Branch-and-price algorithms (cont.)

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

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

Ciclo di Seminari 'Column Generation'
Metodi e Modelli per l'Ottimizzazione Combinatoria
Corso di Laurea Magistrale in Informatica
Dipartimento di Matematica Pura e Applicata
Università degli Studi di Padova
19^{th} - 28^{th} October 2011

Global outline

- Part I - Decomposition methods
- Part II - Applications
- Part III - Branch-and-price algorithms
- Part IV - Branch-and-price algorithms (cont.)
- Part V - Practical issues, stabilization, accelerating strategies and heuristics
- Bibliography

Part IV

Branch-and-price algorithms (cont.)

Outline

- Branch-and-price: general integer variables
- Arc-flow model
- Application example: cutting stock problem
- Extension to the Multiple lengths cutting stock problem

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 $w_1, \ldots, w_d, \ldots, w_m$.
- Oriented acyclic graph $G = (V, A)$.
- $V = \{0, 1, 2, \ldots, W\}$.

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 $G = (V, A)$.
- $V = \{0, 1, 2, \ldots, W\}$.
- $A = \{(i,j) : 0 \leq i < j \leq W \text{ and } j - i = w_d,\ d = 1, \ldots, m\}$: 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.

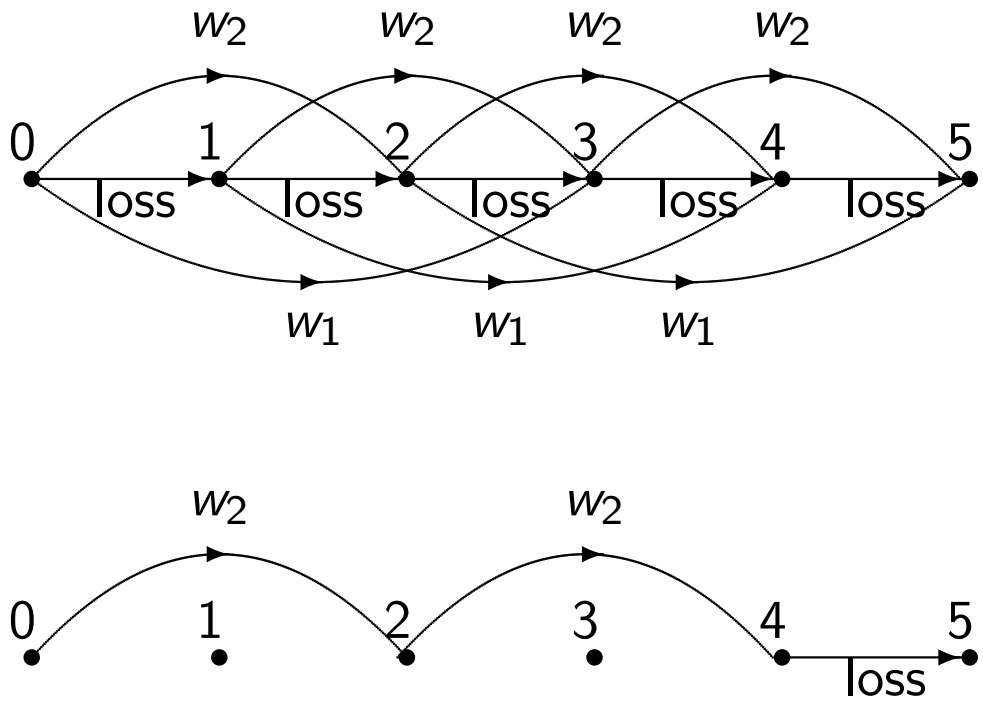
Cutting Stock Problem: arc flow model [VC, 1999]

- 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 $w_1, \ldots, w_d, \ldots, w_m$.
- 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.
- 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.

Ahuja, Magnanti, Orlin'93:

Every nonnegative arc flow can be represented as a path and cycle flow (though not necessarily uniquely) with the following two properties:
i) every oriented path with positive flow connects a deficit node to an excess node.
ii) At most $n+m$ paths and cycles have nonzero flow; out of these, at most m cycles have nonzero flow.

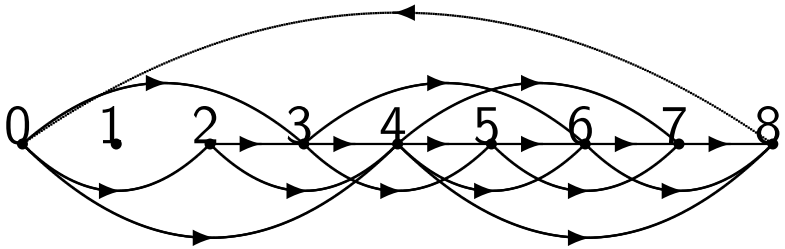
Arc flow model

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.

$$\begin{aligned} \min \quad & z \\ \text{subj. to} \quad & + \sum_{(i,j)\in A} x_{ij} - \sum_{(j,k)\in A} x_{jk} = \begin{cases} -z & , \text{ if } j = 0 \\ 0 & , \text{ if } j = 1,\ldots,W-1 \\ z & , \text{ if } j = W \end{cases} \\ & \sum_{(k,k+w_d)\in A} x_{k,k+w_d} \geq b_d \ , \ d = 1,2,\ldots,m \\ & x_{ij} \geq 0 \ \text{ and integer } \ , \ \forall (i,j) \in A \end{aligned}$$

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.

Symmetry

A model has symmetry when different solutions, in terms of the values of the decision variables, correspond to the same physical solution in the real system.

Example Kantorovich model: decision variable $x_{ij} = 1$, if item j is placed in roll i.

Two solutions: items 1 and 2 are placed in a bin, and the items 3 and 4 in the other:

solution 1: $x_{11} = x_{12} = x_{23} = x_{24} = 1$;

solution 2: $x_{21} = x_{22} = x_{13} = x_{14} = 1$.

correspond to the same packing. □

Example Arc-flow model:

solution 1: $x_{03} = x_{35} = x_{57} = x_{78} = 1$;

solution 2: $x_{02} = x_{24} = x_{47} = x_{78} = 1$.

correspond to the same cutting pattern. □

Reduction of symmetry

Criterion

An arc of size w_e, denoted as $x_{k,k+w_e}$, can only have its tail at a node k that is the head of another arc of size w_d, $x_{k-w_d,k}$, for $w_d \geq w_e$, or, else, at node 0, i.e., the left border of the roll.

If a cutting pattern has loss, it will appear at the end of the roll:

Criterion

Arcs of loss $x_{k,k+1}$ may be set to zero, for $k < w_m$.

In a cutting pattern, the number of consecutive arcs of a given size should be less than or equal to the number of demanded items of that size.

Criterion

Given a node k that is the head of another arc of size w_d $(w_d > w_e)$ or $k = 0$, the only valid arcs of size w_e are those that start at nodes $k + sw_e, s = 0, 1, 2, \ldots, b_e - 1$ and $k + sw_e \leq W$, being b_e the demand of items of size w_e.

Symmetry does not arise in the following solution methodology.

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:

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

Example: first branching constraint

- Fractional optimal solution of the linear relaxation:

A_j	A_1	A_3	A_4
	4	2	3
		2	
	4	2	3
		2	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:

$$x_{ij} \leq \lfloor x_{ij} \rfloor$$
$$\text{and}$$
$$x_{ij} \geq \lceil x_{ij} \rceil$$

- 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

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

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

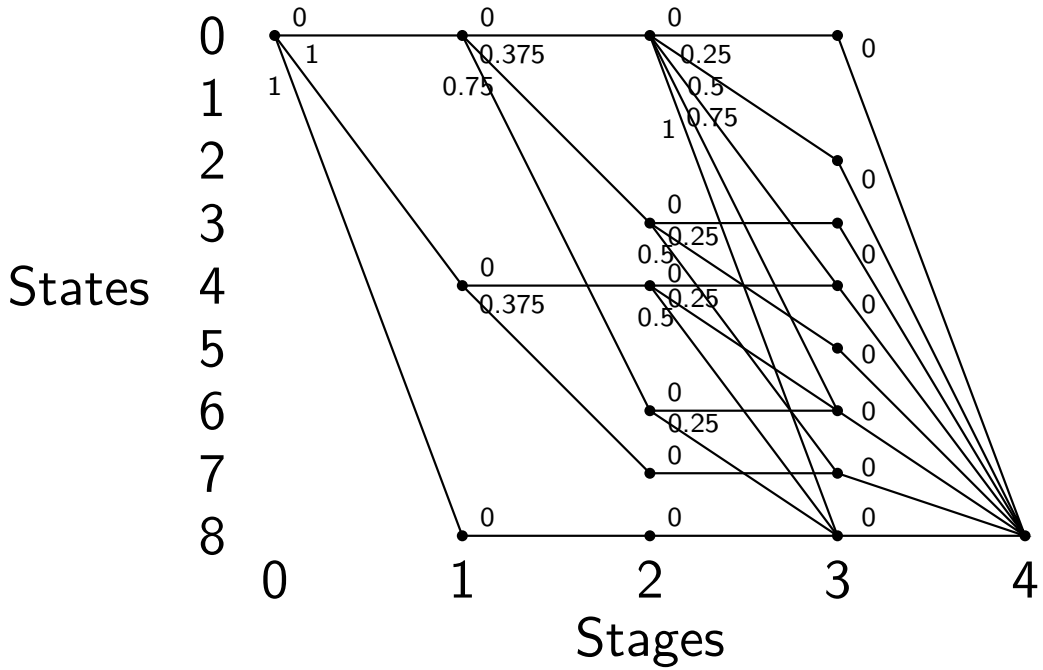
Insert branching constraint $x_{04} \geq 3$, and reoptimize:

	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!

Multiple lengths cutting stock problem (MLCSP)

Multiple stock lengths available, instead of a single roll size.
Proposed by Gilmore-Gomory'63 (machine balance problem).
Many heuristic approaches: Chu,La'2001, Holthaus'2002
Few exact solution approaches: Monaci'2002, Belov,Scheithauer'2002

Equivalent counterpart in bin-packing literature is the Variable sized bin-packing problem (VSBPP).

Multiple lengths cutting stock problem: model

a_{ikr} : number of items of width w_i obtained in stock length k using pattern r.
λ_{kp} : number of times a pattern p from stock length k is cut.
P_k : set of feasible cutting patterns of stock length k, $k = 1, \ldots, K$.

$$\begin{aligned}
\min \quad & \sum_{k=1}^{K} \sum_{p \in P^k} W_k \lambda_{kp} \\
\text{subj. to} \quad & \sum_{k=1}^{K} \sum_{p \in P^k} a_{ikp} \lambda_{kp} \geq b_i, \ i = 1, \ldots, m, \\
& \sum_{p \in P^k} \lambda_{kp} \leq B_k, \ k = 1, \ldots, K, \\
& \lambda_{kp} \geq 0 \text{ and integer}, \ k = 1, \ldots, K, \ p \in P^k.
\end{aligned}$$

constraint set 1: demand constraints.
constraint set 2: availability constraints on each stock length.
(Gilmore-Gomory'63)

Multiple lengths cutting stock problem: example

Stock : availabilities $\mathscr{B} = (B_1, B_2, B_3)$, sizes $\mathscr{W} = (9, 6, 5)$.
Demand: quantities $b = (20, 10, 20)$, sizes $w = (4, 3, 2)$.
Sum of item sizes: $\sum_i w_i b_i = 80 + 30 + 40 = 150$.
Available capacity of stock lengths: $\sum_k W_k B_k = 9B_1 + 6B_2 + 5B_3$.

	λ_{11}	λ_{12}	λ_{13}	λ_{14}	λ_{15}	λ_{16}	λ_{17}	λ_{21}	λ_{22}	λ_{23}	λ_{24}	λ_{31}	λ_{32}	λ_{33}	
$w_i =$ 4	2	1	1					1				1			≥ 20
3		1		3	2	1			2	1			1		≥ 10
2		1	2		1	3	4	1		1	3		1	2	≥ 20
$W_k =$ 9	1	1	1	1	1	1	1								$\leq B_1$
6								1	1	1	1				$\leq B_2$
5												1	1	1	$\leq B_3$
min	9	9	9	9	9	9	9	6	6	6	6	5	5	5	

Arc-flow model (multiple lengths cutting stock problem)

$$\min \sum_{k=1}^{K} W_k z_k$$

subj. to

$$-\sum_{(d,e)\in A'} x_{de} + \sum_{(e,f)\in A'} x_{ef} = \begin{cases} \sum_{k=1}^{K} z_k & , \text{ if } e = 0 \\ -z_k & , \text{ for } e = W_k, k = 1,\ldots,K \\ 0 & , \text{ otherwise} \end{cases}$$

$$\sum_{(d,d+w_i)\in A'} x_{d,d+w_i} \geq b_i \ , \quad \forall i \in I$$

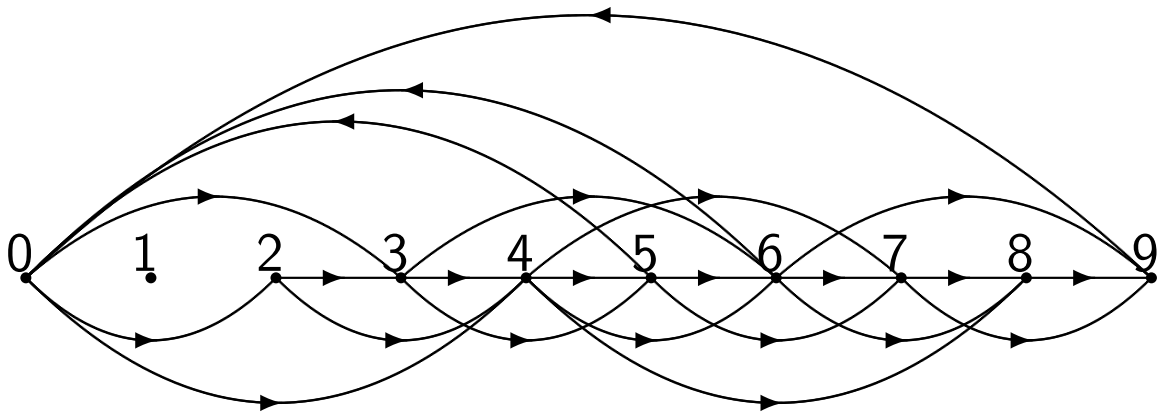
$$z_k \leq B_k \ , \ k = 1,\ldots,K$$

$$x_{de} \geq 0 \text{ and integer} \ , \ \forall (d,e) \in A'$$

$$z_k \geq 0 \text{ and integer} \ , \ k = 1,\ldots,K$$

Example (multiple lengths cutting stock problem)

Stock : availabilities $\mathscr{B} = (B_1, B_2, B_3)$, sizes $\mathscr{W} = (9, 6, 5)$.
Demand: quantities $b = (20, 10, 20)$, sizes $w = (4, 3, 2)$.
Sum of item sizes: $\sum_i w_i b_i = 80 + 30 + 40 = 150$.
Available capacity of stock lengths: $\sum_k W_k B_k = 9B_1 + 6B_2 + 5B_3$.



LP model (multiple lengths cutting stock problem)

		z_1	z_2	z_3	x_{04}	x_{48}	x_{03}	x_{36}	x_{47}	x_{69}	x_{02}	x_{24}	x_{35}	x_{46}	x_{57}	x_{68}	x_{79}	x_{23}	x_{34}	x_{45}	x_{56}	x_{67}	x_{78}	x_{89}		
node	0	1	1	1	-1		-1				-1														=	0
	1																								=	0
	2										1	-1						-1							=	0
	3						1	-1					-1					1	-1						=	0
	4				1	-1			-1			1		-1					1	-1					=	0
	5		-1										1		-1					1	-1				=	0
	6	-1						1		-1				1		-1					1	-1			=	0
	7								1						1		-1					1	-1		=	0
	8					1										1							1	-1	=	0
	9	-1								1							1							1	=	0
$w_i =$	4				1	1																			≥	20
	3						1	1	1	1															≥	10
	2										1	1	1	1	1	1	1								≥	20
$W_k =$	9	1																							≤	B_1
	6		1																						≤	B_2
	5			1																					≤	B_3
	min	9	6	5																						

Lower bounds z_{IP}^w for MLCSP at node w of branching tree

z_{LP}^w : LP bound at node w of branch-and-price tree.
Any MLCSP solution uses a combination of stock rolls with integer lengths.
Stronger integer lower bound z_{IP}^w for node w : smallest integer combination ($\geq z_{LP}^w$) of modified list of stock lengths available.
Modified list of stock lengths available (branching constraints on z arcs):
i) $z_k \geq l_k^w$: remove l_k^w rolls of width W_k from list, and consider them separately.
ii) $z_k \leq u_k^w$: use list of u_k^w rolls instead of B_k, otherwise $u_k^w = B_k$.

$$
\begin{aligned}
z_{IP}^w = \min \quad & \sum_{k=1}^{K} W_k y_k + \sum_{k=1}^{K} W_k l_k^w \\
\text{subj. to} \quad & \sum_{k=1}^{K} W_k y_k \geq \lceil z_{LP}^w \rceil - \sum_{k=1}^{K} W_k l_k^w, \\
& y_k \leq u_k^w - l_k^w, \ k = 1, \ldots, K, \\
& y_k \geq 0 \text{ andinteger}, \ k = 1, \ldots, K.
\end{aligned}
$$

Level Cuts for MLCSP

Use z_{IP}^{w} to enforce "Level Cuts" in one of the following ways:

1. trim loss has to appear somewhere in the cutting plan:

$$\sum_{k=1}^{K} \sum_{p \in P_k} \left(W_k - \sum_{i=1}^{m} w_i a_{ikp} \right) \lambda_{kp} \geq z_{IP}^{w} - \sum_{i=1}^{m} w_i b_i.$$

2. use, at least, z_{IP}^{w} length of rolls:

$$\sum_{k=1}^{K} \sum_{p \in P_k} W_k \lambda_{kp} \geq z_{IP}^{w}.$$

Dual information easily transferable to subproblem.

Feasibility cuts for MLCSP [Vanderbeck'99]

m cuts are derived, one for each item size.
Only feasibility cuts that depend on a single item size: dual info variables easily reported to the pricing subproblems.
If the availability of the largest roll $k = 1$ is enough to cut all the items of size w_i, for $i = 1, \ldots, m$, we will have

$$\sum_{k=1}^{K} \sum_{p \in P_k : a_{ikp} > 0} \lambda_{kp} \geq \left\lceil \frac{b_i}{\left\lfloor \frac{W_1}{w_i} \right\rfloor} \right\rceil .$$

Otherwise, if $\left\lceil \frac{b_i}{\left\lfloor \frac{W_1}{w_i} \right\rfloor} \right\rceil > B_1$, and the availability of stock rolls with length W_2 is enough to cut the remaining items,

$$\sum_{k=1}^{K} \sum_{p \in P_k : a_{ikp} > 0} \lambda_{kp} \geq B_1 + \left\lceil \frac{b_i - B_1 \left\lfloor \frac{W_1}{w_i} \right\rfloor}{\left\lfloor \frac{W_2}{w_i} \right\rfloor} \right\rceil ,$$

and so forth.

Implementation issues (MLCSP)

Branching scheme

- Branch first on larger fractional z arcs.
- Use branching information to improve lower bounds (Level cuts).
- Then, branch on x_{de} arcs.

Subproblem

- Single dynamic programming recursion solves subproblems for all lengths.

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	$\approx$ 1 sec.
	15	25	$\approx$ 1 sec.
Hard instances $\rightarrow$	5	100	$\approx$ 30 sec.

MLCSP - Computational results: summary

Branch-and-price-and-cut with simple rounding heuristic compares favorably to other approaches (results with dual cuts shown in Part V):

	K	m	av. time	
largest group inst. (Monaci)	5	100	$\approx$ 4 sec.	
our instances	15	25	$\approx$ 1 sec.	
hard instances (Belov) $\rightarrow$	5	100	$\approx$ 30 sec.	(for 45 instances solved; 5 unsolved in time limit: 900 sec. optimality gap $\leq$ 0.0025%)

K : different large roll widths (Alves'PhD2005, Alves,VC'2006).

300 instances (Monaci'2002 - Combinatorial enumeration)
Monaci solved 78% of the instances (time limit: 900 seconds).
50 hard instances (Belov'2002 - Column generation with Chvátal-Gomory cutting planes and elaborate heuristic)
Belov's approach solves less instances and takes more time (85 sec. vs. 30 sec.)

Concluding remarks

- Some heuristics provide good results for CSP; often optimal solutions.
- Heuristics do not solve optimally the MLCSP so easily.
- (GG + arc-flow) methodology is better than arc-flow solely, even with dynamic constraint generation (less symmetry and smaller size of the LP basis).