



**UNIVERSITY AT BUFFALO**

*State University of New York*

***THE SCHOOL OF ENGINEERING  
AND APPLIED SCIENCES***

---

# Branch and Price

Column Generation for Solving Huge Integer Programs

Mohan Akella  
Sharad Gupta  
Avijit Sarkar



# Introduction to Branch & Price

---

- **Branch & Cut** algorithms modify the basic Branch & Bound strategy by attempting to strengthen the linear programming relaxation (LPR) of an IP with new inequalities before branching a fractional solution.
- Basically, Branch & Cut = Branch & Bound + Cutting Planes (in other words, Row Generation)
- **Branch & Price** similar to Branch & Cut, except that procedure focuses on **Column Generation** rather than **Row Generation**.
- In fact, Pricing and Cutting are complementary procedures for tightening a LP relaxation.

- Branch & Price integrates Branch & Bound and Column Generation methods for solving large-scale IPs .
- At each node of the Branch & Bound tree, columns may be generated to tighten (improve) the LP relaxation.
- In Branch & Price, sets of columns are left out of the LP relaxation of large IPs because there are too many columns to handle efficiently and most of them will have their associated variables equal to zero in an optimal solution anyway.
- Then to check optimality, a sub-problem, also called the “*pricing problem*” is solved to identify columns to enter the basis.
- If such columns are found, the LP is reoptimized.
- Branching occurs when no columns “price” out to enter the basis and the LP solution does not satisfy integrality conditions.



# Branch & Price : Application Areas

---

- Generalized Assignment Problem (GAP) (discussed in detail later)
- Airline Crew Scheduling : finding a minimum cost assignment of flight crews to a given flight schedule such that certain constraints are satisfied.
- Optimal shift scheduling : with multiple rest breaks, meal breaks etc.
- Multi-commodity Flow Problems
- Routing Problems



# GAP : Definition & Problem Formulation

---

Definition: Given 'n' jobs and 'm' machines, objective is to find a maximum profit assignment of jobs to machines, such that each job is assigned to precisely one machine subject to capacity restrictions on the machines.

Problem Formulation:

$$\max \sum_{i=1}^m \sum_{j=1}^n p_{ij} x_{ij}$$

*s . t .*

$$\sum_{i=1}^m x_{ij} = 1, \forall j = 1(1)n$$

$$\sum_{j=1}^n w_{ij} x_{ij} \leq c_i \forall i = 1(1)m$$

$$x_{ij} \in \{0,1\}, i = 1(1)m, j = 1(1)n$$



# Problem Formulation-Contd.

---

$p_{ij}$  = profit associated with assigning job 'j' to machine 'i'

$$x_{ij} = \begin{cases} 1, & \text{if job 'j' is assigned to machine 'i'} \\ 0, & \text{otherwise} \end{cases}$$

$w_{ij}$  = claim on the capacity of machine 'i' by job 'j'

$c_i$  = capacity of machine 'i'



# GAP : Problem Reformulation

Let  $K_i = \{ x_1^i, x_2^i, \dots, x_{k_i}^i \}$  be the set of possible feasible assignments of jobs to machine 'i', i.e.  $x_k^i = \{ x_{1k}^i, x_{2k}^i, \dots, x_{nk}^i \}$  is a feasible solution to

$$\sum_{j=1}^n w_{ij} x_{jk}^i \leq c_i$$

Let  $y_k^i = \begin{cases} 1, & \text{if feasible assignment } x_k^i \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$

The GAP can now be reformulated as follows : called "Master Problem"

$$\max \sum_{i=1}^m \sum_{k=1}^{k_i} \left( \sum_{j=1}^n p_{ij} x_{ij} \right) y_k^i$$

$$s.t. \sum_{i=1}^m \sum_{k=1}^{k_i} x_{jk}^i y_k^i = 1, \forall j = 1(1)n$$

$$\sum_{k=1}^{k_i} y_k^i \leq 1, \forall i = 1(1)m$$

$$y_k^i \in \{0,1\}, i = 1(1)m, k \in k_i$$



# Points to be noted in the Master Problem

---

- What is the advantage of reformulating the problem ?
- The reformulated problem is essentially obtained by applying Dantzig-Wolfe decomposition to the original formulation.
- The reformulated problem sub-divides the original problem into a Master Problem and a sub-problem.
- The knapsack constraints have been placed in the sub-problem.
- A column  $y_k^i$  in the master problem represents a feasible assignment of jobs to machine 'i'.
- Master Problem cannot be solved directly due to exponential number of columns.
- Restricted Master Problem (RMP) : Master Problem that considers only a subset of the columns.

# UB Sub-Problem (Pricing Problem)

Additional columns can be generated for the RMP by solving the following two sub-problems :

$$\max_{1 \leq i \leq m} \{ z(KP_i) - v_i \}$$

where:  $v_i$  = dual associated with convexity constraint of machine 'i'  
 $z(KP_i)$  = optimal solution to the following knapsack problem

$$\max \sum_{j=1}^n (p_{ij} - u_j) x_j^i$$

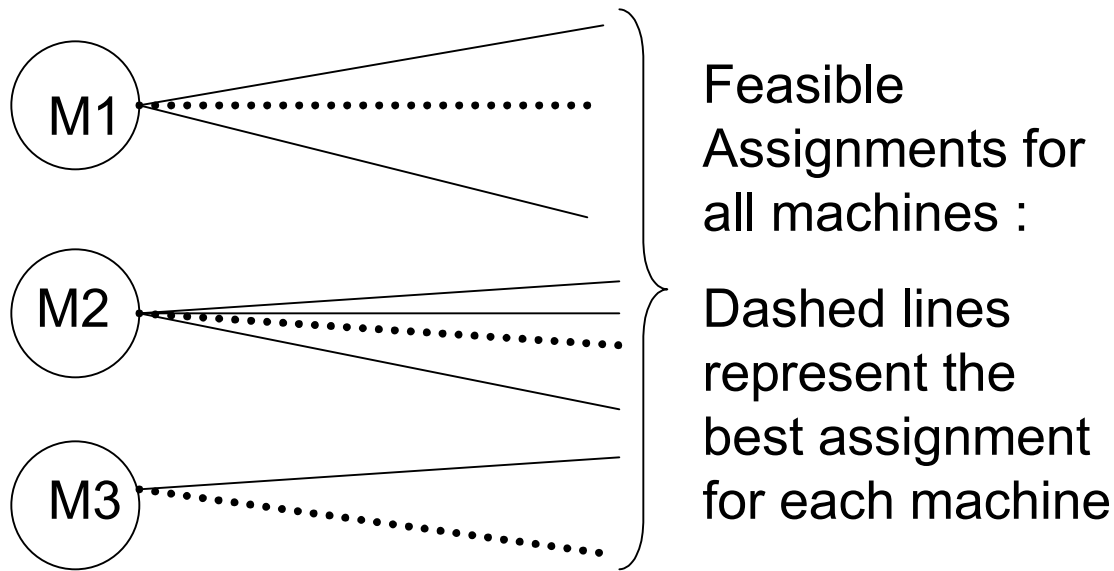
$$s.t. \sum_{j=1}^n w_{ij} x_j^i \leq c_i$$

$$x_j^i \in \{0,1\}, \quad \forall j = 1(1)n$$

$u_j$  = optimal dual price from the solution to the RMP associated with the partitioning constraint for job 'j'

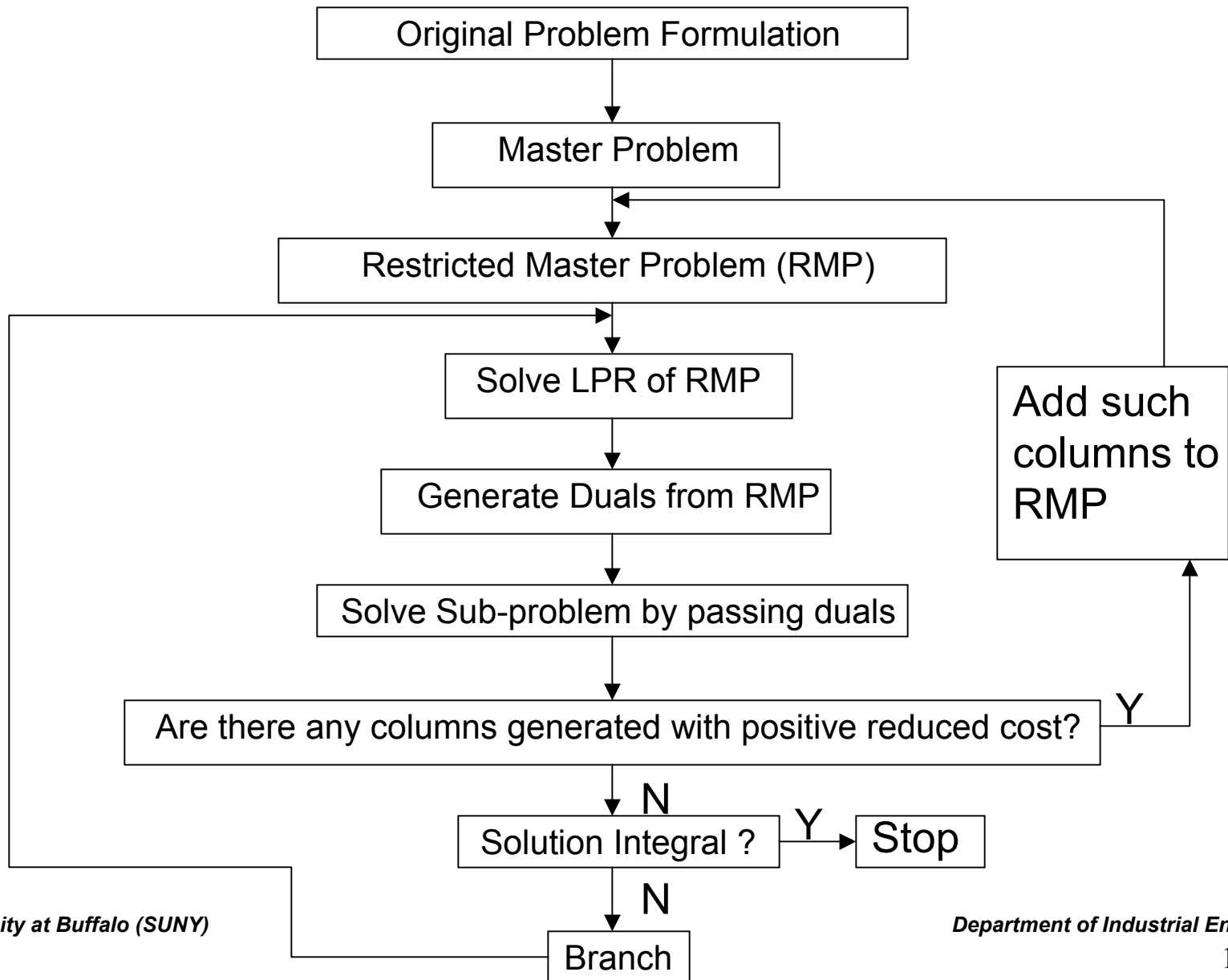
# UB Explanation of Sub-problems

➤ The second sub-problem generates the best feasible assignment for each machine 'i'.



➤ Now the objective is to search for the best assignment over all machines, which precisely is done by the first sub-problem.

$$\max_{1 \leq i \leq m} \{ z ( KP_i ) - v_i \}$$



- Column Generation always works in the feasible domain. Why?
- Column Generation sub-problem generates column with highest reduced cost.
- During the column generation process, the RMP keeps growing.
- A column prices out favorably to enter the basis, if its reduced cost is positive.
- It is not necessary to solve the sub-problem to optimality, any column with a positive reduced cost can enter.
- Consequently, if the objective function value of the column generation sub-problem is less than or equal to zero, then the current optimal solution for the RMP is also optimal for the Master Problem.

- If a solution to the Master problem is fractional, then the corresponding solution to the standard formulation is also fractional.
- Every branch in the standard formulation has an equivalent branch in the Master problem.
- Fixing a variable in the standard formulation corresponds to fixing a set of variables in the Master problem.
- In the standard formulation, fixing  $x_{ij}$  to zero prohibits job 'j' to be assigned to machine 'i'. In the Master problem, this can be achieved as follows : if  $x_{jk}^i = 1$  , then  $y_k^i = 0$  for all  $k \in K_i$ .
- Conversely, want job 'j' to be assigned to machine 'i' : if  $x_{jk}^i = 0$  , then  $y_k^i = 0$  for all  $k \in K_i$  and for some  $l \neq i$ , if  $x_{jk}^l = 1$ , then  $y_k^l = 0$  for  $1 \leq l \neq i \leq m$  and  $k \in K_l$ .

# UB Computational Results for a GAP(Savelsbergh)

		Branch-and-Bound		Branch-and-Price	
Problem	n/m	# nodes	CPU	# nodes	CPU
1	10	15	0.11	5	4.21
2	6	84	0.47	5	1.80
3	3	168	1.61	3	0.86
4	2	556	6.65	2	0.55
5	10	83	1.16	5	11.09
6	5	333	7.23	6	6.31
7	2.5	979	19.93	3	2.34
8	1.5	770	3.7	3	0.44

✓MINTO 2.0 / CPLEX 3.0

✓IBM / RS6000 model 590



# Branch & Price : Conclusion

---

- Extremely versatile for solving large-scale IPs.
- Procedure integrates Branch & Bound with Column Generation.
- Largely problem specific.

Finally .....

Branch-and-Price.....a nice name

which hides a well-known process relatively **EASY TO APPLY**

.....*Desrosiers*

***Next step : Presenting***

***Branch-and-Price-and-Cut = Branch-and-Bound + CG + Cutting Planes***

- A Branch-and-Price Algorithm for the Generalized Assignment Problem – M. Savelsbergh, *OR Vol 45. No.6, 1997*
- Branch and Price : Column Generation for Solving Huge Integer Programs – Barnhart et. al. *OR Vol 46. No.3, 1998*
- Using Branch-and-Price-and-Cut to solve Origin-Destination Integer Multi-commodity Flow problems – Vance et. al. *OR Vol 48. No.2, 2000*
- On Dantzig-Wolfe Decomposition in Integer Programming and ways to perform branching in a Branch-and-Price Algorithm – F. Vanderbeck  
*OR Vol 48. No.1, 2000*
- Airline Crew Scheduling : A New Formulation and Decomposition Algorithm – Vance, Barnhart et. al. *OR Vol 48. No.2, 2000*
- A Heuristic Branch-and-Price Approach for the Airline Crew Pairing Problem – Vance et. al.

## Branch-and-Price Example

Let us consider the Generalized Assignment Problem (GAP) with  $m = 2$  machines and  $n = 3$  jobs.

$p_{ij}$  = profit associated with assigning job  $j$  to machine  $i$ .

The  $p_{ij}$  matrix is given below:

Job( $j$ )			
	1	2	3
Machine $i = 1$	5	7	3
Machine $i = 2$	2	10	5

The capacities of the machines ( $C_i$ ) are given below:

$i$	$C_i$
1	5
2	8

The amount of capacity of machine  $i$  taken up by each job  $j$  is also tabulated below:

Job( $j$ )			
	1	2	3
Machine $i = 1$	3	4	2
Machine $i = 2$	4	3	4

$k_i$  denotes the set of possible feasible assignments to machine  $i$ .

Thus,  $k_1 = (1, 0, 0), (0, 1, 0), (0, 0, 1), (1, 0, 1)$

and  $k_2 = (1, 0, 0), (0, 1, 0), (0, 0, 1), (1, 1, 0), (0, 1, 1)$

The **Master Problem** is now formulated as follows:

$$\mathbf{max} \ z = 5y_1^1 + 7y_2^1 + 3y_3^1 + 8y_4^1 + 2y_1^2 + 10y_2^2 + 5y_2^3 + 12y_2^4 + 15y_2^5$$

*s.t.*

$$y_1^1 + 0 + 0 + y_4^1 + y_1^2 + 0 + 0 + y_2^4 + 0 = 1 \quad (u_1)$$

$$0 + y_1^2 + 0 + 0 + 0 + y_2^2 + 0 + y_2^4 + y_2^5 = 1 \quad (u_2)$$

$$0 + 0 + y_1^3 + y_4^1 + 0 + 0 + y_2^3 + 0 + y_2^5 = 1 \quad (u_3)$$

$$y_1^1 + y_1^2 + y_1^3 + y_4^1 \leq 1 \quad (v_1)$$

$$y_1^2 + y_2^2 + y_2^3 + y_2^4 + y_2^5 \leq 1 \quad (v_2)$$

where  $u_j$  and  $v_i$  denote the duals associated with job  $j$  and machine  $i$  respectively.

Let us arbitrarily choose columns  $y_1^1$  and  $y_2^5$ , which is a feasible solution. Now, the Restricted Master Problem (RMP) is as follows:

$$\mathbf{max} \ z = 5y_1^1 + 15y_2^5$$

s.t

$$y_1^1 + 0 = 1 \quad (u_1)$$

$$0 + y_2^5 = 1 \quad (u_2)$$

$$0 + y_2^5 = 1$$

$$y_1^1 + 0 = 1$$

$$0 + y_2^5 = 1$$

The last 3 constraints are redundant, hence their corresponding duals are 0.

Now

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (1)$$

Therefore

$$B^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2)$$

$$c_B B^{-1} = \begin{bmatrix} 5 & 15 \end{bmatrix} \quad (3)$$

Hence  $u_1 = 5, u_2 = 15, v_1 = 0, v_2 = 0$

**Subproblem for Machine 1**

$$\mathbf{max}(5 - 5)x_1^1 + (7 - 15)x_1^2 + (3 - 0)x_3^1$$

$$\text{s.t. } 3x_1^1 + 4x_1^2 + 2x_3^1 \leq 5$$

Optimal solutions are  $(1, 0, 1)$  and  $(0, 0, 1)$ , each with  $z = 3$

$$\text{Hence } z(KP_1) - v_1 = 3 - 0 = 3$$

**Subproblem for Machine 2**

$$\mathbf{max}(2 - 5)x_1^2 + (10 - 15)x_2^2 + (5 - 0)x_3^2$$

$$\text{s.t. } 5x_1^2 + 3x_2^2 + 4x_3^2 \leq 8$$

Optimal solution is  $(0, 0, 1)$  (i.e. sequence 3 of machine 2) with  $z = 5$

$$\text{Hence } z(KP_2) - v_2 = 5 - 0 = 5$$

Since the reduced cost for machine 2 is higher, sequence 3 of machine 2 ( $y_3^2$ ) is the column generated.

Now the new RMP is:

$$\mathbf{max} \ z = 5y_1^1 + 15y_2^5 + 5y_3^2$$

s.t

$$y_1^1 + 0 + 0 = 1 \quad (u_1)$$

$$0 + y_5^2 + 0 = 1 \quad (u_2)$$

$$0 + y_5^2 + y_3^2 = 1 \quad (u_3)$$

$$y_1^1 + 0 + 0 = 1$$

$$0 + y_3^2 + y_5^2 = 1$$

The last 2 constraints are redundant, hence their corresponding duals are 0.

Now

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Therefore

$$B^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$c_B B^{-1} = [ 5 \quad 15 \quad 5 ] \quad (6)$$

Hence  $u_1 = 5, u_2 = 15, u_3 = 5, v_1 = 0, v_2 = 0$

**Subproblem for Machine 1**

$$\mathbf{max}(5 - 5)x_1^1 + (7 - 15)x_1^2 + (3 - 5)x_3^1$$

$$\text{s.t. } 3x_1^1 + 4x_1^2 + 2x_3^1 \leq 5$$

Optimal solutions are  $(1, 0, 0)$  and  $(0, 0, 0)$ , each with  $z = 0$

$$\text{Hence } z(KP_1) - v_1 = 0 - 0 = 0$$

**Subproblem for Machine 2**

$$\mathbf{max}(2 - 5)x_1^2 + (10 - 15)x_2^2 + (5 - 5)x_3^2$$

$$\text{s.t. } 5x_1^2 + 3x_2^2 + 4x_3^2 \leq 8$$

Optimal solutions are  $(0, 0, 1)$  and  $(0, 0, 0)$  each with  $z = 0$

$$\text{Hence } z(KP_2) - v_2 = 0 - 0 = 0$$

Therefore, reduced costs for all columns are 0. Hence the solution  $y_1^1 = 1, y_5^2 = 1, y_3^2 = 0$  is optimal.

**Optimal assignments are:**

(1,0,0) for machine 1, i.e. job 1 is assigned to machine 1.

(0,1,1) for machine 2, i.e. jobs 2 and 3 are assigned to machine 2.

$$z_{optimal} = 5 + 10 + 5 = 20$$