

## **A New Heuristic Procedure for Quadratic Assignment Problems**

**Najla Akram AL-Saati**

*dr.najla\_alsaati@uomosul.edu.iq*

*College of Computer Sciences and Mathematics*

*University of Mosul, Iraq*

**Received on: 27/07/2002**

**Accepted on: 01/04/2003**

### **Abstract**

In this work, a new heuristic procedure is developed for the solution of Quadratic assignment problems after illustrating various known procedures, and an attempt to increase the efficiency of near-optimal solutions obtained from known heuristic procedures is carried out. Quadratic assignments are used for solving a wide range of problems such as the design of control panels to minimize execution time and the assignment of manufacturing departments among locations to achieve optimal product movements.

This work is based on developing a new heuristic procedure using an improved decision procedure developed from the NP-complete nature and environment of assignment problems. In order to assess the efficiency of the new procedure, which depends on constructing the solution, a comparison is made with that of VNZ (for Vollmann, Nugent and Zartler) which is considered to be one of the most efficient procedures used for solving such problems, and which depends on improving a given initial solution. Final results show a distinctive improvement in the solutions of the various randomly generated problems, and obviously, evidence of the exact solution gained using total enumeration techniques is far essential, as a measure, in the comparisons carried out between the resulting solutions of the mentioned procedures.

Three different tests are carried out in this work including 26 randomly generated problems, each is represented by a

distance and a volume flow matrix with different matrix dimensions, to be employed in the process of evaluating the efficiency of the improved heuristic procedure in terms of solution accuracy and processing time consumption. C++ was employed in constructing the program structures of the mentioned procedures.

**Keywords:** Quadratic Assignment Problems, Heuristic.

### إجراء حدسي جديد لمسائل التعيين التربيعة

نجلاء اكرم الساعاتي

كلية علوم الحاسوب والرياضيات، جامعة الموصل

تاريخ استلام البحث: 2002/07/27 تاريخ قبول البحث: 2003/04/01

#### الملخص

يشمل هذا العمل تطوير إجراء حدسي إيجادي جديد لحل مسائل التعيين التربيعة وذلك بعد استعراض الإجراءات المتوافرة ومحاولة زيادة كفاءة الحلول القريبة الأمثلية المستحصلة باستخدام الإجراءات الإيجادية الحدسية المعروفة. تستخدم التعيينات التربيعة في حل مجموعة واسعة من المسائل منها مسائل تصميم لوحات التحكم لتقليص وقت التنفيذ وتوزيع أقسام الإنتاج على المواقع لتحقيق أفضل كلفة لانتقال المنتجات بين الأقسام.

يرتكز العمل الأساسي للبحث على تطوير إجراء إيجادي جديد من خلال عملية اتخاذ قرار محسنة مشتقة من طبيعة مسائل التعيين التربيعة والتي تعتبر من نوع آ. (NP-Complete). ولتقييم كفاءة الطريقة المطورة ، والتي اعتمدت أسلوب بناء الحل ، ومقارنتها مع إجراء (Vollmann, Nugent and Zartler –VNZ) التي تعتبر إحدى كفاً الإجراءات المستخدمة لحل مثل هذه المسائل ، والتي تعتمد أسلوب تحسين حل أولي معطى. توضح النتائج النهائية تحسناً ملحوظاً في حلول المسائل المختلفة المولدة عشوائياً ، ومن الطبيعي أن وجود الحل الدقيق المستحصل

بإجراءات السرد الكلي أكثر أهمية وضرورة كمقياس لعمليات المقارنة بين الحلول الناتجة من الإجراءات المذكورة.

تم إجراء ثلاثة أنواع مختلفة من التجارب في هذا العمل حيث تضمنت 26 مسألة مولدة بشكل عشوائي يمثل كل منها بمصفوفة البعد أو المسافة ومصنوفة انتقال الحمل وبإبعاد مصفوفات مختلفة ، وذلك لتستخدم في عملية تقييم كفاءة الإجراء المحسن في هذا العمل من حيث دقة الحل والوقت التنفيذي المستغرق. تم توظيف لغة ++C في بناء الهياكل البرمجية للإجراءات المستخدمة.  
**كلمات مفتاحية:** مسائل التعيين التربيعة، الحدس.

### **1- Introduction:**

The problem of assigning  $n$  new facilities to  $n$  sites when there is an interchange between new facilities is referred to as a Quadratic Assignment Problem (QAP). It must be considered to assign exactly one facility to each site. For example, sites may be rooms in a plant, and the facilities might be departments to be assigned to the rooms. Alternatively, the facilities might be new machines to be located in a job shop, and sites might be the possible locations of new facilities.

The QAP is a combinatorial problem that falls into the NP-Complete type of problems, due to the large number of possible assignments ( $n!$ ) available in the solution space of the problem. [Lauriere, 1990]

A number of location problems can be correctly formulated as quadratic assignment problems. The design of control panels to minimize the expected time required to execute a sequence of operations is one illustration of a quadratic assignment problem. The location of items in storage bins in a storeroom is another example of a QAP. Another problem is that of the stock keeper, who would want the items to be placed in such a way that the time required for him to fill a customer's order would be minimized. [Frances, White 1974]

Procedures that have been developed to find least-cost solutions to the quadratic assignment problem are of two types: Exact and Near Optimal. Exact procedures do in fact find least and optimal cost assignment, but they have not yet been developed to the point where they are of significant practical value due to the need of total enumeration; all the exact procedures of interest developed thus far are implicit enumeration procedures. Thus heuristic solution procedures have received a considerable attention and many algorithms have evolved to generate good solutions [Nanda, Khaopravetch, 1987]. For the problem being considered, a heuristic procedure may be characterized as one that has intuitive appeal and seems reasonable, such a procedure might be called commonsense procedure.

Many heuristic procedures have been developed for the QAP such as that of Hillier [White 1974], Kumphol Khaopravetch [Khaopravetch 1990] described a program to calculate the cost and efficiency of any solution developed by the user. Nanda and Weinrarten [Nanda, 1974] proposed a formula to calculate the mean and standard deviation of all ( $n!$ ) assignment costs without actually enumerating them. Nanda [Nanda, 1987] put forward an interactive decisional algorithm for facility planning. Vollmann, Nugent and Zartler [White 1974] state that their procedure produces good results, which has less storage needs than any procedure examined.

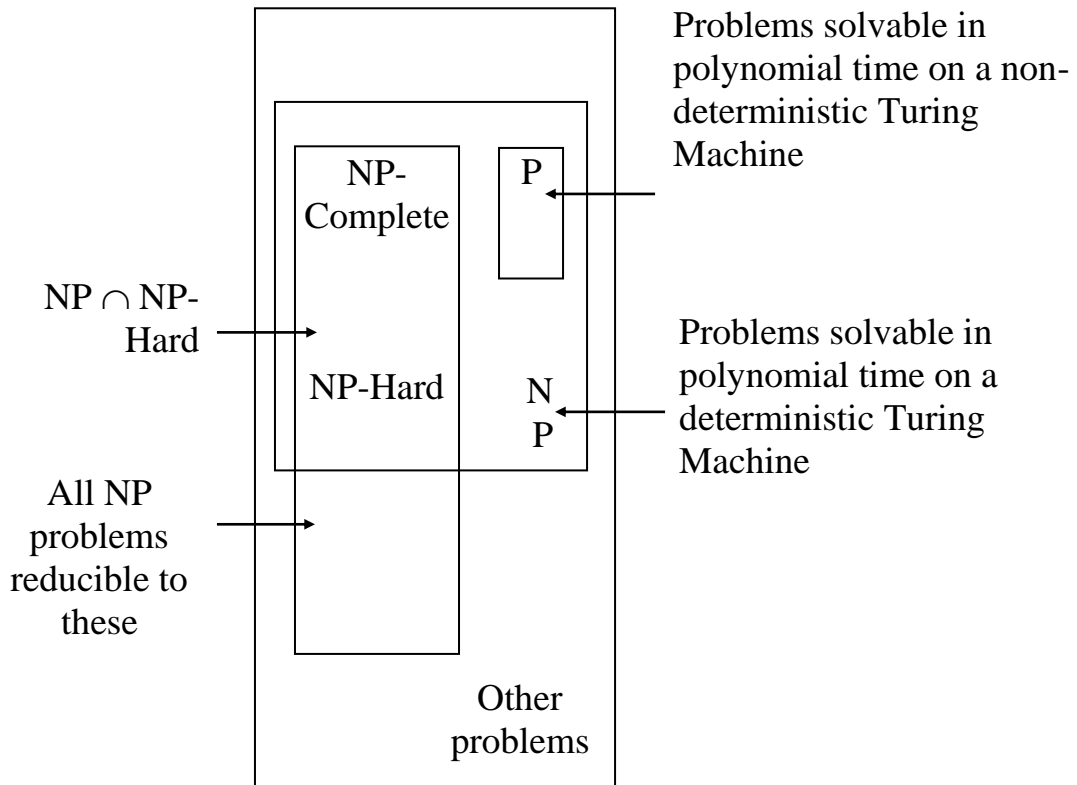
More recently Ahuja, Orlin and Tiwari [Ahuja et. al., 1997] proposed a greedy genetic algorithm for solving this problem, they state that out of 132 instance of test problems with variant sizes, the greedy genetic algorithm obtained the best known solution for 103 instances, and for the remaining instances (except one) found solutions within 1% of the best known solution.

But due to the fact that genetic algorithms are applied to improve already obtained solutions, this work focuses on

building better initial solutions to be more appropriate for later genetic applications.

**2- Problem Formulation:**

This type of problems is an NP-Complete problem, a problem is said to be NP-Complete if and only if it is NP-Hard and it is in NP, the whole relationship is clarified in (Figure-1). [Lauriere, 1990]



**Figure-1 Relationship between problem classes**

The QAP problem can be formulated as follows:

Given  $n$  new facilities and sites, it is required to:

$$\text{Minimize } f(x) = \frac{1}{2} \sum_{i=1}^n \sum_{k=1}^n \sum_{j=1}^n \sum_{h=1}^n C_{ikjh} X_{ik} X_{jh} \dots\dots\dots(1)$$

Subject to  $\sum_{i=1}^n X_{ik} = 1, k=1, \dots, n \dots\dots\dots(2)$

$\sum_{k=1}^n X_{ik} = 1, i=1, \dots, n. \dots\dots\dots(3)$

$X_{ik}=0$  or  $1$ , for all  $i, k$

Where  $C_{ikjh}$  denote the annual cost of having facility  $i$  located at site  $k$  and facility  $j$  located at site  $h$ .  
 $X_{ik}$  (the decision variable) equal one if facility  $i$  is located at site  $k$  and equal zero otherwise.

If facility  $i$  is located at site  $k$  and facility  $j$  at site  $h$  then  $X_{ik}$  and  $X_{jh}$  both equal 1 and the cost term  $C_{ikjh}$  is included in the total cost calculation.

The first set of constraints (eq.2) ensures that exactly one facility is assigned to each site; the second set (eq.3) of constraints results in each facility being assigned to one site. [Frances, White, 1974]

Given an assignment of facilities to sites, it has been found instructive to let  $a(i)$  denote the number of the site to which facility  $i$  is assigned, and let  $a$  be the assignment vector  $a = ( a(1) , a(2) , \dots, a(n) )$  so that the  $i$ 'th component of the assignment vector  $a$  is the number of the site to which facility  $i$  has been assigned.

The total cost for the assignment  $a$  is computed by:- (The division by 2 is due to the symmetry of D and W matrixes)

$$TC(a) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n W_{ij} D_{a(i)a(j)} \dots\dots\dots(4)$$

Where  $D_{kh}$  (site-distance) denote an appropriately determined distance between site  $k$  and site  $h$

$W_{ij}$  is a constant of proportionality converting the distance facilities  $i$  and  $j$ , for all  $i < j$ , into a cost.

Given an assignment of facilities to locations, the total cost for any facility  $k$  is given by:

$$P_k(a) = \sum_{j=1}^n W_{kj} D_{a(k)a(j)}, \text{ for } k = 1, \dots, n \dots\dots\dots(5)$$

A simple approach for finding an assignment with a lower cost is to try interchanging different combinations of two facilities given  $n$  facilities, and pointing out two facilities, say  $u$  and  $v$ , which by interchanging their locations will yield an assignment ( $a'$ ) with a lower total cost ( $TC(a')$ ).

As  $TC(a)$  and  $TC(a')$  include a number of common terms, that is, all those not involving facilities  $u$  or  $v$ , it is easy to compute  $TC(a) - TC(a')$  by subtracting those terms involving facilities  $u$  or  $v$  in  $TC(a')$  from those in  $TC(a)$ .

Thus the change in cost obtained by interchanging the locations of facilities  $u$  and  $v$  for a given assignment, denoted by  $DTC_{uv}(a)$ , is readily obtained [White 1974]:

$$DTC_{uv}(a) = \sum_{i=0}^n (W_{iu} - W_{iv}) [D_{a(i)a(u)} - D_{a(i)a(v)}] - 2W_{uv} D_{a(u)a(v)} \dots\dots\dots(6)$$

**3- Solution Procedures:-**

**3-1-Enumerative & Heuristic Procedures:**

Solving quadratic assignment problems using enumerative procedures is guaranteed to yield the optimal solution, but total enumeration requires the explicit consideration of all ( $n!$ ) assignments. When  $n=12$ ,  $n! = 4.79 \times (10^8)$ , it has been estimated that the time required for solving a problem with  $n=14$  using total enumeration on a Pentium III (1GHz) computer would take about **128 Hour** as shown in (Table-2) and increase rapidly with  $n$ . Even if a later-generation computer is to be used, it will still be evident that the use of total enumeration would be infeasible (i.e., cannot be solved by polynomial time algorithms).

So it is clear that as the size of the problem grows larger, the enumerative techniques become more time consuming and collapse, at the end, in the combined explosion (increase exponentially with  $n$ ) failing to produce any solution. It was this fact that gave the starting trigger for heuristic algorithm to evolve. Although not promising to find the optimal solution, heuristics are designed to generate near-optimal solution close to optimal [Flouds 1984].

Most heuristic procedures belong to a one or a mixture of four general strategies [Daellenbach 1978], they are:

Solution-Building (Construction) Strategies.

Break-Make (component analysis) Strategies.

Solution-Modification (improvement) Strategies.

Search-Learning (learning) Strategies.

The procedures referred to in this work belong to two of the strategies above, the first, that is VNZ, which is to be explained in the next section, belong to the third strategy and works by accepting an initial solution in an attempt to improve it using certain rules. The new heuristic procedure presented here belongs to the first strategy, it works by trying to construct the solution completely from certain rules.

### **3-2- The Vollmann, Nugent and Zartler Procedure:**

In an attempt to find a satisfactory procedure that requires less computational effort, Vollmann, Nugent, and Zartler have devised a procedure referred to as VNZ. The VNZ procedure consists of two phases.

Given an initial assignment  $a$ , along with the matrices  $D$  and  $W$ , phase 1

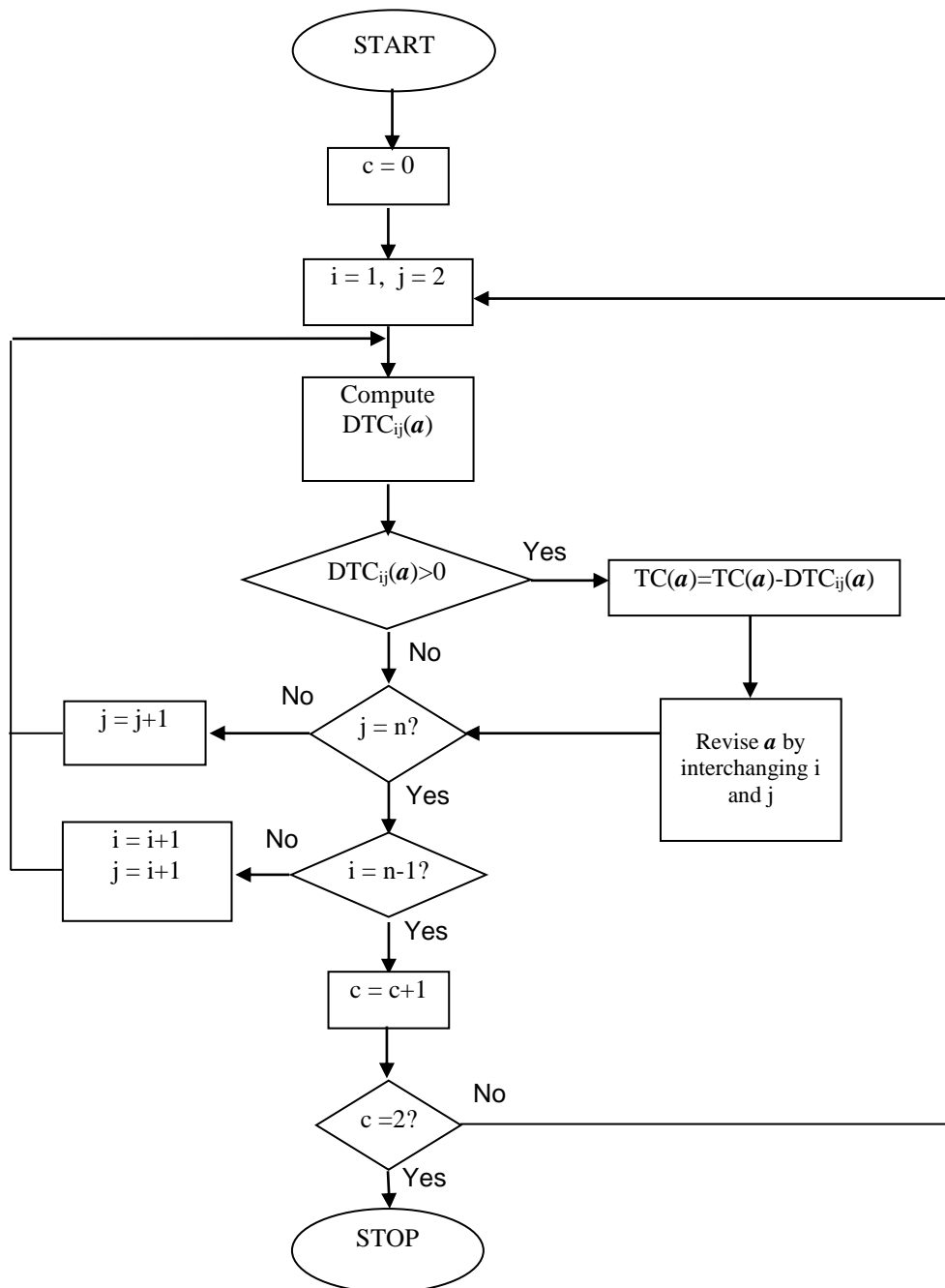
(Figure-2) chooses only two facilities, and concentrates on interchanging the locations of these two facilities with others. The choice of the two facilities is motivated by the fact that each



has a high total cost, which suggests that interchanging these two facilities with others will lead to a greater reduction in total cost than that obtained by most other choices of two facilities. The procedure of phase 1 repeats itself until two facilities (M1) and (M2) are found such that neither can be interchanged with any facility on their respective lists and the total cost will decrease. At this point, phase 1 stops.

Phase 2 (Figure-3) of the procedure is essentially a double check on phase 1. In phase 2 all pair-wise interchange of facilities are checked twice, and interchanges are made when the total cost is reduced. Since all facilities are considered when making interchanges in phase 2, it is possible that facilities may be interchanged during phase 2 that were not considered for an interchange during phase 1. All pair-wise interchanges are checked twice simply to prevent the computational effort in phase 2 from becoming excessive. [Frances, White, 1974]





**Figure-3 Phase two of the VNZ Procedure**

The total cost (Eq.4) for an assignment can be programmed

as:-

```
int cost()
{int i,j,sum;
sum=0;
for(i=0;i<n;i++)
for(j=0;j<n;j++)
sum+=(w[i][j]*d[a[i]][a[j]]);
sum*=.5;
return sum;
}
```

The DTC function (Eq.6) is programmed as:-

```
int dtc(int u1,int v1)
{int i,wd,dd,sum;
sum=0
for(i=0;i<n;i++)
{wd=w[i][u1]-w[i][v1];
dd=d[a[i]][a[u1]]-d[a[i]][a[v1]];
sum+=wd*dd;
}
sum-=(2*w[u1][v1]*d[a[u1]][a[v1]]);
return sum;
}
```

And the core process of VNZ which is the computation of  $P_k$  from (Eq.5) and is programmed as:-

```
int pk(int k)
{int s,j;
s=0;
for(j=0;j<n;j++)
s+=w[k][j]*d[a[k]][a[j]];
return s;
}
```

**4- A New Approach for the Solution: -**

A new heuristic procedure is developed here using the idea of constructing and building the solution in a step-by-step fashion, in contrast to the solution modification process of VNZ procedure, which necessitates an initial solution to start the process of improving such a solution. An initial solution can have a great impact on the resulting final solution and the efficiency of the employed rules.

Gupta's heuristic procedure [Gupta, 1971] developed originally for solving Flow Shop assignment problems (FSP) is modified here according to the adjustments imposed by the structure of quadratic assignment problems (QAP), and phase 1 of VNZ procedure is replaced with the modified heuristic rule to be correctly and appropriately applied to QAP.

Phase 1 of VNZ, relies heavily on the computation of the  $P_k$  function which represents the total cost for each facility  $k$  needed to determine M1 and M2 (the two facilities with the highest total cost), modifying phase 1 will require the elimination of that particular function and employing the improved rule to build the solution..

Adjustments of Gupta's heuristic rule involve pointing out the significant differences between the formulation of QAPs and FSPs. Observing that a FSP has only one asymmetrical matrix for determining the cost of an assignment, the situation is completely different for a QAP, as it involves two symmetric matrixes W and D.

Gupta's heuristic rule for FSP is given by:

$$f(i) = \frac{A}{\min(t_{i,m} + t_{i,m+1})}, \quad 1 \leq m \leq (M - 1) \dots\dots\dots (7)$$

Where A=1 if  $t_{i,M} \leq t_{i,1}$   
A= -1 otherwise

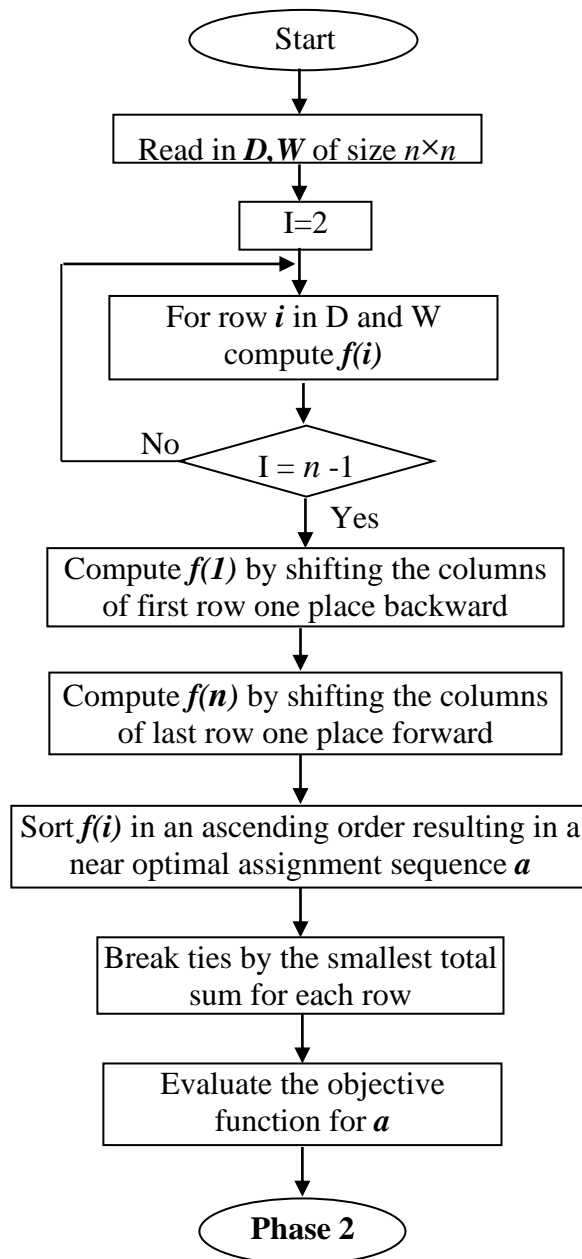
$t_{ij}$  is the time needed for processing job  $i$  on machine  $j$   
 $M$  is the number of machines

For quadratic assignment problems,  $M$  can be taken as  $n$  the number of facilities to be assigned, and  $t_{ij}$  is equivalent to multiplying the distance between facility  $i$  and facility  $j$  by the corresponding weight value with special treatment of the zero diagonal. This is done by isolating row 1 and row  $n$  from the general case of the rule, and shifting  $j$  one place backward for row 1 and forward for row  $n$ .

$$f(i) = \frac{A}{\min(W_{i,j}D_{i,j} + D_{i,j+1}W_{i,j+1})}, \quad 1 \leq j \leq (n-1) \quad \dots\dots\dots (8)$$

The facilities are then arranged in an ascending order of the vector  $f$  obtained after applying the new rule, if the application of the rule results in more than one element of  $f$  having equal values, then the ordering will be based on choosing the facility with the smallest sum of  $W_{ij}D_{ij}$  before others. In the end, an assignment is produced that gives a better near-optimal total cost solution; the algorithm is shown in Figure-4.

This rule substitutes the process of computing the  $P_k$  function and make no significance for the sorting process, the new rule is a much more direct forward procedure that needs less computational effort and computer storage. Efficiency of the obtained solutions is tested and discussed in the following section.



**Figure–4 The New Modified Procedure**

## **5- Results and comparisons:**

Three tests were carried out in this work on different computers for validating the efficiency of the present procedure, 26 different sizes of randomly generated problems were used in this investigation:-

### **Test 1:-**

Twenty problems of different small sizes were applied and tested by the VNZ procedure and the present procedure, and were then verified by the exact solution obtained by total enumerative procedures. Tests were performed on a Pentium III (1GHz) computer.

A distinctive improvement was observed from the obtained results. Table-1 shows the cost of assigning  $n$  facilities to  $n$  sites with matrix  $D$  and  $W$  being randomly generated, the cost of this assignment was computed using VNZ procedure and the present heuristic procedure.

The present procedure was capable of improving absolutely all the cases considered, and as can be noticed, some improved cases converged to the exact solution pointed out by total enumeration, these are shown shaded to be distinguished from other situations. The computational effort measured by time is very small (less than 0.04) for both heuristics, so it is not included in this table.



Table-1 A Comparison between different procedures according to solution efficiency

Problem		VNZ	Improved Sol.	Exact Sol.
No	Size ( $n$ )	Cost	Cost	Cost
1	6×6	231	224	224
2		219	215	213
3		345	345	345
4		144	144	144
5	7×7	249	249	244
6		200	195	195
7		295	272	272
8		294	275	274
9	8×8	379	362	362
10		499	492	488
11		419	419	414
12		432	432	432
13	9×9	451	434	418
14		376	376	356
15		487	467	467
16		462	458	437
17	10×10	718	715	703
18		667	664	618
19		742	721	711
20		810	780	755

**Test 2:-**

Four problems of larger sizes were tested on a Pentium III (1GHz) computer to point out the differences based on computational effort; the two procedures still have no considerable computation time, both were less than 0.04 since the computer cannot measure such small time (18.2 ticks/sec). But there is a huge difference in the time needed for computing optimal solutions using total enumeration as indicated by Table-2. Extracting the most outer loops and then multiplying the results by the upper bound of the extracted loops, gave the

estimated enumeration time. The explosion shown in Table-2 has overcome the expected practical expectations and made the use of such procedures infeasible.

Table-2 Execution time for large problems using total enumeration

Problem		VNZ Sol.		New Sol.		Enumeration
No.	Size	Min Cost	Time	Min Cost	Time	Estimated time
1	12 × 12	2007	<0.04	1978	<0.04	32 minutes
2	13 × 13	1117	<0.04	1102	<0.04	8 Hours
3	14 × 14	1430	<0.04	1424	<0.04	128 Hours
4	15 × 15	1547	<0.04	1536	<0.04	2000 Hours

**Test 3:-**

Both heuristics were investigated using the two very large problems on a slower and less powerful computer (Pentium I 100MHz) to indicate the difference in execution time required. Results in Table-3 show that the improved procedure has produced better solutions with an increasing amount of reduction in execution time.

Table-3 Execution time for very large problems on a slower computer

Problem		VNZ Sol.		New Sol.	
No.	Size	Min Cost	Time	Min Cost	Time
1	100 × 100	291129	1.00 sec	291067	0.80 sec
2	125 × 125	457119	2.30 sec	457090	1.42 sec

**6- Conclusion: -**

This work draws an important conclusion, which can be issued by the application of a new heuristic procedure built carefully by observing the solution's near optimality. Empirical testing has found that the heuristic presented here provides better performance in comparison to a well known efficient heuristic procedure. This better performance is stated in terms of less computational effort, production of more reliable solutions (as verified by exact solutions gained by total enumeration), in addition to the reduction in the execution time required to solve a given problem.

This heuristic procedure can be applied to a wide number of assignment problems that have a similar mathematical formulation, and can further be extended to apply to other types of problems depending on the mathematical formulation of such problems and the constraints they are subjected to, thus solving large-size problems more quickly and efficiently.

**REFERENCES**

- [1] Ahuja, R.K., Orlin, J.B. ,Tiwari, A.,**“A Greedy Genetic Algorithm for the Quadratic Assignment Problem,”**1997, site:web.mit.edu/jorlin/www/papersfolder
- [2] Daellenbach, H.G., **“Introduction to Operation Research,”** 1978, Allyn and Bacon, Inc.
- [3] Foulds, L.R., **“Combinatorial Optimization for Undergraduates,”** Springer-Verlag, New York, 1984.  
Frances R., White, J., **“Facility Layout and Location,”** Prentice-hall, 1974.
- [4] Gupta, J.N.D., **“A Functional Heuristic Algorithm for the Flow Shop Scheduling Problem,”** “ Operational Research Quarterly, 22 (1971) pp. 39-47.
- [5] Jean-Louis Lauriere, **Problem Solving and Artificial Intelligence,** Prentice Hall 1990.
- [6] Khaopravetch, K., Nanda, R., **“Assessing Solution Efficiency For Quadratic Assignment Problems,”** Microcomputers, April 1990.
- [7] Nanda R., Khaopravetch, K., **“An Interactive Decisional Algorithm for Facility Planning,”** AIIE International Engineering Conference Proceedings, 1987, PP. 272-277.
- [8] Nanda, R., Weingarten, A., **“A Practical Solution for the Quadratic Assignment Problem,”** “ AIIE Annual Conference Proceedings, 1974, v.6, PP.114-123.