

*AIRO 2003: Venezia 2-5 September 2003*

Alberto Ceselli, Giovanni Righini

Università degli Studi di Milano

DTI - Dipartimento di Tecnologie dell'Informazione



A branch-and-price algorithm for the multilevel  
generalized assignment problem

# Outline of the talk:

1. Problem formulations.
2. A branch-and-price algorithm.
3. Implementation details.
4. Computational results.

# The multilevel generalized assignment problem

## ● Formulation:

MGAP)

$$\min v = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ijk} x_{ijk}$$

s.t.

$$\left\{ \begin{array}{lll} \sum_{i \in I} \sum_{k \in K} x_{ijk} = 1 & \forall j \in J & \text{Partitioning} \\ \sum_{j \in J} \sum_{k \in K} a_{ijk} x_{ijk} \leq b_i & \forall i \in I & \text{Capacity} \\ x_{ijk} \in \{0,1\} & \forall i \in I, \forall j \in J, \forall k \in K & \text{Integrality} \end{array} \right.$$

agents	$i \in I = \{1 \dots M\}$
tasks	$j \in J = \{1 \dots N\}$
levels	$k \in K = \{1 \dots L\}$

# State of the art

Approximation algorithms:

- Tabu Search  
(Laguna M., Kelly J.P., Gonzàles-Velarde J.L., Glover F.,  
EJOR 82 - 1995)
- Heuristics (French A.P., Wilson J.M., Journal of Heuristics 8 - 2002)

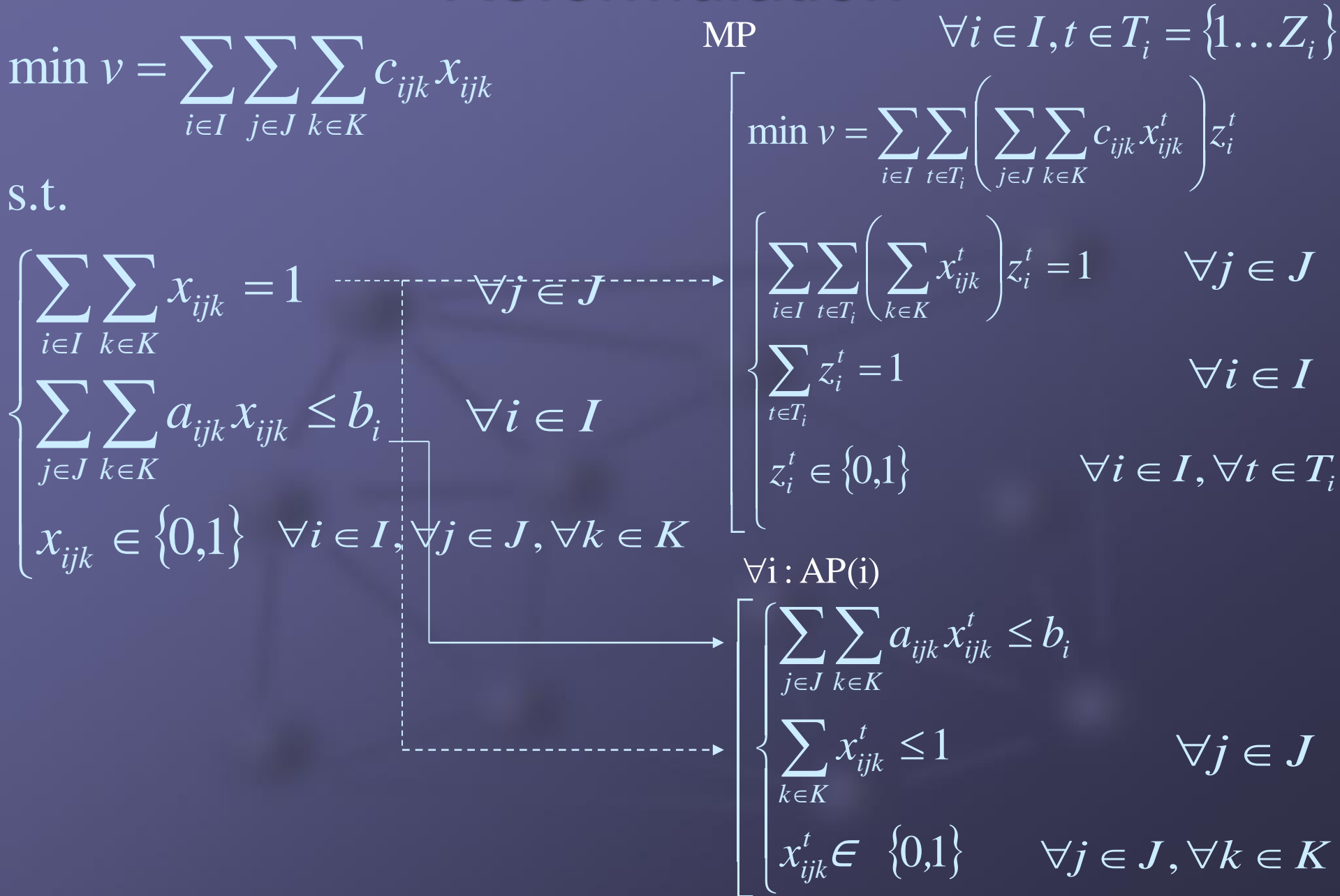
Exact Methods:

- (General purpose MIP solver)
- Logic Cuts (Osorio M.A., Laguna M., to appear on EJOR - 2003)

We propose:

- Branch-and-price algorithm

# Reformulation



# Master Problem - Relaxation

MP

$$\min v = \sum_{i \in I} \sum_{t \in T_i} \left( \sum_{j \in J} \sum_{k \in K} c_{ijk} x_{ijk}^t \right) z_i^t$$

$$\left\{ \begin{array}{l} \sum_{i \in I} \sum_{t \in T_i} \left( \sum_{k \in K} x_{ijk}^t \right) z_i^t = 1 \quad \forall j \in J \\ \sum_{t \in T_i} z_i^t = 1 \quad \forall i \in I \\ z_i^t \in \{0,1\} \quad \forall i \in I, \forall t \in T_i \end{array} \right.$$

L-MP

$$\min v = \sum_{i \in I} \sum_{t \in T_i} \left( \sum_{j \in J} \sum_{k \in K} c_{ijk} x_{ijk}^t \right) z_i^t$$

$$\left\{ \begin{array}{l} \sum_{i \in I} \sum_{t \in T_i} \left( \sum_{k \in K} x_{ijk}^t \right) z_i^t \geq 1 \quad \forall j \in J \quad (\lambda_j) \\ - \sum_{t \in T_i} z_i^t \geq -1 \quad \forall i \in I \quad (\mu_i) \\ z_i^t \in \mathbb{R}_+ \quad \forall i \in I, \forall t \in T_i \end{array} \right.$$

↓  
L-RMPs

# Pricing problem

$$\forall i : \text{PP}(i)$$

$$\min \bar{c}_i^t = \sum_{j \in J} \sum_{k \in K} c_{ijk} x_{ijk}^t - \sum_{j \in J} \lambda_j \left( \sum_{k \in K} x_{ijk}^t \right) + \mu_i$$

s.t.

$$\begin{cases} \sum_{j \in J} \sum_{k \in K} a_{ijk} x_{ijk}^t \leq b_i \\ \sum_{k \in K} x_{ijk}^t \leq 1 \\ x_{ijk}^t \in \{0,1\} \end{cases}$$

$$\forall j \in J$$

$$\forall j \in J, \forall k \in K$$

$$\max \tau_i = \sum_{j \in J} \sum_{k \in K} (\lambda_j - c_{ijk}) x_{ijk}^t$$

$$\begin{cases} \sum_{j \in J} \sum_{k \in K} a_{ijk} x_{ijk}^t \leq b_i \\ \sum_{k \in K} x_{ijk}^t \leq 1 \\ x_{ijk}^t \in \{0,1\} \end{cases}$$

$$\forall j \in J$$

$$\forall j \in J, \forall k \in K$$

$$\begin{pmatrix} \sum_{k \in K} x_{i1k}^t \\ \vdots \\ \sum_{k \in K} x_{iNk}^t \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

i →

# Primal Bound 1





# Primal Bound 1

## ● Step 1: coefficient definition

$$f_{ij} = \sum_{t \in T_i} \sum_{k \in K} x_{ijk}^t z_i^t$$

“Desirability” of assigning task  $j$  to agent  $i$

## ● Step 2: task-agent allocation

- Assign each task  $j$  to the agent  $i^*$  for which  $f_{ij}$  is maximum:

$$\forall j \longrightarrow i^* = \arg \max_{i \in I} \{f_{ij}\}$$

# Primal Bound 1

## ● Step 3: task-level selection

- For each agent, solve a MCKP considering only its assigned tasks
- If no feasible solution exists, then FAIL

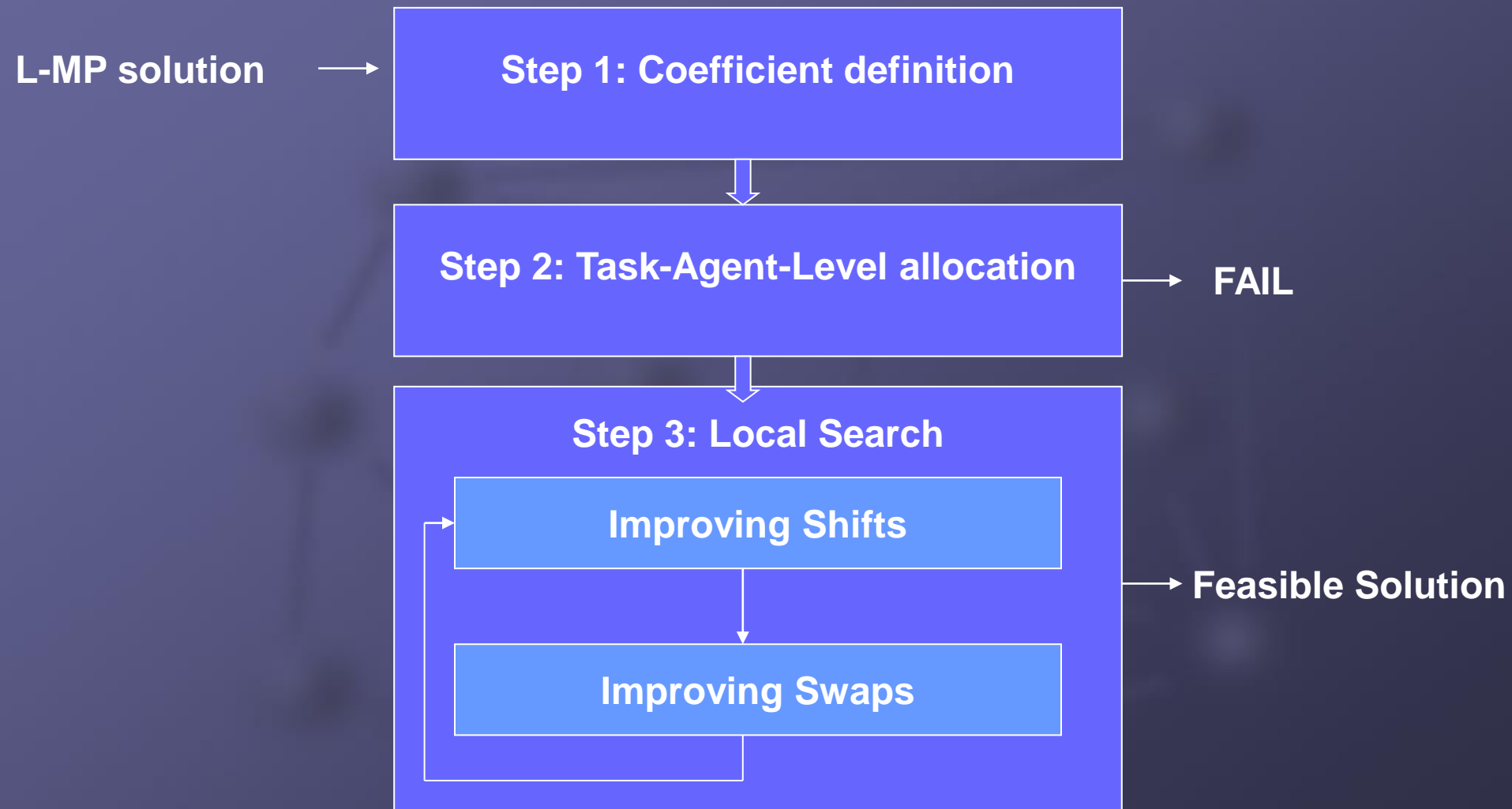
$$O(NL(M + \sum_{i \in I} b_i))$$

## ● Step 4: Local search

- Improving shifts: (selected in a greedy way)

$$O(NML)$$

# Primal Bound 2



# Primal Bound 2

- Step 1: Coefficient definition

$$g_{ijk} = \sum_{t \in T_i} x_{ijk}^t z_i^t \quad \text{“Desirability” of assigning task } j \text{ to agent } i \text{ at level } k$$

- Step 2: Task-agent-level allocation  
(French and Wilson MGAPH2)

- Repeat

- For each unassigned task  $j$ , compute a regret value:

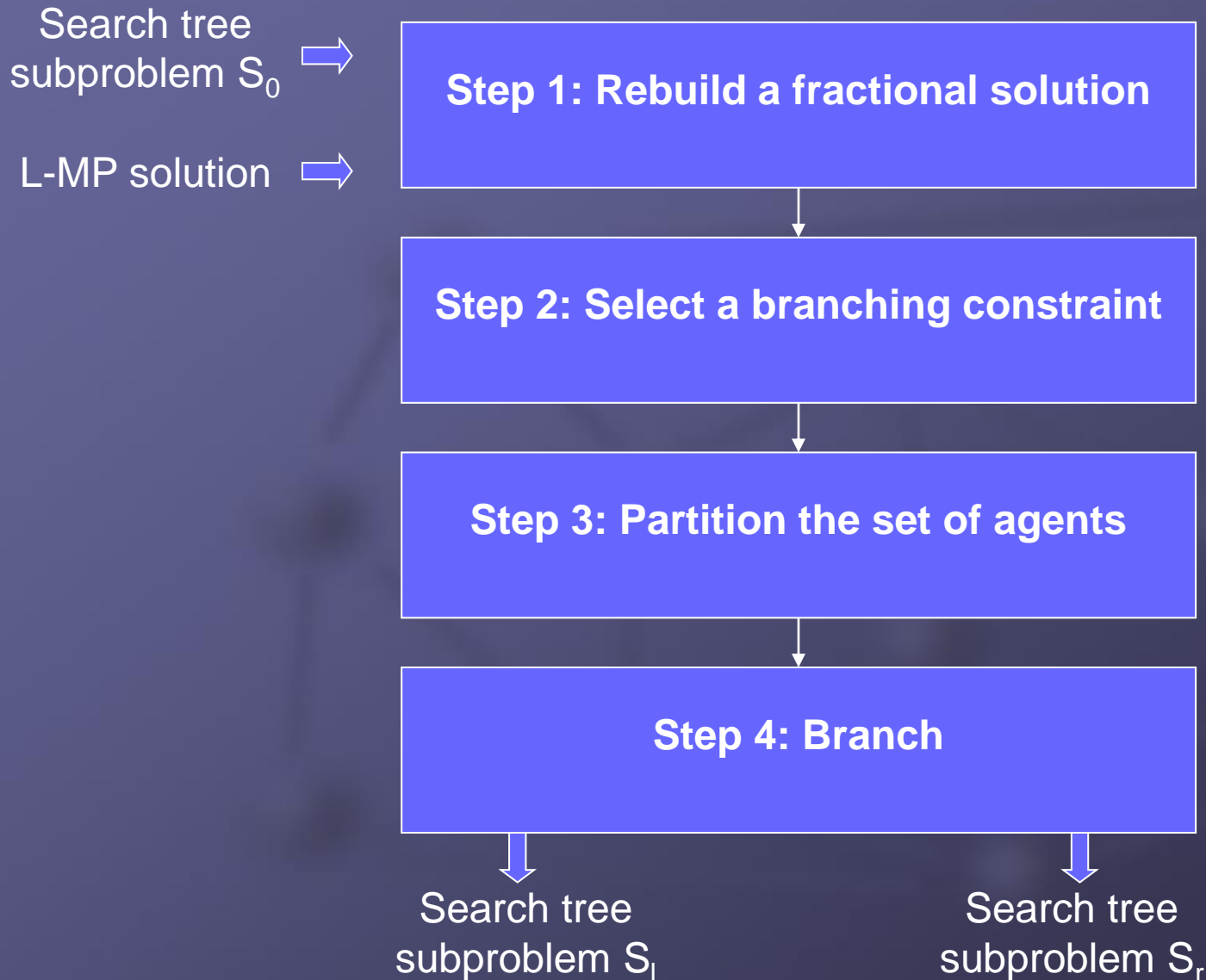
$$r_j = g_{i_j j k_j^1} - g_{i_j j k_j^2}$$

- Select the task  $j^*$  with highest regret value
- Make the most “desirable” agent-level assignment for  $j^*$
- Until all tasks are assigned or infeasibility is met
- $O(N ML \log ML)$

# Primal Bound 2

- Step 3a: Local Search – Improving shifts
  - Perform all improving shifts in a greedy way
  - If any change is made, go to step 3b else STOP (a local optimum is reached)
  - $O(NML)$
- Step 3b: Local search – Improving swaps
  - Evaluate all improving swaps
  - Perform only the best swap, then repeat evaluation  $O(LN^2)$
  - When no more improving swaps can be made, go to step 3a

# Branching Scheme



# Branching Scheme

- Step 1: Rebuild a fractional solution for the original formulation:

$$f_{ij} = \sum_{t \in T_i} \sum_{k \in K} x_{ijk}^t z_i^t$$

- Step 2: Let

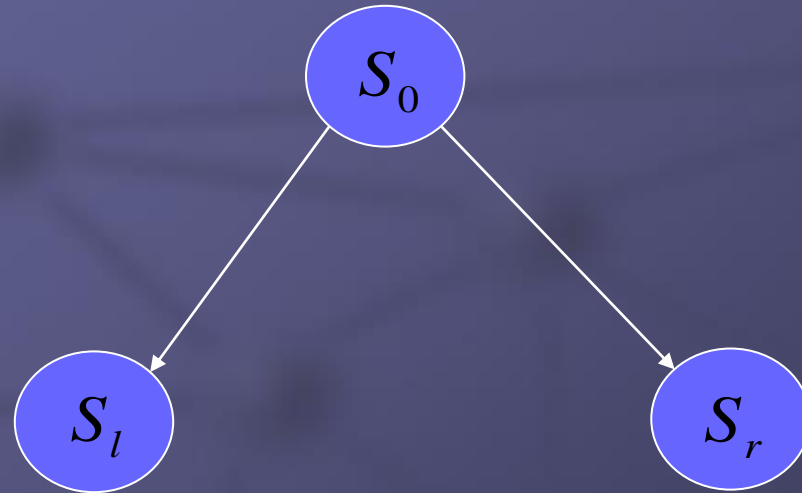
- $H_j$  be the set of agents to which task  $j$  is assigned in a fractional solution:  
 $H_j = \{i \mid f_{ij} > 0\}$
- Select the task  $j^*$  with the highest  $|H_j|$  value

- Step 3:

- Partition  $H_{j^*}$  in two subsets  $H_{j^*}^l$  and  $H_{j^*}^r$ , with  $|H_{j^*}^l| \approx |H_{j^*}^r|$   
(... and use an heuristic method to obtain a balanced partition)
- Let  $R_j$  be the set of agents to which task  $j$  is not assigned in a fractional solution:  
 $R_j = \{i \mid f_{ij} = 0\}$
- Partition  $R_{j^*}$  in two subsets  $R_{j^*}^l$  and  $R_{j^*}^r$ , with  $|R_{j^*}^l| \approx |R_{j^*}^r|$

# Branching Scheme (2)

- Step 4: Binary branch on the partitioning constraint  $j^*$ , imposing



$$\sum_{i \in H_{j^*}^l \cup R_{j^*}^l} \sum_{k \in K} x_{ij^*k} = 0$$

$$\sum_{i \in H_{j^*}^r \cup R_{j^*}^r} \sum_{k \in K} x_{ij^*k} = 0$$

Best first search strategy: first on the dual bound, then on the number of fixed variables



# Implementation details

- Preprocessing
- Fathoming - tailing off control
- Column pools management
- RMP initialisation

# Preprocessing

- For each agent  $i$ , for each task  $j$ , check if there are two efficiency levels  $k'$  and  $k''$  such that

$$c_{ijk'} \geq c_{ijk''} \wedge a_{ijk'} \geq a_{ijk''}$$

- Assignment  $(i, j, k'')$  *dominates*  $(i, j, k')$ : variable  $x_{ijk'}$  can be fixed to 0

# Fathoming – tailing off (1)

## ● Relaxation

- Task-agent semi-assignment constraints

$$\begin{array}{l}
 \min v = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ijk} x_{ijk} \\
 \text{s.t.} \\
 \left\{ \begin{array}{l} \sum_{i \in I} \sum_{k \in K} x_{ijk} \geq 1 \\ \sum_{j \in J} \sum_{k \in K} a_{ijk} x_{ijk} \leq b_i \\ \sum_{k \in K} x_{ijk} \leq 1 \\ x_{ijk} \in \{0,1\} \end{array} \right. \quad \forall i \in I, \forall j \in J, \forall k \in K
 \end{array}
 \quad \xrightarrow{\text{LR)}} \quad
 \begin{array}{l}
 \min v = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ijk} x_{ijk} + \sum_{j \in J} \lambda_j \left( 1 - \sum_{i \in I} \sum_{k \in K} x_{ijk} \right) \\
 \text{s.t.} \quad (\forall i) \\
 \left\{ \begin{array}{l} \sum_{j \in J} \sum_{k \in K} a_{ijk} x_{ijk} \leq b_i \\ \sum_{k \in K} x_{ijk} \leq 1 \\ x_{ijk} \in \{0,1\} \end{array} \right. \quad \forall j \in J, \forall i \in I, \forall k \in K
 \end{array}$$

Redundant in the original formulation

# Fathoming – tailing off (2)

Lagrangean relaxation: LR(i)

$$\min v_i = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} (c_{ijk} - \lambda_j) x_{ijk}$$

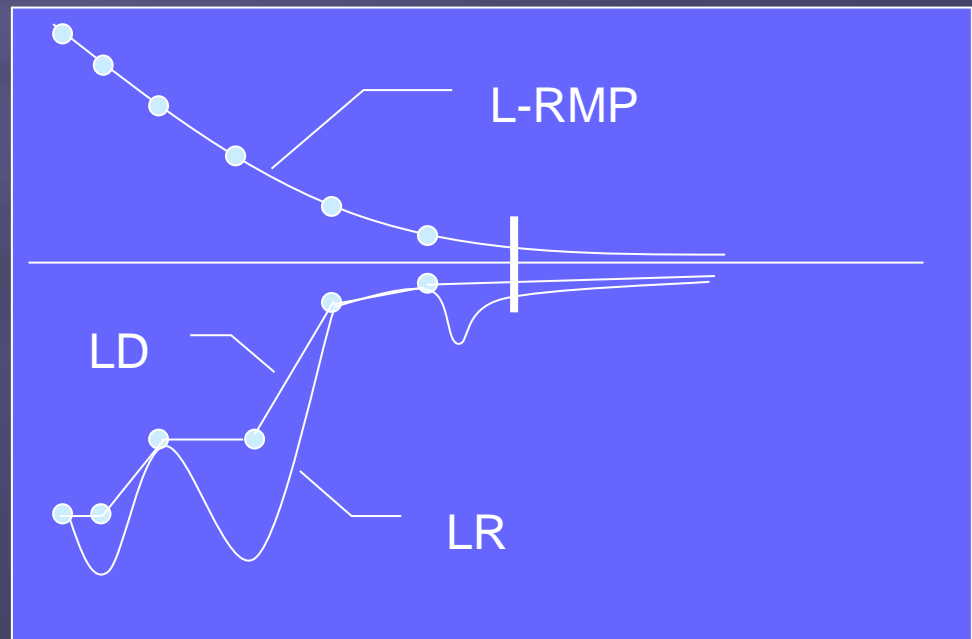
$$\begin{cases} \sum_{j \in J} \sum_{k \in K} a_{ijk} x_{ijk} \leq b_i \\ \sum_{k \in K} x_{ijk} \leq 1 & \forall j \\ x_{ijk} \in \{0,1\} & \forall j, \forall k \end{cases}$$

Pricing Problem: PP(i)

$$\min \tau_i = \sum_{j \in J} \sum_{k \in K} (c_{ijk} - \lambda_j) x_{ijk}^t$$

$$\begin{cases} \sum_{j \in J} \sum_{k \in K} a_{ijk} x_{ijk}^t \leq b_i \\ \sum_{k \in K} x_{ijk}^t \leq 1 & \forall j \\ x_{ijk}^t \in \{0,1\} & \forall j, \forall k \end{cases}$$

Dual Bound evaluation

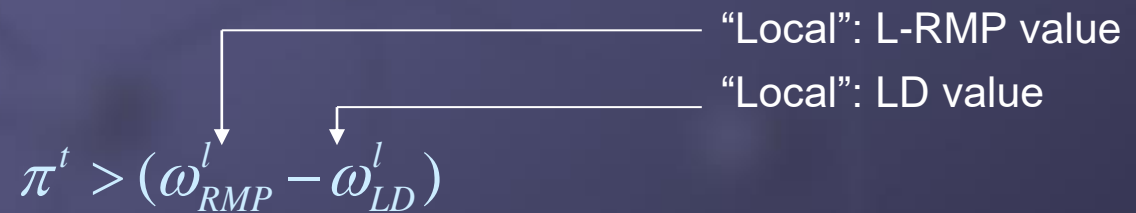


# Column management

- “All Negative” pricing strategy
- RMP remotion (limit of 3000 columns reached): at each step I
  - Red remotion



- Yellow remotion



- Green remotion

$$\pi^t > \left( \frac{\omega_{RMP}^l - \omega_{LD}^l}{M} \right)$$

- M different pools
- RMP reinsertion
- Pool deletion (6 consecutive inspections)

# RMP-Initialisation

- 1 Dummy column (worst solution)

$$\underbrace{(1, \dots, 1)}_N, \underbrace{(0, \dots, 0)}_M)^T$$

- Root Node: Random Columns (10 random sets)

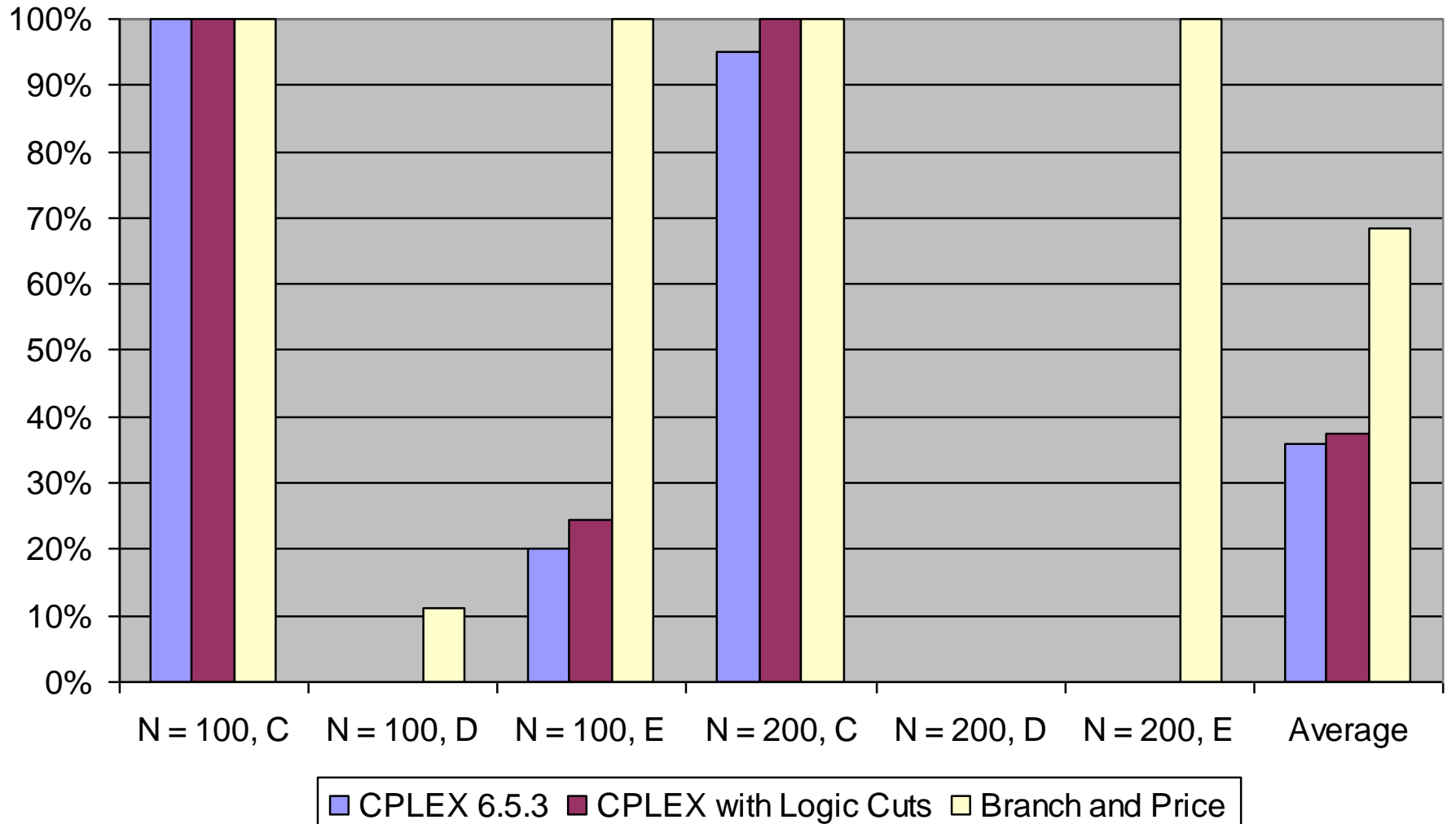
- Search tree nodes:

- Feasible columns from the most recent optimisation
- Columns from the pool  
(using optimal dual variables of the ancestor)

# Computational results

- C++
- ILOG CPLEX 6.5.3
- Intel Pentium IV 1.6 GHz, 512MB RAM
- Linux RedHat 7.3
- gcc/g++ 2.96 , -O3 optimisations
  
- 215 randomly generated instances (from 3000 to 30000 binary variables)
  
- Correlation type: C, D, E ( $p = 1.0$ ,  $p = 0.8$ )
  
- $N = 100$ ;  $M = 10, 20, 30$ ;  $L = 3, 4, 5$
- $N = 200$ ;  $M = 15, 30$ ;  $L = 4, 5$
  
- Time limit: 2 hours
- Memory overflow

# Comparison of methods % problems solved





# Comparison of methods N = 100

	M	N	L	CPLEX 6.5.3			CPLEX 6.5.3 + Logic Cuts			Branch and Price		
				time	# opt	slack (%)	time	# opt	slack (%)	time	# opt	slack (%)
C	10	100	3	3,6	5	0,00	6,6	5	0,00	6,3	5	0,00
	10	100	4	8,2	5	0,00	11,0	5	0,00	12,4	5	0,00
	10	100	5	5,4	5	0,00	7,5	5	0,00	8,5	5	0,00
	20	100	3	4,5	5	0,00	7,9	5	0,00	2,8	5	0,00
	20	100	4	6,2	5	0,00	8,0	5	0,00	3,4	5	0,00
	20	100	5	12,6	5	0,00	15,0	5	0,00	2,6	5	0,00
	30	100	3	5,9	5	0,00	7,0	5	0,00	2,3	5	0,00
	30	100	4	8,1	5	0,00	13,5	5	0,00	2,7	5	0,00
	30	100	5	9,9	5	0,00	11,4	5	0,00	1,7	5	0,00
				7,2	45	0,00	9,8	45	0,00	4,7	45	0,00
D	10	100	3	-	0	0,64	-	0	0,58	4699,4	2	0,03
	10	100	4	-	0	0,57	-	0	0,50	2231,4	1	0,07
	10	100	5	-	0	0,66	-	0	0,52	560,8	2	0,06
	20	100	3	-	0	1,48	-	0	1,32	-	0	0,71
	20	100	4	-	0	1,33	-	0	1,30	-	0	0,52
	20	100	5	-	0	1,28	-	0	0,99	-	0	0,43
	30	100	3	-	0	1,87	-	0	1,68	-	0	0,89
	30	100	4	-	0	1,86	-	0	1,68	-	0	0,54
	30	100	5	-	0	1,51	-	0	1,46	-	0	0,51
				-	0	1,24	-	0	1,12	2497,2	5	0,42
E	10	100	3	791,1	2	0,14	3820,3	4	0,14	63,8	5	0,00
	10	100	4	521,7	3	0,06	3191,3	3	0,08	11,2	5	0,00
	10	100	5	466,0	4	0,05	2278,7	4	0,04	15,7	5	0,00
	20	100	3	-	0	1,77	-	0	1,49	118,2	5	0,00
	20	100	4	-	0	1,52	-	0	1,71	100,5	5	0,00
	20	100	5	-	0	1,58	-	0	2,60	95,8	5	0,00
	30	100	3	-	0	3,90	-	0	4,02	952,6	5	0,00
	30	100	4	-	0	2,97	-	0	3,57	600,0	5	0,00
	30	100	5	-	0	3,18	-	0	3,04	481,1	5	0,00
				592,9	9	1,69	3096,8	11	1,86	271,0	45	0,00
				300,0	54	0,98	1553,3	56	0,99	924,3	95	0,14

# Comparison of methods N = 200

	M	N	L	CPLEX 6.5.3			CPLEX 6.5.3 + Logic Cuts			Branch and Price		
				time	# opt	slack (%)	time	# opt	slack (%)	time	# opt	slack (%)
C	15	200	4	91,9	5	0,00	93,9	5	0,00	137,9	5	0,00
	15	200	5	82,6	5	0,00	131,5	5	0,00	326,2	5	0,00
	30	200	4	273,3	5	0,00	385,8	5	0,00	27,0	5	0,00
	30	200	5	243,1	4	0,39	586,5	5	0,00	17,2	5	0,00
				172,7	19	0,10	299,4	20	0,00	127,1	20	0,00
D	15	200	4	-	0	0,63	-	0	0,54	-	0	0,08
	15	200	5	-	0	0,60	-	0	0,48	-	0	0,08
	30	200	4	-	0	0,91	-	0	0,99	-	0	0,56
	30	200	5	-	0	0,80	-	0	0,83	-	0	0,25
				-	0	0,73	-	0	0,71	-	0	0,24
E p 0.8	15	200	4	-	0	0,17	-	0	0,37	1336,3	5	0,00
	15	200	5	-	0	0,17	-	0	0,64	679,9	5	0,00
	30	200	4	-	0	1,17	-	0	2,15	1966,7	5	0,00
	30	200	5	-	0	1,10	-	0	2,26	350,8	5	0,00
				-	0	0,65	-	0	1,36	1083,4	20	0,00
E p 1.0	15	200	4	-	0	0,25	-	0	0,85	700,5	5	0,00
	15	200	5	-	0	0,24	-	0	0,89	1165,3	5	0,00
	30	200	4	-	0	1,06	-	0	2,11	367,0	5	0,00
	30	200	5	-	0	1,17	-	0	1,97	1578,0	5	0,00
				-	0	0,68	-	0	1,45	952,7	20	0,00
				172,7	19	0,54	299,4	20	0,88	721,1	60	0,06

# Quality of the relaxation

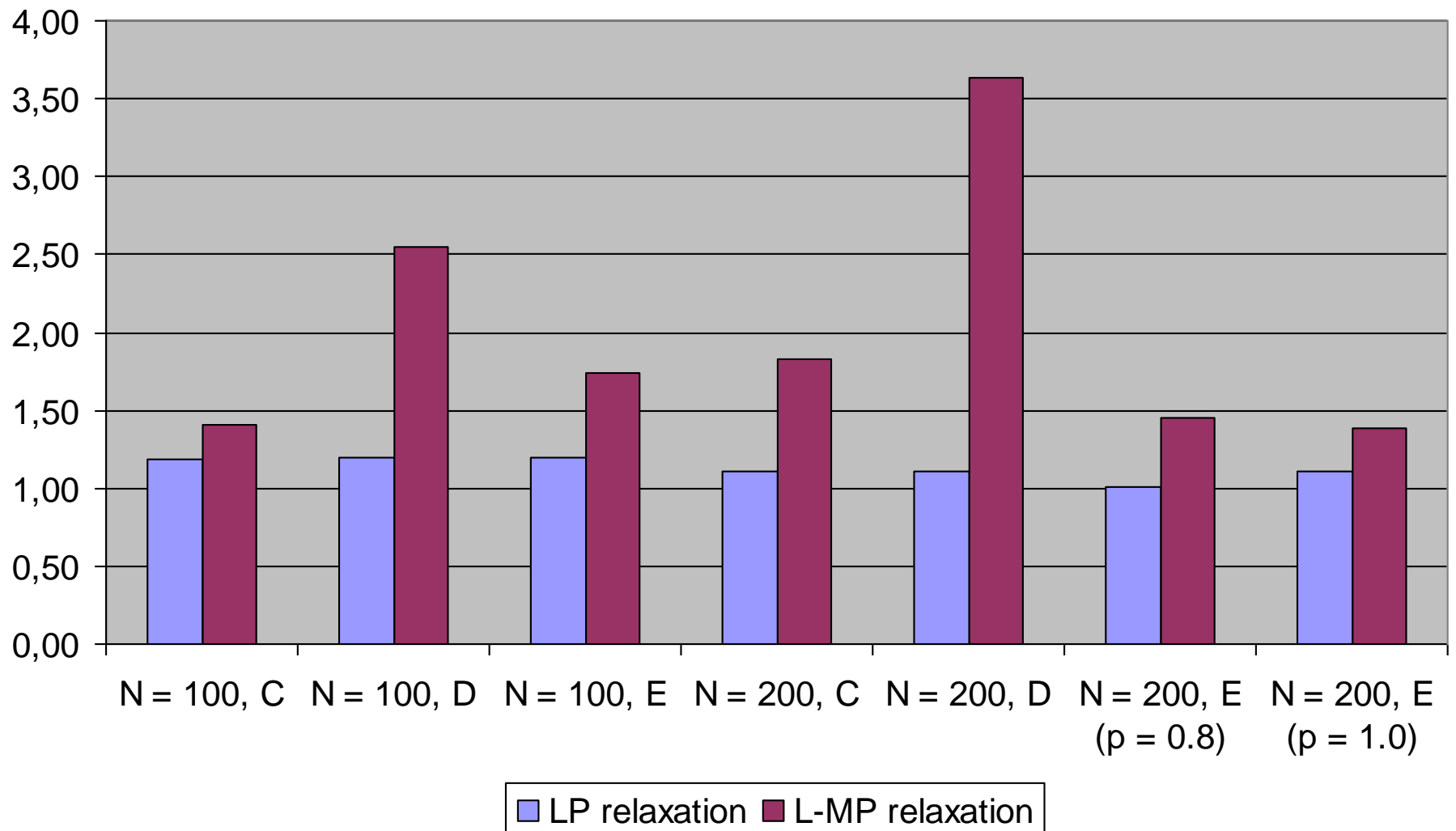
N = 100, Correlation C D E (p 1.0), 5 random instances each combination

		LP relaxation			L-MP relaxation		
		Int. Gap (%)	Time (s)	Fract.	Int. Gap (%)	Time (s)	Fract.
Levels	3	1,94	0,16	1,19	0,30	3,91	1,82
	4	1,44	0,22	1,20	0,27	4,24	1,85
	5	0,95	0,30	1,19	0,21	4,00	2,03
Agents	10	0,81	0,09	1,10	0,16	7,81	1,45
	20	1,61	0,23	1,19	0,31	2,58	1,99
	30	1,91	0,36	1,29	0,32	1,76	2,27

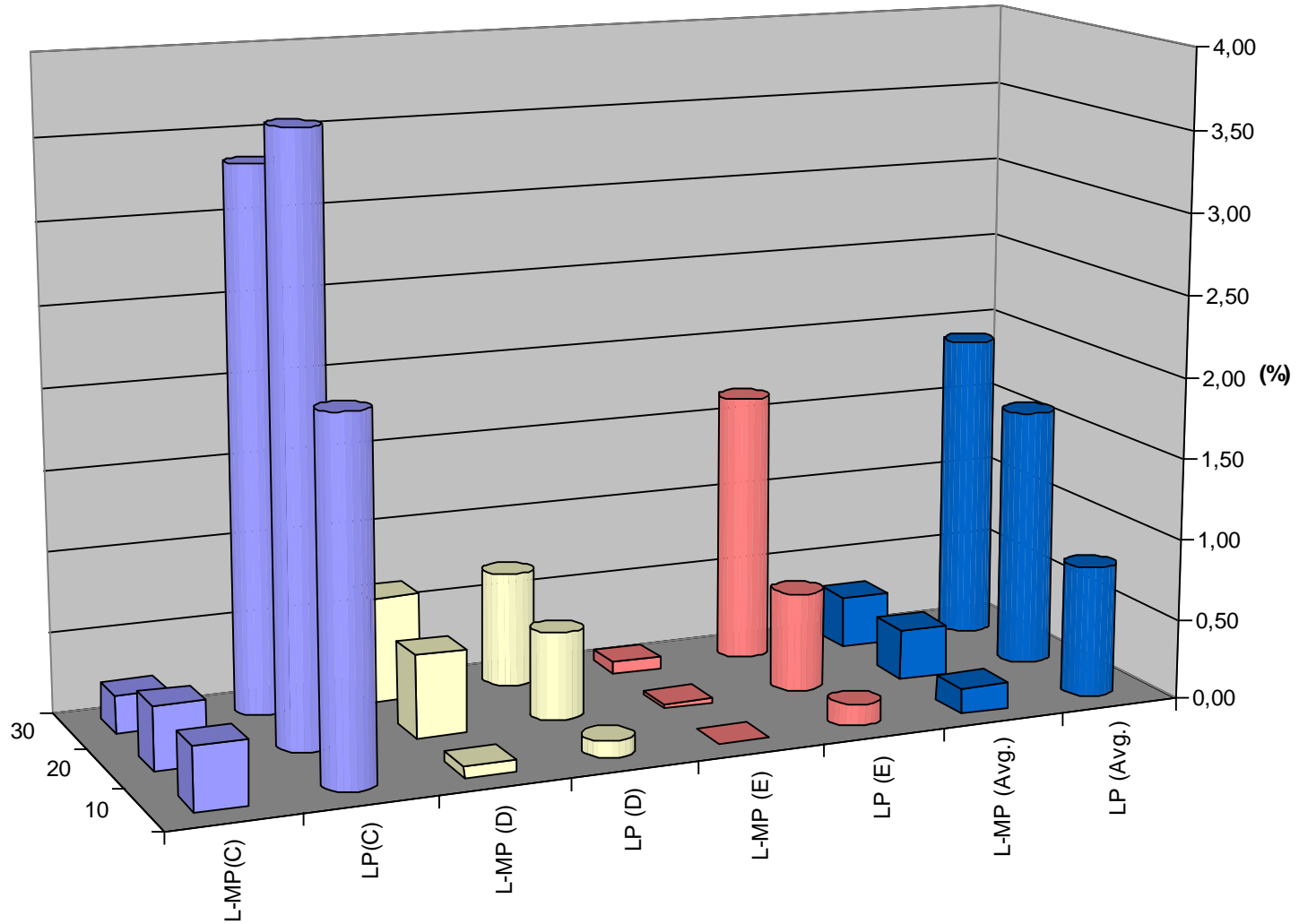
N = 200, Correlation C D E (p 1.0 and p 0.8), 5 random instances each combination

		LP-relaxation			L-MP relaxation		
		Int. Gap (%)	Time (s)	Fract.	Int. Gap (%)	Time (s)	Fract.
Levels	4	0,41	0,81	1,08	0,19	102,25	2,15
	5	0,32	1,17	1,08	0,13	114,84	2,46
Agents	15	0,41	0,49	1,05	0,12	184,39	1,80
	30	0,32	1,50	1,10	0,20	32,70	2,81

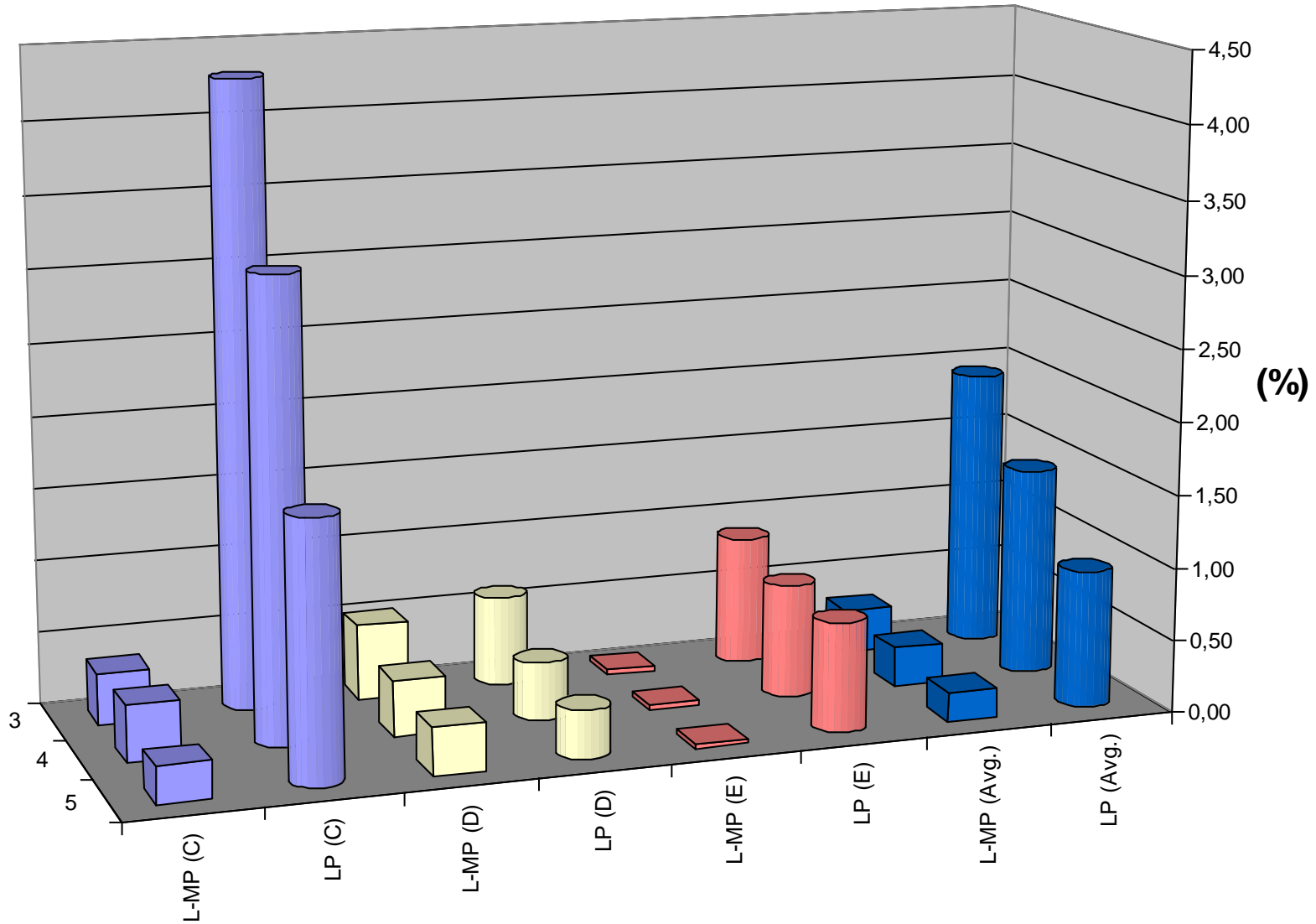
# (Avg.) Number of fractional assignments



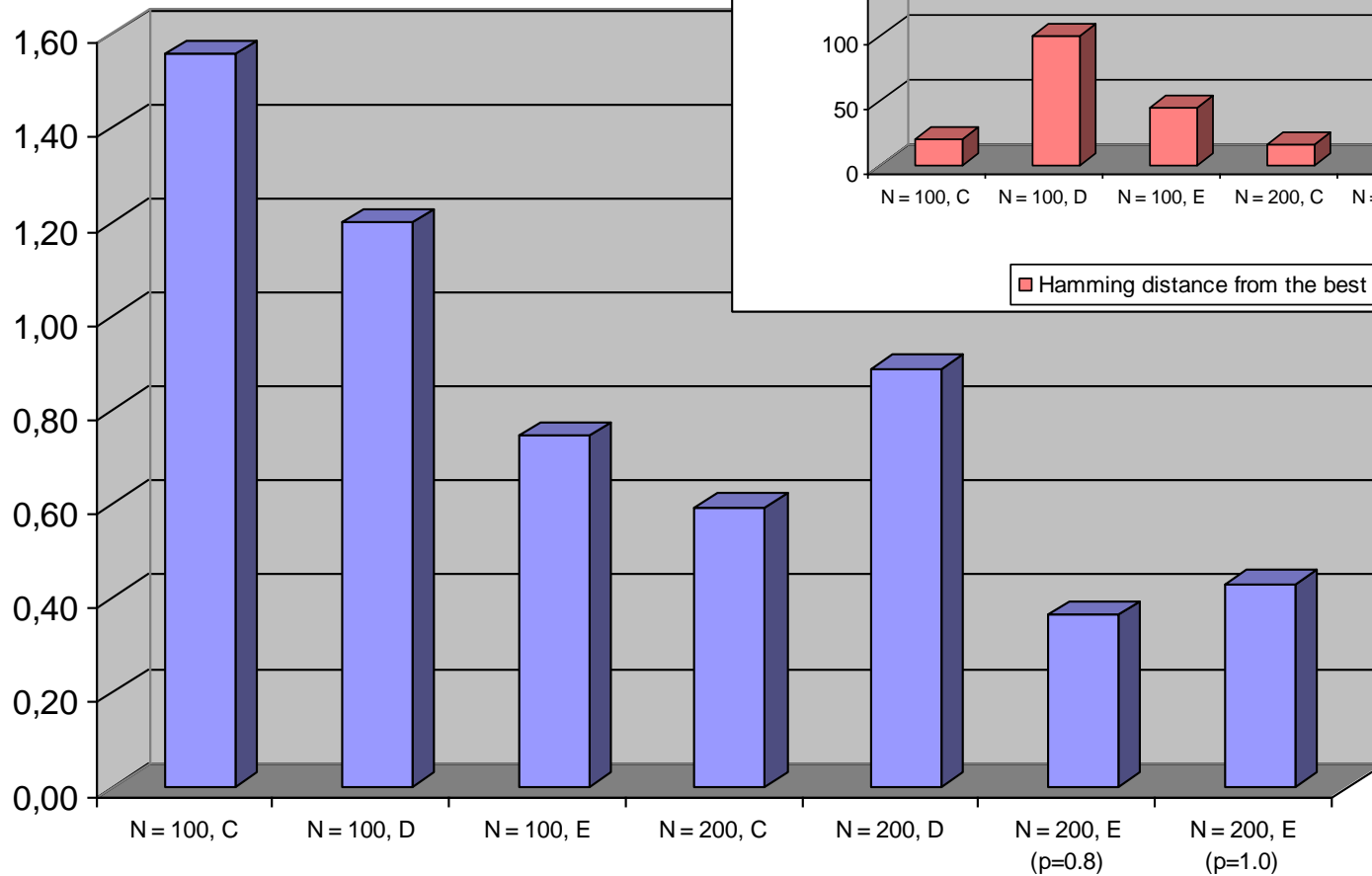
# Integrality gap (%) – sort by agent



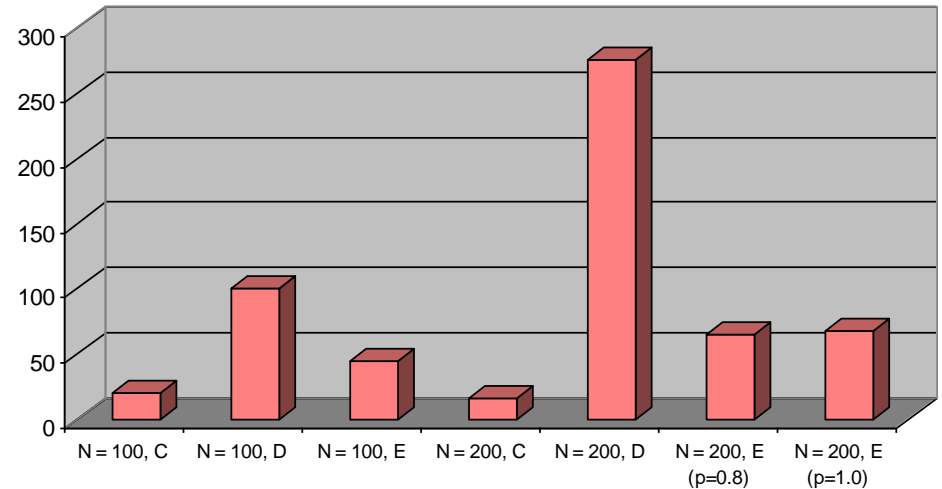
# Integrality gap (%) – sort by level



# Performances on the root node



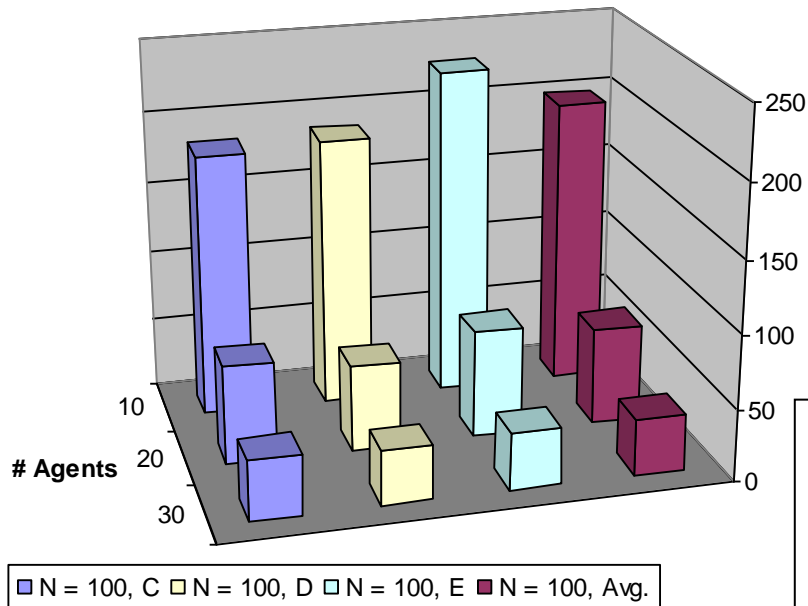
■ Primal-dual gap (%)



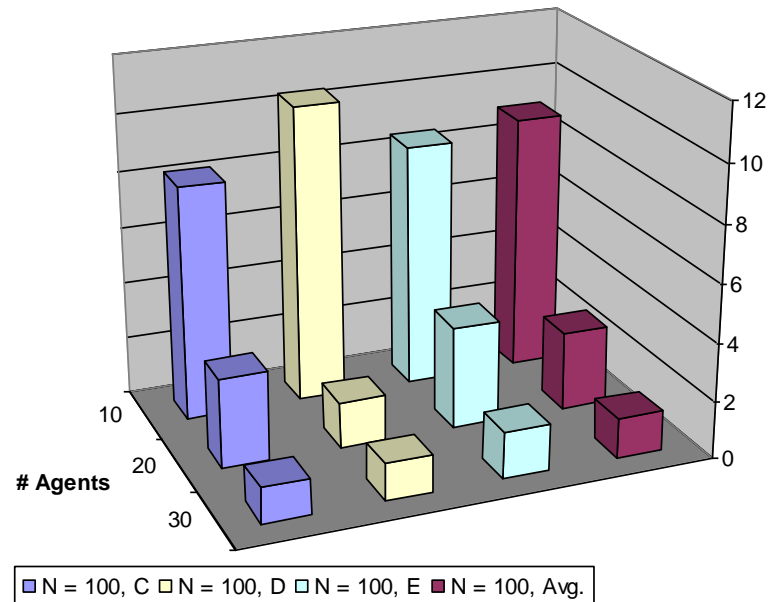
■ Hamming distance from the best solution known

# Multiple Pricing

CG iterations - Root node - Agents



CG iterations - Search tree - Agents





# Branch and Price N = 100

	M	N	L	Root Node								Search Tree			
				Primal Gap(%)	Hamming	Int. Gap (%)	P-D Gap (%)	CGs	Time (s)	Fract.	Final Gap	Avg CGs	Time(s)		
C	10	100	3	1,49	16,0	0,34	1,83	198,4	5,5	1,23	0,00	5,00	9,27	6,3	
	10	100	4	0,55	23,6	0,45	1,00	182,4	5,8	1,33	0,00	5,00	7,81	12,4	
	10	100	5	0,67	19,2	0,39	1,05	169,0	5,1	1,22	0,00	5,00	7,43	8,5	
	20	100	3	1,46	20,0	0,38	1,84	70,8	1,5	1,31	0,00	5,00	2,75	2,8	
	20	100	4	1,44	16,4	0,40	1,84	71,8	1,6	1,43	0,00	5,00	4,02	3,4	
	20	100	5	0,85	18,8	0,40	1,24	62,6	1,4	1,41	0,00	5,00	2,53	2,6	
	30	100	3	2,26	29,2	0,34	2,60	45,4	1,0	1,41	0,00	5,00	2,14	2,3	
	30	100	4	1,67	22,4	0,35	2,02	44,2	1,0	1,50	0,00	5,00	1,31	2,7	
	30	100	5	0,58	16,8	0,00	0,58	33,6	0,8	1,83	0,00	5,00	0,33	1,7	
				1,22	20,3	0,34	1,56	97,6	2,6	1,41	0,00	45,00	4,18	4,7	
D	10	100	3	0,45	52,0	0,06	0,51	190,4	8,2	1,73	0,03	2,00	14,34	4699,4	
	10	100	4	0,33	50,8	0,09	0,42	205,0	9,2	1,71	0,07	1,00	11,13	2231,4	
	10	100	5	0,64	93,2	0,06	0,70	166,4	7,8	2,03	0,06	2,00	5,74	560,8	
	20	100	3	1,06	96,8	0,60	1,65	65,2	3,2	2,45	0,71	0,00	1,66 -		
	20	100	4	0,79	116,4	0,52	1,31	59,0	3,0	2,67	0,52	0,00	1,76 -		
	20	100	5	0,95	148,8	0,43	1,38	53,6	2,9	3,32	0,43	0,00	1,38 -		
	30	100	3	0,89	102,0	0,91	1,80	40,2	2,3	2,81	0,89	0,00	1,37 -		
	30	100	4	0,92	95,2	0,55	1,46	38,8	2,5	2,87	0,54	0,00	1,35 -		
	30	100	5	1,05	145,6	0,50	1,56	33,4	2,3	3,37	0,51	0,00	1,16 -		
				0,79	100,1	0,41	1,20	94,7	4,6	2,55	0,42	5,00	4,43	2497,2	
E	10	100	3	0,38	34,0	0,02	0,40	220,0	8,8	1,43	0,00	5,00	13,66	63,8	
	10	100	4	0,04	6,4	0,00	0,04	232,8	9,8	1,13	0,00	5,00	2,36	11,2	
	10	100	5	0,28	19,2	0,00	0,28	231,0	10,1	1,21	0,00	5,00	9,58	15,7	
	20	100	3	0,82	40,4	0,03	0,85	73,6	3,0	1,71	0,00	5,00	3,39	118,2	
	20	100	4	0,43	54,8	0,02	0,45	75,0	3,2	1,78	0,00	5,00	3,33	100,5	
	20	100	5	0,60	57,6	0,03	0,62	75,8	3,3	1,77	0,00	5,00	4,01	95,8	
	30	100	3	1,84	63,6	0,07	1,91	38,2	1,7	2,32	0,00	5,00	1,42	952,6	
	30	100	4	1,03	64,4	0,08	1,11	39,0	1,9	2,20	0,00	5,00	1,62	600,0	
	30	100	5	0,97	66,4	0,07	1,04	39,4	2,4	2,13	0,00	5,00	1,77	481,1	
				0,71	45,2	0,04	0,75	113,9	4,9	1,74	0,00	45,00	4,57	271,0	
				0,91	55,2	0,26	1,17	102,0	4,1	1,90	0,14		4,39	924,3	

# Branch and price N = 200

	M	N	L	Root Node								Search Tree			
				Primal Gap(%)	Hamming	Int. Gap (%)	Primal-Dual $\zeta$ CGs	Time (s)	Fract.	Final Gap	Avg CGs	Time(s)			
C	15	200	4	0,45	16,4	0,23	0,68	389,6	116,0	1,25	0,00	5,00	9,48	137,9	
	15	200	5	0,75	27,2	0,20	0,94	463,8	192,1	1,26	0,00	5,00	34,71	326,2	
	30	200	4	0,56	8,4	0,09	0,65	88,8	18,8	2,07	0,00	5,00	1,34	27,0	
	30	200	5	0,10	12,4	0,00	0,10	63,8	15,2	2,74	0,00	5,00	0,20	17,2	
					0,46	16,1	0,13	0,59	251,5	85,5	1,83	0,00	20,00	11,43	127,1
D	15	200	4	0,55	208,8	0,15	0,70	319,2	187,3	2,56	0,08	0,00	10,94	-	
	15	200	5	0,51	242,8	0,11	0,62	310,2	223,5	3,13	0,08	0,00	9,50	-	
	30	200	4	0,59	317,6	0,66	1,25	91,0	35,9	4,13	0,56	0,00	2,11	-	
	30	200	5	0,52	336,4	0,46	0,98	84,8	33,6	4,71	0,25	0,00	2,28	-	
					0,54	276,4	0,34	0,89	201,3	120,1	3,63	0,24	0,00	6,21	-
E 0.08	15	200	4	0,57	66,8	0,01	0,58	537,6	207,0	1,31	0,00	5,00	38,80	1336,3	
	15	200	5	0,34	50,8	0,00	0,34	517,6	180,5	1,32	0,00	5,00	27,43	679,9	
	30	200	4	0,29	84,0	0,01	0,30	173,6	48,4	1,60	0,00	5,00	7,94	1966,7	
	30	200	5	0,23	59,6	0,01	0,23	157,2	44,2	1,58	0,00	5,00	6,91	350,8	
					0,36	65,3	0,01	0,36	346,5	120,0	1,45	0,00	20,00	20,27	1083,4
E 1.00	15	200	4	0,45	52,8	0,01	0,46	496,0	174,5	1,32	0,00	5,00	34,78	700,5	
	15	200	5	0,18	62,4	0,01	0,19	543,4	203,2	1,25	0,00	5,00	34,11	1165,3	
	30	200	4	0,53	78,4	0,01	0,53	149,2	410,1	1,42	0,00	5,00	8,32	367,0	
	30	200	5	0,54	75,2	0,01	0,54	150,2	418,5	1,57	0,00	5,00	7,58	1578,0	
					0,42	67,2	0,01	0,43	334,7	301,6	1,39	0,00	20,00	21,20	952,7

# References

- Laguna M, Kelly JP, Gonzáles-Velarde JL, Glover F (1995) Tabu search for the multilevel generalized assignment problem, European Journal of Operational Research 82: 176-189
- French AP, Wilson JM (2002) Heuristic Solution Methods for the Multilevel Generalized Assignment Problem, Journal of Heuristics 8: 143-153
- Osorio MA, Laguna M, Logic cuts for multilevel generalized assignment problems, European Journal of Operational Research, to appear
- Pisinger D (1995) A minimal algorithm for the Multiple Choice Knapsack Problem, European Journal of Operational Research 83: 758-767

ceselli@dti.unimi.it