Research Article

Developing a mathematical model for the dynamic facility layout problem considering material handling system and optimizing it using cloud theory-based simulated annealing algorithm

N. Pournaderi[1] · V. R. Ghezavati[2] · M. Mozafari[1]

Abstract

The optimization of production systems has always been noteworthy for researchers. Today's world experienced rapid changes in technology and a reduction in the product life cycle. So, manufacturing must be able to respond quickly to these alterations, especially in the product mix and demand, with minimizing transport and handling equipment cost. Therefore, it is necessary to examine the dynamic cell formation problem. In respect of layout replacement (changing) needs a huge budget, compliance with budget constraints is one of the realistic aspects of modeling and problem solving in these types of issues. Hence, in this study, budgetary limitations are considered in modeling as well. The carrier equipment that transports materials between machines also is examined, to select optimal ones under consideration of the cost. Although, transporting materials costs is applied to the system constantly and during a period replacement facility costs and carrier equipment fixed costs are imposed immediately and at the beginning of the period. For this reason, these two costs are different in nature; thus, minimizing these two types of costs is considered as separate objectives in the model by this study. As the issue is complex, the multi-objective based on could theory and simulated annealing algorithms will be offered for solving the model. Finally, the Taguchi method is applied to adjust the algorithms parameters and algorithms performance is measured by solving some random sample.

Keywords Dynamic layout · Budget constraints · Carrier equipment · Multi-objective cloud theory algorithms · Taguchi method

1 Introduction

The planning of formation facility plays an important role in modern systems and it affects the competitiveness of a firm seriously. Most firms have to update the arrangement of the production cells because of the quantity change of the productions, mixed models and short life cycle of a product such as a cell phone. Moreover, due to creativity and innovation in production technologies, we need to use the facilities change. The efficient arrangement leads to a reduction in material handling costs by reducing the materials flow and decreasing the distance between cells, then decreasing production costs and increasing competition power of the organization will be achieved. An effective arrangement helps the other operations that are related to workflow, work better. According to the nature of the data of material flow, the arrangement classifies into two categories: static and dynamic. The arrangement where the flow of materials between sectors does not change over time is known as a static facility layout problem (SFLP). When the flow of material between sectors varies through the planning horizon, it changes to dynamic facility layout problem (DFLP). MacKendall and Hakobyan [1] stated that factors such as design changing to produce new products, add or remove products, change and relocation of existing production tools, short product

✉ V. R. Ghezavati, v_ghezavati@azad.ac.ir; vrghezavati@gmail.com | [1]Department of Industrial Engineering, Islamic Azad University, Electronic Branch, Tehran, Iran. [2]School of Industrial Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran.

SN Applied Sciences (2019) 1:832 | https://doi.org/10.1007/s42452-019-0865-x

Received: 2 April 2019 / Accepted: 1 July 2019 / Published online: 5 July 2019

life cycles, changes in production volume and the production planning may change the material flow. The solving of DFLS is as an arrangement plan. An arrangement plan for DFLP could be shown as a series of plans that each of them associated with a period. Therefore, the arrangement plan is determined on the Planning horizon and minimizes the total of the material handling costs and rearrangement costs of the units in consecutive periods.

Basic question about the two-objective DFLS is how to choose the layout of the machines and how to assign the transporters to the equipment in order to minimize the total cost of operation including the cost of materials handling between production cells and cost of movement production cells. The purpose of this study is to provide a model to answer this basic question.

Koopmans and Beckmann [2] were the first to define the statistic facility layout problem as a current industrial problem and declared the aim of this kind of production cell layout problem is minimizing the cost of material handling between units. Statistic facility layout is studied completely and it dated more than five decades but DFLP is studied in 1986 for the first time by Rosenblatt [3].

In general, different approaches to solving dynamic facility layout problems raised in the literature can be classified into four groups (Kulturel-Konak [4]):

1. Exact methods
2. Heuristics approaches
3. Metaheuristic approaches
4. Hybrid approaches.

Rosenblatt [3], Lacksonen and Enscere [5] used exact methods to solve. Urban [6] used a heuristic method to solve. For the first time, Conway and Venkataramanan [7] used genetic algorithms to solve DFLP. Kochhar and Heragu [8] solved the multi-level facilities layout problem by genetic algorithms and the function of proposed algorithm problems was tested with 6, 15 and 30 departments in 5 and 10 periods. Burkard and Rendl [9] were the first who used SA to solve the facility layout problems. In literature, it is proved that the genetic algorithm has less performance than the simulated annealing algorithm.

Balakrishnan et al. [10], Lee and Lee [11], Dunker et al. [12] used a hybrid genetic algorithm to solve. Kaku and Mazzola [13], Sahin and Turkbey [14] as well as McKendall and Liu [15] used a tabu search algorithm to solve. Baykosaglu and Gindy [16], McKendall et al. [17] used a simulated annealing algorithm to solve.

Azimi [18] used simulation as an efficient tool in the salesman problem and showed how simulation is able to solve such problems efficiently. Chen [19] provided a new representation way to display the answer. The proposed representation method can be used in all of the meta-heuristic algorithms and can enhance search capabilities. The performance of this display method is examined by the ant colony algorithm and it can lead to better answers in less calculation time. Xu and Song [20] implied the movement cost between machines in the form of phase parameters and considered facilities in the form of two-dimensional shapes, then the proposed problem was optimized by using the PSO algorithm. Finally, the efficiency of the proposed algorithm was evaluated and approved using a case study.

Evans et al. [21], Grobelny [22], Raoot and Rakshit [23], Gen et al. [24], Dweiri and Meier [25], Aiello and Enea [26] as well as Deb and Bhattacharyya [27] used fuzzy data.

Chen and Rogers presented a multi-objective dynamic model to explore several aspects of planning layout facility including time, distance-based objective and neighborhood-based objective. They applied ant colony optimization for solving the problem. Although our model and the model of Chen and Rogers are both multi-objectives, Chen and Rogers used Urban' method for developing the model, and finally, only an objective function is defined as total.

Despite lots of researches have been done in this field, the researches need improvement in planning methods. According to the literature review, assuming unequal facilities and budget constraints, have been considered in previous studies, but the type of transportation vehicles, limited numbers and fixed costs of these vehicles have not considered in any of the studies. Also, almost in all the research, the material handling costs and the cost of moving machinery considered as an objective, but each of these costs in the objective function may have different importance for decision-makers. Therefore, it is better to consider the objective function as two separate functions. This type of formulation makes the decision makers be able to impose their views. On the other hand, due to the single-objective, most researchers used GA, TS and SA methods or hybrid algorithms to solve.

Therefore, we decide to consider the DFLP problem as a multi–objective problem, consider the cost of transporter means in the function objective, and simultaneously consider budget limits and kind of transportation vehicles as constraints in this paper. Despite the reputation of assignment quadratic problem, the problem is difficult to solve by the use of traditional optimization algorithms (Garey and Rogers [28]). As Francis et al. [29] stated, the assignment quadratic problem for more than 15 to 20 departments is NP-hard. In order to find the optimal solution in a DFLP with N production cells and T planning period, the $(N!)^T$ options should be evaluated. Therefore, we will design and apply a multi-objective metaheuristic method in this study.

2 Problem statement and modeling

The facility layout problems often are modeled to assignment quadratic problem.

Dynamic facility layout problem depends on the changes in material flow and it is predicted that it happens in the future. Future divides into the number of periods of time and it is the problem of the arrangement planning for some periods, so solutions are in the form of layout planning. A layout planning of the DFLP could be shown as a series of plans each related to a period. Therefore, layout planning is determined in a planning horizon and minimizes the total material handling cost and the rearrangement cost in consecutive periods. When the facility layout is planned in consecutive periods of time and production cells and machinery are moved from one location to another, the rearrangement cost appears. The rearrangement cost can produce another cost, too. For example, you may need special equipment and some may be employed or unemployed.

2.1 Decision variables and parameters

Decision variables and symbols and parameters considered in this model are as follows:

2.1.1 Symbols and parameters of the problem

(A) Index:

i, k = facility index i, k = {1,2, …, N}
j, l = locations index j, l = {1,2, …, N}
tr = transporters index tr = {1,2, …, M}
t = period index t = {1, 2, …, T}
N = the number of facilities and locations
T = number of planned periods
M = number of transporters type

(B) Parameters:

$C_{t,i,k,m}$ = the materials handling cost between facility *i* and facility *k* by transporter *m* at period *t*
FC_{tm} = fixed cost of transporter *m* at period *t*
A_{tijl} = cost of moving facility *i* from the location *j* to *l* at period of *t*
$D_{j,l}$ = distance between location *j* and *l*
LB_t = remaining budget from period *t* to period *t + 1*
B_t = available budget at the period *t*
AB_t = allocated budget for the period *t*
$A\text{-}TR_{t,m}$ = maximum number of available transporter *m* at period *t*

2.1.2 Decision variables

$$X_{tij}\begin{cases}1 & \text{facility } i \text{ assign to location } j \text{ at the period } t\\ 0 & \text{Otherwise}\end{cases}$$

$$Y_{t,i,k,m}\begin{cases}1 & \text{If transporter m has been selected to move the material from machine i to the machine k at the period } t\\ 0 & \text{Otherwise}\end{cases}$$

In research, the assumptions are as follows:

1. In each situation can only be one facility.
2. Every facility can be in only one situation.
3. The amount of material flow is certain and pre-determined in all periods of time.
4. The movement cost of every facility is definite and certain in each period.
5. The number of facilities and situations are the same.
6. Each facility can be placed in any position.
7. Available transporter is predetermined.

2.2 Modeling

$$MinZ_1 = \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N} D_{j,l} * F_{t,i,k} * X_{tij} * X_{tkl} * Y_{t,i,k,m} * C_{t,i,k,m} \tag{1}$$

$$MinZ_2 = \sum_{t=2}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{l=1}^{N} A_{tijl} * X_{t-1,ij} * X_{til} + \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{k=1}^{N}\sum_{m=1}^{M} Y_{t,i,k,m} * FC_{tm} \tag{2}$$

Subject to:

$$\sum_{j=1}^{N} X_{t,i,j} = 1 \qquad i = 1, 2, \ldots, N; \; t = 1, 2, \ldots, T \tag{3}$$

$$\sum_{i=1}^{N} X_{t,i,j} = 1 \qquad j = 1, 2, \ldots, N; \; t = 1, 2, \ldots, T \tag{4}$$

$$\sum_{m=1}^{M} Y_{t,i,k,m} = R_{t.i.k}$$
$$R_{t,i,k} = \begin{cases} 1 & if\ F_{t,i,k} \geq 0 \\ 0 & else \end{cases} \quad \begin{array}{l} i = 1, 2, \ldots, N-1 \\ k = i+1, 2, \ldots, N \end{array} \tag{5}$$

$$\sum_{i=1}^{N-1} \sum_{k=i+1}^{N} Y_{t,i,k,m} \leq A\_TR_{t,m} \quad \begin{array}{l} m = 1, 2, \ldots, M \\ t = 1, 2, \ldots, T \end{array} \tag{6}$$

$$Y_{t,i,k,m} = Y_{t,k,i,m} \quad \begin{array}{l} i, k = 1, 2, \ldots, N \\ t = 1, 2, \ldots, T \\ m = 1, 2, \ldots, M \end{array} \tag{7}$$

$$\begin{aligned} LB_t = B_t - \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{l=1}^{N} A_{tijl} * X_{t-1,ij} * X_{til} \\ - \sum_{i=1}^{N} \sum_{k=1}^{N} \sum_{m=1}^{M} Y_{t,i,k,m} * FC_{tm}, \quad t = 1, 2, \ldots, T \end{aligned} \tag{8}$$

$$B_t = AB_t + LB_{t-1}, \quad t = 1, 2, \ldots, T \tag{9}$$

$$\begin{aligned} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{l=1}^{N} A_{tijl} * X_{t-1,ij} * X_{til} \\ + \sum_{i=1}^{N} \sum_{k=1}^{N} \sum_{m=1}^{M} Y_{t,i,k,m} * FC_{tm} \leq B_t, \quad t = 1, 2, \ldots, T \end{aligned} \tag{10}$$

$$X_{t,i,j} \in \{0, 1\} \quad \begin{array}{l} i, j = 1, 2, \ldots, N \\ t = 1, 2, \ldots, T \end{array} \tag{11}$$

$$Y_{t,i,k,m} \in \{0, 1\} \quad \begin{array}{l} i, k = 1, 2, \ldots, N \\ m = 1, 2, \ldots, M \\ t = 1, 2, \ldots, T \end{array} \tag{12}$$

$$LB_t, B_t, AB_t \geq 0, \quad t = 1, 2, \ldots, T \tag{13}$$

The first objective function (1) minimizes the summation of carrying material cost between cells. The second objective function (2) is associated with the costs at the beginning of the period and tries to minimize the costs of machinery movement and fixed cost of transporters. Constraint (3) ensures that every facility will only be in one situation and constraint (4) ensures that in each situation, only one facility takes place. Constraint (5) states that for a pair of machines if there is a material flow, one transporter should be assigned. Constraint (6) is to control the number of transporters used in each period. Constraint (7) ensures that for each pair of machines, one type of transporter would be assigned. The constraint (8) relates that the budget transferred to the next period is equal to the current budget minus the relocation and transportation costs in the current period. Constraint (9) shows that the available budget for every period is equal to the total budget allocated to the same period and the remaining budget from the previous period. Constraint (10) indicates a max budget for each period. Constraints (11), (12) and (13) controls the values of the decision variables.

After modeling, the validity of the model must be examined. For this purpose, a small example was resolved by Lingo software (version 8). The model runs without error and the answer was reasonable and any of the answers did not violate limitations. Examples including 6 departments, 6 situations and 3 types of the transporter and horizontal time also comprised of two periods.

In this paper, the maximum time suitable to solve a problem was considered 1000 s and the examples were solved with an exact algorithm. After solving the problems, it was determined that in the problem with 7 departments and 3 periods of time, it is impossible to achieve the answer within 1000 s. During this period of time of performance, the software can't achieve the answer. Therefore, to solve problems with the size of more than 7 departments and 3 periods in an acceptable time, it is needed to develop heuristic and meta-heuristic methods. So, a new metaheuristic algorithm is proposed for solving the dynamic facility layout problem. The proposed algorithm is able to solve the model efficiently by using the annealing metals principle and cloud theory.

3 Solution approaches

Optimization algorithms presented in this paper are the following:

1. Non-dominated Sorting Genetic Algorithm (NSGA-II)
2. Non-dominated Ranked Genetic Algorithm (NRGA)
3. Multi-Objective Cloud Simulated Annealing Algorithm (MOCSA).

In this research, the binary tournament selection strategy is used. The selection strategy is used to select two parents from the population. In this method, for the selection of two parents, first of all, two members of the population are selected by random and then they are selected based on compared rank. Anyone who has less rank is selected and if they have the same rank then they are compared according to the crowding distance and each one has higher crowding distance is chosen as a parent.

Ranking of the population occurs using the following two steps:

1. Fast Non-dominated sort
2. Crowding distance: Less crowding distance shows that the answers have more density.

Operators which are used in order to produce generation:

1. Crossover operator
2. Mutation operator.

3.1 Non-dominated sorting genetic algorithm (NSGA-II)

This algorithm was presented in 2002 by Deb et al. To perform non-dominated sorting genetic algorithm the following steps should be done [30]:

1. The parents population (P_t) and offspring population (Q_t) merged together in order to create a population (R_t). We must now select N top answers of R_t and make new parents population P_{t+1}.
2. Non-dominated sorting performance on Rt and its different boundary is shown with (F_i: i = 1, 2, …) will be determined and identified.
3. The process of putting the answer in the population begins from the answers belonging to the first border that is F_1. The increase of the members of every border will continue until the number of members of the population is still less than N.
4. If the sum of the members of i th border and the number of current members of the population is more than N i.e.: $|Fi| + |P_{t+1}| > N$, crowding distance of any answer will be calculated and the number of $(N - |P_{t+1}|)$ from the answers which have the highest density are added into the population.
5. The population of new offspring (Q_{t+1}) will be created by applying the selection, crossover and mutation operators on P_{t+1}.

In this research, designing the structure of chromosomes is so that, the produced answer satisfies constraints of the arrangement and transporter automatically. But about the constraints of the budget must be used in a different approach. To solve this problem and to evaluate the chromosomes in the algorithms actually by penalizing, the function objective changes into an appropriate fitness function and we balance the objective function by a linear transformation:

$$Z_i = Z_i + a * b \tag{14}$$

In this research, the stop criteria of the algorithm is to reach the predefined maximum iteration. The amount of Iteration is calculated in the parameter adjustment discussion and by Taguchi experimental design.

3.2 Non-dominated ranked genetic algorithm (NRGA)

Al Jadaan et al. [31] used a modified selection algorithm based on the roulette wheel where it is assigned an amount for each answer equal to the rank of the answer in the population. The difference of NRGA with NSGA-II is in the selection strategy, population arranging and selection for the next generation.

First, we arrange the population answers in the non-dominated borders so that the first border will have the best answers in the population. So, if at this stage, we have 5 non-dominated borders for the population, we allocate 5 scores for the first border and 1 score to the fifth border. Therefore, the higher the score shows answers to that border are better. After ranking the borders, the answer in each border also will be ranked based on crowding distance. Therefore, after calculating the crowding distance for all responses present in every border, the response that has the most crowding distance will have the highest rank and for the response with the lowest crowded distance will have rank 1. In the selection section, we use the roulette wheel operator based on ranking. This selection methodology selects better members with more probability for reproduction and generates the next generation.

First, the roulette wheel is defined on the two intervals [0, S1] and [0, S2], $S_1 = \sum_{i=1}^{n} p_i$ and $S_2 = \sum_{j=1}^{m} p_{0j}$. Then, answers present in the border occupy some of these two intervals based on the probability of their selection. Then two random numbers are selected between 0 and 1 and the first random number is used for selecting the border in the range [0, S1] and the second random number is used for selecting one of the responses present in the selected border in the range [0, S2].

3.3 Multi-objective cloud simulated annealing algorithm (MOCSA)

In SA, in addition to accepting a solution that improves the objective function, worse solutions are accepted by probability too. The probability function is shown in Eq. (15):

$$P_{SA} = e^{\frac{-\Delta}{T}}; \quad \Delta = \frac{f(w_n) - f(w)}{f(w_n)} \times 100 \tag{15}$$

where Δ is the rate of the changes of the objective function and T represents temperature. If P_{SA} is more than a random number between zero and one, the bad answer is accepted. As a general rule, every iteration of SA algorithm creates a neighbor state likes and based on a probability, the problem goes from s state to s′ state or remain in the same s state. This step repeats until we achieve a relatively optimal solution or the maximum number of iterations have been done. In the case of multi-objective, first of all, some initial solutions were produced and this solution searches the solved space in parallel. At the end of each iteration, all solutions are put together and non-dominated solutions are removed. Remain solutions are considered as the initial solution of the next iteration.

Cloud theory is the innovation and development of membership function at the fuzzy theory which is obtained from the transformation of uncertainty between quantity and quality based on the time value concept. The concept of physical annealing, molecules as the dropped temperature move randomly from large-scale to small-scale. Cloud theory is easy in Linguistic description but difficult to simulate on the computer. However, because cloud theory can explain the transfer from a qualitative concept and present it as a numeric form, it is used as a guide for performance. Cloud theory is used for producing continuous annealing temperature. Cloud theory has an indiscriminate characteristic and tends to be stable. Although annealing temperature varies randomly, its rupture reduces variability in the search. In this condition and based on cloud theory when the system is in equilibrium, the temperature of the system will be generated around a point and accumulate.

The structure of the MOCSA algorithm solution presentation proposed is similar to the proposed NSGA-II presentation.

The generator of neighborhood structure has the task of producing the next states and the movement state of an algorithm is determined according to the calculation of current point cost and the next point cost. The movement generator changes the solution from the current state to the neighboring state. It is done by 2-opt replacement, in this way, first, one substring representing the layout plan related to a period, is selected then two situations are determined by random and their inner facilities are replaced.

Different component cooling programs include:

A. Initiate temperature: we produce a large number of random solutions and determine their objective function; then we calculate the rate of the standard deviation of the obtained results and used to determine the initial temperature. In the proposed algorithm, 1.5 times of standard deviation present in the primary responses have been used to determine the initial solution. It should be noted that for each objective, a separate temperature will be considered.
B. The final temperature
C. Reduction of the temperature in each stage: temperature reduction can usually be achieved by a simple linear relation:

$$T_k = \alpha T_{k-1} \tag{16}$$

The parameter α is determined through experiments analysis.

Of course, as stated in the SA algorithm based on cloud theory, the temperature at each level is not stable and should have minor changes. In this paper the following linear equation is used to reduce the temperature:

$$T_A = T_k + RAND\left[-0.1 * T_k, 0.1 * T_k\right] \tag{17}$$

In this equation, T_k is the amount of base temperature in a certain balance and T_A is the temperature used in the reception function. The RAND [a, b] function gives a random number in the range of a to b.
D. iteration at any temperature: the number of iterations at each temperature will be determined through experiments analysis.

The probability of bad movement acceptation is calculated by Eq. (18):

$$\begin{matrix} P1 = \exp\left(\frac{-\Delta C1}{t1}\right)\rangle r \\ P2 = \exp\left(\frac{-\Delta C2}{t2}\right)\rangle r \end{matrix} \quad P = (P_1 + P_2)/2 \tag{18}$$

where ΔC1 and ΔC2 are differences between the current solution and the neighborhood solution in the first and second evaluation functions; t1 and t2 are the current temperatures of the system for each objective. r is a random number between zero and one and p, the probability of moving to a new solution.

The move will be to the new solution if the new solution is better than the current one or the amount of movement probability function is greater than a random number of the range (0, 1]. Otherwise, the explorer will generate and evaluate another new solution. The process continues step by step until stop criteria are achieved. In this algorithm, such as proposed NSGA-II, stop criteria is achieving a defined iteration.

In order to increase the efficiency of the proposed three algorithms for the proposed model, input parameters of algorithms are set in their best quantity by using the parameter setting method.

3.4 Setting parameters and calculation results

In this paper for parameter setting, at the first determine controllable factors, uncontrollable factors, performance evaluation Criterion and levels related to every factor, then select the appropriate orthogonal arrays and ultimately the optimum levels will be determined and data will be analyzed using the signal ratio to noise.

Table 1 shows the domain of searching input parameters levels of three algorithms.

Table 1 Controllable factors and their levels

Algorithm	Parameters	Range	Level 1	Level 2	Level 3
NSGA-II	Pop size	100–300	100	200	300
	Pc	0.7–0.9	0.7	0.8	0.9
	Pm	0.05–0.1	0.05	0.075	0.1
	Iteration	300–700	300	500	700
NRGA	Pop size	100–400	100	250	400
	Pc	0.7–0.9	0.7	0.8	0.9
	Pm	0.05–0.1	0.05	0.075	0.1
	Iteration	500–1000	500	750	1000
MOCSA	Inner-loop	10–30	10	20	30
	Outer-loop	200–400	200	300	400
	Alpha	0.95–0.99	0.95	0.97	0.99
	Pop size	100–200	100	150	200

The complete factorial experimental project for the above four factors needs 81 tests or treatment combination. But this type of project in terms of cost and time is not economical. On the other hand, statistically no need to experiment with all combinations of factor levels. So, we will use fractional repeat projects. In order to select an appropriate orthogonal array, we must calculate the required number of degrees of freedom. In this case, one degree of freedom for total mean and two degrees of freedom for each of three level factors are needed. Therefore, the total required degrees of freedom is equal to 9 and the orthogonal array is L27. Structure of designed arrays and the results of experiments for the NSGA-II, NRGA, and MOCSA algorithms were calculated. The intended response variable to solve the proposed model is the number of solutions present in the front of Algorithms (NOS).

The aim of these experiments is the optimization of the control factors levels combination. To achieve this purpose, the performance measurements provided by Taguchi, that is to say, the ratio S/N is considered as a response variable. The response variable must be as large as possible, so its variable matches with the state “bigger is better”. According to this context, the ratio of S/N for the mentioned variable is the following:

$$\eta = \left(\frac{S}{N}\right) = -10\log\left(\frac{1}{n}\right)\sum_{i}\left(1/y_i^2\right) \tag{19}$$

In the tables taken from the software, the manner how the index quantities S/N change at different levels of algorithms is studied, and the levels in which the index S/N achieved its peak are selected as optimal level and their quantities are given in the Table 2.

In this study, due to multi-objective structure of the proposed model, we applied two curves for convergence ability of the proposed solution approaches. The first curve indicates convergence of the Pareto front after the determined iterations for the best solution approach (MOCSA). The second curve shows the convergence of all solution methods in terms of optimality. Since the objectives of the model are minimization, so the method, which can find the solutions with minimal cost, is more favorable. For this purpose, a sample of example is generated according to the following data:

- 6 Departments and 10 periods;
- Distance matrix

	1	2	3	4	5	6
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

- Flow of materials matrix

	1	2	3	4	5	6	1	2	3	4	5	6
1	0	90	689	194	165	494	0	257	1632	330	117	285
2	668	0	1324	811	241	206	159	0	1309	297	803	404
3	631	387	0	125	281	375	98	82	0	271	222	383
4	80	495	615	0	222	221	110	404	1174	0	750	386
5	276	204	1127	490	0	676	73	507	1679	190	0	107
6	109	409	1780	394	200	0	152	487	355	646	315	0
1	0	1348	490	447	186	169	0	159	1103	218	297	95
2	625	0	74	307	777	326	631	0	1618	95	253	109
3	114	1645	0	288	975	68	552	213	0	432	397	141

	1	2	3	4	5	6	1	2	3	4	5	6
4	156	578	447	0	554	212	418	122	797	0	108	495
5	353	732	118	373	0	283	115	154	1610	425	0	158
6	328	1071	387	352	199	0	167	214	2092	471	323	0
1	0	315	456	2340	187	73	0	375	319	558	745	183
2	581	0	195	2370	162	207	703	0	209	789	428	502
3	431	179	0	1090	233	248	496	234	0	481	109	508
4	301	58	56	0	124	170	237	1008	439	0	508	451
5	173	286	396	575	0	189	533	848	394	570	0	96
6	123	159	143	1753	411	0	288	202	386	729	653	0
1	0	1112	505	422	414	132	0	191	1623	264	433	90
2	627	0	560	99	227	86	422	0	455	240	101	418
3	373	2007	0	235	384	205	269	127	0	131	339	584
4	482	1638	262	0	233	129	275	272	834	0	477	551
5	223	1196	520	55	0	75	434	326	1526	810	0	569
6	200	782	271	292	235	0	276	327	1040	245	331	0
1	0	191	390	239	215	107	0	379	141	116	321	39
2	1868	0	126	448	271	108	1167	0	194	186	434	224
3	1870	121	0	116	256	19	1399	99	0	498	247	205
4	517	249	574	0	168	111	1718	289	308	0	281	86
5	1701	172	249	457	0	91	2474	127	122	180	0	51
6	1761	317	482	471	318	0	1466	117	142	568	404	0

Table 2 Ideal levels of factors for the proposed algorithms

Algorithm	Parameters	Definition of parameters	Ideal quantities
NSGA-II	Pop-size	Population size	200
	Pc	Cross-over rate	0.8
	Pm	Mutation rate	0.075
	Iteration	The number of iterations	500
NRGA	Pop-size	Population size	250
	Pc	Cross-over rate	0.8
	Pm	Mutation rate	0.075
	Iteration	The number of iterations	750
MOCSA	Inner-loop	Number of iterations on the inner-loop	20
	Outer-loop	Number of iterations on the outer-loop	300
	Alpha	Rate of temperature reduction	0.97
	Pop size	Population size	150

- Handling cost = [898 911 627 538 738 977];
- The number of handling systems type: 3;
- Number of handling system 1: 14;
- Number of handling system 2: 9;
- Number of handling system 3: 4;
- Variable cost = rand (0–10);
- Fixed cost = 500 + rand (0–100)
- Budget = 5000 + rand (0–1000);

Figure 1 shows the convergence of the best solutions approach (MOCSA) to the minimal Pareto front after the determined iterations. As can be seen, three Pareto fronts are illustrated by three colors. The blue front is the first front, the green front is the second one, and red front is the third front. It can be found that after the required iterations, the Pareto front is converged to the red front due to the minimization type of the model. Therefore, convergence of the method can be validated.

In addition, three proposed solution approaches are compared in terms of convergence to the optimal front. From Fig. 2, three Pareto fronts are illustrated for the proposed methods. As it can be seen, the Pareto front that

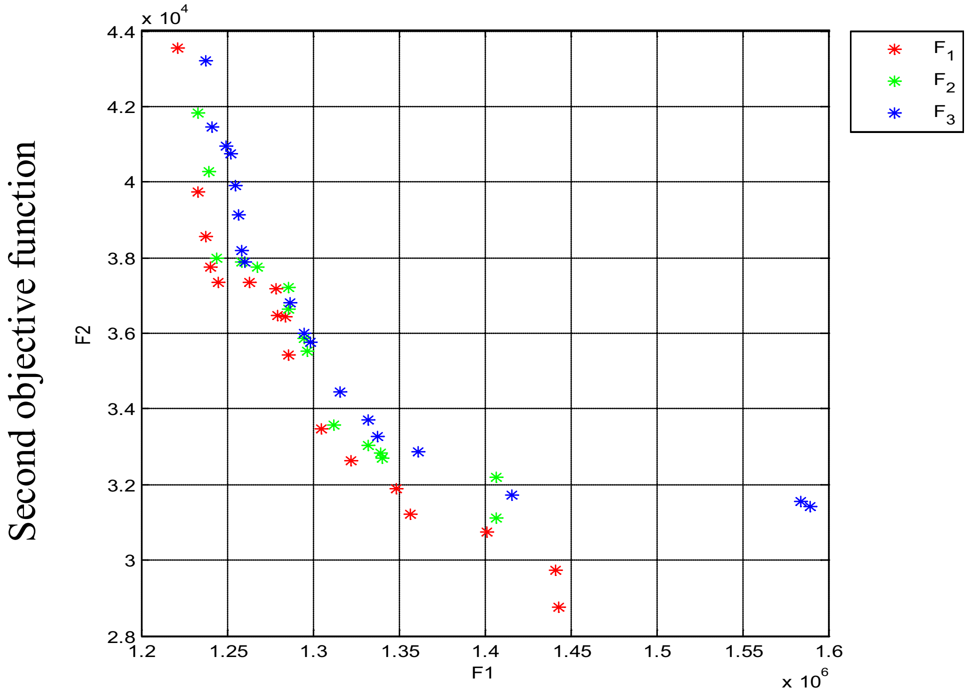


Fig. 1 The Pareto solutions of MOCSA algorithm for a test problem

is found by MOCSA method has the better performance against other methods due to finding the Pareto front with minimal costs.

4 Results analysis

In this section, we analyze the calculated results of the presented solution methods. To analyze the results, 32 samples of the problem were performed. After that, to evaluate the performance of algorithms, four criteria of comparison are presented. At last, the performance of the proposed algorithms is compared with each other based on the indices. Spacing, maximum spread, algorithm running time, and the number of Pareto solutions measures have been used for analyzing and comparing.

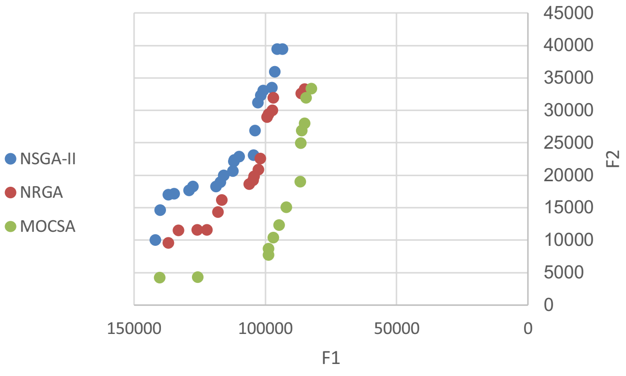


Fig. 2 Comparison of Pareto front in problem number 1

4.1 Comparison criteria

1. Spacing criteria: the algorithm is better which final non-dominate solutions have small spacing.

$$S = \sqrt{\frac{1}{n-1}\sum\nolimits_{i=1}^{n}\left(d_i - \bar{d}\right)^2} \tag{20}$$

2. Maximum diversity: in the case of the two objectives, this criterion is equal to the Euclidean distance between two border solutions in target space. The bigger this criterion, the better it will be.

$$D = \sqrt{\sum\nolimits_{j=1}^{m}(\max f_i^j - \min f_i^j)^2} \tag{21}$$

3. The number of Pareto's solution criterion: the quantity of NOS Criterion indicates the number of Pareto's optimal solutions that could be found in every algorithm.
4. Algorithm running time criterion

Pareto-based multi-objective algorithms were calculated and compared with above criterions for all generated experimental problems which results are shown in Tables 3, 4, 5, 6 and Figs. 3, 4, 5, 6. For evaluating the algorithms, first, all results have been normalized by using the relative percentage division (RPD). The RDP shows the distance between solutions and obtained the optimal solution at each algorithm.

After defining the standard criteria for comparing Pareto-based multi-objective algorithms, that criteria

Table 3 Results and RPD values for spacing criteria

Problem no.	Current value			RPD value		
	MOCSA	NRGA	NSGA-II	MOCSA	NRGA	NSGA-II
1	3395	4046	2743	23.7696	47.50273	0
2	4134	4144	3181	29.95913	30.2735	0
3	3449	3223	3666	7.012101	0	13.74496
4	4456	4402	3222	38.29919	36.62322	0
5	3773	5135	2664	41.62913	92.75526	0
6	2848	4336	3524	0	52.24719	23.73596
7	3810	3525	2910	30.92784	21.13402	0
8	2734	4605	2718	0.588668	69.42605	0
9	2925	4073	3969	0	39.24786	35.69231
10	2765	3870	2965	0	39.96383	7.233273
11	2852	4380	3947	0	53.57644	38.39411
12	2600	4020	3338	0	54.61538	28.38462
13	3405	4238	2802	21.52034	51.24911	0
14	2582	4085	2572	0.388802	58.82582	0
15	2822	4604	4161	0	63.1467	47.44862
16	2871	4520	2970	0	57.43643	3.448276
17	6725	7015	6998	0	4.312268	4.05948
18	6668	6235	7520	6.944667	0	20.60946
19	7375	7774	7241	1.850573	7.360862	0
20	7144	7051	7397	1.318962	0	4.907105
21	5592	6575	7729	0	17.57868	38.21531
22	6020	6992	7076	0	16.14618	17.54153
23	7470	8217	6026	23.96283	36.35911	0
24	7908	7280	6856	15.34422	6.184364	0
25	9561	12,286	10,742	0	28.5012	12.35226
26	9087	11,277	7677	18.36655	46.89332	0
27	10,922	12,055	10,688	2.189371	12.79004	0
28	8993	11,090	10,804	0	23.31814	20.13789
29	10,009	9516	7106	40.8528	33.915	0
30	9596	10,321	10,693	0	7.555231	11.43185
31	8802	11,634	7279	20.9232	59.82965	0
32	8752	11,550	9911	0	31.96984	13.24269

are calculated for all generated experimental samples in Table 3, 4, 5 and 6. To evaluate the algorithms, all the results are normalized by using the relative percentage division (RPD). The RPD shows the distance between solutions and obtained optimal solution at each algorithm. RDP is calculated according to the formula:

RDP is calculated according to the formula:

$$RPD_{ij} = \left|\left(sol_{ij} - sol_{j,best}\right)\right| / sol_{j,best} \tag{22}$$

i is the number of the algorithm and j is the problem number. The Tables 3, 4, 5 and 6 show the RPD values for all algorithms.

Figure 3 shows the performance of Algorithms in the spacing index. It can be concluded that the NSGA-II algorithm has the best performance among other algorithms because this criterion should be as small as possible. Furthermore, in some cases, the MOCSA had high quality. Figure 4 shows the algorithm efficiency in the index of the maximum spread. In this index, any of the algorithms is not superior to the other. Figure 5 shows the performance of proposed algorithms in the index of NOS. It is obvious that almost in all of the samples, NSGA-II performs better and can find the most solution at the first forefront of Pareto. Figure 6 shows the superiority of the proposed algorithm (MOCSA) to NSGA-II and NRGA at the run time. According to this figure, the MOCSA algorithm in the small size problems performs the same as the other two algorithms, but as the problem size grows, the performance of the MOCSA algorithm highly increases so that it creates a significant difference with two other algorithms, especially NSGA-II.

Table 4 Results and RPD values for maximum diversity criteria

Problem no.	Current value			RPD value		
	MOCSA	NRGA	NSGA-II	MOCSA	NRGA	NSGA-II
1	122,842	104,863	131,950	6.902615	20.52823	0
2	123,993	100,532	139,882	11.35886	28.13085	0
3	138,844	110,587	127,107	0	20.35162	8.453372
4	137,233	104,633	132,819	0	23.75522	3.216428
5	113,978	98,775	121,549	6.228764	18.73648	0
6	110,229	103,611	134,105	17.80396	22.7389	0
7	118,093	92,749	132,221	10.68514	29.85305	0
8	122,491	104,779	137,238	10.74557	23.65161	0
9	119,652	99,169	112,609	0	17.11881	5.886237
10	100,840	87,210	88,307	0	13.51646	12.4286
11	101,088	106,056	81,734	4.684318	0	22.93317
12	112,357	110,646	107,583	0	1.522825	4.248956
13	127,176	57,686	119,338	0	54.64081	6.163113
14	108,242	74,118	109,041	0.732752	32.0274	0
15	121,239	75,795	96,993	0	37.48299	19.99852
16	115,758	95,772	114,340	0	17.26533	1.224969
17	121,756	89,668	71,450	0	26.35435	41.31706
18	115,791	67,022	62,385	0	42.11813	46.12276
19	115,980	99,199	84,853	0	14.46887	26.83825
20	97,464	88,288	93,712	0	9.414758	3.849627
21	107,398	90,967	91,796	0	15.29917	14.52727
22	92,127	63,555	91,175	0	31.01371	1.033356
23	96,905	63,818	87,505	0	34.14375	9.700222
24	84,310	99,737	70,194	15.46768	0	29.6209
25	71,523	107,759	82,797	33.62689	0	23.16465
26	99,344	92,669	54,614	0	6.719077	45.02537
27	119,957	98,780	72,660	0	17.65383	39.4283
28	90,167	57,647	87,306	0	36.06641	3.173001
29	104,660	81,144	76,897	0	22.46895	26.52685
30	56,133	111,726	53,196	49.75834	0	52.38709
31	66,430	106,621	110,736	40.01048	3.716045	0
32	57,543	86,517	90,598	36.48535	4.504514	0

In order to evaluate and compare precisely, the statistical analyses were used especially variance analysis. In such a way that the algorithms were variance analyzed by software in relation to any criteria and the results were analyzed. Less than 0.05% of *P* value indicates a significant difference between the responses of the two algorithms in relation to specific criteria, otherwise, it can be said that there is no significant difference between the performances of two algorithms in relation to that criteria. The report of software output of variance analysis, it was observed that the quantities of *P* values are less than 0.05% in all criteria, so according to statistical output, there are significant differences between the algorithms in the Spacing, Diversity, NOS and run time criteria.

Also in order to determine the efficiency of algorithms in the indexes that have a significant difference, the 95% confidence intervals test are used and the results of that are shown at the Figs. 7, 8, 9 and 10 in the form of confidence intervals. We analyze the charts considering the fact that the performance of the algorithm at the concerned index are better when RPD amount is less and as much nearer to zero.

As shown in Fig. 7, the average of the results of the NSGA-II and MOCSA algorithm is close together, but it is far away from the average results of the NRGA. Thus, according to the analysis results, the performance of the NSGA-II and MOCSA algorithm is better at the spacing index.

Table 5 Results and RPD values for number of Pareto's solution criteria

Problem no.	Current value			RPD value		
	MOCSA	NRGA	NSGA-II	MOCSA	NRGA	NSGA-II
1	24	30	45	46.66667	33.33333	0
2	27	36	54	50	33.33333	0
3	30	30	39	23.07692	23.07692	0
4	42	33	63	33.33333	47.61905	0
5	33	42	48	31.25	12.5	0
6	36	42	39	14.28571	0	7.142857
7	33	27	36	8.333333	25	0
8	42	30	36	0	28.57143	14.28571
9	45	27	39	0	40	13.33333
10	24	21	33	27.27273	36.36364	0
11	45	42	48	6.25	12.5	0
12	24	21	42	42.85714	50	0
13	24	33	54	55.55556	38.88889	0
14	27	42	54	50	22.22222	0
15	21	33	54	61.11111	38.88889	0
16	24	33	51	52.94118	35.29412	0
17	39	27	57	31.57895	52.63158	0
18	33	33	51	35.29412	35.29412	0
19	30	27	63	52.38095	57.14286	0
20	45	21	39	0	53.33333	13.33333
21	42	36	42	0	14.28571	0
22	30	30	51	41.17647	41.17647	0
23	21	36	60	65	40	0
24	36	39	36	7.692308	0	7.692308
25	33	36	48	31.25	25	0
26	45	33	54	16.66667	38.88889	0
27	24	33	54	55.55556	38.88889	0
28	36	36	33	0	0	8.333333
29	39	27	39	0	30.76923	0
30	36	39	54	33.33333	27.77778	0
31	45	21	33	0	53.33333	26.66667
32	39	24	33	0	38.46154	15.38462

According to Fig. 8, the results obtained from 95% confidence interval test for the maximum spread criteria showed the performance of the proposed algorithm MOCSA is better than the two other algorithms.

Also according to the Fig. 9, the performance of the NSGA-II is much better than the other two algorithms in optimal Pareto's queue solutions number. The performance of the proposed algorithm is also slightly better than NRGA.

According to the results of variance analysis in Fig. 10, the performance of the proposed algorithm is much better than the two other algorithms in the run time index.

The applicability of the research in this paper can be discussed in two ways in terms of model and methodology as follows:

The Proposed model Facility layout issues are often modeled as a Quadratic Assignment Problem (QAP which is a problem that uses not only for the planning of the firm layout, but also uses for the planning of hospitals and universities layout, and even the keyboard layout. According to Francis and his colleagues' research [29], the QAP computation is Intractable for a problem with more than 15 to 20 departments. The demand changes in different periods, so we need to develop the model for the dynamic state which the flow of materials is predicted on each period (Balakrishnan and Cheng [10]) and also the transportation costs are determined based on that. To determine these costs exactly, it is necessary to consider the different handling machines type., in this model, we are going to create conditions for decision-makers to they have all the options ahead and can easily make their own choices based on

Table 6 Results and RPD values for running time criteria

Problem no.	Current value			RPD value		
	MOCSA	NRGA	NSGA-II	MOCSA	NRGA	NSGA-II
1	404	428	498	0	5.940594	23.26733
2	470	468	442	6.334842	5.882353	0
3	500	390	376	32.97872	3.723404	0
4	378	518	368	2.717391	40.76087	0
5	466	504	474	0	8.154506	1.716738
6	442	470	402	9.950249	16.91542	0
7	426	460	496	0	7.981221	16.43192
8	450	362	418	24.30939	0	15.46961
9	732	1096	1122	0	49.72678	53.27869
10	664	1126	1338	0	69.57831	101.506
11	990	1064	1208	0	7.474747	22.0202
12	904	1008	1282	0	11.50442	41.81416
13	864	1220	944	0	41.2037	9.259259
14	892	788	934	13.19797	0	18.52792
15	898	1066	820	9.512195	30	0
16	616	658	1202	0	6.818182	95.12987
17	2056	2590	3272	0	25.97276	59.14397
18	2066	2270	2566	0	9.874153	24.20136
19	2106	2230	2618	0	5.887939	24.31149
20	1926	2582	3268	0	34.06023	69.67809
21	1924	2244	2644	0	16.63202	37.42204
22	1704	2046	2850	0	20.07042	67.25352
23	2118	2132	2404	0	0.661001	13.50331
24	2108	2218	3096	0	5.218216	46.86907
25	8579	8441	15,852	1.634877	0	87.79765
26	9059	11,883	14,330	0	31.17342	58.18523
27	8472	9391	13,955	0	10.8475	64.71907
28	8163	10,543	13,387	0	29.15595	63.99608
29	7567	9226	15,991	0	21.92414	111.3255
30	8189	11,374	13,444	0	38.89364	64.17145
31	9049	9031	15,099	0.199313	0	67.19079
32	8402	11,458	15,137	0	36.37229	80.15949

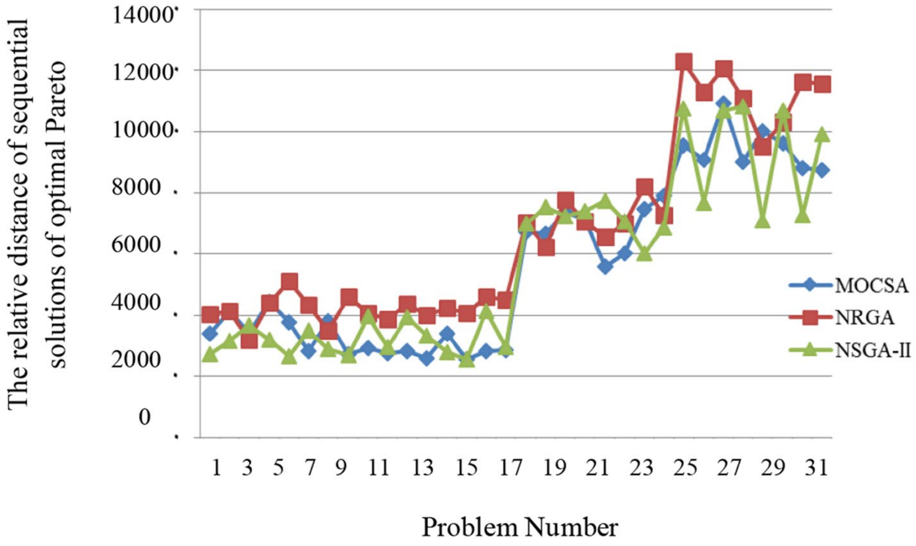


Fig. 3 Efficiency comparison of the proposed algorithms in a spacing criterion

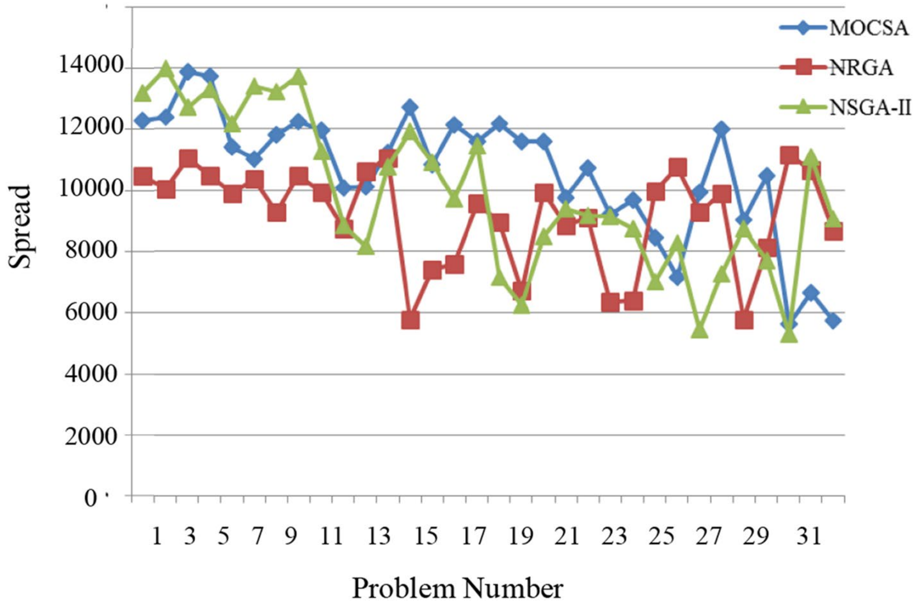


Fig. 4 Efficiency comparison of the proposed algorithms in the maximum spread criterion

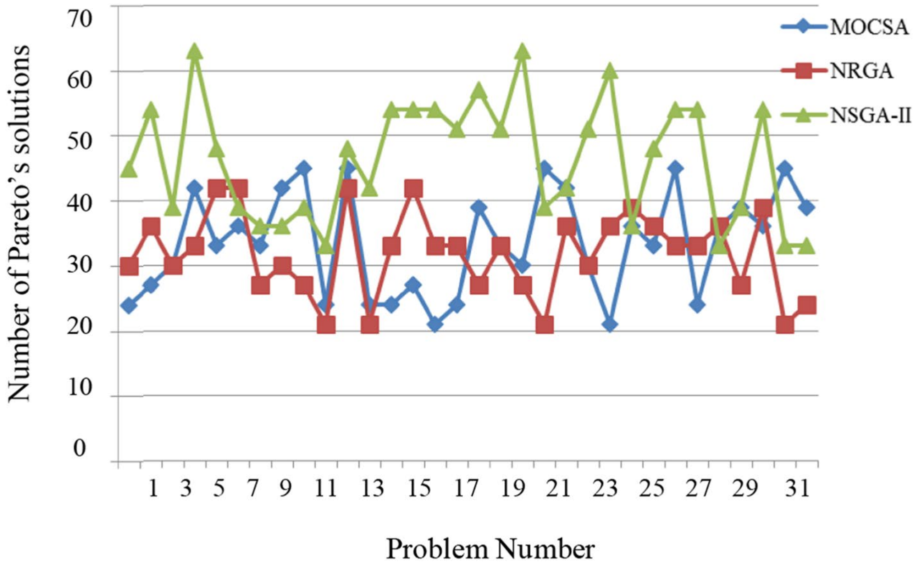
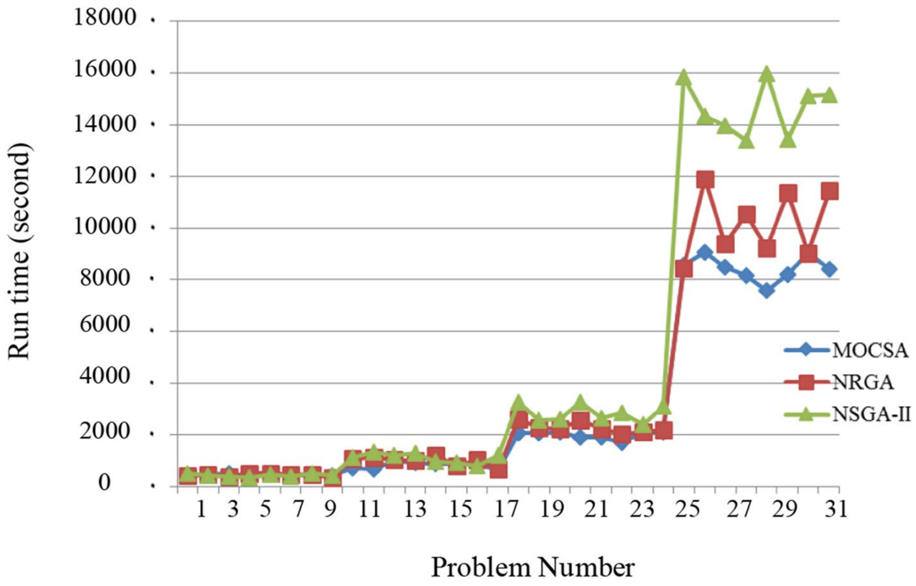


Fig. 5 Efficiency comparison of the proposed algorithms in the number of Pareto's solutions criterion

their goals. A large budget is required for changing layout. Therefore, regarding the budget in modeling and solving the problem is one of the realistic aspects of the problem. On the other hand, re-arrangement cost and the material handling cost are a different matter and it should be considered in the manager decision. So we considered those as two distinct functions in the proposed model. If these two types of goals have not the same Currency value, the higher cost will cover the lower cost and will not have a significant impact on decision-makers views. Also, in this modeling, decision-makers can choose best carriers by analyzing the variable and fixed costs of the handling system during the planning horizon. For example, a robot may have a higher fixed cost than a forklift, but its variable cost is much lower, so managers and decision makers can decide easily by considering their planning and priorities.

The Proposed methodology DCFP is an intractable problem, so because of that complexity, developing a Meta-heuristic algorithm is necessary. Simulated annealing is a local search algorithm that can pass the local optimal. This Meta-heuristic algorithm has become more used in recent years because it is used easily and it can cross from the local optimal by using the Hill-climbing. Also it applicable to discrete problems and a few continuous problems. Although the annealing temperature changes randomly which this discrepancy reduces the search variation, the cloud theory

Fig. 6 Efficiency comparison of run time of proposed algorithms



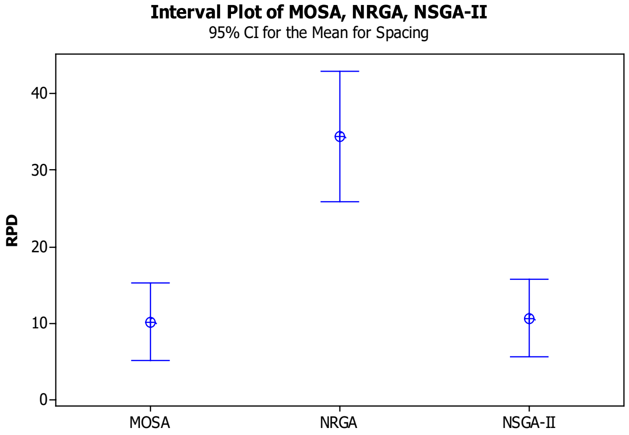


Fig. 7 The results of variance analysis for the spacing indicator

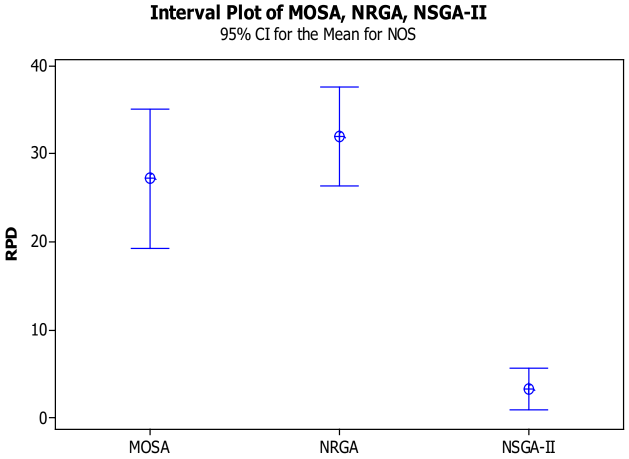


Fig. 9 Results of variance analysis to Pareto's solution number index

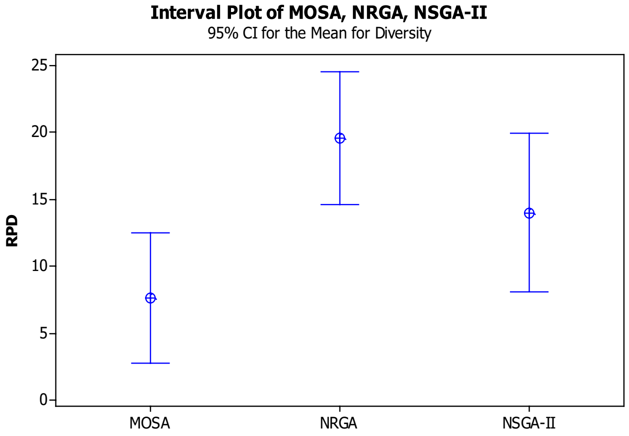


Fig. 8 The results of variance analysis for the maximum spread

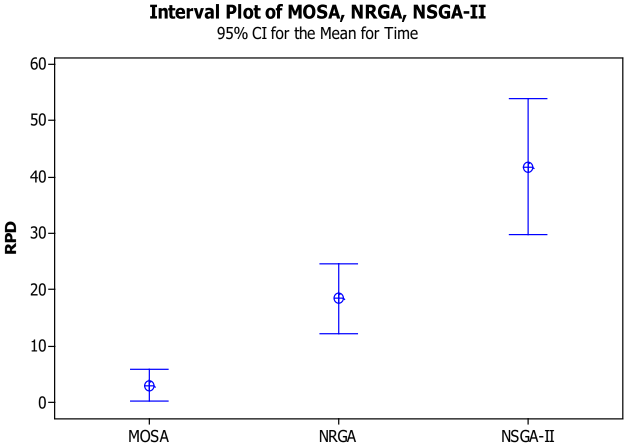


Fig. 10 The results of variance analysis for run time index

is used to produce a continuous annealing temperature. The cloud theory has a random attribute and tends to be stable. So, in the real world, this solving method gives better solutions.

In today's manufacturing environment, the facility layout needs to be sufficiently adaptable to changes in product design, process design, and flexible scheduling in order to stay in the competition. Therefore, I think this model and method are not only applied to actual problems but also it has high flexibility. I hope this method be useful in real world.

5 Conclusion

As we have seen, the main objective of this research is developing the mathematical model as a two-objective problem by considering the transportation costs at objective function, also applying budget and type of transporter limitation simultaneously and presenting a solving method for dynamic layout facilities problem. Since the exact methods often are not suitable for large size problems, proposed heuristic and meta-heuristic approaches have been developed. The proposed algorithm is a combination of simulated annealing and cloud theory. The performance of the proposed algorithm has been compared with the algorithms present in the literature and the results indicate the effectiveness of the proposed algorithm. In order to compare this issue, first of all the proposed algorithm parameters placed on its best quantities by using the Taguchi method, so that algorithm will have the maximum efficiency; then the random sample problems have been generated and they have been solved by the proposed algorithm and results were compared with the results of two other algorithms. The results show that the NSGA-II performs significantly better than the other two algorithms in the number of Pareto's solutions criteria. While the performance of MOCSA and NRGA are very close together. NSGA-II and MOCSA perform the same way at the spacing criteria too. However, NRGA performs weaker than two other algorithms. At the maximum spread criterion, MOCSA performs significantly better than NRGA. In this criterion, NSGA-II and NRGA have not a significant difference. Also, although two NSGA-II and MOCSA algorithms perform statistically the same, we can say that the MOCSA algorithm performs better. MOCSA performs highly better than the two other algorithms at run time criterion.

Finally, even though this study concern directly about the production sites, but its results can be used to other arrangements especially Office Layout because it is a generic search.

This research concerns only a small portion of the combined problems about facility layout, so further researches in this field can be done. Among the fields to be continued in this research are as follows:

- Considering the size of the facility unequally and solving it by the proposed algorithm
- Taking account the intracellular flows and entry and exit points of each cell
- Taking account random parameters of layout problem in a dynamic state
- Considering the time value of money in various periods as well as the fuzzy costs.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. McKendall AR, Hakobyan A (2010) Heuristics for the dynamic facility layout problem with unequal-area departments. Eur J Oper Res 201(1):171–182. https://doi.org/10.1016/j.ejor.2009.02.028
2. Koopmans TC, Beckmann M (1957) Assignment problems and the location of economic activities. Econom J Econom 25(1):53–76. https://doi.org/10.2307/1907742
3. Rosenblatt MJ (1986) The dynamics of plant layout*. Manag Sci 32(January):76–87. https://doi.org/10.1287/mnsc.32.1.76
4. Kulturel-Konak S (2007) Approaches to uncertainties in facility layout problems: perspectives at the beginning of the 21st century. J Intell Manuf 18(2):273–284. https://doi.org/10.1007/s10845-007-0020-1
5. Lacksonen TA, Emory Enscere E (1993) Quadratic assignment algorithms for the dynamic layout problem. Int J Prod 31(3):503–517
6. Urban TL (1993) A heuristic for the dynamic facility layout problem. IIE Trans 25(4):57–63. https://doi.org/10.1080/07408179308964304
7. Conway DG, Venkataramanan MA (1994) Genetic search and the facility layout problem. Comput Oper Res 21(8):955–960
8. Kochhar JS, Heragu SS (1999) Facility layout design in a changing environment. Int J Prod Res 37(11):2429–2446. https://doi.org/10.1080/002075499190590
9. Burkard RE, Rendl F (1984) A thermodynamically motivated simulation procedure for combinatorial optimization problems. Eur J Oper Res 17(2):169–174. https://doi.org/10.1016/0377-2217(84)90231-5
10. Balakrishnan J, Cheng CH, Conway DG, Lau CM (2003) A hybrid genetic algorithm for the dynamic plant layout problem. Int J Prod Econ 86(2):107–120. https://doi.org/10.1016/S0925-5273(03)00027-6
11. Lee YH, Lee MH (2002) A shape-based block layout approach to facility layout problems using hybrid genetic algorithm. Comput Ind Eng 42:237–248

12. Dunker T, Radons G, Westkämper E (2005) Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem. Eur J Oper Res 165(1):55–69. https://doi.org/10.1016/j.ejor.2003.01.002
13. Kaku BK, Mazzola JB (1997) A Tabu-search heuristic for the dynamic plant layout problem. INFORMS J Comput 9(4):374–384. https://doi.org/10.1287/ijoc.9.4.374
14. Şahin R, Türkbey O (2009) A new hybrid Tabu-simulated annealing heuristic for the dynamic facility layout problem. Int J Prod Res 47(24):6855–6873. https://doi.org/10.1080/00207540802376323
15. McKendall AR, Liu W-H (2012) New Tabu search heuristics for the dynamic facility layout problem. Int J Prod Res 50(3):867–878. https://doi.org/10.1080/00207543.2010.545446
16. Baykasoğlu A, Gindy NNZ (2001) A simulated annealing algorithm for dynamic layout problem. Comput Oper Res 28(14):1403–1426. https://doi.org/10.1016/S0305-0548(00)00049-6
17. McKendall AR, Shang J, Kuppusamy S (2006) Simulated annealing heuristics for the dynamic facility layout problem. Comput Oper Res 33(8):2431–2444. https://doi.org/10.1016/j.cor.2005.02.021
18. Azimi P (2010) An efficient heuristic algorithm for the traveling salesman problem. In: Proceedings of 8th Heinz Nixdorf symposium on advanced manufacturing and sustainable logistics. Paderborn University, Germany
19. Yu-Hsin Chen G (2013) A new data structure of solution representation in hybrid ant colony optimization for large dynamic facility layout problems. Int J Prod Econ 142(2):362–371. https://doi.org/10.1016/j.ijpe.2012.12.012
20. Xu J, Song X (2015) Multi-objective dynamic layout problem for temporary construction facilities with unequal-area departments under fuzzy random environment. Knowl Based Syst 81:30–45. https://doi.org/10.1016/j.knosys.2015.02.001
21. EVANSt GW, Wilhelm MR, Karwowskij W (1987) A layout design heuristic employing the theory of fuzzy sets. Int J Prod Res 25:1431–1450. https://doi.org/10.1080/00207548708919924
22. Grobelny J (1987) The fuzzy approach to facility layout problems. Fuzzy Sets Syst 23:175–190
23. Raoot AD, Rakshit A (1991) "A 'fuzzy' approach to facilities lay-out planning. Int J Prod Res 29:835–857. https://doi.org/10.1080/00207549108930105
24. Gen M, Ida K, Cheng C (1995) Multi row machine layout problem in fuzzy environment using genetic algorithms. Comput Ind Eng 29(1–4):519–523
25. Dweiri F, Meier FA (1996) Application of fuzzy decision-making in facilities layout planning. Int J Prod Res 34(11):3207–3225. https://doi.org/10.1080/00207549608905085
26. Aiello G, Enea M (2001) Fuzzy approach to the robust facility layout in uncertain production environments. Int J Prod Res 39(18):4089–4101. https://doi.org/10.1080/00207540110061643
27. Deb SK, Bhattacharyya B (2005) Fuzzy decision support system for manufacturing facilities layout planning. Decis Support Syst 40(2):305–314. https://doi.org/10.1016/j.dss.2003.12.007
28. Garey C, Rogers J (2009) Managing dynamic facility layout with multiple objectives. PICMET proceeding, August 2–6, Portland, OR, USA
29. Francis RL, McGinnis LF, White JA (1992) Facility layout and layout: an analytical approach. Prentice Hall, Englewood Cliffs
30. Deb K (2002) Multi objective using evolutionary algorithms. Wiley, New York
31. Al Jadaan O, Rao CR, Rajamani L (2006) Parametric study to enhance genetic algorithm performance, using ranked based roulette wheel selection method. InSciT2006 2:274–278

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.