COMPUTATIONAL EXPERIMENTS WITH LOCATION PROBLEMS USING AUTOMATIC CONSTRAINT GENERATION

Jaume BARCELÓ

RR85/06

juny

COMPUTATIONAL EXPERIMENTS WITH LOCATION PROBLEMS USING AUTOMATIC CONSTRAINT GENERATION

J. Barceló
Dept. d'Investigació Operativa i Estadística
Facultat d'Informàtica
Jordi Girona Salgado, 31
08034 Barcelona, Spain

ABSTRACT

We report computational experience with the use of automatic constraint generation in the case of capacitated location problems involving between 10 and 30 plants and up to 50 demand centers. For different types of model formulations different surrogate knapsack constraints are investigated and found to substantially impact in many cases the actual numerical results.

RESUMEN

Se reporta la experiencia computacional derivada del uso de procedimientos de generación automática de restricciones para problemas de localización de plantas con restricciones de capacidad, de dimensiones entre 10 y 30 plantas y hasta 50 centros de demanda. Para distintas formulaciones del modelo se investigan diferentes tipos de restricciones knapsack surrogadas. Los resultados obtenidos presentan en muchos casos una reducción sustancial del esfuerzo computacional.

COMPUTATIONAL EXPERIMENTS WITH LOCATION PROBLEMS USING AUTOMATIC CONSTRAINT GENERATION

J. Barceló
Dept. d'Investigació Operativa i Estadística
Facultat d'Informàtica
Jordi Girona Salgado, 31
08034 Barcelona, Spain

ABSTRACT

We report computational experience with the use of automatic constraint generation in the case of capacitated location problems involving between 10 and 30 plants and up to 50 demand centers. For different types of model formulations different surrogate knapsack constraints are investigated and found to substantially impact in many cases the actual numerical results.

1. INTRODUCTION

This computational study is the result of a blending of empirical and theorical facts about capacitated plant location problems. Among the empirical facts one should mention the strength of the dissaggregated formulations for plant location problem as, for instance those including dissaggregated constrainsts of the form

$$x_{ij} < b_j Y_j \quad \forall i \in I, \forall j \in J \qquad (1)$$

where Y_j are the 0-1 decision variables associated with the plants, $J=\{1,\ldots,m\}$, is the set of potential locations for the plants, and x_{ij} are the variables associated with shipments from plant j to center i, being $I=\{1,\ldots,n\}$, the set of centers whose demands d_i must be satisfied and $b_j, j \in J$, the capacities of the plants. In a former work Spielberg, /12/, stressed the importance of such constraints and later on Schrage, /10/, tested widely the computational effects of adding constraints (1).

Geoffrion and McBride, /5/, explained why these constraints are computationaly efficient by proving that they are facets of the convex hull of the capacitated plant location problem.

However from an LP approach to solve capacitated plant location problems includinf constraints (1) in the LP formulation of the problem has effort and computational costs, given the increase in the size of the problem, which can compensate the other advantages, at least one disposes of

a modified LP code able to deal implicity with such constraints as a special case of general upperbounding constraints, (Schrage, /10/, /11/).

Another empirical fact has been the positive experience of including also aggregated constraints like:

$$\sum_{j \in J} b_j Y_j \geqslant d \qquad (2)$$

where

$$d = \sum_{i \in I} d_i$$

is the aggregated demand, in the mixed integer programming formulation of capacitated plant location problems. Computational experience of such a fact is reported by Guignard and Spielberg, /6/, and by Van Roy, /13/, in a different algorithmic approach showing that constraints (2) help to compute stronger Benders cuts in the cross-decomposition procedure.

In the above referenced paper Guignard and Spielberg recommend also adding "logical"constraints, or cuts, of the form:

$$\sum_{j \in S} Y_j \geqslant L \,, \quad S \subseteq J \qquad (3)$$

but leaving open the question of how to build up systematically the best ones of such logical constraints, that is how to identify the best subset S and compute the corresponding best lower bound L.

Taking into account all the above considerations we try to study the computational behaviour of capacitated plant location problems, formulated as mixed integer programming problems, when dissaggregated constraints (1) are not included given that the LP code at hand is an ordinary one (in our case the XMP of Roy Marsten), and an automatic procedure to generate constraints (3) is available.

2. AUTOMATIC CONSTRAINT GENERATION FOR CAPACITATED PLANT LOCATION PROBLEMS.

When in the mixed integer programming formulation of capacitated plant location problems one considers the "surrogate knapsack" problem

$$\sum_{j \in J} b_j Y_j \geqslant d \ , \quad Y_j \in \{0,1\} \ , \ \forall j \in J \tag{4}$$

and the associated knapsack polytope

$$K = \text{CONV} \left\{ Y \in R^m \,\middle|\, \sum_{j \in J} b_j Y_j \geqslant d \ , \ Y_j \in \{0,1\} , \ \forall j \in J \right\} \tag{5}$$

if P is the polytope associated with the capacitated plant location problem, then Padberg, Van Roy and Wolsey, /9/, prove the following proposition:

Proposition 2.1

Every valid inequality for K is a valid inequality for P.
(which eventually, if some additional condictions hold -see reference above- could be also a fact of P).

As it is well known the concept of cover inequalities for knapsack problems, Balas, /1/, provides a procedure of computing valid inequalities and facets for the knapsack polytope K. Thus, when the formulation of capacitated plant location problems includes surrogate knapsack constraints like (4), the procedure of computing valid inequalities from minimal covers provides a systematic procedure of generating logical constraints of the form

$$\sum_{j \in S} \pi_j Y_j \geqslant \pi_o \qquad (6)$$

which include the family of constraints (3) as a particular case given that in certain cases constraints (6) can be generated with coefficients $\pi_j > 1$, being consequently more powerful than constraints (3).

That answers the first question stated above. The second one, that of identifying the best of such inequalities, was answered by Padberg proving that the best one of the minimal cover inequalities was the most violated by the current solution $\bar{Y}$ to the LP relaxation of the problem.

Let $\bar{Y}$ be the current solution to the LP relaxation of the surrogate knapsack constraint (4), then Crowder, Johnson and Padberg, /4/, prove the following proposition:

Proposition 2.2

Let (z,s) be an optimal solution to the knapsack problem

$$z=\text{MIN}\left\{\sum_{j\in J}\bar{Y}_j s_j \;\middle|\; \sum_{j\in J} b_j s_j \geqslant \sum_{j\in J} b_j - d,\ s_j \in \{0,1\}\ ,\ \forall j \in J\right\}$$

where $S \subseteq J$ is the support of s. Then:

- If $z<1$ the valid inequality

$$\sum_{j\in S} Y_j \geqslant 1 \qquad (7)$$

for K (and therefore for P) is violated by $\bar{Y}$.

However, whenever such an inequality (7) exists, it will be a facet of the polytope K^S defined by

$$K^S = K \cap \left\{ y \in R^m \;\middle|\; Y_j = 0\ ,\ \forall j \notin S \right\} \qquad (8)$$

facet which can be strengthened, that is, lifted into the m-space to yield a facet of K, by means of the following lifting procedure, /4/, based on the lifting theorems, /7/ proved in Padberg, /8/, Balas and Zemel, /3/.

Sequential Lifting Procedure

Set $\pi_j = 1 \quad , \quad \forall j \in S$

$f_o = |S| - 1$

Iterative step

For $k \in J \setminus S$

Compute $Z_k = \text{MAX} \sum_{j \in S} \pi_j s_j$

$$\text{s.t.} \sum_{j \in S} b_j s_j \leqslant \sum_{j \in S} b_j - d - b_k$$

$$s_j \in \{0,1\} \quad , \quad \forall j \in S$$

Define

$\pi_k = f_o - Z_k$

$S = S \cup \{k\}$

Repeat until $J \setminus S = \emptyset$

Then define $\pi_o = \sum_{j \in J} \pi_j - f_o$

The inequality

$$\sum_{j \in J} \pi_j Y_j \geqslant \pi_o \qquad (9)$$

is the facet of K most violated by the current LP solution $\bar{Y}$, and as it is a valid inequality (eventually a facet) of P, this valid inequality cuts -off the current LP solution.

The computational experience that this paper reports is based on the algorithmic approach represented in the flow chart of (Fig. 1), and consists on given a mixed integer formulation for the capacitated plant location problem which includes a surrogate knapsack constraint (4), compute from the solution to the successive LP relaxation as

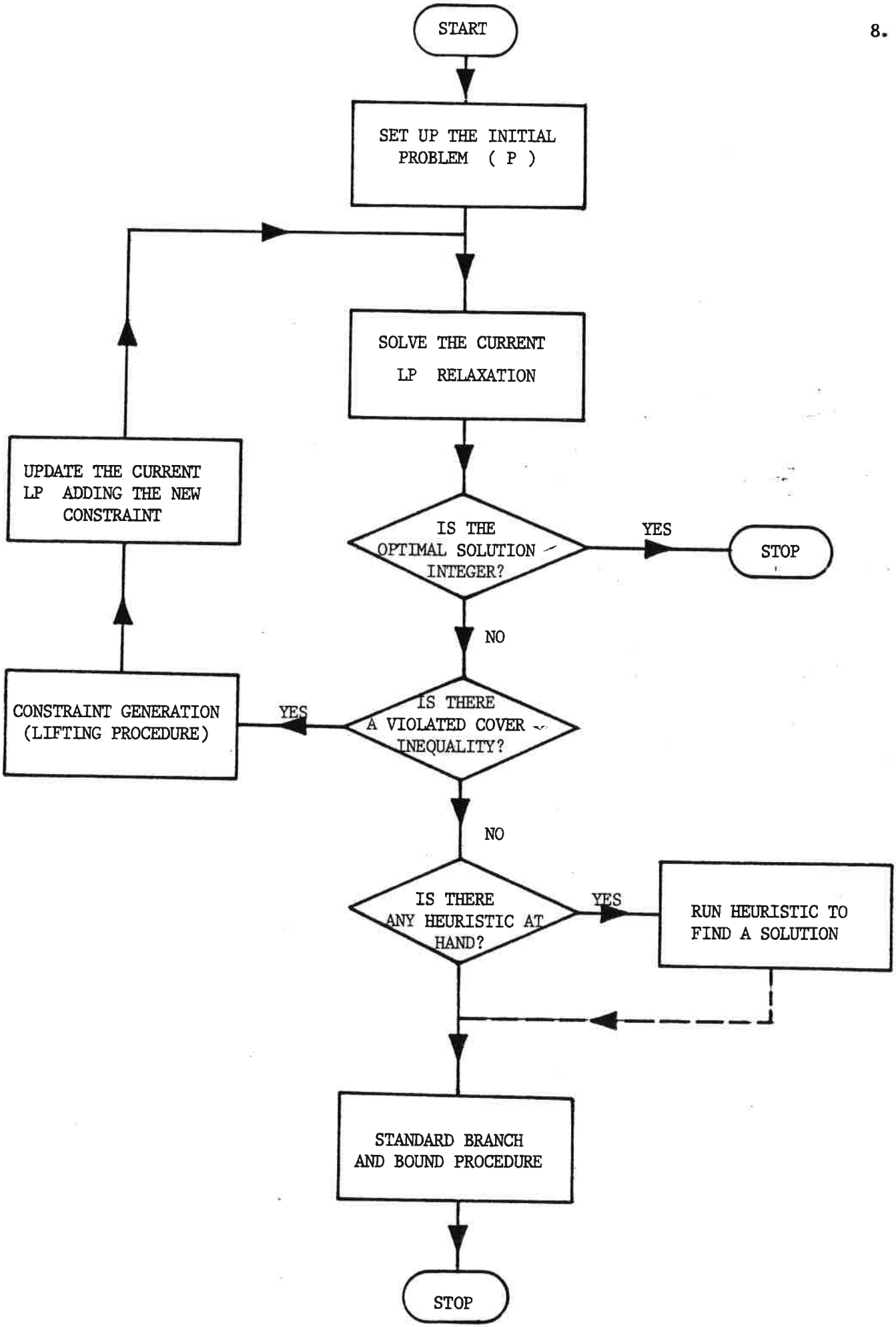
START
SET UP THE INITIAL PROBLEM (P)
SOLVE THE CURRENT LP RELAXATION
IS THE OPTIMAL SOLUTION INTEGER?
YES
STOP
NO
IS THERE A VIOLATED COVER INEQUALITY?
YES
CONSTRAINT GENERATION (LIFTING PROCEDURE)
UPDATE THE CURRENT LP ADDING THE NEW CONSTRAINT
NO
IS THERE ANY HEURISTIC AT HAND?
YES
RUN HEURISTIC TO FIND A SOLUTION
STANDARD BRANCH AND BOUND PROCEDURE
STOP

FIGURE 1.

much constraints (9) as possible adding them to the problem formulation, before to proceed to a branch and bound when addition of constraints (9) does not provides with an integer solution. In our experiences, as we were using the XMP, we used the pivot and complementing heuristic of Balas and Martin, /2/, to generate a starting upper bound for the branch and bound.

3. PROBLEM FORMULATION

The problem formulation used to perform the computational experiments reported here has been as follows:

$$z = \text{MIN} \sum_{i \in I} \sum_{j \in J} c_{ij}\, x_{ij} + \sum_{j \in J} f_j\, y_j$$

$$\sum_{j \in J} x_{ij} = 1 \,, \quad \forall i \in I \tag{10}$$

$$\sum_{j \in J} y_j \leqslant K \tag{11}$$

$$\sum_{j \in J} b_j\, y_j \geqslant d \,, \quad d = \sum_{i \in J} d_i \tag{12}$$

$$\sum_{i \in I} d_i\, x_{ij} \leqslant b_j\, y_j \,, \quad \forall j \in J \tag{13}$$

$$0 \leqslant x_{ij} \leqslant 1 \,, \quad \forall i \in I \,, \quad \forall j \in J \tag{14}$$

$$y_j \in \{0,1\} \,, \quad \forall j \in J \tag{15}$$

where as usually c_{ij} are the supplying costs from plant j to center i and f_j are the fixed costs of opening a plant at the potential site j. The set of constraints (10) assures that the demand of all centers will be satisfied, constraint (11) limits the number of plants to be open to a maximum of K, constraint (12) is the surrogate knapsack constraint discussed above, and the set of constraints (13) prevents trying to supply a center from a not opened plant ($y_j = 0$ $x_{ij} = 0$, $\forall i \in I$), and does not allow the supply of more than the total capacity of the plant when it is open. These constraints are an aggregation of constraints of type (1) leading to a weaker LP relaxation but requiring less computational effort as has been commented at the beginning when one not disposes of a modified LP code at hand.

Constraints (15) are replaced by

$$0 \leqslant y_j \leqslant 1 \ , \quad \forall j \in J \tag{16}$$

to derive the ordinary LP relaxation, note that constraints (14) and (16) can then be treated as ordinary upperbound variables.

After each LP solution constraint (12) generates a constraint of the form

$$\sum_{j \in J} \pi_j y_j \geqslant \pi_o \tag{9}$$

following the procedure described above.

An alternative to be also studied with this formulation is that of the pure integer capacitated problem, corresponding to the case when each center is allowed to be supplied only from one plant. This problem is obtained replacing constraints (14) by

$$X_{ij} \in \{0,1\}, \quad \forall i \in I, \forall j \in J \qquad (17)$$

In such a case constraints (13) are also knapsack-like constraints and then can generate additional constraints of the form:

$$\sum_{i \in I} \pi_i X_{ij} \leq b_j Y_j, \quad j \in J^* \qquad (18)$$

being J^* the set of open plants, in the same way as constraints (9) were generated from (12).

4. COMPUTATIONAL EXPERIENCE

Six series of problemes of different sizes and features were solved and the results are shown in tables 1 to 6. All these problems were generated using a random test problem generator with the following characteristics. Given the parameters DMIN, minimum value of the demand, DPROM approximate average value of the demands, CMIN minimum transportation cost, CPROM approximate average transportation cost, and DCOEF average ratio between the capacity of K plants and the total demand d, problem data are generated in

the following way: K, limit number of plants to be open, is generated in the interval [0.1n,n], being n the number of centers defined by the users; demands d_i are generated as randomly distributed in an interval centered on DPROM which has DMIN as lower limit; once the demands are generated, BPROM, average plant capacities, is computed as (TOTAL DEMAND)/(DCOEF*K), and then capacities b_j are generated as randomly distributed in the interval [BMIN, 2*BPROM-BMIN], where BMIN is fixed to DMIN. Transportation costs are computed as $c_{ij}=a_{ij}+r_{ij}.d_i$, assuming that they have two components, the first one a_{ij}, a fixed component depending on the plant, determined randomly, and the second $r_{ij}.d_i$ proportional to the demand. Fixed costs f_j can be computed in two ways, either as proportional to capacities or as proportional to capacities plus a random component but always assuming that they satisfy some proporcionality with transportation costs.

Computational results for each serie of experiments are recorded in the corresponding table, and are the following: first column on each table identifies the problem, second column identifies the final status of the problem. Problem status a means that for this problem the constraint identification technique has been successfull, and has reduced significatively the computational effort required for solving the problem, while problem status b has the opposite meaning, for this class of problems the constraint identification technique not only was not successfull but in fact increased the amount of computations for solving the problem. Some other problems have no defined status, thas means that for them the technique was not applicable because

no cutting planes were possible to compute. Third and fourth columns give information about the structure of the problem. They contain respectively the values of the ratios between the total demand d and the total capacity $\sum_{j \in J} b_j$ offered by the plants, and the mean ratio between the fixed costs f_j and the total transportation costs $\sum_{i \in I} c_{ij}$ for each plant. Columns five and six give notice of the computational effort required for solving the problem using a standard branch and bound code without cutting planes. Column five reports the number of nodes explored and column six the total number of LP pivots. Column seven gives more information about the features of the problem, values in this column represent the gap:

(optimal Integer value-optimal LP value).100/optimal LP value

As we mentioned before the strategy was that of computing as many cutting planes as possible before proceeding to branch and bound. Column eight gives the number of cutting planes computed for each problem. Columns nine and ten report the computational effort required for solving the problem by the standard branch and bound code once the cutting planes were added. Column nine gives the number of nodes explored and column ten gives the total number of LP pivots, included the ones required to restore primal feasibility after adding cutting planes.

The last column gives an estimate of the percentual gap reduction due to the cutting planes added.

Problems in tables 1, 2 and 3 are of size 20x10, that is 20 centers and 10 plants. Table 1 includes 15 problems generated with the assumption that fixed costs f_j were proportional to capacities b_j, problems in Tables 2 and 3 were generated using the same seeds as problems 1.3 to 1.13, in Table 1, but assuming that f_j were proportional to capacities b_j plus a random amount (Table 2), and also assuming different ratios between f_j and $\sum_{i \in I} c_{ij}$, (Table 3). Table 4 includes 18 problems of size 30x15, problems 4.1, to 4.9, were generated assuming f_j proportional to b_j, and problems 4.10, to 4.18 assuming the more general hypothesis that there is also a random modifier added to the proportional relationship. Table 5 reports the results for 14 problems of size 40x20, and Table 6 the results for 12 problems of size 40x30 plus a problem of size 50x25.

All the experiences were done in a VAX 785 using the XMP code of Roy Marsten, with the pivot and complementing heuristic for getting an initial feasible solution.

5. CONCLUSIONS

As it was told above, problems marked with a are those for which the constraint generation technique was successful, a comparison between columns five and nine, and six and ten gives us an idea about the reduction in the computational effort. For instance, for problem 1.1 the 48 nodes and 1130 LP pivots needed for solving directly the

problems, were reduced to 16 nodes and 199 LP pivots after adding one cutting plane. Problem 5.7 represents a more spectacular case were after 362 nodes and more than 10000 LP pivots, computations were stopped without getting a feasible solution and after adding 4 cutting planes optimal solution was found after exploring 117 nodes and performing 3329 LP pivots.

Within classes a and b there exist two subclasses identified respectively by a* and b. in the Tables. Problems which status is marked a* are problems for which the final result was successful but such that if instead of adding as many cutting planes as possible one had stopped the procedure before then the results had been even better. For instance problem 2.7, requires 66 nodes and 1188 LP pivots to be solved directly but after 5 cutting planes these effort is reduced to only 43 nodes and 362 LP pivots, but if we had stopped to add cutting planes after the third one then to solve the problem we had required to explore 19 nodes and 263 LP pivots, a very much better result.

Subclass b. has a similar behaviour. Problem 6.9, for instance, the one showing the worse results, increases the computations from 241 nodes and 7341 LP pivots to 357 nodes and 10049 LP pivots after adding 8 cutting planes, but if we had added the planes one by one we had got the following results, after the first cutting plane we had required only 160 nodes and 5140 which represents a nice improvement, the second and third cutting planes had provided similar results but after the third cutting plane a worsening process starts ending after the eigth cutting plane with the results reported above.

SIZE 20 X 10 , $f_j = K_j\ b_j$

TABLE 1

PROBLEM	status	RATIO $r=\Sigma d/\Sigma b$	RATIO $f/\Sigma c$	BRANCH AND BOUND NODES	BRANCH AND BOUND LP PIVOTS	GAP %	NUMBER OF CUTTING PLANES	BRANCH AND BOUND NODES	BRANCH AND BOUND LP PIVOTS	GAP REDUCTION (%)
1.1	a	0.70	1.40	48	1130	10.75	1	16	199	99.71
1.2	a	0.75	1.40	48	466	5.76	1	0	132	100.00
1.3	-	0.47	4.45	44	512	9.83	-	-	-	--
1.4	a	0.58	3.18	16	276	1.34	2	16	267	1.65
1.5	a	0.29	7.26	58	646	2.34	-	-	-	--
1.6	-	0.23	6.99	43	587	6.50	-	-	-	--
1.7	b	0.58	3.23	53	1407	0.57	1	53	1564	1.83
1.8	-	0.43	4.77	44	526	4.04	-	-	-	--
1.9	-	0.38	4.58	44	552	7.11	-	-	-	--
1.10	-	0.29	6.57	53	1714	1.60	-	-	-	--
1.11	b.	0.81	1.27	54	469	1.56	3	51	505	22.04
1.12	a	0.75	1.33	40	417	1.35	1	41	394	18.44
1.13	a	0.77	0.63	51	461	0.68	1	16	269	4.72
1.14	b	0.64	0.61	67	523	6.65	1	67	580	0.015
1.15	a	0.77	0.53	22	350	5.92	3	16	172	10.36

SIZE 20 X 10 , $f_j = K_j\, b_j + r_j$ (r_j random)

TABLE 2

PROBLEM	status	$r=\Sigma d/\Sigma b$	$f/\Sigma c$	BRANCH AND BOUND		GAP %	NUMBER OF CUTTING PLANES	BRANCH AND BOUND		GAP REDUCTION (%)
				NODES	LP PIVOTS			NODES	LP PIVOTS	
2.1	a	0.64	0.79	46	785	6.19	5	16	301	17.41
2.2	b.	0.47	4.77	37	399	3.00	5	19	409	72.64
2.3	a	0.58	3.67	63	763	3.40	2	51	451	32.17
2.4	b	0.29	6.95	41	393	25.48	3	46	681	14.48
2.5	b.	0.23	6.49	19	240	7.25	4	16	251	95.40
2.6	b	0.58	2.82	49	442	5.24	3	52	538	11.78
2.7	a*	0.43	5.57	66	1188	2.13	5	43	362	23.84
2.8	b.	0.38	4.13	34	346	1.53	2	19	357	36.68
2.9	a	0.29	6.70	37	409	11.05	4	16	195	41.81
2.10	a	0.64	1.94	64	1161	1.00	1	64	1078	0.87

SIZE 20 X 10 , $f_j = K_j\ b_j + r_j$ (r_j random)

TABLE 3

PROBLEM	status	$r=\Sigma d/\Sigma b$	$f/\Sigma c$	BRANCH AND BOUND		GAP %	NUMBER OF CUTTING PLANES	BRANCH AND BOUND		GAP REDUCTION (%)
				NODES	LP PIVOTS			NODES	LP PIVOTS	
3.1	b	0.66	1.42	44	329	2.59	3	43	408	5.41
3.2	a	0.68	1.99	37	390	1.78	3	36	293	8.50
3.3	a	0.73	1.75	46	498	4.90	4	16	221	96.13
3.4	b.	0.64	1.54	35	363	7.83	2	36	409	39.43
3.5	a	0.59	1.77	46	438	2.36	7	16	230	89.28
3.6	b.	0.68	2.26	35	325	2.05	6	34	374	49.92
3.7	b	0.72	1.44	16	254	3.57	3	19	287	26.32
3.8	a	0.72	1.72	35	565	5.28	4	16	248	97.51
3.9	a	0.72	1.55	41	526	2.50	1	41	519	4.25
3.10	-	0.79	1.29	45	461	0.29	-	-	-	--

SIZE 30 x 15

TABLE 4

PROBLEM	status	$r=\Sigma d/\Sigma b$	$f/\Sigma c$	BRANCH AND BOUND		GAP %	NUMBER OF CUTTING PLANES	BRANCH AND BOUND		GAP REDUCTION (%)
				NODES	LP PIVOTS			NODES	LP PIVOTS	
4.1	-	0.66	1.00	100	1482	0.87	-	-	-	--
4.2	b	0.70	0.98	82	1289	0.53	2	90	1257	8.34
4.3	a	0.87	0.36	43	746	0.76	6	37	590	16.16
4.4	-	0.67	0.46	98	1350	1.91	-	-	-	--
4.5	-	0.77	0.40	66	850	0.35	-	-	-	--
4.6	-	0.67	0.44	72	3825	0.93	-	-	-	--
4.7	-	0.70	0.43	84	1245	2.23	-	-	-	--
4.8	-	0.78	0.54	77	1167	1.65	-	-	-	--
4.9	b	0.81	0.54	16	387	0.09	3	37	756	99.26
4.10	a	0.89	0.63	48	640	4.83	4	36	423	90.77
4.11	a	0.67	0.36	84	1548	1.70	2	49	838	12.08
4.12	b	0.67	0.48	73	1749	2.70	1	214	4956	6.77
4.13	a	0.78	0.58	70	791	2.04	4	48	689	14.83
4.14	b	0.71	1.29	16	385	0.71	2	16	424	54,67
4.15	a	0.39	1.84	138	2316	3.30	3	72	818	51.27
4.16	b.	0.33	1.77	49	779	4.30	4	40	815	43.43
4.17	a*	0.44	1.90	96	2613	4.60	5	44	628	49.75
4.18	a*	0.38	1.75	113	1272	5.02	6	64	825	61.39

SIZE 40 x 20

TABLE 5

PROBLEM	status	$r=\Sigma d/\Sigma b$	$f/\Sigma c$	BRANCH AND BOUND		GAP %	NUMBER OF CUTTING PLANES	BRANCH AND BOUND		GAP REDUCTION (%)
				NODES	LP PIVOTS			NODES	LP PIVOTS	
5.1	a	0.78	0.47	103	2118	2.42	3	74	1219	6.77
5.2	a*	0.75	0.84	122	1779	1.26	6	42	958	27.77
5.3	a*	0.81	0.92	84	2032	1.13	5	48	1389	15.22
5.4	a*	0.77	0.98	97	2078	1.02	3	75	1434	22.04
5.5	a	0.8	0.80	93	3151	2.76	2	75	2160	3.40
5.6	a*	0.76	1.00	222	5123	1.87	5	95	1777	38.61
5.7	a*	0.64	1.36	362	> 10000	4.33	4	117	3329	19.97
5.8	b	0.7	1.16	74	1479	1.05	2	83	1615	2.95
5.9	a	0.63	1.30	44	771	0.33	1	16	492	79.42
5.10	b	0.57	1.48	101	1790	1.90	1	116	2052	18.80
5.11	a	0.19	3.30	73	1560	9.05	8	35	760	35.07
5.12	a	0.42	1.75	44	889	2.90	4	39	869	7.24
5.13	a	0.49	1.82	84	1533	6.10	5	62	1217	80.04
5.14	a	0.17	3.64	56	886	3.73	3	38	753	20.43

SIZE 40 x 30

TABLE 6

PROBLEM	status	$r=\sum d/\sum b$	$f/\sum c$	BRANCH AND BOUND		GAP %	NUMBER OF CUTTING PLANES	BRANCH AND BOUND		GAP REDUCTION (%)
				NODES	LP PIVOTS			NODES	LP PIVOTS	
6.1	b.	0.39	1.47	48	680	0.66	3	35	1030	10.84
6.2	b.	0.69	0.83	57	1360	0.28	5	49	1418	13.69
6.3	b	0.48	0.88	216	3737	1.48	1	172	4930	0.39
6.4	a	0.3	2.21	94	1958	0.54	5	41	585	26.40
6.5	a	0.48	0.5	130	3124	0.75	1	43	812	71.83
6.6	a	0.69	0.72	109	2160	0.26	2	22	765	6.46
6.7	b	0.51	1.25	22	563	0.13	4	25	781	14.71
6.8	a*	0.67	1.23	165	3460	0.47	5	16	659	94.78
6.9	b.	0.4	1.14	241	7341	2.28	8	357	10049	8.87
6.10	b.	0.71	0.58	72	1394	0.25	3	72	1649	12.30
6.11	a*	0.59	0.66	-	--	1.31	5	148	3185	26.60
6.12	b.	0.30	1.60	117	2560	4.28	3	125	2545	31.58
(50 x 25)										
7.1	a	0.40	1.58	151	3091	1.86	2	135	2843	41.93

Table 7 sumarizes the results for the eigthy problems reported in tables 1 to 6.

Class \ Table	1	2	3	4	5	6	Total Number of Problems in class
a	6	5	5	7	12	6	41
b.	1	3	2	1	0	5	12
b	2	2	2	4	2	2	14
No cutting planes	6	0	1	6	0	0	13
	15	10	10	18	14	13	80

strictly speaking this technique was successful in only a fifty per cent of cases (41 out of 80), but it is obvious that this number can be improved if a stopping rule for problems in subclasses a* and b. is at hand, and if it was possible to identify a priori when a problem belongs to class b.

In an attempt to derive such a stopping rule problems in subclasses a* and b. have been solved again adding one cutting plane each time instead of all the cutting planes computable for each problem. The behaviour shown by all those problems was the following: the first cutting planes, the ones leading to an improvement usually produced a strong reduction in the gap, and the next ones produced a

smaller reduction in the gap. Consequently comparing the gap reduction produced each time a new cutting plane is added with the previous ones one can decide whether or not to continue adding more cutting planes and then avoid situations as b. and a*.

Trying to answer the question of it was possible or not to identify a priory problems belonging to class b, a cluster analysis in the space of the principal components has been done. The analysis was done using for each problem the values of the following variables: ratio $R1=\sum_{i \in I} d_i / \sum_{j} b_j$ (values in column 3 in tables 1 to 6), mean ratio $R2= f_j / \sum_{i \in I} c_{ij}$ (values in column 4 in tables 1 to 6), RSIZ= Number of centers/Number of plants, IND=(ZHEU-ZLP).100/ZLP, where ZHEU is the objective function value for the pivot and complement heuristic, and ZLP is the optimal LP value. IND can be interpreted as an upper bound estimate of the duality gap. The last variable used in the analysis was BR= (ZCUT-ZLP).100/ZLP, where ZCUT is the optimal LP value after the first cutting plane.

Figure 2, displays graphically the results of the analysis on the plane determined by the two principal components, explaining the 65.66% of the total variance. They appear four meaningful classes. Class 1 collects problems which can be considered as outlayers. Class 2, clearly includes problems with high values of R2 and IND, that is problems with big values for ratio $f_j / \sum_{i \in I} c_{ij}$, presenting also a large gap, while class 4, is determined by values of parameter R1, ratio between total demand and total capacity, and class 3 is defined mainly by ratio Number of centers/

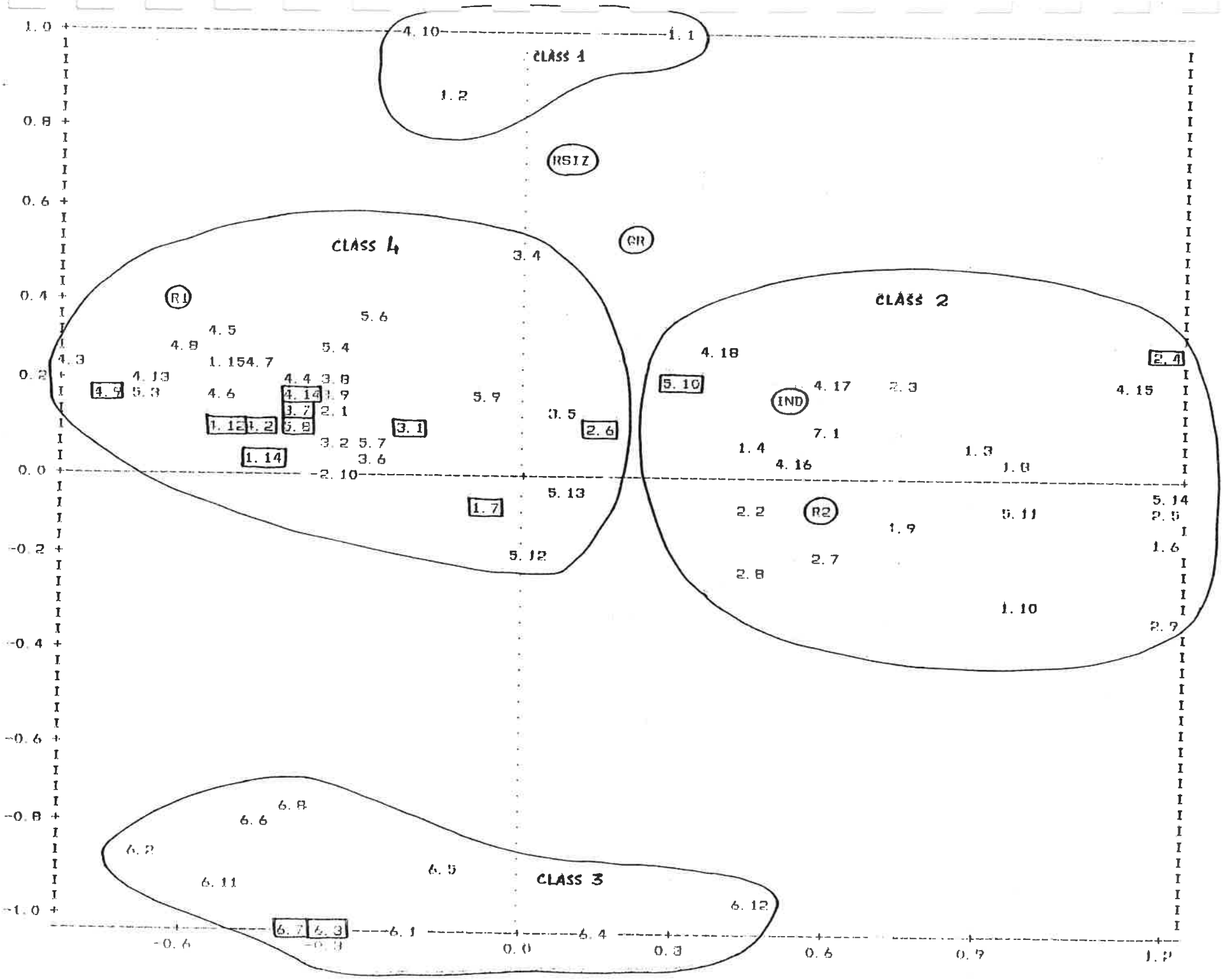
CLASS 1
CLASS 4
CLASS 2
CLASS 3
RSIZ
GR
IND
R1
R2
4. 10
1. 1
1. 2
3. 4
5. 6
4. 5
4. 8
5. 4
4. 3
1. 15
4. 7
4. 13
4. 4
3. 8
4. 9
5. 3
4. 6
4. 14
1. 9
3. 7
2. 1
1. 12
4. 2
6. 8
3. 1
3. 2
5. 7
1. 14
3. 6
2. 10
5. 9
3. 5
2. 6
5. 13
1. 7
5. 12
4. 18
2. 4
5. 10
4. 17
2. 3
4. 15
7. 1
1. 4
1. 3
4. 16
1. 8
5. 14
2. 2
5. 11
2. 5
1. 9
1. 6
2. 7
2. 8
1. 10
2. 9
6. 8
6. 6
6. 2
6. 5
6. 11
6. 12
6. 7
6. 3
6. 1
6. 4
1. 0
0. 8
0. 6
0. 4
0. 2
0. 0
-0. 2
-0. 4
-0. 6
-0. 8
-1. 0
-0. 6
-0. 3
0. 0
0. 3
0. 6
0. 9
1. 2

FIGURE 2.

(Multiple points : point represented/hidden point)

4.8	3.10
1.15	1.11
4.7	1.13
4.13	5.5
4.4	5.2
5.3	5.1
4.6	1.12
3.7	4.1
2.1	3.3
4.12	4.11
1.6	1.5
6.1	6.9

Number of plants. However problems b appear everywhere, and the only partial conclusions, given the small size of the sample, is that they seem to be more likely in class 4, that would mean that problems with ratios R1 higher that the average and ratios R2 lower than the average, and no big gap tend to behave as b more often than others.

We conclude that more computational research is required to drawn definitive conclusions, but anyway the constraints identification technique gives good results in a significative number of cases: 53 out of 67 (given that for the 13 remaining cases the technique was not applicable), given that as we show ed before problems b. with the additional stopping rule behave as problems a. Further computational research should include also alternative formulations leading to other valid inequalities and facets, like those derived of (1-k) configurations.

R E F E R E N C E S

/1/ E. BALAS
Facets of the knapsack Polytope
Mathematical Programming, 8, (1975), pp. 146-164.

/2/ E. BALAS and C.H. MARTIN
Pivot and Complement: a Heuristic for 0-1 Programming.
Management Science, 26, (1980), pp. 86-96.

/3/ E. BALAS and E. ZEMEL
Facets of the knapsack Polytope from Minimal Covers.
Siam Journal on Applied Mathematics, 34, (1978), pp. 119-148.

/4/ H. CROWDER, E.L. JOHNSON and M.W. PADBERG
Solving Large-Scale Zero-One Linear Programming Problems
Operations Research, 31, (1983), pp. 803-834.

/5/ A.M. GEOFFRION and R. McBRIDE
Lagrangean Relaxation Applied to Capacitated Facility Location Problems.
AIIE Transactions,10, (1978), pp. 40-47.

/6/ M. GUIGNARD and K. SPIELBERG
A Direct Dual Method for the Mixed Plant Location Problem with some Side Constraints.
Mathematical Programming, 17, (1979), pp. 198-228.

/7/ M.W. PADBERG
Covering, Packing and Knapsack Problems.
Annals of Discrete Mathematics, 4, (1979), pp. 265-287.

/8/ M.W. PADBERG
A Note on Zero-One Programming.
Operations Research, 23, (1975), 833-837.

/9/ M.W. PADBERG, T.J. VAN ROY and L.A. WOLSEY
Valid Inequalities for Fixed Charge Problems.
Core Discussion Paper, No. 8232, Université Catholique de Louvain, (1982).

/10/ L. SCHRAGE
Implicit Representation of Variable Upper Bounds in Linear Programming.
Mathematical Programming Study, 4, (1975), pp. 118-132.

/11/ L. SCHRAGE
Implicit Representation of Generalized Variable Upper Bounds in Linear Programming.
Mathematical Programming, 14, (1978), pp. 11-20.

/12/ K. SPIELBERG
Plant Location with Generalized Search Origin.
Management Science, 16, (1969), pp. 165-178.

/13/ T.J. VAN ROY
A Cross-Decomposition Algorithm for Capacitated Facility Location.
Working Paper, Afdeling Industrial Beleid, Katholieke Universiteit Leuven, (1983).