Branch-and-price and related topics

applications in cutting stock

J.M. Valério de Carvalho
vc@dps.uminho.pt

Departamento de Produção e Sistemas
Escola de Engenharia, Universidade do Minho

Seminário, CEOC
Universidade do Aveiro
26 de Outubro de 2007

Outline

1. Integer programming modelling
   - Strength of integer programming models
   - Dantzig-Wolfe decomposition
   - 3 different models: IP, LP, DW
2. Cutting Stock Problem (CSP)
   - CSP models
   - Column generation for CSP
3. Branch-and-price
   - Compatibility and robust branching schemes
   - Arc flow model
   - Branch-and-price algorithm for CSP
4. Acceleration of column generation with dual cuts
   - Motivation
   - Dual cuts for CSP

Integer programming: strength of models

Integer Programming Problems (IP):

$$\begin{aligned} z_{IP} = \min \quad & cx \\ subject\ \ to \quad & Ax = b \\ & x \geq 0 \text{ and integer} \end{aligned}$$

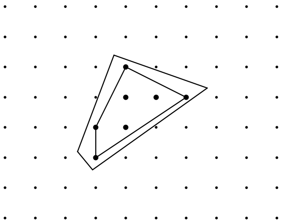
can be solved by branch-and-bound using the Linear Programming (LP) relaxation that results from relaxing the integrality conditions:

$$\begin{aligned} z_{LP} = \min \quad & cx \\ subject\ \ to \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

Crucial issue: some IP models are stronger, because their LP relaxations:

- provide closer description of convex hull of valid integer solutions.
- have LP optimal solution values closer to IP optimal solution values (smaller gap).

Motivation for branch-and-price



Some strong IP models have an exponential number of variables.

Solve them combining **column generation** and **branch-and-bound**.

Dantzig-Wolfe decomposition

May provide strong models (stronger than plain LP relaxation)...
... with an exponential number of variables.

$$
\begin{array}{ll}
\min & cx \\
suj. & Ax = b \\
 & x \in X \\
 & x \geq 0 \text{ and integer}
\end{array}
$$

Constraints decomposed in two sets:

- first set: general constraints → **Master Problem**.
- second set: constraints with special structure → **Subproblem**

Subproblem must be amenable for separate solution.

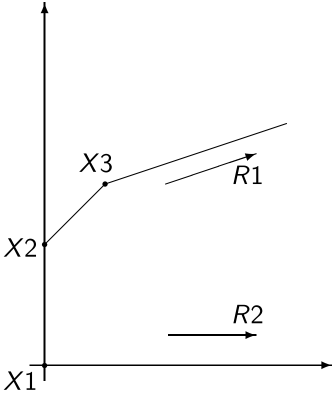
Dantzig-Wolfe decomposition: representation of a point

$$\begin{array}{ll} \min & cx \\ suj. & Ax = b \\ & \boxed{x \in X} \\ & x \geq 0 \end{array}$$

- Polyhedron X has I extreme points, denoted as $X_1, X_2, \ldots, X_I$, and K extreme rays, denoted as $R_1, R_2, \ldots, R_K$.
- Any point $x \in X$ is expressed as a convex combination of the extreme points of X plus a non-negative combination of the extreme rays of X :

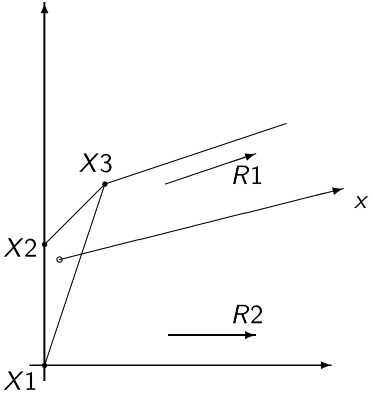
$$X = \left\{ x = \sum_{i=1}^{I} \lambda_i X_i + \sum_{k=1}^{K} \mu_k R_k, \sum_{i=1}^{I} \lambda_i = 1, \lambda_i \geq 0, \forall i, \mu_k \geq 0, \forall k \right\}$$

Dantzig-Wolfe decomposition: graphical representation



X_1,X_2 and X_3 are extreme points, and R_1 and R_2 are extreme rays.
Valid space is unbounded.

Dantzig-Wolfe decomposition: graphical representation



x is expressed as a convex combination of X_1, X_2 and X_3 plus a non-negative combination of R_1 and R_2.

Some rewriting work...

replacing x in $\min\{cx : Ax = b, x \in X, x \geq 0\}$, we obtain

$$
\begin{array}{ll}
\min & c(\sum_{i=1}^{I} \lambda_i X_i + \sum_{k=1}^{K} \mu_k R_k) \\
suj. & A(\sum_{i=1}^{I} \lambda_i X_i + \sum_{k=1}^{K} \mu_k R_k) = b \\
 & \sum_{i=1}^{I} \lambda_i = 1 \\
 & \lambda_i \geq 0, \forall i \\
 & \mu_k \geq 0, \forall k
\end{array}
$$

Reformulation of the problem: master problem

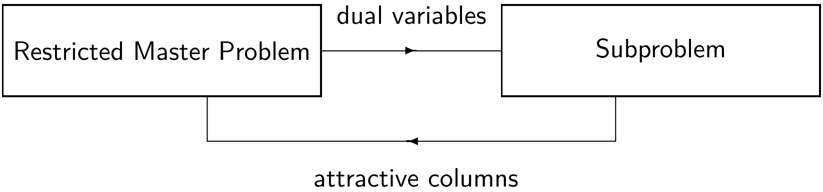
$$
\begin{aligned}
\max \quad & \sum_{i=1}^{I} (cX_i)\lambda_i + \sum_{k=1}^{K} (cR_k)\mu_k \\
\textit{subj. to} \quad & \sum_{i=1}^{I} (AX_i)\lambda_i + \sum_{k=1}^{K} (AR_k)\mu_k = b \\
& \sum_{i=1}^{I} \lambda_i = 1 \\
& \lambda_i \geq 0, \forall i \\
& \mu_k \geq 0, \forall k
\end{aligned}
$$

Decision variables: λ_i and μ_k.
Reformulated model is equivalent to original model.
Number of extreme points and extreme rays can be exponentially large.
Use column generation!

Column generation

Solve linear programming relaxation using column generation:

Choose an initial restricted set of columns
While (there is a column with negative reduced cost) do
    add column to restricted problem
    reoptimize
End While



[Dantzig, Wolfe, 1960; Ford, Fulkerson, 1958]

Integrality property

If X does not have the integrality property, the reformulated model is stronger than the linear programming relaxation.

Instead of searching extreme points and extreme rays in:

$$x \in Conv\{x \in X\},$$

search in:

$$x \in Conv\{x \in X \text{ and integer}\}.$$

That may not be too hard: in the Cutting Stock Problem, we have to find an integer solution of the subproblem (knapsack problem).

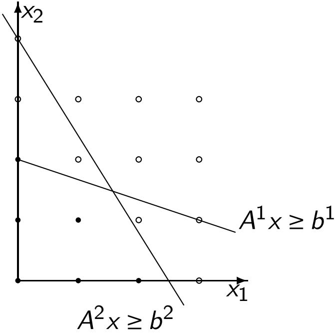
3 different models: IP, LP, DW

$$\begin{aligned} z_{IP} = \min \quad & cx \\ subj.\ to \quad & Ax = b \\ & x \in X \\ & x \geq 0 \text{ and integer} \end{aligned}$$

$$\begin{aligned} z_{LP} = \min \quad & cx \\ subj.\ to \quad & Ax = b \\ & x \in X \\ & x \geq 0 \end{aligned}$$

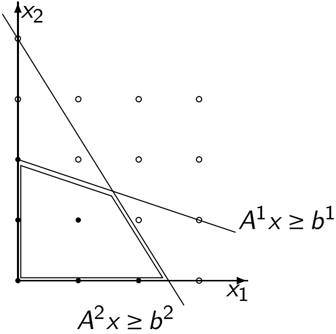
$$\begin{aligned} z_{DW} = \min \quad & cx \\ subj.\ to \quad & Ax = b \\ & x \in Conv\{x \in X \text{ and integer}\} \\ & x \geq 0 \end{aligned}$$

Integer Problem: domain is a finite set of points



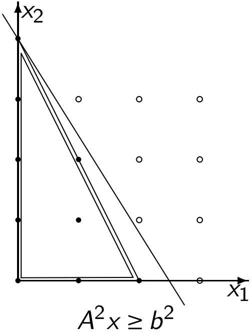
$$\begin{aligned} z_{IP} &= \min cx \\ \textit{subj. to} \quad & A^1x \geq b^1 \\ & A^2x \geq b^2 \\ & x \geq 0 \text{ and integer} \end{aligned}$$

Linear programming relaxation



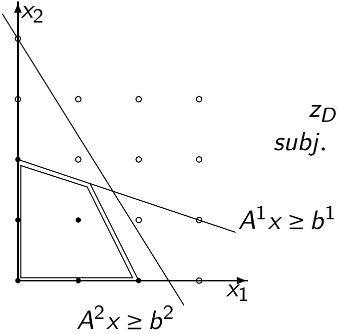
$$\begin{aligned} z_{LP} &= \min cx \\ \textit{subj. to} \quad & A^1x \geq b^1 \\ & A^2x \geq b^2 \\ & x \geq 0 \end{aligned}$$

$A^2x \geq b^2$ does not have the integrality property



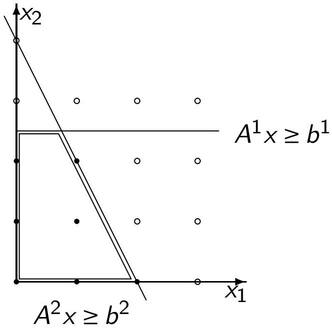
$x \in Conv\{A^2x \geq b^2 \text{ and integer}\}$

Reformulated model

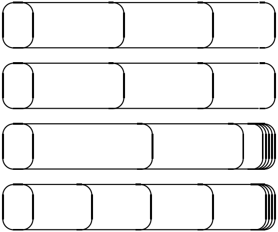


$$\begin{aligned} z_{DW} &= \min cx \\ \text{subj. to} \quad & A^1x \geq b^1 \\ & x \in Conv\{A^2x \geq b^2 \text{ and integer}\} \\ & x \geq 0 \end{aligned}$$

If X is an integer polytope $\Rightarrow$ same bound as LP



Cutting Stock Problem



W: width of large rolls
w_i : width of rolls for client i, $i = \ldots, m$
b_i : demand of rolls of width w_i (many items of each size)
Objective: cut the minimum number of rolls to satisfy demand.

Cutting Stock Problem: glimpse of a weak model

Decision variables $x_{ij} = \begin{cases} 1 & \text{, if item } j \text{ is placed in roll } i \\ 0 & \text{, otherwise} \end{cases}$

Decision variables $y_i = \begin{cases} 1 & \text{, if roll } i \text{ is used} \\ 0 & \text{, otherwise} \end{cases}$

$$
\begin{aligned}
\min z_{IP} &= \sum_{i=1}^{n} y_i \\
\text{subj. to} \quad & \sum_{j=1}^{n} w_j x_{ij} \leq W y_i, \ \forall i \in I \\
& \sum_{i=1}^{n} x_{ij} = 1, \ \forall j \in J \\
& y_i = 0 \text{ or } 1, \ \forall i \\
& x_{ij} = 0 \text{ or } 1, \ \forall i, j
\end{aligned}
$$

L. Kantorovich, "Mathematical methods of organising and planning production" (translated from a paper in Russian, dated 1939), Management Science 6, 366–422, 1960.

Cutting Stock Problem: Gilmore-Gomory model

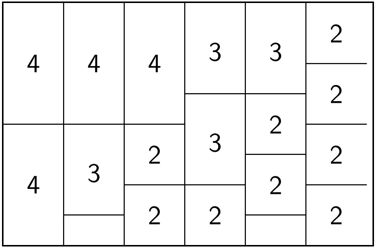
Cutting Pattern: possible arrangement of items in the roll:

$$\sum_{i=1}^{m} a_{ij} w_i \leq W$$
$$a_{ij} \geq 0 \text{ and integer }, \ \forall j \in J.$$

a_{ij} : number of items of width w_i obtained in the cutting pattern j
J : set of valid cutting patterns.
x_j : number of rolls cut according cutting pattern j.

$$\begin{aligned} min\ z_{IP} &= \sum_{j \in J} x_j \\ subject\ \ to \quad & \sum_{j \in J} a_{ij} x_j \geq b_i, \ i = 1, 2, \ldots, m \\ & x_j \geq 0 \text{ and integer }, \ \forall j \in J \end{aligned}$$

(Very Small) Example



$W = 8$	cutting patterns						Demand b_i	
	x_1	x_2	x_3	x_4	x_5	x_6		
$w_i = 4$	2	1	1				$\geq$	5
3		1		2	1		$\geq$	4
2			2	1	2	4	$\geq$	8
min	1	1	1	1	1	1		

(Very Small) Example (cont.)

$W = 8$	cutting patterns						Demand b_i	
	x_1	x_2	x_3	x_4	x_5	x_6		
$w_i = 4$	2	1	1				$\geq$	5
3		1		2	1		$\geq$	4
2			2	1	2	4	$\geq$	8
min	1	1	1	1	1	1		

Optimal fractional solution

x_1	x_2	x_3	x_4	x_5	x_6	
2.5			2.0		1.5	6 rolls

Fractional solution rounded up

x_1	x_2	x_3	x_4	x_5	x_6	
3.0			2.0		2.0	7 rolls

Excess production: 1 item of width 4 and 2 items of width 2

Column generation for CSP [Gilmore, Gomory, 1961]

Generally, it is unpractical to enumerate all valid cutting patterns.

Solve linear programming relaxation of CSP using column generation:

Choose an initial restricted set of cutting patterns
While (there is an a attractive cutting pattern) do
    add attractive cutting pattern to restricted problem
    reoptimize
End While

To get an integer solution, round up fractional values of cutting patterns.
Solutions are of good quality, if the quantities demanded are high.

Cutting Stock Problem: Restricted Problem

$$\begin{aligned} \textit{min} \quad & z_{LP} = \sum_{j \in \overline{J}} x_j \\ \textit{subject to} \quad & \sum_{j \in \overline{J}} a_{ij} x_j \geq b_i, \ \ i = 1, 2, \ldots, m \\ & x_j \geq 0, \ \ \forall j \in \overline{J}, \end{aligned}$$

$\overline{J}$: subset of cutting patterns in restricted problem
$\pi = \pi(\overline{J}) = (\pi_1, \pi_2, \ldots, \pi_m)$: optimal dual solution with subset $\overline{J}$

Pricing cutting patterns out of the restricted problem:

Reduced cost of cutting pattern j: $1 - \sum_{i=1}^{m} a_{ij}\pi_i$
Column is attractive if its reduced cost < 0

Find most attractive cutting pattern $\in J \setminus \overline{J}$:

$$\min_{j \in J \setminus \overline{J}} 1 - \sum_{i=1}^{m} a_{ij}\pi_i$$

Cutting Stock Problem: knapsack subproblem

Columns in $\overline{J}$ have reduced costs ≥ 0; so, search over J:

$$\min_{j \in J} \; 1 - \sum_{i=1}^{m} a_{ij}\pi_i \;\equiv\; \max_{j \in J} \; \sum_{i=1}^{m} a_{ij}\pi_i - 1$$

Knapsack subproblem:

$$\begin{aligned} \max \; z_s = \; & \sum_{i=1}^{m} \pi_i y_i \\ \textit{subject to} \; & \sum_{i=1}^{m} w_i y_i \leq W \\ & y_i \geq 0 \text{ and integer}, \; i = 1, 2, \ldots, m \, , \end{aligned}$$

y_i : number of items of size w_i in the new cutting pattern

If optimum $z_s^* > 1$, cutting pattern is attractive.
If no attractive columns, solution is optimal.

Restricted problem: first iteration

Initial solution: 3 columns, each with items of the same size.
Optimal solution:

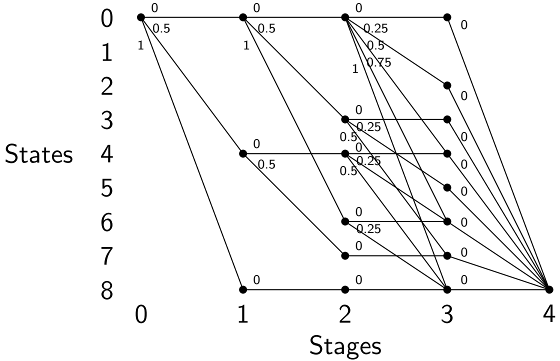
	x_1	x_2	x_3			dual
$w_d = 4$	2			$\geq$	5	0.5
3		2		$\geq$	4	0.5
2			4	$\geq$	8	0.25
min	1	1	1			

primal	2.5	2.0	2.0

$z^0 = 6.5$

Subproblem: first iteration

$$\begin{array}{lrrrl} \max\ z_s = & 0.5y_1 + & 0.5y_2 + & 0.25y_3 & \\ \textit{subject to} & 4y_1 + & 3y_2 + & 2y_3 & \leq 8 \\ & \multicolumn{4}{l}{y_j \geq 0 \text{ and integer}, \forall j} \end{array}$$



Optimal solution: $(y_1, y_2, y_3) = (0, 2, 1), z_s^* = 1.25 \rightarrow$ Attractive

Restricted problem: second iteration

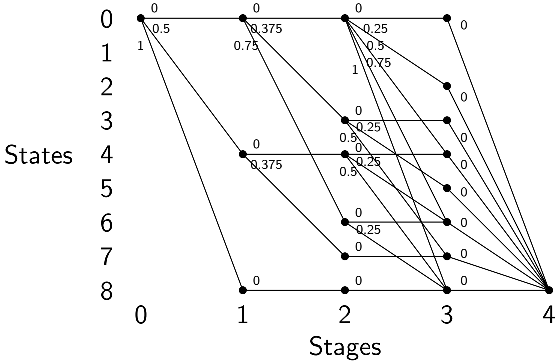
Attractive cutting pattern: 2 items of size 3 and 1 item of size 2.
Insert attractive column in the restricted problem, and reoptimize.
Optimal solution:

	x_1	x_2	x_3	x_4			dual
$w_d = 4$	2				$\geq$	5	0.5
3		2		2	$\geq$	4	0.375
2			4	1	$\geq$	8	0.25
min	1	1	1	1			
primal	2.5	0.0	1.5	2.0			

$z_{LP} =$ 6.0

Subproblem: second iteration

$$\begin{array}{llllll} \max \ z_s = & 0.5y_1 + & 0.375y_2 + & 0.25y_3 & \\ \textit{subject to} & 4y_1 + & 3y_2 + & 2y_3 & \leq 8 \\ & y_j \geq 0 \text{ and integer}, \forall j & & & \end{array}$$



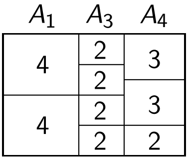
Alternative optima (Value $z_s^* = 1.0$) → No attractive columns. So...

Optimal solution of the linear relaxation

	x_1	x_2	x_3	x_4			dual
$w_d = 4$	2				$\geq$	5	0.5
3		2		2	$\geq$	4	0.375
2			4	1	$\geq$	8	0.25
min	1	1	1	1			

primal: 2.5 0.0 1.5 2.0

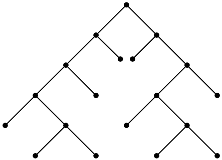
$z_{LP} = 6.0$



$x_j =$ 2.5 1.5 2.0

Getting integer solutions with branch-and-price

Branch-and-price = branch-and-bound + column generation



Methodology

- Branching constraints are introduced in the restricted master.
- After branching, deep in the tree, new columns may be needed.
- Column generation still has to work correctly.

Compatibility between Master Problem and Subproblem

Structure of the restricted master problem

- Branching constraints change the structure of the restricted master problem.
- Subproblem has to identify correctly the attractive and non-attractive columns with respect to the new structure.

Robust branching scheme

- Branching scheme should not induce intractable changes in the structure of the subproblem.
- Desirably, subproblem should be the same optimization problem both during the linear relaxation and branch-and-price.

Branching schemes

Branching on variables of the reformulated model

Regeneration of variables: a column set to zero by a branching constraint in the restricted master problem may turn out to be the most attractive column generated by the subproblem.

Branching on original variables

Original variables: variables of model to which the Dantzig-Wolfe decomposition is applied.
Successful in many applications.
Often, original variables are related with flows in arcs.

Cutting Stock Problem: arc flow model [VC, 1999]

- Rolls of integer capacity W and items of integer size $w_1, \ldots, w_d, \ldots, w_m$.

Cutting Stock Problem: arc flow model [VC, 1999]

- Rolls of integer capacity W and items of integer size $w_1, \ldots, w_d, \ldots, w_m$.
- Oriented acyclic graph $G = (V, A)$.

Cutting Stock Problem: arc flow model [VC, 1999]

- Rolls of integer capacity W and items of integer size $w_1, \dots, w_d, \dots, w_m$.
- Oriented acyclic graph $G = (V, A)$.
- $V = \{0, 1, 2, \dots, W\}$.

Cutting Stock Problem: arc flow model [VC, 1999]

- Rolls of integer capacity W and items of integer size $w_1, \ldots, w_d, \ldots, w_m$.
- Oriented acyclic graph $G = (V, A)$.
- $V = \{0, 1, 2, \ldots, W\}$.
- $A = \{(i, j) : 0 \leq i < j \leq W$ and $j - i = w_d,\ d = 1, \ldots, m\}$: length of oriented arc defines size of item.

Cutting Stock Problem: arc flow model [VC, 1999]

- Rolls of integer capacity W and items of integer size $w_1, \ldots, w_d, \ldots, w_m$.
- Oriented acyclic graph $G = (V, A)$.
- $V = \{0, 1, 2, \ldots, W\}$.
- $A = \{(i,j) : 0 \leq i < j \leq W \text{ and } j - i = w_d,\ d = 1, \ldots, m\}$: length of oriented arc defines size of item.
- Additional arcs $(k, k+1), k = 0, 1, \ldots, W-1$, correspond to loss.

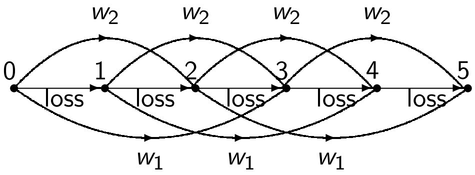
Cutting Stock Problem: arc flow model [VC, 1999]

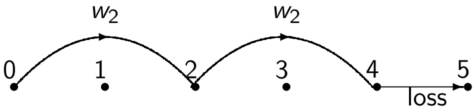
- Rolls of integer capacity W and items of integer size $w_1, \ldots, w_d, \ldots, w_m$.
- Oriented acyclic graph $G = (V, A)$.
- $V = \{0, 1, 2, \ldots, W\}$.
- $A = \{(i,j) : 0 \leq i < j \leq W \text{ and } j - i = w_d,\ d = 1, \ldots, m\}$: length of oriented arc defines size of item.
- Additional arcs $(k, k+1), k = 0, 1, \ldots, W-1$, correspond to loss.
- Valid cutting pattern is a path between vertices 0 and W.

Cutting Stock Problem: arc flow model [VC, 1999]

- Rolls of integer capacity W and items of integer size $w_1, \dots, w_d, \dots, w_m$.
- Oriented acyclic graph $G = (V, A)$.
- $V = \{0, 1, 2, \dots, W\}$.
- $A = \{(i,j) : 0 \leq i < j \leq W \text{ and } j - i = w_d,\ d = 1, \dots, m\}$: length of oriented arc defines size of item.
- Additional arcs $(k, k+1), k = 0, 1, \dots, W-1$, correspond to loss.
- Valid cutting pattern is a path between vertices 0 and W.
- The number of variables is $O(mW)$.

Example: rolls of width $W = 5$, items of sizes 3 and 2





Path corresponds to 2 items of size 2 and 1 unit of loss.

Arc flow model: main ideas

- Flow of one unit from vertex 0 to vertex W corresponds to one cutting pattern.
- Larger flow corresponds to the same cutting pattern in several rolls.
- Flow Decomposition property (graph G is acyclic): any flow can be decomposed in oriented paths connecting the only supply node (node 0) to the only terminal node (node W).
- Solution with integer flows is decomposed into an integer solution for the Cutting Stock Problem.

Arc flow model

Decision variables x_{ij} : flow in arc $(i,j) \equiv$ number of items of size $j-i$ placed in any roll at a distance i of the border of the roll.

$$\min \quad z$$

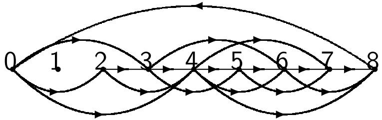
$$\text{subject to} \quad + \sum_{(i,j)\in A} x_{ij} - \sum_{(j,k)\in A} x_{jk} = \begin{cases} -z & , \text{if } j=0 \\ 0 & , \text{if } j=1,\ldots,W-1 \\ z & , \text{if } j=W \end{cases}$$

$$\sum_{(k,k+w_d)\in A} x_{k,k+w_d} \geq b_d \ , \ d=1,2,\ldots,m$$

$$x_{ij} \geq 0 \text{ and integer} \ , \ \forall (i,j) \in A$$

Constraint set 1: flow conservation $\equiv$ valid cutting patterns.
Constraint set 2: sum of flows in arcs of each size $\geq$ demand.
Objective: minimize $z \equiv$ flow between vertex 0 and vertex W.

Arc flow model: example



		x_{04}	x_{48}	x_{03}	x_{36}	x_{47}	x_{02}	x_{24}	x_{35}	x_{46}	x_{57}	x_{68}	z	
node	0	-1		-1			-1						1	$= 0$
	1													$= 0$
	2						1	-1						$= 0$
	3			1	-1				-1					$= 0$
	4	1	-1			-1		1		-1				$= 0$
	5								1		-1			$= 0$
	6				1					1		-1		$= 0$
	7					1					1			$= 0$
	8		1									1	-1	$= 0$
$w_d =$	4	1	1											≥ 5
	3			1	1	1								≥ 4
	2						1	1	1	1	1	1		≥ 8

The loss arcs in the Figure are omitted in the LP model.

Equivalence with Gilmore-Gomory model

Proposition

Arc flow model is equivalent to classical Gilmore-Gomory model.

Proof: applying a DW decomposition to arc flow model gives Gilmore-Gomory model.

- Keep demand constraints in the master problem and flow constraints in the subproblem.
- Each path (cutting pattern) corresponds to an integer solution of the knapsack subproblem.
- Each path is part of a circulation flow (includes the z variable), which is an extreme ray of the subproblem.
- Null solution is the only extreme point.
- Otherwise, there are extreme rays: no convexity constraint.

□

Branch-and-price methodology for CSP

Master Problem: Gilmore-Gomory model + branching constraints based on arc flow variables.

Finding a fractional arc flow variable for branching:

Find arc flows x_{pq} reading Gilmore-Gomory variables:

- Assumption: items in cutting pattern placed by decreasing size.
- Cutting pattern contributes x_j to flow of original variable x_{pq} if there is an item of size $q - p$ beginning at p,
- *i.e.*, value of the flow x_{pq} is given by:

$$x_{pq} = \sum_{j \in \overline{J}} x_j$$

Example: first branching constraint

- Fractional optimal solution of the linear relaxation:

A_j	A_1	A_3	A_4
	4, 4	2, 2, 2, 2	3, 3, 2
$x_j =$	2.5	1.5	2.0

- Flows in arcs: $x_{04} = 2.5$, $x_{48} = 2.5$, $x_{03} = 2.0$, $x_{36} = 2.0$, $x_{02} = 1.5$, $x_{24} = 1.5$, $x_{46} = 1.5$, and $x_{68} = 3.5$.
- First branching constraint: $x_{04} \geq 3$.

Branching scheme

- Branching rule (simple): create 2 branches:

$$x_{ij} \leq \lfloor x_{ij} \rfloor$$

and

$$x_{ij} \geq \lceil x_{ij} \rceil$$

- Variable selection: fractional largest item size, closer to the top border of the roll.
- Search: depth-first search ($\geq$ branch explored first).
- Branching constraint respects to a single arc in position (i, j).
- Branching constraint only affects the cutting patterns with an arc in position (i, j).

Restricted master problem in node w of the search tree

$$
\begin{aligned}
\min \quad & \sum_{j \in J} x_j \\
\text{s. to} \quad & \sum_{j \in J} a_{dj} x_j \geq b_d, \;\; d = 1, 2, \ldots, m \qquad \leftarrow \textbf{GG model} \\
& \sum_{j \in J} \delta_j^l x_j \leq \lfloor x_{ij}^l \rfloor \;, \;\; \forall l \in G^w \\
& \sum_{j \in J} \delta_j^l x_j \geq \lceil x_{ij}^l \rceil \;, \;\; \forall l \in H^w \qquad \leftarrow \textbf{branching constraints} \\
& x_j \geq 0, \;\; \forall j \in J,
\end{aligned}
$$

G^w, H^w : sets of branching constraints of the types $\leq$ and $\geq$, respectively.
x_{ij}^l : the fractional values of flow $0 < x_{ij}^l < b_d$.
$\delta_j^l = 1$, if the arc $(i, i + w_d) \in$ cutting pattern j; or 0, otherwise.

Note: CSP has general integer variables

Most applications have binary variables.

Dual information for the subproblem

- Prize / penalty from a branching constraints of type $\geq$ and $\leq$, respectively, only change reduced cost of one arc in the subproblem.
- In node w, the reduced cost of arc (i,j) is

$$\overline{c}_{ij} = \pi_d - \sum_{l \in G^w_{(i,j)}} \mu_l + \sum_{l \in H^w_{(i,j)}} \nu_l,$$

$G^w_{(i,j)} \subseteq G^w$, $H^w_{(i,j)} \subseteq H^w$: sets of branching constraints on arc (i,j).

- Structure of subproblem remains unchanged during branch-&-price.
- Subproblem is solved using dynamic programming (pseudopolynomial).

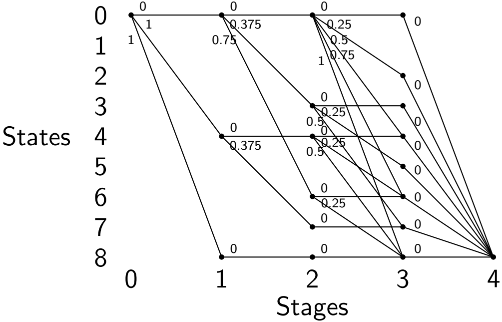
Restricted problem: first node of branch-and-price tree

Insert branching constraint $x_{04} \geq 3$, and reoptimize:

	x_1	x_2	x_3	x_4			dual
$w_d = 4$	2				$\geq$	5	0.0
3		2		2	$\geq$	4	0.375
2			4	1	$\geq$	8	0.25
$x_{04} \geq 3$	1				$\geq$	3	1.0
min	1	1	1	1			
primal	3.0	0.0	1.5	2.0		$z^1 =$	6.5

Dual info: prize of 1 associated to branching constraint $x_{04} \geq 3$..

Subproblem: first node of branch-and-price tree



In stage 0, placing 1 or 2 items has a contribution equal to 1.
First decision: arc (0,4); second decision: arcs (0,4) and (4,8).
Optimal solution: 1 item of size 4 and 2 items of size 2 (value=1.5).

Optimal integer solution

The new column has a 1 in the branching constraint (sum of flows in arc (0,4) across all cutting patterns must be ≥ 3).
After reoptimizing:

	x_1	x_2	x_3	x_4	x_5			dual
$w_d = 4$	2				1	$\geq$	5	0.5
3		2		2		$\geq$	4	0.375
2			4	1	2	$\geq$	8	0.25
$x_{04} \geq 3$	1				1	$\geq$	3	0.0
min	1	1	1	1	1			
primal	2.0	0.0	1.0	2.0	1.0		$z^* =$	6.0

The solution is integer, with a value equal to the LP relaxation.
Optimal solution!

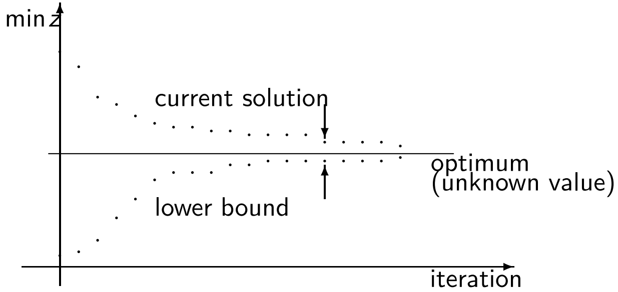
Cutting Stock Problem: some computational results

CSP: problems with 1 large roll width and m =200 item different sizes solved in reasonable time (triplet instances) [VC, 1999].

Multiple lengths CSP: problems with K different large roll widths (instances from literature) [Cláudio Alves, VC, 2008].

	K	m	av. time
	5	100	≈ 1 sec.
	15	25	≈ 1 sec.
Hard instances →	5	100	≈ 30 sec.

Acceleration of column generation



Slow convergence: large changes in the values of the dual variables, which oscillate from one iteration to the next.
Degeneracy: in many iterations, adding new columns to restricted master problem does not improve objective value.

Column generation: dual perspective

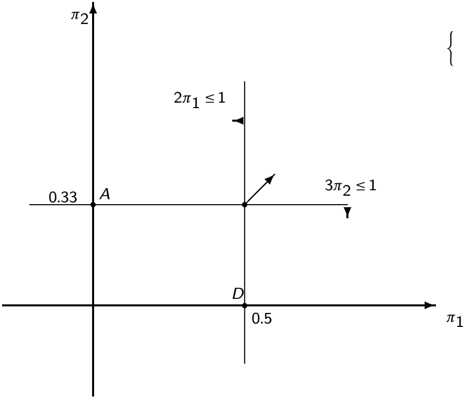
Cutting plane algorithm: adding a column in the primal is equivalent to adding a cut in the dual.

$$(Primal)\quad \begin{array}{ll} \min & cx \\ s.t. & Ax \geq b \\ & x \geq 0 \end{array} \qquad\qquad (Dual)\quad \begin{array}{ll} \max & \pi b \\ s.t. & \pi A \leq c \end{array}$$

CSP Example: rolls of width 10, items of size 4 and 3

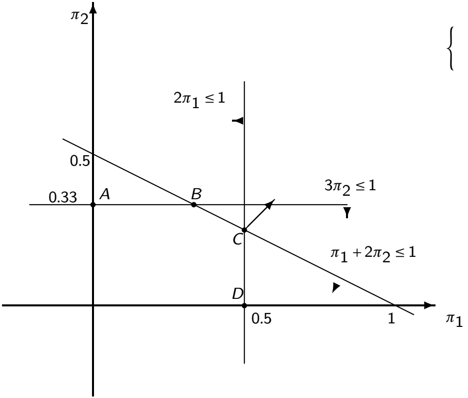
$$(Primal)\quad \begin{array}{lllll} \min & 1x_1 & +1x_2 & +1x_3 & \\ s.t. & 2x_1 & +1x_2 & & \geq b_1 \\ & & +2x_2 & +3x_3 & \geq b_2 \\ & & x_1, x_2, x_3 \geq 0 & & \end{array} \qquad (Dual)\quad \begin{array}{llll} \max & b_1\pi_1 & +b_2\pi_2 & \\ s.t. & 2\pi_1 & & \leq 1 \\ & 1\pi_1 & +2\pi_2 & \leq 1 \\ & & 3\pi_2 & \leq 1 \\ & \pi_1, \pi_2 \geq 0 & & \end{array}$$

Dual space of CSP: first iteration



$$\begin{cases} 2\pi_1 & & \leq 1 \\ & 3\pi_2 & \leq 1 \end{cases}$$

Dual space of CSP: second iteration



$$\begin{cases} 2\pi_1 & & \leq 1 \\ & 3\pi_2 & \leq 1 \\ 1\pi_1 & +2\pi_2 & \leq 1 \end{cases}$$

Acceleration of column generation: motivation

Restricting the dual space may accelerate column generation.
Better convergence: smaller number of attractive columns in subproblem.
Less degeneracy: alternative dual solutions ≡ degenerate primal solutions.

How to do it [VC, 2005]:
Add valid dual cuts to the model before starting column generation.

Dual cuts

$$(P)\quad \begin{array}{ll} \min & cx \\ s.t. & Ax = b \\ & x \geq 0 \end{array} \qquad\qquad (D)\quad \begin{array}{ll} \max & \pi b \\ s.t. & \pi A \leq c \end{array}$$

Adding a set of inequalities to the dual problem, $\pi D \leq d$, we get the extended primal-dual pair:

$$(P^e)\quad \begin{array}{ll} \min & cx + dy \\ s.t. & Ax + Dy = b \\ & x, y \geq 0 \end{array} \qquad\qquad (D^e)\quad \begin{array}{ll} \max & \pi b \\ s.t. & \pi A \leq c \\ & \pi D \leq d \end{array}$$

Usually, restricting the dual ≡ relaxing the primal.
In this case, that does not happen.

A family of valid dual cuts

Proposition

For any width w_i, and a set S of item widths, indexed by s, such that $\sum_{s \in S} w_s \leq w_i$, the dual cuts

$$-\pi_i + \sum_{s \in S} \pi_s \leq 0, \quad \forall i, S,$$

are valid inequalities to the space of optimal solutions of the dual of the cutting stock problem.

Proof.

(contradiction): there would be an attractive cutting pattern. □

Primal point of view: an item of size w_i can be cut, and used to fulfill the demand of smaller orders, provided the sum of their widths is $\leq w_i$.

Example

Combining a cutting pattern and a valid dual cut gives a new cutting pattern.

$W = 100$	A_1		D_1		A_1		A_1^{new}
25	2	+	0	$-->$	2	+	2
10	4		-1		4		0
6	1		1		1		5
3	0		1		0		4
2	2		0		2		2
x_j	0.3		0.8		0.1		0.2

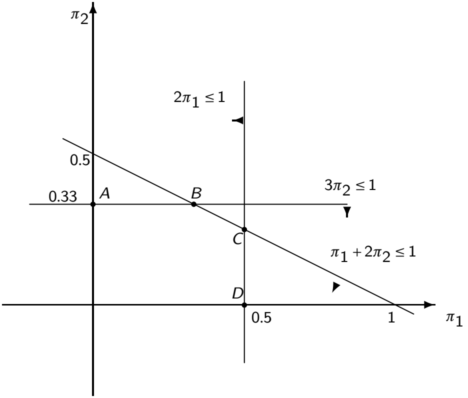
Implementation issues

- Exponential number of cuts of this family.
- Use only cuts from sets S of small cardinality.
- Sets of size 1 and 2 provide a polynomial number $O(m^2)$ of cuts.

Cuts selected:

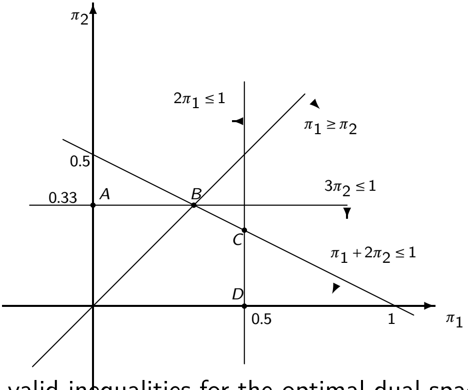
- **Cuts of Type 1:** $-\pi_i + \pi_{i+1} \leq 0, \quad i = 1, 2, \ldots, m-1$
- **Cuts of Type 2:** $-\pi_i + \pi_j + \pi_k \leq 0, \quad \forall i, j, k : w_i \geq w_j + w_k$

Dual space of CSP: rolls of size 10, items of size 4 and 3



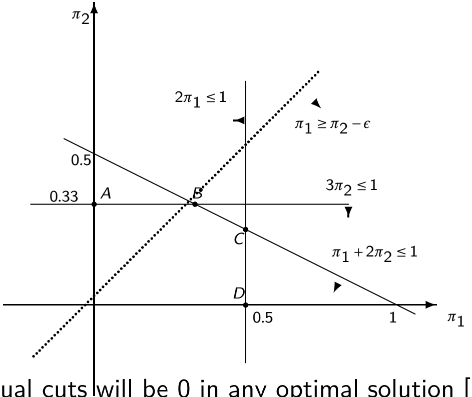
knapsack:
$K = \{(y_1, y_2) : 4y_1 + 3y_2 \leq 10, y_1, y_2 \geq 0 \text{ and integer}\} = \{(2,0),(1,2),(0,3)\}$

Dual space of CSP with cut $\pi_1 \geq \pi_2$



Dual cuts are valid inequalities for the optimal dual space: $\pi_1 \geq \pi_2$ cuts the dual space but obeys all the dual optimal solutions.

Dual space of CSP with cut $\pi_1 \geq \pi_2$ perturbed by ϵ



Columns of dual cuts will be 0 in any optimal solution [Ben Amor, Desrosiers, VC, 2006].

Methodology

- Add dual cuts to model before starting column generation.
- Add starting solution: as suggested by GG, or any other.
- Proceed as usual.

100	dual cuts							GG initial solution				
25	-1				-1			4				
10	1	-1			1	-1			10			
6		1	-1		1	1	-1			16		
3			1	-1		1	1				33	
2				1			1					50

Computational results with dual cuts

Speed-up factor = 4.5 times faster.
Reduction in degenerate pivots: percentage falls from 40% to 8.5%.

Instances [Vance 1993] :
- rolls with widths of 100, 120 or 150
- number of items equal to 200 or 500
- items randomly generated from uniform distribution u(1,100)

Concluding remarks

- Strength of models is of crucial importance.
- Branch-and-price implementation is a mess, but outcome pays for.
- Branch-and-price is very competitive (LP solvers are getting much better) and successful in real world applications.
- Dual cuts make column generation faster keeping models strong.

Some related lines of research at U.Minho - I

Strengthening column generation models with primal cuts

- Primal cuts lead to stronger models.
- After primal cut is inserted, in the subproblem, we must be able to anticipate the coefficient of the column in the primal cut, so that the column is correctly evaluated.
- Some function used for deriving *robust* cuts:
  - Superadditive non-decreasing functions (SANDF).
  - Dual feasible functions (DFF).
  - Chvátal-Gomory cuts (CGC) from arcflow model.

Some related lines of research at U.Minho - II

Dual feasible functions (DFF)

- DFF are valid inequalities for knapsack constraints.
- New ways of deriving stronger dual feasible functions.
- Their use in **dual** space cutting.
- Their use for deriving stronger lower bounds for n-dimensional CSP.

Some related lines of research at U.Minho - III

Dual cuts

- Dual cuts are problem dependent (not so general as stabilization procedures), but do not need adjustments.
- in multiple-length CSP.
- in 2-dimensional guillotine cutting CSP.
- in CSP with other objective functions.
- in (planar) multicommodity flow problems.

Some related lines of research at U.Minho - IV

2-dimensional guillotine cutting for furniture industry

- Branch-and-price algorithms.
- Cutting pattern sequencing for minimization of number of open stacks.
- Heuristic approaches.

References

- G. Dantzig and P. Wolfe, "Decomposition Principle for Linear Programs", Operations Research 8, 101–111, 1960.
- L. Ford and D. Fulkerson, "A suggested computation for maximal multi-commodity network flows", Management Science 5, 1, 97-101, 1958.
- P. Gilmore and R. Gomory, "A Linear Programming Approach to the Cutting Stock Problem", Operations Research 9, 849–859, 1961.
- J.M. Valério de Carvalho, "Exact solution of bin-packing problems using column generation and branch-and-bound", Annals of Operations Research, Vol. 86, pp. 629–659, 1999.
- J.M. Valério de Carvalho, "Using extra dual cuts to accelerate convergence in column generation", INFORMS Journal on Computing, 17, 2, pp. 175-182, 2005.
- Hatem Ben Amor, Jacques Desrosiers, J.M. Valério de Carvalho, "Dual-optimal Inequalities for Stabilized Column Generation", Operations Research 54 (3) 2006, pp. 454–463.
- Hatem Ben Amor, J.M. Valério de Carvalho, "Cutting Stock Problems", in "Column Generation", Guy Desaulniers, Jacques Desrosiers, and Marius M. Solomon (eds.), pp. 131-162, Springer, 2005, ISBN: 0-387-25485-4
- Cláudio Alves, J.M. Valério de Carvalho, "A Stabilized Branch-and-Price-and-Cut Algorithm for the Multiple Length Cutting Stock Problem", Computers and Operations Research, 35, 4, pp. 1315-1328, 2008.