TONGJI UNIVERSITY

Introduction to Column Generation

Zhenzhen ZHANG

School of Economics and Management, Tongji University

2024

Outline

01 Basic Theories

02 Column Generation

03 Branch-and-Price

04 Applications

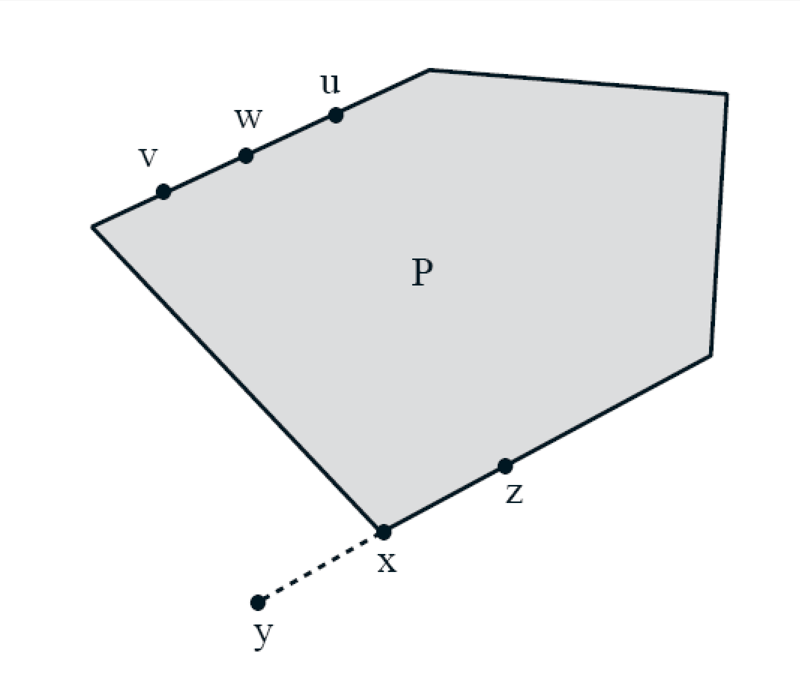
Useful Concepts

- **Extreme point:**
  - Let $P \subseteq \mathbb{R}^n$ be a non-empty, closed convex set. Then $\boldsymbol{x}$ is an extreme point of P if there are no two points $\boldsymbol{y}, \boldsymbol{z} \in P$ and $\lambda \in (0, 1)$, $s.t.\, \boldsymbol{x} = \lambda \boldsymbol{y} + (1-\lambda)\boldsymbol{z}$.

- Polyhedreon $P = \{\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}\}$

- $x \in P$ is an **extreme point** of P if

$$\nexists \boldsymbol{y}, \boldsymbol{z} \in P \;:\; \boldsymbol{x} = \lambda \boldsymbol{y} + (1-\lambda)\boldsymbol{z}, \lambda \in (0,1)$$



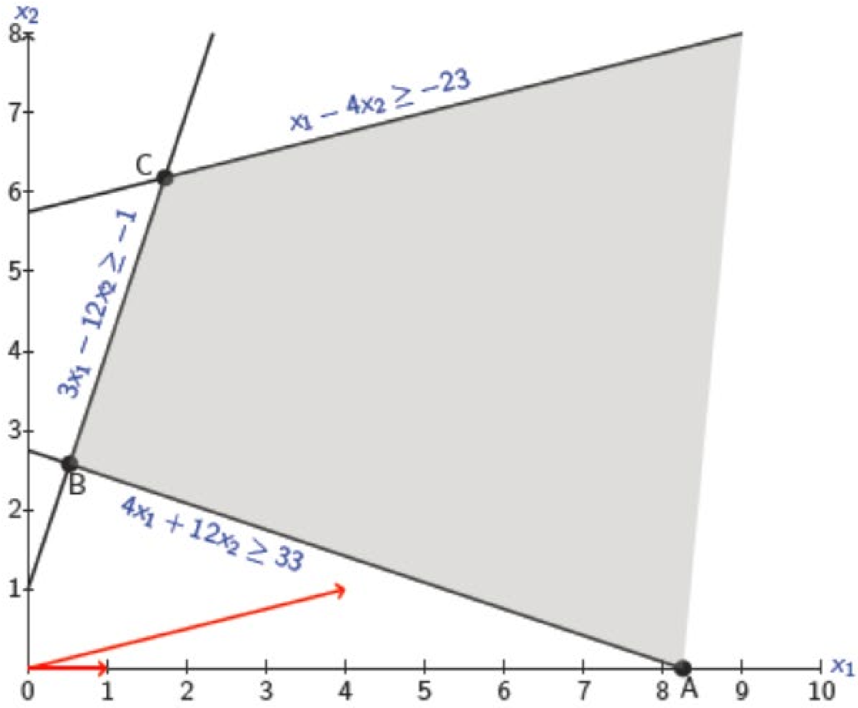
Useful Concepts

- Given a convex set, a nonzero vector $\boldsymbol{d}$ is called **direction** of the set, if for each $\boldsymbol{x}_0$ in the set, the point $\{\boldsymbol{x}_0 + \lambda \boldsymbol{d}: \lambda \geq 0\}$ also belongs to the set.
  - Hence, starting at any point $\boldsymbol{x}_0$ in the set, one can recede along $\boldsymbol{d}$ for any step length $\lambda \geq 0$ and remain within the set.
  - For polyhedron $P = \{\boldsymbol{x} \in R^n | \boldsymbol{Ax} \leq \boldsymbol{b}\}$, let $P^0 = \{\boldsymbol{d} \in R^n | \boldsymbol{Ad} \leq \boldsymbol{0}\}$. Any $\boldsymbol{d} \in P^0 \setminus \{\boldsymbol{0}\}$ is a direction of P.
  - Clearly, if the set is bounded, then it has no directions.

- An **extreme direction** of a convex set is a direction of the set that cannot be represented as a positive combination of *two distinct* directions of the set.
  - A direction $\boldsymbol{d}$ of polyhedron $P = \{\boldsymbol{x} \in R^n | \boldsymbol{Ax} \leq \boldsymbol{b}\}$ is called an extreme direction if there are $n - 1$ linearly independent constraints that are active at $\boldsymbol{d}$.

Extreme Points and Directions:

- $P = \{x \in \mathbb{R}^2_+ : 4x_1 + 12x_2 \geq 33, 3x_1 - x_2 \geq -1, x_1 - 4x_2 \geq -23\}$



- $x^A = (33/4, 0)^T$
- $x^B = (21/40, 103/40)^T$
- $x^C = (19/11, 68/11)^T$
- $d^A = (1, 0)^T$
- $d^B = (4, 1)^T$

Resolution Theorem by Minkowski (Convexification)

- Let $X = \{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}\}$ be a nonempty (polyhedral) set:
  - The set of **extreme points** is nonempty and has a finite number of elements, i.e., $\boldsymbol{x}_1, \dots \boldsymbol{x}_k$
  - The set of **extreme directions** is empty if and only if X is bounded.
  - If X is not bounded, then the set of extreme directions is nonempty and has a finite number of elements, i.e., $\boldsymbol{d}_1, \dots \boldsymbol{d}_\ell$.

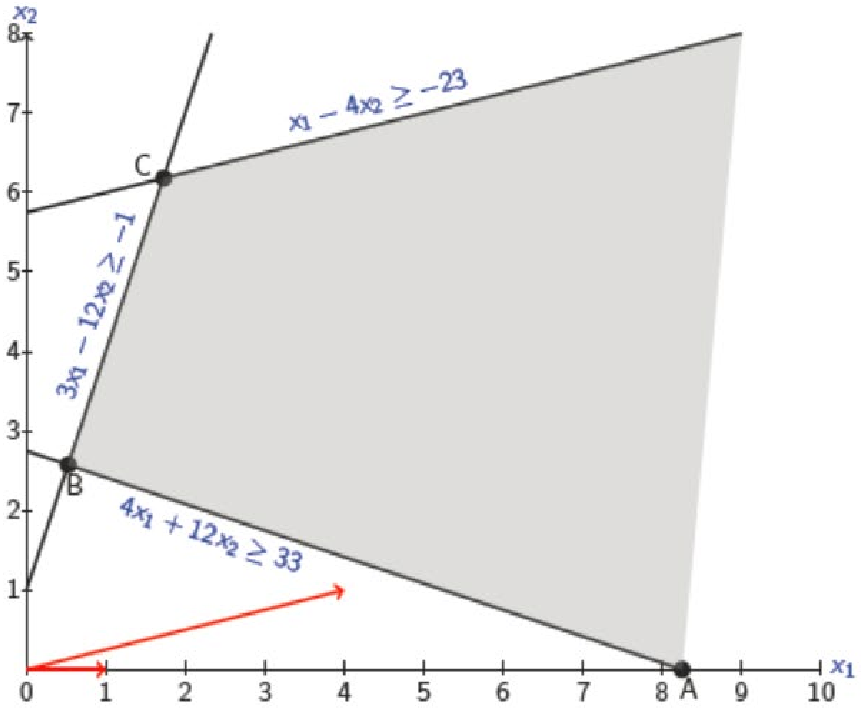
- **Resolution Theorem:**
  - $\boldsymbol{x} \in X$ if and only if it can be represented as a *convex combination* of $\boldsymbol{x}_1, \dots \boldsymbol{x}_k$ plus a *nonnegative linear combination* of $\boldsymbol{d}_1, \dots \boldsymbol{d}_\ell$:

$$\boldsymbol{x} = \sum_j^k \lambda_j \boldsymbol{x}_j + \sum_j^\ell \mu_j \boldsymbol{d}_j, \sum_j^k \lambda_j = 1, \boldsymbol{\lambda} \in \mathbb{R}_+^k, \boldsymbol{\mu} \in \mathbb{R}_+^\ell.$$

Resolution Theorem

- $P = \{x \in \mathbb{R}^2_+ : 4x_1 + 12x_2 \geq 33, 3x_1 - x_2 \geq -1, x_1 - 4x_2 \geq -23\}$

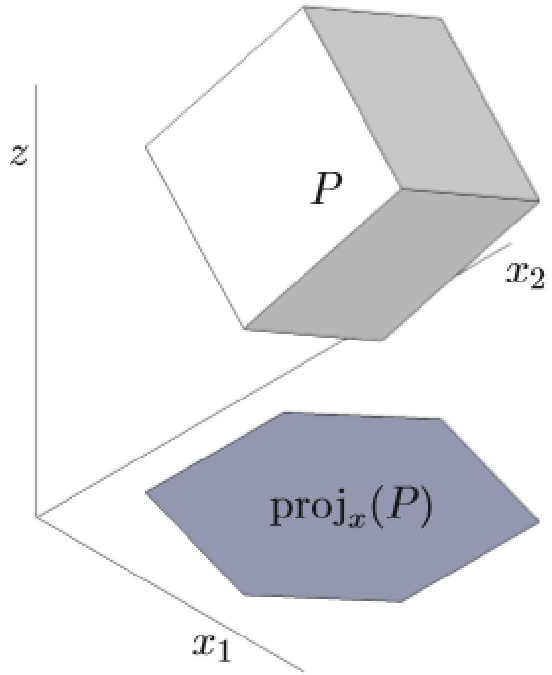
$$Q = \{(x, \lambda, \mu) \in \mathbb{R}^2 \times \mathbb{R}^3_+ \times \mathbb{R}^2_+ : \\ x = x^A\lambda_1 + x^B\lambda_2 + x^C\lambda_3 + \\ \begin{pmatrix}1\\0\end{pmatrix}\mu_1 + \begin{pmatrix}4\\1\end{pmatrix}\mu_2, \\ \lambda_1 + \lambda_2 + \lambda_3 = 1\}$$



- $x^A = (33/4, 0)^T$
- $x^B = (21/40, 103/40)^T$
- $x^C = (19/11, 68/11)^T$

Projection

$$\text{proj}_x(S) := \{x \in \mathbb{R}^n \ : \ \exists z \in \mathbb{R}^p \text{ s.t. } (x, z) \in S\}.$$



Resolution Theorem (Discretization)

- Every IP set $X = \{x \in \mathbb{Z}^n: Ax \geq b\}$ can be represented in the form $X = proj_x(Q_I)$, where:

$$Q_I = \{(x, \lambda, \mu) \in \mathbb{R}^n \times \mathbb{Z}_+^k \times \mathbb{Z}_+^\ell : x = \sum_j^k \lambda_j x_j + \sum_j^\ell \mu_j d_j, \sum_j^k \lambda_j = 1\}.$$

  - $\{x_1, \dots x_k\}$ is the finite set of **integer points** in X
  - $\{d_1, \dots d_\ell\}$ are the extreme directions (scaled to be integer) of $conv(X)$.

- Remark 1: if X is **bounded**, the set $\{x_1, \dots x_k\}$ contains **all integer points** in X, and $\{d_1, \dots d_\ell\}$ is empty.

- Remark 2: if X is **unbounded**, the set $\{x_1, \dots x_k\}$ contains some integer points which are not extreme points of $conv(X)$.

Discretization Representation

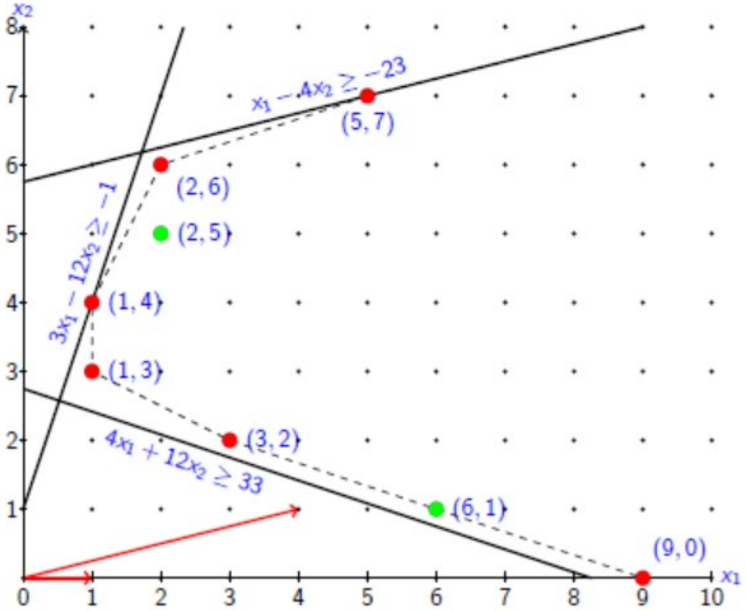
- The set of integer points $X = P \cap \mathbb{Z}^2$ where

$$P = \{x \in \mathbb{R}^2_+ : 4x_1 + 12x_2 \geq 33, 3x_1 - x_2 \geq -1, x_1 - 4x_2 \geq -23\}$$

$$Q = \{(x, \lambda, \mu) \in \mathbb{R}^2 \times \mathbb{Z}^6_+ \times \mathbb{Z}^2_+ :$$

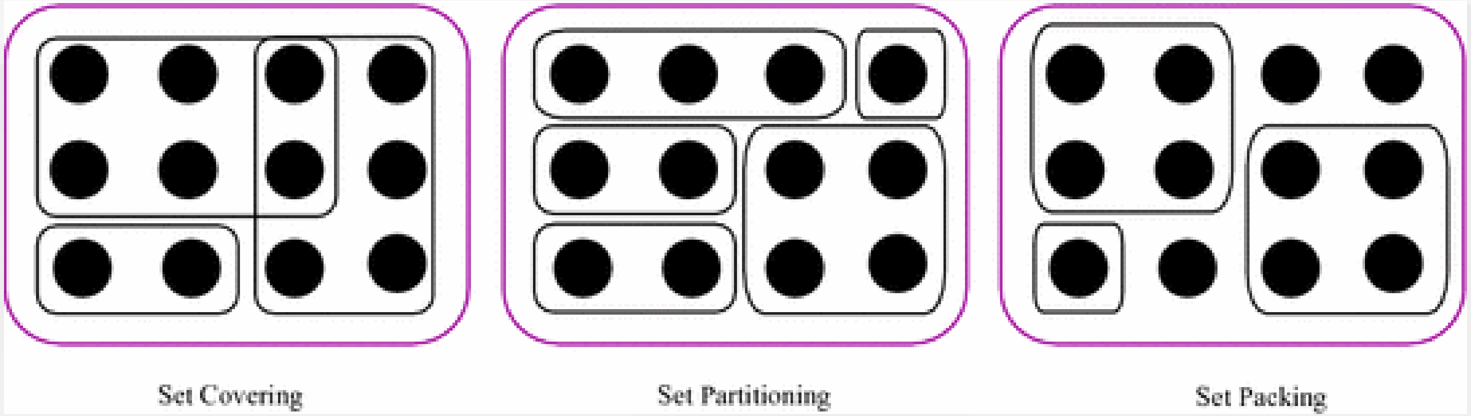
$$x = \begin{pmatrix}9\\0\end{pmatrix}\lambda_1 + \begin{pmatrix}3\\2\end{pmatrix}\lambda_2 + \begin{pmatrix}1\\3\end{pmatrix}\lambda_3 + \begin{pmatrix}1\\4\end{pmatrix}\lambda_4 + \begin{pmatrix}2\\6\end{pmatrix}\lambda_5 + \begin{pmatrix}5\\7\end{pmatrix}\lambda_6 + \begin{pmatrix}2\\5\end{pmatrix}\lambda_7 + \begin{pmatrix}6\\1\end{pmatrix}\lambda_8 + \begin{pmatrix}1\\0\end{pmatrix}\mu_1 + \begin{pmatrix}4\\1\end{pmatrix}\mu_2 +$$

$$\sum_{p=1}^{8} \lambda_p = 1\}$$



Packing, Covering, Partitioning

- Let N be a finite set and $\mathcal{M} = \{N_j : j \in M\}$ a class of nonempty subsets of N (M is the index set of these subsets)
- $S \subseteq M$ is a **cover** of N if $\cup_{j \in S} N_j = N$
- $S \subseteq M$ is a **packing** of N if the subsets $N_j, j \in S$, are pairwise disjoint
- $S \subseteq M$ is a **partition** of N if S is both a cover and a packing



Packing, Covering, Partitioning

- Let N be a finite set and $\mathcal{M} = \{N_j : j \in M\}$ a class of nonempty subsets of N (M is the index set of these subsets)
- $S \subseteq M$ is a **cover** of N if $\cup_{j\in S} N_j = N$
- $S \subseteq M$ is a **packing** of N if the subsets $N_j, j \in S$, are pairwise disjoint
- $S \subseteq M$ is a **partition** of N if S is both a cover and a packing

Set Covering Problem (SCP)	Set Packing Problem (SPK)	Set Partitioning Problem (SPP)
$\min \sum_{j\in M} c_j x_j$ s.t. $Ax \geq e$	$\max \sum_{j\in M} c_j x_j$ s.t. $Ax \leq e$	$\min / \max \sum_{j\in M} c_j x_j$ s.t. $Ax = e$

Dantzig-Wolfe Reformulations

- Originally developed by George Dantzig and Philip Wolfe and initially published in 1960 for solving linear programming problems with special structure.
- Consider two closely related extended formulations for the problem:

$$\min\{cx\colon Dx \geq d, x \in Z\}.$$

- Assume $\mathrm{Z} = \{x \in \mathbb{Z}_+^n\colon Bx \geq b\}$ is ***bounded***, and $\{x_g\}_{g \in G^c}$ are the extreme points of $conv(Z)$.
- The Dantzig-Wolfe reformulation based on the **convexification** is:

$$
\begin{aligned}
(\mathrm{DW_c}) \quad z^c = \min\ & \textstyle\sum_{g \in G^c} (cx_g)\lambda_g \\
\text{s.t.}\ & \textstyle\sum_{g \in G^c} (Dx_g)\lambda_g \geq d \\
& \textstyle\sum_{g \in G^c} \lambda_g = 1 \\
& x = \textstyle\sum_{g \in G^c} x_g \lambda_g \in \mathbb{Z}^n \\
& \lambda \in \mathbb{R}_+^{|G^c|}.
\end{aligned}
$$

Dantzig-Wolfe Reformulations

- Originally developed by George Dantzig and Philip Wolfe and initially published in 1960 for solving linear programming problems with special structure.

- Consider two closely related extended formulations for the problem:

$$\min\{cx: Dx \geq d, x \in Z\}.$$

- Assume $Z = \{x \in \mathbb{Z}_+^n: Bx \geq b\}$ is ***bounded***, and $\{x_g\}_{g \in G^d}$ are the points of Z.
- The Dantzig-Wolfe reformulation based on the **discretization** is:

$$
\begin{aligned}
(\mathrm{DW_d}) \quad z^d = \min\ & \textstyle\sum_{g \in G^d} (cx_g)\lambda_g \\
\text{s.t.}\ & \textstyle\sum_{g \in G^d} (Dx_g)\lambda_g \geq d \\
& \textstyle\sum_{g \in G^d} \lambda_g = 1 \\
& \lambda \in \mathbb{B}^{|G^d|}.
\end{aligned}
$$

Dantzig-Wolfe Reformulations

- There is no difference between DW_c and DW_d when $Z \in \mathbb{B}^n$ as every point $x \in Z$ is an extreme point of $conv(Z)$:

$$x = \sum_{g \in G^c} x_g \lambda_g \in \mathbb{B}^n \quad in\ DW_c \quad \Leftrightarrow \quad \lambda \in \mathbb{B}^{|G^d|}\ in\ DW_d$$

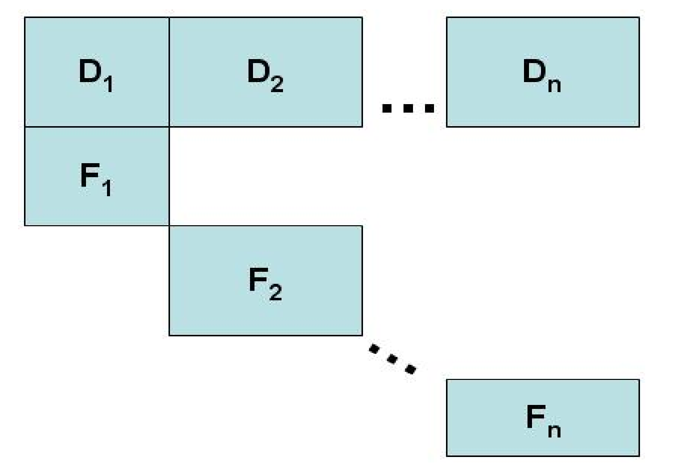
- Many large-scale applications belong to this class
  - especially many decomposition procedures that give rise to
    - set-partitioning problems
    - set-covering problems

Block Diagonal Structure

- $Bx \geq b$ may usually have **block diagonal** (BD) structure:
  - $Z = \{Z_1 \times Z_2 \times \cdots \times Z_K\}$, and
  - $(IP_{BD}) = \min\{\sum_{k=1}^{K} c_k x_k : (x_1, \dots, x_K) \in Y, x_k \in Z_k \ \forall k = 1, \dots, K\}$

- IP_{BD} can be explicitly written as:

$$
\begin{array}{llcccccl}
IP_{BD} & \min & c_1x_1 & + c_2x_2 & + \cdots & + c_Kx_K & & \\
 & \text{s.t.} & D_1x_1 & + D_2x_2 & + \cdots & + D_Kx_K & \geq & d \\
 & & B_1x_1 & & & & \geq & b_1 \\
 & & & B_2x_2 & & & \geq & b_2 \\
 & & & & \ddots & & \geq & \vdots \\
 & & & & & B_Kx_K & \geq & b_K \\
 & & x_1 \in \mathbb{Z}_+^n, \dots & & & x_K \in \mathbb{Z}_+^n & &
\end{array}
$$



Block Diagonal Structure

- Relaxing the constraints $\sum_{k=1}^{K} D_k x_k \geq d$, the problem decomposes into K *smaller* problems
  - Assume $\{x_g\}_{g\in G_k^d}$ for Z_k, and $x_k = \sum_{g\in G_k^d} x_g \lambda_{kg} \in Z_k$ for all k
  - The multi-block Dantzig-Wolfe reformulation resulting from **discretization** approach is:

$$\min \sum_{k=1}^{K} \sum_{g\in G_k^d} (c_k x_g) \lambda_{kg}$$

$$\text{s.t.} \sum_{k=1}^{K} \sum_{g\in G_k^d} (D_k x_g)\, \lambda_{kg} \geq d$$

$$\sum_{g\in G_k^d} \lambda_{kg} = 1, \quad \forall k = 1, \dots, K$$

$$\lambda \in \mathbb{B}^{\sum_k^K |G_k^d|}$$

- How about identical subproblem?

Example

- The cutting stock problem (CSP)
- An unlimited number of rolls of length L are available
- Given $d \in \mathbb{Z}_+^n$ and $s \in \mathbb{R}_+^n$, the problem is to obtain d_i pieces of length s_i for $i = 1, \ldots, n$ by cutting up the smallest possible number of rolls

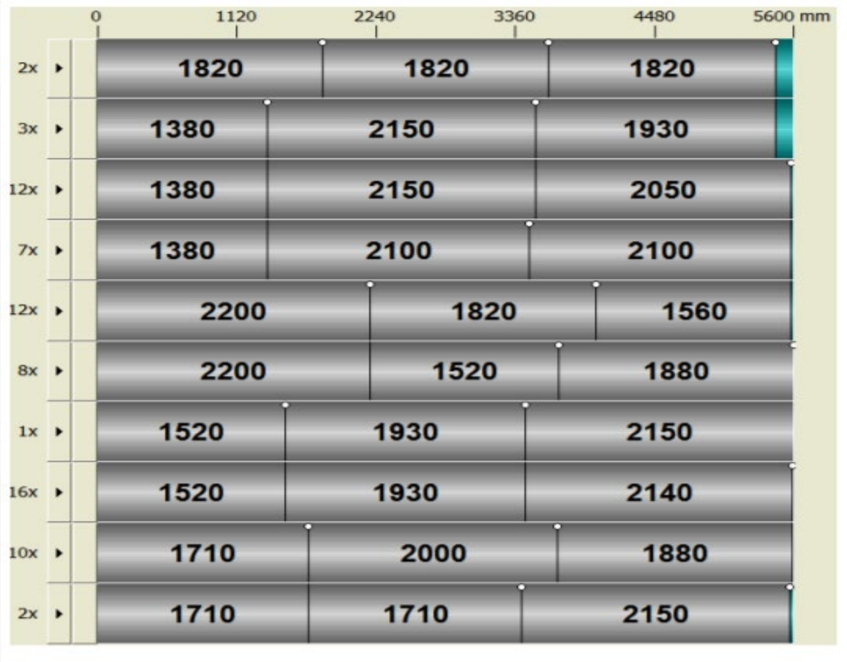
Example

- The cutting stock problem (CSP)

The width of large rolls: 5600mm. The width and demand of customers:

Width	1380	1520	1560	1710	1820	1880	1930	2000	2050	2100	2140	2150	2200
Demand	22	25	12	14	18	18	20	10	12	14	16	18	20

- An optimal solution:



Example

- The cutting stock problem (CSP)
- Notations:
  - K: index set of available rolls
  - y_k: =1 if roll k is cut, 0 otherwise
  - x_{ki}: number of times that item i is cut on roll k

- The IP formulation:

$$
\begin{aligned}
(P_1)\quad & \min \textstyle\sum_{k\in K} y_k \\
& s.t. \textstyle\sum_{k\in K} x_{ki} \geq d_i, \ \forall i = 1 \dots n \\
& \textstyle\sum_{i=1}^{n} s_i x_{ki} \leq L y_k, \ \forall k \in K \\
& x_{ki} \in \mathbb{Z}_+, y_k \in \{0, 1\}.
\end{aligned}
$$

Example

- Identical subproblems: $Z^* = \{x \in \mathbb{Z}_+^n : \sum_{i=1}^n s_i x_i \leq L\}$
- Each point x_g of Z^* corresponds to a cutting pattern
- Notations:
  - J: index set of all feasible patterns
  - a_{ij}: the number of times that item i is cut in pattern $j \in J$
  - y_j: number of rolls cut with pattern j

- The set-covering formulation:

$$
\begin{aligned}
(P_2) \quad & \min \sum_{j \in J} y_j \\
& s.t. \sum_{j \in J} a_{ij} y_j \geq d_i, \ \forall i = 1 \ldots n \\
& \quad y_j \in \mathbb{Z}_+, \forall j \in J.
\end{aligned}
$$

Example

- The cutting stock problem (CSP)
- **Comparisons:**
  - Number of variables?
  - Number of constraints?
  - Which formulation has a tighter LP relaxation?

$$
\begin{aligned}
(P_1) \quad & \min \textstyle\sum_{k\in K} y_k \\
& s.t. \textstyle\sum_{k\in K} x_{ki} \geq d_i, \ \forall i = 1 \dots n \\
& \quad \textstyle\sum_{i=1}^{n} s_i x_{ki} \leq L y_k, \ \forall k \in K \\
& \quad x_{ki} \in \mathbb{Z}_+, y_k \in \{0, 1\}.
\end{aligned}
$$

$$
\begin{aligned}
(P_2) \quad & \min \textstyle\sum_{j\in J} y_j \\
& s.t. \textstyle\sum_{j\in J} a_{ij} y_j \geq d_i, \ \forall i = 1 \dots n \\
& \quad y_j \in \mathbb{Z}_+, \forall j \in J.
\end{aligned}
$$

Example

- **Capacitated vehicle routing problem (CVRP)**
  - R: index set of all feasible routes
  - a_{ir}: binary coefficient, =1 if $i \in V$ belongs to route $r \in R$
  - c_r: the total travel cost associated with the route $r \in R$
  - θ_r: binary variable, =1 if route $r \in R$ is chosen in the solution
- **Set-partitioning model:**

$$
\begin{aligned}
\min\ & \textstyle\sum_{r\in R} c_r\theta_r \\
\text{s.t.}\ & \textstyle\sum_{r\in R} a_{ir}\theta_r = 1, \forall i \in V_c && \text{(customer)} \\
& \textstyle\sum_{r\in R} \theta_r \le m && \text{(vehicle number)} \\
& \theta_r \in \{0,1\}, \forall r \in R.
\end{aligned}
$$

- Issues?

Outline

01 Basic Theories

02 Column Generation

03 Branch-and-Price

04 Applications

Solving by Column Generation

- The LP relaxation of DW_c is traditionally called the (Dantzig-Wolfe) master problem (MLP)
  - Compute the optimal value z_{MLP} of MLP?
  - An exponential number of variables

- **Column Generation:**
  - Consider only a subset of points $\{x_g\}_{g \in \bar{G}}$ with $\bar{G} \subset G$
  - Dynamically introduce other necessary points
  - Reoptimize via *revised simplex* method

Solving by Column Generation

- Restricted master problem (RMLP) with $\bar{G}$:

$$
\begin{aligned}
(RMLP)\quad z_{RMLP} = \min\ & \textstyle\sum_{g\in\bar{G}}(cx_g)\lambda_g \\
& \textstyle\sum_{g\in\bar{G}}(Dx_g)\lambda_g \ge d && (\pi) \\
& \textstyle\sum_{g\in\bar{G}}\lambda_g = 1 && (\sigma) \\
& \lambda \in \mathbb{R}_+^{|\bar{G}|}
\end{aligned}
$$

- The dual of $RMLP$:

$$
\begin{aligned}
\max\ & \pi d + \sigma \\
\text{s.t.}\quad & \pi D x_g + \sigma \le c x_g, \quad \forall g \in \bar{G} \\
& \pi \ge 0, \sigma \in \mathbb{R}^1
\end{aligned}
$$

- Let λ' and (π', σ') represent the primal and dual solutions of $RMLP$.

Solving by Column Generation

- **Observations:**
  - $z_{RMLP} = \sum_{g \in \bar{G}} (cx_g) \lambda'_g$ gives an upper bound on z_{MLP}
  - The reduced cost of column x_g associated with (π', σ') is $cx_g - \pi' D x_g - \sigma'$
  - Instead of examining the reduced costs of the huge number of columns, pricing can be carried out implicitly by solving a single IP over the set Z:

$$\xi = \min_{g \in G} (cx_g - \pi' D x_g) = \min_{x \in Z} (c - \pi' D) x \qquad (*)$$

  - MLP is solved when $\xi - \sigma' = 0$, i.e., when there is no column with negative reduced cost

Solving by Column Generation

- **Observations:**
  - Instead of examining the reduced costs of the huge number of columns, pricing can be carried out implicitly by solving a single IP over the set Z:

$$\xi = \min_{g \in G}(cx_g - \pi' D x_g) = \min_{x \in Z}(c - \pi' D)x \qquad (*)$$

  - The **pricing** problem $(*)$ is equivalent to the Lagrangian subproblem, hence, each pricing step provides a ***Lagrangian dual bound*** $\xi + \pi' d$
    - ✓ (π', ξ) forms a feasible solution for the dual of MLP

$$\{\max \pi d + \sigma : \ \pi D x_g + \sigma \le c x_g \ \forall g \in G, \pi \ge 0, \sigma \in \mathbb{R}^1\}$$

    therefore $\xi + \pi' d$ gives a lower bound on z_{MLP}.

  - Alternative method to update the dual values (Subgradient method)

Algorithm of Column Generation

i) Initialize primal and dual bounds $PB = +\infty$, $DB = -\infty$. Set $t = 1$.
Generate a subset of points $\{x^g\}_{g \in G^1}$ so that $RMLP$ is feasible

ii) Iteration t:

a) Solve $RMLP$ over the current set of columns $\{x_g\}_{g \in G^t}$ and let λ^t and (π^t, σ^t) be the primal and the dual solution respectively

b) If λ^t defines an integer solution of IP update PB. If $PB = DB$, **stop**

c) Solve the pricing problem:

$$(SP^t) \quad \zeta^t \quad = \min\{(c - \pi^t D)x : x \in Z\}$$

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

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

iii) Increment t and return to ii)

Algorithm of Column Generation

- When problem IP has a block diagonal structure with the k_{th} subproblem having optimal value ξ_k:
  - Upper bounds on value z_{MLP} are of the form $\pi' d + \sum_{k=1}^{K} \sigma'_k$
  - Lower bounds are of the form $\pi' d + \sum_{k=1}^{K} \xi_k$

- When the K subproblems are identical, these bounds take the form $\pi' d + K\sigma'$ and $\pi' d + K\xi$, respectively

Column Generation: Example

- Consider the following LP program:

$$
\begin{array}{llrrrrl}
(P) & \min & -2x_1 & -x_2 & -x_3 & +x_4 & \\
& s.t. & x_1 & & +x_3 & & \leq 2 \\
& & x_1 & +x_2 & & +2x_4 & \leq 3 \\
& & x_1 & & & & \leq 2 \\
& & x_1 & +2x_2 & & & \leq 5 \\
& & & -x_3 & & +x_4 & \leq 2 \\
& & & & 2x_3 & +x_4 & \leq 6 \\
& & x_1, & x_2, & x_3, & x_4, & \geq 0
\end{array}
$$

- Set Z consists of the last four constraints and the nonnegative restrictions
  - The third and forth constraints involve only x_1 and x_2,
  - The fifth and sixth constraints involve only x_3 and x_4

Column Generation: Example

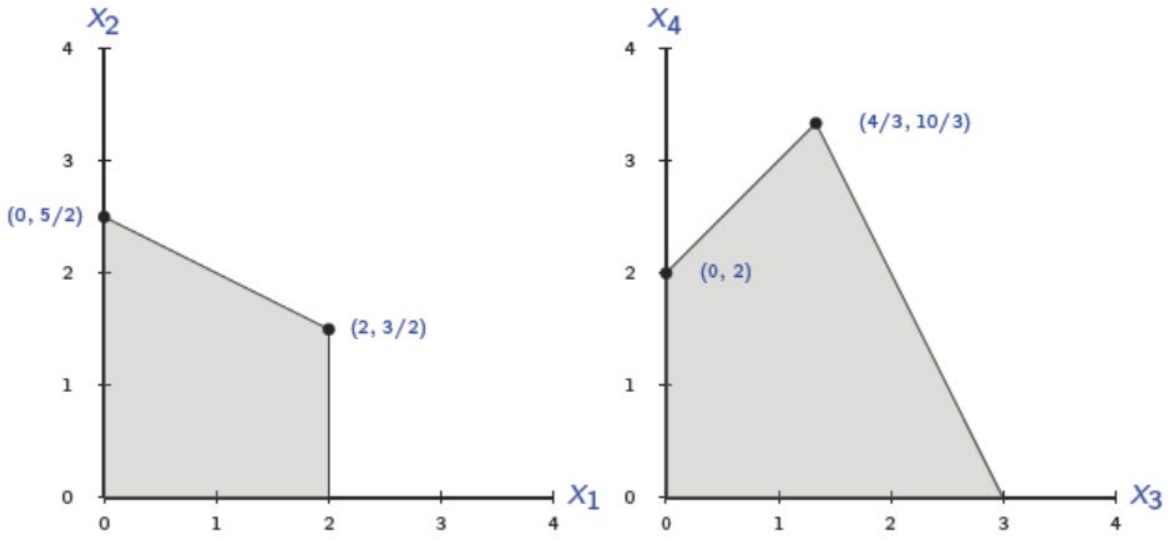
- Consider the following LP program:

$$
\begin{array}{llrrrrl}
(P) & \min & -2x_1 & -x_2 & -x_3 & +x_4 & \\
 & s.t. & x_1 & & +x_3 & & \leq 2 \\
 & & x_1 & +x_2 & & +2x_4 & \leq 3 \\
 & & x_1 & & & & \leq 2 \\
 & & x_1 & +2x_2 & & & \leq 5 \\
 & & & -x_3 & & +x_4 & \leq 2 \\
 & & & & 2x_3 & +x_4 & \leq 6 \\
 & & x_1, & x_2, & x_3, & x_4, & \geq 0
\end{array}
$$

- We handle the first two constraints as $Dx \leq d$:
  - $D = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 2 \end{bmatrix}$ and $d = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

Column Generation: Example

- Minimizing a linear function over Z becomes a simple process, since the subproblem can be decomposed into two subproblems:



(a) Z_1 (b) Z_2

where $Z_1 = \{(x_1, x_2) : x_1 \leq 2, x_1 + 2x_2 \leq 5, x_1, x_2 \geq 0\}$ and $Z_2 = \{(x_3, x_4)$ $-x_3 + x_4 \leq 2, 2x_3 + x_4 \leq 6, x_3, x_4 \geq 0\}$

Column Generation: Example

- The problem is reformulated as follows, where $x_1, x_2, \dots, x_t$ are the extreme points of Z, $\hat{c}_j = cx_j$ for $j = 1, \dots, t$ and $s \geq 0$ is the slack vector:

$$
\begin{array}{lll}
(P) & \min \sum_{j=1}^{t} \hat{c}_j \lambda_j & \\
& s.t. \sum_{j=1}^{t} (Dx^j)\lambda_j + s = d & \\
& \sum_{j=1}^{t} \lambda_j = 1, & \\
& \lambda_j \geq 0, & j = 1, \dots, t \\
& s \geq 0 &
\end{array}
$$

- $RMLP$ can be initialized with the extreme point $x_1 = (0,0,0,0)$ of Z of cost $cx_1 = 0$:

$$
\begin{array}{llllll}
(RMLP) & z_{RMLP} = \min 0\lambda_1 & & & & \\
& s.t. \quad s_1 & & & = 2 & (\pi_1) \\
& & s_2 & & = 3 & (\pi_2) \\
& & & \lambda_1 & = 1 & (\sigma) \\
& s_1, s_2, \lambda_1 \geq 0 & & & &
\end{array}
$$

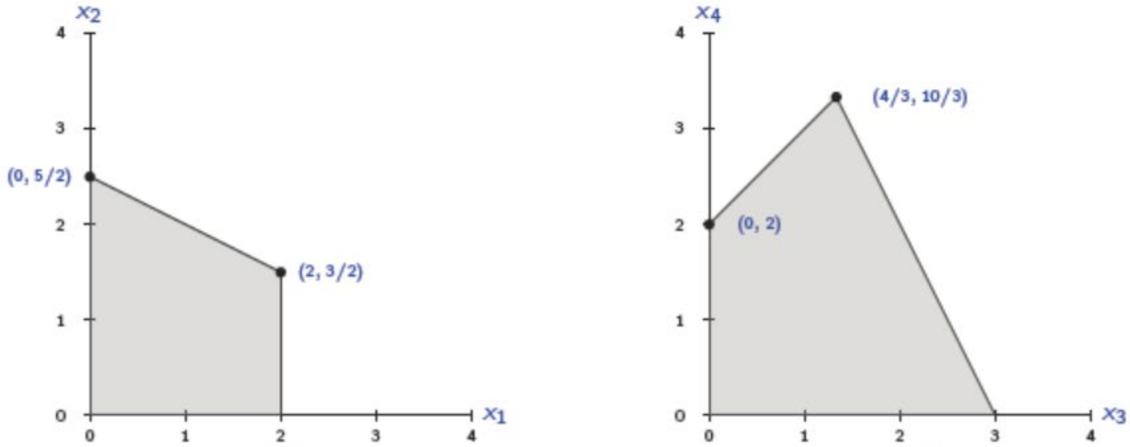
Column Generation: Example

- Iteration $t = 1$

a) Solve *RMLP*: $\lambda_1 = 1$, $s_1 = 2$, $s_2 = 3$, $z_{RMLP} = 0$ and $(\pi_1, \pi_2, \sigma) = (0, 0, 0)$

c) Solve the pricing problem (SP) $\zeta = \min\{(c - \pi D)x : x \in Z\}$:

$$(SP) \quad \min \; -2x_1 - x_2 - x_3 + x_4$$
$$s.t. \; x \in Z, \text{ or } (x_1, x_2) \in Z_1, (x_3, x_4) \in Z_2$$



(a) Z_1 (b) Z_2

The optimal solution is $x^2 = (2, 3/2, 3, 0)$ with $\zeta = -8.5$ $(\zeta - \sigma = -8.5 - 0 < 0)$

d) Dual bound: $L(\pi) = \pi d + \zeta = 0 - 8.5 = -8.5 \Rightarrow DB = \max\{-\infty, -8.5\} = -8.5$

Column Generation: Example

- Updating problem $RMLP$ with extreme point x^2:

$$
\begin{array}{llllll}
(RMLP) & \min & \hat{c}_1\lambda_1 & +\hat{c}_2\lambda_2 & & \\
 & s.t. & (Dx^1)\lambda_1 & +(Dx^2)\lambda_2 & +s & = d \\
 & & \lambda_1 & +\lambda_2 & & = 1 \\
 & & \lambda_1, & \lambda_2 & & \geq 0 \\
 & & & & s & \geq 0
\end{array}
$$

and as $Dx^1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $Dx^2 = \begin{bmatrix} 5 \\ 7/2 \end{bmatrix}$, $\hat{c}_2 = cx^2 = -17/2$ we have:

$$
\begin{array}{lllllll}
(RMLP) & z_{RMLP} = \min & 0\lambda_1 & +(-17/2)\lambda_2 & & & \\
 & s.t. & 0\lambda_1 & +5\lambda_2 & +s_1 & & = 2 \\
 & & 0\lambda_1 & +(7/2)\lambda_2 & & +s_2 & = 3 \\
 & & \lambda_1 & +\lambda_2 & & & = 1 \\
 & & \lambda_1, & \lambda_2, & s_1, & s_2 & \geq 0
\end{array}
$$

Column Generation: Example

- Iteration $t = 2$

a) Solve *RMLP*: $\lambda_1 = 3/5$, $\lambda_2 = 2/5$, $s_1 = 0$, $s_2 = 8/5$, $z_{RMLP} = -17/5 = -3.4$ and $(\pi_1, \pi_2, \sigma) = (-17/10, 0, 0)$

Notice that the best-known feasible solution to the overall problem of cost $z_{RMLP} = -3.4$ is give by $x = \lambda_1 x^1 + \lambda_2 x^2 = 3/5x^1 + 2/5x^2 = (4/5, 3/5, 6/5, 0)$

c) Solve the pricing problem (SP) $\quad \zeta \quad = \min\{(c - \pi D)x : x \in Z\}$:

$$c - \pi D = (-2, -1, -1, 1) - (-17/10, 0) \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 2 \end{bmatrix} = (-3/10, -1, 7/10, 1)$$

$$\begin{aligned} (SP) \quad & \min \; -3/10x_1 - x_2 + 7/10x_3 + x_4 \\ & \text{s.t.} \; x \in Z, \text{ or } (x_1, x_2) \in Z_1, (x_3, x_4) \in Z_2 \end{aligned}$$

The optimal solution is $x^3 = (0, 5/2, 0, 0)$ with $\zeta = -2.5$ ($\zeta - \sigma = -2.5 - 0 < 0$)

d) Dual bound: $L(\pi) = \pi d + \zeta = (-17/10, 0, 0)d - 2.5 = -17/5 - 2.5 = -5.9 \Rightarrow DB = \max\{-8.5, -5.9\} = -5.9$

Column Generation: Example

- Updating problem $RMLP$ with extreme point x^3:

$$
\begin{array}{llllll}
(RMLP) & \min & \hat{c}_1\lambda_1 & +\hat{c}_2\lambda_2 & +\hat{c}_3\lambda_3 & \\
 & s.t. & (Dx^1)\lambda_1 & +(Dx^2)\lambda_2 & +(Dx^3)\lambda_3 & +s = d \\
 & & \lambda_1 & +\lambda_2 & +\lambda_3 & = 1 \\
 & & \lambda_1, & \lambda_2, & \lambda_3 & \geq 0 \\
 & & & & & s \geq 0
\end{array}
$$

$\hat{c}_3 = cx^3 = -5/2$ and $Dx^3 = \begin{bmatrix} 0 \\ 5/2 \end{bmatrix}$ we have:

$$
\begin{array}{lllllllll}
(RMLP) & z_{RMLP} = \min & 0\lambda_1 & -17/2\lambda_2 & -5/2\lambda_3 & & & \\
 & s.t. & 0\lambda_1 & +5\lambda_2 & +0\lambda_3 & +s_1 & & = 2 \\
 & & 0\lambda_1 & +7/2\lambda_2 & +5/2\lambda_3 & & +s_2 & = 3 \\
 & & \lambda_1 & +\lambda_2 & +\lambda_3 & & & = 1, \\
 & & \lambda_1, & \lambda_2, & \lambda_3 & & & \geq 0 \\
 & & & & & s_1 & s_2 & \geq 0
\end{array}
$$

Column Generation: Example

- Iteration $t = 3$

a) Solve $RMLP$: $\lambda_1 = 0$, $\lambda_2 = 2/5$, $\lambda_3 = 3/5$, $s_1 = 0$, $s_2 = 1/10$, $z_{RMLP} = -49/10 = -4.9$ and $(\pi_1, \pi_2, \sigma) = (-6/5, 0, -5/2)$

The best-known feasible solution to the overall problem of cost $z_{RMLP} = -4.9$ is give by $x = \lambda_1 x^1 + \lambda_2 x^2 + \lambda_3 x^3 = 2/5x^2 + (3/5)x^3 = (4/5, 21/10, 6/5, 0)$

c) Solve the pricing problem (SP) $\quad \zeta \quad = \min\{(c - \pi D)x : x \in Z\}$:

$$c - \pi D = (-2, -1, -1, 1) - (-6/5, 0) \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 2 \end{bmatrix} = (-4/5, -1, 1/5, 1)$$

$$\begin{aligned} (SP) \quad & \min \; -4/5x_1 - x_2 + 1/5x_3 - x_4 \\ & \text{s.t.} \;\; x \in Z, \text{ or } (x_1, x_2) \in Z_1, \; (x_3, x_4) \in Z_2 \end{aligned}$$

The optimal solution is $x^4 = (2, 3/2, 0, 0)$ with $\zeta = -3.1$ ($\zeta - \sigma = -3.1 - (-5/2) = -0.6 < 0$)

d) Dual bound: $L(\pi) = \pi d + \zeta = (-6/5, 0, 0)d - 3.1 = -12/5 - 3.1 = -5.5 \Rightarrow DB = \max\{-5.9, -5.5\} = -5.5$

Column Generation: Example

- Updating problem *RMLP* with extreme point x^3, $\hat{c}_4 = cx^4 = -11/2$ and $Dx^4 = \begin{bmatrix} 2 \\ 7/2 \end{bmatrix}$:

$$
\begin{array}{llrrrrrrl}
(RMLP) & z_{RMLP} = \min & 0\lambda_1 & -17/2\lambda_2 & -5/2\lambda_3 & -11/2\lambda_4 & & & \\
& \text{s.t.} & 0\lambda_1 & +5\lambda_2 & +0\lambda_3 & +2\lambda_4 & +s_1 & & = 2 \\
& & 0\lambda_1 & +7/2\lambda_2 & +5/2\lambda_3 & +7/2\lambda_4 & & +s_2 & = 3 \\
& & \lambda_1, & +\lambda_2 & +\lambda_3 & +\lambda_4 & & & = 1 \\
& & \lambda_1, & \lambda_2, & \lambda_3, & \lambda_4, & s_1 & s_2 & \geq 0
\end{array}
$$

Column Generation: Example

- Iteration $t = 4$

a) Solve *RMLP*: $\lambda_1 = 0$, $\lambda_2 = 1/3$, $\lambda_3 = 1/6$, $\lambda_4 = 1/2$, $s_1 = 0$, $s_2 = 0$, $z_{RMLP} = -5$ and $(\pi_1, \pi_2, \sigma) = (-1, -1, 0)$
The best-known feasible solution to the overall problem of cost $z_{RMLP} = -5$ is give by $x = \lambda_2 x^2 + \lambda_3 x^3 + \lambda_4 x^4 = 1/3x^2 + (1/2)x^3 + (1/6)x^4 = (1, 2, 1, 0)$

c) Solve the pricing problem (SP) $\quad \zeta \quad = \min\{(c - \pi D)x : x \in Z\}$:

$$c - \pi D = (-2, -1, -1, 1) - (-1, -1)\begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 2 \end{bmatrix} = (0, 0, 0, 3)$$

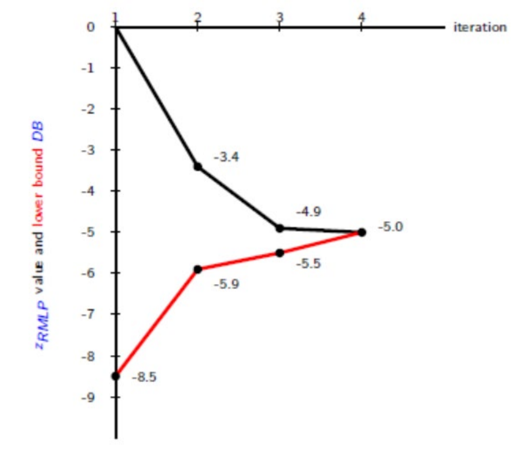
$$\begin{array}{lll} (SP) & \min & 0x_1 + 0x_2 + 0x_3 + 3x_4 \\ & s.t. & x \in Z, \text{ or } (x_1, x_2) \in Z_1, \ (x_3, x_4) \in Z_2 \end{array}$$

The optimal solution is $x^5 = (0, 0, 0, 0)$ with $\zeta = 0$ $(\zeta - \sigma = 0) \Rightarrow DB = -5$, stop

d) Dual bound: $L(\pi) = \pi d + \zeta = (-1, -1, 0)d - 0 = -5 - 0 = -5$

Column Generation: Example

- The optimal solution is given by $(x_1, x_2, x_3, x_4) = (1,2,1,0)$ with objective value -5.
- Progress of the lower bounds and the objective values for the primal feasible solutions generated:



- Note that $(1, 2)$ is not an extreme point of Z_1 and $(1, 0)$ is not an extreme point of Z_2
- In general, the decomposition algorithm may not provide an optimal extreme point of the overall problem if alternative optima exist

Improving Basic Column Generation

- **Main drawback**: degeneracy (upper bound Z_{RMLP} remains stuck at the same value)

- More sophisticated and robust mechanisms:
  - Warm start or proper initialization. The procedure is initialized with a dual solution π computed, i.e., using a dual heuristic
  - Stabilization techniques, that penalize deviations of the dual solutions from a stability center $\hat{\pi}$
  - Smoothing techniques that moderate the current dual solution based on previous iterates

Improving Basic Column Generation

- **Main drawback**: degeneracy (upper bound Z_{RMLP} remains stuck at the same value)

- More sophisticated and robust mechanisms:
  - Interior point approach, providing dual solutions corresponding to points in the center of the face of optimal solutions of $RMLP$

  - Lagrangian Column Generation. Instead of using the simplex algorithm, Lagrangian relaxation is used to approximate the optimal dual variables

    Baldacci, R., Christofides, N., & Mingozzi, A. (2008). An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. Mathematical Programming, 115(2), 351–385.
  - Reformulation strategies to avoid degeneracy or symmetries

Outline

01 Basic Theories

02 Column Generation

03 Branch-and-Price

04 Applications

Get Integral Solutions

- To solve *IP* problem based on its Dantzig-Wolfe reformulation, column generation and branch-and-bound must be combined

- The resulting algorithm is known as *branch-and-price* or *IP* column generation

- When cutting planes are used to strengthen the relaxation, the resulting algorithm is called *branch-price-and-cut* or *branch-and-cut-and-price*

- How to select **branching** constraints in order to enforce integrality and, their impact on problem RMLP and on the pricing problem?

Branch on Master Variables

- A standard branching scheme for DW_d consisting in imposing a disjunctive constraint on a fractional variable $\lambda_g^* = v$ of the Dantzig-Wolfe reformulation:
  - Up-branch (U): $\lambda_g \geq \lceil v \rceil$
  - Down-branch (D): $\lambda_g \leq \lfloor v \rfloor$

- Branching on the master variables is either not feasible (as for DW_c) or not advisable:
  - Branch D is weakly constraining while branch U significantly changes the solution $\rightarrow$ unbalanced enumeration tree
  - On branch D we need to add the constraint $x \neq x_g$ in the sub-problem which destroys its structure

Branch on Master Variables

- The alternative is to work simultaneously with original and master formulations, i.e., branching on the original variables x:

- Recall that when $Z \subseteq \mathbb{B}^n$:

$$x = \sum_{g \in G^c} x_g \lambda_g \in \mathbb{B}^n \quad \Leftrightarrow \quad \lambda \in \mathbb{B}^{|G^d|},$$

  - branching on the original variables is equivalent to branch on the master variables
  - Up-branch (U): $x_j \geq \lceil x_j^* \rceil$, the new IP problem is:

$$Z_U = \min\{cx: Dx \geq d, x \in Z, x_j \geq \lceil x^* \rceil\}$$

  - Down-branch (D): $x_j \leq \lfloor x_j^* \rfloor$, the new IP problem is defined similarly
- The branching decision can be enforced either in the master (Option 1) or in the pricing problem (Option 2)

Single or Multiple Distinct Subproblems: Option 1

- We have $Y_U^1 = \{x \in \mathbb{Z}^n : Dx \geq d, x_j \geq \lceil x^* \rceil\}$ and $Z_U^1 = Z$

Solving the new MLP

$$(MLP_1)\ \ Z_{MLP_1} = \min \textstyle\sum_{g\in G}(cx_g)\lambda_g$$

$$s.t.\ \ \textstyle\sum_{g\in G}(Dx_g)\,\lambda_g \geq d \qquad (\pi)$$

$$\textstyle\sum_{g\in G} x_{g,j}\,\lambda_g \geq \lceil x_j^* \rceil \qquad (\mu)$$

$$\textstyle\sum_{g\in G} \lambda_g = 1 \qquad (\sigma)$$

$$\lambda \in \mathbb{R}_+^{|G|}$$

where $\{x_g\}_{g\in G}$ is the set of points of Z

Solving the new subproblem

$$(SP_1)\ \zeta_1 = \min\{(c - \pi D)x - \mu x_j : x \in Z\}$$

where $(\pi, \mu, \sigma) \in \mathbb{R}_+^m \times \mathbb{R}_+^1 \times \mathbb{R}^1$

Single or Multiple Distinct Subproblems: Option 2

- We have $Y_U^2 = \{x \in \mathbb{Z}^n : Dx \geq d\}$ and $Z_U^2 = Z \cap \{x_j \geq \lceil x^* \rceil\}$

Solving the new MLP

$$(MLP_2)\ \ Z_{MLP_2} = \min \sum_{g \in G_2} (cx_g)\lambda_g$$

$$s.t.\ \ \sum_{g \in G_2} (Dx_g)\,\lambda_g \geq d \qquad (\pi)$$

$$\sum_{g \in G_2} \lambda_g = 1 \qquad (\sigma)$$

$$\lambda \in \mathbb{R}_+^{|G_2|}$$

where $\{x_g\}_{g \in G_2}$ is the set of points of Z_U^2.

Solving the new subproblem

$(SP_2)\ \zeta_2 = \min\{(c - \pi D) : x \in Z \cap \{x_j \geq \lceil x_j^* \rceil\}\}$

where $(\pi, \sigma) \in \mathbb{R}_+^m \times \mathbb{R}^1$

- As Z is partitioned in Z_U^2 and $Z \setminus Z_U^2$, and $\sum_{g \in G_2} \lambda_g = 1$, then $\sum_{g \in G \setminus G_2} \lambda_g = 0$, i.e., columns of $Z \setminus Z_U^2$ are removed from the master

Comparison of Options 1 and 2

- ***Strength*** of the linear programming bound

$$Z_{MLP_1} = \min\{cx: Dx \geq d, x \in conv(Z), x_j \geq \lceil x_j^* \rceil\}$$

$$\leq \quad Z_{MLP_2} = \min\{cx: Dx \geq d, x \in conv(Z \cap \{x_j \geq \lceil x_j^* \rceil\})\}$$

$\rightarrow$ Option 2 leads to better bound

- ***Complexity*** of the subproblem
  - Option 1: the subproblem is unchanged
  - Option 2: the subproblem may become more complicated

$\rightarrow$ Option 2 may be preferable if the modified subproblem remains “tractable”

- Getting Integer Solutions
  - Option 2 allows to generate points in the interior of $conv(Z)$

Useful References for Pricing Problem

- The pricing problems usually are solved via label-setting algorithm
  - A type of dynamic programming: state representation and extension
  - **Dominance rules** are essential for efficiency

- References for Capacitated Vehicle Routing Problem

  - Vanderbeck, F. (2005). Implementing Mixed Integer Column Generation. In M. Desaulniers, Guy and Desrosiers, Jacques and Solomon (Ed.), Column Generation (pp. 331–358). Springer US.
  - Feillet, D., Dejax, P., Gendreau, M., & Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. Networks, 44(3), 216–229.
  - Righini, G., & Salani, M. (2008). New dynamic programming algorithms for the resource constrained elementary shortest path problem. Networks, 51(3), 155–170.
  - Martinelli, R., Pecin, D., & Poggi, M. (2014). Efficient elementary and restricted non-elementary route pricing. European Journal of Operational Research, 239(1), 102–111.
  - Zhang, Z., Luo, Z., Qin, H., & Lim, A. (2019). Exact Algorithms for the Vehicle Routing Problem with Time Windows and Combinatorial Auction. Transportation Science, 53(2), 427–441.

Branch-and-Price Algorithm: Practical Aspects

- ***Issues*** that must be considered in developing a branch-and-price algorithm:
  - Initialization of the restricted master program
  - Stabilization of the column generation procedure
  - Combining column and cut generation
  - Branching strategies
  - Primal heuristics and preprocessing techniques
  - Master feasibility

Outline

01 Basic Theories

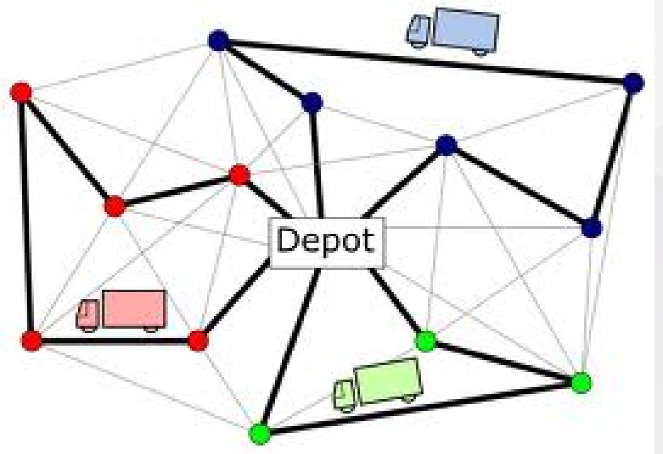
02 Column Generation

03 Branch-and-Price

04 Applications

Capacitated Vehicle Routing Problem (CVRP)

- **Problem description**
  - A fleet of K identical vehicles located at a central depot 0 can be used to serve a set N_c of customers which geographically surround the depot.
  - Each vehicle has a given capacity Q, and each customer has a given demand q_i and can be serviced exactly once. Each arc (i,j) has a distance d_{ij}.
  - Each customer is served exactly once by one vehicle.
  - The goal is to minimize the total travel distance.
  - Can you provide the mathematical model?



Capacitated Vehicle Routing Problem (CVRP)

- **data and parameters**
  - $G = \{N, A\}$: undirected graph
  - $N = \{0\} \cup N_c$: set of nodes with 0 as the depot and N_c as the set of customers
  - $A = \{(i, j): i, j \in N, i \neq j\}$: the set of arcs
  - K: set of identical vehicles
  - d_{ij}: the distance of arc $(i, j) \in A$
  - q_i: the demand of customer $i \in N_c$, and let $q_0 = 0$
  - Q: the vehicle capacity

Two-index Model

- **decision variables**
  - x_{ij}: binary variable, =1 if arc (i, j) is traversed
  - u_i: continuous variable, the vehicle load after serving the customer i
- **Mathematical model:**

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

$$s.t. \quad \sum_{j\in N:(i,j)\in A} x_{ij} = \sum_{j\in N:(j,i)\in A} x_{ji} = 1, \ \forall i \in N_c$$

$$\sum_{i\in N_c} x_{0i} \le |K|$$

$$u_i \ge u_j + q_j - Q\left(1 - x_{ij}\right), \ \forall i, j \in N, i \ne j$$

$$0 \le u_i \le Q, \ \forall i \in N$$

$$x_{ij} \in \{0,1\}, \ \forall (i, j) \in A$$

Set-partitioning model

- **Notations and variables**
  - R: index set of all feasible routes
  - a_{ir}: binary coefficient, =1 if $i \in V$ belongs to route $r \in R$
  - c_r: the total travel cost associated with the route $r \in R$
  - θ_r: binary variable, =1 if route $r \in R$ is chosen in the solution
- **Mathematical model:**

$$\min \sum_{r \in R} c_r \theta_r$$

$$\text{s.t.} \quad \sum_{r \in R} a_{ir} \theta_r = 1, \forall i \in V_c \qquad \text{(customer)}$$

$$\sum_{r \in R} \theta_r \leq m \qquad \text{(vehicle number)}$$

$$\theta_r \in \{0, 1\}, \forall r \in R.$$

- **Advantages?**

Column Generation

- Restricted Master Problem Linear Relaxation (RMLP): R $\rightarrow \bar{R}$

- **Dual variables**
  - $\lambda_i, i \in V_c; \quad and \quad \lambda_0$
- **Dual problem:**

$$\max \sum_{i \in V_c} \lambda_i + m\lambda_0$$

$$\text{s.t.} \quad \boxed{\sum_{i \in V_c} a_{ir}\lambda_i + \lambda_0 \le c_r, \quad \forall r \in \bar{R}}$$

$$\lambda_i \in \mathbb{R}, \forall i \in V_c$$

$$\lambda_0 \le 0.$$

- **Reduced cost:** $c_r - \sum_{i \in V_c} a_{ir}\lambda_i - \lambda_0 \ge 0, \forall r \in \bar{R}$

Pricing Problem

- Find the column with negative reduced cost or prove none exist

$$\min \quad \sum_{(i,j)\in A} d_{ij}x_{ij} - \sum_{(i,j)\in A} \lambda_i x_{ij} - \lambda_0$$

$$s.t. \quad \sum_{j\in N:(i,j)\in A} x_{ij} = \sum_{j\in N:(j,i)\in A} x_{ji}, \ \forall i \in N_c$$

$$u_i \geq u_j + q_j - Q(1 - x_{ij}), \ \forall i,j \in N, i \neq j$$

$$0 \leq u_i \leq Q, \ \forall i \in N$$

$$x_{ij} \in \{0,1\}, \ \forall (i,j) \in A$$

Pricing Problem

- Find the column with negative reduced cost or prove none exist

$$c_r - \sum_{i \in V_c} a_{ir}\lambda_i - \lambda_0 < 0 \quad \Longrightarrow \quad r \in \bar{R}$$

$$\sum_{(i,j)\in A} x_{ij}\left(d_{ij} - \lambda_j\right) < 0$$

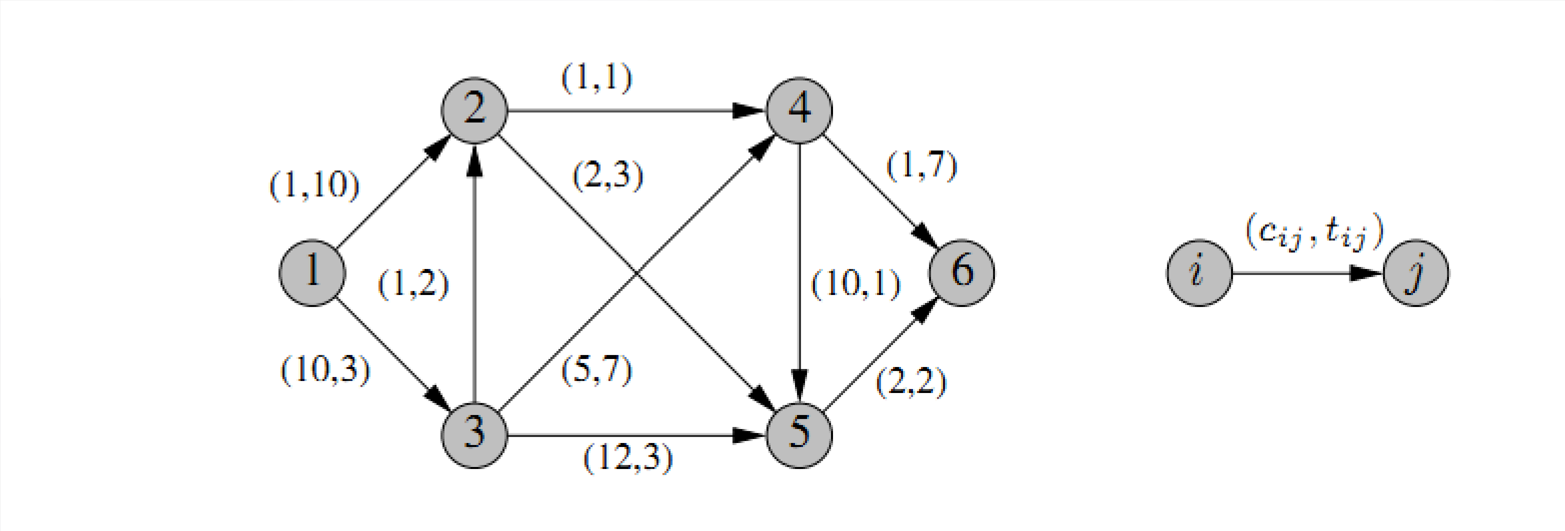
$$min \sum_{(i,j)\in A} x_{ij}\boxed{\left(d_{ij} - \lambda_j\right)}$$

Shortest Path:

Pricing Problem

- Elementary Shortest Path Problem with Resource Constraints ***(NP-hard)***

- Shortest Path Problem with Resource Constraints (Pseudo-polynomial)



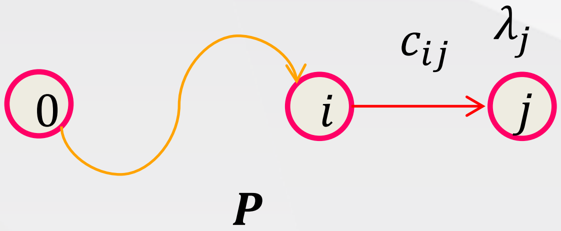
Time Constraint Shortest Path Problem (Ahuja et al. 1993)

Pricing Problem

- **Common algorithm:** Label setting algorithm/ Dynamic programming

- **Label:** $L_i = \{C_i, q_i, t_i, r_i\}$ for the partial path $(0, \dots, i)$
  - C_i : reduced cost
  - q_i: consumed capacity
  - t_i: consumed time
  - r_i: the set of visited customers

- **Label extension:** update the resource information



- **Dominance rule**: discard the unpromising labels to avoid enumeration

Branching

- Branching on the vehicle number: $\sum_{r \in R} \theta_r^*$

- Branching on the two consecutive arcs: $i \to j \to k$

- Branching on the arc: $x_{ij} = \sum_{r \in R} \sum_{i,j} b_{ij}^r \theta_r^*$

$$
\begin{aligned}
\min\ & \sum_{r \in R} c_r \theta_r \\
\text{s.t.}\ & \sum_{r \in R} a_{ir} \theta_r = 1, \forall i \in V_c \\
& \sum_{r \in R} \theta_r \leq m \\
& \theta_r \in \{0, 1\}, \forall r \in R.
\end{aligned}
$$

- Note that some branching may add constraints to the RMLP, while some can be dealt in the pricing problem by modifying the graph

Acceleration and Improvements

- **Pricing problem**:
  - Non-elementary routes:
    - ✓ Q-route relaxation: $n(Q+1)$ vector with complexity $O(n^2Q)$
    - ✓ Q-routes with k-Cycle Elimination: avoid the cycle with less than k visits
    - ✓ Ng-routes (Baldacci et al.): ng-set (memory) for each customer
  - Strong dominance rules
  - Bi-directional search
  - Completion bounds
  - Decremental State Space Relaxation (DSSR)

Acceleration and Improvements

- **Column Generation**:
  - Dual stabilization (Du Merle et al.)
  - Faster pricing heuristics first
  - Dynamically update and control the Column pool
  - Routes enumeration
- **Branching**
  - Strong branching:
    - ✓ A simpler branching over individual edges (Fractional variables)
    - ✓ Quick evaluation of 30 candidates and produce ranking. Evaluate the best one fully. Other candidates with good ranking are better evaluated
    - ✓ Collect the previous evaluations of a variable, which could be a good predictors of future evaluations

Acceleration and Improvements

- **Cut Generation:**
  - Robust cuts:
    - ✓ not change the structure of pricing problem
    - ✓ Cuts applied to two-index formulations are both robust cuts

$$\sum_{i,j} \beta_{ij} x_{ij} \leq \beta_0 \quad \rightarrow \quad \sum_{r \in R} \sum_{i,j} \beta_{ij} b_{ij}^{r} \theta_r \leq \beta_0$$

      - ✓ (Strengthened) Rounded capacity cuts
      - ✓ 2-path cuts
      - ✓ Strengthened com cuts
      - ✓ (Partial) multi-star cuts
  - Non-robust cuts: $\sum_{r \in R} \beta_r \theta_r \leq \beta_0$
    - ✓ β_r might not linear functions of the arc flows
    - ✓ Subset-row cuts (Jepsenetal.[2008]), Limited memory subset-row cuts

Acceleration and Improvements

- **Machine learning techniques?**
  - Column generation
  - Cut generation
  - Node exploration
  - Branching
  - Decomposition

References

- Bertsimas, D. and Tsitsiklis, J.N., 1997. Introduction to linear optimization (Vol. 6, pp. 479-530). Belmont, MA: Athena Scientific.
- Wolsey, L.A., 1998. Integer programming (Vol. 52). John Wiley & Sons.
- Desaulniers, G., Desrosiers, J. and Solomon, M.M. eds., 2006. Column generation (Vol. 5). Springer Science & Business Media.
- Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W. and Vance, P.H., 1998. Branch-and-price: Column generation for solving huge integer programs. Operations Research, 46(3), pp.316-329.
- Feillet, D., 2010. A tutorial on column generation and branch-and-price for vehicle routing problems. 4or, 8(4), pp.407-424.
- Costa, L., Contardo, C. and Desaulniers, G., 2019. Exact branch-price-and-cut algorithms for vehicle routing. Transportation Science, 53(4), pp.946-985.

THANK YOU

Zhenzhen ZHANG

School of Economics and Management, Tongji University

zhenzhenzhang@tongji.edu.cn