,

$$\{w \in \mathbb{R}^n_+ : Hw \geq (d - G\,\tilde{x}\,)\} \neq \emptyset$$

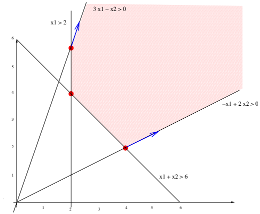
if and only if

$$\forall v \in \mathbb{R}^m_+ : vH \leq 0, \quad v(d - G\,\tilde{x}\,) \ \leq 0.$$

Hence, a polyhedral description of the projection in the x-space is:

$$\text{proj}_x(Q) = \{x \in \mathbb{R}^n : v^j(d - Gx) \ \leq 0 \quad j \in J\}$$

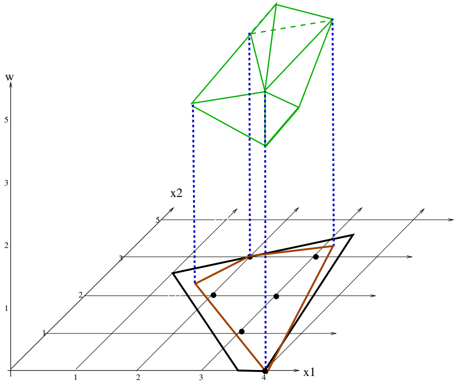
$\{v^j\}_{j \in J}$, exteme rays. of $\{v \in \mathbb{R}^m_+ : vH \leq 0\}$.



An extended formulation for an IP set $P_I \subseteq \mathbb{N}^n$

is a polyhedron $Q = \{(x,w) \in \mathbb{R}^{n+e} : Gx + Hw \geq d\}$ such that

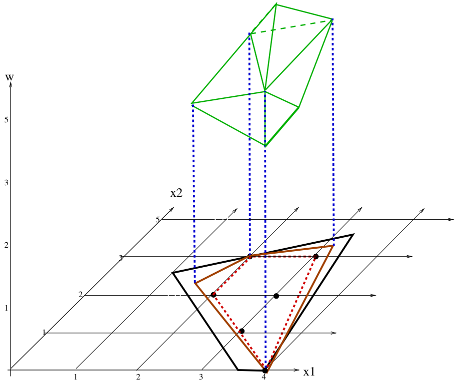
$$P_I = \text{proj}_x(Q) \cap \mathbb{N}^n.$$



An extended formulation for an IP set $P_I \subseteq \mathbb{N}^n$

is a polyhedron $Q = \{(x, w) \in \mathbb{R}^{n+e} : Gx + Hw \geq d\}$ such that

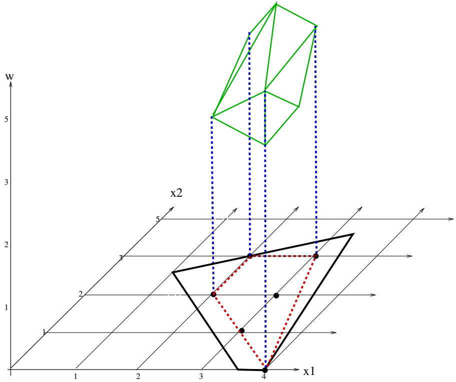
$$P_I = \text{proj}_x(Q) \cap \mathbb{N}^n.$$



A tight extended formulation for an IP set $P_I \subseteq \mathbb{N}^n$

is a polyhedron $Q = \{(x,w) \in \mathbb{R}^{n+e} : Gx + Hw \geq d\}$ such that

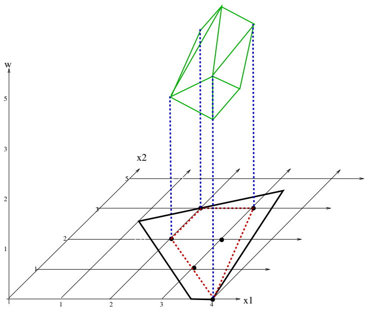
$$\mathbf{conv}(P_I) = \mathrm{proj}_x(Q).$$



Compact Extended Formulations

A formulation (resp. extended f.) is "Compact"

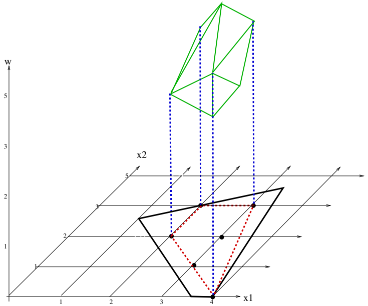
if the length of the description of P (resp. Q) is polynomial in the input length of the description of CO.



Compact Extended Formulations

A formulation (resp. extended f.) is "Compact"

if the length of the description of P (resp. Q) is polynomial in the input length of the description of CO.



Compactness of an Ideal Formulation

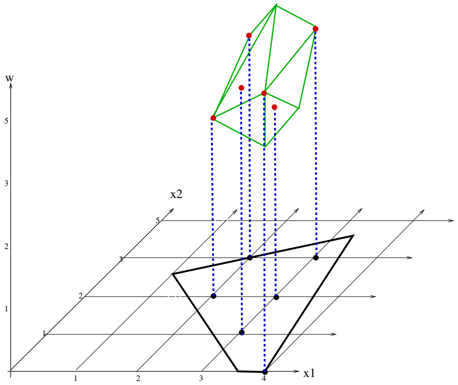
An ideal formulation cannot be compact unless CO is in $\mathscr{P}$.

IP Extended Formulations

An extended IP-formulation for an IP set $P_I \subseteq \mathbb{N}^n$

is an IP-set $Q_I = \{(x,w) \in \mathbb{R}^n \times \mathbb{N}^e : Gx + Hw \geq b\}$ s.t.

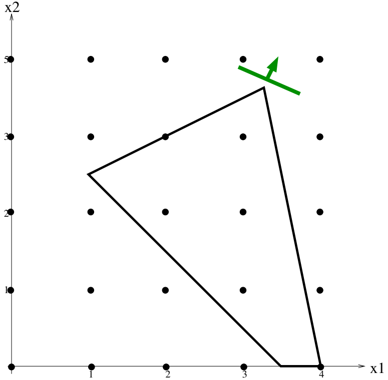
$$P_I = \text{proj}_x Q_I.$$

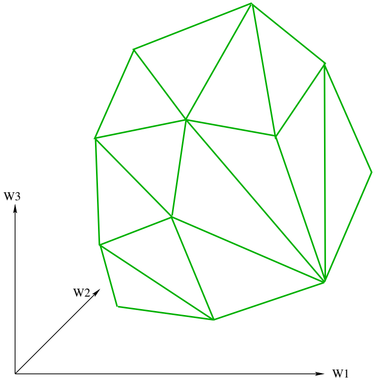


Outline

1. Extented Formulations
   - Formulation
   - Extended Formulation
   - Reformulation
   - Decomposition & Reformulation
2. Examples
   - Steiner Tree Problem
   - Traveling Salesman Problem
   - Capacitated Network Design
3. How to build them
   - Variable Splitting
   - DP based reformulation
   - LP separation
   - Union of Polyhedra
   - Reduced coefficient & basis
4. Interests of Reformulations

Change of variables: $x = T\,w$





Reformulation: a special case of extended formulation

An extended formulation based on a **change of variables**: $\mathbf{x} = \mathbf{Tw}$.

$$Q = \{(x,w) \in \mathbb{R}^{n+e} : \begin{array}{rcl} Tw & = & x \\ Ew & \geq & e \end{array}\}.$$

Then,

$$\text{proj}_x(Q) = T(W) := \{x = Tw \in \mathbb{R}^n : \underbrace{Ew \geq e, w \in \mathbb{R}^e}_{w \in W}\}.$$

A reformulation for an IP-set $P_I \subseteq \mathbb{N}^n$

is a polyhedron W along a linear transformation, $\mathbf{x} = \mathbf{Tw}$, s.t.

$$P_I = T(W) \cap \mathbb{N}^n$$

A IP-reformulation for an IP-set $P_I \subseteq \mathbb{N}^n$

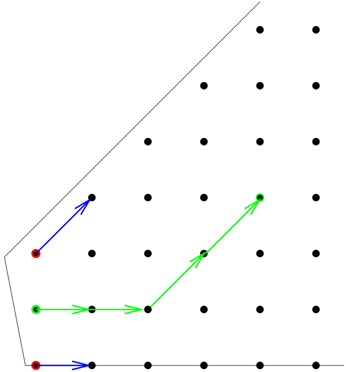
is an IP-set $W_I = W \cap \mathbb{N}^e$ along a linear transformation, $\mathbf{x} = \mathbf{Tw}$, s.t.,

$$P_I = T(W_I)$$

Minkowski's representation: a special case of reformulation

Polyhedron $\mathrm{conv}(P_I)$ can be defined by its extreme points and rays:

$$Q = \{(x,\lambda,\mu) \in \mathbb{R}^n \times \mathbb{R}_+^{|G|} \times \mathbb{R}_+^{|R|} : x = \sum_{g \in G} x^g \lambda_g + \sum_{r \in R} v^r \mu_r, \sum_{g \in G} \lambda_g = 1\}$$



change of variables: $\mathbf{x} = \mathbf{X}\,\lambda + \mathbf{V}\,\mu$.

Outline

1. Extented Formulations
   - Formulation
   - Extended Formulation
   - Reformulation
   - Decomposition & Reformulation
2. Examples
   - Steiner Tree Problem
   - Traveling Salesman Problem
   - Capacitated Network Design
3. How to build them
   - Variable Splitting
   - DP based reformulation
   - LP separation
   - Union of Polyhedra
   - Reduced coefficient & basis
4. Interests of Reformulations

Extended formulation based on a subset of constraints

Original formulation

$$[F] \equiv \min \left\{ c\,x \quad \begin{array}{rcl} A\,x & \geq & a \\ B\,x & \geq & b \\ x & \in & \mathbb{N}^n \end{array} \right\}$$

Subproblem

$$\mathsf{P} \equiv \left\{ \begin{array}{rcl} B\,x & \geq & b \\ x & \in & {\mathbb{R}_+}^n \end{array} \right\}$$

$$P_I = \mathsf{P} \cap \mathbb{N}^n$$

Decomposition + SP Reformulation

Extended formulation based on a subset of constraints

Original formulation

$$[F] \equiv \min \left\{ c\,x \quad \begin{array}{rcl} A\,x & \geq & a \\ B\,x & \geq & b \\ x & \in & \mathbb{N}^n \end{array} \right\}$$

Subproblem

$$\mathsf{P} \equiv \left\{ \begin{array}{rcl} B\,x & \geq & b \\ x & \in & \mathbb{R}_+{}^n \end{array} \right\}$$

$$P_I = \mathsf{P} \cap \mathbb{N}^n$$

$$P_I \to \left\{ x = Tw : Ew \geq e, w \in \mathbb{N}^e \right\}$$

Extended reformulation

$$[R] \equiv \min \left\{ c\,T\,w \quad \begin{array}{rcl} A\,T\,w & \geq & a \\ E\,w & \geq & e \\ w & \in & \mathbb{N}^p \end{array} \right\}$$

Extended formulation based on a subset of constraints

Original formulation

$$[\text{F}] \equiv \min \left\{ c\,x \quad \begin{array}{rcl} A\,x & \geq & a \\ B\,x & \geq & b \\ x & \in & \mathbb{N}^n \end{array} \right\}$$

Subproblem

$$\text{P} \equiv \left\{ \begin{array}{rcl} B\,x & \geq & b \\ x & \in & \mathbb{R}_+{}^n \end{array} \right\}$$

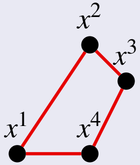
$$P_I = \text{P} \cap \mathbb{N}^n$$

Special case: Dantzig-Wolfe Reformulation

$$[\text{M}] \equiv \min \left\{ \sum_{g \in G} c\,x^g\,\lambda_g \quad \begin{array}{rcl} \sum_{g \in G} A\,x^g\,\lambda_g & \geq & a \\ \sum_{g \in G} \lambda_g & = & 1 \\ \lambda & \in & \{0,1\}^{|G|} \end{array} \right\}$$

Applying Minkowski

$$x = \sum_{g \in G} x^g\,\lambda_g$$



Price Decomposition (Dantzig)

$$(IP) \qquad z = \min\{cx : Ax \geq a, Bx \geq b, x \in \mathbb{Z}^n_+\}$$

where $Ax \geq a$ represent "complicating constraints" while the set $Bx \geq b$ is **"more tractable"**

- Relaxing $Ax \geq a$ while penalizing (pricing) their violation in the objective $\longrightarrow$ **Lagrangian relaxation**
- Reformulate the problem as selection of solutions to set $Bx \geq b$ that satisfy $Ax \geq a$ $\longrightarrow$ **Dantzig-Wolfe Reformulation – Column Generation**

Dantzig-Wolfe Decomposition: The block diagonal case

$$\begin{array}{lccccccccc}
\min & c^1 x^1 & + & c^2 x^2 & + & \dots & + & c^K x^K & \\
 & A^1 x^1 & + & A^2 x^2 & + & \dots & + & A^K x^K & \geq a \\
 & B^1 x^1 & & & & & & & \geq b^1 \\
 & & & B^2 x^2 & & & & & \geq b^2 \\
 & & & & & \ddots & & & \geq \vdots \\
 & & & & & & & B^K x^K & \geq b^K \\
 & x^1 & \in \mathbb{Z}^{n_1}_+, & x^2 & \in \mathbb{Z}^{n_2}_+, & \dots & x^K & \in \mathbb{Z}^{n_K}_+. &
\end{array}$$

Relaxing the constraints $\boldsymbol{Ax \geq a}$ decomposes the problem into $\boldsymbol{K}$ smaller size optimization problems:

$$\min\{c^k x^k : B^k x^k \geq b^K\}$$

The “complicating” constraints only depend on the aggregate variables:

$$\boldsymbol{y = \sum_{k=1}^K x^k} \qquad Y = \{y \in \mathbb{Z}^n_+ : \boldsymbol{Ay \geq a}\}.$$

Extended formulation based on a subset of variables

Original formulation

$$[F] \equiv \min \left\{ c\,x + h\,y \right.$$
$$Gx + Hy \geq d$$
$$\left. x \in \mathbb{N}^n, \quad y \in \mathbb{N}^p \right\}$$

Subproblem

$$\mathrm{P} \equiv \left\{ Hy \geq d - Gx \right.$$
$$\left. y \in \mathbb{N}^q \right\}$$

$$P_I = \mathrm{P} \cap \mathbb{N}^q$$

$$P_I \rightarrow \left\{ y = Tw : Ew \geq e(x), w \in \mathbb{R}^e \right\}$$

Extended reformulation

$$[R] \equiv \min \left\{ cx + h\,T\,w \right.$$
$$G\,x + H\,T\,w \geq d$$
$$E\,w \geq e(x)$$
$$\left. x \in \mathbb{N}^n, \quad w \in \mathbb{R}^e \right\}$$

$$\min cx + hy$$
$$Gx + Hy \geq d$$
$$\mathbf{x} \in \mathbb{Z}^{\mathbf{n}}, \mathbf{y} \in \mathbb{R}_+^{\mathbf{p}}$$

- The integer variables $\mathbf{x}$ are seen as the **"important" decisions**: ex. network design
- **Fix** $\mathbf{x}$ and compute the associated optimal $\mathbf{y}$ (solve SP).
- A **feedback loop** allowing one to adjust the $\mathbf{x}$ solution after obtaining the associated $\mathbf{y}$: Bender's cuts.

Benders Decomposition

$$\min\{cx + hy : \; Gx + Hy \geq d, \; x \in Z^n, y \in R^p_+\}$$

$$\min\{cx + \phi(x) : x \in \text{proj}_x(Q) \cap \mathbb{Z}^n\} \rightarrow \text{ a MIP}$$

where

$$Q = \{(x,y) \in R^n \times R^p_+ : Gx + Hy \geq d\}$$

$$\begin{aligned}
\phi(\mathbf{x}) &= \min\{hy : Hy \geq d - G\mathbf{x}, y \in R^p_+\} \\
&= \max\{u(d - G\mathbf{x}) : uH \leq h, u \in R^m_+\} \\
&= \max_{t=1,\ldots,T} \{u^t(d - G\mathbf{x})\} \quad \text{for } x \in \text{proj}_x(Q)
\end{aligned}$$

u^t are extreme points of $U = \{u \in \mathbb{R}^m_+ : uH \leq h\}$, v^r are extreme rays;

Bender's Master $\equiv \min c\mathbf{x} + \sigma$

$$\begin{aligned}
\sigma &\geq u^t(d - G\mathbf{x}) \;\; t = 1, \cdots, T \\
v^r(d - G\mathbf{x}) &\leq 0 \;\; r = 1, \cdots, R \\
\mathbf{x} &\in \mathbb{Z}^n
\end{aligned}$$

Benders Decomposition: The block diagonal case

$$
\begin{array}{rclclcllcl}
\min c\mathbf{x} & + & h^1y^1 & + & h^2y^2 & +\cdots & & + & h^Ky^K & & \\
G^1\mathbf{x} & + & H^1y^1 & & & & & & & \geq & d^1 \\
G^2\mathbf{x} & + & & & H^2y^2 & & & & & \geq & d^2 \\
 & \vdots & & & & & \ddots & & & \geq & \vdots \\
G^K\mathbf{x} & + & & & & & & & H^Ky^K & \geq & d^K \\
 & \mathbf{x} & \in \mathbb{N}^n, & & y^k & \in \mathbb{R}^q & k = 1,\ldots,K & & & &
\end{array}
$$

- Fixing $\mathbf{x}$ leads to a decomposition per block in $\mathbf{y^k}$ variables
- If moreover, blocks are identical, i.e. $(\mathbf{H^k}, \mathbf{h^k}) = (H, h) \forall k$, Benders cut generators obtained for one SP are valid forall k

Resource Splitting (Dantzig)

$$
\begin{array}{llllll}
\min c\mathbf{x} & +h^1y^1 & +h^2y^2 & +h^3y^3 & & \\
\mathbf{x} & -\mathbf{x^1} & -\mathbf{x^2} & -\mathbf{x^3} & = & 0 \\
 & G^1\mathbf{x^1}+H^1y^1 & & & \geq & d^1 \\
 & & G^2\mathbf{x^2}+H^2y^2 & & \geq & d^2 \\
 & & & G^3\mathbf{x^3}+H^3y^3 & \geq & d^3 \\
\mathbf{x} & \in \mathbb{N}^n, & (x^k, y^k) & \in \mathbb{R}^{n+q} & k=1,\ldots,3 &
\end{array}
$$

- split x using $\mathbf{x} = \sum_{\mathbf{k}} \mathbf{x^k}$ (or $x = x^k \forall k$)
- Lagrangian dualization of constraints $x = \sum_k x^k$ (or $x = x^k \forall k$)

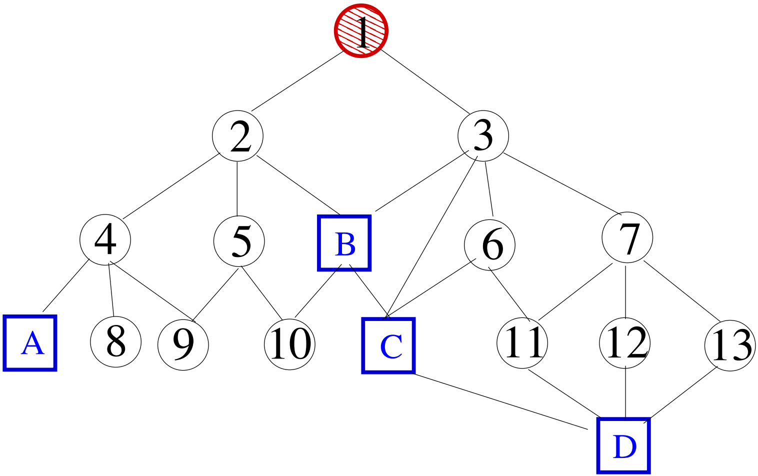
Outline

1. Extented Formulations
   - Formulation
   - Extended Formulation
   - Reformulation
   - Decomposition & Reformulation
2. Examples
   - Steiner Tree Problem
   - Traveling Salesman Problem
   - Capacitated Network Design
3. How to build them
   - Variable Splitting
   - DP based reformulation
   - LP separation
   - Union of Polyhedra
   - Reduced coefficient & basis
4. Interests of Reformulations

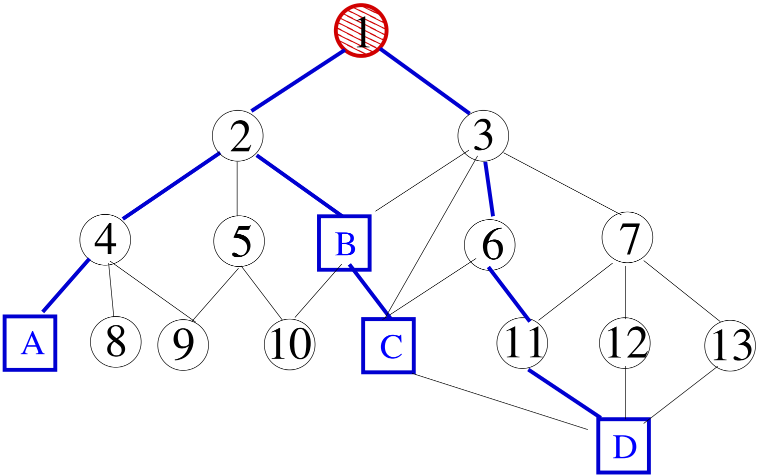
Outline

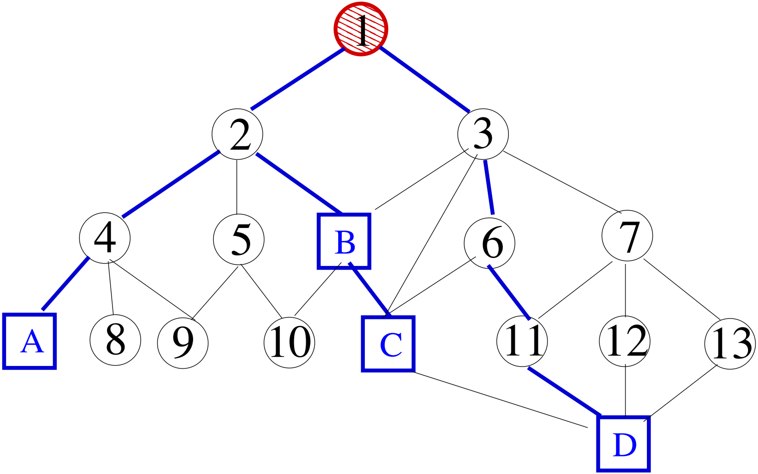
1. Extented Formulations
   - Formulation
   - Extended Formulation
   - Reformulation
   - Decomposition & Reformulation
2. Examples
   - Steiner Tree Problem
   - Traveling Salesman Problem
   - Capacitated Network Design
3. How to build them
   - Variable Splitting
   - DP based reformulation
   - LP separation
   - Union of Polyhedra
   - Reduced coefficient & basis
4. Interests of Reformulations

Example: Steiner Tree



Example: Steiner Tree





Special cases (that are “easy”):

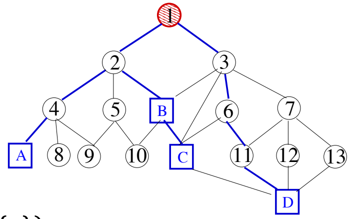
- $T = \{i\}$: shortest path from r to i
- $T = V \setminus \{r\}$: minimum cost spanning tree

Steiner Tree: Arc flow formulation

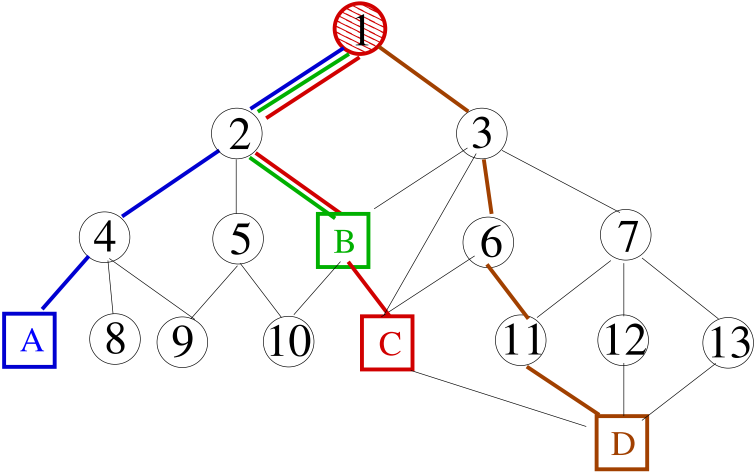
Variables

- $x_{ij} \in \{0,1\}$ — arc (i,j) is used or not
- $y_{ij} \in \mathbb{N}$ — number of connections going through (i,j)

$$
\begin{aligned}
\min \sum_{(i,j)\in A} c_{ij} x_{ij} & & \\
\sum_{j\in V^+(r)} y_{rj} &= |T| & \\
\sum_{j\in V^-(i)} y_{ji} - \sum_{j\in V^+(i)} y_{ij} &= 1 & i \in T \\
\sum_{j\in V^-(i)} y_{ji} - \sum_{j\in V^+(i)} y_{ij} &= 0 & i \in V \setminus (T \cup \{r\}) \\
y_{ij} &\leq |T|\, x_{ij} & (i,j) \in A \\
y &\in \mathbb{R}_+^{|A|}, & \\
x &\in \{0,1\}^{|A|} &
\end{aligned}
$$



Example: Steiner Tree



Steiner Tree: Multi commodity flow formulation

Variable splitting

- $w^t_{ij} \in \{0,1\}$ — arc (i,j) is used to connect terminal t
- $y_{ij} = \sum_k w^t_{ij}$ — defines a linear transformation

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

$$\sum_{j \in V^+(r)} w^t_{rj} = 1 \quad t \in T$$

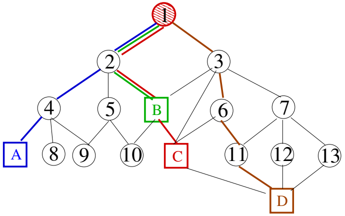
$$\sum_{j\in V^-(i)} w^t_{ji} - \sum_{j \in V^+(i)} w^t_{ij} = 1 \quad i = t \in T$$

$$\sum_{j\in V^-(i)} w^t_{ji} - \sum_{j \in V^+(i)} w^t_{ij} = 0 \quad i \in V \setminus \{r,k\},\ t \in T$$

$$w^t_{ij} \leq x_{ij} \quad (i,j) \in A,\ t \in T$$

$$w \in \mathbb{R}_+^{|K|\times|A|},$$

$$x \in \{0,1\}^{|A|}$$

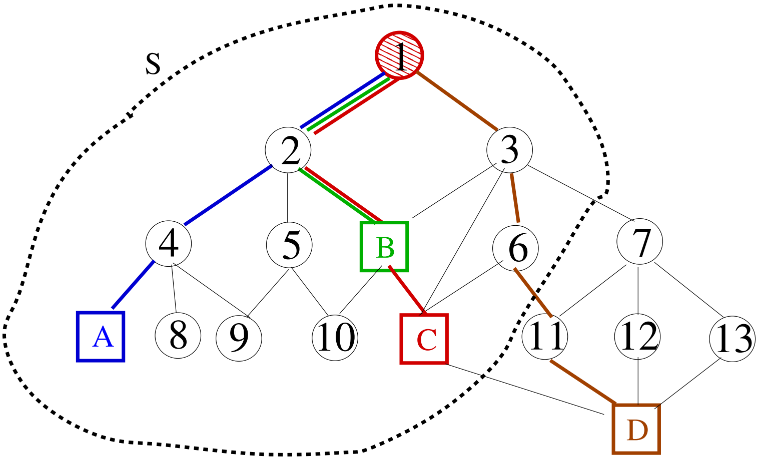


Steiner Tree: Path flow formulation

Decomposition

- $\lambda_p^t \in \{0,1\}$ — path p is used to connect terminal t

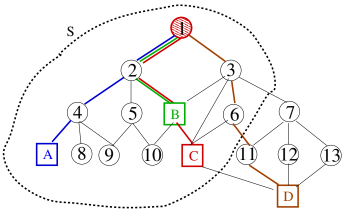
$$
\begin{aligned}
\min \sum_{(i,j)\in A} c_{ij} x_{ij} & & \\
\sum_{p\in P(k)} \lambda_p^t &= 1 \quad t \in T \\
\sum_{p\in P(k)} \delta_{ij}^p \lambda_p^t &\leq x_{ij} \quad (i,j) \in A,\ t \in T \\
\lambda_p^t &\in \{0,1\}^{|P(k)|} \quad t \in T \\
x_{ij} &\in \{0,1\}
\end{aligned}
$$


S
1
2
3
4
5
B
6
7
A
8
9
10
C
11
12
13
D

Steiner Tree: Network design formulation

projection in the x-space

$$
\begin{aligned}
\min \sum_{(i,j)\in A} c_{ij} x_{ij} & & & \\
\sum_{(i,j)\in\delta^+(S)} x_{ij} & \geq & 1 & \quad S \ni r, T \setminus S \neq \emptyset \\
x & \in & \{0,1\}^{|A|}, &
\end{aligned}
$$



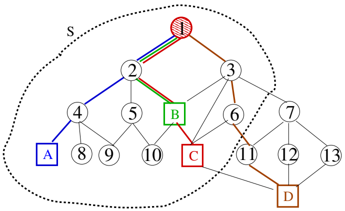
Steiner Tree: Network design formulation

projection in the x-space

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

$$\sum_{(i,j)\in\delta^+(S)} x_{ij} \geq 1 \quad S \ni r, T \setminus S \neq \emptyset$$

$$x \in \{0,1\}^{|A|},$$

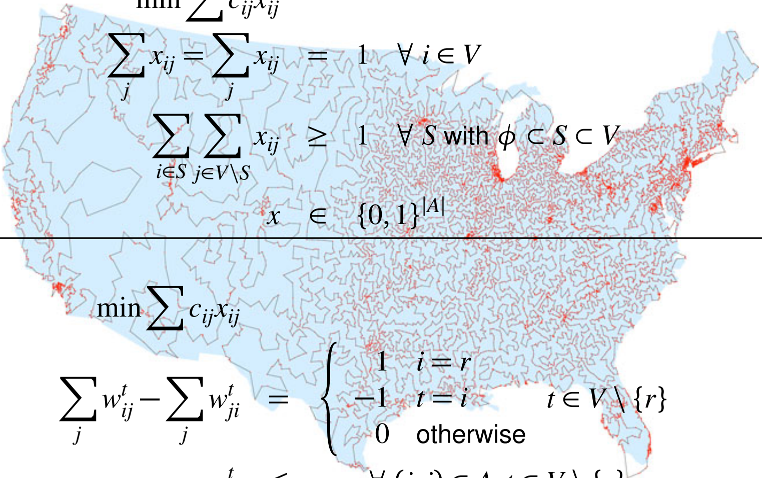


Note: This projection onto the x space

- has the **same LP value** than the multi-commodity flow formulation
- is **better than** the initial **compact** aggregate flow formulation.

Outline

1. Extented Formulations
   - Formulation
   - Extended Formulation
   - Reformulation
   - Decomposition & Reformulation
2. Examples
   - Steiner Tree Problem
   - Traveling Salesman Problem
   - Capacitated Network Design
3. How to build them
   - Variable Splitting
   - DP based reformulation
   - LP separation
   - Union of Polyhedra
   - Reduced coefficient & basis
4. Interests of Reformulations



$$\min \sum c_{ij} x_{ij}$$

$$\sum_j x_{ij} = \sum_j x_{ij} \quad = \quad 1 \quad \forall\, i \in V$$

$$\sum_{i \in S} \sum_{j \in V \setminus S} x_{ij} \quad \geq \quad 1 \quad \forall\, S \text{ with } \phi \subset S \subset V$$

$$x \quad \in \quad \{0,1\}^{|A|}$$

$$\min \sum c_{ij} x_{ij}$$

$$\sum_j w^t_{ij} - \sum_j w^t_{ji} \quad = \quad \begin{cases} 1 & i = r \\ -1 & t = i \\ 0 & \text{otherwise} \end{cases} \qquad t \in V \setminus \{r\}$$

$$w^t_{ij} \quad \leq \quad x_{ij} \quad \forall\, (i,j) \in A, t \in V \setminus \{r\}$$

$$x \in \{0,1\}^{|A|}, w \quad \in \quad [0,1]\ \forall (i,j) \in A, t \in V \setminus \{r\}$$

Outline

1. Extented Formulations
   - Formulation
   - Extended Formulation
   - Reformulation
   - Decomposition & Reformulation
2. Examples
   - Steiner Tree Problem
   - Traveling Salesman Problem
   - Capacitated Network Design
3. How to build them
   - Variable Splitting
   - DP based reformulation
   - LP separation
   - Union of Polyhedra
   - Reduced coefficient & basis
4. Interests of Reformulations

Multi-Commodity Capacitated Network Design

$$
\begin{aligned}
[\mathrm{F}] \equiv \min\{\sum_{ijk} c_{ij}^k x_{ij}^k + \sum_{ij} f_{ij} y_{ij} \\
\sum_j x_{ji}^k - \sum_j x_{ij}^k &= d_i^k \quad \forall i,k \\
\sum_k x_{ij}^k &\leq u_{ij} y_{ij} \quad \forall i,j \\
x_{ij}^k &\geq 0 \quad \forall i,j,k \\
y_{ij} &\in \mathbb{N} \quad \forall i,j\}
\end{aligned}
$$

$$
\begin{aligned}
[SP^{ij}] \equiv \min\{\sum_k c^k x^k + f\, y : \\
\sum_k x^k &\leq u\, y \\
x^k &\leq \min\{d^k, u\, y\} \forall k\}
\end{aligned}
$$

Network Design: Extended form. for the SPs

Let $y_{ij}^{s} = 1$ and $x_{ij}^{ks} = x_{ij}^{k}$ if $y_{ij} = s$.

$$[SP^{ij}] \equiv \min\{\sum_{ks} c_{ij}^{k}\, x_{ij}^{ks} + \sum_{s} f_{ij}\, s\, y_{ij}^{s} :$$

$$\begin{aligned} \sum_{s} y_{ij}^{s} &\leq 1 \\ (s-1)\, u_{ij}\, y_{ij}^{s} \leq \sum_{k} x_{ij}^{ks} &\leq s\, u_{ij}\, y_{ij}^{s} \quad \forall s \\ x_{ij}^{ks} &\leq \min\{d^{k}, s\, u_{ij}\}\, y_{ij}^{s} \quad \forall k, s\} \end{aligned}$$

Extended formulation for the arc design subproblem (Union of Polyhedra) **[Croxton, Gendron and Magnanti OR07]**

Network Design: extended formulation

$$[\mathrm{R}] \equiv \min\{\sum_{ijks} c_{ij}^k x_{ij}^{ks} + \sum_{ijs} f_{ij}\, s\, y_{ij}^s$$

$$\sum_{js} x_{ji}^{ks} - \sum_{js} x_{ij}^{ks} = d_i^k \quad \forall i,k$$

$$(s-1)\, u_{ij}\, y_{ij}^s \leq \sum_k x_{ij}^{ks} \leq s\, u_{ij}\, y_{ij}^s \quad \forall i,j,s$$

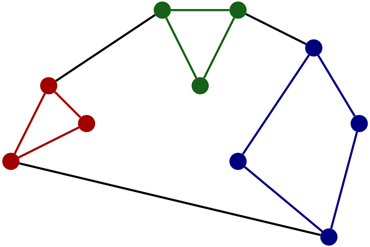
$$0 \leq x_{ij}^{ks} \leq d^k y_{ij}^s \quad \forall i,j,k,s$$

$$\sum_s y_{ij}^s = 1 \quad \forall i,j$$

$$y_{ij}^s \in \{0,1\} \quad \forall i,j,s\}$$

[Frangioni & Gendron, DAM09]

Network Design: Union of Polyhedra



Network Design: column genenration formulation

$$[\mathbf{M}] \equiv \min\{ \sum_{i,j,s,g\in G^{ij}} (c^k_{ij}\, x^g_{ks} + f_{ij}\, s\, y^g_s)\, \lambda^{ij}_g$$

$$\sum_{js} \sum_{g\in G^{ij}} x^g_{ks}\, \lambda^{ij}_g - \sum_{js} \sum_{g\in G^{ij}} x^g_{ks}\, \lambda^{ij}_g = d^k_i \quad \forall i,k$$

$$\sum_{g\in G^{ij}} \lambda^{ij}_g \leq 1 \quad \forall i,j$$

$$\lambda^{ij}_g \in \{0,1\} \quad \forall i,j,g \in G^{ij}\}$$

[Frangioni & Gendron WP10]

Outline

1. Extented Formulations
   - Formulation
   - Extended Formulation
   - Reformulation
   - Decomposition & Reformulation
2. Examples
   - Steiner Tree Problem
   - Traveling Salesman Problem
   - Capacitated Network Design
3. How to build them
   - Variable Splitting
   - DP based reformulation
   - LP separation
   - Union of Polyhedra
   - Reduced coefficient & basis
4. Interests of Reformulations

Outline

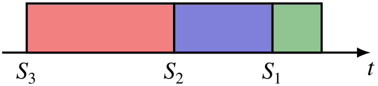
1. Extented Formulations
   - Formulation
   - Extended Formulation
   - Reformulation
   - Decomposition & Reformulation
2. Examples
   - Steiner Tree Problem
   - Traveling Salesman Problem
   - Capacitated Network Design
3. How to build them
   - Variable Splitting
   - DP based reformulation
   - LP separation
   - Union of Polyhedra
   - Reduced coefficient & basis
4. Interests of Reformulations

Ways to obtain extended formulations

- Variable Splitting
  - Multi-Commodity Flow: $x_{ij} = \sum_k x_{ij}^k$
  - Unary expansion: $x = \sum_{q=0}^{u} q\, w_q,\ \sum_{q=0}^{u} w_q = 1, w \in \{0,1\}^{u+1}$
  - Binary expansion: $x = \sum_{p=0}^{\log\lfloor u \rfloor} w_p,\ , w \in \{0,1\}^{\log u}$
- Dynamic Programming Solver → Network Flow LP [Martin et al]
- Separation is easy → Separation LP [Martin et al]
- Reduced coefficent / basis reformulations [Aardal et al]
- Union of Polyhedra [Balas]
- ...

Unary expansion: Time-Indexed Formulation

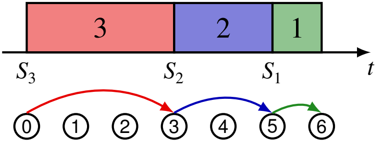
Single machine scheduling problem (with integer data):



$$S_j \geq S_i + p_i \text{ or } S_i \geq S_j + p_j \;\forall\; i,j$$

requires big M formulation: $S_j \geq S_i + p_i - M(1 - x_{ij})$.

Single machine scheduling problem (with integer data):



$$S_j \geq S_i + p_i \text{ or } S_i \geq S_j + p_j \ \forall \ i,j$$

Change of variables: $S_j = \sum_t t \, w_{jt}$

with $w_{jt} = 1$ iff job j starts at the beginning of $[t, t+1]$.

$$\sum_{j \in J} w_{j0} = 1$$

$$\sum_{j \in J} w_{jt} - \sum_{j \in J} w_{j,t-p_j} = 0 \quad \forall t \geq 1$$

Ways to obtain extended formulations

- Variable Splitting
  - Multi-Commodity Flow: $x_{ij} = \sum_k x_{ij}^k$
  - Unary expansion: $x = \sum_{q=0}^{u} q\, w_q,\ \sum_{q=0}^{u} w_q = 1, w \in \{0,1\}^{u+1}$
  - Binary expansion: $x = \sum_{p=0}^{\log\lfloor u\rfloor} w_p,\ , w \in \{0,1\}^{\log u}$
- Dynamic Programming Solver → Network Flow LP [Martin et al]
- Separation is easy → Separation LP [Martin et al]
- Reduced coefficent / basis reformulations [Aardal et al]
- Union of Polyhedra [Balas]
- ...

Outline

1 Extented Formulations
- Formulation
- Extended Formulation
- Reformulation
- Decomposition & Reformulation

2 Examples
- Steiner Tree Problem
- Traveling Salesman Problem
- Capacitated Network Design

3 How to build them
- Variable Splitting
- DP based reformulation
- LP separation
- Union of Polyhedra
- Reduced coefficient & basis

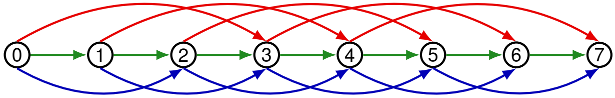
4 Interests of Reformulations

$$\max\{\sum_i p_i x_i : \sum_i a_i x_i \leq b, x_i \in \mathbb{N}\}$$

- **DP Recursion:** $V(c) = \max_{i=1,\ldots,n: c \geq a_i}\{V(c - a_i) + p_i\}$
- **in LP form:**

$$\begin{aligned} \min V(b) & & & \\ V(c) - V(c - a_i) & \geq & p_i & \qquad i = 1, \ldots, n,\ c = a_i, \cdots, b \\ V(0) & = & 0 & \end{aligned}$$

- **its Dual:** "longest path problem"



DP based reformulation: the knapsack example

$$\max\{\sum_i p_i x_i : \sum_i a_i x_i \leq b, x_i \in \mathbb{N}\}$$

- **DP Recursion:** $V(c) = \max_{i=1,\ldots,n: c \geq a_i}\{V(c-a_i)+p_i\}$
- **in LP form:**

$$\begin{array}{rcll} & \min V(b) & & \\ V(c)-V(c-a_i) & \geq & p_i & i=1,\ldots,n,\ c=a_i,\cdots,b \\ V(0) & = & 0 & \end{array}$$

- **its Dual:** "longest path problem"

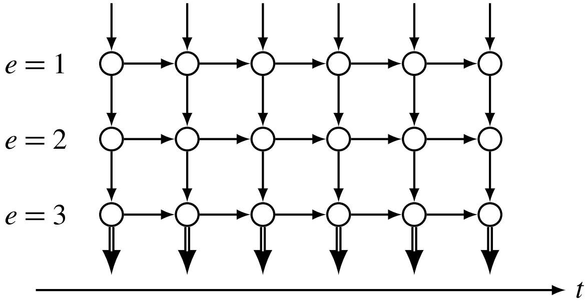
$$\begin{array}{rcll} \max \sum_{j=1}^{n}\sum_{r=0}^{b-a_i} c_i w_{ic} & & & \\ \sum_i w_{ic} & = & 1 & c=0 \\ \sum_i w_{ic} - \sum_i w_{i,c-a_i} & = & 0 & c=1,\cdots,b-1 \\ \sum_i w_{i,c-a_i} & = & 1 & c=b \\ w_{ic} & \geq & 0 & i=1,\cdots,n;\ c=0,\cdots,b-a_i \end{array}$$

DP based reformulation: Multi-Echelon Lot-Sizing

Variables

- $x_{e,t}$ — production of intermediate product of echelon e in period t
- $s_{e,t}$ — stock of echelon e product at the end of period t

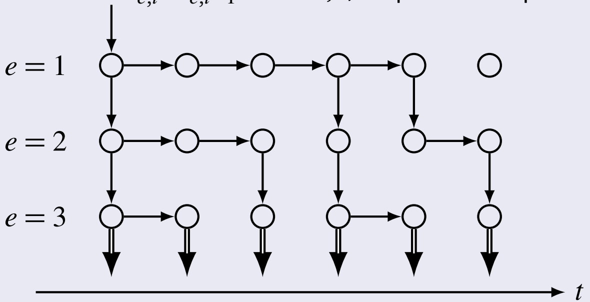
$$\begin{aligned} x_{e,t} + s_{e,t-1} &= x_{e+1,t} + s_{e,t} && \text{for } e = 1, \ldots, E-1 \\ x_{e,t} + s_{e,t-1} &= d_t + s_{e,t} && \text{for } e = E \end{aligned}$$



DP based reformulation: Multi-Echelon Lot-Sizing

Dominance property

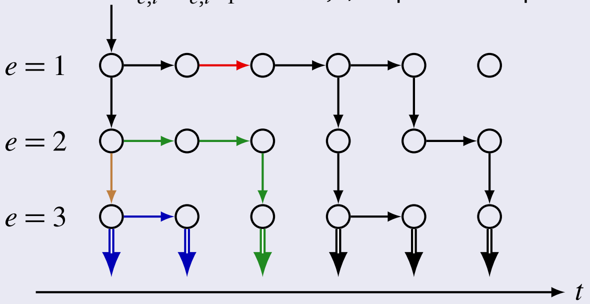
$\exists$ opt solution where $x_{e,t} \cdot s_{e,t-1} = 0 \; \forall e, t, \Rightarrow$ production plan is a tree:



DP based reformulation: Multi-Echelon Lot-Sizing

Dominance property

$\exists$ opt solution where $x_{e,t} \cdot s_{e,t-1} = 0 \;\forall e, t, \Rightarrow$ production plan is a tree:



Dynamic programming

State (e, t, a, b) corresponds to accumulating at echelon e in period t a production covering exactly the demand of periods $a, \ldots, b$.

$$
\begin{aligned}
V(e,t,a,b) &= \min\{V(e,t+1,a,b), \\
&\quad \min_{l=a,\ldots,b}\{V(e+1,t,a,l) + c^k_{et} D^k_{al} + f^k_{et} + V(e,t+1,l+1,b)\}\}
\end{aligned}
$$

- **DP Recursion:**

$$V(e,t,a,b) = \min\{V(e,t+1,a,b), \min_{l=a,\dots,b}\{V(e+1,t,a,l) + c^k_{et} D^k_{al} + f^k_{et} + V(e,t+1,l+1,b)\}\}$$

- **in LP form:**

$$\begin{aligned} &\max V(1,1,1,T) \\ V(e,t,a,b) &\leq V(e,t+1,a,b) \;\forall e,t,a,b \\ V(e,t,a,b) &\leq V(e+1,t,a,l) + c^k_{et} D^k_{al} + f^k_{et} + V(e,t+1,l+1,b) \;\forall e,t,a,b,l \\ V(E+1,t,a,b) &= 0 \;\forall t,a,b \end{aligned}$$

- **its Dual:** flow on hyper-arcs

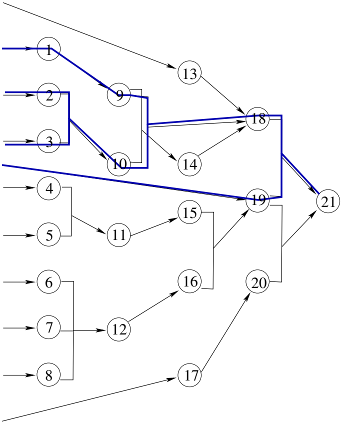
  $w_{e,t,a,l,b} = 1$ if at echelon e in period t production covers demands from period a to period l, while the rest of demand up to b, shall be covered in the future.

DP based reformulations

[Martin et al OR90] When a problem can be solved by dynamic programming,

$$V(l) = \min_{(J,l)\in\mathscr{A}} \{\sum_{j\in J} V(j) + c(J,l)\},$$

an extended formulation consist in modeling a decision tree in an hyper-graph



Ways to obtain extended formulations

- Variable Splitting
  - Multi-Commodity Flow: $x_{ij} = \sum_k x_{ij}^k$
  - Unary expansion: $x = \sum_{q=0}^{u} q\, w_q,\ \sum_{q=0}^{u} w_q = 1, w \in \{0,1\}^{u+1}$
  - Binary expansion: $x = \sum_{p=0}^{\log\lfloor u \rfloor} w_p,\ , w \in \{0,1\}^{\log u}$
- Dynamic Programming Solver → Network Flow LP [Martin et al]
- Separation is easy → Separation LP [Martin et al]
- Reduced coefficent / basis reformulations [Aardal et al]
- Union of Polyhedra [Balas]
- …

Outline

1 Extented Formulations
- Formulation
- Extended Formulation
- Reformulation
- Decomposition & Reformulation

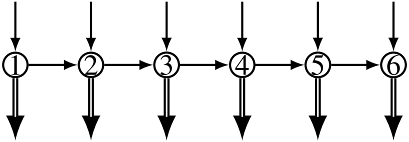
2 Examples
- Steiner Tree Problem
- Traveling Salesman Problem
- Capacitated Network Design

3 How to build them
- Variable Splitting
- DP based reformulation
- LP separation
- Union of Polyhedra
- Reduced coefficient & basis

4 Interests of Reformulations

Reformulation of the Uncapacitated Lot-Sizing: LP sep. & reform.

$$\min \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} h_t s_t + \sum_{t=1}^{n} q_t y_t$$
$$s_{t-1} + x_t = d_t + s_t \ \forall\ t$$
$$x_t \leq M y_t \ \forall\ t$$
$$s, x \in R_+^n, y \in \{0,1\}^n$$



Reformulation of the Uncapacitated Lot-Sizing: LP sep. & reform.

$$\min \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} h_t s_t + \sum_{t=1}^{n} q_t y_t$$
$$s_{t-1} + x_t = d_t + s_t \;\forall\, t$$
$$x_t \le M y_t \;\forall\, t$$
$$s, x \in R_+^n, y \in \{0,1\}^n$$

Facet-defining inequalities: $L = \{1, \ldots, l\}$, $S \subseteq L$

$$\sum_{j \in S} x_j + \sum_{j \in L \setminus S} d_{jl} y_j \ge d_{1l}$$

Let $\mu_{jl} = \min\{x_j, d_{jl} y_j\}$ for $1 \le j \le l \le n \Rightarrow$ a tight and compact extended formulation is obtained from the OF by adding:

$$\begin{aligned} \sum_{j=1}^{l} \mu_{jl} &\ge d_{1l} \quad 1 \le l \le n \\ \mu_{jl} &\le x_j \quad 1 \le j \le l \le n \\ \mu_{jl} &\le d_{jl} y_j \quad 1 \le j \le l \le n. \end{aligned}$$

LP separation & reformulation: example 2

Robust Optimization:

$$\begin{aligned} \min\{ \quad & c\,x \\ & A^{\xi} x \geq a \quad \forall \xi \in \Xi \\ & x \in \mathbb{N}^n \}, \end{aligned}$$

with $A^{\xi} = A + \sum_k A^k \xi_k$

and $\Xi = \{\xi \in \mathbb{R}^K : B\,\xi \geq b\}$.

The **separation problem**:

$$\sum_j a_{ij} x + \min\{\sum_k \sum_j a^k_{ij} \xi_k x_j : B\xi \geq b, \xi \in \mathbb{R}^K\} \geq a_{i0}? \quad \forall i$$

$$\sum_j a_{ij} x + \max\{u\,b : \; u\,B \leq \sum_j a^k_{ij} x_j \;\forall k, \; u \in \mathbb{R}^m\} \geq a_{i0}? \quad \forall i$$

The **extended formulation**:

$$\begin{aligned} \min\{ \quad & c\,x \\ \sum_j a_{ij} x + u\,b & \geq a_{i0} \quad \forall i \\ u\,B & \leq \sum_j a^k_{ij} x_j \quad \forall k \\ u \in \mathbb{R}^m \quad & x \in \mathbb{N}^n \}. \end{aligned}$$

Outline

1 Extented Formulations
- Formulation
- Extended Formulation
- Reformulation
- Decomposition & Reformulation

2 Examples
- Steiner Tree Problem
- Traveling Salesman Problem
- Capacitated Network Design

3 How to build them
- Variable Splitting
- DP based reformulation
- LP separation
- Union of Polyhedra
- Reduced coefficient & basis

4 Interests of Reformulations

The $1-k$ Configuration example

$$Y = \{(x_0, x) \in \{0,1\}^{n+1} : kx_0 + \sum_{j=1}^{n} x_j \leq n\}.$$

$$Y^0 = \{x_0 = 0, \sum_{j=1}^{n} x_j \leq n\} \quad \cup \quad Y^1 = \{x_0 = 1, \sum_{j=1}^{n} x_j \leq n-k\}$$

Tight extended formulation:

$$\begin{array}{rcl} x_j & = & x_j^0 + x_j^1 \; j = 1,\ldots,n \\ x_j^0 & \leq & 1 - x_0 \; j = 1,\ldots,n \\ x_j^1 & \leq & x_0 \quad j = 1,\ldots,n \\ \sum_{j=1}^{n} x_j^1 & \leq & (n-k)x_0 \\ x & \in & [0,1]^{3n-2} \end{array}$$

Outline

1. Extented Formulations
   - Formulation
   - Extended Formulation
   - Reformulation
   - Decomposition & Reformulation
2. Examples
   - Steiner Tree Problem
   - Traveling Salesman Problem
   - Capacitated Network Design
3. How to build them
   - Variable Splitting
   - DP based reformulation
   - LP separation
   - Union of Polyhedra
   - Reduced coefficient & basis
4. Interests of Reformulations

The knapsack problem example

$$X = P^1 \cap \mathbb{Z}^n = P^2 \cap \mathbb{Z}^n$$

where

$$P^1 = \{x \in [0,1]^5 : 97x_1 + 65x_2 + 47x_3 + 46x_4 + 25x_5 \leq 136\}$$

$$P^2 = \{x \in [0,1]^5 : 5x_1 + 3x_2 + 3x_3 + 2x_4 + 1x_5 \leq 6\}$$

$$P^2 \subset P^1$$

Outline

1. Extented Formulations
   - Formulation
   - Extended Formulation
   - Reformulation
   - Decomposition & Reformulation
2. Examples
   - Steiner Tree Problem
   - Traveling Salesman Problem
   - Capacitated Network Design
3. How to build them
   - Variable Splitting
   - DP based reformulation
   - LP separation
   - Union of Polyhedra
   - Reduced coefficient & basis
4. Interests of Reformulations

Extended formulation: Interests

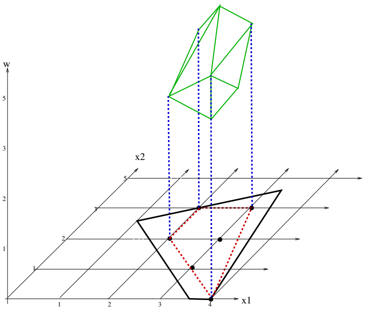
1. **Improved formulation** (better LP bound & rounding heuristic)

extra variables

↓

tighter relations,

linearisation



Extended formulation: Interests

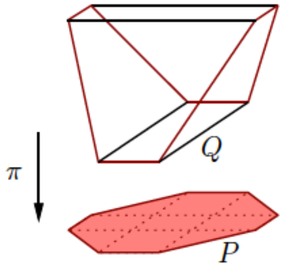
1. **Improved formulation** (better LP bound & rounding heuristic)
2. **Simpler formulation** (captures the combinatorial structure)

extra variables

↓

fewer constaints

structure built into var. definitions



Extended formulation: Interests

1. **Improved formulation** (better LP bound & rounding heuristic)
2. **Simpler formulation** (captures the combinatorial structure)
3. **Direct use of a MIP-Solver** (solved by standard tools)

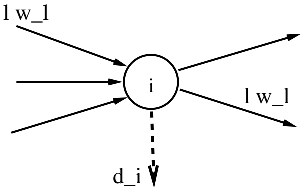
Extended formulation: Interests

1. **Improved formulation** (better LP bound & rounding heuristic)
2. **Simpler formulation** (captures the combinatorial structure)
3. **Direct use of a MIP-Solver** (solved by standard tools)
4. **Rich variable space** (to express cuts or branching)

Vehicle routing: $x_a = \sum_{l=0,\dots,C} w_l^a$
$w_q^a = 1$ if vehicle on arc a with load l,

$$\sum_l \sum_{a \in \delta^-(i)} l w_l^a - \sum_l \sum_{a \in \delta^+(i)} l w_l^a = d_i$$

$\rightarrow$ knapsack cover cuts.

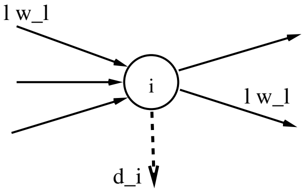


[Uchoa]

1. **Improved formulation** (better LP bound & rounding heuristic)
2. **Simpler formulation** (captures the combinatorial structure)
3. **Direct use of a MIP-Solver** (solved by standard tools)
4. **Rich variable space** (to express cuts or branching)

Vehicle routing: $x_a = \sum_{l=0,\dots,C} w_l^a$
$w_q^a = 1$ if vehicle on arc a with load l,

$$\sum_l \sum_{a\in\delta^-(i)} l w_l^a - \sum_l \sum_{a\in\delta^+(i)} l w_l^a \quad = \quad d_i$$



→ knapsack cover cuts.

[Uchoa]

5. Reformulation can help to eliminate **Symmetries**