International Journal of Applied Operational Research
Vol. 8, No. 1, pp. 23-32, Winter 2018

Journal homepage: ijorlu.liau.ac.ir

# Solve the facility's layout problem by developed genetic algorithm

H. Jafari*, M. Faraji, M. Ehsanifar

**Received:** 10 August 2017 **;** **Accepted:** 20 December 2017

**Abstract** The quadratic assignment problem (QAP) is one of the problems of combinatorial optimization belonging to the NP-hard problems' class and has a wide application in the placement of facilities. Thus far, many efforts have been made to solve this problem and countless algorithms have been developed to achieve optimal solutions, one of which is the Genetic Algorithm (GA). This paper aims at finding a suitable layout for the facilities of an industrial workshop by using a developed genetic algorithm. The research method in the current study is mathematical modeling and data was analyzed using genetic algorithm in MATLAB. The results show that the developed genetic algorithm (DGA) is highly efficient, as it has the power to discover many optimal solutions.

**Keyword:** Optimization, Facility's Layout, Genetic Algorithm, Quadratic Assignment Problem.

## 1 Introduction

The facility layout or the quadratic assignment problem is a spatial layout of goods production or service provision facilities. Koopmans and Beckmann were the first to define the issue of layout of facilities as a common industrial problem [1].

The design of the layout is an optimization problem that tries to make the deployment more efficient, taking into account the various interactions between the facilities and materials transportation system [2].

The layout problem is used in many production systems. Typically, the problem of placing facilities (including offices and machinery) is in the factory space. Since the layout is affecting transportation costs, usually the main cost in manufacturing organizations, efficient layout will have a significant role in the performance of the organization. Transportation costs account for 20 to 50 percent of the total operating costs and also 15 to 70 percent of the cost of producing a commodity. This cost is calculated based on the flow of materials among the departments and the distance between them, and the best option is the arrangement that would have the lowest transportation cost [3].

For more than five decades, scientists have studied the quadratic assignment problem and have made remarkable discoveries in this regard. Most mathematical scientists, computer

* **Corresponding Author. (✉)**
**E-mail:** Hossein_jafari_123@yahoo.com (H. Jafari)

**H. Jafari**
Young Researchers and Elite Club, Arak Branch, Islamic Azad University, Arak, Iran.

**M. Faraji**
Department of Industrial Engineering, Islamic Azad University of Arak, Arak, Iran.

**M. Ehsanifar**
Department of Industrial Engineering, Islamic Azad University of Arak, Arak, Iran.

science experts, operations research analysts, and economists use the quadratic assignment problem to model a variety of optimization problems [4].

## 2 Mathematical form of the classical problem

Assignment implies that each facility conforms to one location and vice versa. In QAP, the number of facilities and locations should be equal. The mathematical format of this problem is as follows [5]:

$$Min\ C = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{s=1}^{n} d_{i,k} w_{j,s} x_{i,j}\ x_{k,s} \tag{1}$$

$s.t.$

$$\sum_{j=1}^{n} x_{i,j} = 1\ ;\ i = 1,2,\dots,n \tag{2}$$

$$\sum_{i=1}^{n} x_{i,j} = 1\ ;\ j = 1,2,\dots,n \tag{3}$$

$$x_{i,j} = \begin{cases} 0 \\ 1 \end{cases}\ ;\ i = 1,2,\dots,n \text{ and } j = 1,2,\dots,n \tag{4}$$

## 3 Problem definition

The quadratic assignment problem is an exponential complex problem, as issues with dimensions larger than fifteen (n> 15) cannot be solved or a great amount of time is needed to solve them [6].

Hence, an evolutionary algorithm will be used to solve the layout problem in this research. The results of evolutionary algorithms, especially the genetic algorithm, are highly dependent on the primary population or the first generation [7].

In other words, if an appropriate original population is not available to the genetic algorithm, the likelihood of finding better solutions (in this research, better layouts) is reduced. On the other hand, according to the rules of the genetic algorithm, the primary population must be created completely randomly and no precise operator has been introduced for creating the first generation so far. Therefore, in this research, which is dedicated to a case study, a simple method is suggested to solve this problem.

## 4 Methodology

This research study aims to find a solution to a mathematical problem and formulate and solve the problem using mathematical literature. Hence, the mathematical modeling method (nonlinear allocation model) was used. In other words, the problem is modeled through mathematics and the final model is solved using a meta-heuristic algorithm (genetic algorithm).

## 5 The theory of evolution

According to Darwin's theory of evolution, generations with superiority over other generations will enjoy a greater chance of survival, and their superior characteristics will be passed on to their next generations. Also, the second part of Darwin's theory states that when a child's organ is propagated, some accidental events change the characteristics of the child organ; if these changes are beneficial to the child organ, it will increase the probability of the survival of that child organ. In computerized calculations, methods are suggested for optimization problems, according to Darwin's theory; all these methods come from evolutionary processing in nature. Search methods are called evolutionary search algorithms [8].

## 6 Genetic algorithm

A genetic algorithm is a search technique in computer science to find an approximate solution for optimization and search issues. The genetic algorithm is a particular type of evolutionary algorithms that uses evolutionary biology techniques such as inheritance and mutation. This algorithm was first introduced by John Henry Holland. In fact, genetic algorithms use Darwin's natural selection principles to find the optimal formula for predicting or matching patterns. Genetic algorithms are often a good option for regression-based prediction techniques. In artificial intelligence, genetic algorithm is a programming technique that uses genetic evolution as a problem solving model. The problem to be solved possesses inputs that are converted into solutions in a process modeled from genetic evolution; then, solutions are evaluated by the evaluation function as candidates, and the algorithm ends if the conditions required for exit are met. Genetic algorithm is generally a repetition-based algorithm, most of whose parts are selected as random processes [8]. The operators of this algorithm are described below.

### 6.1 Coding

Genetic algorithms deal with their coded form rather than working on the parameters or variables of the problem. Binary coding is a method of encoding, in which the goal is to transform the problem’s solution into a string of binary numbers. This operator is also possible as a permutation [9].

### 6.2 Evaluation operator

The fitness function is acquired by implementing the proper transformation of the target function, that is, the function to be optimized. This function evaluates each string with a numerical value that specifies its quality. The higher the quality of the response’s string, the more efficient the response will be, and the probability of participation in the next generation will also increase [9].

### 6.3 Crossover operator

The most important operator in the genetic algorithm is the crossover operator. A crossover is a process in which the old generation of chromosomes is mixed and combined to create a new generation of chromosomes. The couples considered in the selection section as the parent exchange their genes together and creating new members in this section [9].

### 6.4 Mutation operator

Mutation is also another operator that gives rise to other possible answers. In the genetic algorithm, after a member has been created in a new population, each gene mutate with the probability of mutation. In a mutation, a gene may be removed from a population of genes, or a gene that has not been present in the population may be added [9].

### 6.5 Different steps in the Genetic algorithm

The main phases of the genetic algorithm from the beginning to the end include [10]:
1. Start
2. Creating a primary population.
3. Estimating the initial population and sort the members.
4. Determining the best member in the primary population.
5. Performing reproduction in the previous generation.
6. Making a jump in the previous generation.
7. Selecting the optimal members of the population from the previous generation, population from reproduction and mutated population.
8. Assessing the population in the new generation and sorting the members.
9. Determining the best member in all generations.
10. If the stop condition is not met, go to step 5 otherwise go to step 11.
11. End

These steps are also shown in the following flowchart, Figure 1.

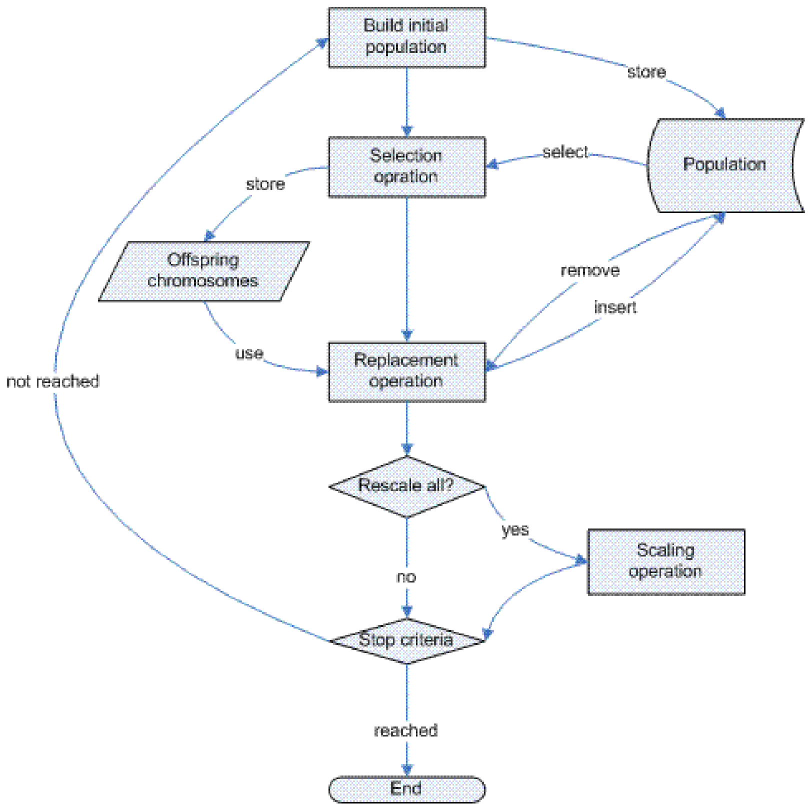


**Fig. 1** Genetic algorithm flow chart.

Figure 2 shows the structure of a chromosome, in fact, this chromosome is the coded shape of an answer to the layout problem. In Figure 3, an instance of crossover operator is shown, and Figure 4 illustrates an instance of mutation operator.

| 1 | 5 | 7 | 4 | 2 | 6 | 3 | 8 | 9 | 10 | 13 | 11 | 12 | 17 | 15 | 16 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| gene1 | gene2 | gene3 | gene4 | gene5 | gene6 | gene7 | gene8 | gene9 | gene10 | gene11 | gene12 | gene13 | gene14 | gene15 | gene16 | gene17 |

**Fig. 2** The structure of a chromosome [Author].

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Parent1: | 8 | 3 | 2 | 7 | 5 | 6 | 4 | 1 | 9 | 10 | 12 | 11 | 13 | 15 | 14 | 17 | 16 |
| Parent2: | 1 | 5 | 7 | 4 | 2 | 6 | 3 | 8 | 9 | 10 | 13 | 11 | 12 | 17 | 15 | 16 | 14 |
| offspring 1: | 8 | 3 | 2 | 7 | 5 | 6 | 4 | 1 | 9 | 10 | 13 | 11 | 12 | 17 | 15 | 16 | 14 |
| offspring 2: | 1 | 5 | 7 | 4 | 2 | 6 | 3 | 8 | 9 | 10 | 12 | 11 | 13 | 15 | 14 | 17 | 16 |

**Fig. 3** A sample of crossover operator [Author].

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Befor Mutation: | 1 | 5 | 7 | 4 | 2 | 11 | 3 | 8 | 9 | 10 | 13 | 6 | 12 | 17 | 15 | 16 | 14 |
| After Mutation: | 1 | 5 | 7 | 4 | 2 | 17 | 3 | 8 | 9 | 10 | 13 | 6 | 12 | 11 | 15 | 16 | 14 |

**Fig. 4** A sample of mutation operator [Author].

Note that the fitness function in the current genetic algorithm is the same objective function in the QAP problem.

## 7 Proposed approach

After completing the computer codes associated with the genetic algorithm, the algorithm is placed in a repeating loop and it will be executed countless times (N). This causes the algorithm to be very diverse in the first generation. Finally, the best scenario is chosen from the recovery scenarios. Depending on the importance of the problem, researchers can select the N number large or small. Obviously, choosing larger Ns will yield more reliable results.

## 8 Case study

A case study in this research involves an industrial workshop producing a variety of wood and metal products. The workshop consists of 17 facilities and 17 stations. The purpose of this paper is to establish optimal facilities at stations, based on the distance between stations and the machines transportation flow.

### 8.1 Stations distance for the studied workshop

The distance between the stations is shown in Table 1.

**Table 1** Station Distance Matrix [Author].

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 25 | 38 | 50 | 60 | 77 | 22 | 35 | 49 | 63 | 77 | 28 | 43 | 55 | 67 | 79 | 101 |
| 2 | 25 | 0 | 16 | 28 | 40 | 55 | 5.5 | 17.5 | 31.5 | 44.5 | 58.5 | 23 | 38 | 50 | 62 | 74 | 98 |
| 3 | 38 | 16 | 0 | 17 | 29 | 41 | 17.5 | 6.5 | 19.5 | 33.5 | 47.5 | 23 | 24 | 36 | 48 | 60 | 68 |
| 4 | 50 | 28 | 17 | 0 | 16 | 31 | 29.5 | 15.5 | 7.5 | 21.5 | 35.5 | 35 | 20 | 28 | 32 | 44 | 56 |
| 5 | 60 | 40 | 29 | 16 | 0 | 19 | 41.5 | 27.5 | 13.5 | 9.5 | 23.5 | 47 | 32 | 20 | 24 | 36 | 44 |
| 6 | 77 | 55 | 41 | 31 | 19 | 0 | 56.5 | 42.5 | 28.5 | 14.5 | 8.5 | 62 | 47 | 35 | 23 | 23 | 25 |
| 7 | 22 | 5.5 | 17.5 | 29.5 | 41.5 | 56.5 | 0 | 7 | 31 | 45 | 59 | 8 | 23 | 35 | 47 | 59 | 81 |
| 8 | 35 | 17.5 | 6.5 | 15.5 | 27.5 | 42.5 | 7 | 0 | 7 | 31 | 45 | 14 | 9 | 21 | 32 | 44 | 68 |
| 9 | 49 | 31.5 | 19.5 | 7.5 | 13.5 | 28.5 | 31 | 7 | 0 | 7 | 31 | 28 | 13 | 7 | 19 | 31 | 55 |
| 10 | 63 | 44.5 | 33.5 | 21.5 | 9.5 | 14.5 | 45 | 31 | 7 | 0 | 7 | 44 | 27 | 15 | 5 | 17 | 41 |
| 11 | 77 | 58.5 | 47.5 | 35.5 | 23.5 | 8.5 | 59 | 45 | 31 | 7 | 0 | 56 | 41 | 29 | 17 | 5 | 26 |
| 12 | 28 | 23 | 23 | 35 | 47 | 62 | 8 | 14 | 28 | 44 | 56 | 0 | 20 | 32 | 44 | 56 | 80 |
| 13 | 43 | 38 | 24 | 20 | 32 | 47 | 23 | 9 | 13 | 27 | 41 | 20 | 0 | 16 | 27 | 39 | 63 |
| 14 | 55 | 50 | 36 | 28 | 20 | 35 | 35 | 21 | 7 | 15 | 29 | 32 | 16 | 0 | 16 | 28 | 52 |
| 15 | 67 | 62 | 48 | 32 | 24 | 23 | 47 | 32 | 19 | 5 | 17 | 44 | 27 | 16 | 0 | 16 | 40 |
| 16 | 79 | 74 | 60 | 44 | 36 | 23 | 59 | 44 | 31 | 17 | 5 | 56 | 39 | 28 | 16 | 0 | 28 |
| 17 | 101 | 98 | 68 | 56 | 44 | 25 | 81 | 68 | 55 | 41 | 26 | 80 | 63 | 52 | 40 | 28 | 0 |

### 8.2 Percentage of transportation flows for the studied workshop

The percentage of displacement between machines is shown in Table 2.

**Table 2** Matrix of transportation percentage between facilities [Author].

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 16.9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 12.15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.97 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.83 | 0 | 10.35 | 0 | 0 | 0 | 0 | 0 | 3.65 |
| 4 | 0 | 0 | 0 | 0 | 0 | 12.27 | 2.79 | 2.79 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 12.61 | 0 | 0 | 2.23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.01 | 0 | 0 | 0 | 0.01 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.03 | 0.03 | 0 | 0 | 0 | 0 | 0.09 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0.37 | 0 | 0 | 0 | 0 | 0 | 0.41 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 3.67 | 1.2 | 5.27 | 2.39 | 3.99 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### 8.3 Building a mathematical model of the facility layout problem for the studied workshop

The model is as follows:

$$Min \quad Cost = \sum_{i=1}^{17}\sum_{j=1}^{17}\sum_{k=1}^{17}\sum_{s=1}^{17} d_{i,k} w_{j,s} x_{i,j}\, x_{k,s} \tag{5}$$

s.t.

$$\sum_{j=1}^{17} x_{i,j} = 1\ ;\ i = 1,2,\ldots,17 \tag{6}$$
$$\sum_{i=1}^{17} x_{i,j} = 1\ ;\ j = 1,2,\ldots,17 \tag{7}$$
$$x_{12,j} = 0\ ; j\epsilon\{1,2,3,4,5,6,7,8,9,10,11,12,13,15,16\} \tag{8}$$
$$x_{6,j} = 0\ ; j\epsilon\{1,2,3,4,5,6,7,8,9,10,11,12,13,15,16\} \tag{9}$$
$$x_{1,j} = 0\ ; j\epsilon\{1,2,3,4,5,6,7,8,9,10,11,12,13,14,17\} \tag{10}$$
$$x_{17,j} = 0\ ; j\epsilon\{1,2,3,4,5,6,7,8,9,10,11,12,13,14,17\} \tag{11}$$
$$x_{7,j} = 0\ ; j\epsilon\{1,3,4,5,8,10,11,13,14,15,16,17\} \tag{12}$$
$$x_{8,j} = 0\ ; j\epsilon\{1,3,4,5,8,10,11,13,14,15,16,17\} \tag{13}$$
$$x_{9,j} = 0\ ; j\epsilon\{1,3,4,5,8,10,11,13,14,15,16,17\} \tag{14}$$
$$x_{10,j} = 0\ ; j\epsilon\{1,3,4,5,8,10,11,13,14,15,16,17\} \tag{15}$$
$$x_{11,j} = 0\ ; j\epsilon\{1,3,4,5,8,10,11,13,14,15,16,17\} \tag{16}$$
$$x_{2,j} = 0\ ; j\epsilon\{2,6,7,9,12,14,15,16,17\} \tag{17}$$
$$x_{3,j} = 0\ ; j\epsilon\{2,6,7,9,12,14,15,16,17\} \tag{18}$$
$$x_{4,j} = 0\ ; j\epsilon\{2,6,7,9,12,14,15,16,17\} \tag{19}$$
$$x_{5,j} = 0\ ; j\epsilon\{2,6,7,9,12,14,15,16,17\} \tag{20}$$
$$x_{13,j} = 0\ ; j\epsilon\{2,6,7,9,12,14,15,16,17\} \tag{21}$$
$$x_{14,j} = 0\ ; j\epsilon\{2,6,7,9,12,14,15,16,17\} \tag{22}$$

$$x_{15,j} = 0 \,;\, j\epsilon\{2,6,7,9,12,14,15,16,17\} \quad (23)$$
$$x_{16,j} = 0 \,;\, j\epsilon\{2,6,7,9,12,14,15,16,17\} \quad (24)$$
$$x_{12,14} = x_{1,16} \quad (25)$$
$$x_{12,17} = x_{1,15} \quad (26)$$
$$x_{6,14} = x_{17,16} \quad (27)$$
$$x_{6,17} = x_{17,15} \quad (28)$$
$$x_{i,j} = \begin{cases}0\\1\end{cases} ;\, i = 1,2,\dots,17 \text{ and } j = 1,2,\dots,17 \quad (29)$$

Excessive constraints on the classic mathematical model indicate the inability to install some machines in some departments.

### 8.4 Solving the model with the help of MATLAB software and proposed method

To determine the worst-case layout, the proposed algorithm has been used with N = 100, population 12, 50% reproduction, and 50% mutation. In Figure 5, the amount of best costs is presented in various algorithm performances.

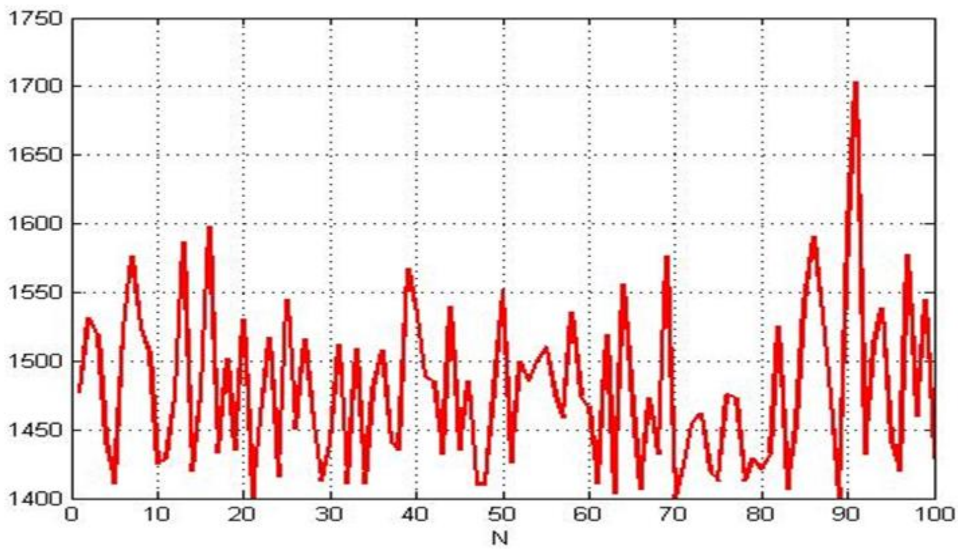


**Fig. 5** The amount of the best costs in various algorithm performances [Author].

The results show that the layout obtained from the proposed method to find the best layout costs 1400.845. The optimal layout is also shown in Table 3.

**Table 3** Optimal layout

| | $F1$ | $F2$ | $F3$ | $F4$ | $F5$ | $F6$ | $F7$ | $F8$ | $F9$ | $F10$ | $F11$ | $F12$ | $F13$ | $F14$ | $F15$ | $F16$ | $F17$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $L1$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| $L2$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $L3$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $L4$ | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $L5$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| $L6$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $L7$ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $L8$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $L9$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| $L10$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $L11$ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $L12$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| $L13$ | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $L14$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $L15$ | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $L16$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| $L17$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

## 9 Discussion

Most studies on facility placement assume that all machinery can be deployed in all locations. In real-world problems, however, this assumption may not be true. This study deals with a case in which certain machinery cannot be deployed in certain locations. This problem clearly requires more constraints than the basic problem. Therefore, a specific mathematical model was defined for the research problem. Since this model does not have an exact solution, genetic algorithm was employed to solve it. The results of the problem solution indicate an acceptable placement for the workshop under study. Moreover, the study is significant in another aspect. Here, different initial populations were used in the data analysis to arrive at different solutions, since the initial solutions played a crucial role in the genetic algorithm in the final solutions.

## 10 Conclusion

Proper layout of facilities has a direct relationship with the final cost of goods in large and small manufacturing units. If an incorrect location is found for facilities in industrial units, it is evident that the more the interactions between workstations or departments of production unit increase, the more the manufacturing costs will increase. Meanwhile, the mentioned problem is one of the exponential complex problems that generally needs meta-analysis

methods to be solved. In this study, by adding a simple step with genetic algorithm, attempts were made to create generations with greater diversity so that the algorithm could achieve better results and it does not stick in local extrema.
Future studies can address the problem of facility placement with unequal areas using meta-heuristic algorithms.

## References


1. Koopmans, T.C., and Beckmann, M. J. (1957). Assignment problems and the location of economic activities, Econometrica, (25),53-76.
2. Azadivir, F., and Wang, J. (2000). Dynamic facility layout design using simulation and genetic algorithms. International Journal of Production Research, (38), 4369–4383.
3. Gupta, T., and Seifoddini, H. (1990). Production Data Based Similarity Coefficient for Machine–Component Grouping Decisions in the Design of Cellular Manufacturing System. International Journal of Production Research, 28 (4), 1247-1269.
4. Afentakis, P., Millen, R.A., and Solomon, M.M. (1990). Integrated Approach to Facilities Layout using Expert Systems. International Journal of Production Research, 28 (2), 311-323.
5. Xia, Y. (2010). An efficient continuation method for quadratic assignment problems. Computers & Operations Research, 6(37), 1027-1032.
6. Michael, R.G., and David, S.J. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman. ISBN 0-7167-1045-5. A2.5: ND43, pg.218.
7. Schmitt, L.M., Nehaniv, C.N, and Fujii, R.H. (1998). Linear analysis of genetic algorithms, Theoretical Computer Science,(208),111–148.
8. Hassan, M.M. (2013). Optimization of stay cables in cable-stayed bridges using finite element, genetic algorithm, and B-spline combined technique.Engineering Structures, (49), 643–654.
9. Schmitt, L.M., Nehaniv, C.N., and Fujii, R.H. (1998). Linear analysis of genetic algorithms, Theoretical Computer Science, 208,111–148.
10. Schmitt, L.M. (2004). Theory of Genetic Algorithms II: models for genetic operators over the string-tensor representation of populations and convergence to global optima for arbitrary fitness function under scaling, Theoretical Computer Science, 310, 181–231.