SIMULATED ANNEALING ALGORITHM FOR SOLVING FACILITY LAYOUT PROBLEM

***Salih Eltahir Elmarioud Ali*[1], *Elamin Hussein*[2]**

[1] ***Faculty of Engineering &Technology, Nile Valley University(Atbara).***
[2] ***Faculty of Engineering, Karary University (Khartoum).***

Received December 2010, accepted after revision June 2011

مُستَخْلَص

مشكلة تخطيط وسائل الإنتاج توجد في عدة أنواع من نظم التصنيع. وعادة ما ترتبط مشاكل تخطيط وسائل الإنتاج بمواقع وسائل الإنتاج كالآلات و الأقسام الإنتاجية في المصنع. من المعروف أن مشكلة تخطيط وسائل الإنتاج تؤثر بشكل كبير على أداء النظام في محل العمل يتم تحديد العلاقات الأساسية المتبادلة بين العمليات حول ما إذا كان أو لم تكن تتطلب التقارب أو التجاور. و المقياس المستخدم إيجاد الحد الأدنى من تكلفة حركة البند ، حيث يتم التعبير عن هذه التكاليف بوصفها وظيفة خطية بالنسبة للمسافة المقطوعة. وتتعلق هذه الورقة بدراسة أسباب التقارب أو الانفصال مع الأخذ في الاعتبار حركة المواد بين العمليات. نظراً للعدد الكبير من العمليات الحسابية والترتيبات اللازمة من أجل حل مشكلة تخطيط وسائل الإنتاج ، هناك صعوبة في التعامل معها يدويا. أفضل نهج هو استخدام الكمبيوتر كأداة لتحليل مشاكل تخطيط وسائل الإنتاج. على أساس تخطيط أولي(ابتدائي). تهدف هذه الورقة إلي إيجاد تخطيط كتلة أمثل لعرض المواقع النسبية للأقسام . تم استخدام الطريقة الزوجية لإجراء تبادل لتوليد الحلول البديلة عن طريق اعتماد : خوارزمية محاكاة إجرائية التلدين بالتحمية ثم التبريد (الانصهار الزائف). تم تصميم وتطوير خوارزمية تستند على الانصهار الزائف لحل مشكلة تخطيط وسائل الإنتاج للأقسام ذات المساحات المتساوية وتم تطبيق الخوارزمية على بعض المسائل القياسية وأعطت نتائج جيدة.

ABSTRACT

Layout problems are found in several types of manufacturing systems. Typically, layout problems are related to the location of facilities (e.g., machines, departments) in a plant. They are known to greatly impact the system performance. In job shop primary interrelations between operations are identified as to whether or not they require closeness. The criterion employed is concerned with minimization of the cost of item movement, where this cost is expressed as a linear function of distance traveled. This paper studies the reasons for closeness or separation that needed with consideration of materials movement between operations. Due to large number of calculations and arrangements needed for solving facility layout problems, the computer is used as an aided tool for analyzing layout problems. Based on an *initial layout* the objective is to produce an *optimal block layout* showing the relative positioning of departments. Pair wise exchange procedure is used to generate alternative solutions via adopting *simulated annealing heuristic*. Simulated annealing based software system for solving equal area static facility layout problem is developed. The developed algorithm is tested by using some standard problems in the literature.

Keywords: *Facility Layout, Block Layout, Quadratic Assignment Problem, Simulated Annealing, Optimization.*

1. INTRODUCTION

To operate production and service systems efficiently, the systems not only have to be operated with optimal planning and operational policies, but also well designed. Optimal design of physical layout is an important issue in the early stage of the system design. The facility layout problem is the problem of designing a physical layout of departments with a certain objective such as minimizing the total material handling costs. Generally, layout design is done in two steps: design of a block layout and completion of details. In the first step, shapes and relative locations of departments are determined, while the second step specifies the location of primary equipment used in each department and incidental equipment such as gas and air lines, and lighting fixtures. Because the second step is usually quite system-dependent, researchers have concentrated on development of an efficient algorithm to produce a good block layout. Block layout is usually a precursor to these subsequent design steps, termed “detailed layout”.

A facility layout is an arrangement of everything needed for production of goods or delivery of services. A facility is an entity that facilitates the performance of any job. It may be a machine tool, a work centre, a manufacturing cell, a machine shop, a department, a warehouse, etc. [1].

2. BLOCK LAYOUT

A typical approach to the facility layout problem is to combine tasks or equipment into functional groups, or blocks. Once knowledge of the materials flow, process details, and support activities is known, it is possible to locate different blocks on the layout based on their relationships with each other. Specifying the relative location and size of each department within a facility, this common representation of solutions to the facility layout problem is referred to as the block layout . Block layouts are used to provide preliminary information to architects and engineers involved in the construction of a new facility. The block layout is typically represented in either a discrete or continuous fashion. A discrete representation of the block layout uses a collection of grids to represent departments . However, a continuous representation uses the centroid, area, perimeter, width and/or length of a department to specify the exact location of the department within a facility layout. In the literature, most of the facility layout algorithms use a discrete representation to generate the block layout.[2].

3. FORMULATION OF FACILITY LAYOUT PROBLEMS

A number of formulations have been developed for the facility layout problem. When the real shapes and sizes of the facilities are disregarded, the facility layout problem is generally formulated as a quadratic assignment problem (QAP) of allocating equal area facilities to discrete locations on a grid with the objective of minimizing a given cost function. The QAP model is used to assign facilities (departments) to locations such that the distance materials travel is minimized. The number of departments to be located and the number of locations are equal, and all the locations are of equal size. The location site diagram of a 6-department problem is given in Figure 1. The distance between any two adjacent sites is one distance unit, and distance is measured from the centroid of one department to the centroid of another using the rectilinear distance measure.

4. CLASSIFICATION OF FACILITY LAYOUT PROBLEMS

The flow data used for determining the layout classifies the layout problem into two categories: static and dynamic. If the flow data between the departments does not change over time, then the problem is defined as the **S**tatic **F**acility **L**ayout **P**roblem (SFLP). [3]& [4].

When the flow changes over time, then the problem is defined as the **D**ynamic **F**acility **L**ayout **P**roblem (DFLP). [5]. Furthermore, the nature of the flow data can be characterized as deterministic or probabilistic. Deterministic flow data is fixed and known with certainty. That is, during the planning horizon, **material** flow between the departments is known with certainty. When the flow data are not known with certainty, they can be represented as random variables. That is, the behavior of the flow data can be approximated by a probability distribution. In other words, the flow data are said to be probabilistic.[6].

5. ALGORITHMS CLASSIFICATION

Layout algorithms can also classified according to their objective functions. There are two objectives, one aims at *minimizing* the sum of *flows* times *distance*, while other aims at *maximizing* an adjacency score.

Generally, the former, that is the distance based objective, which is similar to the classical quadratic assignment problem

1	2	3
4	5	6

Figure 1: Location Site diagram.

(QAP), is more suitable when the input data is expressed as a from-to-chart [is used for measuring Flow quantitatively in terms of amount of movements between departments]. And the later, that is, the adjacency – based objective, is more suitable for a relationship chart[Flows measured qualitatively using the closeness relationships values conjunction with reasons for the closeness value].

Consider first the distance based objective. Let m denote the number of departments, $\mathbf{f_{ij}}$ denote the flow from department i to department j, expressed in number of unit loads moved per unit time, and $\mathbf{c_{ij}}$ denote the cost of moving a unit load one distance unit from department i to department j. The objective is to minimize the cost per unit time for movement among the departments. Expressed mathematically, the objective can be written as :

$$\min \ z = \sum_{i=1}^{m} \sum_{j=1}^{m} f_{ij} c_{ij} d_{ij} \qquad (1)$$

Where d_{ij} is the distance from department i to department j. in many layout algorithms d_{ij} is measured rectilinearly between department centroids.

c_{ij} Values in equation (1) are assumed to be independent of the utilization of the handling equipment, and they are linearly related to the length of the move. In those cases where c_{ij} values do not satisfy the above assumptions, set c_{ij} =1 for all i and j and focus only on total unit load travel in the facility, that is, the product of f_{ij} and d_{ij} values.

Now considering the adjacency-based objective where the adjacency score is computed as sum of all flow values (relationship values) between those departments that are adjacent in the layout. let $x_{ij} = 1$ if departments i and j are adjacent(that is they are share boarder) in the layout, and zero otherwise, the objective is to maximize the adjacency score; that is [7]:

$$\max \ z = \sum_{i=1}^{m} \sum_{j=1}^{m} f_{ij} x_{ij} \qquad (2)$$

6. SIMULATED ANNEALING

Simulated annealing (SA) is a random-search technique which exploits an analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a more general system; it forms the basis of an optimization technique for combinatorial (Finding the minimum of a given function depending on many variables) and other problems.

Analogy: If a liquid material cools and anneals too quickly, then the material will solidify into a sub-optimal configuration. If the liquid material cools slowly, the crystals within the material will solidify optimally into a state of minimum energy (i.e. ground state). This ground state corresponds to the minimum of the cost function in an optimization problem.

One of the primary strengths of SA is that, while trying to improve a layout, it may accept non-improving solutions to allow the algorithm to explore other regions of the solution space (instead of stopping at the first seemingly good solution. In fact a SA-based procedure may accept non-improving solutions several times during the search in order to push the algorithm out of solution which may be only locally optimal. As result the objective function value (OFV) may actually increase more than once. The amount of increase in (OFV) that the algorithm will tolerate is carefully controlled through the search.

The concept of occasionally accepting *non-improving solutions* is as follows: given a current arrangement of atoms or "elements", randomly making incremental changes to current arrangement in order to obtain a new arrangement, to measure decrease in *energy*, say ΔE. If $\Delta E > 0$ (i.e. energy decrease), the new arrangements is accepted as the current one and use it to make subsequent changes. However, if $\Delta E \leq 0$, the new arrangement is accepted with probability,

$$P(\Delta E) = \exp(\Delta E / K_b T) \qquad (3)$$

Where T is temperature and K_b is a constant. To apply the above procedure to *optimization problems*, treat:

a. The current arrangement as the current solution.

b. The *new* arrangement as the *candidate* solution.

c. The *energy* as the *objective function* value. A random incremental change is made for example, by changing two randomly picked departments from the current layout.

d. Setting $K_b = 1$, since it has no known significance in optimization problems.

It should be observed that, as the increase in the objective function gets larger, the probability of accepting candidate solution gets smaller (see Equation (3)). Also it should be noted that, the probability of accepting a non-improving solution decreases as the temperature decreases (cooling), so that, it is more likely to accept non-improving solutions early in the annealing process, due to the relatively high temperatures.

It is concluded that the probability depends on the change in the objective function relative to temperature.

So that, setting an initial temperature value, and how fast to 'cool' the system, are two very important issues in designing SA-based algorithm. Concerning the initial temperature setting, one possible approach is to set it according to (OFV) of starting solution; other approaches consider high constant value.

The above process of generating candidate solutions and updating the current solution continuous until the system reaches steady state or equilibrium (the condition in which a further improvement in the solution using additional interchanges is highly unlikely to occur) at the current temperature. One equilibrium is reached the temperature is reduced according to a predetermined temperature reduction factor (a fraction number between zero and one), then continue to generate and evaluate candidate solutions with the new temperature setting.

The search is terminated either when a user –specified final temperature is reached (which can be expressed as the maximum number of temperature reductions be considered) or a user defined number of successive temperature reductions does not produce an improvement in the current best solutions.

7. The Developed Simulated Annealing Algorithm

The algorithm begins with setting the initial solution S and a corresponding objective function value z, which is calculated by using equation (1) concentrating on the product of flow (f_{ij}) and distance (d_{ij}) and neglecting (c_{ij}) values . Two candidate departments in the sequence of the initial configuration in the initial layout sequence are sequentially selected for swapping; the potential pair-swaps are examined in the order:

(1, 2), (1, 3) . . . . . (1, n), (2, 3) . . . . . (n- 1, n), (1, 2), ..., and the change in the objective function value is compared with the initial value z. If the change in cost is a reduction, then the swap is made and the configuration is updated. If the cost is increased, then the configuration change is accepted only if it meets the acceptance criteria described in flow chart presented in Figure 2.

In this algorithm the temperature is controlled by the criterion taken from [8] scheme were the temperature drops after each attempted pair-swop, from a specified starting temperature T_i to a specified final temperature T_f by the recurrence (iterated) relation:

$$T_{i+1} = \frac{T_i}{(1+\alpha T_i)} \quad where\, \alpha < T \ (4)_o$$

Where α is a cooling rate.

This cooling scheme could be completed in a specified number of steps (S). This is achieved by setting:

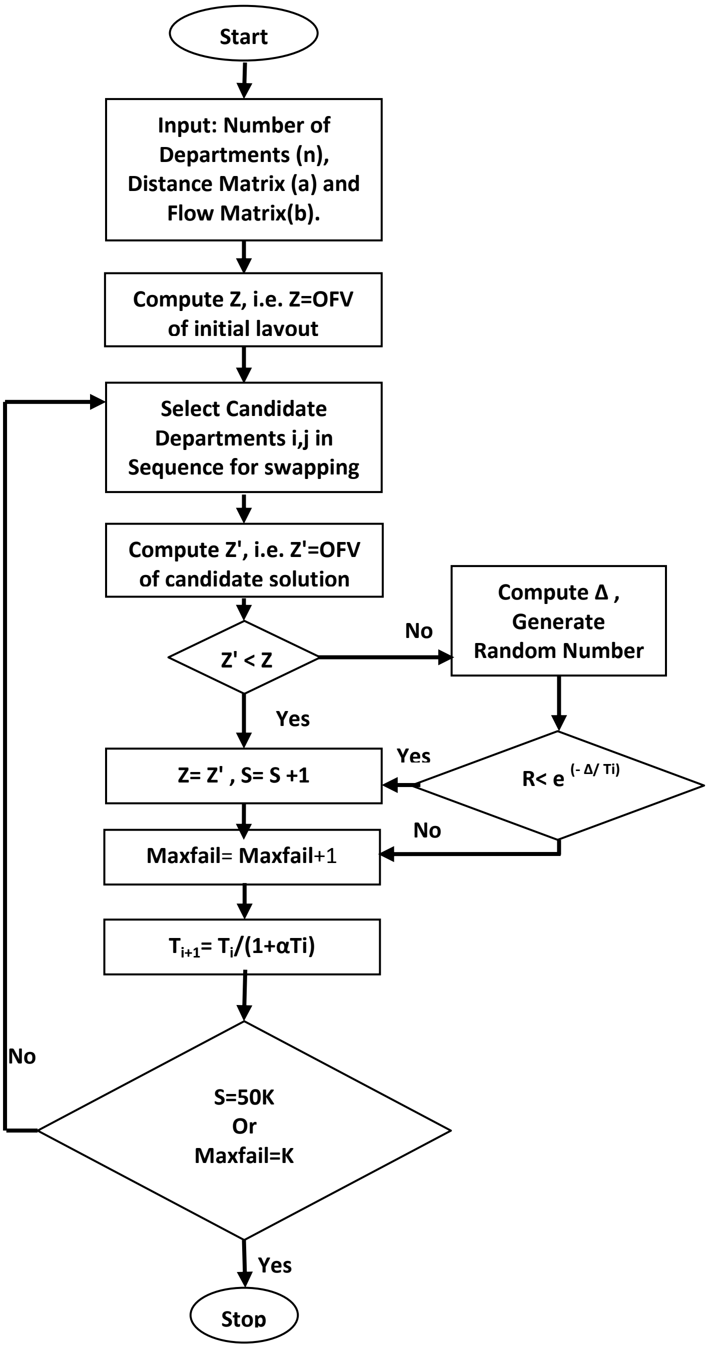

Figure 2: The developed Simulated Annealing

$$\alpha = \frac{T_i - T_f}{S T_o T_f} \qquad (5)$$

Where S, the number of swops examined, equal to 50K, K represents the size of the neighbourhood structure. (Where K = ½n (n - 1).

S/100 random swops are used to determine Δmax and Δmin and T_0, T_f are given the values:

$$T_0 = \Delta_{min} + {(\Delta_{max} - \Delta_{min})}/{10} \qquad (6)$$

$$T_f = \Delta_{min}$$

If Consecutive Uphill Steps (CUS=size of neighbourhood=½n (n - 1)) are rejected (maximum number of fails), then:

(i) the next uphill is accepted,
(ii) Cooling is stopped by setting α = 0.

8. PROBLEMS DATA

The algorithm should be tested on four problems of different sizes: 12, 15, 20, and 30-departments, the problems are taken from Nugent *test problems* [9], Wherever possible the plant shapes are rectangular: 3 by 4 (i.e., 3 rows by 4 columns), 3 by 5, 4 by 5, and 5 by 6 for the 12, 15, 20, and 30-department problems respectively. All departments for all problems are square and of equal area, and the distances between departments are measured rectangularly. The problems are shown in figure 3. The numbers are given in the order of locations 1, 2… n. Location numbers follow the conventional department sequence of left to right, top to bottom. The flow and distance data for the test problems are presented in the form of charts shown in Tables 2 to 5.

Ten trials were performed on each of these problems with **S**, the number of swops examined, and equal to 50K. The department numbers given indicate the assignment of departments to locations.

The flow and distance data are assumed symmetrical (Fij= Fji, Dij= Dji, for all i, J) for all problems the data are compactly presented in Table1

1	2	3	4
5	6	7	8
9	10	11	12

a: Twelve-Department Plant

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

b: Fifteen-Department Plant

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

c: Twenty-Department Plant

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

d: Thirty-Department Plant

Figure 3: Plant Shapes of the Test Problems

Table 1: The input form of flow and distance data.

	1	**2**	**3**	**- -**	**n**
1	-	D12	D13	- -	D1n
2	F21	-	D23	- -	D2n
3	F31	F32			
-					
n	Fn1	Fn2			

Table 2: Flow and Distance Data for n=12

	1	2	3	4	5	6	7	8	9	10	11	12
1	-	1	2	3	1	2	3	4	2	3	4	5
2	5	-	1	2	2	1	2	3	3	2	3	4
3	2	3	-	1	3	2	1	2	4	3	2	3
4	4	0	0	-	4	3	2	1	5	4	3	2
5	1	2	0	5	-	1	2	3	1	2	3	4
6	0	2	0	2	10	-	1	2	2	1	2	3
7	0	2	0	2	0	5	-	1	3	2	1	2
8	6	0	5	10	0	1	10	-	4	3	2	1
9	2	4	5	0	0	1	5	0	-	1	2	3
10	1	5	2	0	5	5	2	0	0	-	1	2
11	1	0	2	5	1	4	3	5	10	5	-	1
12	1	0	2	5	1	0	3	0	10	0	2	-

Table 3: Flow and Distance Data for n=15

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	-	1	2	3	4	1	2	3	4	5	2	3	4	5	6
2	10	-	1	2	3	2	1	2	3	4	3	2	3	4	5
3	0	1	-	1	2	3	2	1	2	3	4	3	2	3	4
4	5	3	10	-	1	4	3	2	1	2	5	4	3	2	3
5	1	2	2	1	-	5	4	3	2	1	6	5	4	3	2
6	0	2	0	1	3	-	1	2	3	4	1	2	3	4	5
7	1	2	2	5	5	2	-	1	2	3	2	1	2	3	4
8	2	3	5	0	5	2	6	-	1	2	3	2	1	2	3
9	2	2	4	0	5	1	0	5	-	1	4	3	2	1	2
10	2	0	5	2	1	5	1	2	0	-	5	4	3	2	1
11	2	2	2	1	0	0	5	10	10	0	-	1	2	3	4
12	0	0	2	0	3	0	5	0	5	4	5	-	1	2	3
13	4	10	5	2	0	2	5	5	10	0	0	3	-	1	2
14	0	5	5	5	5	5	1	0	0	0	5	3	10	-	1
15	0	0	5	0	5	10	0	0	2	5	0	0	2	4	-

Table 4: Flow and Distance Data for n=20

.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
1	-	1	2	3	4	1	2	3	4	5	2	3	4	5	6	3	4	5	6	7
2	0	-	1	2	3	2	1	2	3	4	3	2	3	4	5	4	3	4	5	6
3	5	3	-	1	2	3	2	1	2	3	4	3	2	3	4	5	4	3	4	5
4	0	10	2	-	1	4	3	2	1	2	5	4	3	2	3	6	5	4	3	4
5	5	5	0	1	-	5	4	3	2	1	6	5	4	3	2	7	6	5	4	3
6	2	1	5	0	5	-	1	2	3	4	1	2	3	4	5	2	3	4	5	6
7	10	5	2	5	6	5	-	1	2	3	2	1	2	3	4	3	2	3	4	5
8	3	1	4	2	5	2	0	-	1	2	3	2	1	2	3	4	3	2	3	4
9	1	2	4	1	2	1	0	1	-	1	4	3	2	1	2	5	4	3	2	3
10	5	4	5	0	5	6	0	1	2	-	5	4	3	2	1	6	5	4	3	2
11	5	2	0	10	2	0	5	10	0	5	-	1	2	3	4	1	2	3	4	5
12	5	5	0	2	0	0	10	10	3	5	5	-	1	2	3	2	1	2	3	4
13	0	0	0	2	5	10	2	2	5	0	2	2	-	1	2	3	2	1	2	3
14	0	10	5	0	1	0	2	0	5	5	5	10	2	-	1	4	3	2	1	2
15	5	10	1	2	1	2	5	10	0	1	1	5	2	5	-	5	4	3	2	1
16	4	3	0	1	1	0	1	2	5	0	10	0	1	5	3	-	1	2	3	4
17	4	0	0		5	1	2	5	0	0	0	1	0	1	0	0	-	1	2	3
18	0	5	5	2	2	0	1	2	0	5	2	1	0	5	5	0	5	-	1	2
19	0	10	0	5	5	1	0	2	0	5	2	2	0	5	10	2	2	1	-	1
20	1	5	0	5	1	5	10	10	2	2	5	5	5	0	10	0	0	1	6	-

Table 5: Flow and Distance Data for n=30

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	-	1	2	3	4	5	1	2	3	4	5	6	2	3	4	5	6	7	3	4	5	6	7	8	4	5	6	7	8	9
2	3	-	1	2	3	4	2	1	2	3	4	5	3	2	3	4	5	6	4	3	4	5	6	7	5	4	5	6	7	8
3	2	4	-	1	2	3	3	2	1	2	3	4	4	3	2	3	4	5	5	4	3	4	5	6	6	5	4	5	6	7
4	0	0	3	-	1	2	4	3	2	1	2	3	5	4	3	2	3	4	6	5	4	3	4	5	7	6	5	4	5	6
5	0	10	4	0	-	1	5	4	3	2	1	2	6	5	4	3	2	3	7	6	5	4	3	4	8	7	6	5	4	5
6	2	4	0	0	5	-	6	5	4	3	2	1	7	6	5	4	3	2	8	7	6	5	4	3	9	8	7	6	5	4
7	10	0	5	0	2	1	-	1	2	3	4	5	1	2	3	4	5	6	2	3	4	5	6	7	3	4	5	5	7	8
8	5	0	5	2	0	2	10	-	1	2	3	4	2	1	2	3	4	5	3	2	3	4	5	6	4	3	4	4	6	7
9	0	2	5	2	0	2	10	1	-	1	2	3	3	2	1	2	3	4	4	3	2	3	4	5	5	4	3	3	5	6
10	5	2	1	0	0	1	5	3	10	-	1	2	4	3	2	1	2	3	5	4	3	2	3	4	6	5	4	4	4	5
11	2	1	4	6	0	4	10	5	2	5	-	1	5	4	3	2	1	2	6	5	4	3	2	3	7	6	5	5	3	4
12	5	0	1	0	2	10	10	0	1	5	0	-	6	5	4	3	2	1	7	6	5	4	3	2	8	7	6	5	4	3
13	0	0	0	2	0	10	6	0	5	6	0	5	-	1	2	3	4	5	1	2	3	4	5	6	2	3	4	5	6	7
14	0	0	4	5	0	2	0	0	2	0	1	5	2	-	1	2	3	4	2	1	2	3	4	5	3	2	3	4	5	6
15	2	2	0	2	0	5	0	2	0	1	2	2	0	2	-	1	2	3	3	2	1	2	3	4	4	3	2	3	4	5
16	0	0	4	5	0	5	10	4	3	5	1	0	4	1	4	-	1	2	4	3	2	1	2	3	5	4	3	2	3	4
17	5	0	0	1	2	0	2	5	0	5	0	0	2	0	5	0	-	1	5	4	3	2	1	2	6	5	4	3	2	3
18	6	2	6	1	1	5	1	2	2	0	2	0	2	5	1	3	2	-	6	5	4	3	2	1	7	6	5	4	3	2
19	3	0	3	1	0	0	10	10	0	5	0	0	1	3	0	0	2	5	-	1	2	3	4	5	1	2	3	4	5	6
20	0	1	2	1	0	0	1	6	0	2	0	2	0	10	1	2	0	1	0	-	1	2	3	4	2	1	2	3	4	5
21	1	6	5	2	2	0	5	0	4	3	0	0	6	0	0	2	0	2	5	5	-	1	2	3	3	2	1	2	3	4
22	10	1	5	2	0	10	5	5	0	5	6	4	2	0	5	0	0	10	5	2	4	-	1	2	4	3	2	1	2	3
23	0	0	2	4	5	0	2	5	5	0	6	5	1	4	0	2	6	10	1	1	0	5	-	1	5	4	3	2	1	2
24	10	1	1	0	1	0	3	2	2	5	0	10	5	2	2	0	5	4	0	3	1	0	0	-	6	5	4	3	2	1
25	2	2	0	2	0	0	5	5	0	2	4	1	5	0	0	5	3	0	5	1	0	4	4	5	-	1	2	3	4	5
26	1	2	0	0	2	4	0	0	5	10	5	0	0	0	0	0	5	0	2	5	0	4	4	5	1	-	1	2	3	4
27	1	5	3	2	1	0	2	5	2	10	3	0	0	4	5	5	0	5	1	6	0	5	1	0	0	0	-	1	2	3
28	1	1	1	2	0	10	0	5	2	1	2	0	1	2	1	2	0	0	2	5	5	0	0	1	10	0	0	-	1	2
29	0	10	0	5	2	1	1	0	5	5	2	0	5	5	1	5	5	0	10	5	0	2	2	0	1	0	0	2	-	1
30	1	5	2	5	1	1	3	2	2	2	10	1	5	5	0	10	1	0	10	3	0	5	2	0	0	0	10	2	2	-

9. PROBLEMS SOLUTIONS

Problem 1:

Table 6: n=12, number of iterations (swops examined) per restart =3300.

Restart (run)	Best	Current
1	578	578 *
2	578	586
3	578	600
4	578	592
5	578	586
6	578	586
7	578	582
8	578	586
9	578	586
10	578	586

Best solution value found at iteration number 1295,Best solution value=578 ,Best permutation: 2 10 6 5 1 11 8 4 3 9 7 12.

Problem 2:

Table 7: n=15, number of iterations (swops examined) per restart =5250.

Restart	Best	Current
1	1166	1166
2	1166	1170
3	1152	1152
4	1152	1180
5	1150	1150 *
6	1150	1174
7	1150	1160
8	1150	1152
9	1150	1152
10	1150	1150 **

Best solution value=1150

Best permutation: 12 5 6 15 10 11 7 14 3 4 9 8 13 2 1

* the best value found at iteration number 945.

** the best value found at iteration number 1542.

Problem 3:

Table 8: n=20, of number of iterations (swops examined) per restart=9500

Restart	Best	Current
1	2632	2632
2	2628	2628
3	2580	2580
4	2570	2570*
5	2570	2632
6	2570	2604
7	2570	2584
8	2570	2596
9	2570	2622
10	2570	2620

Best value found at iteration number 2654,Best solution value=2570,Best permutation found: 17 5 7 1 6 19 15 20 8 13 4 2 12 11 16 18 14 10 3 9.

Problem 4:

Table 9: n=30, number of iterations (swops examined) per restart=21750

Restart	Best	Current
1	6196	6196
2	6194	6194
3	6178	6178
4	6174	6174
5	6174	6194
6	6174	6224
7	6172	6172
8	6170	6170*
9	6170	6222
10	6170	6174

Best value found at iteration number 15035,Best solution value=6170,Best permutation found: 4 3 29 21 25 14 30 19 9 13 28 20 16 8 7 25 6 27 11 22 10 1 12 15 18 23 26 17 24.

10. ANALYSIS

The results of the four problems are compared with the best known values in the literature and other heuristics as shown in Table 10.

Table 10: The Developed Algorithm compared with best known values and other heuristics values.

(a)	n	(b)	(c)	(d)	(e)
1	12	578	612	***578***	600
2	15	1150	1182	***1150***	1226
3	20	2570	2620	***2570***	2676
4	30	6124*	6236*	***6170****	6390*

Where (a) is the column of the problem number, (n) stands for the problem size,(b) represents the best known values in the literature taken from [10], column (c) presents the results of Burkard Simulation Procedure known as (QAPH4 heuristic) were taken from [11],the results of column (e) are outcome of Monte-Carlo heuristic for the QAP and column(d) values are the results of the developed SA algorithm.

From the analysis presented above, It should be noted that THE DEVELOPED ALGORITHM coincide with the best known values in the literature, for the Biggest Nugent test problem (n=30) THE DEVELOPED ALGORITHM resulted in best solution of 6170 cost with 10 runs compared with 6124 as the best known solution found in literature, however THE DEVELOPED ALGORITHM with 100 runs approaching this value and resulted in 6128 cost value found at iteration 6276 in run number 98.

due to the large number of iterations performed at each run, THE DEVELOPED ALGORITHM gives best values compared with Burkard Simulation Procedure in which , the number of iterations at constant temperature is (2n) as default where[n is the number of departments], then during the annealing process the number of iterations is increased by ten percentage at each step by using multiplying factor.

Since Monte-Carlo heuristic is a local optimization based method searching for the best solutions in a downhill direction only (accepting only improving solutions) , it is not surprising that THE DEVELOPED ALGORITHM gives better values in comparing with.

11. **CONCLUSION**

Based on simulated annealing approach this study provides computerized tool to solve the static facility layout problem. The tool is a software, for solving departments of equal areas, that is direct implementation of quadratic assignment problems (QAP**)**. The software tested on 4 problems of different sizes: 12, 15, 20, and 30-departments, the results are compared with other known heuristics like QAPH4, Monte-Carlo Heuristic for QAP and best known values in the literature. The analysis reveals that the software gives good and acceptable results.

REFERENCES

1. Heragu, S. “Facilities Design”, PWS Publishing Company, Boston, 1996.
2. Sheth, V. S. “Facilities Planning and Materials Handling: Methods and Requirements.” New York: Marcel Dekker, Inc., 1995, pp. 67-86.
3. Kusiak, A. and Heragu, S.S., 1987, “The facility layout problem,” European Journal of Operational Research, Vol. 29, pp. 229-251.
4. Meller, R. D. and K.-Y. Gau, "The Facility Layout Problem: Recent and Emerging Trends and Perspectives," Journal of Manufacturing Systems, vol. 15, no. 5, pp.351-366, 1996.
5. Rosenblatt, M.J., 1986, “The dynamics of Plant Layout,” Management Science, Vol. 32, No. 1, pp. 76-86.
6. Kouvelils, P., Kurawarwala, A.A., and Gutierrez, G.J., 1992, “Algorithms for Single and Multiple Period Layout Planning for Manufacturing Systems,” European Journal of Operational Research, Vol. 63, pp. 287-303.
7. Tompkins, J. A. White, Y. A. Boozer, E. H. Frazelle, J. M. Tanchoco, and J.Trevino, Facilities Planning, 2nd ed. New York: John Wiley and Sons, 1996
8. Lundy, M., and Mees, A., "Convergence of an annealing algorithm", mathematical Programming 34 (1986) 111-124.
9. Nugent, C.E., T.E. Vollmann and J. Rural, An experimental comparison of techniques for the assignment of facilities to locations, J. Operations Res. 16 (1968) 150-173
10. Burkard, R.E. and T. Brnniger, A heuristic for quadratic Boolean programs with applications to quadratic assignment problems, European J. Operational Res. 13(4) (1983) 374-386.
11. Burkard, R.E., and Rendl, F., "A thermodynamically motivated simulation procedure for combinatorial optimization problems", European Journal of Operational Research 17 (1984) 169-174.
12. Misevicius A (2003) A modified simulated annealing algorithm for quadratic assignment problem. Informatica 14(4):497–514.