DEPARTEMENT TOEGEPASTE ECONOMISCHE WETENSCHAPPEN

ONDERZOEKSRAPPORT NR 9533

A Polyhedral Approach for the Generalized Assignment Problem

by

Zeger DEGRAEVE

Jurgen TISTAERT

Katholieke Universiteit Leuven

Naamsestraat 69, B-3000 Leuven

ONDERZOEKSRAPPORT NR 9533

A Polyhedral Approach for the Generalized Assignment Problem

by

Zeger DEGRAEVE
Jurgen TISTAERT

D/1995/2376/34

A Polyhedral Approach for the Generalized Assignment Problem

Zeger Degraeve

Jurgen Tistaert[1]

Katholieke Universiteit Leuven, Belgium

Department of Applied Economic Sciences
Naamsestraat 69
3000 Leuven

October 1995

[1]A word of gratitude towards Prof. Z. Degraeve and Prof. W. Gochet. The results in this paper wouldn't have been possible without letting me participate in enriching educations.

A Polyhedral Approach for the Generalized Assignment Problem

Zeger Degraeve

Jurgen Tistaert

Katholieke Universiteit Leuven, Belgium

Abstract:

The generalized assignment problem (*GAP*) consists of finding a maximal profit assignment of n jobs over m capacity constrained agents, whereby each job has to be processed by only one agent. This contribution approaches the *GAP* from the polyhedral point of view. A good upper bound is obtained by approximating the convex hull of the knapsack constraints in the *GAP*-polytope using theoretical work of Balas. Based on this result, we propose a procedure for finding close-to-optimal solutions, which gives us a lower bound. Computational results on a set of 60 representative and highly capacitated problems indicate that these solutions lie within 0.06% of the optimum. After applying some preprocessing techniques and using the obtained bounds, we solve the generated instances to optimality by branch and bound within reasonable computing time.

1. Introduction

In this paper, we report a new algorithmic result for the classical Generalized Assignment Problem (*GAP*). The *GAP*-polytope is defined in the unit hypercube and the optimization problem can be formulated as:

$$Max \sum_{j=1}^{n}\sum_{i=1}^{m} p_{ji}x_{ji}$$

subject to:

$$\sum_{i=1}^{m} x_{ji} = 1 \qquad for\ j = 1,\ldots,n \qquad (1)$$

$$\sum_{j=1}^{n} a_{ji}x_{ji} \le b_i \qquad for\ i = 1,\ldots,m \qquad (2)$$

$$x_{ji} \in \{0,1\} \qquad for\ \begin{matrix} j = 1,\ldots,n \\ i = 1,\ldots,m \end{matrix} \qquad (3)$$

where:

j = *index of job,*

i = *index of agent,*

x_{ji} = *1 if job j is processed by agent i ; 0 otherwise,*

p_{ji} = *profit of processing job j by agent i,*

a_{ji} = *resource consumption of job j processed by agent i,*

b_i = *resource availibility of agent i.*

Constraints (1) force each job to be processed by only one of the agents, while constraints (2) express the capacity restrictions on each agent. Constraints (3) enforce the ordinary dichotomy conditions on the decision variables. Fisher, Jaikumar and Van Wassenhove (1986) show that this problem is NP-hard, since the basic NP-hard two-partition problem is reducible to *GAP*.

The *GAP* appears as a subproblem in important real-life applications of mathematical programming. Examples include resource scheduling, design of communications networks (Grigordiadis, Tang and Woo, 1974), routing and distribution problems (Campbell and Langevin, 1995).

Several codes have been developed to tackle this problem. An excellent overview can be found in Cattrysse and Van Wassenhove (1992). Most algorithms are based on relaxation methods, set partitioning techniques and branch and bound procedures. The fastest and most powerful algorithms are from Ross and Soland (1975), Martello and Toth (1990), Fisher, Jaikumar and Van Wassenhove (1986) and Cattrysse, Salomon and Van Wassenhove (1994).

This contribution approaches the *GAP* from the polyhedral point of view and our procedure is outlined in the following sections. First, in section 2, we describe the cutting-plane algorithm that is used to get a good upper bound (*UB*). Based on this result, we propose in section 3 a LP-based methodology to obtain a new lower bound (*LB*). Computational experiments indicate that the bounds are quite tight. After applying some tailor-made preprocessing techniques reducing the size of the problem instances, optimality is reached by using a classical branch and bound (*B&B*) procedure in section 4. The report concludes with a formal discussion of computational results and gives some directions for future research.

2. Upper Bound calculations

In order to obtain a *UB*, define $\overline{GAP}$ as the LP relaxation of *GAP*. We construct a cutting-plane algorithm using strong valid inequalities derived from the m 0/1 knapsack constraints (2) in $\overline{GAP}$. Strong valid inequalities and facet defining inequalities were studied simultaneously by Balas (1975), Hammer, Johnson and Peled (1975) and Balas and Zemel (1978). Crowder, Johnson and Padberg (1983) applied these techniques successfuly on large-scale 0/1 programming problems. Consider an arbitrary capacity constraint i and define the solution space for this constraint as:

$$S_i = \left\{ \sum_{j=1}^{n} a_{ji} x_{ji} \leq b_i \; ; \; x_{ji} \in \{0,1\} \, , \, j = 1,\ldots,n \right\}$$

We assume a_{ji}, b_i integer and order the coefficients monotonically such that $a_{1i} \geq a_{2i} \geq a_{3i} \ldots \geq a_{ni}$ with $a_{1i} \leq b_i$. S_i is called an independence system and $\dim\left(CONV\left(S_i\right)\right) = n$. Let $N = \{1,..,n\}$, $M = \{1,..,m\}$ and $\mathbf{x_{j\bullet}} = \left\{x_{j1},..,x_{jm}\right\}$.

A cover C_i is a subset of N for which $\sum_{j \in C_i} a_{ji} > b_i$. This cover is minimal if all of its subsets are independent, or equivalently, if for each $k \in C_i$, $\sum_{j \in C_i} a_{ji} - a_{ki} \leq b_i$. A cover C_i leads to the valid inequality:

$$\sum_{j \in C_i} x_{ji} \leq |C_i| - 1$$

Generally, an inequality of this type can be strenghtened by lifting it to its full dimensionality in order to produce a facet. Using a sequential lifting procedure, one has to solve $|N \backslash C_i|$ knapsack problems to determine the lifting coefficients (e.g. using dynamic programming). Balas (1975) however, proposed the following procedure:

(1) Find a strong cover $C_i \subseteq N$ not yet considered; i.e., a set $C_i \subseteq N$ such that

(i) C_i is a minimal cover
(ii) if $E(C_i) \neq N$, then $\sum_{j \in (C_i - \{j_1\}) \cup \{j_2\}} a_{ji} \leq b_i$

where $a_{j_1} = \max_{j \in C_i} \{a_{ji}\}$, $a_{j_2} = \max_{j \in N - E(C_i)} \{a_{ji}\}$,
and $E(C_i) = C_i \cup \{j \in N - C_i : a_{ji} \geq a_{j_1}\}$

if there is none, stop. Otherwhise go to (2)

(2) Let $\alpha_{0i} = |C_i| - 1$ and define the coefficients α_{ji} and the index sets N_h^i, i.e.:

$$\alpha_{ji} = h \text{ for all } j \in N_h^i,\ h = 0,1,\ldots,q;$$
$$N_0^i = N - E(C_i), N_1^i = E(C_i) - \bigcup_{h=2}^{q} N_h^i;$$
$$N_h^i = \left\{ j \in E(C_i) : \sum_{j \in R_h} a_{ji} \leq a_{ji} < \sum_{j \in R_{h+1}} a_{ji} \right\},\ h = 2,\ldots,q$$

where R_h is the set of the first h elements of C_i, for $h = 2,\ldots,q+1$, $q = |C_i| - 1$.

Then the inequality

$$\sum_{j \in N} \alpha_{ji} x_{ji} \leq \alpha_{0i} \qquad (4)$$

is a valid cut (i.e., is satisfied by all $x \in vert(GAP)$); and if

$$\sum_{j \in C_i - R_{h+1}} a_{ji} + a_{pi} \leq b_i$$

holds for all $p \in N_h^i$, $h = 1,\ldots,q$, then (4) defines a facet of *GAP*.

This procedure enables us to determine immediately all the lifting coefficients α_{ji}. One can only apply this proposition, after having determined a strong cover to start lifting from. In order to find a cover, we first have to solve the following separation problem. Given a fractional solution $\theta_i = \{\overline{x_{ji}} | j = 1,\ldots,n\}$, we want to find a cover C_i for each agent i. We represent the unknown set C_i by the vector $Q_i = \{q_{1i}, q_{2i}, .., q_{ni}\} \in B^N$ and obtain the optimization problem F_i (Nemhauser and Wolsey, 1988):

$$Min \sum_{j \in N} (1 - x_{ji}^*) q_{ji}$$

subject to:

$$\sum_{j \in N} q_{ji} a_{ji} \geq b_i + 1$$

$$q_{ji} \in \{0,1\} \qquad j = 1,\ldots,n$$

From above, separation involves solving a 0/1 knapsack problem, using a *B&B* procedure or a dynamic programming recursion. In our code, we use the Gilmore-Gomory *B&B*-procedure implemented by Schrage (1987). After the cover is found, it is adjusted to make

it minimal. Then condition (ii) from the procedure of Balas for a strong cover is tested. Starting from the solution to the LP relaxation $\overline{GAP}$, we solve a separation problem to determine a minimal cover C_i and apply the procedure of Balas subsequently for each agent i. The resulting inequalities which cut off the current fractional solution are added to the problem and the extended formulation is reoptimized. We repeat this process until no inequality can be created which cuts off the current fractional solution. At this point, the objective function value of the LP-solution gives the *UB*.

3. Lower Bound calculations

To compute the *LB*, we develop an iterative LP-based procedure. Starting from the solution to the LP relaxation which generates the *UB*, all variables equal to one are fixed (assigned to that agent) and a reduced problem is created. The cutting-plane algorithm is then applied again to obtain the *UB* of the reduced problem. We repeat this process until the LP-solution resulting from the *UB*-calculation on the reduced problem only contains fractional assignments. At this moment, the reduced problem is solved by branch and bound (*B&B*). This method by iteratively reducing the problem provides us with close-to-optimal solutions. This heuristic can be formalized in the following five steps:

1. Set $LB = 0$. Let the current problem formulation $\overline{CP}$ be $\overline{GAP}$.

2. Compute the UB on $\overline{CP}$ as outlined in section 2. If the associated LP-solution $\theta = \{x_{ji} \mid i = 1,\ldots,m;\ j = 1,\ldots,n\}$ contains no decision variables x_{ji} with value 1, then goto step 4. If all variables are integral, let ξ be the objective value and goto step 5. Otherwise, let :

   $$J^* = \{j \in N:\ \exists\, i \in M \text{ such that } x_{ji} = 1 \text{ in } \theta\}$$
   $$X = \{x^*_{ji} : x^*_{ji} = 1 \text{ in } \theta\}$$

   X is the set of decision variables which are at their upperbound in the LP- solution θ, resulting from the UB-calculation. Having a $x^*_{ji} = 1$ in a LP relaxation means that job j is being assigned uniquely to agent i. J^* contains all indices of jobs j, which have been assigned uniquely in θ. Save θ and continue with step 3.

3. Delete all cuts from $\overline{CP}$ which were generated during the *UB*-calculation in step 2. Recall the solution θ from step 2 and alter $\overline{CP}$ by applying following substeps for each $x^*_{ji} \in X$:

   a) Detect and delete constraint $j \in J^*$ of type (1) in which x^*_{ji} occurs and eliminate the set of variables $\mathbf{x_{j\bullet}}$ belonging to constraint j from $\overline{CP}$.
   b) Since $x^*_{ji} = 1$ in θ and eliminated in (a), set $b_i = b_i - a^*_{ji}$
   c) Set $LB = LB + c^*_{ji}$

   Notice that each time we pass through step 3, we end up with the original formulation $\overline{GAP}$ in which at every turn sets $\mathbf{x_{j\bullet}}$ are being deleted for each $j \in J^*$. The capacity limits are being adjusted appropriately in (b). Define this altered problem as the new $\overline{CP}$. Continue with step 2.

4. Put a dichotomy condition on all variables in the $\overline{CP}$ and solve with *B&B*. Let ξ be the optimal objective value. Continue with step 5.

5. Set $LB = LB + \xi$. We obtain an upper bound.

4. Solution Scheme to Optimality

Optimality can be achieved by *B&B*, incorporating the information of the obtained bounds. However, even with tight bounds, strongly capacitated *GAP*'s tend to have large *B&B* trees. In order to reduce the size of these, we propose two powerful preprocessing techniques in sequence.

4.1 Preprocessing 1

A number of integer variables in *GAP* can be taken out from consideration, using following basic theorem from Nemhauser and Wolsey (1988):

Theorem 1:

Let RC_{ji} be the reduced cost of variable x_{ji} in the solution θ, corresponding to the *UB* calculation on $\overline{GAP}$. If $x_{ji} = 0$ and nonbasic in the solution θ and $RC_{ji} \geq UB - LB$, then there exists an optimal solution to the *GAP* with $x_{ji} = 0$.

With good bounds, theorem 1 allows us to eliminate a set of variables from the original formulation $\overline{GAP}$. If ρ equals the number variables which are left after applying preprocessing 1, then $[1-\frac{\rho}{n*m}] * 100$ expresses by how much we were able to reduce the original problemsize.

4.2 Preprocessing 2

After preprocessing 1, it may occur that there exist constraints of type (1) with only one variable x_{ji}^* left. In an optimal solution, this variable has to be equal to one, since i is the only agent where job j can be assigned to. This allows us to make some further reductions in the problemsize. Now, let $LB' = LB$ and $\delta = 0$. Consider:

$$R^* = \{Constraints\ j \in N\ with\ only\ one\ variable\ x_{ji}\ left\}$$

$$T^* = \{x_{ji} \mid j \in R^*\}$$

And perform for each $x_{ji} \in T^*$ each of the following tree operations:

(a) Eliminate constraint $j \in R^*$, corresponding to x_{ji} and delete x_{ji} from the formulation.

(b) Set $b_i = b_i - a_{ji}$.

(c) Set $LB' = LB' - c_{ji}$ and $\delta = \delta + c_{ji}$

When going through preprocessing 1 and 2, we are left with an adapted *GAP* formulation with $\rho - |T^*|$ variables. We define this reduced problem as GAP_{red}. *LB'* is the adapted lower bound for GAP_{red}. Before putting dichotomy conditions on the variables and turning to *B&B* to solve GAP_{red}, we calulate *UB'* on the relaxation $\overline{GAP_{red}}$, as outlined in section 2. Using *UB'* and *LB'*, we proceed by *B&B* to solve GAP_{red}. If φ is the optimal objective value, then $\varphi + \delta$ is the optimal objective value of our original formulation *GAP*. Figure 1 summarizes the approach, as outlined in foregoing sections.

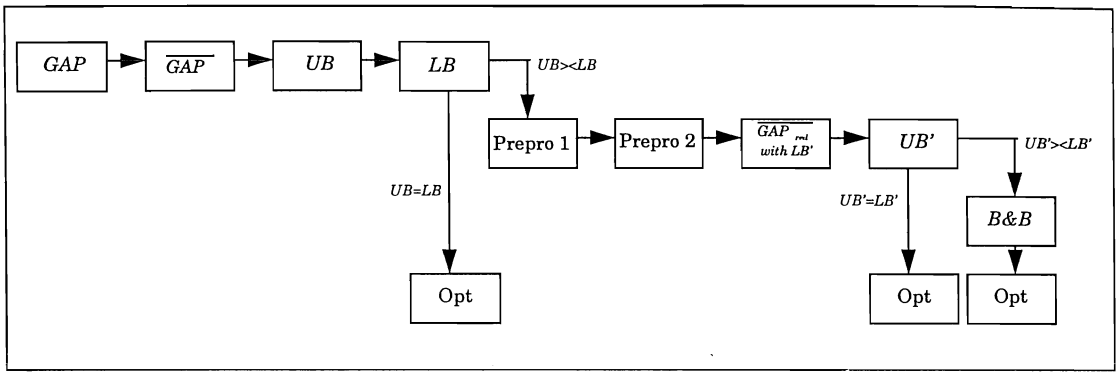

Figure 1: Solution scheme

5. Computational Results

The procedure is implemented using the NDP486 FORTRAN compiler V4.3.0 and linked with the LINDO library version 5.1 from Schrage (1987). All experiments were run on an IBM compatible 486 66 Mhz machine. We tested our procedure on a set of representative problems, which were provided in Cattrysse, Salomon and Van Wassenhove (1994). These datasets are known to be computationally difficult and give rise to highly capacitated ($\frac{1}{m}\sum_{j=1}^{n}\sum_{i=1}^{m} a_{ji} \approx \sum_{i=1}^{m} b_i$) *GAP*'s. They have following characteristics:

- $m \in \{5,8,10\}$ and the ratio $\frac{n}{m} \in \{3,4,5,6\}$. For each of the 12 possible combinations, 5 testsets were generated, yielding 60 problems in total, coded by problem C[m][n]-[1-2-3-4-5].

- the data were all drawn from a discrete uniform distribution (DU) with:

$c_{ji} \sim$ DU(15, 25)

$a_{ji} \sim$ DU(5, 25)

$b_i \sim \left(\frac{0.8}{m}\right)\sum_{j=1}^{n} a_{ji}$

There doesn't seem to be unanimity in the literature concerning maximizing or minimizing the objective value in computational experiments on *GAP*. Fisher, Jaikumar and Van Wassenhove (1986) and Martello and Toth (1990) generate instances on which they minimize the objective value, while Ross and Soland (1975) and Cattrysse and al. (1994) consider maximization problems. In order to make some comparisons with the recent procedures, we present computational results for maximization. We discuss results by means of summarizing tables. However, these averages are somewhat biased because of differences between 5 instances for one combination of n and m. Therefore, it is suggested to consult the detailed tables which were joined as an appendix.

Table I contains averages of the following statistics for each of the 12 types of datasets:

- $\left[\frac{UB - LB}{LB}\right] * 100$: relative deviation between *UB* and *LB*

- $\left[\frac{UB - Opt}{Opt}\right] * 100$: relative deviation of *UB* from the optimum
- $\left[\frac{Opt - LB}{LB}\right] * 100$: relative deviation of *LB* from the optimum

Our heuristic for the *LB* deviates on average 0.06% from the optimal solution. The bounds obtained by applying the procedure of Balas to get the *UB* deviate on average 0.38% from the optimum.

Table I: Average deviations from the optimum (in percent)

Problem	$\left[\frac{UB - LB}{LB}\right] * 100$	$\left[\frac{UB - Opt}{Opt}\right] * 100$	$\left[\frac{Opt - LB}{LB}\right] * 100$
C0515	0.842	0.842	0.000
C0520	0.605	0.558	0.046
C0525	0.352	0.352	0.000
C0530	0.639	0.547	0.092
C0824	0.535	0.499	0.035
C0832	0.345	0.345	0.000
C0840	0.232	0.147	0.085
C0848	0.515	0.337	0.177
C1030	0.536	0.422	0.113
C1040	0.441	0.293	0.147
C1050	0.189	0.172	0.017
C1060	0.152	0.097	0.055
Average	0.449	0.384	0.064

In table II, we measure the quality of our bounds (DT). Our *UB* is compared with the LP-relaxation, with the *UB* from Cattrysse, Salomon and Van Wassenhove (1994) (CSV). and with *UB* from Fisher, Jaikumar and Van Wassenhove (1986) (FJV). The *LB* is compared with the *LB* from CSV and with the heuristic solution of Martello and Toth (1990) (MT). The numbers present the relative deviations from the optimum for all bounds. The CSV-bounds dominate the other known results from the literature. Our *LB* is better than the *LB* from CSV. Our *UB* is not as tight as CSV, though is still very good when it is compared to the other procedures.

Table II: Quality of *LB*'s and *UB*'s (relative deviations from optimum)

	Lower Bound				Upper Bound		
Problem	MT	CSV	DT	LP	CSV	FJV	DT
C0515	3.781	0.061	0.000	2.989	0.169	3.82	0.842
C0520	2.168	0.190	0.046	1.832	0.118	2.21	0.558
C0525	1.478	0.035	0.000	0.847	0.007	1.86	0.352
C0530	1.484	0.214	0.092	1.158	0.156	2.38	0.547
C0824	2.180	0.072	0.035	1.544	0.132	1.96	0.499
C0832	2.450	0.053	0.000	1.078	0.122	1.68	0.345
C0840	1.805	0.000	0.085	0.599	0.059	1.44	0.147
C0848	2.174	0.195	0.177	0.586	0.071	2.68	0.337
C1030	2.186	0.141	0.113	1.288	0.143	2.05	0.422
C1040	1.704	0.189	0.147	0.732	0.052	1.58	0.293
C1050	1.320	0.017	0.017	0.489	0.053	0.79	0.172
C1060	1.092	0.028	0.055	0.287	0.037	0.75	0.097
Average	1.985	0.100	0.064	1.120	0.093	1.93	0.384

Table IIIa contains average CPU-times in seconds. Note that these numbers are cumulative. The CSV-procedure was implemented on an IBM387 16 Mhz. Although comparison of running times on different machines is difficult, IBM experts estimate the speed ratio between an IBM486 66 Mhz and an IBM387 16 Mhz at 11.
On average 180.7 seconds are needed to solve an instance to optimality. The CSV

procedure needed 788 seconds (i.e. 8665/11) on average to reach optimality. This implies that our CPU-times to reach the optimal solution are about **4 times faster** than CSV. This is due to the tightening effect of the cuts and the preprocessing techniques. Note that still 69% of total CPU time is put into the *B&B* process.

Table IIIa: Average CPU times in seconds (values are cumulative)

Problem	Upper Bound	Lower Bound	Optimum	% total CPU for UB	% total CPU for UB	% total CPU for *B&B*
C0515	0.60	3.00	5.80	0.103	0.414	0.483
C0520	0.20	1.80	3.80	0.053	0.421	0.526
C0525	1.00	2.20	4.00	0.250	0.300	0.450
C0530	0.80	2.00	12.40	0.065	0.097	0.839
C0824	1.80	20.60	46.60	0.039	0.403	0.558
C0832	3.20	7.60	25.80	0.124	0.171	0.705
C0840	4.00	10.20	46.80	0.085	0.132	0.782
C0848	3.40	35.20	904.60	0.004	0.035	0.961
C1030	4.00	22.20	47.20	0.085	0.386	0.530
C1040	4.80	525.80	843.80	0.006	0.617	0.377
C1050	5.40	13.80	73.80	0.073	0.114	0.813
C1060	11.60	20.40	153.80	0.075	0.057	0.867
Average	3.40	55.40	180.70	0.020	0.290	0.690

Table IIIb contains CPU-times to obtain the bounds. The CSV procedure needs about 26 seconds less to complete the bounding procedure. As noted before, we regain this loss of time by significantly reducing CPU times to obtain optimality.

Table IIIb: Comparison of average CPU-times for bounding procedure*

	Time to obtain LB and UB	
Problem	CSV	DT
C0515	0.89	3.00
C0520	1.95	1.80
C0525	3.29	2.20
C0530	6.70	2.00
C0824	2.78	20.60
C0832	8.57	7.60
C0840	10.00	10.20
C0848	33.08	35.20
C1030	13.33	22.20
C1040	33.97	525.80
C1050	65.58	13.80
C1060	175.54	20.40
Average	29.64	55.40

* Based on benchmark runs, an IBM486 66 Mhz is about 11 times faster than an IBM387 16 Mhz. To obtain comparable times, the entries in the CSV-column were divided by 11.

6. Conclusions and Directions for Future Research

Approaching the *GAP* from the polyhedral point of view seems to bear fruit in generating good bounds and finding close-to-optimal solutions on a set of computationally hard problems. It has the additional advantage to be time-efficient. Our bounds are competitive with those of CSV. At the moment, the authors are making an extensive time comparison between several algorithms on one machine. A further sharpening of the lowerbound would reduce the branching time considerably, though this is a NP-hard problem on itself.

Our procedure is LP-based and the generated heuristic solutions may be interesting vectors to be shot off in a column generation approach where the *GAP* appears as a subproblem. Several applications lend themselves to this purpose.

References

Balas, E. 1975. Facets of the knapsack polytope. *Mathematical Programming*, **8**, 146-164.

Balas, E. and Zemel, E. 1978. Facets of the Knapsack Polytope from Minimal Covers. *SIAM Journal on Applied Mathematics* , **34**, 119-148.

Campbell, J.F. and Langevin, A. 1995. The Snow Disposal Assignment Problem. *Journal of the Operational Research Society*, **46**, 919-929.

Cattrysse, D.G. and Van Wassenhove L.N. 1992. A survey of algorithms for the generalised assignment problems. *European Journal of Operational Research*, **60** (3), 260-272.

Cattrysse, D.G., Salomon, S. and Van Wassenhove L.N. 1994. A set partitioning heuristic for the generalised assignment problem. *European Journal of Operational Research*, **72** (1), 167-174.

Crowder H., Johnson, E.L. and Padberg, M. 1983. Solving Large-Scale Zero-One Linear Programming Problems. *Operations Research*, **31** (5), 803-834.

Fisher, M.L., Jaikumar, R. and Van Wassenhove, L.N. 1986. A multiplier adjustment method for the generalized assignment problem. *Management Science*, **32** (9), 1095-1103.

Grigordiadis, M.D., Tang, D.J. and Woo, L.S. 1975. Considerations in the optimal synthesis of some communications networks. *Paper presented at 45th joint National Meeting of ORSA/TIMS*, Boston, MA.

Hammer, P.L., Johnson, E.L. and Peled, U.N. 1975. Facets of Regular 0-1 Polytopes. *Mathematical Programming*, **8**, 179-206.

Martello, S. and Toth, P. 1990. Knapsack Problems, Algorithms and Computer Implementations. *Wiley Interscience Series in Discrete Mathematics and Optimization*, New York, NY.

Nemhauser, G.L. and Wolsey, L.A. 1988. Integer and Combinatorial Optimization. *Wiley Interscience Series in Discrete Mathematics and Optimization*, New York, NY.

Ross, G.T. and Soland, R.M. 1975. A branch and bound algorithm for the generalized assignment problem. *Mathematical Programming*, **8**, 91-103.

Schrage, L. 1987. Users manual for Linear, Integer and Quadratic Programming with LINDO. *The Scientific Press*, Redwood City, CA.

Appendix

Table IV: Detailed computational results (absolute values)

Problem	UB	LB	Opt	$\left[\frac{UB-LB}{LB}\right]*100$	$\left[\frac{UB-Opt}{Opt}\right]*100$	$\left[\frac{Opt-LB}{LB}\right]*100$
C0515-1	337	336	336	0.30	0.30	0.00
C0515-2	329	327	327	0.61	0.61	0.00
C0515-3	342	339	339	0.88	0.88	0.00
C0515-4	344	341	341	0.88	0.88	0.00
C0515-5	331	326	326	1.53	1.53	0.00
C0520-1	437	434	434	0.69	0.69	0.00
C0520-2	441	435	436	1.38	1.15	0.23
C0520-3	421	420	420	0.24	0.24	0.00
C0520-4	422	419	419	0.72	0.72	0.00
C0520-5	428	428	428	0.00	0.00	0.00
C0525-1	581	580	580	0.17	0.17	0.00
C0525-2	567	564	564	0.53	0.53	0.00
C0525-3	574	573	573	0.17	0.17	0.00
C0525-4	572	570	570	0.35	0.35	0.00
C0525-5	567	564	564	0.53	0.53	0.00
C0530-1	658	656	656	0.30	0.30	0.00
C0530-2	649	643	644	0.93	0.78	0.16
C0530-3	679	672	673	1.04	0.89	0.15
C0530-4	649	647	647	0.46	0.31	0.15
C0530-5	667	664	664	0.45	0.45	0.00
C0824-1	565	563	563	0.36	0.36	0.00
C0824-2	560	558	558	0.36	0.36	0.18
C0824-3	564	563	564	0.18	0.00	0.00
C0824-4	570	568	568	0.35	0.35	0.00
C0824-5	567	559	559	1.43	1.43	0.00
C0832-1	764	761	761	0.39	0.39	0.00
C0832-2	761	759	759	0.26	0.26	0.00
C0832-3	759	758	758	0.13	0.13	0.00
C0832-4	756	752	752	0.53	0.53	0.00
C0832-5	750	747	747	0.40	0.40	0.00
C0840-1	944	940	942	0.43	0.21	0.21
C0840-2	951	949	949	0.21	0.21	0.00
C0840-3	969	968	968	0.10	0.10	0.00
C0840-4	945	945	945	0.00	0.00	0.00
C0840-5	953	949	951	0.42	0.21	0.21
C0848-1	1136	1130	1133	0.53	0.26	0.27
C0848-2	1138	1134	1134	0.35	0.35	0.00
C0848-3	1143	1138	1141	0.44	0.18	0.26
C0848-4	1123	1116	1117	0.63	0.54	0.09
C0848-5	1131	1124	1127	0.62	0.35	0.27
C1030-1	711	706	709	0.71	0.28	0.42
C1030-2	720	717	717	0.42	0.42	0.00
C1030-3	714	712	712	0.28	0.28	0.00
C1030-4	724	723	723	0.14	0.14	0.00
C1030-5	713	705	706	1.13	0.99	0.14
C1040-1	959	958	958	0.00	0.00	0.00
C1040-2	968	962	963	0.62	0.52	0.10
C1040-3	963	958	960	0.52	0.31	0.21
C1040-4	948	944	947	0.42	0.11	0.32
C1040-5	952	946	947	0.63	0.53	0.11
C1050-1	1141	1139	1139	0.18	0.18	0.00
C1050-2	1180	1177	1178	0.25	0.17	0.08
C1050-3	1195	1195	1195	0.00	0.00	0.00
C1050-4	1174	1171	1171	0.26	0.26	0.00
C1050-5	1174	1171	1171	0.26	0.26	0.00
C1060-1	1452	1449	1451	0.21	0.07	0.14
C1060-2	1451	1449	1449	0.14	0.14	0.00
C1060-3	1435	1433	1433	0.14	0.14	0.00
C1060-4	1447	1446	1447	0.07	0.00	0.07
C1060-5	1448	1445	1446	0.21	0.14	0.07
Average in percentage				0.44	0.38	0.06

Table VI: Detailed CPU-times in seconds

Problem	time UB	time LB	time Opt	%UB	%LB	%Opt
C0515-1	1	1	1	0.00	0.00	0.00
C0515-2	1	1	2	0.50	0.00	0.50
C0515-3	1	1	2	0.50	0.00	0.50
C0515-4	0	2	4	0.00	0.50	0.50
C0515-5	0	10	20	0.00	0.50	0.50
C0520-1	0	2	6	0.00	0.33	0.67
C0520-2	0	3	5	0.00	0.60	0.40
C0520-3	1	2	3	0.33	0.33	0.33
C0520-4	0	2	5	0.00	0.40	0.60
C0520-5	0	0	0	0.00	0.00	0.00
C0525-1	1	2	3	0.33	0.33	0.33
C0525-2	1	1	2	0.50	0.00	0.50
C0525-3	1	2	3	0.33	0.33	0.33
C0525-4	1	2	5	0.20	0.20	0.60
C0525-5	1	4	7	0.14	0.43	0.43
C0530-1	1	2	5	0.20	0.20	0.60
C0530-2	1	3	25	0.04	0.08	0.88
C0530-3	1	1	16	0.06	0.00	0.94
C0530-4	1	3	7	0.14	0.29	0.57
C0530-5	0	1	9	0.00	0.11	0.89
C0824-1	1	8	9	0.11	0.78	0.11
C0824-2	2	4	7	0.29	0.29	0.43
C0824-3	2	3	4	0.50	0.25	0.25
C0824-4	2	6	11	0.18	0.36	0.45
C0824-5	2	82	202	0.01	0.40	0.59
C0832-1	2	5	24	0.08	0.13	0.79
C0832-2	2	10	19	0.11	0.42	0.47
C0832-3	3	7	10	0.30	0.40	0.30
C0832-4	4	6	43	0.09	0.05	0.86
C0832-5	5	10	33	0.15	0.15	0.70
C0840-1	5	22	168	0.03	0.10	0.87
C0840-2	4	8	17	0.24	0.24	0.53
C0840-3	1	4	5	0.20	0.60	0.20
C0840-4	5	5	6	0.83	0.00	0.17
C0840-5	5	12	38	0.13	0.18	0.68
C0848-1	2	7	1087	0.00	0.00	0.99
C0848-2	6	12	196	0.03	0.03	0.94
C0848-3	3	14	2418	0.00	0.00	0.99
C0848-4	3	8	454	0.01	0.01	0.98
C0848-5	3	135	368	0.01	0.36	0.63
C1030-1	9	62	104	0.09	0.51	0.40
C1030-2	4	18	50	0.08	0.28	0.64
C1030-3	2	7	14	0.14	0.36	0.50
C1030-4	3	7	11	0.27	0.36	0.36
C1030-5	2	17	57	0.04	0.26	0.70
C1040-1	9	11	11	0.82	0.18	0.00
C1040-2	2	51	169	0.01	0.29	0.70
C1040-3	7	493	757	0.01	0.64	0.35
C1040-4	4	12	406	0.01	0.02	0.97
C1040-5	2	2062	2876	0.00	0.72	0.28
C1050-1	9	17	61	0.15	0.13	0.72
C1050-2	5	22	38	0.13	0.45	0.42
C1050-3	2	3	3	0.67	0.33	0.00
C1050-4	5	14	42	0.12	0.21	0.67
C1050-5	6	13	225	0.03	0.03	0.94
C1060-1	6	12	141	0.04	0.04	0.91
C1060-2	3	13	89	0.03	0.11	0.85
C1060-3	12	20	179	0.07	0.04	0.89
C1060-4	21	30	81	0.26	0.11	0.63
C1060-5	16	27	279	0.06	0.04	0.90
Total	204	3324	10842			
Average	3.40	55.40	180.70	0.02	0.29	0.69

Table VIII: Detailed quality of obtained bounds

MT: Martello and Toth
CVS: Catrysse, Salomon and Van Wassenhove
LP: LP-relaxation

Problem	LP-UB	MT-LB	CSV-UB	CSV-LB	Our-UB	Our LB
C0515-1	2.26	8.74	0.30	0.00	0.30	0.00
C0515-2	3.79	2.51	0.00	0.31	0.61	0.00
C0515-3	3.15	3.35	0.15	0.00	0.88	0.00
C0515-4	2.76	1.79	0.00	0.00	0.88	0.00
C0515-5	2.99	2.52	0.40	0.00	1.53	0.00
C0520-1	2.31	1.64	0.23	0.00	0.69	0.00
C0520-2	2.50	4.31	0.00	0.23	1.15	0.23
C0520-3	1.22	3.19	0.21	0.00	0.24	0.00
C0520-4	2.23	1.70	0.14	0.72	0.72	0.00
C0520-5	0.89	0.00	0.00	0.00	0.00	0.00
C0525-1	0.50	1.22	0.00	0.00	0.17	0.00
C0525-2	0.93	3.49	0.00	0.00.	0.53	0.00
C0525-3	0.82	0.17	0.00	0.17	0.17	0.00
C0525-4	0.74	0.88	0.00	0.00	0.35	0.00
C0525-5	1.25	1.62	0.04	0.00	0.53	0.00
C0530-1	0.97	0.61	0.11	0.61	0.30	0.00
C0530-2	1.70	1.90	0.39	0.00	0.78	0.16
C0530-3	1.33	2.12	0.21	0.15	0.89	0.15
C0530-4	0.78	1.25	0.08	0.31	0.31	0.15
C0530-5	1.01	1.53	0.00	0.00	0.45	0.00
C0824-1	1.00	1.99	0.18	0.00	0.36	0.00
C0824-2	1.26	2.39	0.00	0.00	0.36	0.18
C0824-3	0.86	2.17	0.02	0.00	0.00	0.00
C0824-4	2.09	2.34	0.14	0.00	0.35	0.00
C0824-5	2.56	2.01	0.32	0.36	1.43	0.00
C0832-1	0.95	1.06	0.16	0.00	0.39	0.00
C0832-2	1.01	2.29	0.13	0.26	0.26	0.00
C0832-3	0.97	3.55	0.07	0.00	0.13	0.00
C0832-4	1.48	3.30	0.13	0.00	0.53	0.00
C0832-5	0.98	2.05	0.12	0.00	0.40	0.00
C0840-1	0.67	1.18	0.12	0.00	0.21	0.21
C0840-2	0.64	0.96	0.06	0.00	0.21	0.00
C0840-3	0.43	1.57	0.00	0.00	0.10	0.00
C0840-4	0.54	3.17	0.00	0.00	0.00	0.00
C0840-5	0.71	2.15	0.12	0.00	0.21	0.21
C0848-1	0.49	0.89	0.05	0.27	0.26	0.27
C0848-2	0.69	3.28	0.16	0.00	0.35	0.00
C0848-3	0.38	2.33	0.00	0.44	0.18	0.26
C0848-4	0.82	2.10	0.13	0.00	0.54	0.09
C0848-5	0.56	2.27	0.01	0.27	0.35	0.27
C1030-1	1.28	1.43	0.14	0.42	0.28	0.42
C1030-2	1.35	3.31	0.06	0.14	0.42	0.00
C1030-3	1.36	1.14	0.15	0.14	0.28	0.00
C1030-4	0.74	1.83	0.14	0.00	0.14	0.00
C1030-5	1.70	3.22	0.23	0.00	0.99	0.14
C1040-1	0.49	1.48	0.00	0.00	0.00	0.00
C1040-2	1.06	2.01	0.11	0.21	0.52	0.10
C1040-3	0.77	1.69	0.01	0.21	0.31	0.21
C1040-4	0.39	2.05	0.00	0.21	0.11	0.32
C1040-5	0.95	1.28	0.14	0.32	0.53	0.11
C1050-1	0.53	1.24	0.00	0.00	0.18	0.00
C1050-2	0.50	1.38	0.03	0.08	0.17	0.08
C1050-3	0.21	0.59	0.02	0.00	0.00	0.00
C1050-4	0.74	1.91	0.10	0.00	0.26	0.00
C1050-5	0.47	1.47	0.12	0.00	0.26	0.00
C1060-1	0.21	1.11	0.00	0.00	0.07	0.14
C1060-2	0.33	0.98	0.06	0.07	0.14	0.00
C1060-3	0.27	1.20	0.03	0.00	0.14	0.00
C1060-4	0.21	0.91	0.05	0.00	0.00	0.07
C1060-5	0.41	1.26	0.04	0.07	0.14	0.07
Average	1.12	1.99	0.09	0.10	0.38	0.06