

N. M. M. DE ABREU

T. M. QUERIDO

P. O. BOAVENTURA-NETTO

**RedInv-SA : a simulated annealing for the
quadratic assignment problem**

RAIRO. Recherche opérationnelle, tome 33, n° 3 (1999),
p. 249-273

http://www.numdam.org/item?id=RO_1999__33_3_249_0

© AFCET, 1999, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
<http://www.numdam.org/>

REDINV-SA: A SIMULATED ANNEALING FOR THE QUADRATIC ASSIGNMENT PROBLEM (*)

by N.M.M. de ABREU ⁽¹⁾, T.M. QUERIDO ⁽²⁾
and P.O. BOAVENTURA-NETTO ⁽¹⁾

Communicated by Catherine ROUCAROL

Abstract. – An algebraic and combinatorial approach to the study of the Quadratic Assignment Problem produced theoretical results that can be applied to (meta) heuristics to give them information about the problem structure, allowing the construction of algorithms. In this paper those results were applied to inform a Simulated Annealing-type heuristic (which we called RedInv-SA). Some results from tests with known literature instances are presented.

Keywords: Quadratic assignment problem, meta-heuristics, simulated annealing, and combinatorial programming.

Résumé. – Une étude algébrique et combinatoire du Problème d'Affectation Quadratique a produit des résultats théoriques qui peuvent être appliqués à des (méta)-heuristiques. Donnant des informations sur la structure du problème, ils conduisent de cette façon à la formulation d'algorithmes. Dans cet article, ces résultats ont été appliqués à l'information d'une heuristique de type Recuit Simulé, appelée RedInv-SA. Des résultats de l'application de l'algorithme à des problèmes classiques de la littérature sont présentés à la fin de l'article.

Mots clés : Problèmes d'affectation quadratique, métaheuristiques, recuit simulé et programmation combinatoire.

1. INTRODUCTION

We consider the Quadratic Assignment Problem (QAP) in the classical form defined by Koopmans and Beckmann (KB57) as that of minimizing a sum of flow-distance products, which is the objective function of an allocation problem of n “machines” to n “positions” with pairwise cost dependency. This description has been applied to a variety of practical situations, from hospital plants through university campi to integrated circuits and keyboards.

(*) Received January 1996.

⁽¹⁾ COPPE/UFRJ, CP 68.507, 21.945-970 Rio de Janeiro, RJ, Brazil.

⁽²⁾ CEFET/RJ, Av. Maracanã n° 229, 20.271-110 Rio de Janeiro, RJ, Brazil.

The purposes of this paper are to present the essential features of an algebraic-combinatorial approach for QAP study and to apply some of these features in the building of a Simulated Annealing (SA) heuristic algorithm, as well as to allow for a solution quality evaluation. The theoretical basis, following Abreu (Ab84) involves a number of results concerning the permutation set. Among the results there presented it is important, for what follows, a condition for partial ordering of the permutations corresponding to the solutions of QAP instances and the connection between the permutation inversion number and the solution costs.

The K-B formulation is that of equations (1.1).

$$\min Z = \sum_{i,j,p,q=1}^n f_{ij}d_{pq}x_{ip}x_{jq}$$

subject to

$$\begin{aligned} \sum_{i=1}^n x_{ij} &= 1 & (j = 1, \dots, n) \\ \sum_{j=1}^n x_{ij} &= 1 & (i = 1, \dots, n) \\ x_{ij} &= 0 \text{ or } 1 \end{aligned} \tag{1.1}$$

where the f_{ij} and the d_{pq} are non-negative integers defining matrices F and D respectively. In this work we will consider these matrices as being symmetric, which allows us to substitute the Gavett-Plyter vector description (GP66) for them.

The QAP has some amazing properties, as it is extremely well structured and, on the other hand, it is a NP-hard problem for whose resolution the present tendency is the design of efficient heuristics. Such algorithms require instance banks for comparative testing and, in fact, a QAP instance library, QAPLIB [BKR94] has been constantly utilized and amplified. Techniques for generating instances with known optima have also been developed by Palubetskis [Pa88] and Li and Pardalos [LP92], allowing researchers to design their own test instances. An important reference on QAP is [PW94ed] where, among several important articles, a good review by the same authors [PW94] can be found.

2. SOME CONCEPTS AND DEFINITIONS

The graph theory basic concepts and definitions used in this work follow Berge [Be73].

The Koopmans-Beckmann description is equivalent to the Burkard and Stratman form [BS78], considering the assignment of facilities to locations as a permutation φ on $\{1, 2, \dots, n\}$ by setting $\varphi(i) = j$ if facility i is assigned to location j , such that

$$\min Z = \sum_{i,j=1}^n f_{ij}d_{\varphi(i)\varphi(j)}. \tag{2.1}$$

At this point it is convenient to recall some basic definitions.

DEFINITION 2.1: A *permutation* (or *t-permutation*) is a bijective function $\xi : T \rightarrow T$ (where T is the set of the first t naturals) represented as

$$\xi = \begin{pmatrix} 1 & 2 & \dots & t \\ \xi(1) & \xi(2) & \dots & \xi(t) \end{pmatrix}. \tag{2.2}$$

The natural t is known as the *order* of the permutation.

Let S_t be the set of all permutations over the elements in T . The *identity permutation* is the function $\iota_t \in S_t$ such that $\iota_t(j) = j$, for $j = 1, \dots, t$. Two elements $x, y \in T$ are *adjacent* in $\xi \in S_t$ when there exists $i \in T - \{t\}$ such that $x = \xi(i)$ and $y = \xi(i + 1)$ or $y = \xi(i)$ and $x = \xi(i + 1)$. We say that x *precedes* y in the image of ξ when there exist $i, j \in T$ such that $i < j$, $x = \xi(i)$ and $y = \xi(j)$. An ordered pair $(x, y) \in T \times T$ such that x precedes y in the image of ξ is an *inversion* in ξ when $x > y$.

It is frequently useful to shorthand a permutation by its image $\xi = (\xi(1)\xi(2)\dots\xi(t))$.

DEFINITION 2.2: Let ξ be a permutation. Then

$$\vartheta(\xi) = \{(x, y) \in T \times T | (x, y) \text{ is an inversion in } \xi\} \tag{2.3}$$

where $\vartheta(\xi)$ is the *inversion set* in ξ . The cardinality $|\vartheta(\xi)|$ is the number of inversions in ξ .

DEFINITION 2.3: A *cycle* (*k-cycle*) in S_t is a t -permutation μ with $(t - k)$ *fixed elements* $\mu(j) = j$, the remaining k elements being given

by $\mu(x_i) = x_{i+1}$ ($i = 1, \dots, k - 1$), $\mu(x_k) = x_1$. A cycle is usually represented as (x_1, \dots, x_k) . A 2-cycle is also called a *transposition*.

DEFINITION 2.4: A QAP instance of order n is the association of two n -cliques K_f and K_d , respectively valued by the material flows between machines and by the distances between positions (Fig. 2.1).

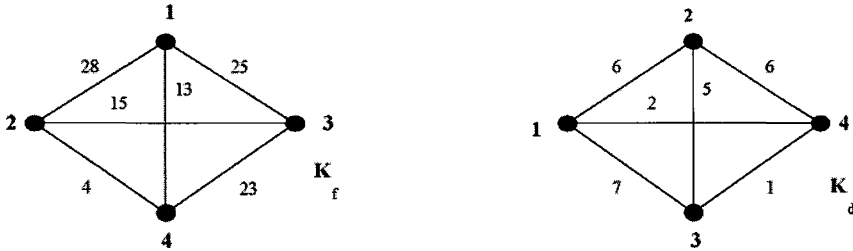


Fig. 2.1. - Cliques K_f and K_d .

This example is the Gavett-Plyter instance [GP66]. These authors developed a QAP relaxation (which is a Linear Assignment Problem of size $N = C_{n,2}$) defined by two column vectors, $F = [f_r]$ and $D = [d_s]$ ($r, s = 1, \dots, N$) where the components are respectively the edges of K_f and K_d taken in lexicographic order.

DEFINITION 2.5: We define a flow-distance edge-edge matrix Γ , as

$$\Gamma = FD^T = [\gamma_{rs}]. \tag{2.4}$$

Every QAP feasible solution is a solution for the relaxed problem defined by Γ . The reciprocal, however, is not true, because the relaxed problem has $N!$ solutions, of which only $n!$ are feasible.

The original and the relaxed problems can be associated to each other by the bijection between lexicographic and numerical orders; for a pair (i, j) ($1 \leq i < j \leq n$) the numerical ordering ψ_{ij} is

$$\psi_{ij} = (i - 1)n - i(i + 1)/2 + j. \tag{2.5}$$

DEFINITION 2.6: By sorting F components in non-decreasing order and D components in non-increasing order and calling \bar{F} and \bar{D} the vectors thus obtained, we define a standard matrix $\bar{\Gamma}$ as

$$\bar{\Gamma} = \bar{F} \bar{D}^T. \tag{2.6}$$

and ϕ_F and ϕ_D as the permutations taking respectively F into \bar{F} and D into \bar{D} .

Example matrices are shown in Figures 2.2 and 2.3, where lines and columns are identified by both orderings and by the edge values.

$\Psi(i, j)$	i, j	F	1	2	3	4	5	6	$\Psi(i, j)$ i, j D
			12	13	14	23	24	34	
			6	7	2	5	6	1	
1	12	28	<u>168</u>	196	56	140	168	28	
2	13	25	150	175	50	125	<u>150</u>	25	
3	14	13	78	91	26	<u>65</u>	78	13	
4	23	15	90	105	<u>30</u>	75	90	15	
5	24	4	24	<u>28</u>	8	20	24	4	
6	34	23	138	161	46	115	138	<u>23</u>	

Fig. 2.2. - Γ matrix - [GP66] instance.

ϕ_F^{-1}	$\Psi^{-1}\phi_F^{-1}$	\bar{F}	2	1	5	4	3	6	ϕ_D^{-1} $\Psi^{-1}\phi_D^{-1}$ \bar{D}
			13	12	24	23	14	34	
			7	6	6	5	2	1	
5	24	4	<u>28</u>	24	24	20	8	4	
3	14	13	91	78	78	<u>65</u>	26	13	
4	23	15	105	90	90	75	<u>30</u>	15	
6	34	23	161	138	138	115	46	<u>23</u>	
2	13	25	175	150	<u>150</u>	125	50	25	
1	12	28	196	<u>168</u>	168	140	56	28	

Fig. 2.3. - Standard $\bar{\Gamma}$ matrix for the [GP66] instance.

The feasible solution corresponding to [2 1 4 3] in the graphs of Figure 2.1, is shown in both matrices as underlined characters (with a cost equal to 464). As a consequence of $\bar{\Gamma}$ ordered structure we can obtain a lower bound $\text{tr}(\bar{\Gamma})$ for the optimal QAP solution; Hardy *et al.* [HLP52] and Wimmert [Wi58] show it is as being equal to the trace of $\bar{\Gamma}$, that is, the sum of its diagonal elements. (For this instance we have $\text{tr}(\bar{\Gamma}) = 389$, the optimal solution being [4 1 3 2] with cost 403.)

DEFINITION 2.7: To a given matrix $\bar{\Gamma}$ we associate a set $QA(\bar{\Gamma})$ of all instances I_{Γ} which can be obtained by different \bar{F} and \bar{D} orderings. Each of these instances will be defined by a given (F, D) pair.

3. THE FEASIBILITY PROBLEM

DEFINITION 3.1: Let us now consider a given matrix $\bar{\Gamma}$ and a given N -permutation $\rho_{\bar{\Gamma}}$. We say that the N -vector $X = (\bar{f}_1 \bar{d}_{\rho_{\bar{\Gamma}}(1)}, \dots, \bar{f}_N \bar{d}_{\rho_{\bar{\Gamma}}(N)})$ is *feasible* for a given instance $P \in QA(\bar{\Gamma})$ if we can find a vertex permutation φ such that

$$\bar{f}_r \bar{d}_{\rho_{\bar{\Gamma}}(r)} = f_{\phi_F^{-1}(r)} d_{\phi_D^{-1}(\rho_{\bar{\Gamma}}(r))} = f_{\Psi_{ij}} d_{\Psi_{\varphi(i)\varphi(j)}} \quad (r = 1, \dots, N). \quad (3.1)$$

The feasibility is guaranteed by the coherent application of K_d edges over K_f edges, that is, preserving vertex over vertex.

For a given $I_{\Gamma} \in QA(\bar{\Gamma})$ we consider the permutations ρ_{Γ} and $\rho_{\bar{\Gamma}}$ as acting respectively on Γ and $\bar{\Gamma}$ columns. It is easy to see that

$$\rho_{\Gamma} = \phi_F \circ \rho_{\bar{\Gamma}} \circ \phi_D^{-1}. \quad (3.2)$$

In order to clarify the composition operation shown in the example that follows, we will use here the notation presented in (2.2). This composition is done from left to right.

Example: We take again the [GP66] instance, with the solution defined in Γ as $\rho_{\Gamma} = [1 \ 5 \ 4 \ 3 \ 2 \ 6]$ and in $\bar{\Gamma}$ as $\rho_{\bar{\Gamma}} = [1 \ 4 \ 5 \ 6 \ 3 \ 2]$.

$$\rho_{\Gamma} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 5 & 4 & 3 & 2 & 6 \end{pmatrix}.$$

Then

$$\phi_F \circ \rho_{\bar{\Gamma}} \circ \phi_D^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 5 & 2 & 3 & 1 & 4 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 5 & 6 & 3 & 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 1 & 5 & 4 & 3 & 6 \end{pmatrix} = \rho_{\Gamma}.$$

To identify whether an edge-permutation ρ_{Γ} corresponds or not to a feasible solution for a given instance I_{Γ} , we have to verify if there is a vertex-permutation $\varphi \in S_n$ leading to an edge allocation in $\bar{\Gamma}$ corresponding to

$$\rho_{\Gamma} = \begin{pmatrix} \Psi(1, 2) = 1 & \Psi(1, 3) = 2 & \dots & \Psi(n - 1, n) = N \\ \Psi(\varphi(1), \varphi(2)) & \Psi(\varphi(1), \varphi(3)) & \dots & \Psi(\varphi(n - 1), \varphi(n)) \end{pmatrix}.$$

For the [GP66] instance we have, as an example (see Figs. 2.2 and 2.3),

Matrix	—>	$\bar{\Gamma}$	Γ	$\bar{\Gamma}$	Γ
Position	—>	(6,2)	(1,1)	(5,3)	(2,5)
Entry value	->	168		150.	

4. INVERSIONS AND COSTS

Through this item we will consider every N -permutation as a permutation of $\bar{\Gamma}$ columns; for sake of simplicity we will not use the $\bar{\Gamma}$ index.

The translation of (1.1) into edge permutations associated to permutations of $\bar{\Gamma}$ columns allows us to write, for a given N -permutation ξ ,

$$Z_\xi = \sum_{i=1}^N \bar{f}_i \bar{d}_{\xi(i)} \tag{4.1}$$

DEFINITION 4.1: The *increase* associated to a permutation $\xi \in S_N$ is

$$\Delta_\xi = Z_\xi - Z_I \tag{4.2}$$

where $Z_I = \text{tr}(\bar{\Gamma})$ is the problem lower bound associated to the identity permutation ι_N .

THEOREM 4.1: *The increase associated to a cycle $\mu = (x_1, \dots, x_k) \in S_N$ is given by*

$$\Delta(\mu) = \sum_{j=1}^{k-1} \bar{f}(x_j) (\bar{d}(x_{j+1}) - \bar{d}(x_j)) + \bar{f}(x_k) (\bar{d}(x_1) - \bar{d}(x_k)).$$

Proof: By considering (4.1) and (4.2). □

We know that every permutation can be expressed as a composition of disjoint cycles (see for instance Berge [Be68]). The next theorem allows the increase of a permutation to be defined as a function of the increases associated to its cycles.

THEOREM 4.2: *Let $\xi \in S_N$ be a permutation expressed as a composition of r disjoint cycles $\xi = \mu_1 \circ \mu_2 \circ \dots \circ \mu_r$. The increase associated to ξ is*

$$\Delta\xi = \sum_{j=1}^r \Delta(\mu_j).$$

Proof: As the cycles are disjoint, we can write

$$Z_\xi = \sum_{j=1}^r \sum_{y \in \mu_j} \bar{f}(y) \bar{d}(\mu_j(y)) \quad (4.3)$$

and therefore

$$\Delta_\xi = Z_\xi - Z_I = \sum_{j=1}^r \left[\sum_{y \in \mu_j} \bar{f}(y) [\bar{d}(\mu_j(y)) - \bar{d}(y)] \right] = \sum_{j=1}^r \Delta(\mu_j). \quad (4.4)$$

□

This rather simple result has a strong consequence; as every parcel in the sum is non-negative, we can say that, for every j , $j = 1, \dots, r$,

$$\Delta(\mu_j) \leq \Delta(\xi). \quad (4.5)$$

Then we can have permutation pairs (ξ, π) such that $\Delta\xi \leq \Delta\pi$ or $\Delta\xi \geq \Delta\pi$, **regardless** of the instance values.

DEFINITION 4.2: Two increases $\Delta\xi_1$ and $\Delta\xi_2$ are *freely comparable* if it is possible to verify either $\Delta\xi_1 \leq \Delta\xi_2$ or $\Delta\xi_1 \geq \Delta\xi_2$, regardless of $\bar{\Gamma}$ entry values.

THEOREM 4.3: *If a permutation ξ_2 is obtained from a permutation ξ_1 by applying a transposition θ , then ξ_1 and ξ_2 are freely comparable.*

Proof: Let us consider $\theta = (x, y)$, $x < y$ and $\xi_2 = \xi_1 \circ \theta$. We will have $\xi_2(x) = \xi_1(y)$, $\xi_2(y) = \xi_1(x)$ and $\xi_2(i) = \xi_1(i)$ for $i \neq x, y$. Then

$$\Delta(\xi_2) - \Delta(\xi_1) = (\bar{f}(x) - \bar{f}(y))[\bar{d}(\xi_1(y)) - \bar{d}(\xi_1(x))]. \quad (4.6)$$

As \bar{F} has a non-descending ordering the first factor of the right-hand member will always be non-positive. For the same ordering reason, the second factor will always be non-negative if $\xi_1(y) < \xi_1(x)$ and non-positive if $\xi_1(y) > \xi_1(x)$. □

According to this proof the comparison direction (either $\Delta(\xi_2) \leq \Delta(\xi_1)$ or $\Delta(\xi_2) \geq \Delta(\xi_1)$) depends on whether $\xi_1(x)$ is lesser or greater than $\xi_1(y)$. This comparison corresponds to an order of scalar products, an approach used by [Re85]. The next theorem relates this result to the inversion number of ξ_1 and ξ_2 .

THEOREM 4.4: *Let $\xi_1, \xi_2, \theta \in S_N$, where θ is a transposition. If $\xi_2 = \xi_1 \circ \theta$ and $|\vartheta(\xi_1)| < |\vartheta(\xi_2)|$, then ξ_1 and ξ_2 are freely comparable and $Z_{\xi_1} \leq Z_{\xi_2}$.*

Proof: Consider $\theta = (x, y)$ where $1 \leq x < y \leq N$. Then the permutations ξ_1 and ξ_2 are respectively of the form

$$\xi_1 = \left(\begin{array}{cccc} \dots & x & \dots & y & \dots \\ \dots & \xi_1(x) & \dots & \xi_1(y) & \dots \end{array} \right) \quad \text{and} \quad \xi_2 = \left(\begin{array}{cccc} \dots & x & \dots & y & \dots \\ \dots & \xi_1(y) & \dots & \xi_1(x) & \dots \end{array} \right).$$

Taking into account that $|\vartheta(\xi_1)| < |\vartheta(\xi_2)|$ implies $\xi_1(x) < \xi_1(y)$ we have, from (4.6),

$$\Delta_{\xi_1} - \Delta_{\xi_2} = (\bar{f}(x) - \bar{f}(y))[\bar{d}(\xi_1(y)) - \bar{d}(\xi_1(x))].$$

The orderings defined for \bar{F} and \bar{D} imply respectively $\bar{f}(x) \leq \bar{f}(y)$ and $\bar{d}(\xi_1(y)) - \bar{d}(\xi_1(x))$, from which we obtain $\Delta_{\xi_1} \leq \Delta_{\xi_2}$ and, consequently,

$$Z_{\xi_1} \leq Z_{\xi_2}. \quad \square$$

DEFINITION 4.5: Theorem 4.4 allows us to build a directed graph $GI_N = (S_N, W)$ where W is the set of pairs (ξ_i, ξ_j) such that $\vartheta(\xi_i)$ has exactly one more inversion than $\vartheta(\xi_j)$. This graph is known as the *inversion graph*.

GI_N is a circuitless graph whose level ordinal function is the number of inversions. Level 0 has exactly one element (which is ϑ_N , the identity permutation corresponding to the lower bound $\text{tr}(\bar{\Gamma})$). It has a lattice structure [Be68, AV95].

Figure 4.1 is a scheme of GI_N vertex distribution for $N = 6$; the graph is regular of degree $N - 1$, so the single vertex in Level 0 is adjacent to every vertex in Level 1; from now on, the adjacency relations depend on which positions of each N -permutation are affected by the possible inversions. For $N = 6$ we have $C_{6,2} + 1 = 16$ levels and their cardinalities N_i can

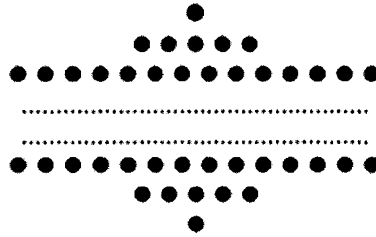


Fig. 4.1. - Scheme of GI_6 .

be easily found, through a generating function, to be 1, 5, 14, 29, ..., 29, 14, 5, 1 [Ab84].

Figure 4.2 shows the first three levels of GI_6 with the corresponding permutations. We consider the edges to be directed from Level $k + 1$ to Level k , according to Definition 4.5.

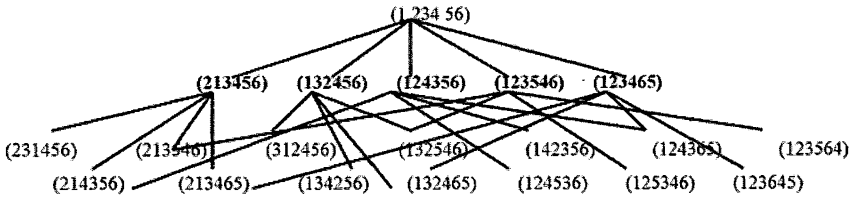


Fig. 4.2. - GI_6 first three levels.

THEOREM 4.5: *If there exists a directed path between ξ_i and ξ_j in GI_N , then $\Delta(\xi_i)$ and $\Delta(\xi_j)$ are freely comparable, with $|\vartheta(\xi_i)| < |\vartheta(\xi_j)|$ and consequently $Z_{\xi_i} \leq Z_{\xi_j}$.*

Proof: It follows from Definition 4.5 and Theorem 4.4. □

Theorem 4.5 allows us to identify a class of freely comparable permutation pairs where the respective number of inversions grow with the increases and consequently with the costs.

The freely comparable increases are studied in more detail in [Ab84, AbBo89]. One of the most important results there presented is a Maximal Partial Ordering Theorem (MPOT), which gives necessary and sufficient conditions for free comparability between permutations.

A result equivalent to Theorem 4.5 is not available for all MPOT cases. To discuss the application of this theory with regard to heuristic applications we present a conjecture (the Inversion Conjecture) [Ab84, AB89] which

is reinforced, from a practical point of view, by Figures 6.1a and b and 6.2a and b.

CONJECTURE 4.1: The cost function within freely comparable solutions increases with their number of inversions in the corresponding permutations on $\bar{\Gamma}$ columns. \square

5. THE REDINV-SA HEURISTIC

5.1. A brief discussion of Simulated Annealing meta-heuristic

Some exact algorithms have been formulated to work with the QAP [Ro87, CB89, MR94a, RRP96, PRRL97] but its NP-hardness stimulates the use of heuristic techniques and meta-heuristic-based algorithms [BR84, WW87, SK89, Con89, Con90, Fl90, Ta91, MM92, MLP92, Wh93, BM93, LPR94, FF94, CSK94, SK94, MM95].

The Simulated Annealing (SA) meta-heuristics has been used with the QAP [BR84, WW87, Con89, and Con90]. These references will also be considered for a general SA description, allowing us, after a short recall, to deal only with our approach specificities.

Let us consider a minimization problem, where we examine at each iteration the increase δ on the objective function. If δ is negative the new solution will always be accepted, while for worse solutions ($\delta > 0$) we evaluate the expression

$$p = e^{-\delta/T(t)} \quad (5.1)$$

where $T(t)$ is a monotonically non-increasing *temperature function* and p is the probability of acceptance of a new (worse) solution. We then have a strategy to avoid the algorithm path to be stuck to local optima. The probability p will be greater at the beginning as we initialize T with a comparatively high value, a reduction (“cooling”) strategy being applied during the algorithm execution, for which a given number of iterations is specified to use each new T value. The initial T value selection and the possible strategies for its reduction has been extensively discussed in the literature. The general SA form is as follows:

begin

< Data: initial solution x ; function $T(t)$; >

$t \leftarrow 0$;

while $T(t) \notin \{\text{stop criterion}\}$ **do**

begin

$i \leftarrow 0$;

while $i < M$ **do** < M : maximum iteration number for each T value >

begin

generate a new solution x' ;

$\Delta f \leftarrow f(x') - f(x)$;

if $\Delta f < 0$ **then** $x \leftarrow x'$;

else if $\exp(-\Delta f/T(t)) < \text{random}(0,1)$ **then** $x \leftarrow x'$;

$i \leftarrow i + 1$;

end;

$t \leftarrow t + 1$;

end;

end.

The stopping criterion has to do with the algorithm's efficiency in obtaining new better solutions. Eglese [Eg90] does a discussion on this point.

The criteria and the parameters used in the SA version here presented are supported by the theory discussed in Items 2 through 4.

5.2. The formulation of the RedInv-SA heuristic

We want to look for the best possible solution for the function

$$Z_{\rho_{\bar{\Gamma}}} = \sum_{i=1}^n \bar{f}_i \bar{d}_{\rho_{\bar{\Gamma}}(i)} \quad (5.2)$$

where $Z(\rho_{\bar{\Gamma}})$ is the cost associated to the permutation $\rho_{\bar{\Gamma}}$ on $\bar{\Gamma}$ columns. The corresponding Γ column permutation, which defines the instance edge changing, can be obtained by the composition shown in (3.2).

We use Conjecture 4.1 to support the definition of a neighboring criterion for the algorithm to work. The strategy here presented is then a representative of a family of possible heuristic techniques using the decreasing number of inversions as a criterion for choosing elements to be exchanged [Que94].

The permutations on $\bar{\Gamma}$ columns are related to those defined on Γ ones by the expression already presented,

$$\rho_{\bar{\Gamma}} = \phi_F^{-1} \circ \rho_{\Gamma} \circ \phi_D. \quad (5.3)$$

We look for a position m in ρ_{Γ}^{-1} which maximizes $|i - \rho_{\Gamma}^{-1}(i)|$. The exchange will be made with a position j which minimizes $\{|m - \rho_{\Gamma}^{-1}(i)| + |i - \rho_{\Gamma}^{-1}(m)|\}$, so the decrease in the number of inversions will be maximum. The permutation ρ'_{Γ} thus obtained does not correspond to a feasible solution, as only two edges have been exchanged and a vertex exchange would deal with $2(n - 2)$ edges. We will then look for vertex exchanges whose realization implies exchanging edges i and m , thus defining a neighborhood of feasible permutations. At this point, better quality solutions cannot be guaranteed, as the neighborhood could be that of a local optimum. The SA, nevertheless, was designed exactly to deal with such situations.

To find the neighborhood we use (3.2) to return to a Γ column permutation ρ'_{Γ} and then look for the two positions in ρ'_{Γ} which had their images exchanged. From an algorithmic point of view it is not necessary to execute the whole of (3.2) to obtain this result: we simply obtain these two positions, which will be

$$p = \phi_F^{-1}(m) \quad \text{and} \quad q = \phi_F^{-1}(j), \tag{5.4}$$

as well as the corresponding images $t = \rho'_{\Gamma}(p)$ and $u = \rho'_{\Gamma}(q)$.

We finish by applying the inverse function ψ^{-1} of (2.5),

$$\psi^{-1}(t) = (x_t, y_t) \quad \text{and} \quad \psi^{-1}(u) = (x_u, y_u) \tag{5.5}$$

where we can have either four different vertices, if the edges m and j are not adjacent, or, if they are adjacent, only three vertices. In the first case the vertex exchanges will be $x_t \leftrightarrow x_u$, $x_t \leftrightarrow y_u$, $y_t \leftrightarrow x_u$ and $y_t \leftrightarrow y_u$. The second case will give only three exchanges, the fourth one being void.

5.3. An exchange strategy example

The Nugent 5 instance [NVR68] is defined by

$$F = [5, 2, 4, 1, 3, 0, 2, 0, 0, 5] \quad \text{and} \quad D = [1, 1, 2, 3, 2, 1, 2, 1, 2, 1]$$

respectively for K_f and K_d edges in lexicographic order. Then the corresponding bijections defining the orderings for \overline{F} and \overline{D} will be

$$\phi_F = [95847162310] \quad \text{and} \quad \phi_D = [67213849510].$$

Taking as an example a vertex permutation

$$\varphi = [41352]$$

we have, as $\rho_\Gamma = [\psi(\varphi_i, \varphi_j)]$, $1 \leq i < j \leq n$,

$$\rho_\Gamma = [3\ 8\ 10\ 6\ 2\ 4\ 1\ 9\ 5\ 7]$$

with cost 32. By (5.3) the corresponding permutation in $\bar{\Gamma}$ is given by

$$\rho_{\bar{\Gamma}} = [1\ 5\ 3\ 8\ 9\ 6\ 7\ 10\ 2\ 4]$$

with 18 inversions.

We can easily find that $m = 9$, as $\max |i - \rho_{\bar{\Gamma}}(i)| = 7$ and $j = 4$, then we have $\min\{|9 - \rho_{\bar{\Gamma}}(i)| + |i - \rho_{\bar{\Gamma}}(9)|\} = 3$. The modified permutation $\rho'_{\bar{\Gamma}}$ is

$$\rho'_{\bar{\Gamma}} = [1\ 5\ 3\ 2\ 9\ 6\ 7\ 10\ 8\ 4].$$

When returning to Γ we have, at first,

$$\phi_F^{-1}(9) = 1; \rho'_\Gamma(1) = 3 \quad \text{and} \quad \phi_F^{-1}(4) = 4; \rho'_\Gamma(4) = 6.$$

From where we obtain

$$\psi^{-1}(3) = (1, 4) \quad \text{and} \quad \psi^{-1}(6) = (2, 4).$$

The exchanges will then be done between the vertices defining the edges (1,4) and (2,4), in every possible arrangement. In this case the resulting neighborhood is then (1,2), (1,4) and (2,4). The vertex exchanges being defined on the image, we obtain respectively [4 2 3 5 1], [1 4 3 5 2] and [2 1 3 5 4]. This last one has cost 30 and 17 inversions in its $\bar{\Gamma}$ column permutation, which is [1 5 9 8 3 2 7 4 6 10]. We have $\Delta z = -2$, the objective function will decrease and the new solution will be accepted. If Δz were positive, we would use the SA proper criterion, according to (5.1).

5.4. The RedInv-SA algorithm

Given n and Γ , we determine ψ , ϕ_F , ϕ_D and $\bar{\Gamma}$ as previously defined.

We make

$$T_0 = \sum_{i=1}^N \bar{f}_i \bar{d}_{N-i+1} \quad (5.6)$$

and we begin with a randomly chosen vertex permutation from K_d vertices on K_f ones.

The initial temperature value will be taken as the problem upper bound (the element sum from the secondary diagonal of $\bar{\Gamma}$) reduced by an appropriate factor K . We used the function $T(t+1) = 0.9T(t)$, as proposed by Connolly [Con89]. The temperature function loop size M ranged between one and nine (in one case forty) times the problem order n . The stopping criterion was $t < \text{tr}(\bar{\Gamma})/K$, but limited to ten (in two cases five) temperature values. As the edge exchanges are deterministic, a temporary hindrance was put on the last n accepted solutions in order to avoid cycling.

The algorithm is as follows:

```

begin
  T ← T0;
  while (T > tr( $\bar{\Gamma}$ ) or nycyc = maxnycyc) do
    begin
      i ← 0;
      while i ≤ n do
        begin
          procedure change; < return ( $\bar{Z}, \varphi'$ ) >
            ΔZ ←  $\bar{Z} - Z$ ;
            if ΔZ < 0 then  $\varphi \leftarrow \varphi'$ ;
              else if exp(-ΔZ/T) < random (0,1) then  $\varphi \leftarrow \varphi'$ ;
            i ← i + 1;
          end;
        T ← 0.9T;
      end;
    end. <  $\varphi'$ : new solution; Z $_{\varphi'}$ : cost >

```

The operation sequence within the procedure **change** is as follows, according with what was discussed before:

The procedure receives a solution φ' with cost $Z_{\varphi'}$.

We determine

$$\rho_{\Gamma} = \begin{pmatrix} 12 & 13 & \dots & N-1N \\ \varphi(1)\varphi(2) & \varphi(1)\varphi(3) & \dots & \varphi(N-1)\varphi(N) \end{pmatrix}$$

and

$$\rho_{\bar{\Gamma}} = \phi_F^{-1} \circ \rho_{\Gamma} \circ \varphi_D.$$

We then find the positions for changing in $\rho_{\bar{\Gamma}}$:

$$\begin{aligned} m &\leftarrow \text{position corresponding to } \max |i - \rho(i)|; \\ j &\leftarrow \text{position corresponding to } \min \{|m - \rho(i)| + |i - \rho(m)|\}. \end{aligned}$$

We return to a Γ column based permutation: $p = \phi_{\bar{F}}^{-1}(m)$ and $q = \phi_{\bar{F}}^{-1}(j)$.

We find the images $t = \rho'_{\Gamma}(p)$ and $u = \rho'_{\Gamma}(q)$.

We find $\psi^{-1}(t) = (a, b)$ and $\psi^{-1}(u) = (r, s)$.

We generate the neighbor set V_{φ} :

$$\begin{aligned} V_{\varphi} &= \{\varphi' | \varphi'(i) = \varphi(i); i \neq a, b; \varphi'(a) = r \text{ or } s; \varphi'(b) = s \text{ or } r\} \\ \bar{Z} &\leftarrow \min_{\varphi' \in V_{\varphi}} Z_{\varphi'}; \\ \varphi &\leftarrow \varphi'; \end{aligned}$$

Return: new solution φ' with value $Z_{\varphi'}$.

6. DISCUSSION OF RESULTS

The algorithm was programmed in Fortran-77. The computational tests were conducted in two equipments:

- a Pentium-166 microcomputer;
- a DEC workstation with a 170 Mhz Alpha processor.

The presented processes times are in every case those corresponding to P-166.

A sample of 35 symmetric instances from QAPLIB was processed. From these, 21 were lower-order ($12 \leq n \leq 18$) and 14 were higher-order ones ($n \geq 20$). In every case but the two larger instances, a number between 9 and 18 fifty-run tests were executed, combining better initial temperature values with different number of solutions per temperature value. The two larger instances [Tai60a and Sko81] used 25-run tests.

The cooling schedule was based on a constant reduction factor equal to 0.9 for all tests. The initial value was equal to the problem upper limit, reduced by a factor K :

$$T_0 = \bar{F}^T \bar{D}^* / K \quad (6.1)$$

where \overline{D}^* is the vector \overline{D} taken in inverse order. The schedule end was indicated by whichever the following constraints apply first:

- the last temperature going below the problem reduced lower bound, $\text{tr}(\overline{T})/K$;
- ten temperatures having already been obtained (five for Tai60a and Sko81).

The initial temperature range was estimated, for each instance, by running single runs and observing the behavior of the accepted/refused (A/R) solution relation for the set of temperature values, an initial temperature being considered appropriate when giving a fairly initial high A/R (say 4 to 8) and being able to decay to a fairly low A/R (say 0.2 to 0.4), thus showing good sensitivity to problem structure. The values for K were chosen looking for circumscribing this sensible zone.

The number **nsol** of solutions per temperature value is an indicator for the instance difficulty, as some instances required higher values of it to converge while for others the algorithm showed good results with much lesser values.

For each test, the minimum obtained value (Z_{\min}), the average solution value (\overline{Z}) and the average P-166 processing time per run were obtained. The optimal (or better-known) value Z_{opt} for the instance was utilized to calculate a percentile increment of the average on it,

$$p_{\text{av}} = 100 \overline{Z} / Z_{\text{opt}} \quad (6.2)$$

and also a percentual increment of the test minimum obtained value on it,

$$p_{\min} = 100 Z_{\min} / Z_{\text{opt}}.$$

In Tables 6.1 and 6.2 below we specify, for each instance:

- the number **ntest** of tests;
- the range of **nsol** values;
- p_{avs} , the average value of p_{av} for the test set;
- the next column has a double meaning:
 - where Z_{opt} was not obtained, this column shows p_{ms} , the minimum p_{\min} value for the whole set of tests.
 - where Z_{opt} was obtained (so $p_{\text{ms}} = 0$) the cell contains the number n_{opt} of this event for the whole test set.
- the average run time (in seconds) for the test set.

TABLE 6.1
RedInv-SA results with lower-order instances ($12 \leq n \leq 18$).

Instance	n _{test}	nsol	p_{avs}	$p_{ms} - n_{opt}$	Av. time
Rou12	9	30-50	2.04	(23)	0.55
Had12	9	12-36	2.23	(1)	0.15
Nug12	10	10-80	5.01	(39)	0.38
Tai12a	12	12-48	7.36	(17)	0.36
Scr12	12	12-36	9.82	(39)	0.51
Chr12b	9	36-108	34.42	(16)	1.69
Chr12a	16	48-480	40.91	5.73	4.10
Had14	9	20-70	1.64	(4)	0.65
Tai15a	16	15-120	4.78	0.20	1.40
Nug15	10	10-45	4.81	(12)	0.83
Rou15	9	30-90	5.38	(4)	1.77
Scr15	16	30-240	15.90	(2)	1.44
Chr15a	12	135-225	58.61	6.20	2.60
Chr15b	16	45-225	83.40	9.01	1.98
Esc16b	9	16-48	1.89	(89)	0.95
Nug16a	14	32-128	4.31	(2)	3.29
Esc16c	12	16-48	11.19	(29)	2.24
Had16	16	32-112	11.57	0.05	1.55
Esc16d	9	8-32	27.12	(69)	1.38
Had18	16	36-126	1.66	0.15	2.72
Nug18	16	36-144	4.94	(1)	5.86

TABLE 6.2
RedInv-SA results with higher-order instances ($20 \leq n \leq 100$).

Instance	n _{test}	nsol	p_{avs}	$p_{ms} - n_{opt}$	Av. time
Had20	16	40-160	1.64	0.15	5.36
Nug20	9	20-300	3.39	0.15	16.55
Rou20	16	80-300	3.84	0.38	13.72
Tai20a	15	40-160	5.59	1.87	7.68
Scr20	15	60-240	21.29	0.02	17.56
Tai25a	18	50-250	5.98	1.71	25.55
Chr25a	12	100-400	110.00	50.31	61.67
Nug30	12	60-300	8.53	2.68	88.95
Esc32a	10	32-96	51.33	25.69	60.86
Esc32b	12	64-256	49.75	27.97	101.56
Ste36a	12	36-144	52.82	17.23	89.44
Sko42	18	84-672	7.84	3.74	598.16
Tai60a	9	120-360	7.58	5.43	1456.33
Sko81	6	81-324	8.57	6.47	2856.00

The table was ordered by p_{avs} values in crescent order for each problem size. These values can be seen as a proximity measure (between Z_{opt} and the algorithm path) which can be used as a problem difficulty criterion. We can see, for example, that the algorithm is not efficient for some tree-structured Christofides instances: it looks like it cannot easily find a path along that structure. Nevertheless in one case (Chr12b) the p_{avs} criterion does not seem to apply, the average differences staying high whereas a number of optimal values were found. As with any other technique, the problem order has a strong influence: the Nugent series show it clearly. There appears a curious question concerning the instances Esc16 and 32: in the former case we have an important number of different optimal solutions, which allowed the algorithm to present very good results. This is possibly not true for the latter case, where the results were worse than, for example, those shown by Ste36a. We did not consider this question of instance difficulty to be within the scope of this paper; some discussion of it can be found in [MR94b].

Some examples of the algorithm convergence are given in Table 6.3 below. Here we can observe for some sample instances how p_{avn} (which we define as the average of the p_{av} values corresponding to a given **nsol** value) goes to lesser values as **nsol** grows.

TABLE 6.3
Example for RedInv-SA convergence.

Had 18		Nug 18		Tai 25a		Sko 42		Sko 81	
nsol	p_{avn}	nsol	p_{avn}	nsol	p_{avn}	nsol	p_{avn}	nsol	p_{avn}
36	1,90	36	5,89	50	6,75	84	8,56	81	9,39
54	1,78	72	5,01	100	5,98	168	8,09	162	8,37
90	1,53	108	4,48	150	5,76	336	7,38	324	7,94
126	1,42	144	4,39	200	5,41	672	6,96	—	—

Another interesting point to observe in the tests is the comparison between costs and inversion numbers along the algorithm path. The graphs from Figure 6.1a show the costs and the inversion numbers for the solutions examined by the algorithm during a run with the Nugent 12 instance. The graph segments within the rectangles are zoomed in Figure 6.1b, where the solutions are individually shown. The same is shown for Nugent 15 in Figures 6.2a and 6.2b. This type of graph can be found for any instance.

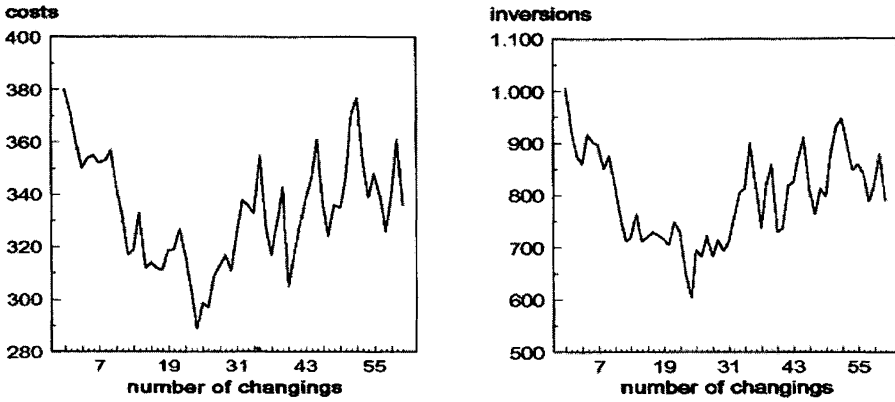


Fig. 6.1a

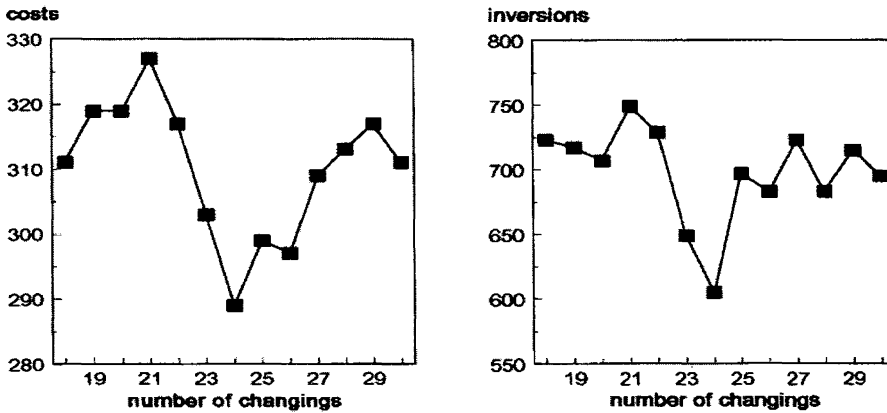


Fig. 6.1b

A detailed analysis would require the determination of free comparability occurrence along the solution trail, but even without that we can see the close relationship between these two solution properties.

6. FINAL REMARKS AND CONCLUSIONS

Although we give information about time processing we do not discuss it, the variety of hardware resources quoted in the literature as supports for computer tests making any time comparison very difficult.

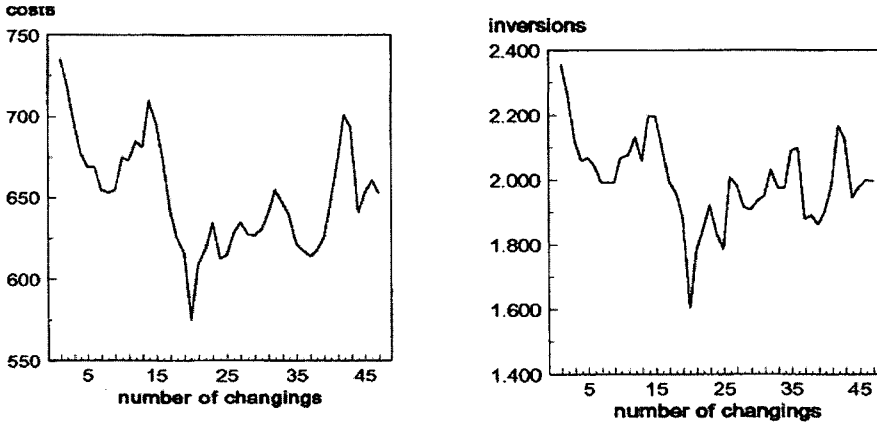


Fig. 6.1c

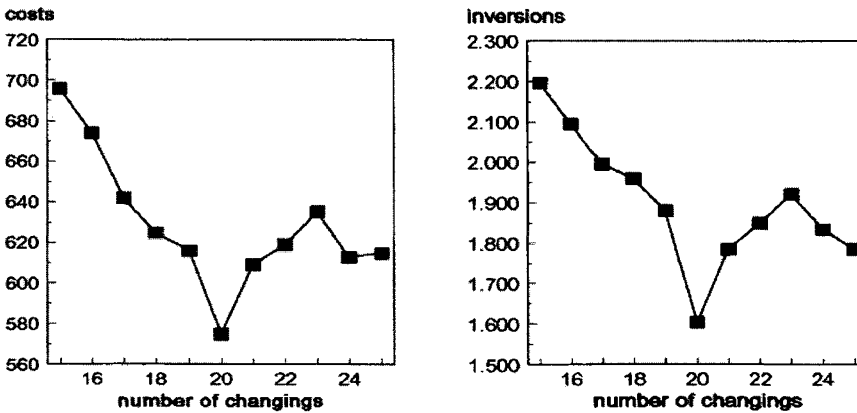


Fig. 6.1d

On the other hand, the number of solution changing made by previously known SA schemes is known to be between

- 1,000 and 4,000 for n between 12 and 15;
- 4,000 and 16,000 for n near 20;
- 5,000 and 30,000 for $n = 30$.

As a comparison, the maximum **nsol** values used in the study, multiplied by 10 (maximum number of different temperatures) gave, for the attained convergence levels, values between

- 360 and 2,250 for n between 12 and 15 (exception: Chr12a with 4,800);

- 320 and 3,840 for n between 16 and 18;
- 960 and 6,720 for n equal or greater than 20.

These values will evidently be pushed up when going further into convergence but we can see there is a fairly good slack, when relating them with previous results, for the work involved with the majority of the tested instances.

For problem order 18 or less the algorithm found the optimal (or best-known) solution one or more times in the majority of the tested instances. With order 20 it went near 0.4% or less in every but one instance. As it could be expected the difficulty arise with the size; nevertheless the convergence can be observed even in higher-order instances.

The second important point concerns Conjecture 4.1. By looking at Figures 6.1a to 6.2b we can see that this conjecture is reinforced by the results, similar graphs being easy to build for any instance. As the inversion number determination is an elementary procedure we think we have in it an efficient parameter for result analysis when using metaheuristics to solve QAP instances.

ACKNOWLEDGEMENTS

The authors are grateful to the referees by their very useful improvement suggestions, which strongly contributed to give more clarity to the text. They are also indebted to the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), whose financial support was essential to the realization of this work. The co-author Tania Maia Querido is also grateful for its support to her work on QAP at Florida University.

REFERENCES

- [Ab84] N. M. M. ABREU, *An Algebraic and Combinatorial Study of the Quadratic Assignment Problem as Defined by Koopmans and Beckmann* (in Portuguese), D.Sc. Thesis, COPPE/UFRJ, Rio de Janeiro, 1984.
- [AB89] N. M. M. ABREU and P. O. BOAVENTURA-NETTO, The Quadratic Assignment Problem: Permutation Ordering and Inversions, *AMSE Rev.*, 1989, 10, p. 21-52.

- [AV95] N. M. M. ABREU and O. VERNET, *Permutation Lattices and the Quadratic Assignment Problem*, Tech. Rep. PEP/PO/8, COPPE/UFRJ, Rio de Janeiro, 1995.
- [Be68] C. BERGE, *Principes de Combinatoire*, Dunod, Paris, 1968.
- [Be73] C. BERGE, *Graphes et Hypergraphes*, 2^e éd., Dunod, Paris, 1973.
- [BR84] R. E. BURKARD and F. RENDL, A Thermodynamically Motivated Simulation Procedure for Combinatorial Optimization Problems, *EJOR*, 1984, 17, p. 169-174.
- [BKR94] R. E. BURKARD, S. E. KARISCH and F. RENDL, *QAPLIB - A Quadratic Assignment Problem Library*, Rep. 287 CDLDO-41, Technische Universität, Graz, 1994.
- [BM93] T. N. BUI and B. R. MOON, *A Genetic Algorithm for a Special Class of the Quadratic Assignment Problem*, P. M. Pardalos, H. Wolkowicz, Ed., *Quadratic Assignment and Related Problems*, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., 16, American Mathematical Society, Providence, 1994.
- [LPR94] Y. LI, P. PARDALOS and M. G. C. RESENDE, *A Greedy Randomized Adaptive Search Procedure for the Quadratic Assignment Problem*, P. M. Pardalos, H. Wolkowicz, Ed., *Quadratic Assignment and Related Problems*, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., 16, American Mathematical Society, Providence, 1994.
- [BS78] R. E. BURKARD and R. H. STRATMAN, Numerical Investigations on Quadratic Assignment Problem, *NRLQ*, 1978, 25, p. 129-140.
- [CB89] N. CHRISTOFIDES and E. BENAVENT, An Exact Algorithm for the Quadratic Assignment Problem on a Tree, *Opns. Res.*, 1989, 37, p. 760-768.
- [Con89] D. CONNOLLY, *Solution Techniques for the Quadratic Assignment Problem and the Development of a General Purpose Simulated Annealing Algorithm*, Ph.D. Thesis, The London School of Economics and Political Science, London, 1989.
- [Con90] D. CONNOLLY, An Improved Annealing Scheme for the QAP, *EJOR*, 1990, 46, p. 93-100.
- [CSK94] J. CHAKRAPANI and J. SKORIN-KAPOV, A Constructive Method for Improving Lower Bounds for a Class of Quadratic Assignment Problems, *Opns. Res.*, 1994, 42, p. 837-845.
- [Eg90] R. W. EGGLESE, Simulated Annealing: a Tool for Operations Research, *EJOR*, 1990, 46, p. 271-281.
- [Fl90] M. M. FLOOD, Exact and Heuristic Algorithms for the Weighted Feedback Arc Set Problem: A Special Case of the Skew-Symmetric Quadratic Assignment Problem, *Networks*, 1990, 20, p. 1-23.
- [FF94] C. FLEURENT and J. A. FERLAND, *Genetic Hybrids for the Quadratic Assignment Problem*, P. M. Pardalos, H. Wolkowicz, Ed., *Quadratic Assignment and Related Problems*, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., 16, American Mathematical Society, Providence, 1994.
- [GP66] J. W. GAVETT and N. V. PLYTER, The Optimal Assignment of Facilities to Locations by Branch-and-Bound, *Opns. Res.*, 1966, 14, p. 210-232.
- [HLP52] G. H. HARDY, J. E. LITTLEWOOD and G. PÓLYA, *Inequalities*, Cambridge University Press, Cambridge, 1934.
- [KB57] T. C. KOOPMANS and M. BECKMANN, Assignment Problems and the Location of Economic Activities, *Econometrica*, 1957, 25, p. 53-76.

- [LP92] Y. LI and P. M. PARDALOS, Generating Quadratic Assignment Test Problems with Known Optimal Permutations, *Comput. Optim. Appl.*, 1992, 1, p. 163-184.
- [LPR94] Y. LI, P. M. PARDALOS and M. G. C. RESENDE, A Greedy Randomized Adaptive Search Procedure for the Quadratic Assignment Problem, P. M. Pardalos, H. Wolkowicz, Ed., Quadratic Assignment and Related Problems, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., 16, American Mathematical Society, Providence, 1994.
- [MLP92] K. A. MURTHY, Y. LI and P. M. PARDALOS, A Local Search Algorithm for the Quadratic Assignment Problem, *Informatica*, 1992, 4, p. 524-538, 596, 604.
- [MM92] J. MILLIS and V. MAGIROU, A Lagrangean Relaxation Algorithm for the Task and the Quadratic Assignment Problems, Helln Res. In Maths. and Informat. '92, Helln. Math. Soc., Athens, 1992, p. 267-279.
- [MM95] J. MILLIS and V. MAGIROU, A Lagrangean Relaxation Algorithm for Sparse Quadratic Assignment Problems, *Oper. Res. Lett.*, 1995, 17, p. 69-76.
- [MR94a] T. MAUTOR and C. ROUCAIROL, A New Exact Algorithm for the Solution of Quadratic Assignment Problems, *Discr. Appl. Maths.*, 1994, 55, p. 281-293.
- [MR94b] T. MAUTOR and C. ROUCAIROL, *Difficulties of Exact Methods for Solving the Quadratic Assignment Problem*, P. M. Pardalos, H. Wolkowicz, Ed., Quadratic Assignment and Related Problems, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., 16, American Mathematical Society, Providence, 1994.
- [NVR68] C. E. NUGENT, T. E. VOLLMANN and J. RUMML, An Experimental Comparison of Techniques for the Assignment of Facilities to Locations, *Opns. Res. Qtrly.*, 1968, 16, p. 150-173.
- [Pa88] G. S. PALUBETSKIS, A Generator of Quadratic Assignment Test Problems with a Known Optimal Solution, *Zh. Vychisl. Mat. I Mat. Fiz*, 1988, 28, p. 1740-1743.
- [PRRL97] P. PARDALOS, K. G. RAMAKRISHNAN, M. G. C. RESENDE and Y. LI, Implementation of a Variance Reduction-Based Lower Bound in a Branch-and-Bound Algorithm for the Quadratic Assignment Problem, *SIAM J. Optim.*, 1997, 7, p. 280-294.
- [PW94] P. M. PARDALOS and H. WOLKOWICZ, *The Quadratic Assignment Problem: A Survey of Recent Developments*, P. M. Pardalos, H. Wolkowicz, Ed., Quadratic Assignment and Related Problems, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., 16, American Mathematical Society, Providence, 1994.
- [PW94ed] P. M. PARDALOS and H. WOLKOWICZ, *Quadratic Assignment and Related Problems*, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., 16, American Mathematical Society, Providence, 1994.
- [Que94] T. M. QUERIDO, *Simulated Annealing in the Inversion Graph of the Quadratic Assignment Problem* (in Portuguese), D. Sc. Thesis, COPPE/UFRJ, 1994.
- [Re85] F. RENDL, Ranking Scalar Products to Improve Bounds for the Quadratic Assignment Problem, *EJOR*, 1985, 20, p. 363-372.
- [Ro87] C. ROUCAIROL, A Parallel Branch-and-Bound Algorithm for the Quadratic Assignment Problem, *Discr. Appl. Maths.*, 1987, 18, p. 211-225.

- [RRP96] K. G. RAMAKRISHNAN, M. G. C. RESENDE and P. M. PARDALOS, A Branch-and-Bound Algorithm for the Quadratic Assignment Problem Using a Lower Bound on Linear Programming, *Nonconvex Optim. Appl.*, 1996, 7, p. 57-73.
- [SK89] J. SKORIN-KAPOV, Tabu Search Applied to the Quadratic Assignment Problem, *ORSA J. Comp.*, 1989, 2, p. 33-45.
- [SK94] J. SKORIN-KAPOV, Extensions of a Tabu Search Adaptation to the Quadratic Assignment Problem, *Comput. Opns. Res.*, 1994, 21, p. 855-865.
- [Ta91] E. TAILLARD, Robust Taboo Search for the Quadratic Assignment Problem, *Parallel Comput.*, 1991, 17, p. 443-455.
- [Wh93] D. J. WHITE, A Parametric-Based Heuristic Program for the Quadratic Assignment Problem, *NRL*, 1993, 40, p. 553-568.
- [Wi58] R. J. WIMMERT, A Mathematical Model of Equipment Location, *J.J.E.*, 1958, 9, p. 498-505.
- [WW87] M. WILHELM and T. L. WARD, Solving Quadratic Assignment Problems by "Simulated Annealing", *IEEE Trans.*, 1987, 19, p. 107-119.