A New Method for Quickly Solving Quadratic Assignment Problems

Zhongzhen ZHANG
School of Management, Wuhan University of Technology, Wuhan 430070, China
Email address: zhangzz321@126.com

Abstract

The quadratic assignment problem is solved by a pivoting based algorithm proposed by the author for quadratic programming where the lower bounds of variables taking 1 or 0 alternatively in the computational process. Experiments are conducted by using instances of Nug, Tai, Chr, Had, Rou, Bur, Esc and Tho for n = 12 to 40. Most problems including the famous Nug30 are solved to optimum in several seconds to several minutes. About 80 optimal solutions are presented that are never published in the literature.

1 Introduction

There are a variety of methods for solving the quadratic assignment problem (QAP) such as the Branch and Bound method, greedy randomizing adaptive search procedure, robust taboo search, simulated annealing approach, and so on. The Branch and Bound method is able to obtain an optimal solution. But it requires too much computational time. It is well known that in 2000 Nug30 was solved in a week by 1000 computers which began with a very small bound 6126 while the minimal value was found to be 6124. Most other methods are heuristic and the obtainment of an optimal solution is by chance to some extent.

This paper proposes a method for solving QAP in a special way that is completely different from the known methods. The QAP is solved as a quadratic programming by a pivoting based algorithm proposed by the author. It may be called a parameter quadratic programming where the lower bounds of variables taking 1 or 0 alternatively in the computational process. In this method all the n^2 variables x_{ij} are divided into two groups, one group has $2n - 1$ variables called nonbasic and another group has $n^2 - 2n + 1$ variables called basic. Usually all the basic variables are zeros. For a given assignment (the current solution), by setting k basic variables to be 1, if a feasible solution is formed, it is called a k order basic assignment, $1 \le k \le n$. For small k, say k = 1 or 2, the k order basic assignments and their values of the objective function can be calculated in very small amount of computation. In this way a better solution may be found which has a value of the objective function smaller than that of the current one. Another way to search for a better solution is to construct a series of k order basic assignments as the current basic assignments from small k to larger k, $2 \le k < n$. In the meanwhile pivoting operations are carried out in which basic variables with values zero and negative costs are interchanged with nonbasic variables so that a high order basic assignment with small value of the objective function is translated into a low order one and is easily detected by the parameter method. It is called contraction effect of the pivoting operation and is the key of our method.

2 The fundamental method

The quadratic assignment problem involves assigning n facilities to n fixed locations so as to minimize the total cost of transferring material between the facilities. Let the n by n matrices $D = (d_{ij})$, $F = (f_{ij})$ and $X = (x_{ij})$, where d_{ij} is the distance from location i to j, f_{ij} is the flow from facility i

to j, and x_{ij} = 1 if facility i is assigned to location j, otherwise x_{ij} = 0. The total cost can be formulated as $\mathrm{tr}(DXF^{\mathrm{T}}X^{\mathrm{T}})$[1] and the problem is as follows:

$$\min \mathrm{tr}(DXF^{\mathrm{T}}X^{\mathrm{T}})$$

$$\text{s.t.} \ \sum_{j=1}^{n} x_{ij} = 1,\ i = 1, 2, \dots, n$$

$$-\sum_{i=1}^{n} x_{ij} = -1,\ j = 1, 2, \dots, n$$

$$x_{ij} = 0 \text{ or } 1,\ i, j = 1, 2, \dots, n. \qquad (2.1)$$

where tr(•) denotes the trace of a square matrix.

We will solve it as a quadratic programming problem in which the constraint

$$x_{ij} = 0 \text{ or } 1,\ i, j = 1, 2, \dots, n$$

is replaced by

$$x_{ij} \geq l_{ij},\ i, j = 1, 2, \dots, n$$

where l_{ij} = 0 or 1, and at the very beginning all the l_{ij} are set to zeros. That is we shall first solve the problem

$$\min \ \frac{1}{2}x^{\mathrm{T}}Hx$$

$$\text{s.t. } Ax = b$$

$$x \geq 0. \qquad (2.2)$$

Where

$$H = \begin{pmatrix} d_{11}(F+F^T) & d_{12}F + d_{21}F^T & \cdots & d_{1n}F + d_{n1}F^T \\ d_{12}F^T + d_{21}F & d_{22}(F+F^T) & \cdots & d_{2n}F + d_{n2}F^T \\ \vdots & \vdots & & \vdots \\ d_{1n}F^T + d_{n1}F & d_{2n}F^T + d_{n2}F & \cdots & d_{nn}(F+F^T) \end{pmatrix},$$

is an $n^2 \times n^2$ Hessian matrix, $x = (x_{11}, \dots, x_{1n}, x_{21}, \dots, x_{2n}, \dots, x_{n1}, \dots, x_{nn})^{\mathrm{T}}$ is an n^2-dimensional column, A is the incidence matrix of the n nodes to n nodes bi-partitioned graph and b is a column vector of components 1 and –1 as determined by the first two equality constraints of (2.1). For brevity, it is supposed that a redundant row of (A b) is deleted. Therefore A is an $(2n - 1) \times n^2$ matrix.

We shall solve (2.2) by the pivoting algorithm presented in [2] which is for solving quadratic programming. This algorithm is divided into three stages: (i)construct an initial table, (ii) preprocessing and (iii) main iterations.

The initial table for solving (2.2) is

Table 2.1 Initial table

	e_h	e_a	
h	H	$-A^{\mathrm{T}}$	$\mathbf{0}$
a	A	O	$-b$

Let $A = (A_1, A_2)$ where A_1 is a nonsingular matrix of dimension $(2n - 1) \times (2n - 1)$.

Correspondingly, Table 2.1 is partitioned as Table 2.2.

Table 2.2 Initial table (the partitioned form)

	e_I	e_{II}	e_{III}	
h_I	H_{11}	H_{12}	$-A_1^T$	0
h_{II}	H_{12}^T	H_{22}	$-A_2^T$	0
a	A_1	A_2	O	$-b$

The preprocessing stage is to solve the inverses A_1^{-1} and $-A_1^{-T}$ which yields Table 2.3 where the columns of basic equality constraints and the corresponding rows are deleted.

Table 2.3 Result of the preprocessing

	e_{II}	h_I	
h_{II}	H_{22}'	$A_2^T A_1^{-T}$	σ_{II}'
e_I	$-A_1^{-1}A_2$	O	$A_1^{-1} b$

Where

$$H_{22}' = H_{22} - H_{12}^T A_1^{-1} A_2 - A_2^T A_1^{-T} (H_{12} - H_{11} A_1^{-1} A_2)$$

is called a reduced Hessian matrix, $-A_1^{-1}A_2$ is called a reduced incidence matrix, and

$$\sigma_{II}' = H_{12}^T A_1^{-1} b - A_2^T A_1^{-T} H_{11} A_1^{-1} b.$$

A_1 being nonsingular means that A_1 is corresponding to a spanning tree[3] in the n nodes to n nodes bi-partitioned graph. Each branch of the tree is associated with a value of x_{ij} which is the solution of the equality constraint $Ax = b$ and is given by $A_1^{-1} b$ here.

In the computing programs of the author, n initial feasible solutions are designed for experiments. Take $n = 4$ for example, they are

12340123, 23410234, 34120341, 41230412

which are called **extended assignments** meaning that

$$x_{11} = x_{22} = x_{33} = x_{44} = 1, x_{21} = x_{32} = x_{43} = 0,$$
$$x_{12} = x_{23} = x_{34} = x_{41} = 1, x_{22} = x_{33} = x_{44} = 0,$$
$$x_{13} = x_{24} = x_{31} = x_{42} = 1, x_{23} = x_{34} = x_{41} = 0,$$
$$x_{14} = x_{21} = x_{32} = x_{43} = 1, x_{24} = x_{31} = x_{42} = 0$$

respectively.

In the main iterations stage, all the pivots are elements of $-A_1^{-1}A_2$ and $A_2^T A_1^{-T}$. As soon as a pivoting operation is carried out on a nonzero element (–1 or 1) of $-A_1^{-1}A_2$, another pivoting operation is carried out on the symmetric element in $A_2^T A_1^{-T}$. It is called a double pivoting. Another form of the pivoting operation for quadratic programming is called principal pivoting. But it is never used for quadratic assignment problems.

3 The basic assignment and local search

Now let us consider a detailed form of Table 2.3 as shown by Table 3.1.

Table 3.1 A table in the main iterations stage

	e_1	…	e_N	h_{N+1}	…	h_{N+M}	
h_1	w_{11}	…	w_{1N}	$-w_{N+1,\,1}$	…	$-w_{N+M,\,1}$	σ_1
…	…	…	…	…	…	…	…
h_N	w_{N1}	…	w_{NN}	$-w_{N+1,\,N}$	…	$-w_{M+N,\,N}$	σ_N
e_{N+1}	$w_{N+1,\,1}$	…	$w_{N+1,\,N}$	0	…	0	σ_{N+1}
…	…	…	…	…	…	…	…
e_{N+M}	$w_{N+M,\,1}$	…	$w_{N+M,\,N}$	0	…	0	σ_{N+M}

Where $N = n^2 - 2n + 1$, $M = 2n - 1$; $\sigma_{N+1}, \ldots, \sigma_{N+M}$ are 1 or 0, i.e., the current solution is feasible; e_i is the coefficient vector of $x_i \geq 0$ and h_i is the coefficient vector of the complementary inequality of $x_i \geq 0$. Here we use x_i ($i = 1, 2, \ldots, n^2$) rather than x_{ij} to represent a variable.

In Table 3.1, each basic unit vector e_j is corresponding to $x_j \geq 0$. Now let us change the right side terms of k basic inequalities, say $x_1 \geq 0, x_2 \geq 0, \ldots, x_k \geq 0$, from 0 to 1. By (2.11) of [2], the last column becomes

$$(\sigma_1 + \sum_{j=1}^{k} w_{1j}, \ldots, \sigma_N + \sum_{j=1}^{k} w_{Nj}, \sigma_{N+1} + \sum_{j=1}^{k} w_{N+1,j}, \ldots, \sigma_{N+M} + \sum_{j=1}^{k} w_{N+M,j})^{\mathrm{T}}. \quad (3.1)$$

This operation is denoted by $\{e_1, e_2, \ldots, e_k\} \to \{e_1^{+1}, e_2^{+1}, \ldots, e_k^{+1}\}$ or $\{e_1, e_2, \ldots, e_k\} \to \{e_1, e_2, \ldots, e_k\}^{+1}$.

If the last M components of (3.1) are 1 or 0, we have a new feasible solution and say it to be ***k* order basic assignment**. Obviously, k can not greater than n. And it is easy to prove there are $n! - 1$ k order basic assignment altogether for $k = 1$ to n. As a special case, the current feasible solution is called 0 order basic assignment.

By (6.10) of [2], the change of the value of the objective function is

$$\Delta f = \frac{1}{2}(1, 1, \ldots, 1)\begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1k} \\ w_{12} & w_{22} & \cdots & w_{2k} \\ \vdots & \vdots & & \vdots \\ w_{1k} & w_{2k} & \cdots & w_{kk} \end{pmatrix}\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} + (1, 1, \ldots, 1)\begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_k \end{pmatrix} \quad (3.2)$$

which is called the cost of the operation $\{e_1, e_2, \ldots, e_k\} \to \{e_1^{+1}, e_2^{+1}, \ldots, e_k^{+1}\}$.

Particularly speaking if we increase the right side term of just one basic inequality, say $x_j \geq 0$, by 1, the cost is

$$\Delta f = \frac{1}{2} w_{jj} + \sigma_j \quad (3.3)$$

The above operation is call +1 operation and denoted by $e_j \to e_j^{+1}$. Since the +1 operation involves a small amount of computation, it is the most common way to find a better solution. For this reason, we call it to be +1 search as well. Another way to find a better solution which requires small computation is the +2 search $\{e_i, e_j\} \to \{e_i^{+1}, e_j^{+1}\}$ whose cost is

$$\Delta f = \frac{1}{2}(w_{ii} + w_{jj} + 2w_{ij}) + \sigma_i + \sigma_j. \quad (3.4)$$

Now suppose that we have a table where k basic unit vectors e_j's are associated with $x_j \geq 1$ and which determine a feasible solution of the value f_1 of the objective function. We say the k

vectors to be **catalyzed basic vectors,** denoted by e_j^{+1}'s, and say other basic unit vectors to be **non-catalyzed basic vectors**.

When there are catalyzed basic vectors in a table we can also change one of them to non-catalyzed one, say e_j^{+1} to e_j, and the cost is

$$\Delta f = \frac{1}{2} w_{jj} - \sigma_j. \tag{3.5}$$

If the change is feasible and the cost is negative, we get a solution whose value of the objective function is less than f_1. This operation is denoted by $e_j^{+1} \to e_j$ and is called –1 search.

Similarly there are –2 search and –1&+1 search. The cost of $\{e_i^{+1}, e_j^{+1}\} \to \{e_i, e_j\}$ is

$$\Delta f = \frac{1}{2}(w_{ii} + w_{jj} + 2w_{ij}) - \sigma_i - \sigma_j \tag{3.6}$$

and the cost of $\{e_i^{+1}, e_j\} \to \{e_i, e_j^{+1}\}$ is

$$\Delta f = \frac{1}{2}(w_{ii} + w_{jj} - 2w_{ij}) - \sigma_i + \sigma_j \tag{3.7}$$

We can also do –2&+1, –1&+2 or –2&+2 to search for a better solution. But these operations especially the –2&+2 requires too much amount of computation and seldom used in the computing program of the author.

The feasible solutions obtained by operations +1, +2, –1, –2, –1&+1, –1&+2, –2&+1 or –2&+2 are in the "vicinity" of the current feasible solution or the current basic assignment. Therefore the above operations may be called local search methods.

4 The pivoting operation

Suppose that we have a table with a feasible solution where some basic unit vectors may be catalyzed and where a part of entries are shown by Table 4.1.

Table 4.1 The current table

	e_s'	h_r	
h_s'	w_{ss}	$-w_{rs}^*$	σ_s
e_r	w_{rs}^*	0	σ_r

Table 4.2 Result of the double pivoting

	e_r	h_s'	
h_r	w_{ss}	$-1 / w_{rs}$	$w_{rs}\sigma_s - w_{ss}\sigma_r$
e_s'	$1 / w_{rs}$	0	$-\sigma_r / w_{rs}$

In Table 4.1 $w_{rs} = -1$ or 1, $\sigma_r = 0$ or 1, and $e_s' = e_s$ if $x_s = 0$ or $e_s' = e_s^{+1}$ if $x_s = 1$. Carrying out a double pivoting $e_r \leftrightarrow e_s'$ and $h_s' \leftrightarrow h_r$ there will be Table 4.2. By (6.15) of [2] the change of the value of the objective function is

$$\Delta f = \frac{1}{2} w_{ss}\sigma_r^2 - w_{rs}\sigma_r\sigma_s \tag{4.1}$$

Some special cases called **forward (backward) descent pivoting** and **par pivoting** are as follows.

(i) Forward descent pivoting

It means that (a) $e_s' = e_s$, and there would be a feasible solution if we performed the +1 operation $e_s \to e_s^{+1}$; (ii) $w_{rs} = -1$, $\sigma_r = 1$ and $\frac{1}{2} w_{ss} + \sigma_s < 0$. After the pivoting there will be a feasible solution and the change of the value of the objective function is $\frac{1}{2} w_{ss} + \sigma_s$.

(ii) Backward descent pivoting

It means that (a) $e_s' = e_s^{+1}$, and there would be a feasible solution if we performed the −1 operation $e_s^{+1} \to e_s$; (ii) $w_{rs} = 1$, $\sigma_r = 1$ and $\frac{1}{2}w_{ss} - \sigma_s < 0$. After the pivoting there will be a feasible solution and the change of the value of the objective function is $\frac{1}{2}w_{ss} - \sigma_s$.

(iii) Par pivoting

It means that $e_s' = e_s$, $w_{rs} = -1$ and $\sigma_r = 0$. The par pivoting is carried out usually when $\frac{1}{2}w_{ss} + \sigma_s < 0$ even if it dose not change the value of the objective function. To see the effect of the par pivoting let us consider Table 4.3 where a nonbasic unit vector e_5 is associated with a deviation 0.

Table 4.3 A table for par pivoting

	e_1	e_2	e_3	e_4	h_5	
h_1	w_{11}	w_{12}	w_{13}	w_{14}	1^*	σ_1
h_2	w_{12}	w_{22}	w_{23}	w_{24}	–1	σ_2
h_3	w_{13}	w_{23}	w_{33}	w_{34}	1	σ_3
h_4	w_{14}	w_{24}	w_{34}	w_{44}	0	σ_4
e_5	-1^*	1	–1	0	0	0

Carrying out a pivoting on –1 and then on 1 with asterisks, there will be Table 4.4.

Table 4.4 Result of the par pivoting

	e_5	e_2	e_3	e_4	h_1	
h_5	w_{11}	$-w_{12}-w_{11}$	$-w_{13}+w_{11}$	$-w_{14}$	1	$-\sigma_1$
h_2	$-w_{12}-w_{11}$	$w_{22}+w_{11}+2w_{12}$	$w_{23}-w_{12}+w_{13}-w_{11}$	$w_{24}+w_{14}$	–1	$\sigma_2+\sigma_1$
h_3	$-w_{13}+w_{11}$	$w_{23}-w_{12}+w_{13}-w_{11}$	$w_{33}+w_{11}-2w_{13}$	$w_{34}-w_{14}$	1	$\sigma_3-\sigma_1$
h_4	$-w_{14}$	$w_{24}+w_{14}$	$w_{34}-w_{14}$	w_{44}	0	σ_4
e_1	–1	1	–1	0	0	0

Suppose that performing a +2 operation $\{e_1, e_2\} \to \{e_1^{+1}, e_2^{+1}\}$ in Table 4.3, in which some basic unit vectors except for e_1 and e_2 may be catalyzed, there results in a feasible solution. Then the change of the value of the objective function is

$$\Delta f = \frac{1}{2}(w_{11} + w_{22} + 2w_{12}) + \sigma_1 + \sigma_2.$$

On the other hand the same solution and change can be obtained by performing a +1 operation $e_2 \to e_2^{+1}$ in Table 4.4. If $\frac{1}{2}w_{11} + \sigma_1$ is negative enough, Δf may be negative and a better solution would be found in Table 4.4 which requires a smaller amount of computation than the +2 operation does in Table 4.3.

The property of the double pivoting especially the par pivoting that translates some high order basic assignments into lower ones is called **contraction effect** and is the key to generate a better solution. Of course the double pivoting would change some low order basic assignments to high order ones in the same time. But it is not required by us.

The par pivoting is frequently performed in order to generate a low order basic assignment with a small value of the objective function. It takes more 90% of the amount of computation in solving a QAP. To see the effect of the par pivoting in details, let us consider the reduced incidence matrix and the associated deviation for n = 4 which is shown by Table 4.5 where e_{ij} is the coefficient vector of $x_{ij} \geq 0$, i, j = 1, 2, 3, 4.

Table 4.5 A reduced incidence matrix of n = 4

	e_{12}	e_{13}	e_{14}	e_{23}	e_{24}	e_{31}	e_{34}	e_{41}	e_{42}	
e_{11}	–1	–1	–1	0	0	0	0	0	0	1
e_{22}	–1	–1	–1	–1	–1	1	0	1	0	1
e_{33}	0	–1	–1	–1	–1	0	–1	1	1	1
e_{44}	0	0	–1	0	–1	0	–1	0	0	1
e_{21}	1	1	1	0	0	-1^*	0	–1	0	0
e_{32}	0	1	1	1	1	–1	0	–1	–1	0
e_{43}	0	0	1	0	1	0	1	–1	–1	0

From Table 4.5 we see that the current solution is 1234 and other 23 basic assignments are as follows.

(i) 1 order basic assignments (6 ones):

e_{12}^{+1} — 2134, e_{13}^{+1} — 3124, e_{14}^{+1} — 4123, e_{23}^{+1} — 1324, e_{24}^{+1} — 1423, e_{34}^{+1} — 1243.

(ii) 2 order basic assignments (6 ones):

$\{e_{12}, e_{34}\}^{+1}$ — 2143, $\{e_{13}, e_{31}\}^{+1}$ — 3214, $\{e_{14}, e_{31}\}^{+1}$ — 4213,

$\{e_{14}, e_{41}\}^{+1}$ — 4231, $\{e_{14}, e_{42}\}^{+1}$ — 4132, $\{e_{24}, e_{42}\}^{+1}$ — 1432.

(iii) 3 order basic assignments (8 ones):

$\{e_{12}, e_{23}, e_{31}\}^{+1}$ — 2314, $\{e_{12}, e_{24}, e_{31}\}^{+1}$ — 2413,

$\{e_{12}, e_{24}, e_{41}\}^{+1}$ — 2431, $\{e_{13}, e_{24}, e_{41}\}^{+1}$ — 3421,

$\{e_{13}, e_{34}, e_{41}\}^{+1}$ — 3241, $\{e_{13}, e_{34}, e_{42}\}^{+1}$ — 3142,

$\{e_{14}, e_{23}, e_{41}\}^{+1}$ — 4321, $\{e_{23}, e_{34}, e_{42}\}^{+1}$ — 1342.

(iv) 4 order basic assignments (3 ones):

$\{e_{12}, e_{23}, e_{34}, e_{41}\}^{+1}$ — 2341, $\{e_{13}, e_{24}, e_{31}, e_{42}\}^{+1}$ — 3412,

$\{e_{14}, e_{23}, e_{31}, e_{42}\}^{+1}$ — 4312.

Let e_{21} enter and e_{31} leave the basis there will be Table 4.6.

Table 4.6 Result of the pivoting

	e_{12}	e_{13}	e_{14}	e_{23}	e_{24}	e_{21}	e_{34}	e_{41}	e_{42}	
e_{11}	–1	–1	–1	0	0	0	0	0	0	1
e_{22}	0	0	0	–1	–1	–1	0	0	0	1
e_{33}	0	–1	–1	–1	–1	0	–1	1	1	1
e_{44}	0	0	–1	0	–1	0	–1	0	0	1
e_{31}	1	1	1	0	0	–1	0	–1	0	0
e_{32}	–1	0	0	1	1	1	0	0	–1	0
e_{43}	0	0	1	0	1	0	1	–1	–1	0

From Table 4.6 we can get 23 basic assignments as follows.

(i) 1 order basic assignments (5 ones):

e_{13}^{+1} — 3214, e_{14}^{+1} — 4213, e_{23}^{+1} — 1324, e_{24}^{+1} — 1423, e_{34}^{+1} — 1243.

(ii) 2 order basic assignments (7 ones):

$\{e_{12}, e_{23}\}^{+1}$ — 2314, $\{e_{12}, e_{24}\}^{+1}$ — 2413, $\{e_{12}, e_{21}\}^{+1}$ — 2134, $\{e_{13}, e_{21}\}^{+1}$ — 3124,
$\{e_{14}, e_{21}\}^{+1}$ — 4123, $\{e_{14}, e_{41}\}^{+1}$ — 4231, $\{e_{24}, e_{42}\}^{+1}$ — 1432.

(iii) 3 order basic assignments (9 ones):

$\{e_{12}, e_{24}, e_{41}\}^{+1}$ — 2431, $\{e_{12}, e_{21}, e_{34}\}^{+1}$ — 2143, $\{e_{13}, e_{24}, e_{41}\}^{+1}$ — 3421,
$\{e_{13}, e_{24}, e_{42}\}^{+1}$ — 3412, $\{e_{13}, e_{34}, e_{41}\}^{+1}$ — 3241, $\{e_{14}, e_{23}, e_{41}\}^{+1}$ — 4321,
$\{e_{14}, e_{23}, e_{42}\}^{+1}$ — 4312, $\{e_{14}, e_{21}, e_{42}\}^{+1}$ — 4132, $\{e_{23}, e_{34}, e_{42}\}^{+1}$ — 1342.

(iv) 4 order basic assignments (2 ones):

$\{e_{12}, e_{23}, e_{34}, e_{41}\}^{+1}$ — 2341, $\{e_{13}, e_{21}, e_{34}, e_{42}\}^{+1}$ — 3142.

We see that 3214 and 4213 are 2 order basic assignments in Table 4.5 and are 1 order ones in Table 4.6; 2314 and 2413 are 3 order basic assignments in Table 4.5 and are 2 order ones in Table 4.6; 3412 and 4312 are 4 order basic assignments in Table 4.5 and are 3 order ones in Table 4.6 due to e_{31} leaving the basis.

Example 4.1 Consider a QAP of n = 4 where the distance matrix and flow matrix are

$$D = \begin{pmatrix} 0 & 2 & 1 & 4 \\ 2 & 0 & 1 & 2 \\ 1 & 1 & 0 & 1 \\ 4 & 2 & 1 & 0 \end{pmatrix} \text{ and } F = \begin{pmatrix} 0 & 2 & 2 & 4 \\ 2 & 0 & 3 & 1 \\ 2 & 3 & 0 & 2 \\ 4 & 1 & 2 & 0 \end{pmatrix}.$$

The table associated with the extended assignment 12340123 is as following.

Table 4.7 The initial table in the main iterations stage

	e_{12}	e_{13}	e_{14}	e_{23}	e_{24}	e_{31}	e_{34}	e_{41}	e_{42}	h_{11}	h_{22}	h_{33}	h_{44}	h_{21}	h_{32}	h_{43}	σ_i
h_{12}	32	20	52	12	12	–16	16	–12	–12	1^*	1	0	0	–1	0	0	–28
h_{13}	20	32	32	12	–4	–16	0	0	12	1	1	1	0	–1	–1	0	–22
h_{14}	52	32	96	12	28	–16	32	0	–20	1	1	1	1	–1	–1	–1	–54
h_{23}	12	12	12	24	8	–12	0	–12	0	0	1	1	0	0	–1	0	–10
h_{24}	12	–4	28	8	24	4	16	4	–16	0	1	1	1	0	–1	–1	–18
h_{31}	–16	–16	–16	–12	4	16	0	16	0	0	–1	0	0	1	1	0	4
h_{34}	16	0	32	0	16	0	16	0	–16	0	0	1	1	0	0	–1	–16
h_{41}	–12	0	0	–12	4	16	0	32	12	0	–1	–1	0	1	1	1	–10
h_{42}	–12	12	–20	0	–16	0	–16	12	24	0	0	–1	0	0	1	1	2
e_{11}	$–1^*$	–1	–1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
e_{22}	–1	–1	–1	–1	–1	1	0	1	0	0	0	0	0	0	0	0	1
e_{33}	0	–1	–1	–1	–1	0	–1	1	1	0	0	0	0	0	0	0	1
e_{44}	0	0	–1	0	–1	0	–1	0	0	0	0	0	0	0	0	0	1
e_{21}	1	1	1	0	0	–1	0	–1	0	0	0	0	0	0	0	0	0
e_{32}	0	1	1	1	1	–1	0	–1	–1	0	0	0	0	0	0	0	0
e_{43}	0	0	1	0	1	0	1	–1	–1	0	0	0	0	0	0	0	0

Where e_{ij} is the coefficient vector of $x_{ij} \geq 0$ and h_{ij} is the coefficient vector of the associated inequality, i, j = 1, 2, 3, 4. The last column is the deviation of the nonbasic vector. The value of the objective function of the current solution is 58.

From Table 4.7 we see that the cost of $e_{12} \rightarrow e_{12}^{+1}$ is 32 / 2 – 28 = –12. Therefore the resulting 1 order basic assignment 2134 has a value of 58 – 12 = 46. Performing a forward descent pivoting $e_{11} \leftrightarrow e_{12}$ and $h_{12} \leftrightarrow h_{11}$ to yield Table 4.8 where the current solution is 2134 with a value $f = 46$.

Table 4.8 Result of the forward descent pivoting

	e_{11}	e_{13}	e_{14}	e_{23}	e_{24}	e_{31}	e_{34}	e_{41}	e_{42}	h_{12}	h_{22}	h_{33}	h_{44}	h_{21}	h_{32}	h_{43}	σ_i
h_{11}	32	12	–20	–12	–12	16	–16	12	12	1	–1	0	0	1	0	0	–4
h_{13}	12	24	–8	0	–16	0	–16	12	24	1	0	1	0	0	–1	0	–6
h_{14}	–20	–8	24	0	16	0	16	12	–8	1	0	1	1	0	–1	–1	–6
h_{23}	–12	0	0	24	8	–12	0	–12	0	0	1	1	0	0	–1	0	2
h_{24}	–12	–16	16	8	24	4	16	4	–16	0	1	1	1	0	–1	–1	–6
h_{31}	16	0	0	–12	4	16	0	16	0	0	–1	0	0	1	1^*	0	–12
h_{34}	–16	–16	16	0	16	0	16	0	–16	0	0	1	1	0	0	–1	0
h_{41}	12	12	12	–12	4	16	0	32	12	0	–1	–1	0	1	1	1	–22
h_{42}	12	24	–8	0	–16	0	–16	12	24	0	0	–1	0	0	1	1	–10
e_{12}	–1	–1	–1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
e_{22}	1	0	0	–1	–1	1	0	1	0	0	0	0	0	0	0	0	0
e_{33}	0	–1	–1	–1	–1	0	–1	1	1	0	0	0	0	0	0	0	1
e_{44}	0	0	–1	0	–1	0	–1	0	0	0	0	0	0	0	0	0	1
e_{21}	–1	0	0	0	0	–1	0	–1	0	0	0	0	0	0	0	0	1
e_{32}	0	1	1	1	1	-1^*	0	–1	–1	0	0	0	0	0	0	0	0
e_{43}	0	0	1	0	1	0	1	–1	–1	0	0	0	0	0	0	0	0

Performing a +1 search in Table 4.8 there are e_{11}^{+1} — 1234, e_{13}^{+1} — 3124, e_{14}^{+1} — 4123, e_{34}^{+1} — 2143 with costs 12, 6, 6, 8 respectively. Therefore we can not perform a forward descent pivoting anymore. The cost of $e_{31} \rightarrow e_{31}^{+1}$ is $16 / 2 - 12 = -4 < 0$. But it results in an infeasible solution. Let us perform a par pivoting $e_{32} \leftrightarrow e_{31}$ and $h_{31} \leftrightarrow h_{32}$ to yield Table 4.9 where the current solution has no change.

Table 4.9 Result of the par pivoting

	e_{11}	e_{13}	e_{14}	e_{23}	e_{24}	e_{32}	e_{34}	e_{41}	e_{42}	h_{12}	h_{22}	h_{33}	h_{44}	h_{21}	h_{31}	h_{43}	σ_i
h_{11}	32	28	–4	4	4	–16	–16	–4	–4	1	–1	0	0	1	0	0	–4
h_{13}	28	40	8	4	4	–16	–16	12	8	1	–1	1	0	1	–1	0	–18
h_{14}	–4	8	40	4	36	–16	16	12	–24	1	–1	1	1	1	–1	–1	–18
h_{23}	4	4	4	16	16	–4	0	0	–4	0	0	1	0	1	–1	0	–10
h_{24}	4	4	36	16	48	–20	16	0	–36	0	0	1	1	1	–1	–1	–18
h_{32}	–16	–16	–16	–4	–20	16	0	0	16	0	1	0	0	–1	1	0	12
h_{34}	–16	–16	16	0	16	0	16	0	–16	0	0	1	1	0	0	–1	0
h_{41}	–4	12	12	0	0	0	0	16	12	0	0	–1	0	0	1	1	–10
h_{42}	–4	8	–24	–4	–36	16	–16	12	40	0	1	–1	0	–1	1	1	2
e_{12}	–1	–1	–1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
e_{22}	1	1	1	0	0	–1	0	0	–1	0	0	0	0	0	0	0	0
e_{33}	0	–1	–1	–1	–1	0	–1	1	1	0	0	0	0	0	0	0	1
e_{44}	0	0	–1	0	–1	0	–1	0	0	0	0	0	0	0	0	0	1
e_{21}	–1	–1	–1	–1	–1	1	0	0	1	0	0	0	0	0	0	0	1
e_{31}	0	1	1	1	1	–1	0	–1	–1	0	0	0	0	0	0	0	0
e_{43}	0	0	1	0	1	0	1	–1	–1	0	0	0	0	0	0	0	0

From Table 4.9 we see that the cost of $e_{23} \to e_{23}^{+1}$ is 16 / 2 – 10 = –2 < 0 and which results in a feasible solution 2314 with f = 46 – 2 = 44.

The above example is very small. Generally speaking, after a table is set up we shall not only do the +1 search but also do the +2 search to find a better solution. In the case that there are catalyzed basic vectors in the table we may do the searches of –1, –2, –1&+1, –2&+1 and –1&+2 besides the +1 and +2 searches. Meanwhile the descent pivoting and the par pivoting are carried out if the cost of the leaving unit vector is negative.

5 Results of experiments

Several computing programs are made in Delphi 6.0. They are conducted in this way: First run the program from the first extended assignment until the optimal solution is obtained or a certain number of operation processes are carried out. If the optimal solution is not obtained, run the program from the second extended assignment, and so on.

An entire operation process which doesn't find out a better solution is as follows.

For a given feasible solution,

(i) Do +1 search and +2 search combining with pivoting operations where no basic unit vectors are catalyzed;

(ii) Construct a 2 order basic assignment from which the number of catalyzed basic vectors is increased by the operation –1&+2. Meanwhile local searches and pivoting operations are performed.

(iii) Update the table by transferring the last k order basic assignment into a 0 order basic assignment or construct another feasible solution by using previous feasible solutions, then return to (i).

The operation –1&+2 of (ii) is imposed no matter whether the cost is negative or not so as to increase the number of catalyzed basic vectors by 1 every time until a certain number, say $n \setminus 2$. Other operations are done only if their costs are negative.

If a better solution is found in the midway of the above operation process, the operation process ends and then another one follows.

The difficulty is how many operation processes should be set if each of which can not find out a better solution. If the number of operation processes is small, a better solution may be missed. And if the number is large, there would run much time without finding a better solution.

Table 5.1 gives the results of the program DelQAPpvt run on a personal computer of a 2GHz CPU and a 2G memory. Since the minimal values of the objective functions are known from QAPLIB Home Page, the experiment is designed as that as soon as the minimal value occurs or 1000 operation processes are carried out without finding a better solution the experiment ends for each initial extended assignment. The "Max No. of processes" is the maximal number of operation processes between two better solutions. Usually the value of the objective function of the initial extended assignment is big. Many better solutions (values of the objective functions become smaller and smaller) occur just in one or two operation processes. As the value becomes very small, the number of operation processes increases rapidly.

From table 5.1 we see that most instances are solved to optimum in several seconds to several minutes.

Table 5.1 Computing results of DelQAPpvt

	Time (seconds)	Initial solution	Max No. of processes		Time (seconds)	Initial solution	Max No. of Processes
Nug12	0.19	1st	30	Rou12	0.14	1st	13
Nug14	2.33	1st	123	Rou15	3.48	1st	215
Nug15	0.14	1st	13	Rou20	46.00	1st	620
Nug16a	1.47	1st	42	Chr12a	1.28	1st	209
Nug16b	0.17	1st	11	Chr12b	0.53	1st	83
Nug17	6.45	1st	249	Chr12c	0.45	1st	44
Nug18	6.03	1st	164	Chr15a	7.91	2nd	392
Nug20	8.69	1st	81	Chr15b	2.92	1st	169
Nug21	17.17	1st	191	Chr15c	11.76	1st	568
Nug22	5.91	1st	41	Chr18a	34.83	1st	326
Nug24	26.28	1st	90	Chr18b	0.69	1st	24
Nug25	11.86	1st	46	Chr20a	33.13	3rd	201
Nug27	90.13	1st	82	Chr20b	76.17	1st	253
Nug28	77.61	1st	125	Chr20c	44.00	1st	322
Nug30	44.73	1st	41	Chr22a	23.63	1st	86
Tai12a	0.17	1st	13	Chr22b	50.41	4th	155
Tai12b	0.37	1st	41	Chr25a	11.17	2nd	27
Tai15a	4.19	1st	298	Bur26a	183.94	1st	131
Tai15b	0.17	1st	14	Bur26b	203.73	1st	398
Tai17a	10.23	1st	196	Bur26c	39.70	1st	52
Tai20a	58.37	1st	649	Bur26d	63.67	2nd	45
Tai20b	10.36	1st	121	Bur26e	271.58	1st	162
Tai25a	387.70	1st	710	Bur26f	41.13	1st	33
Tai25b	433.83	1st	890	Bur26g	54.16	1st	41
Tai30a	291.36	3rd	256	Bur26h	359.09	1st	281
Tai30b	1042.91	1st	481	Esc32a	90.92	1st	45
Tai35a	1462.44	2nd	267	Esc32b	52.91	1st	27
Tai40b	5332.30	2nd	401	Esc32c	1.03	1st	1
Had12	0.31	1st	43	Esc32d	1.25	1st	1
Had14	0.75	1st	81	Esc32e	0.56	1st	1
Had16	0.51	1st	24	Esc32g	0.84	1st	1
Had18	0.75	1st	27	Esc32h	83.45	1st	52
Had20	2.77	1st	43	Tho30	356.61	1st	209
				Tho40	6263.30	1st	981

Some new solutions are given in table 5.2 that are different from those published on QAPLIB Home Page. Each of Esc32a-h has many optimal solutions. Just one of them is listed.

Table 5.2 New solutions for some instances of QAP

	value	Solution
Nug12	578	(5, 6, 10, 2, 4, 8, 11, 1, 12, 7, 9, 3)
Nug15	1150	(9, 8, 13, 2, 1, 11, 7, 14, 3, 4, 12, 5, 6, 15, 10)
		(10, 15, 6, 5, 12, 4, 3, 14, 7, 11, 1, 2, 13 ,8 , 9)
		(12, 5 ,6, 15, 10, 11, 7, 14, 3, 4, 9, 8, 13, 2, 1)
Nug16b	1240	(5, 1, 6, 14, 3, 7, 10, 15, 11, 9, 2, 4, 8, 13, 12, 16)
		(5, 3, 11, 8, 1, 7, 9, 13, 6, 10, 2, 12, 14, 15, 4, 16)
Nug18	1930	(9, 3, 10, 6, 13, 14, 2, 12, 7, 5, 18, 15, 8, 1, 17, 16, 4, 11)
Nug20	2570	(6, 1, 7, 5, 17, 13, 8, 20, 15, 19, 16, 11, 12, 2, 4, 9, 3, 10, 14, 18)
		(9, 3, 10, 14, 18, 16, 11, 12, 2, 4, 13, 8, 20, 15, 19, 6, 1, 7, 5, 17)
		(17, 5, 7, 1, 6, 19, 15, 20, 8, 13, 4, 2, 12, 11, 16, 18, 14, 10, 3, 9)
Nug21	2438	(5, 2, 13, 9, 3, 21, 4, 15, 6, 10, 16, 11, 18, 14, 17, 12, 1, 7, 8, 19, 20)
		(17, 12, 1, 7, 8, 19, 20, 15, 6, 10, 16, 11, 18, 14, 5, 2, 13, 9, 3, 21, 4)
		(20, 19, 8, 7, 1, 12, 17, 14, 18, 11, 16, 10, 6, 15, 4, 21, 3, 9, 13, 2, 5)
Nug22	3596	(5, 13, 6, 12, 16, 11, 22, 18, 4, 14, 15, 2, 21, 9, 10, 7, 3, 1, 19, 8, 20, 17)
		(15, 14, 4, 18, 22, 11, 16, 12, 6, 13, 5, 17, 20, 8, 19, 1, 3, 7, 10, 9, 21, 2)
		(17, 20, 8, 19, 1, 3, 7, 10, 9, 21, 2, 15, 14, 4, 18, 22, 11, 16, 12, 6, 13, 5)
Nug24	3488	(2, 5, 13, 6, 12, 24, 21, 16, 9, 7, 10, 1, 14, 3, 18, 22, 19, 15, 20, 4, 23, 11, 8, 17)
		(17, 24, 1, 19, 8, 20, 15, 12, 22, 18, 23, 14, 5, 6, 10, 7, 3, 11, 2, 13, 21, 9, 16, 4)
		(20, 4, 23, 11, 8, 17, 14, 3, 18, 22, 19, 15, 21, 16, 9, 7, 10, 1, 2, 5, 13, 6, 12, 24)
		(24, 12, 6, 13, 5, 2, 1, 10, 7, 9, 16, 21, 15, 19, 22, 18, 3, 14, 17, 8, 11, 23, 4, 20)
Nug25	3744	(5, 2, 18, 24, 17, 11, 25, 16, 21, 12, 20, 8, 3, 14, 4, 15, 9, 6, 7, 23, 22, 1, 19, 10, 13)
		(13, 10, 19, 1, 22, 23, 7, 6, 9, 15, 4, 14, 3, 8, 20, 12, 21, 16, 25, 11, 17, 24, 18, 2, 5)
		(13, 23, 4, 12, 17, 10, 7, 14, 21, 24, 19, 6, 3, 16, 18, 1, 9, 8, 25, 2, 22, 15, 20, 11, 5)
		(17, 24, 18, 2, 5, 12, 21, 16, 25, 11, 4, 14, 3, 8, 20, 23, 7, 6, 9, 15, 13, 10, 19, 1, 22)
		(22, 1, 19, 10, 13, 15, 9, 6, 7, 23, 20, 8, 3, 14, 4, 11, 25, 16, 21, 12, 5, 2, 18, 24, 17)
Nug27	5234	(5, 10, 25, 27, 1, 11, 23, 16, 21, 13, 24, 2, 20, 9, 8, 26, 4, 7, 14, 18, 19, 15, 6, 3, 22, 12, 17)
		(5, 18, 21, 19, 9, 17, 23, 12, 25, 15, 22, 8, 26, 1, 2, 20, 6, 3, 14, 10, 27, 13, 4, 7, 24, 16,11)
		(13, 10, 25, 27, 1, 11, 22, 24, 21, 12, 15, 2, 20, 9, 18, 26, 6, 16, 23, 8, 19, 14, 7, 4, 5, 3, 17)

Table 5.2 (continue)

	value	Solution
Nug28	5166	(11, 8, 20, 28, 1, 9, 18, 26, 16, 17, 19, 10, 15, 7, 14, 27, 4, 13, 25, 6, 22, 12, 5, 3, 24, 2, 21, 23)
		(11, 14, 16, 22, 7, 13, 18, 24, 20, 19, 17, 12, 21, 1, 8, 23, 4, 9, 25, 2, 28, 10, 3, 5, 26, 6, 15, 27)
		(18, 15, 13, 7, 22, 16, 11, 5, 9, 10, 12, 17, 8, 28, 21, 6, 25, 20, 4, 27, 1, 19, 26, 24, 3, 23, 14, 2)
Nug30	6124	(15, 18, 27, 3, 14, 20, 23, 22,11, 16, 30, 4, 17, 1, 8, 7, 19, 25, 26, 24, 10, 9, 29, 28, 5, 12, 6, 13, 2, 21)
		(20, 14, 3, 27, 18, 15, 4, 30, 16, 11, 22, 23, 25, 19, 7, 8, 1, 17, 28, 29, 9, 10, 24, 26, 21, 2, 13, 6, 12, 5)
		(21, 2, 13, 6, 12, 5, 28, 29, 9, 10, 24, 26, 25, 19, 7, 8, 1, 17, 4, 30, 16, 11, 22, 23, 20, 14, 3, 27, 18, 15)
Chr18b	1534	(2, 1, 3, 4, 6, 5, 9, 8, 12, 7, 15, 10, 18, 13, 17, 14, 16, 11)
		(2, 3, 1, 6, 4, 5, 7, 8, 10, 9, 13, 12, 16, 11, 17, 14, 18, 15)
		(6, 3, 9, 2, 12, 1, 15, 4, 18, 5, 17, 8, 14, 7, 13, 10, 16, 11)
		(7, 4, 10, 1, 13, 2, 16, 3, 17, 6, 14, 9, 11, 12, 8, 15, 5, 18)
		(9, 12, 6, 11, 5, 10, 8, 13, 7, 16, 4, 17, 1, 14, 2, 15, 3, 18)
		(14, 15, 11, 12, 8, 9, 7, 6, 10, 5, 13, 4, 16, 1, 17, 2, 18, 3)
Chr20a	2192	(3, 20, 7, 18, 9, 12, 19, 4, 10, 11, 1, 6, 15, 8, 5, 2, 14, 16, 17, 13)
Chr22b	6194	(10, 19, 3, 1, 20, 2, 6, 4, 7, 8, 17, 13, 11, 15, 21, 12, 9, 5, 22, 14, 18, 16)
Chr25a	3796	(25, 12, 5, 3, 18, 4, 16, 8, 20, 10, 14, 6, 23, 15, 24, 19, 13, 1, 21, 11, 17, 2, 22, 7, 9)
Had14	2724	(8, 13, 10, 11, 12, 5, 2, 14, 3, 6, 7, 1, 9, 4)
Had20	6922	(8, 15, 1, 6, 14, 19, 7, 11, 16, 12, 10, 17, 2, 20, 5, 3, 4, 9, 18, 13)
		(8, 15, 1, 14, 6, 19, 7, 11, 16, 12, 10, 17, 2, 20, 5, 3, 4, 9, 18, 13)
		(8, 15, 16, 6, 14, 19, 7, 11, 1, 12, 10, 5, 3, 20, 2, 17, 4, 9, 18, 13)
		(8, 15, 16, 6, 14, 19, 7, 11, 1, 12, 10, 17, 2, 20, 5, 3, 4, 9, 18, 13)
		(8, 15, 16, 14, 6, 19, 7, 11, 1, 12, 10, 5, 3, 20, 2, 17, 4, 9, 18, 13)
Bur26a	5426670	(11, 15, 26, 7, 4, 12, 13, 6, 2, 18, 9, 5, 1, 21, 8, 14, 3, 19, 20, 10, 17, 25, 16, 24, 22, 23)
		(15, 11, 26, 7, 4, 12, 13, 6, 2, 18, 5, 9, 1, 21, 8, 14, 3, 19, 20, 17, 10, 25, 24, 16, 23, 22)
		(26, 11, 15, 7, 4, 12, 13, 6, 2, 18, 1, 9, 5, 21, 8, 14, 3, 19, 20, 25, 10, 17, 16, 24, 23, 22)
Bur26b	3817852	(10, 15, 25, 7, 4, 6, 14, 22, 23, 18, 9, 5, 1, 21, 8, 12, 3, 19, 20, 11, 17, 26, 16, 24, 13, 2)
		(17, 11, 25, 7, 4, 13, 2, 22, 23, 18, 5, 9, 1, 21, 8, 12, 3, 19, 20, 15, 10, 26, 16, 24, 14, 6)
		(25, 15, 11, 7, 4, 2, 13, 22, 23, 18, 1, 5, 9, 21, 8, 12, 3, 19, 20, 26, 17, 10, 24, 16, 14, 6)

Table 5.2 (continue)

	Value	Solution
		(26, 17, 10, 7, 4, 2, 13, 22, 23, 18, 1, 5, 9, 21, 8, 12, 3, 19, 20, 25, 15, 11, 24, 16, 6, 14)
Bur26c	5426795	(2, 12, 3, 13, 16, 11, 25, 10, 15 ,9, 8, 18, 19, 20, 4, 21, 1, 14, 5, 6, 24, 22, 23, 7, 17, 26)
		(12, 3, 2, 13, 16, 25, 11, 15, 10, 9, 18, 19, 8, 20, 4, 21, 1, 5, 14, 24, 22, 6, 23, 7, 17, 26)
Bur26d	3821225	(3, 13, 11, 2, 16, 8, 26, 21, 15, 9, 19, 12, 18, 20, 23, 25, 14, 1, 5, 6, 22, 24, 7, 4, 10, 17)
		(6, 24, 13, 2, 16, 10, 17, 15, 21, 9, 19, 18, 12, 20, 23, 25, 14, 5, 1, 3, 11, 22, 4, 7, 8, 26)
		(11, 22, 6, 2, 16, 10, 17, 15, 21, 9, 18, 12, 19, 20, 23, 25, 14, 5, 1, 24, 13, 3, 7, 4, 26, 8)
		(24, 3, 22, 2, 16, 10, 17, 21, 15, 9, 18, 19, 12, 20, 23, 25, 14, 1, 5, 11, 6, 13, 7, 4, 26, 8)
Bur26e	5386879	(4, 14, 13, 7, 16, 26, 25, 1, 17, 15, 20, 12, 18, 19, 3, 8, 21, 5, 9, 6, 24, 10, 2, 22, 23, 11)
		(4, 14, 13, 7, 16, 26, 25, 17, 1, 15, 20, 12, 18, 19, 3, 8, 21, 9, 5, 6, 24, 10, 22, 2, 11, 23)
		(13, 14, 4, 7, 16, 26, 25, 17, 1, 15, 18, 12, 20, 19, 3, 8, 21, 9, 5, 10, 24, 6, 22, 2, 11, 23)
Bur26f	3782044	(13, 3, 6, 17, 16, 26, 23, 1, 10, 15, 18, 19, 20, 12, 14, 25, 21, 5, 9, 24, 7, 2, 22, 4, 11, 8)
		(24, 2, 7, 17, 16, 11, 8, 1, 10, 15, 18, 20, 19, 12, 14, 25, 21, 5, 9, 13, 6, 3, 22, 4, 26, 23)
		(24, 6, 7, 17, 16, 26, 23, 1, 10, 15, 18, 20, 19, 12, 14, 25, 21, 5, 9, 13, 2, 3, 22, 4, 8, 11)
Bur26g	10117172	(11, 2, 22, 23, 13, 25, 24, 8, 1, 21, 4, 7, 20, 18, 12, 15, 9, 19, 5, 16, 6, 26, 14, 3, 10, 17)
Bur26h	7098658	(13, 2, 11, 12, 6, 10, 25, 1, 8, 21, 7, 20, 4, 18, 14, 15, 9, 5, 19, 3, 22, 16, 23, 26, 24, 17)
		(13, 11, 22, 12, 6, 10, 25, 1, 8, 21, 7, 4, 20, 18, 14, 15, 9, 5, 19, 3, 16, 2, 26, 23, 24, 17)
		(22, 16, 13, 12, 6, 25, 10, 8, 1, 21, 20, 4, 7, 18, 14, 15, 9, 19, 5, 2, 11, 3, 23, 26, 24, 17)
Esc32a	130	(17, 19, 20, 28, 27, 25, 18, 6, 11, 31, 9, 21, 1, 3, 4, 7, 13, 12, 10, 23, 16, 29, 8, 15, 30, 5, 14, 26, 2, 24, 22, 32)
Esc32b	168	(15, 7, 16, 8, 5, 6, 11, 3, 27, 31, 19, 23, 12, 4, 28, 32, 20, 24, 1, 2, 17, 21, 18, 22, 25,14, 30, 29, 26, 10, 13, 9)
Esc32c	642	(17, 25, 19, 18, 4, 6, 5, 8, 1, 10, 9, 11, 12, 13, 14, 15, 16, 2, 3, 7, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32)

Table 5.2 (continue)

	Value	Solution
Esc32d	200	(2, 12, 3, 18, 9, 7, 6, 5, 8, 10, 4, 11, 26, 1, 15, 16, 13, 14, 19, 20, 21, 22, 17, 23, 24, 25, 27, 28, 29, 30, 31, 32)
Esc32e	2	(8, 1, 3, 4, 5, 6, 20, 7, 2, 10, 11, 12, 9, 13, 14, 16, 15, 17, 18, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32)
Esc32g	6	(5, 2, 1, 3, 4, 6, 14, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32)
Esc32h	438	(1, 19, 29, 22, 12, 4, 30, 25, 9, 7, 27, 11, 21, 6, 5, 13, 14, 31, 10, 28, 8, 3, 23, 26, 17, 2, 32, 15, 24, 18, 20, 16)
Tai30a	1818146	(19, 18, 4, 24, 30, 25, 5, 7, 1, 22, 28, 20, 11, 13, 9, 16, 8, 10, 17, 21, 12, 29, 2, 15, 3, 14, 26, 27, 23, 6)
Tai35a	2422002	(19, 9, 28, 12, 7, 33, 13, 26, 5, 2, 31, 16, 24, 27, 22, 15, 3, 30, 29, 11, 6, 25, 21, 23, 34, 20, 18, 4, 10, 1, 14, 8, 32, 35, 17)
Tho30	149936	(11, 7, 14, 16, 23, 21, 22, 15, 18, 20, 3, 5, 24, 4, 17 , 8, 26, 13, 6, 12, 19, 29, 1, 27, 2, 28, 30, 25, 10, 9)
Tho40	240516	(31, 35, 27, 26, 13, 5, 37, 38, 25, 29, 8, 21, 32, 22, 4, 9, 36, 12, 6, 30, 16, 33, 28, 18, 17, 11, 10, 1, 15, 20, 14, 39, 3, 34, 7, 24, 23, 19, 2, 40)

Tai40a is difficult to solve. It is just found a solution of the value 3146258. It is greater than 3139370 by the Ro-TS method.

The contraction effect of pivoting operations occurs to us the formation of substance that is under certain conditions or control while the formation of a good solution of QAP is depending on the appropriate control of pivoting operations. It is hopeful to discover the deep mechanism of QAP and develop more efficient computing method.

References

[1] G Finke, R E Burkard, F Rendl. Quadratic assignment problems. Annals of Discrete Mathematics, 1987(31): 61-82

[2] Zhongzhen Zhang. An efficient method for solving the local minimum of indefinite quadratic programming, on QUADRATIC PROGRAMMING PAGER, N.I.M. Gould and Ph.L.Toint, http://www.numerical.rl.uk/qp/qp.html

[3] Zhongzhen Zhang. Convex Programming: Pivoting Algorithms for Portfolio Selection and Network Optimization. Wuhan University Press, 2004(in Chinese)