A genetic algorithm to solve process layout problem

Article *in* International Journal of Management and Decision Making · January 2014

DOI: 10.1504/IJMDM.2014.058472

3 authors:


Kaveh Khalili-Damghani
Islamic Azad University, South Tehran Branch


Ali Khatami Firouzabadi
Allameh Tabatabai University


Mohammad Diba
Young Researchers and Elite Club, South Tehran Branch, Islamic Azad University, …

A genetic algorithm to solve process layout problem

Kaveh Khalili-Damghani*

Department of Industrial Engineering,
Islamic Azad University,
South-Tehran Branch,
Imam Hossein SQ., Mazandaran St., Entezari Ave.,
11365-4435, Tehran, Iran
E-mail: kaveh.khalili@gmail.com
*Corresponding author

S.M. Ali Khatami-Firouzabadi

Department of Industrial Management,
Faculty of Management and Accounting,
Allameh Tabataba'i University,
Nezami St., Haf-Peikar St., Tehran, Iran
E-mail: smakhf@hotmail.com

Mohammad Diba

Department of Industrial Engineering,
Islamic Azad University,
Science and Research Branch,
Hormozgan, 7955158455, Iran
E-mail: mh.diba@yahoo.com

Abstract: A model based on a quadratic assignment problem (QAP) is proposed to design a process layout. The objective function of proposed model has three main parts, as: a) maximising the profit of each product; b) minimising stored keeping materials; c) minimising total layout cost. The demand requirement and production capacity are also considered as set of constraints. Eventually, as the global optimum solution is hard to find for the proposed model, a genetic algorithm (GA) is proposed to solve the model. The performance of proposed GA was compared with other well-known algorithms, i.e., simulated annealing (SA) algorithm and tabu search (TS) on several benchmark instances of QAP as well as a series of simulated random large scale instances. The comparison reveals the promising results of GA over SA, and TS.

Keywords: process layout design; quadratic assignment problem; QAP; genetic algorithm; simulated annealing; tabu search.

Reference to this paper should be made as follows: Khalili-Damghani, K., Khatami-Firouzabadi, S.M.A. and Diba, M. (2014) 'A genetic algorithm to solve process layout problem', *Int. J. Management and Decision Making*, Vol. 13, No. 1, pp.42–61.

Biographical notes: Kaveh Khalili-Damghani is an Assistant Professor of Industrial Engineering at the Azad University, South-Tehran Branch in Iran. He holds an MSc and PhD in Industrial Engineering and a PhD in Industrial Management. His research interests include soft computing, fuzzy sets and systems, meta-heuristic methods, multi-criteria decision making, data envelopment analysis, reliability optimisation, quantitative modelling of supply chain, and applied operations research. He is an Associate Editor for six international peer reviewed scientific journals such as *International Journal of Strategic Decision Sciences*, *International Journal of Enterprise Information Systems*, *International Journal of Applied Decision Sciences*, *Journal of Decision Analytics*, *Uncertain Supply Chain Management*, and has published more than 40 papers in high quality journals.

S.M. Ali Khatami-Firouzabadi is an Associate Professor in Allameh Tabataba'i University Business School, Tehran, Iran. He received his BSc and MSc degrees in Industrial Engineering from Iran University of Science and Technology. He received his PhD in Mechanical Engineering from the University of Leeds, UK. His research areas include linear programming, large-scale systems, model evaluation, multi-objective decision analysis, multi-attribute decision making, forecasting, reliability engineering in manufacturing systems and application of operations research methodologies. He has published papers in *International Journal of Computer and Industrial Engineering*, *Expert System Journal*, Euro journals and others. He has contributed papers in an international conference in manufacturing research as well.

Mohammad Diba received his BSc in Industrial Management from Allameh Tabataba'i University Business School, Tehran, Iran and MSc in Industrial Engineering from Science and Research Branch of Islamic Azad University, Hormozgan, Iran. His research interests are quality engineering, applied statistics, data mining, applied operations research, decision analysis, production planning, operations management and meta-heuristics. He has published papers in International Quality Engineering Conference and also has some teaching experience in quality management and operations management in Tehran's Technical and Vocational Training Organization. He has some teaching experience in applied statistics and operations research as well.

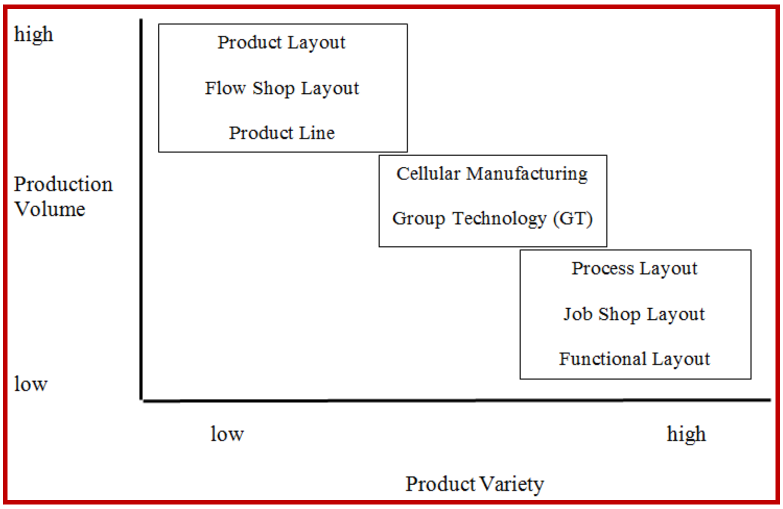
1 Introduction

Facility layout and material handling are important in modern manufacturing systems. Ten to seventy percent of total manufacturing expenses are material handling costs (Francis et al., 1992). An efficient layout can reduce these expenses at least between 10% to 30%. The design process of plant layout is influenced by many factors which can have a great impact on the factory, e.g. plant-engineering, technology, management and also economic-financial. A general definition of plant layout problem is to find the best arrangement of physical facilities to provide an efficient operation.

1.1 Layout problems

Plant layout problem normally depends on the type of production system. In a case where product volumes are high and varieties of products are low, the manufacturing process is known as a flow shop process and the layout is normally based on products. Hence, the layout is called *layout by product* or *product layout*. On the other extreme, a manufacturing plant may have a high variety of products with low production volume, where the process is known as job shop process and the layout known as *job shop layout*, or *functional layout*, or *layout by process* or *process layout* (Tompkins, 2010). Group technology (GT) or cellular manufacturing is normally applied to production systems which are placed between these two extremes with the layout known as the *cellular layout*. Figure 1 illustrates the schematic view of classic layout based on variety and volume of the products.

Figure 1 Comparing different layout types (see online version for colours)



Another type of product layout is the *fixed position layout*. In contrast to other types of layout, in this type the production equipment moves toward the product and not otherwise. This type of layout is common when the products are large in size such as the making of ships or airplanes (Russell and Taylor, 2008).

Layout affects the cost of material handling, time and throughput, and hence affects the overall efficiency and productivity of the plant. The set-up costs and material handling costs for a job in different layout types may be different. Each type of layout has a specific type of material handling equipment. For example, a product layout may transport jobs with a conveyor, while a process layout may use a robot or forklift.

In a process layout, machines with identical or similar processing capabilities are placed in a work centre (department) that can process parts from several families. The process layout is equipment-dominated. This type of layout has advantages such as high flexibility in allocating operations to alternative machines. However, because the material flows of different part families between work centres may follow complicated routes, this can result in long throughput times, high work in process (WIP) levels, and high material handling costs. So, this type of layout is concerned with finding the best arrangement of shops in plant floor. A significant influence on material handling costs can be achieved through changing the layout and place of departments.

The remaining parts of this paper are organised as follows. In the Section 2, the preliminaries of quadratic assignment problem (QAP) are revisited briefly. In Section 3, the formulation of proposed mathematical model for process layout problem is represented. In Section 4, a heuristic solution procedure and a GA are represented. Section 5 is allocated to present the results of solution procedures on several small size problems, benchmark instances, and simulated test problems. Finally, the conclusions are drawn in Section 6.

2 Quadratic assignment problem

Let us revisit the QAP briefly. QAP is a well-known problem in the facility location and layout. Fundamental formulation is due to *Koopman and Beckmann* which was first introduced to solve FLPs (Koopmans and Beckmann, 1957). QAP is a class of NP-hard problems (Sahni and Gonzalez, 1976). So, the medium and large size problems of QAP cannot be solved with exact methods. In general, QAP was proposed in order to address the problem of assigning facilities to locations. With given distances between the locations and given flows between the facilities in order to minimise the sum of product between flows and distances. Assumptions made are that the facilities have equal size (or area) and the locations to place the facilities are known in advance (Peters and Yang, 1997).

2.1 QAP formulations

There are several well-known formulations for QAP problem. One of them is Koopmans-Beckman formulation. The following QAP formulation initially was proposed by Koopmans and Beckmann (1957).

$$Min\ f(x) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{ij} \times d_{kl} \times x_{ik} \times x_{jl} \tag{1}$$

s.t.

$$\sum_{i=1}^{n} x_{ij} = 1, \quad 1 \le j \le n \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad 1 \le i \le n \tag{3}$$

$$x_{ij} \in \{0, 1\} \quad 1 \le i,\ j \le n \tag{4}$$

where n is the number of facilities (locations), f_{ij} is flow of material between facility i and facility j, x_{ik} is a binary decision variable and is 1 if department j is assigned to location l and 0 otherwise. x_{jl} is a binary decision variable and is 1 if department j is assigned to location l and 0 otherwise.

Objective function (1) minimises the sum of the material handling and assignment costs. Constraints set (2) ensure that only one facility is assigned to each location. Constraints set (3) guarantee that each location is occupied by only one facility. Constraints set (4) define the domain of decision variables.

A more general form of QAP was proposed by Lawler (1963) as follows:

$$\text{Min } f(x) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{ij} \times c_{ijkl} \times x_{ik} \times x_{jl} + \sum_{i}^{n}\sum_{k}^{n} a_{ik} \times x_{ik} \tag{5}$$

$$\sum_{i=1}^{n} x_{ik} = 1, \quad \forall k \tag{6}$$

$$\sum_{k=1}^{n} x_{ik} = 1, \quad \forall i \tag{7}$$

$$x_{ik}, x_{jl} \in \{0, 1\}, \quad \forall i, j, k, l \tag{8}$$

where n is the number of facilities (locations), f_{ij} is flow of material between facility i and facility j, c_{ijkl} is the cost of assigning facility i to location k and facility j to location l, simultaneously, a_{ik} is fixed cost of assigning department i to location k, x_{ik} is a binary decision variable and is 1 if department i is assigned to location k and 0 otherwise, x_{jl} is a binary decision variable and is 1 if department j is assigned to location l and 0 otherwise. Objective function (5) minimises the sum of the material handling and assignment costs. Constraint set (6) ensures that only one facility is assigned to each location. Constraint set (7) guarantees that each location is occupied by only one facility. It is notable that the set of constraint (8) promotes the binary state of all decision variables.

The QAP formulation that uses the permutation concept was first introduced in Hillier and Connors (1966). Suppose S_n be the set of all permutations with n elements and $\pi \in S_n$. Consider f_{ij} the flows between facilities i and j and $d_{\pi(i)\pi(j)}$ the distances between locations $\pi(i)$ and $\pi(j)$. If each permutation π represents an allocation of facilities to locations, the problem expression becomes:

$$\min_{x \in S_n} \sum_{i,j=1}^{n} f_{ij} \times d_{\pi(i)\pi(j)}. \tag{9}$$

This formulation is equivalent to the first one presented in (1)–(4), since the constraints (2) and (3) define permutation matrices $X = [x_{ij}]$ related to S_n elements, as in (9), where, for all $1 \le i, j \le n$,

$$x_{ij} = \begin{cases} 1, & \text{if } \pi(i) = j; \\ 0, & \text{if } \pi(i) \ne j, \end{cases} \tag{10}$$

Another equivalent formulation is defined by trace function (the sum of the matrix main diagonal elements) in order to determine the lower bounds of the cost of QAP. This approach can be stated as:

$$\min_{x \in S_n} tr\left(F.X.D.X^t\right) \tag{11}$$

This formulation was first introduced by Edwards (1980). It may be used for a flexible algebraic manipulation of the problem data (Commander, 2005). Other forms of QAP formulations can be found in Loiola et al. (2007).

2.2 Application of QAP on production and services problem

The first occurrence of the QAP as a mathematical model of assigning a set of activities to a set of locations was in the context of facility location problems which still remain one of its major applications (Koopmans and Beckmann, 1957). Nowadays, QAP showed that has a large variety of other applications including such areas as wiring problems in electronics (Steinberg, 1961), scheduling (Geoffrion and Graves, 1976), parallel and distributed computing (Bokhari, 1981), balancing of turbine runners (Laporte and Mercure, 1988), archeology (Grötschel and Wakabayashi, 1989), developing a decision framework for assigning a new facility (police posts, supermarkets, schools) in order to serve a given set of clients (Francis et al., 1992), computer manufacturing (Jünger et al., 1994) and statistical data analysis (Zhao et al., 1998). Recently, the QAP formulation has also been used in website structure improvement (Qahri Saremi et al., 2008). The more precise description of other QAP applications can be found in Loiola et al. (2007).

2.3 Methods used to solve QAPs

There are two major approaches to solve QAPs including optimal and heuristic solution procedures. The first approaches are *exact methods* and the second approaches are heuristic and *metaheuristic methods*. Generally, QAP instances which are larger than 20 cannot be solved using optimal solution procedure in a reasonable CPU time (Li et al., 1994a; Clausen and Perregaard, 1997).

2.3.1 Exact algorithms

There are three main approaches, including *dynamic programming*, *cutting plane techniques*, and *branch and bound procedures*, that are able to find the global optimal solution for a small size instances of QAP. Researches show that the branch and bound is the most successful method for solving QAP instances. Even with fast computers, due to the complexity of the QAP, instances more than $n = 15$ are very hard to solve optimally (Pardalos and Wolkowicz, 1994). Application of dynamic programming to solve QAPs can be seen in Christofides and Benavent (1989) or Urban (1998). Cutting plane was first used to solve QAPs by Bazaraa and Sherali (1982). Gilmore (1962) first solved a QAP of size $n = 8$ with branch and bound technique. Recently, Duffuaa and Fedjki (2012) applied branch and bound technique on several facilities based on QAP.

2.3.2 Heuristic approaches

These approaches produce high-quality solution in a reasonable CPU time. There are three main types of heuristic approaches to QAP as listed in chronological order below (Commander, 2005):

- *Construction approaches:* Construction approaches are the simplest heuristic approaches to solve QAP, from a conceptual and an implementation point of view. These methods are probably the oldest ones that were proposed by Gilmore in the early ‘60s (Gilmore, 1962). Computerised relative allocation of facilities technique (CRAFT) is a famous heuristic to solve QAP. CRAFT was first introduced by Armour and Buffa (1963). Another construction method which yields relatively good results in comparison with other construction methods is proposed by Müller-Merbach (1970).

- *Limited enumeration approaches:* Limited enumeration heuristics are similar to exact methods such as branch and bound approaches or cutting plane algorithms. Heuristics of this type have been developed by Bazaraa and Sherali (1982) and Burkard and Bönniger (1983).
- *Improvement approaches:* Improvement approaches are the most studied class of heuristics for solving QAPs. They are consists of local search methods, that work by starting from an initial basic feasible solution and then attempting to improve it. The local search iteratively seeks a better solution in the neighbourhood of the current solution, terminating when no better solution exists within that neighbourhood (Commander, 2005).

2.3.3 Metaheuristic approaches

Metaheuristics are the most popular and efficient class of solution methods to solve QAPs. These group of methods unlike the other heuristics are not designed to solve a specific problem. They are general problem solving methods. The most popular metaheuristics that were used to solve QAPs are *simulated annealing* (SA), *tabu search* (TS), *genetic algorithms* (GAs), and *greedy randomised adaptive search procedure* (GRASP).

SA has its name from the physical and thermo-dynamic process which it imitates from. This process, called annealing moves high energy particles to lower energy states with the lowering of the temperature. In the initial state of the solving a problem, the algorithm is capable of moving to a worse solution regarding a specific probability. However, with each iteration the algorithm goes towards a better solution (Burkard and Rendl, 1984).

TS works by starting with an initial basic feasible solution and then attempting to improve it. It has an updated list of the best solutions that have been found in the search process. Each solution receives a priority value that helps the algorithm go on (Skorin-Kapov, 1990).

GA is based on Darwin’s theory of natural selection. GAs store a set of solutions and then work to replace these solutions with better ones based on fitness function (Brown et al., 1989). Modified GAs were used to solve QAPs in recent years (Rajmohan and Shahabudeen, 2008).

Other metaheuristics like *ant colony algorithm* (ACO), *particle swarm optimisation* (PSO), *scatter search* (SS), GRASP, *variable neighbourhood search* (VNS) are also used to solve QAPs and are summarised in Table 1. Gambardella et al. (1999), Hardin and Usher (2005), Cung et al. (1997), Li et al. (1994b), Raza and Al-Turki (2010), Ongsakul et al. (2011), See and Wong (2008) and Montoya-Torres et al. (2010).

Table 1 Summery of application of metaheuristics to solve QAPs

Meta-heuristic name	*Founder(s)*	*First application to QAP*
SA algorithm	Kirkpatrick et al. (1983)	Burkard and Rendl (1984)
TS algorithm	Fred Glover (1986)	Skorin-Kapov (1990)
GA	John Holland (1975)	Brown et al. (1989)
ACO	Dorigo et al. (1991)	Gambardella et al. (1999)
PSO	Kennedy, J. and Eberhart, R. (1995)	Hardin and Usher (2005)
SS	Fred Glover (1977)	Cung et al. (1997)
GRASP	Feo and Resende (1989)	Li et al. (1994b)

*Hybrid algorithm*s have had numerous successful implementations in solving combinatorial optimisation problems. *Hybrid algorithm*s were also applied on QAPs (See and Wong, 2008). See and Wong (2008) used *GenANT*, combination of ACO and GA to solve QAP. *Parameter tuning* of meta-heuristics using *design of experiments* (DOEs) or analysis of variance (ANOVA) is one of the new promising areas of research (See and Wong, 2008). Another increasing research area is using metaheuristics for solving multiple objective problems like MOPSO or NSGAII (Ongsakul et al., 2011; Salmasnia et al., 2013).

3 Proposed model

In discussing the importance of minimising interruptions on flow paths in traditional facility layout design, Tompkins (2010) identified three principles that result in effective workflow:

a minimising flow

b minimising the costs of flow

c maximising directed flow paths.

The first principle is addressed by using the work simplification approach; the second is accomplished by classical facility layout analyses (e.g., the QAP). Concerning the third principle, needs a directed and uninterrupted flow path that has fewer intersections with other paths.

In this paper, considering the second and third principle, a mathematical model as well as two solution procedures (i.e., a heuristic method and a GA) for process layout problem are presented. We compare the proposed the performance of GA algorithm with SA and TS on small size problem, several benchmark instances, and simulated examples.

3.1 Model formulation

In this section a mathematical model is proposed for process layout problem based on QAP. The proposed model is a full customisation of model proposed by Jaramillo and McKendall (2010) in which they presented the generalised machine layout problem (GMALP). GMALP was a generalisation of the integrated machine and layout problem.

GMALP integrated the QAP with a multi-commodity flow problem. Jaramillo and McKendall (2010) developed a TS algorithm to solve GMALP.

Consider a factory with N candidate locations and M departments. Figure 2 illustrate the schematic view of plant floor.

Figure 2 Scheme of the plant floor (see online version for colours)

Location N	…..	…..
…..	…..	Location 1

Suppose P products are producing within the factory. Each of product needs to go to Op departments for Op operations. To assigning shops to the plant floor on the basis of QAP, for maximising the total profit (TP) of the factory, the following notations are assumed:

- *Sets and indices:*

n	number of locations
m	number of departments (shops)
o	number of processes
p	number of products
$i, j = 1, 2, \ldots, n$	index of locations (facilities)
$k, l = 1, 2, \ldots, m$	index of departments (shops)

- *Parameters:*

P_p	sales price of product p
C_p	cost of moving every single unit of product p
d_{ij}	distance between facility i and facility j
w_{kl}	exchange between department k and department l
C_k	capacity of working hours of department k in each month
D_p	demand of product p
O_p	number of processes to complete product p
t_{pk}	process time of product p in department k.

- *Decision variables:*

X_p	number of product p produced

$$Y_{ki} = \begin{cases} 1, & \text{if department } k \text{ is assigned to location } i, \\ 0, & \text{otherwise.} \end{cases}$$

$$R_{pko} = \begin{cases} 1, & \text{if product } p \text{ meets department } k \text{ for process } o, \\ 0, & \text{otherwise.} \end{cases}$$

Based on the above notations the following model is proposed to solve the process layout problem:

$$\text{Max } TP = \sum_{k=1}^{m} \sum_{l=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{p=1}^{P} \left(P_p - d_{ij} \times \left(c_p + w_{kl} \right) \right) \times X_p \tag{12}$$

s.t.

$$\sum_{k=1}^{m} Y_{ki} = 1 \quad \forall i \tag{13}$$

$$\sum_{i=1}^{n} Y_{ki} = 1 \quad \forall k \tag{14}$$

$$X_p \le D_p \tag{15}$$

$$\sum_{p=1}^{P}\sum_{o=1}^{O}\sum_{k=1}^{m}\sum_{i=1}^{n} R_{pko} \times t_{pk} \times X_p \times Y_{ki} \le C_k \tag{16}$$

$$\sum_{p=1}^{P}\sum_{o=2}^{O_p-1}\sum_{k=1}^{m}\sum_{i=1}^{n} R_{p(o-1)k} \times X_p \times Y_{ki} = \sum_{p=1}^{P}\sum_{o=2}^{O_p-1}\sum_{k=1}^{m}\sum_{i=1}^{n} R_{p(o+1)k} \times X_p \times Y_{ki} \tag{17}$$

$$X_p \ge 0, \quad \forall p \tag{18}$$

$$Y_{ki} \in \{0, 1\}, \quad \forall k, i \tag{19}$$

$$R_{pko} \in \{0, 1\}, \quad \forall p, k, o \tag{20}$$

Objective function (12) maximises the sum of the TP considering the material handling and assignment costs. Constraint set (13) ensures that each department is allocated to just one location. Constraint set (14) guarantees that each location is occupied just by one department. Constraint set (15) guarantees than the production is at most equal to demand. Constraint set (16) ensures that the capacities of the departments are not exceeded. Constraint set (17) guarantees that the products entering a department is equal to products going out from it. Constraints (18)–(20) define the domains of decision variables.

Model (12)–(20) is a non-linear mixed integer programming which is hard to solve optimally. So, in the following sections proper solution procedures are proposed.

4 Solution procedures

In this section, two solution procedures, i.e., a heuristic method and a GA, are proposed to solve model (12)–(20).

4.1 Heuristic solution procedure

To solve the small size instances of problem (12)–(20) there are several conventional heuristic methods such as *spiral method*, *travel charting method*, *straight line method*, *schematic diagram method*, and *demand sequence method*. All of them try to minimise the total cost by bringing the departments with higher relation near each other. We have used *travel charting method* by Armour and Buffa (1963) for solving small size of instances of problem (12)–(20). Travel charting method is a heuristic method uses *distance* and *cost matrix* for computing transportation cost, the area needed for each department and the production sequence of each product. As the method by Armour and Buffa (1963) is well-know and there are many associated literature, we do not mention the details of it for sake of brevity. Although travel charting method has some weaknesses as

a it is inefficient for large scale problems

b it does not suggest systematic method for improvement

c its final solution depends on quality of initial solution.

4.2 Proposed GA structure for solving proposed model

The GA structure for solving model (12)–(20) is expressed in Figure 3 and will be explained in upcoming sections.

4.2.1 Encoding

Encoding is the first step for using GA. Permutation presentation of chromosome is used here. Each chromosome is filled semi-randomly to make an initial solution for the problem. As mentioned, in the proposed model there are n locations and m departments. Figure 4 presents the chromosome structure of *QAP* for the m departments. Each square shows a locus or location that is candidate position for departments and are numbered, respectively. Each circle shows an allele that indicates a department. Locus and allele together make gene and a set of genes makes a chromosome.

The allele i at locus j in the chromosome indicates that facility i is assigned to location j. For example, the figure shows the department 4 at the first location, 3 at the second location and … department m at the location n.

4.2.2 Initial solution

Initial population is made with generating random permutations. For m departments, there are $M!$ of possible permutations.

4.2.3 Selection procedure

The number of chromosomes required to complete next generation are determined using selection operator. *Roulette wheel* selection method is used at first iterations of the proposed GA. Then, *tournament* selection method is also supplied at middle iterations. The algorithm randomly decided to select based on *roulette wheel* or *tournament.* Finally, just *tournament* selection method with size of 2 is used at last iterations in order to increase the speed of the proposed *GA*.

4.2.4 Crossover

Single point crossover is used in the proposed GA. It is notable that single point crossover may cause infeasibility, so further modifications should be considered. Figure 5 represents the crossover and the associated modification for a five-gene chromosome. As shown in Figure 5, if the cross point be the third gene, the reproduced solutions have repetitive genes. For solving this problem, the old repetitive gene in each child is replaced with the repetitive gene in the other child. The crossover rate (percent of parents participate in crossover operation) is reduced while GA iterates.

Figure 3 The GA structure for solving proposed model (see online version for colours)

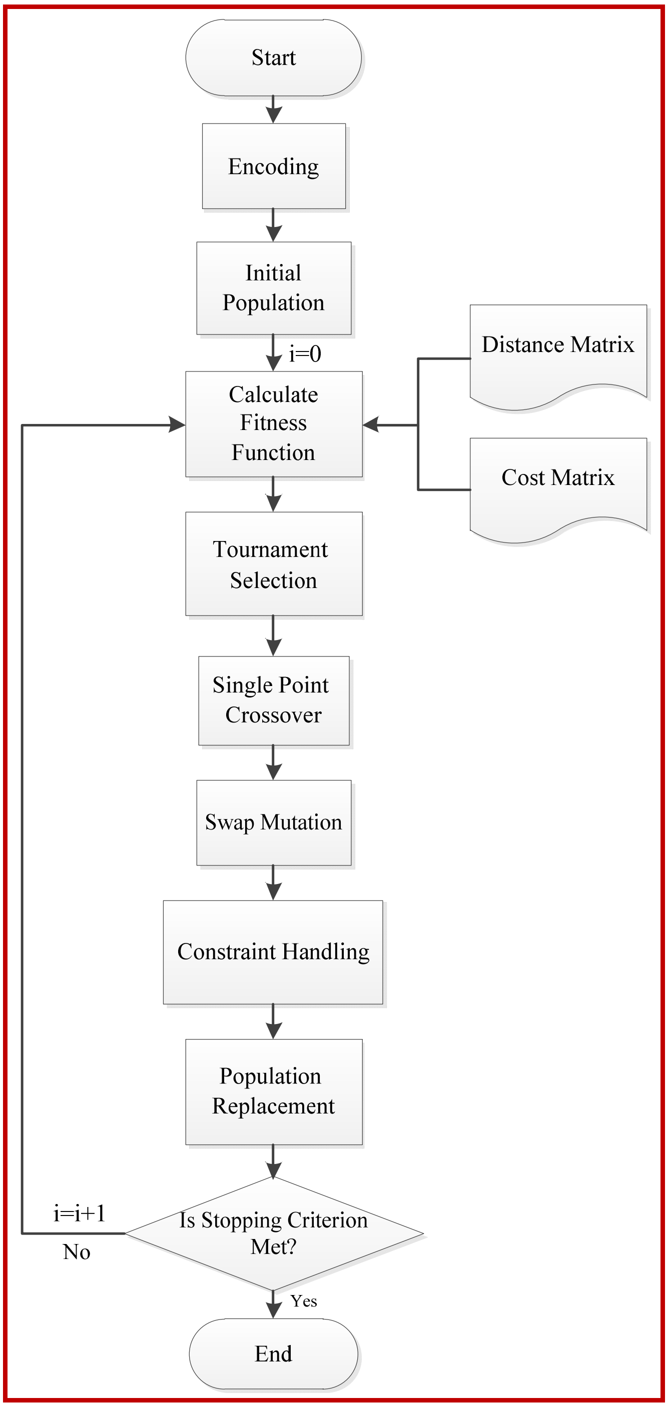


Figure 4 The chromosome structure (see online version for colours)

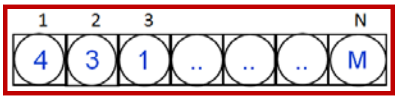
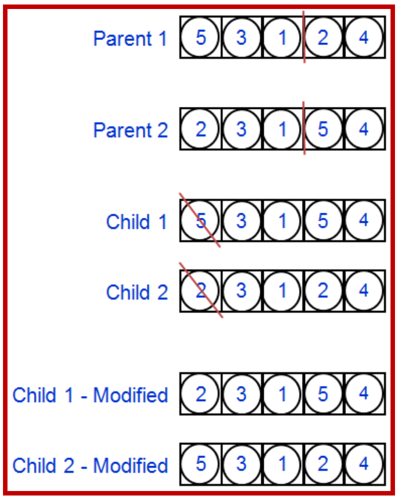


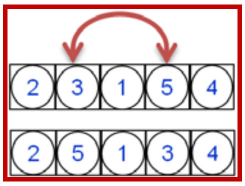
Figure 5 Single-point cross over representation (see online version for colours)



4.2.5 *Mutation*

After the crossover, the mutation is implemented. *Swap* and *reversion* mutation types are used together at early iterations of the proposed GA. At final iteration only *Swap* is used to increase the speed the algorithm. As shown in Figure 6, two genes are randomly selected from selected chromosome and then their locations are substituted.

Figure 6 Swap mutation representation (see online version for colours)



4.2.6 *Constraint handling*

Hybrid strategy is used for constraint handling in this paper. Some of the constraints of proposed model (12)–(20) are tried to meet in the generation of chromosomes, and other constraint are supported by *death penalty strategy* in which, infeasible children are eliminated (Michalewicz, 1995).

4.2.7 Stopping criterion

The GA will stop after a certain number of iterations, called *MaxIte*.

5 Experimental experiences

In this section, a heuristic method and proposed GA are applied to solve a small size instance, and several benchmark instances of QAP, and some simulated test instances. The performance of proposed GA is compared with SA and TS.

5.1 Solving small size instance using heuristic procedure

According to the model (12)–(20), suppose three different products (jobs) are producing and each of them has its own sequence of operations which has to be completed on these departments. Each location is 7 × 7 m; entrance to plant floor is from location 1 and the exit from location 6. Transportation cost is 0.5$ per metre. Figure 7 illustrate the schematic view of plant floor for this instance.

Figure 7 Scheme of the plant floor (see online version for colours)

Location 1	Location 2	Location 3
Location 4	Location 5	Location 6

Table 2 represents the sales price of each product. Table 3 represents the distance matrix between existing locations. Table 4 represents the production sequence of each product.

Table 2 Sales price of products

Product	*Price $*
1	100
2	70
3	55

Table 3 The distance matrix between locations

Distance	*Location 1*	*Location 2*	*Location 3*	*Location 4*	*Location 5*	*Location 6*
Location 1	0	7	7	14	14	21
Location 2	7	0	14	7	21	14
Location 3	7	14	0	7	7	14
Location 4	14	7	7	0	14	7
Location 5	14	21	7	14	0	7
Location 6	21	14	14	7	7	0

Table 4 Production sequence of each product

Product	*Dep.1*	*Dep.2*	*Dep.3*	*Dep.4*	*Dep.5*	*Demand*	*Product ratio*
1	1	2	3	4	5	150	0.375
2	3	1	4	6	-	130	0.325
3	5	2	3	6	-	120	0.3

The solution space of this small size instance has 6! = 720 possible solutions. As mentioned, *travel charting method* proposed by Armour and Buffa (1963) is used to solve this small size instance. Table 5 represents flow matrix between departments.

Table 5 Flow matrix between departments

Distance	*Dep.1*	*Dep.2*	*Dep.3*	*Dep.4*	*Dep.5*
Dep. 1	0	56.25		42.25	
Dep. 2		0	92.25		
Dep. 3	42.25		0	56.25	
Dep. 4				0	56.25
Dep. 5		36			0

A simple initial solution for this problem is: location (i) = department j. For example, $L(1) = 1$, $L(2) = 3$, $L(3) = 5$, $L(4) = 2$, $L(5) = 4$, and $L(6) = 6$. The total cost for this assignment is as follows:

$$\begin{aligned} TP &= 150 \times 0.5 \times (7 + 14 + 7 + 14) + 130 \times 0.5 \times (7 + 7 + 14 + 7) \\ &\quad + 120 \times 0.5 \times (14 + 21 + 14 + 14) = 9,205. \end{aligned}$$

For this situation product of distance and flow matrices is $P = 4,805.5$. For improvement, the departments should be exchanged till the largest numbers put near the main diagonal. The final solution is achieved as $L(1) = 1$, $L(2) = 2$, $L(3) = 3$, $L(4) = 4$, $L(5) = 5$, $L(6) = 6$, and $P = 4,345.25$. The total cost of this assignment is \$7,910 that is lower than initial solution.

$$\begin{aligned} TP &= 150 \times 0.5 \times (7 + 7 + 14 + 7) + 130 \times 0.5 \times (14 + 14 + 7 + 14) \\ &\quad + 120 \times 0.5 \times (14 + 7 + 7 + 7) = 7,910. \end{aligned}$$

As mentioned, heuristic procedures [including method by Armour and Buffa (1963)] are able to solve small size instances of QAP problems. We solved this small size problem in order to illustrate the mechanism of problem and making better sense for readers. In the following sub-sections, we apply our proposed GA algorithm on medium and large size benchmark instances of QAP and also on several simulated large scale test instances.

5.2 *Solving benchmark QAP instances using proposed GA*

The proposed GA has been coded in MATLAB (R2011a) and implemented on a DELL Inspiron laptop equipped with an Intel Core i7-2670 QM processor, 8 GB of memory and windows 7 64-bit operating system. To evaluate the performance of the proposed GA, we selected some QAP instances from the QAP library (Burkard et al., 1997). The QAP library has several instance with their best obtained solution that are all minimisation.

Tournament selection is used for selection procedure, single point crossover is used for crossover and the swap is used for mutation. All runs were performed with *initial population size* of 60 for problems up to size 20 and *initial population size* of 100 for the larger ones, *crossover rate* of 0.7–0.8 and *a rate* of 0.05–0.08. Table 6 represents the results of proposed GA on benchmark instance of QAP. The results of time and cost are average of 50 independent runs of the proposed GA. The obtained results show that the procedure is good for the problems of size 25 and higher.

Table 6 Solutions for benchmark problems

Problem	*Size*	*Iterations*	*Average cost*	*Average time (sec)*	*Optimum cost*	*GAP %*
Nug12	12	100	605	1.0462	578	4.67
Nug15	15	100	1200	1.6839	1150	4.35
Nug25	25	200	3850	6.4054	3744	2.83
Nug28	28	300	5350	9.3051	5166	3.56
Kra32	32	400	95270	14.1337	88700	7.41

5.3 Solving extra simulated instances using proposed GA

For showing the reliability of the proposed GA, seven extra test problems were simulated and solved with proposed GA. A summary of properties of test problems is given in Table 7.

Table 7 Simulation problems specifications

Problem	*Size*	*No. of products*	*Max no. of operations on a product*
Problem #1	12	5	11
Problem #2	26	5	16
Problem #3	15	3	11
Problem #4	18	3	13
Problem #5	24	3	16
Problem #6	16	3	12
Problem #7	28	3	16

In Table 7, size column shows the number of locations/departments, the next column shows the number of product and the last column shows the number of operations. The proposed GA was used to solve the simulated problems. The results were compared to *SA* and *TS* shown in Table 8 that shows the mean of objective function and mean of CPU time of each algorithm for simulated problems. The results are mean of 50 independent runs of each algorithm.

Figure 8 and Figure 9 plot the objective function and CPU time of each algorithm for test problems, respectively. As it is clear, our proposed GA outperforms SA, and TS algorithms on the profit objective function for all seven test problems. Although, our proposed GA is slower than SA and TS.

Table 8 Comparison of average objective function and solution time for simulated problems

Problem	*Average objective function*			*Average solution time*		
	GA	*SA*	*TS*	*GA*	*SA*	*TS*
Problem #1	43,900	41,135	39,536	0.86295722	0.0515029	0.075874625
Problem #2	43,540	35,200	23,531	7.51862774	0.1408183	0.16573836
Problem #3	30,150	24,832	22,413	1.71980116	0.0626733	0.1101217
Problem #4	29,930	22,806	16,785	2.098433	0.07496896	0.11955392
Problem #5	29,985	20,579	11,965	3.0729518	0.12688734	0.15265104
Problem #6	28,170	21,905	19,365	2.0674904	0.06141186	0.05322348
Problem #7	42,400	28,590	18,060	4.48931984	0.1640391	0.19868458

Figure 8 Mean of profit for simulated problems (see online version for colours)

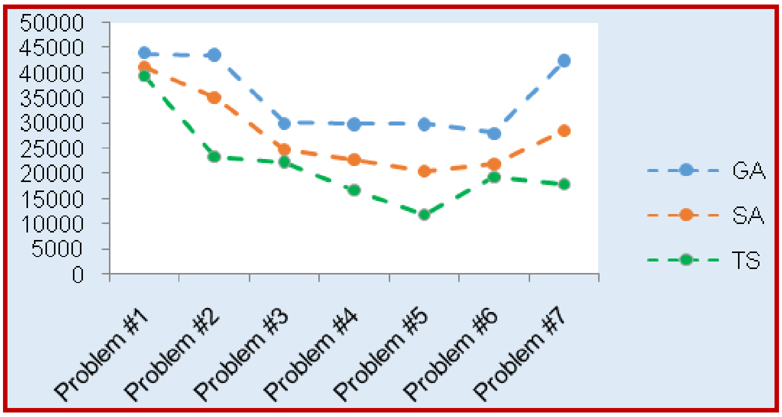
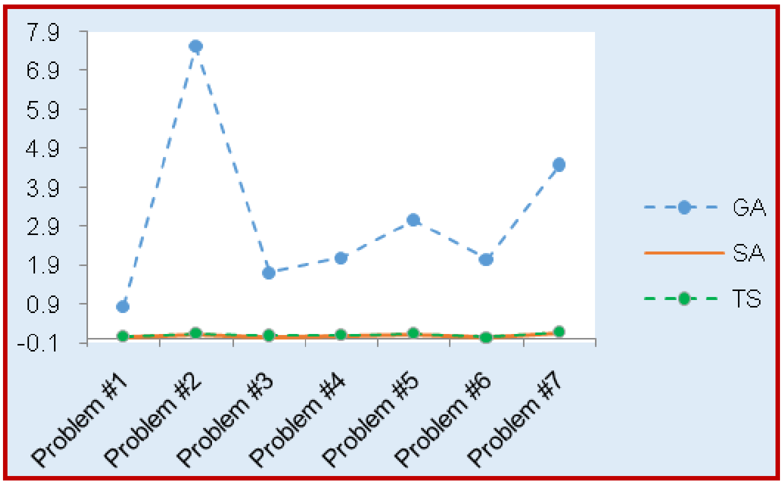


Figure 9 Mean of solution time for simulated problems (see online version for colours)



5.4 Statistical analysis of results

Although Figure 8 represents better performance of proposed GA for objective function, but some statistical analysis were organised to test whether the proposed GA was better than SA and TS meaningfully. Statistical analysis for comparing the objective functions with either *Kolmogorov-Smirnov* and *Anderson-Darling* tests showed that the obtained data even for 50 runs of each category are not normal (Yousefpoor et al., 2013). Also *Levene* test showed that groups does not have equal variances (Parnell et al., 2006), so the *Kruskal-Wallis* test was used for comparing the objective functions (Agasisti and Dal Bianco, 2009). The *Kruskal-Wallis* test with p-value of 0.000, for all the problems showed that according to objective function values there were significant difference between algorithms. Proposed GA dominates SA and SA dominates TS.

6 Conclusions and directions for future research

In this paper, facility location problems for process layout type design were studied. A mathematical model based on QAP was proposed to design the process layout problem. A heuristic procedure and a GA were proposed to solve several benchmark instances of the problem. Several test problems were also generated and solved using proposed GA, and compared with SA and TS algorithm. The results showed that the proposed mathematical model and its associated GA algorithm were practical even for large scale problems. The statistical analysis showed that the proposed procedure outperform SA and TS algorithms meaningfully.

Further reduction in the solution time of the model might be possible adopting and enhancing other metaheuristics. Also *statistical DOE* can be used to *parameter tuning* in metaheuristics. Another area for future research is the expansion of the mathematical model by adding more real life constraints. Furthermore, other types of distance matrix such as *Euclidean* or *squared* Euclidean distance can be used. Another area for further research is using multi-objective model for the problem and solving it using multi-objective metaheuristic algorithms.

References

Agasisti, T. and Dal Bianco, A. (2009) 'Measuring efficiency of higher education institutions', *International Journal of Management and Decision Making*, Vol. 10, Nos. 5–6, pp.443–465.

Armour, G.C. and Buffa, E.S. (1963) 'A heuristic algorithm and simulation approach to relative location of facilities', *Management Science*, Vol. 9, No. 2, pp.294–309.

Bazaraa, M.S. and Sherali, H.D. (1982) 'On the use of exact and heuristic cutting plane methods for the quadratic assignment problem', *Journal of the Operational Research Society*, Vol. 33, No. 11, pp.991–1003.

Bokhari, S.H. (1981) 'On the mapping problem', *IEEE Transactions on Computers*, Vol. C-30, No. 3, pp.207–214.

Brown, D.E., Huntley, C.L. and Spillane, A.R. (1989) 'A parallel genetic heuristic for the quadratic assignment problem', *Proceedings of the 3rd International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers Inc., pp.406–415.

Burkard, R.E. and Bönniger, T. (1983) 'A heuristic for quadratic Boolean programs with applications to quadratic assignment problems', *European Journal of Operational Research*, Vol. 13, No. 4, pp.374–386.

Burkard, R.E. and Rendl, F. (1984) 'A thermodynamically motivated simulation procedure for combinatorial optimization problems', *European Journal of Operational Research*, Vol. 17, No. 2, pp.169–174.

Burkard, R.E., Karisch, S.E. and Rendl, F. (1997) 'QAPLIB – a quadratic assignment problem library', *Journal of Global Optimization*, Vol. 10, No. 4, pp.391–403.

Christofides, N. and Benavent, E. (1989) 'An exact algorithm for the quadratic assignment problem on a tree', *Operations Research*, Vol. 37, No. 5, pp.760–768.

Clausen, J. and Perregaard, M. (1997) 'Solving large quadratic assignment problems in parallel', *Computational Optimization and Applications*, Vol. 8, No. 2, pp.111–127.

Commander, C.W. (2005) 'A survey of the quadratic assignment problem with applications', *Morehead Electronic Journal of Applicable Mathematic*, Vol. 4, No. 4, pp.1–15.

Cung, V-D., Mautor, T., Michelon, P. and Tavares, A. (1997) 'A scatter search based approach for the quadratic assignment problem', *IEEE International Conference on Evolutionary Computation, 1997*, IEEE, pp.165–169.

Duffuaa, S.O. and Fedjki, C.A. (2012) 'General forms of the quadratic assignment problem', *International Journal of Operational Research*, Vol. 13, No. 2, pp.185–199.

Edwards, C. (1980) 'A branch and bound algorithm for the Koopmans-Beckmann quadratic assignment problem', *Combinatorial Optimization II*, Vol. 13, No. 1, pp.35–52, Springer.

Francis, R.L., Mcginnis, L. and White, J.A. (1992) *Facility Layout and Location: An Analytical Approach*, Prentice-Hall, Englewood Cliffs, NJ.

Gambardella, L.M., Taillard, E. and Dorigo, M. (1999) 'Ant colonies for the quadratic assignment problem', *Journal of the Operational Research Society*, Vol. 50, No. 2, pp.167–176.

Geoffrion, A. and Graves, G. (1976) 'Scheduling parallel production lines with changeover costs: practical application of a quadratic assignment/LP approach', *Operations Research,* Vol. 24, No. 4, pp.595–610.

Gilmore, P.C. (1962) 'Optimal and suboptimal algorithms for the quadratic assignment problem', *Journal of the Society for Industrial & Applied Mathematics*, Vol. 10, No. 2, pp.305–313.

Grötschel, M. and Wakabayashi, Y. (1989) 'A cutting plane algorithm for a clustering problem', *Mathematical Programming*, Vol. 45, Nos. 1–3, pp.59–96.

Hardin, C. and Usher, J.S. (2005) 'Facility layout using swarm intelligence', *Proceedings of the Swarm Intelligence Symposium, 2005, SIS 2005*, IEEE, pp.424–427.

Hillier, F.S. and Connors, M.M. (1966) 'Quadratic assignment problem algorithms and the location of indivisible facilities', *Management Science*, Vol. 13, No. 1, pp.42–57.

Jaramillo, J.R. and McKendall Jr., A. (2010) 'The generalised machine layout problem', *International Journal of Production Research*, Vol. 48, No. 16, pp.4845–4859.

Jünger, M., Martin, A., Reinelt, G. and Weismantel, R. (1994) 'Quadratic 0/1 optimization and a decomposition approach for the placement of electronic circuits', *Mathematical Programming*, Vol. 63, Nos. 1–3, pp.257–279.

Koopmans, T.C. and Beckmann, M. (1957) 'Assignment problems and the location of economic activities', *Econometrica: Journal of the Econometric Society*, Vol. 25, No. 1, pp.53–76.

Laporte, G. and Mercure, H. (1988) 'Balancing hydraulic turbine runners: a quadratic assignment problem', *European Journal of Operational Research*, Vol. 35, No. 3, pp.378–381.

Lawler, E.L. (1963) 'The quadratic assignment problem', *Management Science*, Vol. 9, No. 4, pp.586–599.

Li, Y., Pardalos, P., Ramakrishnan, K. and Resende, M. (1994a) 'Lower bounds for the quadratic assignment problem', *Annals of Operations Research*, Vol. 50, No. 1, pp.387–410.

Li, Y., Pardalos, P.M. and Resende, M.G.C. (1994b) 'A greedy randomized adaptive search procedure for the quadratic assignment problem', *Quadratic Assignment and Related Problems, DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, pp.237–261.

Loiola, E.M., de Abreu, N.M.M., Boaventura-Netto, P.O., Hahn, P. and Querido, T. (2007) 'A survey for the quadratic assignment problem', *European Journal of Operational Research*, Vol. 176, No. 2, pp.657–690.

Michalewicz, Z. (1995) 'A survey of constraint handling techniques in evolutionary computation methods', *Evolutionary Programming*, pp.135–155.

Montoya-Torres, J.R., Aponte, A., Rosas, P. and Caballero-Villalobos, J.P. (2010) 'Applying GRASP metaheuristic to solve the single-item two-echelon uncapacitated facility location problem', *International Journal of Applied Decision Sciences*, Vol. 3, No. 4, pp.297–310.

Müller-Merbach, H. (1970) *Optimale Reihenfolgen*, Springer-Verlag, Berlin.

Ongsakul, W., Saksinchai, A. and Boonchuay, C. (2011) 'Multi-objective bidding strategy for GenCo using non-dominated sorting particle swarm optimisation', *International Journal of Applied Decision Sciences*, Vol. 4, No. 4, pp.305–323.

Pardalos, P.P.M. and Wolkowicz, H. (1994) 'Quadratic assignment and related problems', *Dimacs Workshop*, 20–21 May 1993, American Mathematical Soc.

Parnell, J.A., O'Regan, N. and Ghobadian, A. (2006) 'Measuring performance in competitive strategy research', *International Journal of Management and Decision Making*, Vol. 7, No. 4, pp.408–417.

Qahri Saremi, H., Abedin, B. and Meimand Kermani, A. (2008) 'Website structure improvement: Quadratic assignment problem approach and ant colony metaheuristic technique', *Applied Mathematics and Computation*, Vol. 195, No. 1, pp.285–298.

Rajmohan, M. and Shahabudeen, P. (2008) 'Heuristic for solving routing problem in supply chain management', *International Journal of Applied Decision Sciences*, Vol. 1, No. 2, pp.153–178.

Raza, S.A. and Al-Turki, U.M. (2010) 'A new policy for scheduling maintenance on a single machine', *International Journal of Applied Decision Sciences*, Vol. 3, No. 2, pp.132–150.

Russell, R.S. and Taylor III, B.W. (2008) *Operations Management: Creating Value Along the Supply Chain*, 6th ed., John Wiley & Sons.

Sahni, S. and Gonzalez, T. (1976) 'P-complete approximation problems', *Journal of the ACM (JACM)*, Vol. 23, No. 3, pp.555–565.

Salmasnia, A., Moeini, A., Mokhtari, H. and Mohebbi, C. (2013) 'A robust posterior preference decision–making approach to multiple response process design', *International Journal of Applied Decision Sciences*, Vol. 6, No. 2, pp.186–207.

See, P.C. and Wong, K.Y. (2008) 'Using statistical design of experiment to decide the effective parameter values for a new hybrid ant colony algorithm', *International Journal of Applied Decision Sciences*, Vol. 1, No. 3, pp.282–304.

Skorin-Kapov, J. (1990) 'Tabu search applied to the quadratic assignment problem', *ORSA Journal on Computing*, Vol. 2, No. 1, pp.33–45.

Steinberg, L. (1961) 'The backboard wiring problem: a placement algorithm', *Siam Review*, Vol. 3, No. 1, pp.37–50.

Tompkins, J.A. (2010) *Facilities Planning*, Wiley, CA, USA.

Urban, T.L. (1998) 'Solution procedures for the dynamic facility layout problem', *Annals of Operations Research*, Vol. 76, No. 1, pp.323–342.

Peters, B.A. and Yang, T. (1997) 'Integrated facility layout and material handling system design in semiconductor fabrication facilities', *Semiconductor Manufacturing, IEEE Transactions on*, Vol. 10, No. 3, pp.360–369.

Yousefpoor, N., Shirani, A. and Nasehifar, V. (2013) 'Prioritising effective elements on international markets' target audiences via export web-based catalogue', *International Journal of Applied Decision Sciences*, Vol. 6, No. 3, pp.263–280.

Zhao, Q., Karisch, S.E., Rendl, F. and Wolkowicz, H. (1998) 'Semidefinite programming relaxations for the quadratic assignment problem', *Journal of Combinatorial Optimization*, Vol. 2, No. 1, pp.71–109.