Int J Adv Manuf Technol (2013) 68:1665–1675
DOI 10.1007/s00170-013-4965-2

ORIGINAL ARTICLE

# A non-greedy systematic neighbourhood search heuristic for solving facility layout problem

**Rajesh Matai · S. P. Singh · M. L. Mittal**

Received: 15 June 2011 / Accepted: 1 April 2013 / Published online: 15 May 2013


**Abstract** This paper presents a new heuristic for solving a facility layout problem. The proposed heuristic works on the non-greedy systematic pairwise exchange of two facilities, that is 2-exchange neighbourhood search based on non-greedy strategy. The proposed heuristic is applied on a large number of test problems provided by different authors in the quadratic assignment problem (QAP) library (Burkard, Karisch, Rendl, J Glob Optim 10(1):391–403, 1997) with problem size ranging from 12 to 256. Out of the 135 test problems available in the QAP library, the proposed heuristic reached optimal solutions for 64 test problems and matched the best known available solution for the other 15 test problems. For the remaining 56 test problems, the proposed approach reports highly encouraging solutions for 44 test problems (within the 2 % of deviation from the optimal/best known solutions), and for the remaining 12 problems, the proposed approach provides fair solution in reasonable time. Comparison with other meta-heuristic approaches (Ro-TS, RE-TS, GEN and SA) shows the effectiveness of the proposed heuristic.

**Keywords** Facility layout problem · Quadratic assignment problem · Heuristic · Neighbourhood search

R. Matai (✉)
Department of Management, Birla Institute of Technology and Science, Pilani, India
e-mail: rajesh_matai73@yahoo.com

S. P. Singh
Department of Management Studies, Vishwakarma Bhawan, Indian Institute of Technology, Delhi, India

M. L. Mittal
Department of Mechanical Engineering, Malaviya National Institute of Technology, Jaipur, India

## 1 Introduction

Arrangement of facilities in both manufacturing and service environments plays an important role in delivering products and services in an efficient and effective manner. The facility layout problem (FLP) is of crucial importance to industrial engineers and a well researched problem in academics. FLP was first formulated as quadratic assignment problem (QAP) by Koopmans and Beckman [30]. Later, Sahni and Gonzalez [50] showed QAP is a NP-complete problem. To achieve global optimum for QAP branch and bound, cutting planes or combinations of these methods, like branch-and-cut and dynamic programming, are used. However, results by these exact algorithms are modest. Diponegoro and Sarker [14] reported that instances of the QAP of sizes larger than 20 cannot be solved optimally in a reasonable computational time. Therefore, the interest of researchers and practitioners lies in the application of heuristics and meta-heuristic approaches to solve QAP. Some of the well known heuristic approaches applied in the past and available in literature are CRAFT, HC-66, ALDEP, CORELAP, SABLE, etc. But the performance of these heuristics is good only for small- or moderate-sized problems. As the problem size increases, the solution quality decreases. In addition to applying heuristics, nowadays, meta-heuristic approaches like simulated annealing (SA), Tabu Search (TS), ant colony algorithm and genetic algorithm (GA) are also widely applied to solve FLP. But again, the computational time of these meta-heuristic-based approach increases as the problem size increases. A good amount of work on FLP (single and multiobjective) can be found in O'Brien and Abdel Barr [45], Tompkins and Reed [60], Kusiak and Heragu [32], Rosenblatt and Lee [49], Heragu [25], Heragu and Kusiak [24], Meller and Gau [39, 40], Kochhar and Heragu [31], Singh and Sharma [54], Singh and Singh [56, 57] and Matai et al. [37]. Here, an attempt is made to propose a novel heuristic approach for FLP modelled as QAP.

The paper proposes a systematic pairwise exchange heuristic based on non-greedy strategy for solving the FLP having flow as a single objective. The literature relevant to the various approaches for solving the FLP is reviewed in Section 2. The mathematical formulation of FLP modelled as QAP is given in Section 3. A non-greedy systematic neighbourhood search heuristic is described in Section 4. Procedural steps of the proposed heuristic are discussed in Section 5. Section 6 presents performance of the proposed heuristic followed by the conclusions and future research directions in Section 7.

## 2 Literature review

Papers that deal with the research work related to the solution approaches in connection with FLP are reviewed here. In literature, the solution techniques for solving FLP are classified as exact algorithms, heuristics and meta-heuristics. We briefly review the past work in the following three subsections.

### 2.1 Exact approaches

There are three main exact methods used to find the global optimal solution for QAP: dynamic programming, cutting plane techniques, and branch and bound procedures. Research has shown that the branch and bound is the most successful among exact algorithms for solving QAP. The first two branch and bound algorithms were developed by Gilmore [19] and Lawler [34]. In addition to these two algorithms, two other algorithms were developed by Land [33] and Gavett and Plyter [18]. Kaku and Thompson [28] proposed another branch and bound algorithm which performs better than Lawler's [34] algorithm, particularly for problems of bigger size. Bazaraa and Sherali [3] developed a cutting plane algorithm based on Bender's partitioning scheme. Burkard and Bonninger [8] also developed a cutting plane method to solve QAP. Dynamic programming is a technique used for QAP special cases where the flow matrix is the adjacency matrix of a tree. Christofides and Benavent [12] studied this case using a MILP approach to the relaxed problem. All the exact algorithms discussed in the literature have high memory and computational requirements [9].

### 2.2 Approximate approaches

Kusiak and Heragu [32] concluded that all the exact algorithms discussed in literature have the disadvantage of high memory and computational requirements, and the largest problem solved optimally is the problem with 15 facilities. This led researchers to focus on heuristics for solving FLP, which do not provide optimal solution but gives good quality solutions in reasonable computational time. Kusiak and Heragu [32] classified approximate approaches (or heuristics) for FLP as: construction algorithms, improvement algorithms, hybrid algorithms and graph theoretic algorithms.

*Construction algorithms* In construction algorithms, facilities are assigned to a site, one at a time, until the complete layout is obtained. Some of the popular construction algorithms available in literature are: HC66 proposed by Hillier and Connors [27], ALDEP due to Seehof and Evans [52], CORELAP given by Lee and Moore [35], RMA Comp I [41], MAT [15], PLANET [13], LSP [62], linear placement algorithm [42], FATE [5], INLAYT [45], SHAPE [21], and QLAARP [2]. The simplicity of construction algorithm is often associated with a poor quality of the resulting solutions. But these construction algorithms can be used to provide initial solutions for improvement algorithms. Improvement algorithms can be meta-heuristic such as SA and TS which require one feasible solution as starting solution for the execution of these algorithms.

*Improvement algorithms* In improvement algorithms, there is always an initial solution, which is often randomly generated. Based on this initial solution, systematic exchanges between facilities are made and the results are evaluated. The exchange which produces the best solution is retained, and the procedure continues until the solution cannot be improved any further. Hence, the solution quality of improvement algorithms depends upon the initial layout evaluated. Armour and Buffa [1] developed Computerized Relative Allocation of Facilities Technique (CRAFT) which is believed to be the first computerized technique used for the FLP. CRAFT begins by determining the cost of the initial layout. It then evaluates all possible location exchanges between pairs of facilities which either are adjacent to each other or are of the same area. The location exchange, which results in the highest cost reduction, is made. This procedure continues until there is no location exchange which results in a lesser cost than that of the current layout. CRAFT can handle only 40 facilities and does not perform well when the facilities are of unequal areas [17, 51]. Some more improvement algorithms for the FLP are: H63 [26], H63-66 [27], COL [61], Sampling algorithms [43], FRAT [29], COFAD [60], Revised Hillier algorithm [46], LOGIC [59], MULTIPLE [6, 14], SABLE [38] and MSA [55]. It is found that improvement algorithm greatly depends on the initial layout provided and the systematic procedure of the location exchange. The greedy nature of the pairwise exchange makes it susceptible to converge to a local optimum. Therefore, the shortcomings of improvement algorithms originate not only from the initial solution provided but also from the greedy nature of the systematic exchange

procedure. The greedy nature of the procedure is exposed because only the location exchanges, which result in the greatest cost reduction, are accepted. Hence, the nature of the exchange procedure often impedes the algorithm from finding the global optimum and causes the algorithm to converge to a local optimum.

*Hybrid algorithms* Hybrid algorithms have the characteristics of exact and heuristic algorithms or use the principles of construction and improvement algorithms. Examples of such algorithms can be found in Burkard and Stratman [7], Bazaraa and Sherali [3] and Bazaraa and Kirca [4].

*Graph theoretic algorithms* Seppannen and Moore [53] introduced the graph theoretic concept for layout design. Graph theoretic algorithms identify maximal planar subgraphs of a weighted graph which show the relationships between the facilities. The dual of a maximal planar subgraph determines the layout of the facilities. Goetschalckx [20] and Hassan and Hogg [23] also used graph theoretic algorithms. Hassan and Hogg [22] reviewed graph theoretic algorithms for FLP.

Besides these approximate approaches, Ramkumar et al. [47] presented iterated fast local search (IFLS) algorithm for solving QAP. Authors proposed a genetic algorithm-based local search approach for solving QAP. Ramkumar et al. [48] proposed a modification to IFLS with a new recombination crossover operator.

### 2.3 Meta-heuristics

The development of meta-heuristic has greatly influenced the performance of improvement algorithm and uses a general strategy like pairwise exchange heuristic. There are three widely used meta-heuristics in layout problem, i.e. SA, TS and GA. Although performance of meta-heuristic is faster and better than simple heuristic approach, but the quality of solution still depends on the initial solution as in the case of SA or TS. A survey of meta-heuristics solution methods for QAP can be found in Nehi and Gelareh [44].

## 3 Problem formulation

Consider the problem of locating $n$ facilities in $n$ given locations. Each location can be assigned to only one facility, and each facility can be assigned to only one location. There is material flow between the different departments and cost (material handling) associated with the unit flow per unit distance. Thus, different layouts have different total material handling costs depending on the relative location of the facilities. $F_{ik}$ is the flow between facilities $i$ and $k$, and $D_{jl}$ is the distance between locations $j$ and $l$. The FLP has been formulated as follows:

QAP:

$$\text{Min TF} = \sum_{\substack{i=1\\ i\neq k}}^{n}\sum_{\substack{j=1\\ j\neq 1}}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} F_{ik}\times \mathrm{D}_{jl}\times \mathrm{X}_{ij}\times \mathrm{X}_{kl} \tag{1}$$

$$\sum_{i=1}^{n} X_{ij} = 1 \quad \forall \quad j = 1, \ldots\ldots, n \tag{2}$$

$$\sum_{j=1}^{n} X_{ij} = 1 \quad \forall \quad i = 1, \ldots\ldots, n \tag{3}$$

$$X_{ij} \in \{0, 1\} \quad \forall \quad \mathrm{i,j} = 1, \ldots, \mathrm{n} \tag{4}$$

$X_{ij}$=1 if facility $i$ is located/assigned to location $j$ and $X_{ij}$= 0 if facility $i$ is not located/assigned to location $j$, where $n$ is the number of facilities. Equation (1) seeks to minimize the sum of flow multiplied by the distance for all pairs of facilities in a given layout. Equation (2) ensures that each location contains only one facility while Eq. (3) ensures that each facility is assigned to only one location. Solving the QAP optimally in a reasonable computational time is very hard, and therefore, the heuristics and meta-heuristics approach are designed to solve it optimally or near to optimal in reasonable CPU time. Authors propose a new approach to solve QAP optimally or near to optimal in a reasonable time.

## 4 Non-greedy systematic neighbourhood search heuristic

The proposed heuristic works on non-greedy systematic pairwise exchange of two facilities in the neighbourhood locations rather than exchange of facilities randomly. The neighbourhood NB(r) is defined as all assignments that can be reached from a given assignment when $r$ elements are exchanged. For pairwise exchange algorithms, the size of a neighbourhood is $[\mathrm{NB}(2)] = \frac{1}{2}n(n-1)$. Two alternatives exist for neighbourhood search process. First, choose the next potential facility for exchange randomly. Second,

**Table 1** Decision rule comparison of greedy descent and SA with the proposed heuristic

| Algorithm | Rule (conditions) |
|---|---|
| Greedy descent | $\Delta Z<0$ |
| Simulated annealing | $\Delta Z<0$ or random $[0,1] < \mathrm{e}^{-\Delta Z/t}$ |
| Non-greedy systematic neighbourhood search | $\Delta Z<0$or $Z < \left(\frac{\sum Z^{+}}{C}\right)/\varepsilon$ |

Input: $n$, $P_i$, $K$, *Iterations*, $\varepsilon$

/$n$- number of facility or problem size/

/ $P_i$ – initial random solution for proposed heuristic/

/$K$- maximum number of trials of the size of neighbourhood /

/*Iterations*- number of times heuristic is run/

/ $\varepsilon$ - initial value of the intensity factor, $\varepsilon$ [0, 1] /

$P_b = P_i$ /best solution treated so far/

$$h_\varepsilon = \frac{\varepsilon}{\max\left(1, \frac{Kn(n-1)}{2} - 1\right)} \quad \text{/intensity quantum/}$$

*Sum* = 0 /sum of the positive differences of the objective function value/

$C$ = 0 /counter of the positive differences of the objective function value/

$t$ = 0 /current trial number/

**for** (m=1, m<*Iterations*, m++) /for loop for number of Iterations/

**for** (k=1, k<$K$, k++) /for loop for the number of trials of the size of neighbourhood/

**for** (i=1, i<$n$-1, i++) /for loop for pair wise exchange of facilities/

**for** (j=i+1, j<$n$, j++) /for loop for pair wise exchange of facilities/

$P_n = N_2(P_i, i, j)$ / $P_n$ - new solution of pair wise exchange

$\Delta Z = Z(P_n) - Z(P_i)$ /difference of objective function value/

**if** $\Delta Z < 0$ **then** exchange = **TRUE** /start if loop/

**else** *Sum* = *Sum* + $\Delta Z$ , $C = C + 1$

$\varepsilon_c = \varepsilon - (t-1)h_\varepsilon$ /current intensity level/

**if** $\Delta Z < \left(\frac{Sum}{C}\right) / \varepsilon_c$ **then** exchange = **TRUE**

**else** exchange = **FALSE**

**end** /end of if loop/

$t = t + 1$ /next trial number/

**end** /end of for loop for the pair wise exchange of facilities/

**end** /end of the for loop for the K number of trials of size of neighbourhood search/

**return** $P_b$ / return best solution value after each Iteration/

**end** /end of for loop for number of Iterations

**return** $P_b$ / return best solution value after all Iterations

**Fig. 1** Pseudocode for non-greedy systematic neighbourhood search heuristic

explore the neighbourhood in a systematic way having all the possible exchange elements ordered ('shuffled'). The precise order is irrelevant; it is only essential that the neighbourhood is explored thoroughly. Proposed heuristic uses the later approach that is systematic (ordered) neighbourhood search, and this systematic neighbourhood search is based on the non-greedy strategy. The key idea is, pairwise exchanges are accepted if the objective function value after the exchange is lowered or smaller than the average objective function increment divided by an intensity factor. This is in contrast to the greedy improvement approaches where only the exchange, which lowers the objective function value, is allowed. The average objective function increment divided by an intensity factor is called threshold value and evaluated from Eq. (5) given below.

$$\tau = \left(\frac{\sum \Delta Z^+}{C}\right) / \varepsilon \tag{5}$$

$$\Delta Z = Z(P_c) - Z(P_i) \tag{6}$$

$$\Delta Z^+ = \Delta Z \text{ if } \Delta Z > 0 \tag{7}$$

where $P_i$ is the initial random solution, $P_c$ is current solution after pairwise exchange of two facilities and $\Delta Z$ is the current difference of the objective function value of the current solution ($Z(P_c)$) and initial random solution ($Z(P_i)$). $\varepsilon$ is the intensity factor which lies in the interval of [0,1]. $C$ is the total number of increments of objective function value (i.e. when $\Delta Z > 0$) or number of times when objective function value of current solution is more than old solution after pairwise exchange. From Eq. (5), it can be seen that $\tau$ is the average objective function increment (in $C$ number of increments) divided by the factor ($\varepsilon$). The acceptance rule can be finally defined as: the solution in the heuristic during pairwise exchange is accepted if and only if Eq. (8) does hold.

$$\Delta Z < 0 \text{ or } \Delta Z < \left(\frac{\sum \Delta Z^+}{C}\right) / \varepsilon \tag{8}$$

Table 1 shows few other acceptance rules used and available in the literature. From Table 1 it can be seen that the proposed heuristic approach possesses some features similar to those of SA. The main feature of SA is its ability to escape from local

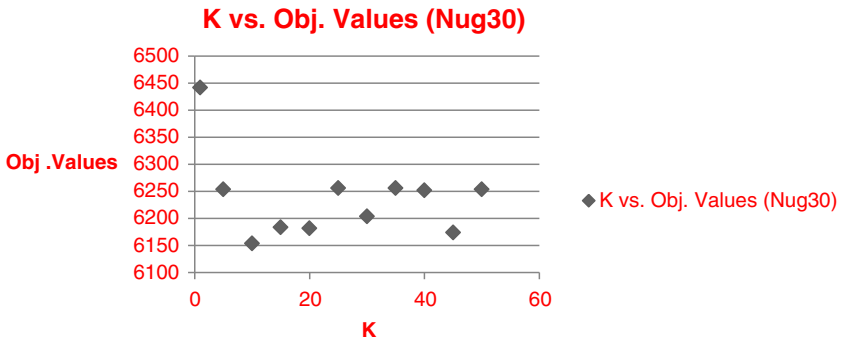


**Fig. 2** Objective function value of the proposed heuristic for different values of $K$

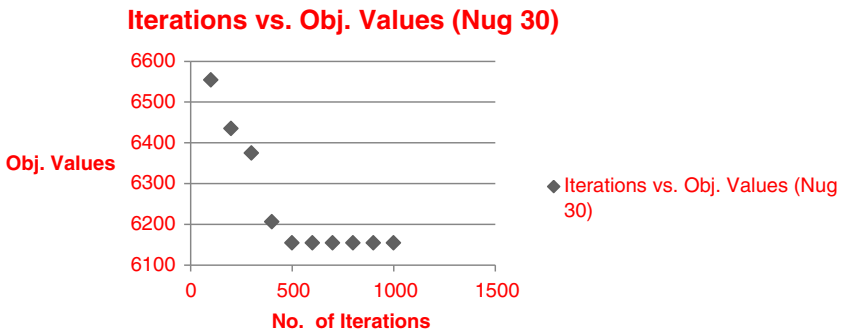


**Fig. 3** Objective function value of the proposed heuristic for different values of iterations

optimum based on the acceptance rule of a candidate solution. If the current solution ($P_c$) has an objective function value smaller (supposing minimization) than that of the initial solution ($P_i$); then, the current solution is accepted. Otherwise, the current solution can also be accepted if the value given by the Boltzmann distribution:

$$e^{-\Delta Z/t} \tag{9}$$

is greater than a uniform random number in [0,1], where $t$ is the 'temperature' control parameter. The solution in the SA during pairwise exchange is accepted if and only if Eq. (10) does hold.

$$\Delta Z < 0 \text{ or random } [0,1] < e^{-\Delta Z/t} \tag{10}$$

The proposed heuristic also tries to escape from local optimum as in SA using non-greedy approach; however, the acceptance rule of the candidate solution in proposed heuristic is different (Eq. 8) than SA (Eq. 10). In SA, 'temperature' is the control parameter; however, in the proposed non-greedy systematic neighbourhood search (NGSNS) heuristic, 'intensity factor' is the control parameter.

We present a pseudo code in Fig. 1 to better understand the proposed heuristic approach. In the pseudo code, $K$ is the number of trials of the size of a neighbourhood. Since the size of a neighbourhood is $[\mathrm{NB}(2)] = \frac{1}{2}n(n-1)$, therefore, in each iteration, the total number of swaps examined is $K\frac{n(n-1)}{2}$. The value of $K$ can influence the quality of results considerably and also computational time. If the value of $K$ is small, then no guarantee exists that the quality of the solution is good. On the other side, if the value of $K$ is large, then the quality of solution is good enough in many cases but at the cost of computational time.

To determine the default value of $K$, we applied NGSNS on Nug30 problem for 500 iterations with the value of $K$ ranging from 1 to 50, with an increment of 5 each time. For each value of $K$, objective function values are noted. Figure 2 shows the results on a graph. It can be seen in graph at $K$=10, objective function value is minimum. Therefore, we set the default value of $K$ equal to 10 for NGSNS heuristic.

The next question arises what should be the number of iterations, since the number of iterations can also influence quality of solution and computational time. To answer this question, we again applied NGSNS on the Nug30 problem taking the default value of $K$=10; however, the number of iterations varied from 100 to 1,000. Again Fig. 3 shows results on a graph. It can be seen easily in the graph that objective function values converge at 500 or more number of iterations. Therefore, we decided the number of iterations to be equal to 500 for the proposed NGSNS.

## 5 Procedural steps of the proposed NGSNS heuristic

Step 1: Randomly generate solution $P_i$ and calculate the corresponding objective value $Z$ using Eq. (1) Initialize: Iterations (specify total number of iterations), sum=0, C (counter)=0 and $t$ (trial number) =0, $K$ (size of neighbourhood)=$n(n-1)/2$ where $n$ is the number of facilities and intensity factor $\varepsilon$=1.

Step 2: Randomly select two facilities and check whether they are the same facilities. If yes, go to step 5. Other interchange facilities to get new solution $P_n$ and calculate new $Z$ and change in objective value $\Delta Z$=new $Z$–old $Z$.

Step 3: If $\Delta Z<0$ is true, accept new solution and new $Z$= old $Z+\Delta Z$. Otherwise, go to step 4.

**Table 2** Distance matrix for example problem

| Locations from/to | Locations | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Location | | | | | | |
| 1 | 0 | 1 | 2 | 1 | 2 | 3 |
| 2 | 1 | 0 | 1 | 2 | 1 | 2 |
| 3 | 2 | 1 | 0 | 3 | 2 | 1 |
| 4 | 1 | 2 | 3 | 0 | 1 | 2 |
| 5 | 2 | 1 | 2 | 1 | 0 | 1 |
| 6 | 3 | 2 | 1 | 2 | 1 | 0 |

**Table 3** Flow matrix for example problem

| From/to | Facilities | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Facility | | | | | | |
| 1 | 0 | 5 | 2 | 4 | 1 | 0 |
| 2 | 5 | 0 | 3 | 0 | 2 | 2 |
| 3 | 2 | 3 | 0 | 0 | 0 | 0 |
| 4 | 4 | 0 | 0 | 0 | 5 | 2 |
| 5 | 1 | 2 | 0 | 5 | 0 | 10 |
| 6 | 0 | 2 | 0 | 2 | 10 | 0 |

Step 4: Sum=sum+$\Delta Z$, increment counter $C=C+1$. Calculate intensity quantum

$$h_\varepsilon = \frac{\varepsilon}{\max\left(1, \frac{Kn(n-1)}{2} - 1\right)}$$

Update current intensity level $\varepsilon_c=\varepsilon-(t-1)h_\varepsilon$. If $\Delta Z < \left(\frac{\text{Sum}}{C}\right)/\varepsilon_c$ is true, accept new solution, i.e. new $Z$=old $Z+\Delta Z$. Else, go to step 5.

Step 5: Increment trial number $t=t+1$.

Step 6: If $t<=K$ is true, repeat steps 2 to 5. Otherwise current iteration is over and return best solution $P_b$ of current iteration and go to step 7.

Step 7: If the specified number of iterations are performed, terminate and select the minimum $Z$ value from all iterations and return corresponding best solution $P_b$; otherwise, repeat steps 1 to 6.

**Table 4** Solution of problem instances with series Nugxx taken from Burkard et al. [11]

| S. no. | Instance | Optimal/ BKS | Proposed heuristic solution | % deviation | CPU |
|---|---|---|---|---|---|
| 1 | Nug12 | **578** | **578** | 0.00 | 0.702 |
| 2 | Nug14 | **1,014** | **1,014** | 0.00 | 0.812 |
| 3 | Nug15 | **1,150** | **1,150** | 0.00 | 2.671 |
| 4 | Nug16a | **1,610** | **1,610** | 0.00 | 3.406 |
| 5 | Nug16b | **1,240** | **1,240** | 0.00 | 3.406 |
| 6 | Nug17 | **1,732** | **1,732** | 0.00 | 1.781 |
| 7 | Nug18 | **1,930** | **1,930** | 0.00 | 5.484 |
| 8 | Nug20 | **2,570** | **2,570** | 0.00 | 8.375 |
| 9 | Nug21 | **2,438** | **2,438** | 0.00 | 20.171 |
| 10 | Nug22 | **3,596** | **3,596** | 0.00 | 12.281 |
| 11 | Nug24 | **3,488** | **3,488** | 0.00 | 70.218 |
| 12 | Nug25 | **3,744** | **3,744** | 0.00 | 202.921 |
| 13 | Nug27 | **5,234** | **5,234** | 0.00 | 227.625 |
| 14 | Nug28 | **5,166** | **5,166** | 0.00 | 637.562 |
| 15 | Nug30 | **6,124** | 6,154 | 0.48 | 646.937 |

### 5.1 Numerical illustration

The working of the proposed heuristic is explained using the data presented in Tables 2 and 3. In Table 2, the distance matrix based on the distance between facilities is presented, and Table 3 shows the flow of materials between facilities for problem size $n$=6.

Step 1: Randomly generate solution $P_i$
$P_i$ 2 3 4 1 6 5 old $Z$=110
Initialize iterations=10, sum=0, $t$=0, $C$=0, $K$=$n(n-1)/2=6\times5/2=15$ and $\varepsilon$=1

Step 2: Randomly, two facilities 2 and 6 are interchanged.
$P_n$ 6 3 4 1 2 5 new $Z$=142 and $\Delta Z$=142–110=32.

Step 3: Since $\Delta Z>0$, go to step 4.

**Table 5** Solution of problem instances with series Taixxx taken from Burkard et al. [11]

| S. no. | Instance | Optimal/BKS | Proposed heuristic solution | % deviation | CPU |
|---|---|---|---|---|---|
| 1 | Tai12a | **224,416** | **224,416** | 0.00 | 11.312 |
| 2 | Tai12b | **39,464,925** | **39,464,925** | 0.00 | 11.234 |
| 3 | Tai15a | **388,214** | **388,214** | 0.00 | 27.109 |
| 4 | Tai15b | **51,765,268** | **51,765,268** | 0.00 | 26.937 |
| 5 | Tai17a | **491,812** | **491,812** | 0.00 | 17.578 |
| 6 | Tai20a | **703,482** | 705,622 | 0.304 | 24.984 |
| 7 | Tai20b | **122,455,319** | **122,455,319** | 0.00 | 24.782 |
| 8 | Tai25a | 1,167,256 | 1,187,088 | 1.69 | 42.796 |
| 9 | Tai25b | **344,355,646** | **344,355,646** | 0.00 | 41.453 |
| 10 | Tai30a | 1,818,146 | 1,855,734 | 2.06 | 64.218 |
| 11 | Tai30b | 637,117,113 | 637,137,951 | 0.003 | 63.484 |
| 12 | Tai35a | 2,422,002 | 2,471,980 | 2.06 | 81.921 |
| 13 | Tai35b | 283,315,445 | 284,350,938 | 0.36 | 81.875 |
| 14 | Tai40a | 3,139,370 | 3,222,972 | 2.66 | 138.853 |
| 15 | Tai40b | 637,250,948 | 637,250,948 | 0.00 | 138.156 |
| 16 | Tai50a | 4,938,796 | 5,069,742 | 2.65 | 337.046 |
| 17 | Tai50b | 458,821,517 | 460,718,405 | 0.41 | 339.343 |
| 18 | Tai60a | 7,205,962 | 7,424,858 | 3.03 | 704.703 |
| 19 | Tai60b | 608,215,054 | 609,491,668 | 0.2 | 678.515 |
| 20 | Tai64c | 1,855,928 | 1,855,928 | 0.00 | 776.76 |
| 21 | Tai80a | 13,499,184 | 13,871,316 | 2.75 | 882.5 |
| 22 | Tai80b | 818,415,043 | 833,128,227 | 1.79 | 883.67 |
| 23 | Tai100a | 21,052,466 | 21,664,160 | 2.9 | 1,423.56 |
| 24 | Tai100b | 1,185,996,137 | 1,201,244,218 | 1.28 | 1,425.87 |
| 25 | Tai150b | 498,896,643 | 508,987,944 | 2.02 | 2,232.76 |
| 26 | Tai256c | 44,759,294 | 44,891,576 | 0.295 | 4,234.78 |

**Table 6** Solution of problem instances with series Stexxx taken from Burkard et al. [11]

| S. no. | Instance | Optimal/BKS | Proposed heuristic solution | % deviation | CPU |
|---|---|---|---|---|---|
| 1 | Ste36a | **9,526** | 9,648 | 1.28 | 176.437 |
| 2 | Ste36b | **15,852** | 16,054 | 1.27 | 176.484 |
| 3 | Ste36c | **8,239,110** | 8,318,780 | 0.97 | 176.593 |

Step 4: Sum=0+32=32, increment counter $C$=0+1=1.

$$h_\varepsilon = \frac{\varepsilon}{\max\left(1, \frac{Kn(n-1)}{2} - 1\right)} = \frac{1}{\max\left(1, \frac{15.6.5}{2} - 1\right)} = \frac{1}{224} = 0.004464$$

Update current intensity level, $\varepsilon_c = \varepsilon - (t-1)h_\varepsilon = 1-(0-1)\times 0.004464 = 1.004464$

$$\left(\frac{\text{Sum}}{C}\right)/\varepsilon_c = 32/1.004464 = 31.857$$

Since $\Delta Z$ (32)>31.857, go to step 5.

Step 5: Increment trial number $t=t+1=0+1=1$.

Step 6: Since trial number $t(1)<K(15)$, go to step 2 (repeat steps 2 to 5). When $t=K$ trials are performed, the current iteration is over, and return the best solution $P_b$ of the current iteration and go to step 7.

Step 7: If iterations=10 are performed, terminate and select the minimum $Z$ value from all iterations and return the corresponding best solution $P_b$; otherwise, repeat steps 1 to 6.

**Table 7** Solution of problem instances with series skoxx, skoxxxx taken from Burkard et al. [11]

| S. no. | Instance | Optimal/ BKS | Proposed heuristic solution | % deviation | CPU |
|---|---|---|---|---|---|
| 1 | sko42 | 15,812 | 15,882 | 0.44 | 161.828 |
| 2 | sko49 | 23,386 | 23,536 | 0.64 | 306.343 |
| 3 | sko56 | 34,458 | 34,736 | 0.8 | 518.39 |
| 4 | sko64 | 48,498 | 48,878 | 0.78 | 892.89 |
| 5 | sko72 | 66,256 | 67,176 | 1.38 | 944.45 |
| 6 | sko81 | 90,998 | 92,004 | 1.1 | 970.56 |
| 7 | sko90 | 115,534 | 116,932 | 1.21 | 1,002.89 |
| 8 | sko100a | 152,002 | 153,528 | 1.004 | 1,076.86 |
| 9 | sko100b | 153,890 | 155,556 | 1.08 | 1,065.675 |
| 10 | sko100c | 147,862 | 149,316 | 0.98 | 1,071.77 |
| 11 | sko100d | 149,576 | 150,940 | 0.91 | 1,085.67 |
| 12 | sko100e | 149,150 | 150,566 | 0.94 | 1,077.36 |
| 13 | sko100f | 149,036 | 150,708 | 1.12 | 1,080.39 |

**Table 8** Solution of problem instances with series Wilxx, Wilxxx taken from Burkard et al. [11]

| S. no. | Instance | Optimal/ BKS | Proposed heuristic solution | % deviation | CPU |
|---|---|---|---|---|---|
| 1 | Wil50 | 48,816 | 48,968 | 0.31 | 330.968 |
| 2 | Wil100 | 273,038 | 274,390 | 0.49 | 2,170.73 |

## 6 Performance of non-greedy systematic neighbourhood search heuristic

The proposed heuristic has been coded in *C* and tested on a Core2duo machine having a processor speed 2.2 GHz with RAM of 2.96 GB. The experiments were conducted in two stages. Stage one is designed to calibrate parameter $K$ to fix the default value of the parameter. Using the default value identified in stage 1, experiments were conducted using benchmark problems from the QAP library [11]. The proposed heuristic is executed for 500 number of iterations keeping the default value of $K$ equal to 10. In each iteration, a new random solution is generated which is improved by pairwise exchanges. Heuristic performs $K$ times $n(n-1)/2$ swaps in each iteration. Different solutions are produced in each iteration, and one best solution is selected from all iterations as final solution.

To validate the performance of the proposed heuristic, all problem instances available in the QAPLIB, an online version of Burkard et al. [11] available at http://www.seas.upenn.edu/qaplib, are tested in the paper. Problem instances of all authors available at QAPLIB are solved, and results are presented in various tables according to the author's names (refer to Tables 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,

**Table 9** Solution of problem instances with series Chrxxx taken from Burkard et al. [11]

| S. no. | Instance | Optimal/ BKS | Proposed | % deviation heuristic solution | CPU |
|---|---|---|---|---|---|
| 1 | Chr12a | **9,552** | **9,552** | 0.00 | 1.14 |
| 2 | Chr12b | **9,742** | **9,742** | 0.00 | 1.156 |
| 3 | Chr12c | **11,156** | **11,156** | 0.00 | 1.156 |
| 4 | Chr15a | **9,896** | **9,896** | 0.00 | 5.265 |
| 5 | Chr15b | **7,990** | **7,990** | 0.00 | 5.265 |
| 6 | Chr15c | **9,504** | **9,504** | 0.00 | 8.156 |
| 7 | Chr18a | **11,098** | **11,098** | 0.00 | 5.515 |
| 8 | Chr18b | **1,534** | **1,534** | 0.00 | 5.518 |
| 9 | Chr20a | **2,192** | 2,232 | 1.82 | 8.406 |
| 10 | Chr20b | **2,298** | 2,432 | 5.83 | 16.796 |
| 11 | Chr20c | **14,142** | **14,142** | 0.00 | 16.921 |
| 12 | Chr22a | **6,156** | 6,210 | 0.87 | 24.546 |
| 13 | Chr22b | **6,194** | 6,268 | 1.19 | 24.937 |
| 14 | Chr25a | **3,796** | 4,106 | 8.16 | 30.265 |

**Table 10** Solution of problem instances with series Thoxx, Thoxxx taken from Burkard et al. [11]

| S. no. | Instance | Optimal/ BKS | Proposed heuristic solution | % deviation | CPU |
|---|---|---|---|---|---|
| 1 | Tho30 | **149,936** | **149,936** | 0.00 | 62.703 |
| 2 | Tho40 | 240,516 | 242,720 | 0.92 | 335.678 |
| 3 | Tho150 | 8,133,398 | 8,222,574 | 1.09 | 1,638.78 |

15, 16, 17, and 18). A comparative analysis of the solutions obtained from the proposed heuristic approach with the optimal and best known solutions available in literature is also tabulated along with the percentage deviation from optimal/best known solutions. In addition to this, computational time of the proposed heuristic approach (in CPU seconds) is also reported for each problem instance tested in the paper.

Out of the 135 test problems available in QAPLIB, 86 were solved to optimality as of June 2011 (optimal values are shown bold in various authors' tables). The remaining 49 had best known solutions reported by various heuristics. In this paper, out of the 135 test problems proposed, heuristic has obtained optimal solutions for 64 problems and matched the best known solution value for 15 problems.

**Table 11** Solution of problem instances with series Escxxx taken from Burkard et al. [11]

| S. no. | Instance | Optimal/ BKS | Proposed heuristic solution | % deviation | CPU |
|---|---|---|---|---|---|
| 1 | Esc16a | **68** | **68** | 0.00 | 1.39 |
| 2 | Esc16b | **292** | **292** | 0.00 | 1.468 |
| 3 | Esc16c | **160** | **160** | 0.00 | 1.421 |
| 4 | Esc16d | **16** | **16** | 0.00 | 1.406 |
| 5 | Esc16e | **28** | **28** | 0.00 | 1.39 |
| 6 | Escf16f | **0** | **0** | 0.00 | 1.384 |
| 7 | Esc16g | **26** | **26** | 0.00 | 1.375 |
| 8 | Esc16h | **996** | **996** | 0.00 | 1.515 |
| 9 | Esf16i | **14** | **14** | 0.00 | 1.406 |
| 10 | Esc16j | **8** | **8** | 0.00 | 1.421 |
| 11 | Esc32a | 130 | 134 | 3.08 | 22.343 |
| 12 | Esc32b | 168 | 168 | 0.00 | 22.093 |
| 13 | Esc32c | 642 | 642 | 0.00 | 22.218 |
| 14 | Esc32d | 200 | 200 | 0.00 | 22.171 |
| 15 | Esc32e | **2** | **2** | 0.00 | 22.125 |
| 16 | Esc32f | **2** | **2** | 0.00 | 22.343 |
| 17 | Esc32g | **6** | **6** | 0.00 | 22.281 |
| 18 | Esc32h | 438 | 438 | 0.00 | 22.171 |
| 19 | Esc64a | 116 | 116 | 0.00 | 360.75 |
| 20 | Esc128 | 64 | 64 | 0.00 | 1,658.732 |

**Table 12** Solution of problem instances with series burxxx taken from Burkard et al. [11]

| S. no. | Instance | Optimal/ BKS | Proposed heuristic solution | % deviation | CPU |
|---|---|---|---|---|---|
| 1 | bur26a | **5,426,670** | **5,426,670** | 0.00 | 24.093 |
| 2 | bur26b | 3,817,852 | 3,817,852 | 0.00 | 19.234 |
| 3 | bur26c | 5,426,795 | 5,426,795 | 0.00 | 23.953 |
| 4 | bur26d | 3,821,225 | 3,821,225 | 0.00 | 23.984 |
| 5 | bur26e | 5,386,879 | 5,386,879 | 0.00 | 23.984 |
| 6 | bur26f | 3,782,044 | 3,782,044 | 0.00 | 23.984 |
| 7 | bur26g | 10,117,172 | 10,117,172 | 0.00 | 24.312 |
| 8 | bur26h | 7,098,658 | 7,098,658 | 0.00 | 23.984 |

In QAPLIB, the best known solution to the non-optimal problems is provided by many different heuristic and meta-heuristic approaches. Of these meta-heuristic approaches, Robust TS [58], Genetic Hybrid [16], GRASP [36] and SA [10] appear more often as the first heuristic to reach the best known solution. Other heuristics may reach the same solution as well, but QAPLIB only recognizes the first such heuristic. It must be noted that no known heuristic reportedly produced uniformly better solutions for all problems.

This raises the question how proposed non-greedy systematic neighbourhood search heuristic compare to the four meta-heuristics mentioned above in the instances not proven to be optimal. Table 19 gives a comparison of these meta-heuristics and proposed heuristic in these non-optimal instances. A checkmark in the first four columns indicates which method first reported the best known solution to that

**Table 13** Solution of problem instances with series lipaxxx taken from Burkard et al. [11]

| S. no. | Instance | Optimal/ BKS | Proposed heuristic solution | % deviation | CPU |
|---|---|---|---|---|---|
| 1 | lipa20a | **3,683** | **3,683** | 0.00 | 3.39 |
| 2 | lipa20b | **27,076** | **27,076** | 0.00 | 3.406 |
| 3 | lipa30a | **13,178** | **13,178** | 0.00 | 125.484 |
| 4 | lipa30b | **151,426** | **151,426** | 0.00 | 42.531 |
| 5 | lipa40a | **31,538** | 31,870 | 1.05 | 138.062 |
| 6 | lipa40b | **476,581** | **476,581** | 0.00 | 137.562 |
| 7 | lipa50a | **62,093** | 62,739 | 1.04 | 132.781 |
| 8 | lipa50b | **1,210,244** | **1,210,244** | 0.00 | 132.671 |
| 9 | lipa60a | **107,218** | 108,225 | 0.93 | 292.375 |
| 10 | lipa60b | **2,520,135** | **2,520,135** | 0.00 | 290.953 |
| 11 | lipa70a | **169,755** | 171,092 | 0.79 | 514.375 |
| 12 | lipa70b | **4,603,200** | 5,501,669 | 19.51 | 516.015 |
| 13 | lipa80a | **253,195** | 254,958 | 0.69 | 876.734 |
| 14 | lipa80b | **7,763,962** | 9,359,096 | 20.54 | 884.265 |
| 15 | lipa90a | **360,630** | 363,005 | 0.65 | 1,411.65 |
| 16 | lipa90b | **12,490,441** | 15,087,930 | 20.79 | 1,412.88 |

**Table 14** Solution of problem instances with series Kraxx taken from Burkard et al. [11]

| S. no. | Instance | Optimal/ BKS | Proposed heuristic solution | % deviation | CPU |
|---|---|---|---|---|---|
| 1 | Kra30a | **88,900** | 89,800 | 1.01 | 16.843 |
| 2 | Kra30b | **91,420** | 91,590 | 0.18 | 84.078 |
| 3 | Kra32 | **88,700** | 90,320 | 1.82 | 109.703 |

**Table 15** Solution of problem instances with series Hadxx taken from Burkard et al. [11]

| S. no. | Instance | Optimal/ BKS | Proposed heuristic solution | % deviation | CPU |
|---|---|---|---|---|---|
| 1 | Had12 | **1,652** | **1,652** | 0.00 | 0.468 |
| 2 | Had14 | **2,724** | **2,724** | 0.00 | 0.812 |
| 3 | Had16 | **3,720** | **3,720** | 0.00 | 1.359 |
| 4 | Had18 | **5,358** | **5,358** | 0.00 | 2.234 |
| 5 | Had20 | **6,922** | **6,922** | 0.00 | 3.421 |

**Table 16** Solution of problem instances with series Rouxx taken from Burkard et al. [11]

| S. no. | Instance | Optimal/ BKS | Proposed heuristic solution | % deviation | CPU |
|---|---|---|---|---|---|
| 1 | Rou12 | **235,528** | **235,528** | 0.00 | 1.14 |
| 2 | Rou15 | **354,210** | **354,210** | 0.00 | 2.687 |
| 3 | Rou20 | **725,522** | **725,522** | 0.00 | 25.203 |

**Table 17** Solution of problem instances with series Scrxx taken from Burkard et al. [11]

| S. no. | Instance | Optimal/ BKS | Proposed heuristic solution | % deviation | CPU |
|---|---|---|---|---|---|
| 1 | Scr12 | **31,410** | **31,410** | 0.00 | 1.156 |
| 2 | Scr15 | **51,140** | **51,140** | 0.00 | 2.671 |
| 3 | Scr20 | **110,030** | **110,030** | 0.00 | 25.828 |

**Table 18** Solution of problem instances with series Elsxx taken from Burkard et al. [11]

| S. no. | Instance | Optimal/ BKS | Proposed heuristic solution | % deviation | CPU |
|---|---|---|---|---|---|
| 1 | Els19 | **17,212,548** | **17,212,548** | 0.00 | 2.718 |

**Table 19** Comparative analysis of past meta-heuristics and proposed heuristic for QAPLIB instances

| S. no. | Name | Ro-TS | GEN | GRASP | SIM | NGSNS |
|---|---|---|---|---|---|---|
| 1 | bur26b | | | √ | | Y |
| 2 | bur26c | | | √ | | Y |
| 3 | bur26d | | | √ | | Y |
| 4 | bur26e | | | √ | | Y |
| 5 | bur26f | | | √ | | Y |
| 6 | bur26g | | | √ | | Y |
| 7 | bur26h | | | √ | | Y |
| 8 | esc32a | √ | | Y | | N |
| 9 | esc32b | √ | | Y | | Y |
| 10 | esc32c | | | Y | √ | Y |
| 11 | esc32d | √ | | Y | | Y |
| 12 | esc32h | √ | | Y | | Y |
| 13 | esc64a | | | Y | √ | Y |
| 14 | esc128 | | | √ | | Y |
| 15 | sko42 | √ | | | N | N |
| 16 | sko49 | √ | | | N | N |
| 17 | sko56 | √ | | | N | N |
| 18 | sko64 | √ | | | N | N |
| 19 | sko72 | √ | | | N | N |
| 20 | sko81 | | √ | | N | N |
| 21 | sko90 | √ | | | N | N |
| 22 | sko100a | | √ | | | N |
| 23 | sko100b | | √ | | | N |
| 24 | sko100c | | √ | | | N |
| 25 | sko100d | | √ | | | N |
| 26 | sko100e | | √ | | | N |
| 27 | sko100f | | √ | | | N |
| 28 | tai25a | √ | | | | N |
| 29 | tai30a | √ | | | | N |
| 30 | tai30b | √ | | | | N |
| 31 | tai35a | √ | | | | N |
| 32 | tai35b | √ | | | | N |
| 33 | tai40a | √ | | | | N |
| 34 | tai40b | √ | | | | Y |
| 35 | tai50a | | √ | | | N |
| 36 | tai50b | √ | | | | N |
| 37 | tai60a | √ | N | | | N |
| 38 | tai60b | √ | | | | N |
| 39 | tai64c | √ | | | | Y |
| 40 | tai80a | | N | | | N |
| 41 | tai80b | √ | | | | N |
| 42 | tai100b | √ | | | | N |
| 43 | tai150b | | √ | | | N |
| 44 | wil50 | Y | | | √ | N |
| 45 | wil100 | N | √ | | N | N |
| 46 | Tho40 | | | | √ | N |
| 47 | Tho150 | | | | √ | N |

√ indicates which heuristic first time reported the best known solution; Y indicates which heuristic matches the best known solution; N indicates which heuristic does not match the best known solution

problem as recognized by QAPLIB. A 'Y' or 'N' indicates whether the heuristic matches the best known solution. Blank entries indicate that the authors did not report a solution in their paper.

## 7 Conclusions and future research direction

In this paper, a non-greedy systematic neighbourhood search heuristic for finding a quality solution in reasonable computational time for the FLP is presented. Heuristic is applied on a large number of instances provided by different authors in QAPLIB with problem sizes ranging from 12 to 256. Out of the 135 test problems in QALIB, the proposed heuristic has obtained optimal solutions for 64 problems and matched the best known solution value for 15 problems. For the remaining 56 problems, the proposed heuristic has given very good quality solutions (44 problem solutions within 0–2 % deviation from the optimal/best known values) as shown in various tables. It must be noted that no known heuristic reportedly produced uniformly better solutions for all problem instances of QAPLIB. The proposed heuristic tries to fill the above gap and performs reasonably well on all instances of QAPLIB. A comparative analysis with past meta-heuristic approaches (Table 19) shows the effectiveness of the proposed NGSNS heuristic. It is also proposed as a future research direction to incorporate some GA parameters such as crossover and mutation in the proposed heuristic approach for further improvement in the performance of the proposed solution methodology.

## References


1. Armour GC, Buffa ES (1963) A heuristic algorithm and simulation approach to relative location of facilities. Manag Sci 9(2):294–309
2. Banerjee P, Montreuil B, Moodie CL, Kashyap RL (1992) A modeling of interactive facilities layout designer reasoning using qualitative patterns. Int J Prod Res 30(3):433–453
3. Bazarra MS, Sherali HD (1980) Bender's partitioning scheme applied to a new formulation of the quadratic assignment problem. Nav Res Logist Q, Issue 1:29–41
4. Bazaraa MS, Kirca O (1983) A branch-and-bound-based heuristic for solving the QAP. Nav Res Logist Q 30(2):287–304
5. Block TE (1978) FATE: a new construction algorithm for facilities layout. J Eng Prod 2:111–120
6. Bozer YA, Meller RD (1994) An improvement-type layout algorithm for single and multiple floor facilities. Manag Sci 40(7):918–932
7. Burkard RE, Stratman KH (1978) Numerical investigations on quadratic assignment problems. Nav Res Logist Q 25(1):129–148
8. Burkard RE, Bonniger T (1983) A heuristic for quadratic Boolean program with application to quadratic assignment problems. Eur J Oper Res 13(4):374–386
9. Burkard RE (1984) Locations with spatial interaction—quadratic assignment problem. In: Francis RL, Mirchandani PB (eds) Discrete location theory. Academic, New York
10. Burkard RE, Rendl F (1984) A thermodynamically motivated simulation procedure for combinatorial optimization problems. Eur J Oper Res 17(2):169–174
11. Burkard RE, Karisch S, Rendl F (1997) QAPLIB—a quadratic assignment problem library. J Glob Optim 10(1):391–403
12. Christofides N, Benavent E (1989) An exact algorithm for the quadratic assignment problem on a tree. Oper Res 37(5):760–768
13. Deisenroth MP, Apple JM (1972) A computerized plant layout analysis and evaluation technique. Technical paper, Annual AIIE Conference, Norcross, GA
14. Diponegoro A, Sarker BR (2003) Machine assignment in a nonlinear multi-product flowline. J Oper Res Soc 54:472–489
15. Edwards HK, Gillett BE, Hale ME (1970) Modular allocation technique (MAT). Manag Sci 17(3):161–169
16. Fleurent C, Ferland JA (1994) Genetic hybrids for the quadratic assignment problem. DIMACS Ser Math Theor Comput Sci 16(IRO-870):190–206
17. Foulds LR (1983) Techniques for facilities layout: deciding which pairs of activities should be adjacent. Manag Sci 29(12):1414–1426
18. Gavett JW, Plyter NV (1966) The optimal assignment of facilities to locations by branch and bound. Oper Res 14(2):210–232
19. Gilmore PC (1962) Optimal and suboptimal algorithms for the quadratic assignment problem. J Soc Ind Appl Math 10(2):305–313
20. Goetschalckx M (1992) An interactive layout heuristic based on hexagonal adjacency graphs. Eur J Oper Res 63(2):304–321
21. Hassan MMD, Hogg GL, Smith DR (1986) SHAPE: a construction algorithm for area placement evaluation. Int J Prod Res 24(5):1283–1295
22. Hassan MMD, Hogg GL (1987) A review of graph theory applications to the facilities layout problem. Omega 15(4):291–300
23. Hassan MMD, Hogg GL (1989) On converting a dual graph into a block layout. Int J Prod Res 27(7):1149–1160
24. Heragu SS, Kusiak A (1990) Machine layout: an optimization and knowledge based approach. Int J Prod Res 28(4):615–635
25. Heragu SS (1992) Recent models and techniques for solving the layout problem. Eur J Oper Res 57(2):136–144
26. Hillier FS (1963) Quantitative tools for plant layout analysis. J Ind Eng 14(1):33–40
27. Hillier FS, Connors MM (1966) Quadratic assignment problem algorithms and the location of indivisible facilities. Manag Sci 13(1):42–57
28. Kaku BK, Thompson GL (1986) An exact algorithm for the general quadratic assignment problem. Eur J Oper Res 23(3):382–390
29. Khalil TM (1973) Facilities relative allocation technique (FRAT). Int J Prod Res 11(2):183–194
30. Koopmans TC, Beckman M (1957) Assignment problems and the location of economic activities. Econometrica 25(1):53–76
31. Kochhar JS, Heragu SS (1998) MULTI-HOPE: a tool for multiple floor layout problems. Int J Prod Res 36(12):3421–3435
32. Kusiak A, Heragu SS (1987) The facility layout problem. Eur J Oper Res 29(3):229–251
33. Land AH (1963) A problem of assignment with interrelated costs. Oper Res Q 14:185–198
34. Lawler EL (1963) The quadratic assignment problem. Manag Sci 9(4):586–599
35. Lee R, Moore JM (1967) CORELAP—computerized relationship layout planning. J Ind Eng 18:195–200
36. Li Y, Pardalos PM, Resende MGC (1994) A greedy randomized adaptive search procedure for the quadratic assignment problem. DIMACS Ser Discret Math Theor Comput Sci 16:237–261

37. Matai R, Singh SP, Mittal ML (2010) Facility layout problem: a state-of-the-art review. Vilakshan, XIMB J Manag 7(2):81–106
38. Meller RD, Bozer YA (1996) A new simulated annealing algorithm for the facility layout problem. Int J Prod Res 34(6):1675–1692
39. Meller RD, Gau KY (1996) The facility layout problem: recent and emerging trends and perspectives. J Manuf Syst 15(5):351–366
40. Meller RD, Gau KY (1996) Facility layout objective functions and robust layouts. Int J Prod Res 34(10):2727–2742
41. Muther R, McPherson K (1970) Four approaches to computerized layout planning. Ind Eng 21:39–42
42. Neghabat F (1974) An efficient equipment layout algorithm. Oper Res 22(3):622–628
43. Nugent CE, Vollman TE, Ruml J (1968) An experimental comparison of techniques for the assignment of facilities to locations. Oper Res 16(1):150–173
44. Nehi HM, Gelareh S (2007) A survey of meta-heuristic solution methods for the quadratic assignment problem. Appl Math Sci 1(46):2293–2312
45. O'Brien C, Abdel-Barr SEZ (1980) An interactive approach to computer aided facility layout. Int J Prod Res 18(2):201–211
46. Picone CJ, Wilhelm WE (1984) A perturbation scheme to improve Hillier's solution to the facilities layout problem. Manag Sci 30(10):1238–1249
47. Ramkumar AS, Ponnambalam SG, Jawahar N, Suresh RK (2008) Iterated fast local search algorithm for solving quadratic assignment problems. Robot Comput-Integr Manuf 24(3):392–401
48. Ramkumar AS, Ponnambalam SG, Jawahar N (2009) A new iterated fast local search heuristic for solving QAP formulation in facility layout design. Robot Comput-Integ Manuf 25(3):620–629
49. Rosenblatt MJ, Lee HL (1987) A robustness approach to facilities design. Int J Prod Res 25(4):479–486
50. Sahni S, Gonzalez P (1976) P-Complete approximation problems. J ACM 23(3):555–565
51. Scriabin M, Vergin RC (1976) Computer and visual methods for plant layout—a rejoinder. Manag Sci 23(l):104–105
52. Seehof JM, Evans WO (1967) Automated layout design program. J Ind Eng 18:690–695
53. Seppannen JJ, Moore JM (1970) Facilities planning with graph theory. Manag Sci 17(4):242–253
54. Singh SP, Sharma RRK (2006) A survey on various approaches to solve facility layout problem. Int J Adv Manuf Technol 30:425–433
55. Singh SP, Sharma RRK (2008) Two-level modified simulated annealing based approach for solving facility layout problem. Int J Prod Res 46(13):3563–3582
56. Singh SP, Singh VK (2010) An improved heuristic approach for multi-objective approach for facility layout problem. Int J Prod Res 48(4):1171–1194
57. Singh SP, Singh VK (2011) Three-level AHP based heuristic approach to solve multi-objective facility layout problem. Int J Prod Res 49(4):1105–1125
58. Taillard E (1991) Robust taboo search for the quadratic assignment problem. Parallel Comput 17:443–445
59. Tam KY (1992) A simulated annealing algorithm for allocating space to manufacturing cells. Int J Prod Res 30(1):63–87
60. Tompkins JA, Reed R (1976) An applied model for the facilities design problem. Int J Prod Res 14(5):583–595
61. Vollman TE, Nugent CE, Zartler (1968) A computerized model for office layout. J Ind Eng 19(7):321–327
62. Zoller K, Adendorff K (1972) Layout planning by computer simulation. AIIE Trans 4(2):116–125