RELAXING THE QUADRATIC SEMI-ASSIGNMENT PROBLEM TO DERIVE LOWER BOUNDS*

Giorgio Gallo, *Dipartimento di Informatica, Università di Pisa*
Elena Moretti Tomasin, *Dipartimento di Matematica Applicata ed Informatica, Università di Venezia*.
Annamaria Sorato, *Dipartimento di Matematica Applicata ed Informatica, Università di Venezia*.

Revised November 1991

Abstract - The problem of assigning elements to sets in order to minimize a quadratic cost function is considered. This problem is well known to belong to the class of NP-hard problems. Here we focus on the computation of lower bounds to its objective function. The main contribution of the paper is the introduction of a relaxed problem from which different bounds can be derived in polynomial time.

Key Words: *Quadratic semi-assignment problem - Scheduling - Clustering - Lower Bounds*.

---

*Partially supported by Consiglio Nazionale delle Ricerche, Progetto Finalizzato Trasporti.

Giorgio Gallo, Dipartimento di Informatica, Università di Pisa, Corso Italia 40, 56100 Pisa (Italy) - email: gallo@di.unipi.it

1.Introduction

The Quadratic Semi-Assignment problem is to assign items to sets in such a way that a quadratic function is minimized. Let N be a set containing n elements; we want to partition it into m<n (possibly empty) subsets. Let $b_{ik} \geq 0$ be the cost of assigning element i to subset k, and $a_{ij} > 0$ be the interaction cost incurred if elements i and j are assigned to the same subset. It is easy to see that the non-negativity assumption on the coefficient b_{ik} is not binding, in fact it is enough to add a properly chosen constant to get positive coefficients. A partition is sought which minimizes the total cost:

$$\text{Min } \sum_k \sum_{i<j} a_{ij} x_{ik} x_{jk} + \sum_k \sum_i b_{ik} x_{ik}$$

$$\text{(QSA)} \qquad \sum_k x_{ik} = 1, \quad i=1,\ldots,n,$$

$$x_{ik} \in \{0,1\}, \quad i=1,\ldots,n, \quad k=1,\ldots,m.$$

QSA has interesting practical interpretations in many fields, such as distributed computation, cluster analysis and scheduling. Originally, the present work has been motivated by a scheduling problem arising in the management of ports: the problem of optimally assigning ships to docks, in order to minimize the interaction cost due to the fact that ships compete for the use of the port resources.

Although it has the nice property that its continuous relaxation has an integer solution [7], QSA is a quite difficult problem; in fact it belongs to the set of NP-hard problems. Also the problem of finding an e-approximate solution cannot be solved in polynomial time unless P=NP [8],[3]. Branch&Bound algorithms for its solution have been proposed by Anthonisse in [1] and by Greenberg in [4]. A related, although different, problem has been recently solved by means of polyhaedral techniques by Grötschel & Wakabayashi [5].

In the following we shall deal with the derivation of lower bounds for QSA, which is a crucial point both in the design of exact algorithms and in the analysis of heuristic ones.

2. Two simple bounds

A rather trivial bound (and shallow indeed) can be obtained by simply deleting the quadratic term from the objective function. This way, the problem is easily solved by setting:

$$x_{ik} = \begin{cases} 1, & \text{if } k=k(i), \\ 0, & \text{otherwise,} \end{cases}$$

where k(i) is such that $b_{ik(i)} = \text{Min}\{b_{ik} : k=1,...,m\}$.

This bound has been used in [1], and will be referred to as LB_0.

The bound LB_0 can be improved as follows. Let r be a lower bound on the number of pairs (i,j) such that $x_{ik}= x_{jk} = 1$ for some k in a feasible solution, i.e.

$r = \text{Min} \{\sum_k \sum_{i<j} x_{ik}x_{jk} : \sum_k x_{ik}=1 \ ,i=1,...,n; \ x_{ik} \in \{0,1\}, \ i=1,...,n, \ k=1,...,m\}$.

Let $a_{i(1)j(1)} \le a_{i(2)j(2)} \le ... \le a_{i(q)j(q)}$, with $q=n(n-1)/2$, be the elements a_{ij} in non-decreasing order. The the following proposition holds true:

Proposition 2.1 $LB_1 = LB_0 + \sum_{1 \le h \le r} a_{i(h)j(h)}$ is a lower bound for QSA.

The bound r can be easily obtained by solving the following problem:

$r=\text{Min} \{(1/2)\sum_k u_k(u_k-1) : \sum_k u_k=n \ , u_k \in \mathbf{Z}_+, \ k=1,...,m\}$,

which has as an optimal solution

$$u^*_k = \begin{cases} \lfloor n/m \rfloor, & k=1,\ldots,m-p, \\ \lceil n/m \rceil, & k=m-p+1,\ldots,m, \end{cases}$$

where $p=n \bmod(m)$, $\lfloor a \rfloor$ is the maximum integer not greater than a, and $\lceil a \rceil$ is the minimum integer not smaller than a.

LB_0 is evaluated in $O(n^2)$ time while LB_1 is evaluated in $(n^2 \log n)$ time, since the n(n-1)/2 reals a_{ij} need to be sorted. Once the elements have been sorted, the cost of computing LB_1, given LB_0, is O(m).

3. A relaxed problem

Now a relaxation of QSA is presented, from which new bounds can be derived. Let $\beta_k = \min\{b_{ik} : i=1,...,n\}$, and assume w.l.o.g. that the subsets are ordered in such a way that $\beta_1 \le \beta_2 \le ... \le \beta_m$. Then problem QSA can be relaxed into the following:

$$(R_1) \qquad \begin{aligned} \text{Min} \ & \sum_k \sum_{i<j} a_{ij}x_{ik}x_{jk} + \sum_k \beta_k(\sum_i x_{ik}) \\ & \sum_k x_{ik} = 1 , \qquad i=1,\ldots,n, \\ & x_{ik} \in \{0,1\} , \qquad i=1,...,n, \quad k=1,...,m. \end{aligned}$$

Proposition 3.1 $v(R_1) \le v(QSA)$.

By v(P) we denote the optimal value of the objective function of problem P.

Proposition 3.2 An optimal solution to R_1, say x*, exists such that $h>k \Rightarrow \sum x^*_{ih} \leq \sum x^*_{ik}$.

Define

$$\alpha_i = \min\{a_{\pi(i)\pi(j)} : \pi(j) > \pi(i)\},\ i = 1, ...,n-1, \qquad (1)$$

where $\pi(1), \pi(2), ..., \pi(n-1)$ is any permutation of the indices 1,2, ..., n-1. We assume $a_{ij} = a_{ji}$ when $i > j$.

In the following we shall denote by $(z_{(1)}, z_{(2)}, ..., z_{(n)})$ the rearrangement of vector $(z_1, z_2, ..., z_n)$ in non-decreasing order of its components, i.e. $z_{(1)} \leq z_{(2)} \leq ... \leq z_{(n)}$. Now we can introduce the following problem, where k ranges from 1 to m:

$$\begin{aligned} &\text{Min } \sum_k \sum_{i<j} \alpha_{(k)} x_{ik} x_{jk} + \sum_k \beta_k (\sum_i x_{ik}) \\ (R_2) \qquad &\sum_k x_{ik} = 1, \qquad i=1,\ldots,n, \\ &x_{ik} \in \{0,1\}, \qquad i=1,...,n, \quad k=1,...,m. \end{aligned}$$

Theorem 3.1 $v(R_2) \leq v(R_1) \leq v(QSA)$.

Proof. Let x* be an optimal solution of R_1 which satisfies the conditions of proposition 3.2. The theorem is proved if we show that

$$\sum_k \sum_{i<j} \alpha_{(k)} x^*_{ik} x^*_{jk} \leq \sum_k \sum_{i<j} a_{ij} x^*_{ik} x^*_{jk};$$

in fact:

$$v(R_2) \leq \sum_k \sum_{i<j} \alpha_{(k)} x^*_{ik} x^*_{jk} + \sum_k \beta_k (\sum_i x^*_{ik}),$$

$$\sum_k \sum_{i<j} a_{ij} x^*_{ik} x^*_{jk} + \sum_k \beta_k (\sum_i x^*_{ik}) = v(R_1) \leq v(QSA).$$

One has

$$\sum_k \sum_{i<j} a_{ij} x^*_{ik} x^*_{jk} - \sum_k \sum_{i<j} \alpha_{(k)} x^*_{ik} x^*_{jk} = \sum_{k=1,t} \sum_{I(k)} (a_{ij} - \alpha_{(k)}),$$

where $I(k) = \{(i,j): x^*_{ik} = x^*_{jk} = 1\}$, k=1,...,t, and $t = \max\{k : \sum_i x^*_{ik} \geq 2\}$.

Let us define $a^*_k = \min\{a_{ij}: (i,j) \in I(k)\}$, k=1, ..., t. Since $(i,j) \in I(k) \Rightarrow (i,q) \notin I(h)$ for all q and $h \neq k$, it is $a^*_{(k)} \geq \alpha_{(k)}$, k = 1,...,t. Then:

$$\sum_{k=1,t} \sum_{I(k)} (a_{ij} - \alpha_{(k)}) \geq \sum_{k=1,t} \sum_{I(k)} (a^*_k - \alpha_{(k)}) = \sum_{k=1,t} |I(k)| (a^*_k - \alpha_{(k)}) =$$

$$\sum_{k=1,t} |I(k)| a^*_k - \sum_{k=1,t} |I(k)| \alpha_{(k)} \geq \sum_{k=1,t} |I(k)| a^*_{(k)} - \sum_{k=1,t} |I(k)| \alpha_{(k)} \geq 0,$$

where the second inequality derives from the fact that for any two vectors a and b it is:

$$\sum_{i=1,n} a_i b_i \geq \sum_{i=1,n} a_{(i)} b_{(n-i+1)},$$

and by hypothesis $|I(1)| \geq |I(2)| \geq ... |I(m)|$. The proof is so completed.◊

Note that R_2 is not a relaxation of QSA in the usual way. In fact it might happen that for some feasible (possibly optimal) solution, say x', it is:

$$\sum_k \sum_{i<j} \alpha_{(k)} x'_{ik} x'_{jk} + \sum_k \beta_k (\sum_i x'_{ik}) > \sum_k \sum_{i<j} a_{ij} x'_{ik} x'_{jk} + \sum_k \sum_i b_{ik} x'_{ik}.$$

The objective function of R_2 does not depend on the indices of the elements assigned to the subsets: it depends only on the number of elements assigned to each subset. Then if y_k denotes the cardinality of set k , we may restate problem R_2 as follows:

$$\text{(QR)} \qquad \begin{aligned} &\text{Min } (1/2)\sum_k \alpha_{(k)} y_k (y_k - 1) + \sum_k \beta_k y_k \\ &\sum_k y_k = n, \\ &y_k \geq 0, \text{ integer, } k=1,...,m. \end{aligned}$$

Clearly $v(QR) = v(R_2) \leq v(QSA)$.

From QR two distinct lower bounds can be derived, $LB_2 = v(QR_c)$, and $LB_3 = v(QR)$, where QR_c is the continuous relaxation of QR. The problem of computing such bounds is addressed in the following sections. Notice that different permutations yield different coefficient α_i, and hence different bounds. The problem of finding ther permutation π which yields the sharpest bound for any given instance of QSA shall not be addressed here. Sharper bounds might be obtained by deriving the above introduced relaxations from an equivalent QSA problem in which, given any real γ_i , i = 1,…,n, the objective function has been transformed as follows:

$$\begin{aligned} &\sum_k \sum_{i<j} a_{ij} x_{ik} x_{jk} + \sum_k \sum_i b_{ik} x_{ik} \\ &= \sum_k \sum_{i<j} a_{ij} x_{ik} x_{jk} + \sum_k \sum_i (b_{ik} - \gamma_i) x_{ik} + \sum_k \sum_i \gamma_i x_{ik} \qquad (2) \\ &= \sum_k \sum_{i<j} a_{ij} x_{ik} x_{jk} + \sum_k \sum_i (b_{ik} - \gamma_i) x_{ik} + \sum_i \gamma_i \\ &= \sum_k \sum_{i<j} a_{ij} x_{ik} x_{jk} + \sum_k \sum_i b'_{ik} x_{ik} + K, \end{aligned}$$

where the second equality derives from the constraint $\sum_k x_{ik} = 1$ (i = 1,…,n), $b'_{ik} = b_{ik} - \gamma_i$ (i = 1,…,n, k = 1,…,m), and $K = \sum_i \gamma_i$. A reasonable choice for γ_i , i = 1,…,n, is:

$\gamma_i = \min\{b_{ik}: k = 1,…,m\}, \quad i = 1,…,n.$

In the following, we shall denote by $\overline{LB}_2$ and $\overline{LB}_3$ the lower bounds LB_2 and LB_3 when the above described transformation is used. For the sake of simplicity, since the only effect of the transformation is to add a constant to the objective function, in the following

two sections we refer only to LB_2 and LB_3, although all the results presented apply to $\overline{LB}_2$ and $\overline{LB}_3$ as well.

4. Evaluation of LB_2

Let us introduce the relaxation of QR_c obtained by dropping the non-negativity constraints:

$$\text{Min } (1/2)\sum_k \alpha_{(k)} y_k(y_k-1) + \sum_k \beta_k y_k$$
$$(QR'_c) \qquad \sum_k y_k = n,$$
$$y_k \in \mathbf{R},\ k=1,...,m.$$

Since the objective function is strictly convex, such a problem has a closed form solution, y', with:

$$y'_k = 1/2 + (\mu'-\beta_k)/\alpha_{(k)},$$

where

$$\mu' = ((n-m/2) + \sum_k(\beta_k/\alpha_{(k)}))/A, \text{ and } A = \sum_k(1/\alpha_{(k)}). \qquad (3)$$

Theorem 4.1 - If y' and y" are optimal solution of QR'_c and of QR_c respectively, then $y'_k \le 0 \Rightarrow y''_k = 0$, k=1, ..., m.

Proof : The solution y" is obtained by solving the following complementarity system

$$\alpha_{(k)} y_k - \mu - u_k = \alpha_{(k)}/2 - \beta_k \text{ , } k = 1, ..., m,$$
$$\sum_k y_k = n,$$
$$y_k u_k = 0, \ k = 1, ..., m, \qquad (4)$$
$$y_k, u_k \ge 0, \ k = 1, ..., m.$$

First observe that $y'' > 0 \Rightarrow y' > 0$, in fact this is the case when the unconstrained optimum and the constrained one coincide. Then if y' has some negative component there must be one zero component at least in y". Assume this to be the case and define $M_1 = \{k: y''_k=0\}$, $M_2 = \{k: y''_k>0\}$, $m_1=|M_1|$, $m_2=|M_2|$, $A=\sum_{k\in M}(1/\alpha_{(k)})$, and $A_2= \sum_{k\in M_2}(1/\alpha_{(k)})$. Then the solution to system (4) can be written as follows:

$$y''_k = \begin{cases} 0, & k\in M_1, \\ 1/2+(\mu''-\beta_k)/\alpha_{(k)}, & k\in M_2, \end{cases}$$
$$\mu'' = (n-m_2/2 + \sum_{k\in M_2} (\beta_k/\alpha_{(k)}))/ A_2,$$

$$u''_k = -\alpha_{(k)}/2 - \mu'' + \beta_k \geq 0 \quad (= 0 \text{ if } k \in M_2).$$

By proper algebraic manipulations it is possible to show that $\mu' \geq \mu''$.

Since $1/2 + (\mu''-\beta_k)/\alpha_{(k)} > 0$, $k \in M_2$

it must be $y'_k = 1/2 + (\mu'-\beta_k)/\alpha_{(k)} > 0$, $k \in M_2$.

We have proved so far that $y''_k > 0 \Rightarrow y'_k > 0$, from which the thesis follows.◊

Theorem 4.1 provides a simple way to solve QR_c wth $O(m^2)$ time complexity; in fact the number of unconstrained problems to be solved is bounded by m, and each of them can be solved in O(m) time. Actually, to evaluate LB_2, problem QR must be set up. That requires the evaluation and the sorting of coefficients α_k and β_k, which can be accomplished in $O(n^2)$ time. But, at least in Branch and Bound algorithms this work can be done just once at the very beginning.

Since the relaxed problem QR'_c has an explicit solution, it is possible to compute the difference between LB_2 and $\bar{LB}_2$. In fact, by simple algebraic manipulations, one obtains:

$$LB_2 - \bar{LB}_2 = \frac{1}{2} \sum_k (\beta_k - \bar{\beta}_k)(y_k - \bar{y}'_k) - K$$

where $\bar{\beta}_k$ and $\bar{y}'_k$ correspond in the transformed problem to b_k and to y'_k respectively. It can be noted that $LB_2 - \bar{LB}_2$ can be either positive or negative.

5. Evaluation of LB_3

Let us define the following convex piecewise linear functions:

$$f_k(y_k) = \begin{cases} +\infty, \text{ if } y_k < 0, \\ (1/2)\alpha_{(k)} y_k (y_k - 1) + \beta_k y_k, \text{ if } y_k \geq 0 \text{ and integer, } k = 1, \ldots, m, \\ (\lceil y_k \rceil - y_k) f(\lfloor y_k \rfloor) + (y_k - \lfloor y_k \rfloor) f(\lceil y_k \rceil), \text{ otherwise.} \end{cases}$$

Clearly QR can be rewritten as:

(QR*) $\quad \text{Min}\{ \sum_k f_k(y_k) : \sum_k y_k = n,\ y_k \geq 0 \text{ integer}, k = 1, \ldots, m\}.$

Since $f_k(y_k)$ is convex for all k [6], then necessary and sufficient condition for a feasible vector y^* to be optimal is that a real λ^* exists such that

$$0 \in \partial L(y^*, \lambda^*)/\partial y_k , \quad k = 1, \ldots, m, \qquad (5)$$

where $\partial L(y,\lambda)/\partial y_k$ is the subdifferential at (y,λ) of the lagrangean function

$$L(y,\lambda) = \sum_k f_k(y_k) - \lambda(\sum_k y_k - n).$$

In our problem conditions (5) become:

$$\alpha_{(k)}(y^*_k - 1) + \beta_k \le \lambda^* \le \alpha_{(k)} y^*_k + \beta_k, \quad y^*_k \ge 1, \qquad (6)$$
$$\lambda^* \le \beta_k, \quad y^*_k = 0.$$

The following results will be used in order to devise a fast algorithm to solve QR*, i.e. to evaluate LB_3.

Proposition 5.1 - If (y^*,λ^*) satisfies the optimality conditions (6) then

$$\beta_1 \le \lambda^* \le \alpha_{(1)} n + \beta_1 .$$

Proof : The proof follows from the fact that an index h must exist such that $y^*_h \ge 1$ and $\beta_1 \le \alpha_{(h)}(y^*_h - 1) + \beta_h \le \lambda^*$, moreover $\lambda^* \le \alpha_{(1)} y^*_1 + \beta_1 \le \alpha_{(1)} n + \beta_1$.◊

Proposition 5.2 - An optimal (y^*,λ^*) exists such that, for some k,

$$\text{either } \lambda^* = \alpha_{(k)}(y^*_k - 1) + \beta_k \text{ or } \lambda^* = \alpha_{(k)} y^*_k + \beta_k .$$

Theorem 5.1 - Given λ^*, a necessary and sufficient condition for the existence of a feasible vector y^* such that (y^*, λ^*) satisfies the optimality conditions (6) is that :

$$\sum_k \lceil (\lambda^* - \beta_k)/\alpha_{(k)} \rceil \le n \le m + \sum_k \lfloor (\lambda^* - \beta_k)/\alpha_{(k)} \rfloor \qquad (7)$$

Proof : (if) Assume that (7) is satisfied, and define $v_k = \lceil (\lambda - \beta_k)/\alpha_{(k)} \rceil$, and $u_k = 1 + \lfloor (\lambda - \beta_k)/\alpha_{(k)} \rfloor$. Then any y such that $v_k \le y_k \le u_k$ satisfies the optimality conditions, and at least one of such y exists which is feasible.

(only if) If the theorem conditions are not satisfied, then a k must exist such that either $v_k > y_k$ or $u_k < y_k$. In the former case one has:

$$\lceil (\lambda - \beta_k)/\alpha_{(k)} \rceil > y_k \text{ and } y_k \text{ integer} \Rightarrow (\lambda - \beta_k)/\alpha_{(k)} > y_k ,$$

and then $\alpha_{(k)} y_k + \beta_k < \alpha_{(k)}[(\lambda - \beta_k)/\alpha_{(k)}] + \beta_k = \lambda$; in the latter case:

$$1 + \lfloor (\lambda - \beta_k)/\alpha_{(k)} \rfloor < y_k \text{ and } y_k \text{ integer} \Rightarrow 1 + (\lambda - \beta_k)/\alpha_{(k)} < y_k ,$$

and then $\alpha_{(k)}(y_k - 1) + \beta_k > \alpha_{(k)}[(\lambda - \beta_k)/\alpha_{(k)}] + \beta_k = \lambda$.

In both cases the optimality conditions do not hold, and the proof is so completed.◊

From the previous results an efficient way to solve problem QR* and hence to compute LB_3 can be easily derived. It is enough to compute a value for λ which satisfies condition (7) ans hence to derive accordingly an optimal vector y. Such operations are performed by procedures OPT-LAMBDA and OPT-Y, which are described next.

```
Procedure OPT-LAMBDA (α,β,λ,v,u,n_min)
    begin
        λ_min:=lower :=β1; λ_max:=upper :=α(1)n+β1; optimal :="false";
        repeat
            λ:=(λ_max+λ_min)/2; n_min:=0; n_max:=0; L_max: = upper ;L_min= lower ;
            for k:=1 to m do
                begin
                    if (λ>β_k) then begin
                            u[k]:= 1 + ⌊(λ-β_k)/α_(k)⌋; n_max:=n_max + u[k];
                            v[k]:= ⌈(λ-β_k)/α_(k)⌉; n_min:=n_min + v[k] end;
                    else u[k]:=v[k]:=0;
                        if (L_min < α_(k)(u[k]-1)+β_k) then L_min := α_(k)(u[k]-1)+β_k ;
                        if (L_max > α_(k)u[k]+β_k) then L_max := α_(k)u[k]+β_k
                end;
            if (n_min ≤ n ≤ n_max) then optimal:="true";
            if (n_min > n) then λ_max:=L_max;
            if (n_max < n) then λ_min:=L_min;
        until optimal = "true"
    end;

Procedure OPT-Y (v,u,n_min,y)
    begin
        Δ:=n - n_min;
        for k:= 1 to m do
            begin
                if (u[k]>v[k] and Δ>0) do
                    begin y[k]:=u[k]; Δ:=Δ - 1 end;
                else y[k]:=v[k]
            end
    end;
```

Theorem 5.2 - The proposed algorithm is correct and runs in O(mlogn) time .

Proof : First observe that when the variable *optimal* takes value "true" in OPT-LAMBDA then λ satisfies conditions (7), and hence the solution provided by the algorithm is optimal.The existence of a λ satisfying conditions (7) in the interval $[\lambda_{min},\lambda_{max}]$ is assured by the fact that at each step, as it can be easily shown, one has:

$$\sum_k \lceil(\lambda_{min}-\beta_k)/\alpha_{(k)}\rceil \le n \le m + \sum_k \lfloor(\lambda_{max}-\beta_k)/\alpha_{(k)}\rfloor . \qquad (8)$$

To show that the algorithm terminates in a finite number of steps, define the sets $\Lambda_k = \{\alpha_{(k)}i+\beta_k : i=1, ...,n,\ \lambda_{min} \le \alpha_{(k)}i+\beta_k \le \lambda_{max}\}$, k=1, ...,m; clearly, at each iteration the cardinality of at least one set Λ_k is strictly decreased. At the beginning it is $\lambda_{max} - \lambda_{min} = n\ \alpha_{(1)}$; then, after $\lceil\log n\rceil$ iteration at most, we get $\lambda_{max} - \lambda_{min} \le \alpha_{(1)}$. At this point , since $\alpha_{(k)} \ge \alpha_{(1)}$,it is $|\Lambda_k| \le 1$ for each k. Then, at most m-1 iterations are now required to termination. In fact when $|\Lambda_k| = \emptyset$ for each k, an optimal λ has been obtained since in this case one has

$$\lceil(\lambda_{min}-\beta_k)/\alpha_{(k)}\rceil = \lceil(\lambda-\beta_k)/\alpha_{(k)}\rceil = 1 + \lfloor(\lambda-\beta_k)/\alpha_{(k)}\rfloor = 1 + \lfloor(\lambda_{max}-\beta_k)/\alpha_{(k)}\rfloor ,$$

and then, because of (8), (7) is satisfied with a strict equality.

To complete the proof it is enough to observe that each iteration of OPT-LAMBDA requires O(m) operations, then the algorithm runs in O(mlogn) since the cost of procedure OPT-Y is O(m).◊

6. Conclusions

A numerical experimentation has been carried out in [2] to compare the lower bounds presented in the previous sections. The relative value of the bounds with respect to LB_0 was considered testing subproblems of the same kind produced by enumerative methods. The results obtained can be briefly summarized as follows:

i) the bounds are generally quite sharp up to a few hundred percent over LB_0;

ii) the relative behavior of the bounds strongly depends on the cost range and on the level of the search tree. For example for cost range [50,100] LB_2, LB_3, $\overline{LB}_2$ and $\overline{LB}_3$ are almost always sharper then LB_1; for cost range [10,100] the opposite is often true depending on the level of the search tree;

iii) for what concernes the computational cost, no significant difference has been observed in the times needed to compute the bounds from LB_0 to $\overline{LB}_3$.

The results suggest that the bounds could be useful in the design of branch & bound algorithms and that it might be proper to choose a dynamic policy to decide which of the bounds is to be used according to the costs and to the level of the search tree.

References

[1] Anthonisse, J.M., "An optimal partitioning", Stiching Mathematisch Centrum, Amsterdam (1968).

[2] Gallo, G., E. Moretti Tomasin, and A. Sorato, "Lower bounds for the quadratic semi-assignment problem", Dipartimento di Informatica, Università di Pisa (1989).

[3] Gonzales, T.F., "On the computational complexity of clustering and related problems", in *System Modeling and Optimization*, Drenik&Kozin, Editor, Springer Verlag (1982).

[4] Greenberg, H., "A quadratic assignment problem without column constraints", *Nav. Res. Logistic Quart.* 16, 417-422 (1969).

[5] Grötschel, M. and Y. Wakabayashi, "A cutting plane algorithm for a clustering problem", *Math. Progr.* 48, 59-96 (1989).

[6] Katoh, N., T. Ibaraki, and Mine, "A polynomial time algorithm for the resource allocation problem with a convex objective function", *J. Opl. Res. Soc.* 30, 449-455 (1979).

[7] Nemhauser, G.L. and R. Carlson, "Scheduling to minimize interaction costs", *Operations Research* 14, 52-58(1966).

[8] Sahni, S. and T. Gonzales , "P-Complete approximation problems", *Journal ACM* 23, 555-565 (1976).