RAIRO-Oper. Res. 50 (2016) 869–890
DOI: 10.1051/ro/2016057

RAIRO Operations Research
www.rairo-ro.org

A NOVEL META-HEURISTIC ALGORITHM FOR MULTI-OBJECTIVE DYNAMIC FACILITY LAYOUT PROBLEM

Seyed Shamsodin Hosseini[1] and Mehdi Seifbarghy[2]

Abstract. This paper proposes an integrated approach for dynamic facility layout problem considering the material handling equipment (MHE). The objectives of this problem are minimization of the fixed costs of MHE, minimization of material handling cost (MHC) and minimization of machine rearrangement costs (MRC). To be more realistic, MHE fixed costs, MRC and MHC, which might be of different importance to decision maker, are considered separately in three objective functions. An integrated model is proposed which is able to simultaneously select the MHE along with the arranging and re-arranging facilities. The model belongs to the class of multi-objective nonlinear mathematical programming models. Considering the NP-hard nature of the model and inspiring the existing water flow like algorithm, a novel Pareto-based meta-heuristic algorithm called multi-objective water flow like algorithm (MOWFA)is developed to solve the problem. We comprehensively discuss the parameter tuning of the algorithms utilizing Taguchi method. Finally, the performance of the proposed MOWFA is evaluated against two well-known meta-heuristic algorithms called non-dominated sorting genetic algorithm (NSGA-II) and non-dominated ranking genetic algorithm (NRGA). Computational results indicate the greater efficiency of the algorithm compared to the two addressed algorithms for solving the given multi-objective problem.

Mathematics Subject Classification. 90B50, 90B80.

Received December 4, 2015. Accepted August 30, 2016.

1. Introduction

Facility layout problem is concerned with determining the efficient layout of machines, cells, or departments. There are several factors which can directly or indirectly affect the efficiency of a layout, among which the most important factors are: variation on demand, adding or removing a product, changing the manufacturing methods and replacing the equipments. All the factors may affect the patterns of material handling between two machines as well as material handling costs (MHC). According to Tompkins *et al.* [1], around 20−50 percent of the total operating costs and 15−70 percent of the total manufacturing costs are concerned with MHC. MHC is known as the most important metric to determine the effectiveness of a layout and is often considered as the single objective of facility layout problems. Due to the variations in product demand and changes in product

Keywords. Multi-objective dynamic facility layout problem, water flow like algorithm, manufacturing facility, material handling, Taguchi method.

[1] Planning Directorate, National Iranian Oil Products Distribution Company, Qazvin, Iran.

[2] Department of Industrial Engineering, Alzahra University, Deh Vank, Tehran, Iran. M.Seifbarghy@alzahra.ac.ir

mix, MHC fluctuates and often increases. This may cause the current layout to lose its efficiency and make re-layout of facilities necessary.

Changes in the current layout should be investigated carefully since these may result in high machine-rearrangement costs (MRC). The problem of preparing the layout and re-layout of the machines in such a way as to minimize MHC and MRC is called dynamic facility layout problem (DFLP). In DFLP, a planning horizon divided into a number of periods, is assumed. Any changes which results in changing MHC, gives the signal of beginning a new period. The objective of DFLP is to determine a layout in each period of horizon planning so that the sum of MHC and MRC is minimized. Apart from the layout of the machines, material batch size is another factor which can affect MHC. Material batch size can reduce the number of the required trips to transfer materials between two machines. The batch size depends on the capacity of material handling equipment (MHE). There are different MHEs such as forklifts, trucks and automated guided vehicles (AGVs) with different capacity used to transport materials between machines.

Thus, there is a close relation between layout of machines and the selected transporters. The final solution is highly dependent on the given part of the problem; in other words, by selecting the best MHE based on an existing layout, or determining the optimal layout based on pre-specified MHE, a considerable degree of freedom may be lost in obtaining an overall optimal design (Chittratanawat [2]); thus, it is recommended to study the problem of allocating MHE to machines in order to handle the materials and solve the problem of machine layout as an integrated problem. As a result, this paper proposes an integrated approach for DFLP considering MHE. To be more realistic, the regular costs of material handling, the fixed costs of establishing MHE and MRC are considered in three separate objective functions. The main reason of this segregation is the nature of these costs which are significantly important to decision makers. Since the behavior of these objectives are not monotonic, the problem falls into the class of multi-objective optimization problems (MOOP). The proposed MOOP is mathematically modeled as a non-linear integer programming model. The mathematical model belongs to the class of NP-hard problems. Soft computing techniques, specifically evolutionary computations, are usually employed to find near-optimum solutions. This paper introduces a multi-objective water flow like algorithm (MOWFA) as an efficient multi-objective evolutionary algorithm to solve the problem. The performance of MOWFA is compared with two popular algorithms including non-dominated sorting genetic algorithm (NSGA-II) and non-dominated ranking genetic algorithm (NRGA). The rest of the paper is structured as follows: Section 2 gives a brief review of the previous studies; then, the mathematical formulation of the problem is introduced in Section 3. Thereafter, Section 4 describes the proposed methodologies to solve the multi-objective dynamic facility layout problem (MODFLP). The parameter tuning and computational results for the proposed algorithm are presented in Section 5. Finally, the conclusion and ideas for future research are given in Section 7.

2. A brief literature review

Layout researches can be categorized into two types of static or dynamic. In static layout, material flow between machines is constant and an optimal layout is designed for a single time period. On the contrary, if the layout is evaluated and modified occasionally with respect to material flow changes, it is called dynamic layout. We give a brief review of the researches regarding the dynamic category. Rosenblatt [3] was the first one who addressed the problem of dynamic facility layout. Rosenblatt [3] developed a procedure based on dynamic programming to determine the optimum design which takes into consideration both MHC and MRC, for multiple periods. In recent years, there have been several efforts to address the DFLP and the research in this area typically hires heuristic and meta-heuristics solution techniques. Urban [4] suggested a heuristic steepest-descent pair-wise interchange procedure combined with the concept of forecast windows similar to CRAFT in order to solve the DFLP. Kouvelis and Kiran [5] proposed a developed dynamic programming to consider the dynamic aspects in automated manufacturing systems. A heuristic tabu search was proposed by Kaku and Mazzola [6] for solving the DFLP. Their heuristic technique was based on initializing a solution to obtain the feasible one. The whole method is divided into two stages: 1. diversifying search and 2. intensifying the search to identify the best solution among the promising solutions identified in stage one. The experimental results showed that tabu search is effective in providing high value solutions for DFLP.

The genetic algorithm was first employed by Conway and Venkataramanan [7] to solve DFLP. The developed methodology evaluated the appropriateness of GA to generate feasible layouts. Balakrishnan and Cheng [8] proposed an improvement in application of GA procedures to solve DFLP. They adopted a different crossover and mutation operators and also used a new generational replacement strategy to help increase population diversity. The computational study showed that their proposed GA was quite more effective than GA proposed by [7]. Baykasoglu and Gindy [9] denoted the simulated annealing (SA) method to solve the DFLP. Their approach was a straightforward implementation of SA to solve the DFLP. Using the test problems from [8], Baykasoglu and Gindy [9] showed that the SA is more capable than developed GAs to achieve better solutions.

Mckendall and Shang [10] proposed three hybrid ant systems (HAS) for solving the DFLPs. In the first system, a HAS with a pair-wise exchange style was used to improve the solution quality. In the second one named modified heuristic HAS, the pair-wise exchange heuristic was replaced by an SA procedure. In the third heuristic system, a look-ahead/look-back strategy was also used. Furthermore, McKendall *et al.* [11] offered two SA methods. The first SA heuristic method called SAI, is a direct adaptation of SA for the DFLP. The second SA heuristic method (SAII) is just like the SAI, except that it has an added look-ahead/look-back strategy. Krishnan *et al.* [12] presented a new tool "Dynamic From Between Chart" to analyze the redesigning layouts. It models the production rate changes using a continuous function.

Rezazadeh *et al.* [13] applied an extended particle swarm algorithm (PSO) for the DFLP. Balakrishnan and Cheng [14] studied the performance of algorithms in static and rolling horizons, under forecast uncertainty for the DFLP. Sahin and Turkbey [15] also suggested a novel hybrid meta-heuristic algorithm based on the SA approach supplemented with a tabu list. Mckendall and Liu [16] proposed three tabu search (TS) heuristics for DFLP. The first heuristic was a simple TS heuristic. The second heuristic added diversification and intensification strategies to the first one, while the third one was a probabilistic TS heuristic. Chen [17] proposed a new encoding and decoding scheme for solution representation within ant colony algorithm framework. It revealed the significant impact of solution representation on the efficiency of heuristics in terms of computational time.

It is only recently that researchers have been proposing multi-objective approaches for DFLP. Chen and Rogers [18] were the first ones who proposed a multi-objective dynamic facility layout model to search about several features of the facility layout problems such as time and distance-based objective as well as the adjacency-based objective. They applied a meta-heuristic optimization algorithm called ant colony optimization to solve the MODFLP. Their results indicated this heuristic technique provides the DFLP with a practical decision support tool.

Jolai *et al.* [19] consider a multi-objective DFLP with unequal fixed size departments and pick up/drop off locations. Their objectives were to minimize the MHC and the MRC and maximize the total adjacency and distance rate between facilities. They implemented a multi-objective PSO algorithm to solve the problem. Emami and Nookabadi [20] proposed a model in which both quantitative and qualitative analyses of dynamic facility layout problems were simultaneously taken into account. In their model, the re-arrangement and material handling costs have been considered as two distinct functions. Similar to the qualitative objective, the adjacency-based objective also aims at maximizing the adjacency scores of the facilities.

Most of the previous studies have focused only on minimizing the sum of MHC and MRC as the most important performance criteria; however, the transporters type used to move the materials between the machines has not been taken into consideration. On the other hand, in the researches available in the literature, the importance of MHC and MRC are identical while they may be of different importance to decision makers. To eliminate these types of shortcomings, this research takes the transporters type into account and proposes a multi-objective model to capture the costs separately. Furthermore, a novel powerful meta-heuristic algorithm is devised to optimize the proposed multi-objective problem. All of these contribute to make our approach more realistic and applicable for solving the layout problem of manufacturing systems.

3. Mathematical model

In this section, the integrated problem is formulated as a multi-objective non-linear integer programming model to minimize the sum of MHC, MHE fixed costs and MRC. In order to help a better understanding of the model presented in the paper, the assumptions, parameters and the decision variables are first defined as follows:

The problem is formulated under the following assumptions:

1. The material flow between machines is dynamic and predetermined.
2. There is a potential set of MHE with predefined fixed costs.
3. The sizes of the machines and locations are equal.
4. The distances between locations are known in advance.

The parameters and indices are:

N	The number of machines/locations;
T	The number of periods in the planning horizon;
TR	The number of available transporters;
i, j, k, l	Index of machines/locations;
t	Index of time periods;
tr	Index of transporters;
A_{tijl}	The cost of shifting machine i from location j to l in period t;
$C_{t,i,k}^{tr}$	The material transporting cost between machine i and machine k in period t by transporter tr;
m_t^{tr}	The maximum number of vehicles tr available in period t;
cap^{tr}	Transport capacity of transporter tr;
$R_{t,i,k}$	1 if any material is transported between machine i and machine k in period t 0 otherwise;
FI^{Tr}	Fixed cost of establishing of transporter tr;
AT_t^{tr}	The available time of transporter tr in period t;
$Time_{t,i,k}^{tr}$	The time is required to complete a tour from machine i to machine k by transporter tr in period t;
$AvgT_{j,l}^{tr}$	The average time is required to complete a tour from location j to location l by transporter tr;
$F_{t,i,k}$	The material flow from machine i to machine j in period t;
$D_{t,j,l}$	The distance between location j and location l in period t.

The decision variables are:

$X_{t,i,j}$	1 if facility i is allocated to location j in period t
	0 otherwise;
$Y_{t,i,k}^{tr}$	1 if transporter tr is selected to handle material from machine i to machine k in period t
	0 otherwise;

The proposed multi-objective dynamic facility layout problem is now formulated as a non-linear integer programming model. This formulation is an extension of the model presented by McKendall *et al.* [10]:

$$\text{Min } Z_1 = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N}\sum_{t=1}^{T}\sum_{tr=1}^{TR}\left(\frac{F_{t,i,k}}{\text{cap}^{tr}}\right) \times D_{t,j,l} \times C_{t,i,k}^{tr} \times X_{t,i,j} \times X_{t,k,l} \times Y_{t,i,k}^{tr} \tag{3.1}$$

$$\text{Min } Z_2 = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{l=1}^{N}\sum_{t=2}^{T} X_{(t-1)ij} \times X_{til} \times A_{t,i,j,l} \tag{3.2}$$

$$\text{Min } Z_3 = \sum_{t=1}^{T} \sum_{i=1}^{N-1} \sum_{k=i+1}^{N} \sum_{tr=1}^{Tr} Y_{t,i,k}^{tr} \times FI^{tr} \tag{3.3}$$

$$\sum_{j=1}^{N} X_{tij} = 1 \quad , \ \forall i = 1, 2, \ldots, N, \quad \forall t = 1, 2, \ldots, T \tag{3.4}$$

$$\sum_{i=1}^{N} X_{tij} = 1 \quad , \ \forall j = 1, 2, \ldots, N, \quad \forall t = 1, 2, \ldots, T \tag{3.5}$$

$$\sum_{tr=1}^{TR} Y_{t,i,k}^{tr} = R_{t,i,k} \ , \ \forall i = 1, 2, \ldots, N-1; \ \forall k = i+1, 2, \ldots, N; \ \forall t = 1, 2, \ldots, T \tag{3.6}$$

$$Y_{t,i,k}^{tr} = Y_{t,k,i}^{tr} \quad , \quad \forall i, k = 1, 2, \ldots, N; \forall t = 1, 2, \ldots, T; \quad \forall tr = 1, 2, \ldots, TR \tag{3.7}$$

$$\sum_{i=1}^{N-1} \sum_{k=i+1}^{N} Y_{t,i,k}^{tr} \leqslant m_t^{tr} \ , \quad \forall tr = 1, 2, \ldots, TR; \ \forall t = 1, 2, \ldots, T \tag{3.8}$$

$$\sum_{i=1}^{N} \sum_{k=1}^{N} \frac{F_{t,i,k}}{cap^{tr}} \times Y_{t,i,k}^{tr} \times \text{Time}_{t,i,k}^{tr} \leqslant AT_t^{tr} \ , \quad \forall tr = 1, 2, \ldots, TR; \ \forall t = 1, 2, \ldots, T \tag{3.9}$$

$$\text{Time}_{t,i,k}^{tr} = \sum_{j=1}^{N} \sum_{l=1}^{N} X_{t,i,j} \times X_{t,k,l} \times \text{Avg}T_{j,l}^{tr},$$
$$\forall i = 1, 2, \ldots, N \ , \ \forall k = 1, 2, \ldots, N \ , \forall tr = 1, 2, \ldots, TR; \ \forall t = 1, 2, \ldots, T \tag{3.10}$$

$$X_{t,i,j} \in \{0, 1\} \ , \quad i, j = 1, 2, \ldots, N \quad , \quad \forall t = 1, 2, \ldots, T \tag{3.11}$$

$$Y_{t,i,k}^{tr} \in \{0, 1\} \ , \quad i, k = 1, 2, \ldots, N \ , \qquad \forall tr = 1, 2, \ldots, TR, \quad \forall t = 2, \ldots, T. \tag{3.12}$$

The objective function (3.1) of this model is to minimize the total MHC. Objective function (3.2) minimizes the MRC and objective function (3.3) minimizes the fixed cost of the MHE. Constraints (3.4) ensure that each facility should be located in one position while constraints (3.5) ensure that in each position only one facility should be allocated. Set of constraints (3.6) state that for any pair of machines, if the materials are handled, a transporter must be assigned. Set of constraints (3.7) guarantee that one type of transporter is assigned to each pair of machines. Set of constraints (3.8) control the number of available vehicles in each period. Constraints (3.9) restrict the functions allocated a transporter to its available time. Constraints (3.10) control the time required to complete a tour between two machines. Constraints (3.11) and (3.12) restrict decision variables. This integer binary programming model is an extension to the classic DFLP and correspondingly is a complex combinatorial optimization problem. Next section presents an efficient algorithm to solve the proposed problem.

4. Solution methodology

On the earth's surface, gravitation force drives water flows to constantly move to lower altitudes. A flow will split into multiple sub-flows while moving from higher altitudes to lower ones on a rugged terrain. The mass and velocity are two main characteristics of a flow which determine the flow's momentum. When a number of water flows move to the same position, they will be merged into a single water flow with higher momentum. Governed by gravity and driven by fluid momentum, flows can run to higher levels or run over bumps to navigate various terrains. Water flowing will cease and stagnate at lowest local or global depression, when the momentum left cannot drive out the water out of the hollow.

Inspiring this behavior of water flow, Yang and Wang [21] proposed a new meta-heuristic optimization algorithm, called water flow-like algorithm (WFA), for solving NP-hard combinational optimization problems. Tran and Ng [22] proposed a water-flow algorithm to solve scheduling problem. They applied the algorithm to a maltose syrup production problem, and illustrated it's ability for solving problems in practical applications. Wu *et al.* [23] and Chang and Wu [24] used the WFA logic and developed a heuristic algorithm for solving the cell formation problem. Their computational results showed that the proposed algorithm has performed better than other benchmarking approaches in terms of both solution effectiveness and efficiency. Recently, Tran and Ng [25] used a hybrid WFA for solving multi-objective flexible flow shop scheduling problem with limited buffers. The performance of the proposed algorithm was investigated by randomly generated test problems. The computational results proved the effectiveness and efficiency of the WFA.

In WFA, the solution space of a problem is mapped as a geographical terrain and the objective function value is considered as the altitude of a water flow while each water flow represent a solution agent. Water splitting and moving to a lower position can be considered as a process of searching for the optimum status. A flow with larger momentum will generate more sub-flows and its surrounding space will be better explored. The mass and velocity of each flow are distributed to its sub-flows so that high quality sub-flows are of higher proportion of the mass and velocity of the addressed flow. When more than two flows move to the same location, they will merge into a single flow. Mass and velocity are then amassed to create an aggregated flow. The new aggregated flow is able to reinforce the searching process around its location. Furthermore, to escape local optima, some water flows may evaporate and return to the terrain by precipitation.

4.1. Multi-objective water flow-like algorithm

In this paper, we have developed a multi-objective version of WFA for discrete optimization problems. Our multi-objective water flow like algorithm (MOWFA) is based on the Pareto approach. The MOWFA consists of five main operations: (3.1) flow splitting and moving; (3.2) flow ranking; (3.3) flow merging based on similarity coefficient method; (3.4) flow evaporation; and (3.5) heuristic precipitation. Physics quantities such as mass, velocity, fluid momentum, energy, and gravitational acceleration are also used as basic parameters to construct this algorithm. In the proposed MOWFA, the flows are ranked based on the two key concepts of non-dominated sorting and crowding distance.

In MOWFA, solution agents are considered as water flows, solution space is mapped to the geographical terrain and the objective function value is considered by the altitude of a water flow. Searching the solution space is modeled as the terrain traversed by the flows. Initially, a cloud, representing an iteration, randomly produces a set of flows into different positions of the ground. The flows are ranked based on non-dominated sorting and crowding distance metrics. Then, based on the rank, each flow is given a momentum. The flows with higher momentums are allowed to generate more sub-flows. Then any tow flows at the same location (sub-flows), are merged into a single flow with higher momentum. In each iteration, the momentum of all flows is decreasing. The flows with zeros momentum will be evaporated and then are returned to the ground by precipitation. The way of encoding the flows, flow splitting and moving, flow merging operation, flow evaporation operation and

```
Initialize the parameters of WFA algorithms ( )
Generate random initial flows (POP)
Perform non dominated sorting
Compute crowding distance
Ranking flows based on non-dominated sorting and crowding distance
While stop criterion is false
{
    For each flow i ∈ {1, 2, ..., N} Do
     Flow splitting and moving
     Searching to find the optimum solution based on neighborhood structures of WFA.
     End For
     Flow merging.
     Water evaporation.
     If rainfall required Do
     Precipitation.
     End If
    Perform non dominated sorting
    Compute crowding distance
    Rank flows based on non-dominated sorting and crowding distance
    Select the flows for next iteration by eliminating the extra flows.
}
```

FIGURE 1. Pseudo-code of MOWFA.

precipitation used in the MOWFA are described later. To give a better understanding of MOWFA, Figure 1 shows its Pseudo-code.

4.2. A fast non-dominated sorting approach

First, the flows are sorted based on non-domination relationship. The non-domination is an individual factor and is said to dominate the other one if its objective function is not worse than the other and at least, one of its objective functions is better than the other one. Figure 2 shows a graphical representation of the fast non-dominated sorting when minimizing the both objectives is desired. Individuals such as x_1, x_4, x_6 and x_7 are assigned ranks as rank 1 since there is no individual superiority to them with respect to $f_1(x)$ and $f_2(x)$. After elimination of the individuals classified as rank 1, individuals with rank 2 are selected, and this process is repeated until all individuals are classified. The non-dominated sorting approach will require at most $O\left(MN^2\right)$ computations, where M is the number of objectives and N is the number of solutions in the population [26].

A set of front-1 individuals is called Pareto-optimal front.

If two solutions (flows) fall into a same front, a new metric called "crowding distance" is used to evaluate the individuals. The crowding distance is a measure of how close an individual is to its neighbors. The basic idea behind the crowding distance is finding the Euclidean distance between each individual in a front based on their m objectives in m-dimensional hyperspace.

Then, the standard operators of the WFA are performed on the flows to enhance the current flows and generate new flows as the next generation. Furthermore, we include a heuristic precipitation process in the algorithm to raise the solution exploitation capability of the search process. The details of the other operations of the MOWFA for the MODFLP are described in the following subsections.

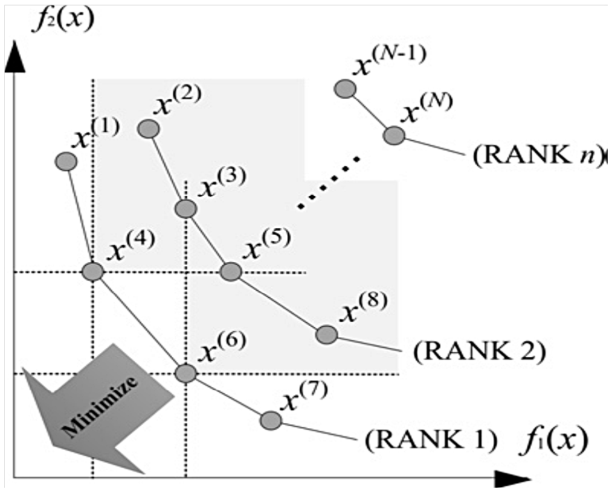

FIGURE 2. Schematic representation of the non-dominated sorting for two objectives [27].

	Period 1					Period 2				
Machine layout	2	4	1	5	3	2	4	5	3	1
Transporters allocation	0	3	2	1	1	0	1	1	2	2
		0	3	2	1		0	1	3	2
			0	1	1			0	1	1
				0	1				0	3
					0					0

FIGURE 3. An example of the proposed solution representation (a water flow).

4.3. Encoding the flows

Initially, a set of flows as initial random solutions are randomly distributed on some positions on the ground. As mentioned earlier, flows are mapped as solutions and must be a representative of decision variables. We present each flow by two sub-matrixes each of which corresponds to a special area of decision making. The first sub-matrix, which is related to the way machines are placed in positions, is presented by a $1 \times NT$ vector in which N is the number of machines and T is the number of periods. The cells 1-T of sub-matrix 1 correspond to machine layout in period one, cells T+*1*-*2*T correspond to machine layout in period two and so on. The value inside each cell denotes the machine number and the cell's rank represents the location of the machine. For example, Figure 3 shows a solution with two periods of time in which during period one, machine 2 is placed in location 1, machine 4 is placed in position 2, machine 1 is placed in position 3, machine 5 is placed in position 4 and machine 3 is placed in position 5. The second sub-matrix is related to assigning the transporters to machines in which the type of selected transporter is pointed to. For instance, in typical solution of Figure 3, the transporter type 3 is responsible for handling the materials between machines 1 and 2 in period 1.

4.4. Flow splitting and moving

Driven by flow momentum, the flows start to move to new locations and explore the solution space for better solutions. A flow with higher momentum generates more streams of sub-flows than one with less momentum.

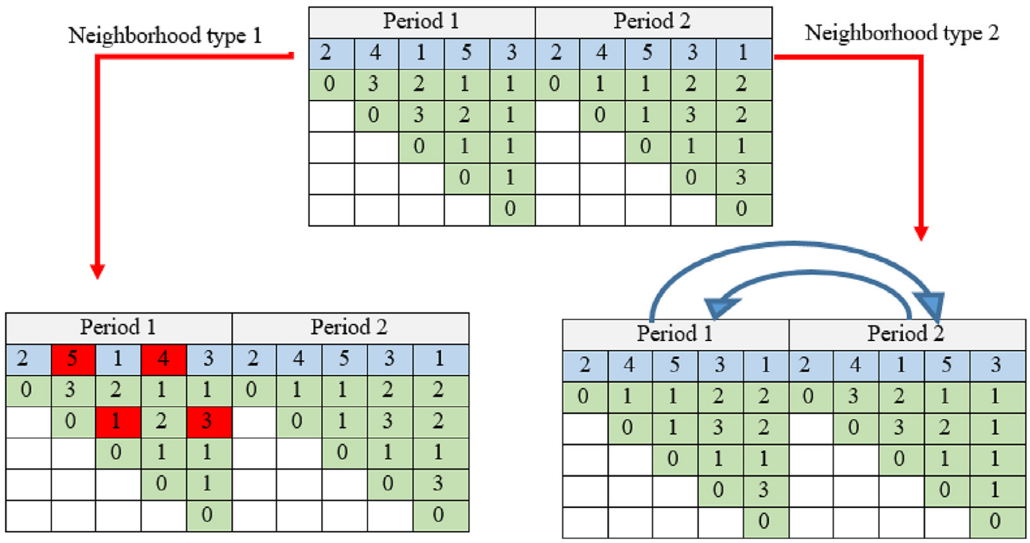

FIGURE 4. An example of neighborhood strategies.

The number of sub-flows split from a flow can be computed by equation (4.1),

$$n_i = \min\left\{\overline{n}, \mathrm{int}\frac{M_i V_i}{T}\right\} \tag{4.1}$$

where $\bar{n}$ is an upper bound imposed on the number of sub-flows, M_i represent the mass and V_i represent the velocity of the flow "i". When the momentum of a flow is lower than a predefined base momentum T, no splitting happens and the flow moves as a single stream to the neighboring location. Furthermore, if a flow has a zero velocity, it will stagnate at its location. The locations of the split sub-flows are derived from the neighboring locations of the original flow. In reality, a flow movement is a search procedure from the current location to a neighboring one. The design of the flow-moving operation is problem-dependent. For more efficiency of proposed MOWFA, we have designed two neighborhood strategies which are capable of keeping the feasibility of solutions.

In neighborhood strategy type 1, after randomly selecting a period, two different cells of sub-matrix 1 and two different cells of sub-matrix 2 are selected and swapped (as is depicted in Fig. 4). In neighborhood strategy type 2, two periods are randomly selected and all the data related to the machine layout and transporter allocation of these periods are exchanged.

As previously mentioned, the mass of flow i is distributed to its sub-flows. For this purpose, the sub-flows of flow i are ranked based on non-dominated sorting and crowding distance metric. The mass of sub-flow j distributed by flow i, U_{ij}, can be obtained from equation (4.2).

$$U_{ij} = \left(\frac{n_i + 1 - \mathrm{rank}_j}{\sum_{r=1}^{n_i} r}\right) \times M_i \tag{4.2}$$

where $rank_j$ represents the rank of sub-flow j with respect to the other sub-flows. For instance, if flow i splits to 5 sub-flows, the mass of rank 1 sub-flow is obtained from equation (4.3).

$$U_{ij} = \left(\frac{5 + 1 - 1}{15}\right) \times M_i = \frac{5}{15} \times M_i. \tag{4.3}$$

Also, velocity of sub-flow j is computed by equation (4.4).

$$\mu_{i,j} = \begin{cases} \sqrt{V_i^2 + 2g\overline{\delta_{i,j}}} & \text{if } V_i^2 + 2g\delta_{i,j} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

$$\overline{\delta_{i,j}} = \text{mean} \lim_k \delta_{i,j}^k$$

where "g" is the gravitational acceleration and $\delta_{i,j}^k \delta_{\mathrm{i,j}}^{\mathrm{k}}$ represents the altitude drop from flow i to its sub-flow j in objective function k which means the improvement of kth objective function when moving from current solution i to neighborhood solution j. The value of $\delta_{i,j}^k \delta_{\mathrm{i,j}}^{\mathrm{k}}$ for each of minimization and maximization objective functions is computed separately (using Eq. (4.5)) and then is averaged,

$$\delta_{i,j}^k = \begin{cases} f_i^k - f_{i,j}^k, & \text{for minimization} \\ f_{i,j}^k - f_i^k, & \text{for maximization} \end{cases} \tag{4.5}$$

where $\mathrm{f}_{\mathrm{i}}^{\mathrm{k}} f_i^k$ represents the value of objective function $k$of flow$i$and $f_{i,j}^k \mathrm{f}_{\mathrm{i,j}}^{\mathrm{k}}$ represents the value of kth objective function of sub-flow j distributed by flow i.

4.5. Flow-merging operation

When more than two flows arrive at the same location, they will merge into a single flow with bigger momentum. Whether the location of two flows are the same or not is investigated by our proposed criterion called similarity coefficient (SC). If the SC between two flows is greater than a predefined threshold, their locations are supposed to be identical and the merging takes place. Figure 5 is proposed to find the SC between two flows a and b:

$$SL_{ab} = \frac{\sum_{i=1}^{N}\sum_{t=1}^{T}\partial\left(X_{tia}, X_{tib}\right)}{N \times T}$$

If $SL_{ab} = 1$

$$SC_{ab} = \frac{\sum_{i=1}^{N}\sum_{k=1}^{N}\sum_{t=1}^{T}\partial\left(Y_{tika}, Y_{tikb}\right)}{N^2 \times T}$$

Else

$SC_{ab} = 0$

End If

/ SL_{ab} is the rate of similarity in terms of layout of machines /

/ X_{tia} and X_{tib} are the location of machine i in period t in the flows a and b respectively /

/ N is the number of machines /

/ T is the number of periods /

/ SC_{ab} is the rate of similarity coefficient /

/ Y_{tika}, Y_{tikb} are selected transporter between machine i and k in the period t in the flows a and b respectively /

FIGURE 5. Determining the similarity coefficient between two flows a and b.

In Figure 5, $\partial(A,B)$ is the similarity between two especial bits given by equation (4.6).

$$\partial(A,B) = \begin{cases} 1 & \text{if } A = B \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

According to Figure 5, the SC between two flows is zero if there is only a small difference in machine layout design. This is because the layout of machines has a great impact on the objective functions. If SC_{ab} exceeds a predefined threshold, the flows a and b are recognized as identical flows and merged. Suppose that the flows a and b are sharing the same location, then both flows a and b will be removed and the new flow will be generated based on the characteristics presented in equations (4.7)–(4.8).

$$M_{new} = M_a + M_b \tag{4.7}$$

$$V_{new} = \frac{V_a M_a + V_b M_b}{M_a + M_b}. \tag{4.8}$$

The new generated flow is quite similar to flow a and b in terms of sub-matrix 1. But, since sub-matrix 2 (transporter allocation) of flows a and b may be different, the structure of the merged flow in sub-matrix 2 (transporter allocation) is followed by the flow with higher rank. Therefore, by reducing the number of agents which are representative of similar positions, MOWFA avoids doing redundant searches.

4.6. Flow evaporation operation

It is natural that flows of water evaporate after possible movement and return to the ground through precipitation. In MOWFA, water evaporation and precipitation mechanism help the algorithm to escape from local optima. If a flow falls into a local optimum, it will be stagnated and so loss the capability of moving, merging or splitting. To overcome this problem and release the flow from the local optima, the trapped flow is forced to evaporate into the atmosphere.

The proposed MOWFA establishes a velocity-based evaporation according to the flows with smaller velocities so that they will evaporate more speedily than flows with larger velocities. The formulation of the velocity-based evaporation proposed by [28], is presented in equations (4.9)–(4.10).

$$M_i = (1 - \rho_i)\, M_i \tag{4.9}$$

$$\rho_i = \begin{cases} 1, & \text{if} \quad \mu_{i,j} = 0 \\ 0, & \text{if} \quad \dfrac{\mu_{i,j}}{V_i} \geqslant 1 \\ 1 - \dfrac{\mu_{i,j}}{V_i} & \text{if} \quad 0 \leqslant \dfrac{\mu_{i,j}}{V_i} \leqslant 1. \end{cases} \tag{4.10}$$

4.7. Heuristic precipitation

After a number of iterations, the evaporated water will return to the ground by precipitation operator. In this paper, a seasonal rainfall is applied periodically in which the precipitation takes place when the number of evaporated flows reaches to a predefined level. In order to enhance the algorithm's diversity, the location of returned flow is deviated far away from the location of flow before evaporation. In this way, the bits of flow are arranged inversely (Fig. 6).

Finally, all the initial flows, all the generated sub-flows and the flows generated *via* precipitation are gathered and form a new population. Then in this population, the members are ranked by non-dominated sorting and crowding distance criteria. By eliminating the overflow members, a population is constructed as initial population of next generation. The process continues to be performed until the stopping criteria are satisfied.

Evaporated water

Period 1					Period 2				
2	4	1	5	3	2	4	5	3	1
0	3	2	1	1	0	1	1	2	2
	0	3	2	1		0	1	3	2
		0	1	1			0	1	1
			0	1				0	3
				0					0

Returned water

Period 1					Period 2				
3	5	1	4	2	1	3	5	4	2
0	1	1	2	3	0	2	2	1	1
	0	1	2	3		0	2	3	1
		0	1	1			0	1	1
			0	1				0	3
				0					0

FIGURE 6. An example of generated solution based on heuristic precipitation.

4.8. Stopping criterion of the algorithms

For stopping MOWFA, first we define the set of m iterations as a round. Then, in every iteration, the best value of every objective function is determined. If, the mean change of all the objective functions between two successive rounds remains constant within 0.95% confidence interval, the algorithm is terminated.

5. Computational experiments

In order to verify and evaluate the results obtained by MOWFA some computational experiments are conducted and the performance of MOWFA is compared with one of the NSGA-II and NRGA algorithms. To do this, a set of test problems are randomly generated. There are 4 different problem sizes. The first contains problems with 6 machines in 5 periods, second contains problems with 6 machines in 10 periods, third contains problems with 15 machines in 5 periods and the fourth contains problems with 15 machines in 10 periods. For each problem size, we generate eight instances. Note that throughout this paper all calculations were performed a PC with Intel Core2 Duo 2.26 GHz CPU and 2 GB RAM. Moreover, algorithms were coded in MATLAB software (Version 7.10.0.499, R2010a).

5.1. Multi-objective performance metrics

In multi-objective optimization algorithms, convergence to the Pareto optimal front and the maintenance of a diverse set of solutions are two objectives. There is no single metric to evaluate the performance of the algorithms to satisfy these objectives. In this paper, five metrics of spacing, mean ideal distance, diversity, number of found solutions in Pareto front and number of function evaluations have been used to measure the algorithms' convergence and diversity.

5.1.1. Spacing metric

Zitzler [29] suggested spacing metric that calculates the relative distance between consecutive solutions in the non-dominated set obtained as equation (5.1).

$$S = \sqrt{\frac{1}{nos-1}\sum_{i=1}^{nos}\left(d_i - \overline{d}\right)^2} \tag{5.1}$$

where d_i is the sum of the differences in objective function values between solution i and it's two nearest neighbors for each objective and $\bar{d}$ is the average of all d_i's. When the solutions are nearly uniformly spaced, the spacing measure will be small. Thus, an algorithm with a smaller spacing is preferred.

5.1.2. Mean ideal distance (MID)

This metric proposed by Zitzler and Thiele [30] measures the convergence rate of Pareto fronts to an ideal point (0, 0),

$$MID = \frac{1}{NOS} \sum_{i=1}^{NOS} c_i \tag{5.2}$$

where $c_i = \sqrt{\sum_{j=1}^{J} f_{ji}^2}$ and f_{ji} is the jth objective function of ith solution. The low value of MID means that the solutions in Pareto front are high-quality solutions.

5.1.3. Diversity metric

This metric, proposed by Zitzler Thiele [30], evaluate scatter of solutions in the Pareto front. As a rule, whatever the scale is bigger, better.

$$D = \sqrt{\sum_{j=1}^{J} \left(\max_i f_{ji} - \min_i f_{ji} \right)^2}. \tag{5.3}$$

5.1.4. Number of found solutions in Pareto front (NOS)

Count the number of the solutions in Pareto optimal front. Thus, more solutions in Pareto frontier imply better performance of the algorithm [31].

5.1.5. Number of function evaluations (NOF)

In order to give a general perspective to the readers about the algorithms' speed, the number of function evaluations is considered as a performance metric. The speed of running the algorithms to find near optimum solutions is one of the most important indexes in order to evaluate the algorithms.

5.2. Tuning algorithm's parameters

The value of the parameters significantly affects the quality of the algorithms. Algorithms cannot reach to the appropriate final solutions if their parameters are not adjusted properly. In this section, we investigate the behavior of proposed MOWFA in different levels of parameters and determine the best level of the parameters. The full factorial design of experiment is a conventional statistical method used for tuning the parameters, but this method is not always well-organized because its calculations are increasingly complex when the number of parameters is high. Therefore, in this paper, to limit the number of the experiments, a practical experimental design technique, known as Taguchi method, is used. Taguchi method is a fractional factorial experiment that is proposed by Taguchi as an efficient alternative for full factorial experiments.

In order to apply the Taguchi method, the levels of the factors should first be determined. The initial levels of factors are shown in Table 1. The presented factors include the actual names along with their brief names.

In the Taguchi method, the values of quality characteristics obtained through the experiments are transferred into a measure called signal to noise (S/N) ratio. The frame of this ratio is different for each response type. equation (5.4) formulates S/N for a larger-the-better response type, where y_i is the ith observed value of the response (quality characteristic) and n is the number of observations in a trial.

$$\frac{S}{N} = -10 \log \left(\frac{1}{n} \sum_{i=1}^{n} y_i^2 \right) \cdot \tag{5.4}$$

Multi-objective coefficient of variation (MOCV), proposed by Rahmati *et al.* (2013), is considered as the response for the experiments.

$$MOCV = \frac{MID}{Diversity} \cdot \tag{5.5}$$

TABLE 1. Algorithm parameter ranges along with their levels.

Solving methodology	Parameter	Description	Level 1	Level 2	Level 3	Level 4
NSGA-II	*Popsize*	Initial pop size	30	70	**140**	200
	P_c	Percent of cross over	0.7	**0.8**	0.85	0.9
	P_m	Percent of mutation	0.1	**0.15**	0.17	0.2
	Round	Number of generations	200	300	**400**	500
NRGA	*Popsize*	Initial pop size	50	100	**150**	200
	P_c	Percent of cross over	0.7	**0.8**	0.85	0.9
	P_m	Percent of mutation	0.1	0.15	0.2	**0.3**
	Round	Number of generations	200	**350**	500	600
MOWFA	*Popsize*	Number of initial flows	30	50	**70**	90
	M_0	Initial mass of original low	20	**30**	40	50
	V_0	Initial velocity of original flow	10	20	**30**	30
	Round	Number of Iterations	100	200	**300**	400

TABLE 2. The orthogonal array and computational results to tune *NSGA-II and NRGA.*

Exp No.			*NSGA-II*			*NRGA*						
	Popsize	P_c	P_m	*Iteration*	$MOCV_1$	$MOCV_2$	$MOCV_3$	S/N	$MOCV_1$	$MOCV_2$	$MOCV_3$	S/N
1	1	1	1	1	1.02	1.29	1.69	−2.68	1.27	1.55	2.06	−4.41
2	1	2	2	2	0.66	0.49	0.42	5.47	0.99	0.85	0.74	1.25
3	1	3	3	3	0.72	0.84	0.77	2.15	0.90	1.03	0.96	0.33
4	1	4	4	4	0.87	0.98	0.82	0.99	1.05	1.20	1.00	−0.71
5	2	1	2	3	0.55	0.58	0.60	4.75	0.67	0.71	0.77	2.86
6	2	2	1	4	1.11	0.69	0.60	1.62	1.33	0.86	0.74	−0.10
7	2	3	4	1	0.89	0.82	0.71	1.83	1.08	1.00	0.90	0.03
8	2	4	3	2	1.11	0.85	0.95	0.21	0.79	0.89	0.74	1.84
9	3	1	3	4	0.39	0.41	0.43	7.79	0.49	0.50	0.55	5.78
10	3	2	4	3	0.52	0.74	0.60	4.07	0.63	0.89	0.73	2.42
11	3	3	1	2	0.68	0.59	0.71	3.58	0.83	0.74	0.89	1.66
12	3	4	2	1	0.68	0.52	0.84	3.17	0.84	0.64	1.01	1.43
13	4	1	4	2	0.68	0.62	0.59	4.00	0.86	0.75	0.75	2.08
14	4	2	3	1	0.35	0.47	0.52	6.89	0.47	0.52	0.65	5.17
15	4	3	2	4	0.57	0.59	0.66	4.32	0.71	0.71	0.80	2.60
16	4	4	1	3	0.61	0.62	0.72	3.70	0.72	0.77	0.92	1.87

As mentioned earlier, in Pareto-based algorithms, two main goals including acceptable convergence and diversity are considered. Since, MID measures the convergence rate of the algorithm and diversity measures, the diversification in Pareto front. MOCV is a comprehensive combination of major metrics which is used as a single response in the Taguchi method. Tables 2 and 3 summarize the experimental results of NSGA-II, NRGA and MOWFA. Regarding equations (5.4) and (5.5) these tables present S/N and MOCV as well.

Figures 7 and 9 shows how the index values of S/N are changing at different levels of the algorithms. Levels where the index S/N has reached the maximum are selected as the optimal levels. Optimal parameter levels of the algorithms are highlighted in Table 1.

5.3. Algorithm's evaluation

In this section, we present results of numerical experiments for the proposed algorithm, compared to those for NSGA-II and NRGA using random generated test problems. Since the meta-heuristic algorithms are naturally stochastic, each instance of the problem is replicated 45 times and the averaged results are reported. The results

TABLE 3. The orthogonal array and computational results to tune *MOWFA*.

Exp No.	*MOWFA*							
	Popsize	M_0	V_0	*Round*	$MOCV_1$	$MOCV_2$	$MOCV_3$	S/N
1	1	1	1	1	1.35	0.99	1.15	−1.38
2	1	2	2	2	1.13	1.08	0.90	−0.36
3	1	3	3	3	0.44	0.29	0.16	10.03
4	1	4	4	4	1.24	1.07	0.91	−0.69
5	2	1	2	3	0.55	0.59	0.74	3.99
6	2	2	1	4	0.44	0.57	0.41	6.40
7	2	3	4	1	1.11	1.33	0.98	−1.21
8	2	4	3	2	0.30	0.19	0.13	13.23
9	3	1	3	4	0.22	0.21	0.04	15.04
10	3	2	4	3	0.34	0.25	0.17	11.69
11	3	3	1	2	0.89	0.98	1.01	0.35
12	3	4	2	1	1.02	1.09	1.12	−0.64
13	4	1	4	2	0.50	0.52	0.58	5.46
14	4	2	3	1	0.09	0.16	0.17	16.80
15	4	3	2	4	0.50	0.49	0.65	5.15
16	4	4	1	3	0.95	1.04	1.06	−0.15

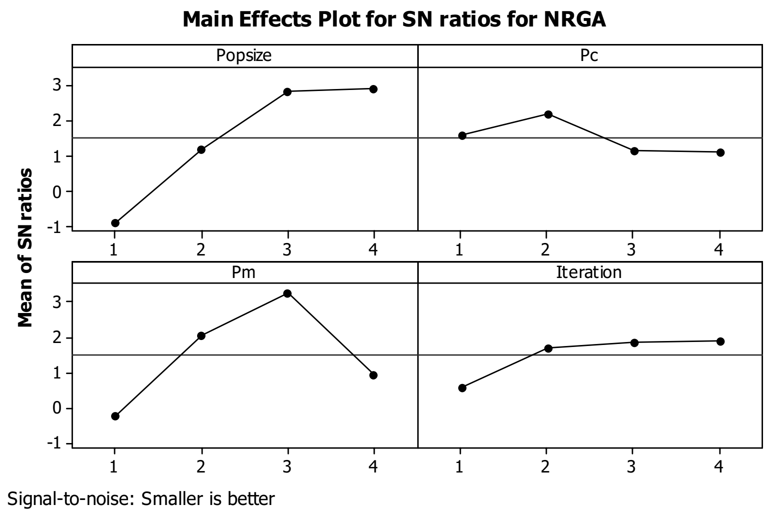

FIGURE 7. The mean S/N ratio plot for each level of the factors for NRGA.

are shown in Appendix A where obtained values of five performance measures are given. Figure 10 shows these results graphically. Figure 10a shows the algorithms' performances using the spacing metric. Since the standard spacing is less, it can be concluded that the NRGA algorithm has the worst performance among the algorithms. MOWFA and NSGA-II are approximately performing similarly.

Figure 10b compares MID metric of the algorithms. It is clear that in almost all instances the MOWFA algorithm performs better in finding high quality solutions. As shown in Figure 10c, in terms of Diversity metric, MOWFA gives better values compared with the two other algorithms for the most of the test problems. Also, it is clear that the performance of NSGA-II and NRGA is at the same level. As shown in Figure 10d, based on NOS metric, the performances of all the algorithm are approximately equivalent; however, as the problem size increases, the quality of NRGA is decreasing.

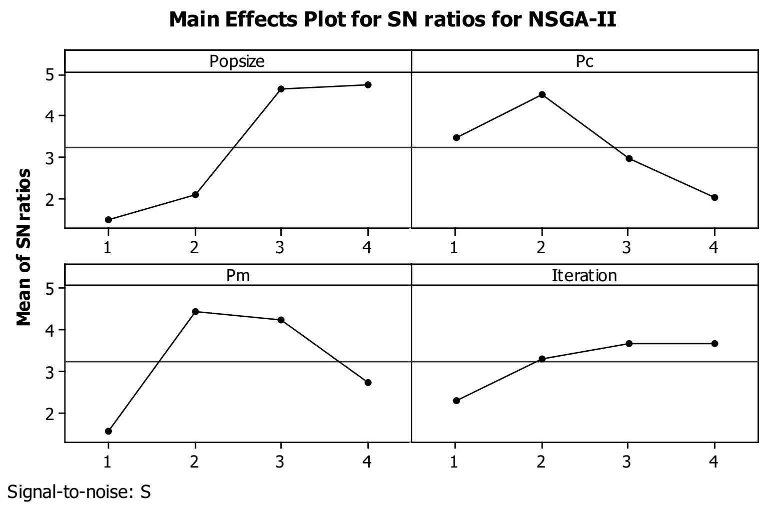

FIGURE 8. The mean S/N ratio plot for each level of the factors for NSGA-II.

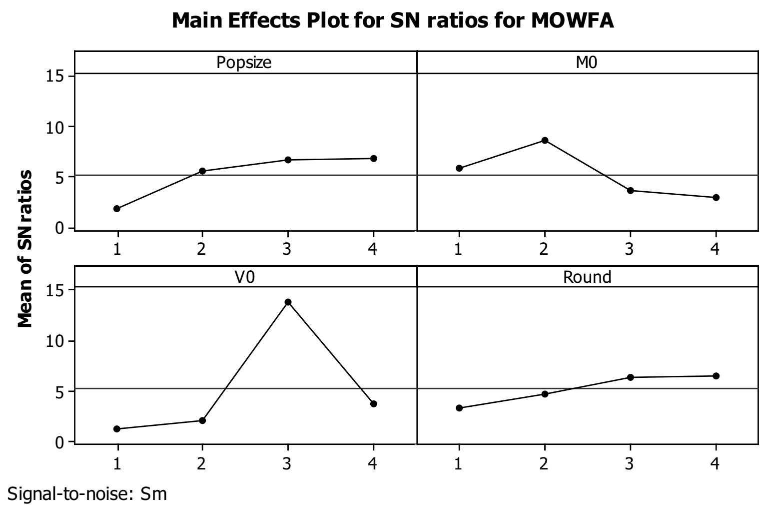

FIGURE 9. The mean S/N ratio plot for each level of the factors for MOWFA.

Figure 10e illustrates the superiority of the proposed MOWFA algorithm in comparison with NSGA-II and NRGA in terms of NOF. According to Figure 10e, for small-sized problems, the MOWFA performs similar to other algorithms, while for the large-sized problems, MOWFA considerably outperforms than the two other algorithms, especially than NSGA-II.

Figure 11 is a graphical representation of the obtained Pareto front for problems 1, 9, 17 and 25. As shown in Figure 11, the solutions from MOWFA are of the best values of all objective functions which in turn is a good evidence for the effectiveness MOWFA in terms of MID.

In order to statistically verify the results shown in Figure 10, we used the analysis of variance (ANOVA) test and statistically compared algorithms according to each of metrics separately. In this case, the value of each metric is transformed to a normalized performance measure named relative percentage deviation (RPD),

TABLE 4. ANOVA test.

Metric's name	F-value	P-value	Test results
Spacing	22.82	8.63e–09	Null hypothesis is rejected
MID	86.28	6.47e–22	Null hypothesis is rejected
Diversity	13.45	7.37e–05	Null hypothesis is rejected
NOS	29.97	8.98e–11	Null hypothesis is rejected
NOF	38.58	6.30e–13	Null hypothesis is rejected

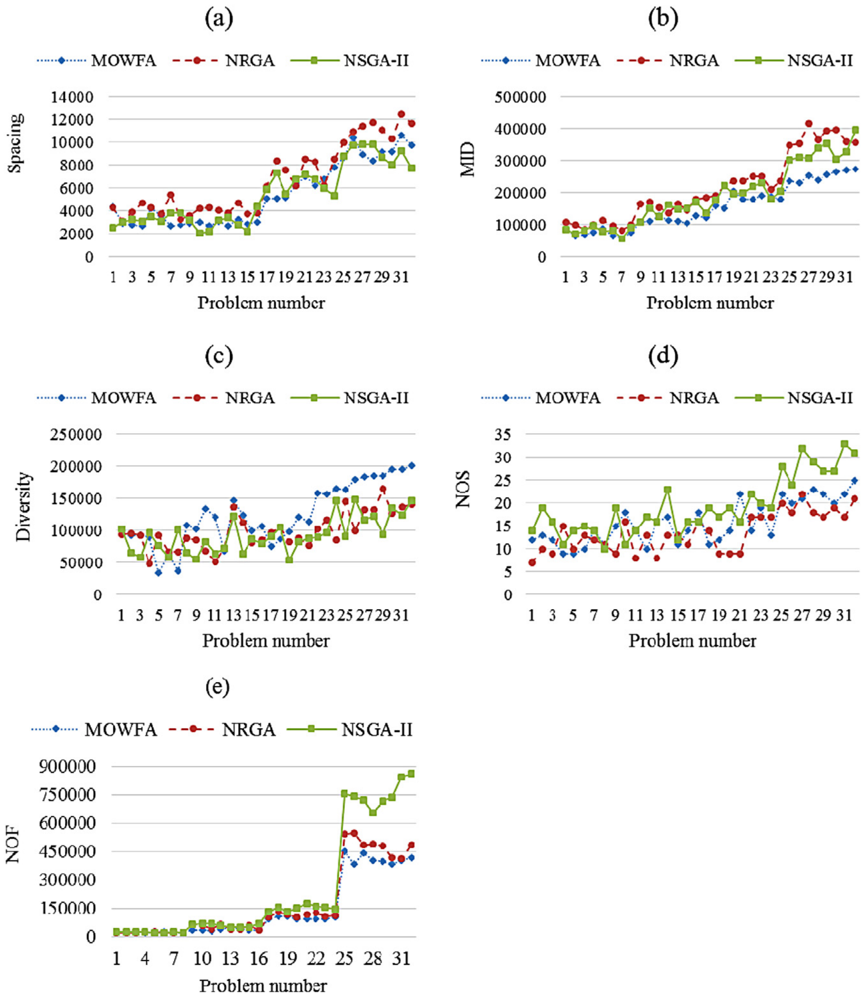

FIGURE 10. Comparison of the proposed algorithm according to spacing metric MID metric, diversity metric, metric NOS and NOF metric.

obtained by the following formula:

$$RPD\,(i,j,k) = \frac{\mathrm{Alg}_{\mathrm{sol}}\,(i,j,k) - \min_{\mathrm{sol}}\,(j,k)}{\min_{\mathrm{sol}}\,(j,k)} \times 100. \tag{5.6}$$

In equation (28), $Alg_{\mathrm{sol}}\,(i,j,k)$ shows the value of performance measure j in problem number k that has been obtained by algorithm i and $\min_{\mathrm{sol}}\,(j,k)$ is the best value of performance measure j between all algorithms for

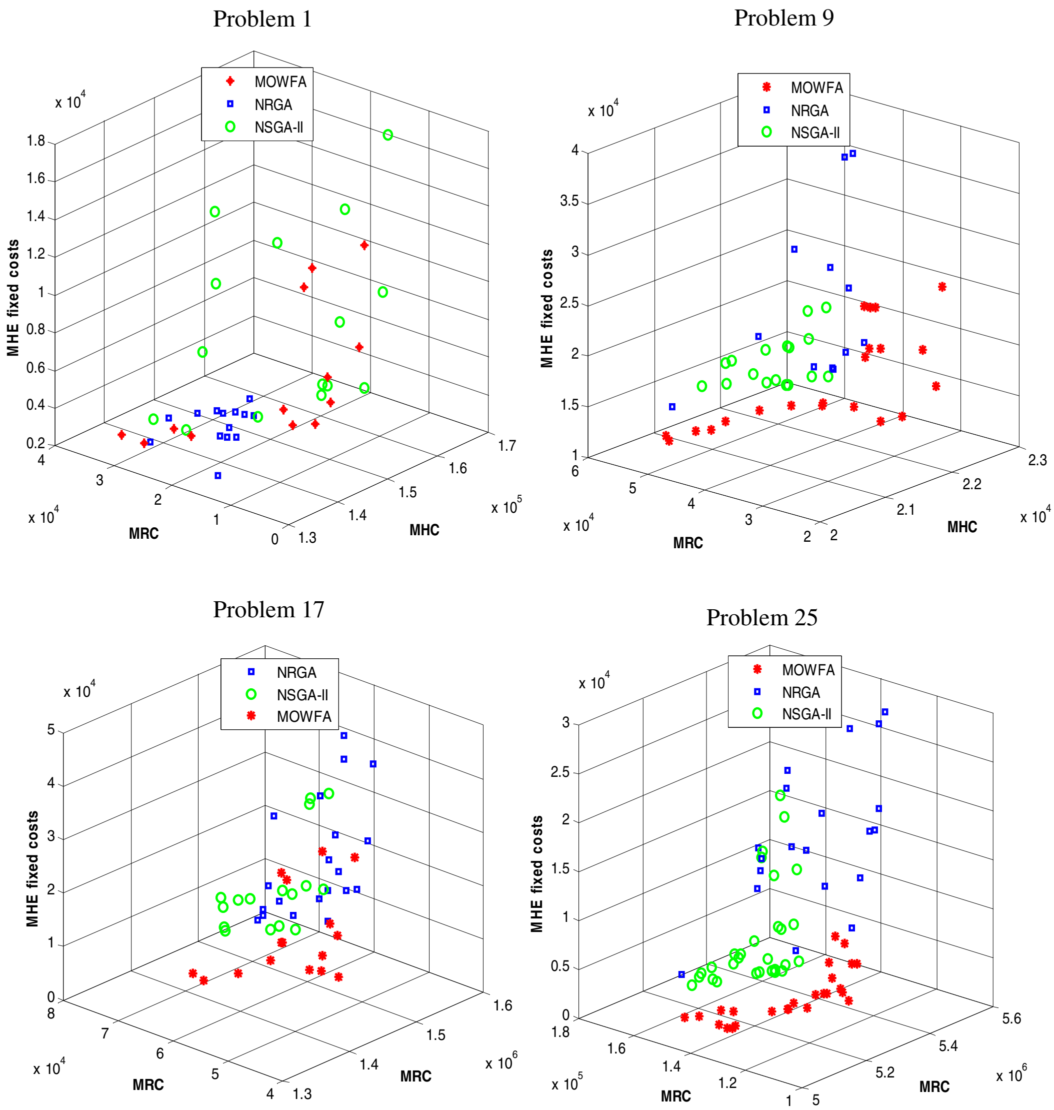

FIGURE 11. Visual presentation of obtained Pareto-front four test problem 1, 9, 17 and 25.

problem number k. The ANOVA results are shown in Table 4, in which rejecting the Null hypothesis indicates that there is significant difference between algorithms.

Based on the outputs of ANOVA test, it is clear that the algorithms have significant differences in terms of all performance metrics which necessitate the use of Tukey test to statistically rank algorithms. The results of the 95% Tukey simultaneous confidence intervals are shown in Figure 12. The results prove well performance of the algorithms based on the mean and Tukey intervals. As Figure 12, we can say that NRGA has the lowest spacing performance among all the algorithms. But, the efficiency of the MOWFA and NSGA-II is at the same quality. In terms of MID metric, the MOWFA algorithm is dramatically better than two other algorithms. Also NSGA-II can perform better than NRGA. Better performance of the MOWFA in terms of Diversity metric can be shown in Figure 12. Based on NOS index the MOWFA and NSGA-II are statistically at the same level. Based on Figure 12 we can prove that MOWFA is the most preferable technique in terms of the required NOF.

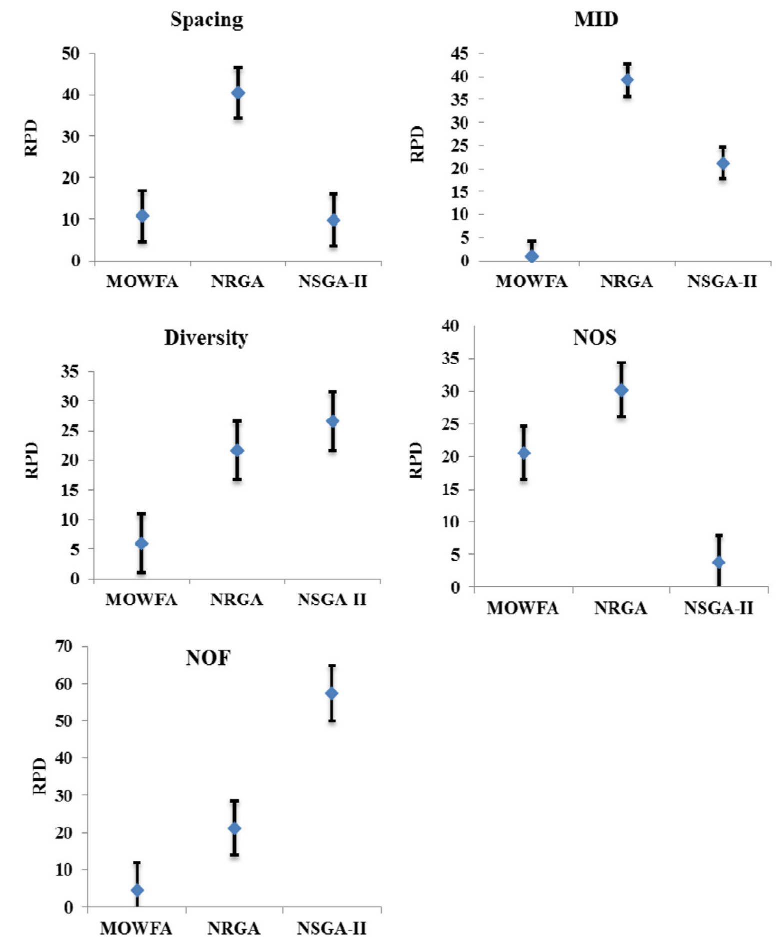

Figure 12. The 95% Tukey simultaneous confidence intervals for the metrics.

6. Conclusion

This paper explained how to develop a more realistic mathematical model for dynamic facility layout problem. Such models can be applied in manufacturing systems with a variety of MHE. In addition to minimizing MHC and MRC, reducing the fixed/fixing cost of the MHE by selecting a safer MHE was also considered as the other objective of the proposed model. To solve the problem, a new meta-heuristic algorithm inspired from the existing water flow like algorithm called MOWFA was proposed. Various test problems were designed to evaluate the performance of the algorithm in comparison with two well-known multi-objective evolutionary algorithms called NSGA-II and NRGA. The experimental results indicated that our proposed algorithm outperforms both NSGA-II and NRGA and it is also able to improve the quality of the solutions. The MID measure showed that the MOWFA could achieve better the objective-function values, especially in large-sized problems. Also, the MOWFA was so faster than the two other algorithms; this can show its applicability for the large-sized real-world problems. The presented model was still open for incorporating other features, such as transporter failure, machine breakdown and random processing time. Another clue for a future research is to consider the proposed algorithm on other types of combinatorial optimization problems. As another direction, it could be interesting to work on some other meta-heuristics and compare them with the given algorithms.

Appendix A.

Table A.1. Multi-objective performance measures obtained for each algorithm.

	Spacing			MID			Diversity		
	MOWFA	NRGA	NSGA-II	MOWFA	NRGA	NSGA-II	MOWFA	NRGA	NSGA-II
P1	4364	4311	2517	89 170	107 245	85 720	102 056	93 258	102 825
P2	2928	3125	2985	63 720	99 720	70 301	92 867	95 507	65 781
P3	2778	3874	3223	68 162	84 268	82 671	92 266	92 721	57 835
P4	2660	4712	3135	77 740	99 636	97 115	89 727	47 798	97 404
P5	4184	4282	3536	88 359	112 199	78 194	33 551	91 797	77 049
P6	3600	3768	3087	64 243	96 525	82 667	62 843	67 231	58281
P7	2659	5385	3829	56 215	82 645	57 337	37 288	66 708	101 320
P8	2780	3240	3817	76 389	99 850	91 132	108 879	88 495	65 230
P9	2969	3539	3150	108 094	164 423	108 605	102 685	85 674	55 295
P10	3016	4193	2081	111 321	171 307	149 924	133 687	68 388	81 878
P11	2679	4327	2245	128 346	154 520	125 884	120 610	50 642	64 082
P12	3092	4056	3164	114 660	137 378	162 938	68 065	73 147	72 590
P13	2675	3846	3386	111 383	165 054	147 971	146 458	136 729	122 052
P14	3241	4713	2785	105 834	146 264	151 001	123 827	112 637	63 483
P15	2832	3749	2221	128 319	178 727	171 941	99 148	80 979	86 176
P16	2979	3782	4342	123 021	183 561	136 524	107 005	85 610	79 217
P17	5074	6149	5935	15 901	190 025	176 880	75 429	96 786	90547
P18	5118	8359	7322	151 633	221 223	221 693	863 43	105 678	105 563
P19	5180	7611	5524	205 728	236 400	196 257	98 305	83 073	53 921
P20	6250	6197	6741	179 172	235 829	198 488	121 580	88 626	81 865
P21	7035	8531	7195	178 895	251 686	218 866	113 301	76 071	88 086
P22	6201	8311	6767	189 820	251 138	231 416	158 760	102 014	90 023
P23	6792	6251	5966	186 711	211 750	179 490	156 124	117 459	96 406
P24	7787	8538	5369	179 168	236 281	204 928	165 063	84 770	146 450
P25	8693	9971	8774	235 733	347 238	300 896	164 189	145 473	90 855
P26	10 368	10 928	9718	230 223	354 887	311 530	179 110	99 619	147 893
P27	8966	11 440	9838	253 287	416 590	307 895	184 045	132 195	116 752
P28	8391	11 718	9819	239 873	364 283	340 194	185 933	132 107	123 084
P29	9207	11063	8729	254 977	394 400	352 149	184 908	164 806	93 441
P30	9209	10 271	8087	263 560	395 843	304 959	195 043	126 198	134 747
P31	10 641	12 461	9246	269 918	359 465	328 036	195 429	136 831	124 167
P32	9729	11 624	7705	273 595	357 401	396 082	200 425	140 932	146 759

	NOS			**NOF* 10^2**		
	MOWFA	NRGA	NSGA-II	MOWFA	NRGA	NSGA-II
P1	12	7	14	23 075	20 833	22 916
P2	13	10	19	24 491	19 287	23 747
P3	12	9	16	18 743	20 912	26011
P4	9	15	11	18 335	23 206	24 098
P5	9	10	14	24 308	24 101	19 158
P6	10	13	15	25 613	18 224	21 047
P7	14	12	14	20 355	21 585	25 784
P8	11	11	10	21 191	19 215	21 807
P9	15	9	19	35 907	62 173	61 021
P10	18	16	11	32 382	59 457	68 730
P11	14	8	14	30 102	37 164	66 658
P12	10	13	17	39 461	65 003	59 471
P13	16	8	16	47 504	41 041	50 179
P14	17	13	23	39 940	36 675	46 817
P15	11	13	12	33 215	60 239	46 830
P16	14	11	16	36 061	33 964	68 700
P17	18	16	16	90 888	102 456	130 278
P18	11	14	19	107 114	128 744	155 434
P19	12	9	17	104 367	114 636	131 606
P20	14	9	19	89 960	100 856	148 224
P21	22	9	16	92 478	116 691	172 430
P22	14	17	22	91 947	127 904	158 870
P23	19	17	20	91 275	105 495	156 189
P24	13	17	19	103 861	110 974	145 701
P25	22	20	28	448 618	541 867	757 899
P26	20	18	24	384 950	545 010	739 484
P27	21	22	32	439 349	483 017	719 259
P28	23	18	29	402 728	489 441	655 397
P29	22	17	27	395 391	478 866	716 287
P30	20	19	27	383 603	416 241	735 609
P31	22	17	33	400 575	411 374	843 477
P32	25	21	31	414 777	482 183	862 108

References

[1] J.A. Tompkins, J.A. White, Y.A. Bozer, E.H. Frazelle, J.M. Tanchoco and J. Trevino, *Facilities planning*. Wiley, New York (1996).

[2] S. Chittratanawat, An integrated approach for facility layout, P/D locations and material handling system design. *Int. J. Prod. Res.* **37** (1999) 683–706.

[3] M.J. Rosenblatt, The dynamics of plant layout. *Manage. Sci.* **321** (1986) 76–86.

[4] T.L. Urban, A heuristic for the dynamic facility layout problem. *IIE Trans.* **25** (1993) 57–63.

[5] P. Kouvelis and A.S. Kiran, Single and Multiple Period Layout Models for Automated Manufacturing Systems. *Eur. J. Oper. Res.* **52** (1991) 300–314.

[6] B. Kaku and J.B. Mazzola, A tabu search heuristic for the plant layout problem. *INFORMS J. Comput.* **94** (1997) 374–384.

[7] D.G. Conway and M.A. Venkataramanan, Genetic search and the dynamic facility layout problem. *Comput. Oper. Res.* **21** (1994) 955–960.

[8] J. Balakrishnan and C.H. Cheng, Genetic search and the dynamic layout problem. *Comput. Oper. Res.* **27** (2000) 587–593.

[9] A. Baykasoglu and N.N.Z. Gindy, A simulated annealing algorithm for dynamic layout problem. *Comput. Oper. Res.* **28** (2001) 1403–1426.

[10] A.R. McKendall and J. Shang, Hybrid ant systems for the dynamic facility layout problem. *Comput. Oper. Res.* **33** (2006) 790–803.

[11] A.R. McKendall, J. Shang and S. Kuppusamy, Simulated annealing heuristics for the dynamic facility layout problem. *Comput. Oper. Res.* **33** (2006) 2431–2444.

[12] K.K. Krishnan, S.H. Cheraghi and C.N. Nayak, Dynamic From-Between Chart: a new tool for solving dynamic facility layout problems. *Int. J. Ind. Syst. Eng.* **1** (2006) 182–200.

[13] H. Rezazadeh, M. Ghazanfari, M. Saidi-Mehrabad and S.J. Sadjadi, An extended discrete particle swarm optimization algorithm for the dynamic facility layout problem. *J. Zhejiang. Univ. Sci. A.* **10** (2009) 520–529.

[14] J. Balakrishnan and C.H. Cheng, The dynamic plant layout problem: Incorporating rolling horizons and forecast uncertainty. *Omega* **37** (2009) 165–177.

[15] R. Şahin and O. Türkbey, A new hybrid tabu-simulated annealing heuristic for the dynamic facility layout problem. *Int. J. Prod. Res.* **47** (2009) 6855–6873.

[16] A.R. McKendall and W.H. Liu, New Tabu search heuristics for the dynamic facility layout problem. *Int. J. Prod. Res.* **50** (2012) 867–878.

[17] G.Y.H Chen, A new data structure of solution representation in hybrid ant colony optimization for large dynamic facility layout problems. *Int. J. Prod. Econ.* **142** (2013) 362–371

[18] G.Y. Chen and K.J. Rogers, Proposition of Two Multiple Criteria Models Applied to Dynamic Multi-objective Facility Layout Problem Based On Ant Colony Optimization. 2009. *IEEE International Conference on Industrial Engineering and Engineering Management*. Hong Kong (2009) 1553–1557.

[19] F. Jolai, R. Tavakkoli-Moghaddam and M. Taghipour, A multi-objective particle swarm optimization algorithm for unequal sized dynamic facility layout problem with pickup/drop-off locations. *Int. J. Prod. Res.* **50** (2012) 4279–4293.

[20] S. Emami and A.S. Nookabadi, Managing a new multi-objective model for the dynamic facility layout problem. *Int. J. Adv. Manuf. Technol.* **68** (2013) 2215–2228.

[21] F.C. Yang and Y.P. Wang, Water flow-like algorithm for object grouping problems. *J. Chin. Inst. Ind. Eng.* **24** (2007) 475–488.

[22] T.H. Tran and K.M. Ng, A water-flow algorithm for flexible flow shop scheduling with intermediate buffers. *J. Sched.* **14** (2011) 483–500.

[23] T.H. Wu, S.H. Chung and C.C. Chang, A water flow-like algorithm for manufacturing cell formation problems. *Eur. J. Oper. Res.* **205** (2010) 346–360.

[24] C.C. Chang and T.H Wu, A novel approach to determine cell formation, cell layout, and intracellular machine layout. In *Information Science and Service Science and Data Mining (ISSDM)*, 2012 6th International Conference on New Trends (2012) 717–721.

[25] T.H. Tran and K.M. Ng, A hybrid water flow algorithm for multi-objective flexible flow shop scheduling problems. *Eng. Optim.* **45** (2013), 483–502

[26] K. Deb, *Multi-objective optimization using evolutionary algorithms*. Wiley and Sons Limited, New York (2001).

[27] S. Honda, T. Igarashi and Y. Narita, Multi-objective optimization of curvilinear fiber shapes for laminated composite plates by using NSGA-II. *Composites Part B: Engineering* **45** (2013) 1071–1078.

[28] H. Wu, S.H. Chung and C.C. Chang, A water flow-like algorithm for manufacturing cell formation problems. *Eur. J. Oper. Res.* **205** (2010) 346–360

[29] E. Zitzler, *Evolutionary algorithms for multi objective optimization: Methods and Applications*. Ph.D. thesis, Swiss Federal Institute of Technology (ETH) Zurich, Switzerland, Germany (1999).

[30] E. Zitzler and L. Thiele, Multi-objective optimization using evolutionary algorithms a comparative case study. In *Fifth International Conference on Parallel Problem Solving*. Berlin, Germany (1998) 292–301.

[31] J.D. Schaffer, Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In Genetic algorithms and their applications: In *Proc. of the first international conference on genetic algorithms*. Hillsdale, NJ: Lawrence Erlbaum (1985) 93–100.