Reformulation and Decomposition of Integer Programs

François Vanderbeck[1] and Laurence A. Wolsey[2]

(Reference: CORE DP 2009/16)

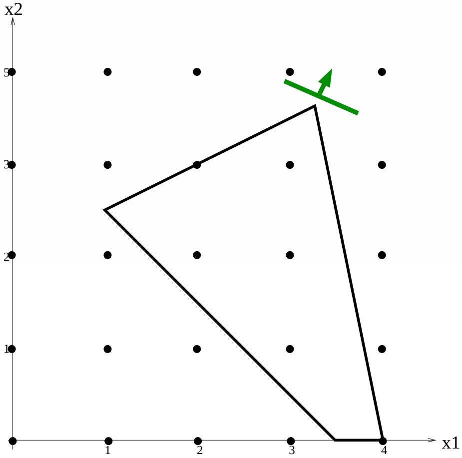
(1) Université Bordeaux 1 & INRIA-Bordeaux
(2) Université de Louvain, CORE.

JFRO, March 6, 2009.

Integer Program

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

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



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 $P = \{x \in \mathbb{R}^n_+ : Ax \geq a\}$.

Mixed Integer Program

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

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

- **MIP solvers** are quite **efficient** but can still **"fail"** on many problems (beyond a certain size).
- They barely exploit **"problem structure"**.

1] To introduce new variables → better LP bounds

- tighter relations between variables
- new variables for branching
- new variables to formulate cuts

2] To work is a smaller dimensional space (if size is an issue)

3] To eliminate symmetry

4] To lead to a decomposition approach and specific (more effective) algorithms

Arc flow formulation

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

Multi commodity arc flow formulation: $y_{ij} = \sum_k w_{ij}^k$

$$
\begin{aligned}
\min \sum_{(i,j)\in A} c_{ij}x_{ij} & & & \\
\sum_{j\in V^+(r)} w_{rj}^k & = & 1 & \quad k \in T \\
\sum_{j\in V^-(i)} w_{ji}^k - \sum_{j\in V^+(i)} w_{ij}^k & = & 0 & \quad i \in V \setminus \{r,k\},\ k \in T \\
\sum_{j\in V^-(i)} w_{jk}^k - \sum_{j\in V^+(i)} w_{kj}^k & = & 1 & \quad k \in T \quad i \in T \\
w_{ij}^k & \leq & x_{ij} & \quad (i,j) \in A,\ k \in K \\
w & \in & \mathbb{R}_+^{|K|\times|A|}, & \\
x & \in & \{0,1\}^{|A|} &
\end{aligned}
$$

Network design formulation

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

Note: This projection onto the x space has the same LP value than the multi-commodity arc flow formulation (itself better than the initial arc flow formulation).

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

1] Constraint decomposition

- $X = Y \cap Z$
  - tighter formulation for Z
  - "implicitly" enumerate set Z (**Lagrangian/Dantzig-Wolfe approach** relying on a tractable optimisation over Z, **OPT(Z)**)
  - "implicitly" give the polyhedral description of $conv(Z)$ (**Cutting Plane approach** relying on a tractable separation over Z, **SEP(Z)**)
- $X = Y \cup Z$ (Variable splitting, Disjunctive Cuts)

2] Variable decomposition: $x = (x^1, x^2)$

- Fixing x^1 yields an "easier" problem in x^2 (**Bender's approach** relying on cut generation for $\mathbf{proj}_{x^1} X$).

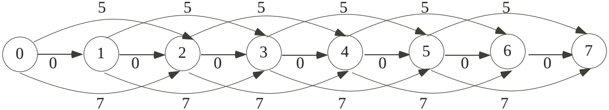
OPT(Z) $\in \mathcal{P}$ *iff* **SEP(Z)** $\in \mathcal{P}$

Item Assignment Formulation

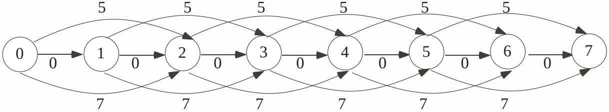
$$\min \sum_{k=1}^{K} u_k$$

$$\begin{aligned} \sum_{k=1}^{K} x_{ik} &= 1 & \forall i \\ \sum_{i} s_i \, x_{ik} &\leq u_k & \forall k \\ u_k, \; x_{ik} &\in \{0,1\} & \forall i, k \end{aligned}$$

$$Z^k = \{(u,x) \in \{0,1\}^{n+1} : \sum_i s_i \, x_i \leq u\}$$



Arc Flow Formulation



$$\min \sum_t w_{0t}$$

$$\sum_t w_{t-s_i,t} = 1 \qquad \forall i$$

$$\sum_j w_{jt} - \sum_j w_{tj} = 0 \qquad t = 1, \cdots, b-1$$

$$w_{jt} \in \{0,1\} \qquad \forall (j,t)$$

Definitions

Combinatorial Optimization Problem

$$(CO) \quad \min\{c(x):\ x \in X\}$$

where X is the "discrete" set of feasible solutions.

Definitions

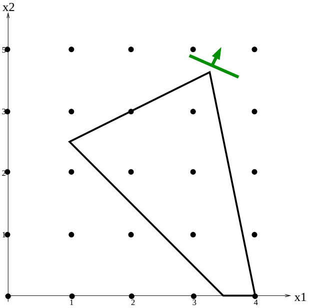
Combinatorial Optimization Problem

$$(CO) \quad \min\{c(x) : \ x \in X\}$$

where X is the "discrete" set of feasible solutions.

Polyheron

$P \subseteq \mathbb{R}^n$ is the intersection of a finite number of half-spaces: $\exists A \in \mathbb{R}^{m \times n}$, $a \in \mathbb{R}^m$ such that $P = \{x \in \mathbb{R}^n : Ax \geq a\}$.



Definitions

Combinatorial Optimization Problem

$$(CO) \quad \min\{c(x) : \; x \in X\}$$

where X is the "discrete" set of feasible solutions.

Polyheron

$P \subseteq \mathbb{R}^n$ is the intersection of a finite number of half-spaces: $\exists A \in \mathbb{R}^{m \times n}$, $a \in \mathbb{R}^m$ such that $P = \{x \in \mathbb{R}^n : Ax \geq a\}$.

Formulation

A polyhedron P is a formulation for (CO) if $X = P \cap \mathbb{Z}^n$ and (CO) can be modeled by the **Integer Program**:

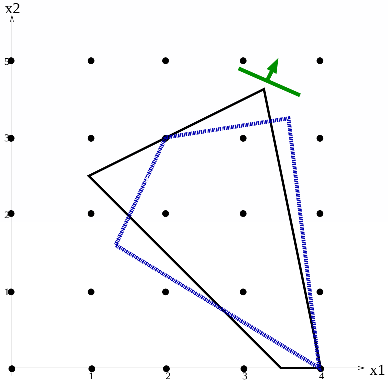
$$(IP) \quad \min\{cx : x \in P \cap \mathbb{Z}^n\}.$$

Definitions

Reformulation (loose definition)

P' is a **reformulation** for (CO) if it provides an alternative polyhedral description:

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



Definitions

Reformulation (loose definition)

P' is a **reformulation** for (CO) if it provides an alternative polyhedral description:
$(CO) \equiv \min\{cx : x \in P \cap \mathbb{Z}^n\} \equiv \min\{c'x' : x' \in P' \cap \mathbb{Z}^{n'}\}$

Stronger formulation

Reformulation $P' \subseteq \mathbb{R}^n$ is a **stronger** than $P \subseteq \mathbb{R}^n$ if $P' \subset P$:
$\min\{cx : x \in X\} \geq \min\{cx : x \in P'\} \geq \min\{cx : x \in P\}$.

Note: If P' and P are in different variable space, one can compare P and $\text{proj}_x(P')$ (or P and TP', T is a lin. transf.).

Definitions

Reformulation (loose definition)

P' is a **reformulation** for (CO) if it provides an alternative polyhedral description:

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

Stronger formulation

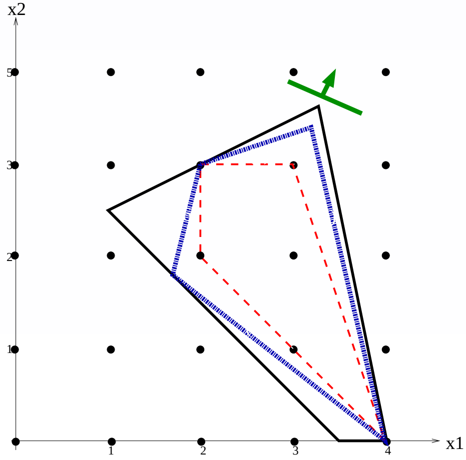
Reformulation $P' \subseteq \mathbb{R}^n$ is a **stronger** than $P \subseteq \mathbb{R}^n$ if $P' \subset P$:
$\min\{cx : x \in X\} \geq \min\{cx : x \in P'\} \geq \min\{cx : x \in P\}$.

Note: If P' and P are in different variable space, one can compare P and $\text{proj}_x(P')$ (or P and TP', T is a lin. transf.).

Ideal formulation

Given $X \subseteq \mathbb{R}^n$, the **convex hull of** X, denoted $\text{conv}(X)$, is the smallest closed convex set containing X. The convex hull of an integer set X defined by rational data is a polyhedron.

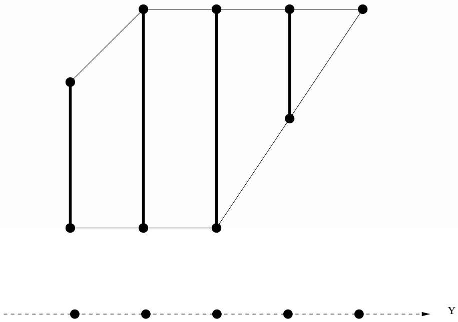
Comparing P, P', and $\mathrm{conv}(X)$



Definitions

Projection

of a set $U \subseteq \mathbb{R}^n \times \mathbb{R}^p$ on the first n variables,
$x = (x_1, \cdots, x_n)$, is
$\text{proj}_x(U) = \{x \in \mathbb{R}^n : \exists\ w \in \mathbb{R}^p \text{ with } (x, w) \in U\}$.



Definitions

Projection

of a set $U \subseteq \mathbb{R}^n \times \mathbb{R}^p$ on the first n variables, $x = (x_1, \cdots, x_n)$, is
$\text{proj}_x(U) = \{x \in \mathbb{R}^n : \exists\ w \in \mathbb{R}^p \text{ with } (x, w) \in U\}$.

An extended Formulation for $P \subseteq \mathbb{R}^n$

is a polyhedron $Q = \{(x, w) \in \mathbb{R}^{n+p} : Gx + Hw \geq d\}$ such that $P = \text{proj}_x(Q)$.

Definitions

Projection

of a set $U \subseteq \mathbb{R}^n \times \mathbb{R}^p$ on the first n variables,
$x = (x_1, \cdots, x_n)$, is
$\text{proj}_x(U) = \{x \in \mathbb{R}^n : \exists\ w \in \mathbb{R}^p \text{ with } (x, w) \in U\}$.

An extended Formulation for $P \subseteq \mathbb{R}^n$

is a polyhedron $Q = \{(x, w) \in \mathbb{R}^{n+p} : Gx + Hw \geq d\}$ such that $P = \text{proj}_x(Q)$.

An extended Formulation for an IP set $X \subseteq \mathbb{Z}^n$

is a polyhedron $Q \subseteq \mathbb{R}^{n+p}$ such that $X = \text{proj}_x(Q) \cap \mathbb{Z}^n$.

Definitions

Projection

of a set $U \subseteq \mathbb{R}^n \times \mathbb{R}^p$ on the first n variables,
$x = (x_1, \cdots, x_n)$, is
$\text{proj}_x(U) = \{x \in \mathbb{R}^n : \exists\ w \in \mathbb{R}^p \text{ with } (x, w) \in U\}$.

An extended Formulation for $P \subseteq \mathbb{R}^n$

is a polyhedron $Q = \{(x, w) \in \mathbb{R}^{n+p} : Gx + Hw \geq d\}$ such that $P = \text{proj}_x(Q)$.

An extended Formulation for an IP set $X \subseteq \mathbb{Z}^n$

is a polyhedron $Q \subseteq \mathbb{R}^{n+p}$ such that $X = \text{proj}_x(Q) \cap \mathbb{Z}^n$.

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

is a set
$Q = \{(x, w^1, w^2) \in \mathbb{R}^n \times \mathbb{Z}^{p_1} \times \mathbb{R}^{p_2} : Gx + H^1 w^1 + H^2 w^2 \geq b\}$
such that $X = \text{proj}_x Q$.

Definitions

Extended formulation & reformulation in a new variable space

If polyhedron Q is an extended IP-formulation for X and a linear transformation $x = Tw$ links the original x variables and the additional variables w, then
$\min\{cTw : ATw \geq a, w \in W\}$ provides an IP-reformulation.

Definitions

Extended formulation & reformulation in a new variable space

If polyhedron Q is an extended IP-formulation for X and a linear transformation $x = Tw$ links the original x variables and the additional variables w, then $\min\{cTw : ATw \geq a, w \in W\}$ provides an IP-reformulation.

A formulation P (resp. extended f. Q) is tight

if $P = \text{conv}(X)$ (resp. $\text{proj}_x(Q) = \text{conv}(X)$).

Definitions

Extended formulation & reformulation in a new variable space

If polyhedron Q is an extended IP-formulation for X and a linear transformation $x = Tw$ links the original x variables and the additional variables w, then $\min\{cTw : ATw \geq a, w \in W\}$ provides an IP-reformulation.

A formulation P (resp. extended f. Q) is tight

if $P = \text{conv}(X)$ (resp. $\text{proj}_x(Q) = \text{conv}(X)$).

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.

Definitions

Extended formulation & reformulation in a new variable space

If polyhedron Q is an extended IP-formulation for X and a linear transformation $x = Tw$ links the original x variables and the additional variables w, then
$\min\{cTw : ATw \geq a, w \in W\}$ provides an IP-reformulation.

A formulation P (resp. extended f. Q) is tight

if $P = \mathrm{conv}(X)$ (resp. $\mathrm{proj}_x(Q) = \mathrm{conv}(X)$).

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 $\mathcal{P}$.

Direct Reformulations

- Variable Splitting
  - Multi-Commodity Flow: $x_{ij} = \sum_k x_{ij}^k$
  - Unary expansion:
    $x = \sum_{q=0}^{C} q\ z_q,\ \sum_{q=0}^{C} z_q = 1, z \in \{0,1\}^{C+1}$
  - Binary expansion:
    $x = \sum_{p=0}^{P} 2^p w_p \leq C,\ , w \in \{0,1\}^{P+1}$, with
    $P = \log_2\lfloor C \rfloor$.
- Dynamic Programming for $OPT \rightarrow$ reformulation
- Separation is easy over a set $\rightarrow$ reformulation
- Union of Polyhedra
- Ad-hoc reformulations

In the context of a decomposition $X = Y \cap Z$, $Z \rightarrow Z'$

Direct Reformulations

Unary expansion: Time-Indexed Formulation

Single machine scheduling problem:

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

requires big M formulation. Instead let
$w_t^j = 1$ iff job j starts at the beginning of interval $[t-1, t]$.

$$\sum_{t=1}^{T} w_t^j = 1 \; \forall \; j$$
$$\sum_{j=1}^{n} \sum_{u=t-p_j+1}^{t} w_u^j \leq 1 \; \forall \; t$$
$$w_t^j \in \{0, 1\} \text{ for } t \in r_j, \ldots, d_j - p_j + 1, \; \forall j$$

$S_j = \sum_t (t-1) w_t^j$.

Direct Reformulations

Dynamic Programming based reformulation: knapsack example

$$G(t) = \max_{j=1,\ldots,n:t-a_j\geq 0}\{G(t-a_j)+c_j\}$$

$$\begin{gathered} \min G(b) \\ G(t) - G(t-a_j) \geq c_j \; j=1,\ldots,n, \; t=a_j,\cdots,b \\ G(0)=0. \end{gathered}$$

is the dual of a longest path problem.

Direct Reformulations

Union of Polyhedra: $1-k$ Configurations

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

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

Tight extended formulation:

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

Direct Reformulations

Separation → reformulation: Uncapacitated Lot-Sizing

$$\min \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} h_t s_t + \sum_{t=1}^{n} q_t y_t$$

$$s_{t-1} + x_t = d_t + s_t \ \forall \ t$$

$$x_t \leq M y_t \ \forall \ t$$

$$s, x \in R^n_+, y \in \{0,1\}^n$$

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

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

Let $\mu_{jl} = \min\{x_j, d_{jl} y_j\}$ for $1 \leq j \leq l \leq n$

⇒ tight and compact extended formulation:

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

Direct Reformulations

Ad hoc direct reformulation: vertex coloring

$$
\begin{array}{rcl}
\min \sum_k y_k & & \\
\sum_k x_{ik} & = & 1 \ \forall i \in V \\
x_{ik} + x_{jk} & \leq & y_k \ \ \forall k, \forall (i,j) \in E \\
x_{ik} & \leq & y_k \ \ \forall k, \forall i \in V \\
x_{ik} & \in & \{0,1\} \ \forall k, \forall i \in V, \\
y_k & \in & \{0,1\} \forall k.
\end{array}
$$

$x_{ik} = 1$ if node i gets color k (symmetries), or
$x_{ik} = 1$ if node i gets the same color as node k ($y_k = x_{kk}$).

Resource Decomposition

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

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

Resource Decomposition

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

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

where

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

$$\begin{aligned}
\phi(x) &= \min\{hy : Hy \geq d - Gx, y \in R^p_+\} \\
&= \max\{u(d - Gx) : uH \leq h, u \in R^m_+\} \\
&= \max_{t=1,\dots,T} u^t(d - Gx)
\end{aligned}$$

where $\{u^t\}_{t=1}^T$ are the extreme points of $U = \{u \in \mathbb{R}^m_+ : uH \leq h\}$.

$$\text{Bender's Ref} \equiv \min cx + \sigma$$

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

Branch-and-Cut

Bender's algorithm: branch-and-cut

i) Solve the restricted master LP. If it is infeasible, that node is infeasible, backtrack. Otherwise, record (σ^*, x^*).

ii) Solve the cut generation subproblem

$$\phi(x^*) = \min\{hy : Hy \geq d - Gx^*, y \in \mathbb{R}^p_+\},$$

or its dual $\max\{u(d - Gx^*) : uH \leq h, u \in \mathbb{R}^m_+\}$.

ii.a) The separation problem is infeasible and one obtains a new extreme ray v^r *with* $v^r(d - Gx^*) > 0$.
A feasibility cut, $v^r(d - Gx) \leq 0$, *is added to the master.*

ii.b) The separation subproblem is feasible, and one obtains a new dual extreme point u^t *with* $\phi(x^*) = u^t(d - Gx^*) > \sigma^*$.
An optimality cut $\sigma \geq u^t(d - Gx)$, *is added to the master.*

ii.c) The separation subproblem is feasible with optimal value $\phi(x^*) = \sigma^*$. *Then,* (x^*, σ^*) *is a solution to the linear master program at the node.*

Resource Decomposition

Block Diagonal Structure: Resource Decomposition

$$
\begin{array}{ccccccccccc}
\min cx & + & h^1y^1 & + & h^2y^2 & + \cdots & +h^Ky^K & & & & \\
G^1x & + & H^1y^1 & & & & & & & \geq & d^1 \\
G^2x & + & & & H^2y^2 & & & & & \geq & d^2 \\
 & \ddots & & & & & & \ddots & & \geq & \vdots \\
G^Kx & + & & & & & & & H^Ky^K & \geq & d^K \\
 & x & \in X, & & y^k & \in Z^k & k = 1, \ldots, K & & & &
\end{array}
$$

$$\zeta^k = \min\{h^k y^k : H^k y^k \geq d^k - G^k x, y^k \in Z^k\}.$$

Resource Decomposition

Multi-Machine Job Assignment Problem: integer SP

$$\min\{\sum_{k=1}^{K}\sum_{j=1}^{n} c_j^k x_j^k :$$

$$\sum_{k=1}^{K} x_j^k = 1 \ \forall \ j,$$

$$x^k \in Z^k \ \forall \ k\}$$

where $x^k \in Z^k$ if and only if the set $S^k = \{j : x_j^k = 1\}$ of jobs can be scheduled on machine k.
Otherwise one generates an infeasibility cut of the form:

$$\sum_{j \in S^k} x_j^k \leq |S^k| - 1$$

Price Decomposition

$$(IP) \qquad z = \min\{cx : \underbrace{Dx \geq d, Bx \geq b, x \in \mathbb{Z}^n_+}_{x \in X}\}$$

where $Dx \geq d$ represent "complicating constraints" while the set $Z = \{x \in \mathbb{Z}^n_+ : Bx \geq b\}$ is **"more tractable"** (**OPT(Z)**)

- Relaxing $Dx \geq d$ while penalizing (pricing) their violation in the objective → **Lagrangian relaxation**
- Reformulate the problem as selecting a solution from Z that satisfy $Dx \geq d$ → **Dantzig-Wolfe Reformulation – Column Generation**

Price Decomposition

The block diagonal case

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

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

$$\min\{c^k x^k : x^k \in Z^k\}$$

The "complicating" constraints only depend on the aggregate variables:

$$\boldsymbol{y = \sum_{k=1}^{K} x^k} \qquad \boldsymbol{Y = \{y \in \mathbb{Z}_+^n : Dy \geq d\}}.$$

- **Lagrangian sub-problem**:
  $L(\pi) = \min_x\{cx + \pi(d - Dx) : \underbrace{Bx \geq b, x \in \mathbb{Z}^n_+}_{x \in Z}\}.$

- **Lagrangian dual**:

$$
\begin{aligned}
z_{LD} &= \max_{\pi \geq 0} L(\pi) = \max_{\pi \geq 0} \min_{x \in Z}\{cx + \pi(d - Dx)\}. \\
&= \max_{\pi \geq 0} \min_{t=1,\ldots,T}\{cx^t + \pi(d - Dx^t)\} \\
&= \max \pi d + \sigma \\
&\quad \pi Dx^t + \sigma \leq cx^t \quad t = 1, \ldots, T \\
&\quad \pi \geq 0, \sigma \in I\!R^1. \\
&= \min \sum_{t=1}^T cx^t \lambda_t \\
&\quad \sum_{t=1}^T Dx^t \lambda_t \geq d \\
&\quad \sum_{t=1}^T \lambda_t = 1, \; \lambda_t \geq 0 \quad t = 1, \ldots, T.
\end{aligned}
$$

Lagrangian duality

$$z_{LD} = \min\{cx : Dx \geq d, x \in \text{conv}(Z)\}$$

Example: the Bin Packing Problem

$$
\begin{aligned}
\min \quad & \sum_{k=1}^{K} u_k \\
\sum_{k=1}^{K} x_{ik} &= 1 \qquad \forall i \\
\sum_{i} s_i \, x_{ik} &\leq u_k \qquad \forall k \\
x_{ik}, u_k &\in \{0,1\} \quad \forall i, k
\end{aligned}
$$

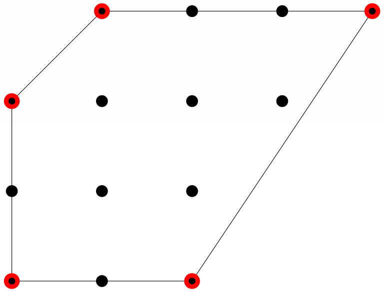
$$
\begin{aligned}
L(\pi) = \sum_i \pi_i + K \min\{(u - \sum_i \pi_i x_i) \\
\sum_i s_i x_i &\leq u \\
u \in \{0,1\},\ x_i &\in \{0,1\}\}
\end{aligned}
$$

Reformulation of $conv(Z) \rightarrow$ convexification

Every Polyhedron P, in particular $conv(Z)$, can be represented as

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

where $\{x^g\}_{g \in G}$ are the extreme points and $\{v^r\}_{r \in R}$ the extreme rays of P.



Reformulation of $conv(Z)$ → convexification

Every Polyhedron P, in particular $conv(Z)$, can be represented as

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

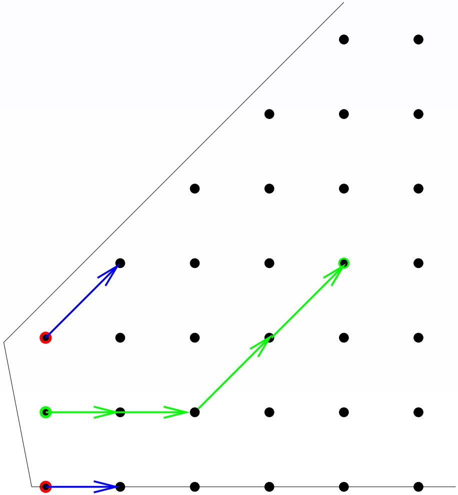
where $\{x^g\}_{g \in G}$ are the extreme points and $\{v^r\}_{r \in R}$ the extreme rays of P.

IP-Reformulation of Z → discretization

Every IP set $Z = \{x \in \mathbb{Z}^n : Bx \geq b\}$ can be represented in the form $Z = \text{proj}_x(Q)$, with

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

where $\{x^g\}_{g \in G}$ is a finite set of integer points in Z, and $\{v^r\}_{r \in R}$ are the extreme integer rays of $\mathrm{conv}(Z)$.



- **The convexification approach:**

$$(DWc)\quad \begin{array}{rcl} \min \sum_{g\in G^c}(cx^g)\lambda_g & & \\ \sum_{g\in G^c}(Dx^g)\lambda_g & \geq & d \\ \sum_{g\in G^c}\lambda_g & = & 1 \\ x=\sum_{g\in G^c}x^g\lambda_g & \in & \mathbb{Z}^n \\ \lambda_g & \geq & 0\ \forall g\in G^c \end{array}$$

- **The discretization approach:**

$$(DWd)\quad \begin{array}{rcl} \min \sum_{g\in G^d}(cx^g)\lambda_g & & \\ \sum_{g\in G^d}(Dx^g)\lambda_g & \geq & d \\ \sum_{g\in G^d}\lambda_g & = & 1 \\ \lambda_g & \in & \{0,1\}\ \forall g\in G^d \end{array}$$

- **The convexification approach:**

$$
(DWc) \quad
\begin{array}{rcl}
\min \sum_{g \in G^c} (cx^g)\lambda_g & & \\
\sum_{g \in G^c} (Dx^g)\lambda_g & \geq & d \\
\sum_{g \in G^c} \lambda_g & = & 1 \\
x = \sum_{g \in G^c} x^g \lambda_g & \in & \mathbb{Z}^n \\
\lambda_g & \geq & 0 \ \forall g \in G^c
\end{array}
$$

- **The discretization approach:**

$$
(DWd) \quad
\begin{array}{rcl}
\min \sum_{g \in G^d} (cx^g)\lambda_g & & \\
\sum_{g \in G^d} (Dx^g)\lambda_g & \geq & d \\
\sum_{g \in G^d} \lambda_g & = & 1 \\
\lambda_g & \in & \{0,1\} \ \forall g \in G^d
\end{array}
$$

Strength of the LP Bound

The linear program modeling LD is precisely the LP relaxation of DWc and equivalent to the LP relaxations of DWd. Hence $z_{LP}^{DWc} = z_{LP}^{DWd} = z_{LD}$.

The block diagonal case with Identical Subsystems

$$
(DWad) \quad
\begin{aligned}
\min \sum_{g \in G} (cx^g) \nu_g & & & \quad (1) \\
\sum_{g \in G} (Dx^g) \nu_g & \geq & d & \quad (2) \\
\sum_{g \in G} \nu_g & = & K & \quad (3) \\
\nu & \in & \mathbb{Z}_+^{|G|}, & \quad (4)
\end{aligned}
$$

The projection of reformulation solution ν into the original variable space will only provide the aggregate variables:

$$
y = \sum_{g \in G} x^g \nu_g \; . \quad (5)
$$

Example: the Bin Packing Problem

$$\begin{array}{rcll} \min & \sum_{k=1}^{K} u_k & & \\ \sum_{k=1}^{K} x_{ik} & = & 1 & \forall i \\ \sum_{i} s_i\, x_{ik} & \leq & u_k & \forall k \\ x_{ik}, u_k & \in & \{0,1\} & \forall i,k \end{array}$$

$$Z = \{x \in \mathbb{Z}_+^n : \sum_{i=1}^{n} s_i x_i \leq 1\} = \{x^g\}_{g \in G}$$

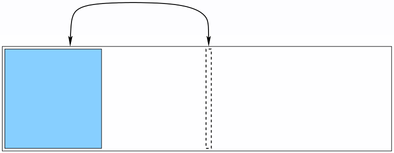
$$\min\{\sum_{g \in G} \nu_g : \sum_{g \in G} x^g \nu_g = 1, \nu \in Z_+^{|G|}\}$$

$$\begin{array}{rcl} \min \sum_{g\in G}(cx^g)\lambda_g & & \\ \sum_{g\in G}(D_i x^g)\lambda_g & \geq & d_i \ \forall i \\ \sum_{g\in G}\lambda_g & = & 1 \\ \lambda_g & \geq & 0 \ \ g\in G \end{array}$$

$$\begin{array}{rcl} \max \sum_i \pi_i d_i + \sigma & & \\ \sum_i \pi_i D_i x^g + \sigma & \leq & cx^g \quad \forall g\in G \\ \pi \geq 0, \sigma & \in & \mathbb{R}^1. \end{array}$$

$$\begin{array}{rcl} \min \sum_{g\in G}(cx^g)\lambda_g & & \\ \sum_{g\in G}(D_i x^g)\lambda_g & \geq & d_i \ \forall i \\ \sum_{g\in G}\lambda_g & = & 1 \\ \lambda_g & \geq & 0 \ \ g\in G \end{array}$$

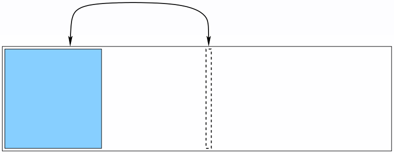
$$\begin{array}{rcl} \max \sum_i \pi_i d_i + \sigma & & \\ \sum_i \pi_i D_i x^g + \sigma & \leq & cx^g \quad \forall g \in G \\ \pi \geq 0, \sigma & \in & I\!R^1. \end{array}$$



Price Decomposition Column Generation

$$\begin{array}{rcl} \min \sum_{g \in G} (cx^g)\lambda_g & & \\ \sum_{g \in G} (D_i x^g)\lambda_g & \geq & d_i \ \forall i \\ \sum_{g \in G} \lambda_g & = & 1 \\ \lambda_g & \geq & 0 \ \ g \in G \end{array}$$

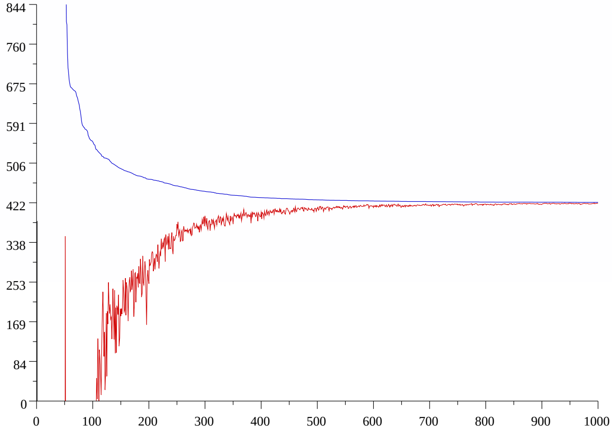
$$\begin{array}{rcll} \max \sum_i \pi_i d_i + \sigma & & & \\ \sum_i \pi_i D_i x^g + \sigma & \leq & cx^g & \forall g \in G \\ \pi \geq 0, \sigma & \in & I\!R^1. & \end{array}$$



- the reduced cost of x^g is $cx^g - \pi D x^g - \sigma$.
- $\zeta = \min_{g \in G}(cx^g - \pi D x^g) = \min_{x \in Z}(c - \pi D)x$. Thus, pricing consists in solving $OPT(Z)$.
- $z^{RMLP} = \sum_{g \in G'} (cx^g)\lambda_g = \pi d + \sigma \geq z_{MLP}$.
- (π, ζ) forms a feasible dual solution / $L(\pi)$ is available after pricing. Hence $L(\pi) = \pi d + \zeta \leq z_{MLP}$.
- If λ is integer, it defines a primal (upper) bound for problem IP.

$$\begin{aligned} \min \textstyle\sum_{t=1}^{T}(cx^t)\lambda_t & & \\ \textstyle\sum_{t=1}^{T}(D_i x^t)\lambda_t &\geq d_i \ \forall i \\ \textstyle\sum_{t=1}^{T}\lambda_t &= 1 \\ \lambda_t &\geq 0 \ \ t=1,\ldots,T. \end{aligned}$$

$$\begin{aligned} \max \textstyle\sum_i \pi_i d_i + \sigma & & \\ \textstyle\sum_i \pi_i D_i x^t + \sigma &\leq cx^t \quad \forall t \\ \pi \geq 0, \sigma &\in I\!R^1. \end{aligned}$$



i) Initialize $PB = +\infty$, $DB = -\infty$. Generate a subset of points x^g so that $RMLP$ is feasible.

ii) Iteration t,

ii.a) Solve $RMLP$; record the primal solution λ and the dual solution (π, σ).

ii.b) Check whether λ defines an integer solution of IP; if so update PB. If $PB = DB$, stop.

ii.c) Solve the pricing problem

$$\zeta = \min\{(c - \pi D)x : x \in Z\}.$$

Let x^* be an optimal solution. If $\zeta - \sigma = 0$, set $DB = z^{RMLP}$ and stop. Otherwise, add x^* to G and include the associated column in $RMLP$ (its reduced cost is $\zeta - \sigma < 0$).

ii.d) Compute the dual bound: $L(\pi) = \pi d + \zeta$; update $DB = \max\{DB, L(\pi)\}$. If $PB = DB$, stop.

Example: the Bin Packing Problem

Numerical example: $n = 5$, $s = (1, 2, 2, 3, 4)$, $S = 6$.

$$[M] \equiv \min \nu_1 + \nu_2 + \nu_3 + \nu_4 + \nu_5$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \\ \nu_4 \\ \nu_5 \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$Z^1 = 5$, $\nu = (1, 1, 1, 1, 1)$, $\pi = (1, 1, 1, 1, 1)$.

$$\begin{aligned} [SP] \equiv \max \ & x_1 + x_2 + x_3 + x_4 + x_5 \\ & x_1 + 2\,x_2 + 2\,x_3 + 3\,x_4 + 4\,x_5 && \leq && 6 \\ & x_i && \in && \{0, 1\} \end{aligned}$$

KNP(π) = 3, solution $x = (1, 1, 1, 0, 0)$

Example: the Bin Packing Problem

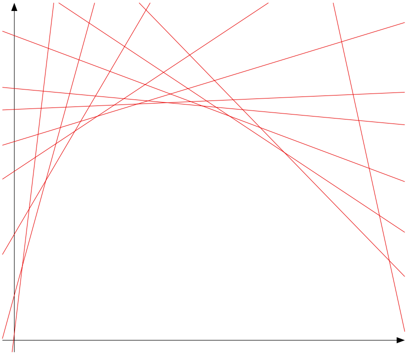
t	Z^t	master sol.	π^t	$L(\pi^t)$	PB	x^t
5	5	$\nu_1 = \nu_2 = \nu_3 = \nu_4 = \nu_5 = 1$	$(1,1,1,1,1)$	-5	5	$(1,1,1,0,0)$
6	3	$\nu_4 = \nu_5 = \nu_6 = 1,$	$(0,0,1,1,1)$	-2	3	$(0,0,1,1,0)$
7	3	$\nu_1 = \nu_4 = \nu_5 = 1$	$(0,1,0,1,1)$	-2	3	$(0,1,0,1,0)$
8	3	$\nu_1 = \nu_6 = \nu_7 = \nu_8 = \frac{1}{2}, \nu_5 = 1$	$(1,0,0,1,1)$	-2	3	$(1,0,0,0,1)$
9	2.5	$\nu_6 = \nu_7 = \nu_8 = \frac{1}{2}, \nu_9 = 1$	$(0,\frac{1}{2},\frac{1}{2},\frac{1}{2},1)$	0	3	$(0,1,0,0,1)$
10	2.33	$\nu_6 = \nu_8 = \nu_{10} = \frac{1}{3}, \nu_7 = \nu_9 = \frac{2}{3}$	$(\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{2}{3},\frac{2}{3})$	$\frac{2}{3}$	3	$(1,1,0,1,0)$
11	2.25	$\nu_6 = \nu_{11} = \frac{1}{4}, \nu_9 = \nu_{10} = \frac{1}{2}, \nu_7 = \frac{3}{4}$	$(\frac{1}{4},\frac{1}{4},\frac{1}{2},\frac{1}{2},\frac{3}{4})$	$\frac{4}{3}$	3	$(0,0,1,0,1)$
12	2	$\nu_{11} = \nu_{12} = 1$	$(0,0,0,1,1)$	2	2	$(0,0,0,0,1)$

$$\min \nu_1 + \nu_2 + \nu_3 + \nu_4 + \nu_5 + \nu_6 + \nu_7 + \nu_8 + \nu_9 + \nu_{10} + \nu_{11} + \nu_{12}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \\ \nu_4 \\ \nu_5 \\ \nu_6 \\ \nu_7 \\ \nu_8 \\ \nu_9 \\ \nu_{10} \\ \nu_{11} \\ \nu_{12} \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$Z^{12} = 2,\ \nu_{11} = 1,\ \nu_{12} = 1,\ u = (0,0,0,1,1).$

$$z_{LD} = \max_{\pi \geq 0} \min_{t=1,\ldots,T} \{cx^t + \pi(d - Dx^t)\}$$



The sub-gradient algorithm

i) Initialize $\pi^0 = 0$, $t = 1$.
ii) Iteration t,

ii.a) Solve the Lagrangian subproblem to obtain the dual bound $L(\pi^t) = \min\{cx + \pi^t(d - Dx)\}$ *and an optimal solution* x^t.

ii.b) Compute x^t*'s violation of the dualized constraints* $(d - Dx^t)$*; this provides a "sub-gradient" that can be used as a "potential direction of ascent" to modify the dual variables.*

ii.c) Update the dual solution using

$$\pi^{t+1} = \max\{0, \pi^t + \epsilon_t(d - Dx^t)\}$$

where ϵ_t *is a appropriately chosen step-size.*

iii) If $t < \tau$*, increment* t *and return to ii).*

The sub-gradient algorithm

- Step size: $\pi^{t+1} = \max\{0, \pi^t + \epsilon_t(d - Dx^t)\}$ with $\epsilon_t = \frac{\alpha_t}{||d-Dx^t||}$
  a) $\alpha_t = C(PB - L(\pi^t))$ with $C \in (0, 2)$;
  b) the α_t form a geometric series: $\alpha_t = C\rho^t$ with $\rho \in (0, 1)$ and $C > 0$;
  c) the α_t form a divergent series: f.i., $\alpha_t = \frac{1}{t}$.

- Candidate primal solution: $\hat{x}$ (no guarantee to satisfy constraints $Dx \geq d$):
  a) $\hat{x} = \sum_{g=1}^{t} x^g \lambda_g$ where $\lambda_g = \frac{\alpha_g}{\sum_{g=1}^{t} \alpha_g}$, or
  b) $\hat{x} = \eta\hat{x} + (1 - \eta)x^t$ with $\eta \in (0, 1)$.

The volume algorithm

- $\hat{x} = \eta\hat{x} + (1-\eta)x^t$ with a suitable $\eta \in (0,1)$;
- $\hat{\pi} = \text{argmax}_{g=1,\dots,t}\{L(\pi^g)\}$;
- the "direction of ascent" is defined by the violation of $\hat{x}$, i.e. $(d - D\hat{x})$, instead of x^t;
- the dual price updating rule is computed from $\hat{\pi}$, instead of π^t: $\pi^{t+1} = \max\{0, \hat{\pi} + \epsilon_t(d - D\hat{x})\}$.
- Stopping criteria: when $||d - D\hat{x}|| \leq \epsilon$ and $||c\hat{x} - \hat{\pi}d|| \leq \epsilon$.

Adapted conjugate gradient method, the method is equivalent to making a suitable correction v^t in the dual price updating direction $\pi^{t+1} = \max\{0, \pi^t + \epsilon_t(d - Dx^t) + v^t\}$.

The bundle method (stabilized col gen)

$$\max_{\pi \geq 0} \{ L(\pi) - \frac{||\pi - \hat{\pi}||^2}{\eta} \}$$

$\Updownarrow$ (in the case $Dx = d$)

$$\min_{x} \{ cx + \hat{\pi}(d - Dx) + \eta ||d - Dx||^2 : x \in conv(Z) \}$$

Thus, the dual restricted master is a quadratic program:

$$\begin{aligned} \max \textstyle\sum_i \pi_i d_i + \sigma - \frac{||\pi - \hat{\pi}||^2}{\eta} & & & \\ \textstyle\sum_i \pi_i D_i x^t + \sigma & \leq & cx^t & \forall t = 1, \ldots, T \\ \pi \geq 0, \sigma & \in & I\!R^1. & \end{aligned}$$

Accelerating column generation

- proper initialization (warm start): meaningful dual solutions π from the outset (using a dual heuristic or a rich initial set of points x^g, produced for instance by the sub-gradient method)
- pricing strategy: multiple column gen., intelligent sequence of pricing problems
- stabilization: penalizing deviations of the dual solutions from a *stability center* $\hat{\pi}$:

$$\max_{\pi \geq 0} \{L(\pi) + S(\pi - \hat{\pi})\}$$

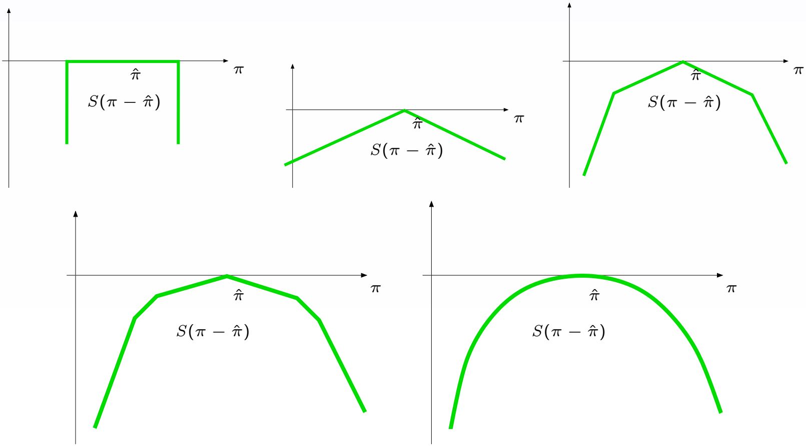
- smoothing:

$$\overline{\pi}^t = \alpha \overline{\pi}^{t-1} + (1-\alpha)\pi^t$$

$$\overline{\pi}^t = \alpha \hat{\pi} + (1-\alpha)\pi^t$$

- interior point approaches: ACCPM, convex combination of dual extreme points.

Stabilization functions



Single subsystem (or multiple non-identical subsystems)

$$\begin{array}{rcl} \min \sum_{g \in G} (cx^g)\lambda_g & & \\ \sum_{g \in G} (Dx^g)\lambda_g & \geq & d \\ \sum_{g \in G} \lambda_g & = & 1 \\ \lambda_g & \geq & 0 \ \forall g \in G \end{array}$$

- Integrality Test. If λ^* is integer, or more generally if $x^* = \sum_{g \in G} x^g \lambda_g^* \in \mathbb{Z}^n$, stop. x^* is an optimal solution of IP.
- Branching. Select a variable x_j for which $x_j^* = \sum_{g \in G} x_j^g \lambda_g^* \notin \mathbb{Z}$. Separate into $X \cap \{x : x_j \leq \lfloor x_j^* \rfloor\}$ and $X \cap \{x : x_j \geq \lceil x_j^* \rceil\}$.
  - Option 1: the branching constraint is dualized as a "difficult" constraint
  - Option 2: the branching constraint is enforced in the sub-problem

Single subsystem: consider the up-branch

- Option 1: The branching constraint goes in the master:

$$
\begin{aligned}
\min \sum_{g \in G} (cx^g)\lambda_g & & \\
\sum_{g \in G} (Dx^g)\lambda_g & \geq & d \\
\sum_{g \in G} x_j^g \lambda_g & \geq & \lceil x_j^* \rceil \\
\sum_{g \in G} \lambda_g & = & 1 \\
\lambda_g & \geq & 0 \;\; g \in G,
\end{aligned}
$$

- Option 2: The branching constraint goes in the subproblem:

$$\zeta_2 = \min\{(c - \pi D)x : x \in Z \cap \{x : x_j \geq \lceil x_j^* \rceil\}\}.$$

Single subsystem: comparing the 2 options

Strength of the linear programming bound

$$\begin{aligned} z^{MLP_1} &= \min\{cx : Dx \geq d, x \in \text{conv}(Z), x_j \geq \lceil x_j^* \rceil\} \\ \leq z^{MLP_2} &= \min\{cx : Dx \geq d, x \in \text{conv}(Z \cap \{x : x_j \geq \lceil x_j^* \rceil\}) \end{aligned}$$

Complexity of the subproblem For option 1 the subproblem is unchanged, whereas for option 2 the subproblem may become more complicated.

Getting Integer Solutions If an optimal solution x^* of IP is not an extreme point of $\text{conv}(Z)$, it cannot be obtained as a solution of the subproblem under Option 1. Under Option 2, one can eventually generate a column $x^g = x^*$ in the interior of $conv(Z)$.

Multiple identical subsystems ($\nu_g = \sum_k \lambda_g^k$): the set partitioning case

$$
\begin{array}{rcl}
\min \sum_{g \in G} (cx^g) \nu_g & & \\
\sum_{g \in G} x^g \nu_g & = & 1 \\
\sum_{g \in G} \nu_g & = & K \\
\nu_g & \geq & 0 \ \forall g \in G
\end{array}
$$

Multiple identical subsystems ($\nu_g = \sum_k \lambda_g^k$): the set partitioning case

$$\begin{array}{rcl} \min \sum_{g \in G} (cx^g)\nu_g & & \\ \sum_{g \in G} x^g \nu_g & = & 1 \\ \sum_{g \in G} \nu_g & = & K \\ \nu_g & \geq & 0 \; \forall g \in G \end{array}$$

- Integrality Test.

  If ν is integer, stop. Else, sort columns and disaggregate ν:

  $\lambda_g^k = \min\{1, \nu_g - \sum_{\kappa=1}^{k-1} \lambda_g^\kappa, (k - \sum_{\gamma \prec g} \nu_g)^+\} \quad \forall k = 1, \ldots, K, g \in G$

  Let $x^k = \sum_{g \in G} x^g \lambda_g^k \; \forall k$. If $x \in \mathbb{Z}^n$,stop.

Multiple identical subsystems ($\nu_g = \sum_k \lambda_g^k$): the set partitioning case

$$\begin{aligned} \min \textstyle\sum_{g\in G}(cx^g)\nu_g & \\ \textstyle\sum_{g\in G} x^g \nu_g &= 1 \\ \textstyle\sum_{g\in G} \nu_g &= K \\ \nu_g &\geq 0 \ \forall g \in G \end{aligned}$$

- Integrality Test.
  If ν is integer, stop. Else, sort columns and disaggregate ν:
  $\lambda_g^k = \min\{1, \nu_g - \sum_{\kappa=1}^{k-1} \lambda_g^\kappa, (k - \sum_{\gamma \prec g} \nu_g)^+\} \quad \forall k = 1, \ldots, K, g \in G$
  Let $x^k = \sum_{g\in G} x^g \lambda_g^k \ \forall k$. If $x \in \mathbb{Z}^n$,stop.

ν_g	1	0	1	$\frac{1}{2}$	0	0	$\frac{1}{2}$	1	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
x_{i_1}	1	1	1	**1**	1	1	**1**	0	0	0	0	0	0	0	0
x_{i_2}	1	1	1	**0**	0	0	**0**	1	1	1	1	0	0	0	0
x_{i_3}	1	0	0	**1**	1	0	**0**	1	1	0	0	1	1	0	0
x_{i_4}	0	1	0	**1**	0	1	**0**	1	0	1	0	1	0	1	0
	x^{k_1}	x^{k_2}		x^{k_3}				x^{k_4}	x^{k_5}						

$$\Rightarrow \boxed{\boldsymbol{x_{i_3}^{k_3} = \frac{1}{2}}}$$

Multiple identical subsystems ($\nu_g = \sum_k \lambda_g^k$): the set partitioning case

$$\begin{array}{rcl} \min \sum_{g\in G}(cx^g)\nu_g & & \\ \sum_{g\in G} x^g \nu_g & = & 1 \\ \sum_{g\in G} \nu_g & = & K \\ \nu_g & \geq & 0 \ \forall g \in G \end{array}$$

- Integrality Test.
  If ν is integer, stop. Else, sort columns and disaggregate ν:
  $\lambda_g^k = \min\{1, \nu_g - \sum_{\kappa=1}^{k-1} \lambda_g^\kappa, (k - \sum_{\gamma \prec g} \nu_g)^+\} \quad \forall k = 1, \ldots, K, g \in G$
  Let $x^k = \sum_{g\in G} x^g \lambda_g^k \ \forall k$. If $x \in \mathbb{Z}^n$,stop.
- Branching. Option 1 and 2 are not useful, as $y_j = \sum_{g\in G} x_j^g \nu_g = 1$ always. Instead, consider two components i and j. Branch on an **auxiliary variable** w_{ij}, using

$$\left(w_{ij} = \sum_{g: x_i^g=1, x_j^g=1} \nu_g = 0\right) \quad \text{or} \quad \left(w_{ij} = \sum_{g: x_i^g=1, x_j^g=1} \nu_g = 1\right)$$

The set partitioning case: branching implementation options

Up-branch $(w_{ij} = \sum_{g: x_i^g=1, x_j^g=1} \nu_g = 1)$

- Option 3: the branching constraint goes in the master

$$\sum_{g: x_i^g=1, x_j^g=1} \nu_g \geq 1$$

$$\zeta_3 = \min\{(c - \pi D)x - \mu w_{ij} : x \in Z, w_{ij} \leq x_i, w_{ij} \leq x_j, w_{ij} \in \{0,1\}\}.$$

- Option 4: it is implicitly enforced in the sub-problem

$$\zeta_4 = \min\{(c - \pi D)x : x \in Z, x_i = x_j\}.$$

- Option 5: differentiate 2 pricing problems, and enforce BC explicitly in SP

$$\zeta_{5A} = \min\{(c - \pi D)x : x \in Z, x_i = x_j = 0\}$$

$$\zeta_{5B} = \min\{(c - \pi D)x : x \in Z, x_i = x_j = 1\}.$$

$$\sum_{g \in G_{5A}} \nu_g = K - 1 \text{ and } \sum_{g \in G_{5B}} \nu_g = 1$$

Strength of the LP bound, Complexity of the SP, Integer Points

Multiple identical subsystems: the general case

$$\min\{\sum_{g\in G}(cx^g)\nu_g : \sum_{g\in G}(Dx^g)\nu_g \geq d, \sum_{g\in G}\nu_g = K, \nu\in\mathbb{R}_+^{|G|}\}.$$

- Option 1: branch on the aggregate variables
  $y_i = \sum_{g\in G} x_i^g \nu_g = \alpha \notin \mathbb{Z}$
  $$\sum_{g\in G} \boldsymbol{x_i^g \nu_g \leq \lfloor \alpha \rfloor} \ \textbf{or} \ \sum_{g\in G} \boldsymbol{x_i^g \nu_g \geq \lceil \alpha \rceil}.$$

- Option 3 & 4: branch on **auxiliary variables** (implicit reformulation) in the master or the SP.
  - VRP: edge use
  - CSP: arc use of arc flow formulation of SP
- Option 2 & 5: Branch on one (or several) components of x and differentiate subproblems: if $\sum_{g:x_j^g\geq l_j}\nu_g = \alpha \notin \mathbb{Z}$,
  $$\sum_{g:x_j^g\geq l_j}\boldsymbol{\nu_g \geq \lceil \alpha \rceil} \ \textbf{or} \ \sum_{g:x_j^g\leq l_j-1}\boldsymbol{\nu_g \geq K - \lfloor \alpha \rfloor}$$
  Pricing is carried out independently over the two sets $\hat{Z} = Z \cap \{x_j \geq l_j\}$ and $Z \setminus \hat{Z}$ on both branches.

Practical implementation issues

- Initialisation: initial columns (heuristic, sub-gradient, volume), artificial columns (dual heuristic)

$$
\begin{array}{rcl}
\min \sum_{t=1}^{T} (cx^t)\lambda_t + \sum_i \overline{\pi}_i \rho_i & & \\
\sum_{t=1}^{T} (D_i x^t)\lambda_t + \rho_i & \geq & d_i \ \forall i \\
\sum_{t=1}^{T} \lambda_t & = & 1 \\
\lambda_t \geq 0 \ \ t = 1, \dots, T. \ \ \rho_i & \geq & 0 \ \forall i
\end{array}
\qquad
\begin{array}{rcl}
\max \sum_i \pi_i d_i + \sigma & & \\
\sum_i \pi_i D_i x^t + \sigma & \leq & cx^t \ \forall t \\
\pi_i & \leq & \overline{\pi}_i \ \forall i \\
\pi \geq 0, \sigma & \in & I\!R^1.
\end{array}
$$

- Preprocessing: “proper columns” (that account for master constraints)
- Stabilization
- Primal heuristics: Restricted master solved as an IP, rounding, local search.
- Cut generation: based on master constraints.
- Branching strategies: branch on constraints.

$$\begin{aligned} \min cy \\ y - z &= 0 \\ y &\in Y \\ z &\in Z. \end{aligned}$$

$$\min \sum_{i \in I} cy^i \lambda_i$$

$$\sum_{i \in I} y^i \lambda_i = \sum_{j \in J} z^j \beta_j$$

$$\sum_{i \in I} \lambda_i = 1 \qquad \sum_{j \in J} \beta_j = 1$$

$$\lambda \in \mathbb{R}_+^{|I|}, \qquad \beta \in \mathbb{R}_+^{|J|}$$

$$\zeta^1 = \min\{\pi x - \pi_0,\ x \in Y\}$$

$$\zeta^2 = \min\{-\pi x - \mu_0, x \in Z\}.$$

The Vehicle Routing Problem

$$\min \sum_{e \in E} c_e \; x_e$$

$$\begin{aligned}
\sum_{e \in \delta(i)} x_e &= 2 \quad \forall i \in V \setminus \{0, n+1\} \\
\sum_{e \in \delta(i)} x_e &= K \quad \forall i \in \{0, n+1\} \\
\sum_{e \in \delta(S)} x_e &\geq 2\, B(S) \quad \forall S \subseteq V \setminus \{0, n+1\} \\
x_e &\in \{0, 1\} \quad \forall e \in E,
\end{aligned}$$

$$Z = \{\text{q-routes}\}$$

$$\min\{\textstyle\sum_{g \in G}(\sum_e c_e x_e^g)\lambda_g : \sum_{g \in G}(\sum_{e \in \delta(i)} x_e^g)\lambda_g = 2 \; \forall i \in V \setminus \{0, n+1\}, \\ \textstyle\sum_{g \in G} \lambda_g \leq K, \; \lambda_g \in \{0, 1\} \; \forall g\}$$

CUTS: $\boxed{\sum_{g \in G, e \in \delta(S)} x_e^g \lambda_g \geq 2\, B(S) \geq 2 \lceil (\sum_{i \in S} d_i)/C \rceil}$