Scientia Iranica E (2017) 24(1), 364–377

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
www.scientiairanica.com

An ANN-based optimization model for facility layout problem using simulation technique

P. Azimi* **and P. Soofi**

Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran.

Received 31 May 2015; received in revised form 12 September 2015; accepted 4 April 2016

KEYWORDS
Facility layout;
Artificial neural network;
Discrete-event simulation;
Non-dominated sorting genetic algorithm.

Abstract. A real manufacturing system faces lots of real-world situations, such as stochastic behaviors; the lack of attention to this issue is noticeable in the previous research. The aim of this paper is to find the optimum layout and the most appropriate handling transporters for the problem by a novel solving algorithm. The new model contains two objective functions including the Material Handling Costs (MHC) and the complication time of jobs (makespan). Real-world situations such as stochastic processing times, random breakdowns, and cross traffics among transporters are considered in this paper. Several experiment designs have been produced using DOE technique in simulation software and an Artificial Neural Network (ANN) as a meta-model is used to estimate the objective functions in the metaheuristic algorithms. A hybrid non-dominated sorting genetic algorithm (H-NSGA-II) is applied for the optimization task. The proposed methodology is evaluated through a real case study. First, simulation model is validated by comparing it with a real data set. Then, the prediction performance of ANN is investigated. Finally, the ability of H-NSGA-II in searching the solution space is compared with the traditional NSGA-II. The results show that the proposed approach, combing simulation, ANN, and H-NSGA-II, provides promising solutions for practical applications.

© 2017 Sharif University of Technology. All rights reserved.

1. Introduction

A facility layout problem is concerned with determining the arrangement of machines, departments, or cells on the shop floor. The most important performance measure to evaluate the efficiency of a layout is the Material Handling Costs (MHC) [1]. Tompkins [2] claimed that 20 to 50 percent of the total operating expenses in manufacturing were attributed to MHC and effective facility layout could reduce these costs by 10 to 30 percent. The flow of materials and the distance between machines are important determinants of MHC. Also, MHC depends on the employed material handling vehicles. Material handling vehicles such as forklifts, trucks, and Automated Guided Vehicles (AGVs) are used to transport materials between various points. The type of these transporters influences the layout of the machines and vice versa. In traditional systems, the decisions related to material handling systems are made after finalizing the layout or vice versa. According to Meller and Gau [3], there exists a lack of parallel engineering in selecting the material handling system with respect to the facility layout. The lack of concurrent engineering results in a high degree of disharmony between the facility layout and material handling system. Therefore, the decisions on determining the type of material handling equipment and the place of machines should be made simultaneously.

While MHC remains the critical index of layout efficiency, shorter cycle times have become much important in today's manufacturing systems [4]. Today's

*. *Corresponding author. Tel: 021 22425773;*
Fax: 021 22425846
E-mail addresses: p.azimi@yahoo.com (P. Azimi);
parham_soofi@yahoo.com (P. Soofi)

consumer market demands that manufacturers must be competitive. This requires efficient operation of manufacturing plants and their ability to satisfy customer demand as quick as possible. On-time delivery and short manufacturing cycle times, as practical issues, should be considered during the layout design process.

This paper considers both MHC and completion time of jobs (makespan) as optimality criteria of the layout. To be more practical, this paper takes the stochastic nature of transporters handling time and transporters failure into account when calculating the makespan. Also, this paper considers effect of workflow interference as a major concern, which has previously been considered very poorly, in estimating the makespan. The cross traffic of transporters can result in considerable delays [5]. These delays lead to an increase in cycle times of production system. Consequently, by taking into account the above factors, we estimate the makespan with the highest possible precision. Because of the described complexity of the manufacturing system, a closed-form analytical expression to calculate makespan does not exist. Therefore, we utilize an Artificial Neural Network (ANN) to estimate the makespan. More specifically, we first build a series of random layout designs and then discrete-event simulation model is used to evaluate the makespan of these designs. The obtained makespan is applied to structure an ANN. The ANN works as makespan estimator during the search process. In order to search the solution space, a hybrid algorithm based on non-dominated sorting genetic algorithm (NSGA-II) and an adaptive local search are developed. The remainder of this paper is organized as follows.

Section 2 reviews the existing literature. Section 3 presents the mathematical formulation and properties of the problem. Section 4 explains the proposed methodology. To show performance of the suggested method, computational experiments are done in Section 5. Finally, the possible extensions of this research are listed along with the conclusions on the proposed approach.

2. Literature review

There is no research similar to the work accomplished in this paper as we review a series of approximately correlated studies. Rosenblatt [6] was the first who introduced the concept of multi-objective approach to facility layout problems. He considered both qualitative and quantitative objectives together to come up with a multi-objective formulation. Rosenblatt also proposed a graphical method to generate a series of efficient solutions based on the conflicting objectives of minimizing the flow cost and maximizing the closeness rating.

Simulation is known as a powerful tool to evaluate various alternatives of facility layout. Computer simulation has been applied in different facility layout problems (for example, see [4,7-11]). Gupta [12] employed a simulation model to select the best layout from a set of candidate layouts. He firstly generated a number of material flows based on a predefined probability distribution and found the optimum layouts of each flow. He defined the best solution as the layout with each department pair being separated by the average distance of the generated layouts. Then, the flexibility of the layout was measured by evaluating its deviation from the ideal distances.

According to Grajo [13], layout optimization and simulation are two tasks that are crucial to any facility planning and layout study. This is because simulation models can reflect all of the attributes of real systems that are difficult to consider using analytical models for the layout optimization problems. Azadivar and Tompkins [14] suggested a simulation model with a GA-based optimization algorithm. In their method, simulation models were used to evaluate the objective functions and GA-based algorithm was used to search the optimum solution. Azadivar and Wang [4] presented a facility layout optimization technique that considered the dynamic features and operational constraints of the system as a whole. In their proposed approach, the performance measures of system, such as cycle time and productivity, were evaluated by simulation. Pagell and Melnyk [9] investigated three layouts consisting of the existing worker-paced assembly line, a modified assembly line, and service cells to improve the overall operation of a service process. They used computer simulation to stimulate a critical analysis of the process.

Kulturel-Konak et al. [15] presented a case in which the demand rate was considered as a stochastic parameter. They also allowed routing flexibility for the products so that they were permitted to follow different routes in the facility. They used a simulation approach to model the uncertainty. In their simulation, the mean, variance, and covariance of interdepartmental flows were estimated; then, these estimations were used in the design process. Tabu search metaheuristic was employed to solve the problem. Jithavech and Krishnan [11] presented a simulation-based method to evaluate the uncertainty associated with the layout. They validated their simulation model against analytical methods. Results from case studies showed that the simulation-based procedure resulted in reduction of risk as high as 80%. Zhou et al. [10] introduced a method where simulation was combined with Genetic Algorithm (GA) to optimize the layout. They tested the optimized site layout within a simulation environment. Altuntas and Selim [16] proposed four different weighted association rule-based data mining approaches to solve facility layout problem. They constructed a simulation model and compared the layouts obtained by the proposed approaches in terms of five performance measures, namely machine utilization, total amount of products produced, cycle time, transfer time, and waiting time in queue. Dombrowskia and Ernst [17] presented a scenario-based simulation approach that used scenario technique, morphological analysis, and discrete event simulation to find out factory layout. Karpe et al. [18] presented a state of the art review of simulation methods for facility layout problems. Azadeh et al. [19] presented an integrated computer simulation-stochastic data envelopment analysis approach to deal with the facility layout problem. In their research, computer simulation network was used for performance modeling of each layout design. The outputs of simulation were average time-in-system, average queue length, and average machine utilization. By comparing their study with some of the relevant studies and methodologies in the literature, they revealed the high ability of the method to handle complex layout problems in manufacturing systems.

This is the first study that introduces an integrated computer simulation, ANN, and H-NSGA-II as an optimization approach for handling imprecision and non-linearity of layout problems in a special case of manufacturing system.

3. Problem description

The manufacturing system addressed here consists of m machines in which n types of parts, each requiring a set of operations, are to be processed. During the manufacturing process, the transporters move the materials from one machine to another until all the processes are completed. The transporters' handling time is stochastic with known probability distribution.

A desired design for the system requires an arrangement of m machines in m predefined positions and assignment of transporters to each pair of machines such that both MHC and makespan are minimized. Since minimization of MHC does not match minimization of makespan, the problem falls into the class of multi-objective optimization problems. To explain the conflict between two objective functions, adapted from Chiang et al. [20], an eight-machine example with the workflow matrix has been shown in Figure 1. The solution to this problem using MHC-based layout is shown in Figure 1(a). While this layout planning will minimize the MHC, it is clear that there are numerous points at which transporters interference occurs. An alternative layout, taking workflow interference into account, can be providing a workflow in which there are any conflicting workflows (Figure 1(b)).

Other assumptions considered are as follows:

- The distances between machines are determined a priori;
- Flow between machines is deterministic;
- Machines and locations are of equal size;
- The initial allocation cost of a machine in a location is ignored;
- The transporters' failure may occur, which leads to increase in their processing time.

3.1. Model formulation

In this section, the nonlinear integer programming formulations of the problem are presented. Before proceeding to the mathematical model, we introduce the indices, parameters, and decision variables:

Indices and parameters

i, j	Index of machines
k, l	Index of locations
tr	Index of transporter
M	The number of machines or the locations
$F_{i,j}$	Amount of material flow among machines i and j
$D_{k,l}$	The distance between the locations k and l
$C_{i,j}$	Unit material handling cost between machines i and j

$$F=\begin{pmatrix}0&0&0&0&7&0&0&0\\7&0&0&0&0&0&0&0\\0&8&0&7&0&0&0&0\\0&3&0&0&0&0&0&0\\0&0&0&0&0&0&4&0\\2&0&0&0&3&0&9&0\\0&0&3&0&0&0&0&4\\6&5&0&5&1&0&0&0\end{pmatrix}$$

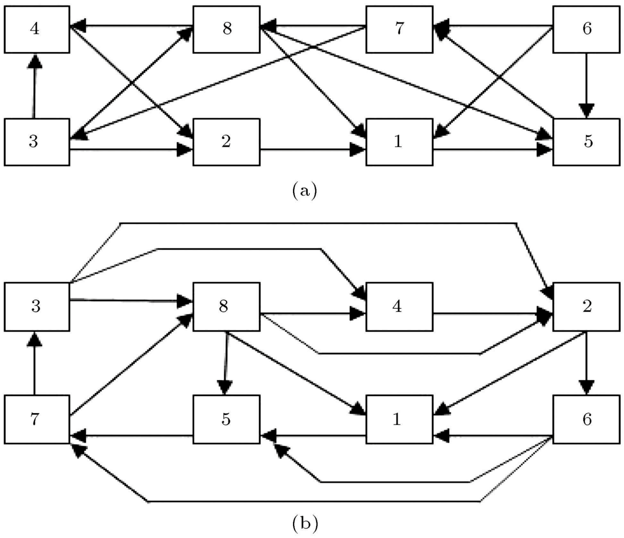


Figure 1. Solution to the eight-facility example: (a) Minimizing MHC, and (b) minimizing workflow interference [20].

FC^{tr} Fixed cost of establishing the transporter type tr

m^{tr} Maximum available transporter type tr

cap^{tr} Capacity of the transporter type tr

Decision variables

$$X_{i,k}\begin{cases}1, & \text{if the machine } i \\ & \text{is assigned to the location } k \\ 0, & \text{otherwise}\end{cases}$$

$$Y_{i,j}^{tr}\begin{cases}1, & \text{if the transporter } tr \text{ is selected to transfer} \\ & \text{parts from the machine } i \text{ to the machine } j \\ 0, & \text{otherwise}\end{cases}$$

$$\text{Min}Z_1 = \left(\sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{M}\sum_{l=1}^{M}\sum_{tr=1}^{TR}\left(\frac{F_{i,j}}{cap^{tr}}\right)\times C_{i,j} \times D_{k,l}\times X_{i,k}\times X_{j,l}\times Y_{i,j}^{tr}\right) + \sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{tr=1}^{TR} Y_{i,j}^{tr}\times FC^{tr}, \quad (1)$$

$$\text{Min}Z_2 = \text{Makespan}, \quad (2)$$

$$\sum_{k=1}^{M} X_{i,k} = 1, \quad \forall i = 1,2,...,M, \quad (3)$$

$$\sum_{i=1}^{M} X_{i,k} = 1, \quad \forall k = 1,2,...,M, \quad (4)$$

$$\sum_{tr=1}^{TR} Y_{i,j}^{tr} = sgn_{i,j}, \quad \forall i = 1,2,...,M-1; \quad \forall k = i+1,2,...,M; \quad (5)$$

$$sgn_{i,j}\begin{cases}1, & \text{if } F_{i,j} > 0 \\ 0, & \text{otherwise}\end{cases} \quad \forall i,j = 1,2,...,M \quad (6)$$

$$Y_{j,i}^{tr} = Y_{i,j}^{tr}, \quad \forall i,j = 1,2,...,M; \quad \forall tr = 1,2,...,TR, \quad (7)$$

$$\sum_{i=1}^{M-1}\sum_{j=i+1}^{M} Y_{i,j}^{tr} \leq m^{tr}, \quad \forall tr = 1,2,...,TR, \quad (8)$$

$$X_{i,k}, Y_{i,j}^{tr} \in \{0,1\}, \quad i,j,k = 1,2,..,M; \quad \forall tr = 1,2,...,TR. \quad (9)$$

The first term of objective function (1) is related to the variable MHC and the second term of it is related to fixed cost of the transporters. The first objective function of the problem is computed by analytical relationships and it focuses on minimizing the handling costs. Objective function (2), which is evaluated by ANNs, focuses on minimizing the whole processing completion time. Constraint set (3) states that each machine is assigned to a location and Constraint set (4) guarantees that each location is occupied by only one machine. Constraint set (5) allocates a transporter for movement between two special machines only when material flow exists. Constraint set (8) controls the maximum available number of each type of transporters. The decision variables are kept either at 1 or at 0 by Constraint set (9).

4. Proposed ANN-based optimization

Because of complexity and uncertainty of many real-life problems, it is very difficult to create a precise analytical model. In such complicated situations, simulation is proven as a powerful computer-based tool that can be used instead of the analytical models to study the behavior of complex real systems [21]. Even though simulation models are skillful in capturing complex system behaviors, simulation is essentially a test approach and the way to get the optimum solutions is not clear in it. Therefore, as Fu [21] pointed out, there is a need to develop algorithms that take advantage of the optimization technique, while being as accurate as simulation. The principle of simulation optimization is that during the optimization process, objective function and constraints are evaluated by simulation model.

Another weakness of simulation model comes from its requirement of a number of replications, which usually make simulation a very time consuming process. In order to take a step towards filling this gap, this paper proposes a novel simulation-based optimization framework, which integrates the simulation modeling, artificial neural network, and metaheuristic optimization algorithm. The framework is shown in Figure 2. A series of different scenarios are generated. Then, based on these scenarios, discrete event simulation model is run. This input-output data is used to train an ANN to approximate the objective function. ANN acts just like a tremendous intelligent brain, which is trained by simulation data and has the capability to estimate the makespan as fast as analytical relationships and as accurate as simulation models.

To search the solution space, a multi-objective optimization algorithm, called NSGA-II, has been combined with an adaptive local search. The NSGA-II showed the capacity to robustly solve large complicated multi-objective problems [22]. In our hybrid NSGA-II (H-NSGA-II), ANN is considered as a chromosomes

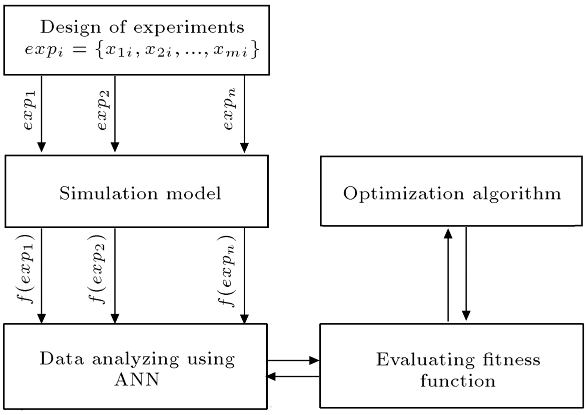


Figure 2. Framework of the proposed method.

fitness function evaluator. The combination of simulation, ANN, and optimization technique provides an effective means for the high complex optimization problems. For more information on simulation-optimization approaches, advances, and applications, we refer readers to recent reviews by Swisher et al. [23] and Fu et al. [24].

4.1. Artificial neural networks

Artificial Neural Networks (ANNs) are known as effective techniques for approximating non-linear model functions [25]. Considering the highly non-linear relation between the makespan and the selected transporters, the ANNs can be effectively applied to find this indefinite relation. ANNs consist of different interconnected processing elements that aim to solve a specific problem. These small computing elements are called neurons (Figure 3). The neuron takes inputs, processes them, and transfers the outputs. Neurons are connected to each other by links known as synapses, and associated with each synapse there is a weight factor. First, all the input signals, design configuration (X_i), transferred by synapses, have to be multiplied by their own weighting (W_i). Then, a special value bias (b) is added to the signals to generate a value (u). Finally, an activation function transfers the value (u) to the output (Y). The bias and activation function form a node. The output can be the input of other nodes. Through a learning algorithm, all the weights (W_i) are iteratively modified to minimize the difference between outputs (Y) and desired outputs. Finally, the trained neural network can be used to immediately predict the simulation results of new configurations.

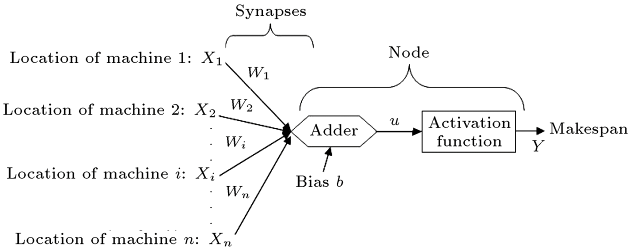

Synapses
Node
Location of machine 1: X_1
Location of machine 2: X_2
Location of machine i: X_i
Location of machine n: X_n
W_1
W_2
W_i
W_n
Adder
u
Activation function
Makespan
Y
Bias b

Figure 3. An example of a neuron for the interested problem.

Location of machines

X_1	X_2	X_3	X_4
3	1	4	1

The allocation of transporters

Y_{12}	Y_{13}	Y_{14}	Y_{23}	Y_{24}	Y_{34}
1	2	2	1	3	1

Figure 4. An example to input in ANN model.

There are many kinds of neural network models. Multi-Layered Perceptron Neural Networks (MLPNNs) with nonlinear transfer functions have been considered in the present study. They are purely empirical models that can theoretically mimic any relationship to any degree of precision ([26,27]). They consist of three layers, including one input layer, one or more hidden layers, and one output layer. The input layer consists of the decision variables associated with machines' location ($X_{i,k}$) and transporters' allocation considerations ($Y_{i,j}^{tr}$); and the output layer gives the outcome of the process or the makespan. An example of the inputs is given in Figure 4. In this configuration, machine 1 is assigned to location 3, machine 2 is assigned to location 1, etc. Also, materials between machines 1 and 2 are moved by transporter 1, materials between machines 1 and 3 are moved by transporter 2, etc.

4.2. Hybrid non-dominated sorting genetic algorithm

NSGA-II is one of the contemporary multi-objective evolutionary algorithms that exhibits high performance and has been widely applied in various disciplines. The algorithm makes use of a fast non-dominating sorting approach to discriminate solutions, which is based on the concept of Pareto dominance and optimality. The concept of Pareto dominance for minimization problem can be expressed as follows:

Consider a multi-objective model with a set of conflict objectives, $f(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), ..., f_n(\vec{x}))$, subject to $g(x) = (g_1(\vec{x}), g_2(\vec{x}), ..., g_m(\vec{x})) \leq 0$, where $\vec{x} \in X$. $\vec{x}$ is the decision vector and X is the feasible solution space. $f(\vec{x})$ is the vector-valued function and $g(x)$ is a vector of constraints. We say solution $\vec{a}$ dominates solution $\vec{b}$ if $f_i(\vec{a}) \leq f_i(\vec{b})$ $\forall i = 1, 2, ..., n$ and $\exists i : f_i(\vec{a}) < f_i(\vec{b})$.

The NSGA-II starts with random generation of population. The binary tournament selection selects the parents based on the rank and crowding distance. Then, genetic operations such as crossover and mutation are used to generate the child populations. The detail of the complete method can be found in Deb's paper [28]. Also, in order to perform a careful search around the most promising area, an adaptive local search is combined with NSGA-II. The local search operator helps to intensify the search in various areas pointed by the genetic mechanisms that in return can improve convergence towards real Pareto front. The local search is applied in a heuristic manner so that it is only applied over some special generations. The main components of the algorithm and the concepts of adaptive local search are explained in the next sections.

4.2.1. Non-dominated sorting

Before selection is performed, every individual (chromosome) in the population is assigned a rank based on non-domination. First, the non-dominated solutions are assigned rank 1. Then, the individuals of rank 1 are eliminated and non-dominated solutions are assigned rank 2. This process is repeated until all individuals are classified. The crowding distance metric proposed by [28] is utilized, where the crowding distance of an individual is the perimeter of the rectangle with its nearest neighbors at diagonally opposite corners. Thus, if two individuals have the same rank, the one with a larger crowding distance is better.

4.2.2. Adaptive local search scheme

The H-NSGA-II presented in this article uses adaptive Simulated Annealing (SA) algorithm as the local search because of its good convergence rate. In order to adapt SA to optimize multiple objectives simultaneously, the Pareto dominance concept is utilized; this means that a dominated or non-dominated neighbor is treated like a worse neighbor and moving towards it is done with a certain probability. The rest of the algorithm is just like a typical SA procedure. Since the SA is applied to all the Pareto front solutions and this may lead to high computational efforts, local search scheme is applied only in some generations. We develop a heuristic index, called Similarity Coefficient (SC), based on the executed local search. First, we compute the SC for each pair of chromosomes by Eq. (10):

$$\mathrm{SC}_{ab}=\frac{\sum_{i=1}^{M}\left\{\partial(X_{ia},X_{ib})+\sum_{i=1}^{M}\sum_{j=1}^{M}\{\partial(Y_{ija},Y_{ijb})\}\right\}}{\overline{M}}, \tag{10}$$

where X_{ia} and X_{ib} are the locations of machine 'i' in the chromosomes 'a' and 'b' and Y_{ija} and Y_{ijb} are the selected transporters to handle pares between machine 'i' and machine 'j' in the chromosomes a and b. $\overline{M'}$ is the number of genes which are not empty. $\partial(\alpha,\beta)$ is the similarity between two especial genes and is expressed by Eq. (11);

$$\partial(\alpha,\beta)=\begin{cases}1 & \text{if } \alpha=\beta\\ 0 & \text{otherwise}\end{cases} \tag{11}$$

The average similarity coefficient of the population is calculated as follows:

$$\overline{\mathrm{SC}}=\frac{\sum_{a=1}^{N-1}\sum_{b=a+1}^{N}\mathrm{SC}_{\ ab}}{\binom{N}{2}}, \tag{12}$$

in which N is the number of chromosomes in population. Finally, considering a pre-defined threshold similarity coefficient (φ) and the obtained average similarity coefficient, the local search scheme will be automatically incorporated into the NSGA-II loop as follows:

$$\begin{cases}\text{apply local search} & \\ \text{ scheme to NSGA-II loop} & \text{if } \overline{\mathrm{SC}}<\varphi\\ & \\ \text{do not use local search} & \\ \text{in NSGA-II loop} & \text{otherwise}\end{cases}$$

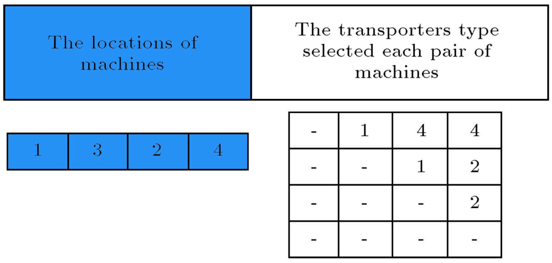
4.2.3. Chromosome structure

The proposed chromosome consists of two matrices, each representing a special area of decision making. The first part shows how machines are placed in locations; the second part represents the allocation of transporters to each pair of machines. Figure 5 shows an example chromosome in which machine 1 is placed in location 1, machine 3 is placed in location 2, machine 2 is placed in location 3, and machine 4 is placed in location 4. The materials between machine 1 and machine 2 are moved by transporter 1, materials between machine 1 and machine 3 are moved by transporter 4, etc.

4.2.4. Crossover operator

The crossover operator combines two chromosomes to produce a new chromosome. We apply two crossover types that only generate feasible solutions. The first proposed crossover operator has the following steps;

1. Strings related to machine layout are selected;
2. A cross point is randomly selected (cross points $1,\dots,M$);
3. The machine numbers before the cross point of parent 1 are copied in the offspring. The remaining machine numbers are put into empty positions according to their relative locations in parent 2.


The locations of machines
The transporters type selected each pair of machines
1
3
2
4

-	1	4	4
-	-	1	2
-	-	-	2
-	-	-	-

Figure 5. An example of chromosome representation.

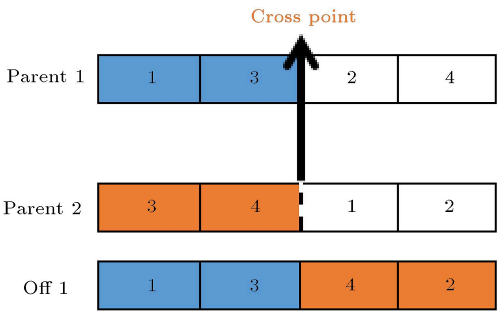


Figure 6. An example of crossover type 1.

Figure 6 shows the crossover for a problem with 4 machines.

In order to cross the sub-matrices related to transporters, a uniform crossover operator is used in which, first, a Boolean matrix is generated; then, in cells in which Boolean matrix is equal to one, the data are filled similar to the offspring from parent 1 and in cells in which Boolean matrix is equal to zero, the offspring is filled similar to that from parent 2.

In order to apply the crossover of type 2, a 1×2 reference vector is first generated. Then, if the first call of reference vector is 1, the machine layout matrix of offspring will be copied from parent 1; if the first call of reference vector is 0, the machine layout matrix of offspring will be copied from parent 2. This way is repeated for transporters matrix (Figure 7).

4.2.5. Mutation

In this paper, the mutation operation is performed only on the machine layout matrix. First, a chromosome is randomly selected. Then, to create a hard change, the genes of the selected chromosome are arranged inversely (Figure 8).

4.2.6. The neighbourhood structure

2-change neighborhood is used to define the neighborhood in the local search algorithm. First, we select a sub-matrix randomly. Then, two different cells are selected randomly and the numbers in these cells are exchanged. Figure 9 shows a neighborhood in which sub-matrix 1 changes in genes 2 and 4.

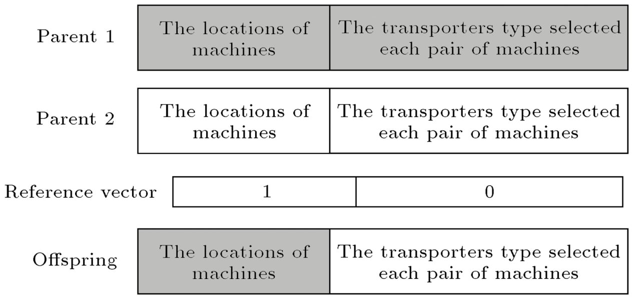

Parent 1
The locations of machines
The transporters type selected each pair of machines
Parent 2
The locations of machines
The transporters type selected each pair of machines
Reference vector
1
0
Offspring
The locations of machines
The transporters type selected each pair of machines

Figure 7. An example of crossover type 2.

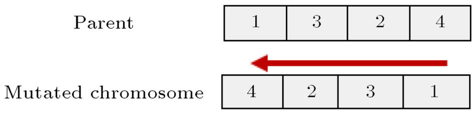


Figure 8. An example of mutation operator.

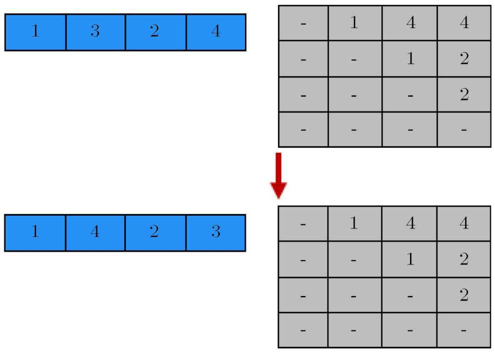


Figure 9. An example of neighborhood operator.

4.2.7. Cooling schedule

The performance of this algorithm also depends on the cooling schedule, which is relevant to the temperature updating function. In the proportional decrement scheme, temperature decreases at steps k and $k+1$ of the outer loop by:

$$T_{k+1} = \alpha T_k, \tag{13}$$

where α is the cooling rate and is obtained by some experiments.

4.2.8. Stopping criterion

To limit the number of replications of both NSGA-II and SA algorithms, some convergence experiments are performed and the best criterion is applied as follows.

NSGA-II will be stopped in the case when total number of iterations reaches a predefined number that is set according to the result of experimental design. For stopping SA in a temperature level, first, we define the set of m iterations as a round. If the mean change between two successive rounds remains fixed within 0.95% confidence interval, we reduce the temperature. For the outer loop of SA, a certain number of iterations is set as the stopping criterion, and this value is determined according to the result of experiment.

5. Computational experiments

To demonstrate and validate the simulation-based optimization framework proposed in this paper, a real-life production system is studied. This case study involves 6 machines, named $M1$, $M2$, $M3$, $M4$, $M5$, and $M6$, and 6 products. The demand of product is known (Table 1) and so the material flow between machines is clear. The distance between locations and capacity and speed of the available transporters are, respectively, shown in Tables 2 and 3. The MHC is fixed at 1 \$/meter. The model is coded in Enterprise

Table 1. Demand rate of products (per day).

	Demand	Sequence
Product 1	300	1→3→6
Product 2	200	1→2→1→3→5→6
Product 3	150	1→3→5→2→6
Product 4	220	1→2→3→6
Product 5	150	1→2→4→6
Product 6	90	1→2→6

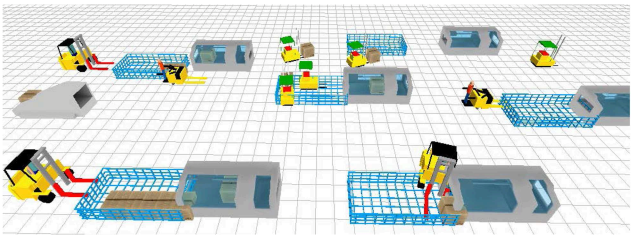


Figure 10. Representation of the real production system in ED simulation software.

Dynamics 8 developer (ED). A graphical representation of the system is shown in Figure 10.

The computational experiments are accomplished in three phases. In the first phase, we validate the simulation model through real data analysis. In the second phase, an ANN is developed and its accuracy in predicting the makespan is investigated. In the last phase, a comparison between H-NSGAII optimization based on artificial neural networks and normal NSGA-II optimization based on artificial neural networks is presented. The metaheuristic optimization algorithms are coded by MATLAB and are implemented in a desktop with a 3.20-GHz CPU running Windows 7 (64 bit).

5.1. Simulation model validation

In this section, we investigate whether the simulation model behaves in accordance with the actual system or not. To assess this, simulation results of 30 days are compared with the actual measurement data in terms of makespan for each day. The results are presented in Table 4. As presented in this table, there is an error less than 2%, showing high accuracy of the designed simulation model.

5.2. Developing an artificial neural network

In this section, a set of proper scenarios are selected for the proposed simulation model to generate training data. Then, a neural network is trained using this data set. Finally, some validation tests are conducted and the ability of designed ANN to predict the makespan is investigated. In this case, there are 16 factors, including 6 factors related to machines' location, each in 6 levels, and 10 factors related to transporters selection, each in 4 levels. The total number of possible experiments is $6^6 * 10^4$, which is so computationally intractable. To overcome this problem, the configuration of all experiments is generated by Uniform Design (UD). UD was proposed by Fang [29]. Its most important feature is that it decreases the number of experimental configurations, especially when the experimental region has many factors and multiple levels. According to the uniform design table of the form $U_n(6^{16})$, the number of experiments can be in the range of 17 to 30 (n is the desired number of experiments). Because n should be the common multiple of all the levels of factors, it can only be 24. Therefore, the uniform design table U_{24}, shown in Table A.1 in Appendix A, is selected. In Appendix A, for factors 1-6 (machines location) the numbers from 1 to 4 signal location 1, the numbers from 5 to 8 signal location 2, etc. For factors 7-16 (transporters selection), the numbers from 1 to 6 signal transporters type 1, the numbers from 7 to 12 signal transporters type 2, etc. The simulation results of all 24 experiments are presented in Appendix A.

The present problem had 16 control factors as the input neurons, and the makespan as the single output. Thus, our ANN includes 16 input nodes and one output node. The number of nodes in the hidden layer can be estimated by Eq. (14) proposed by Chen and Yang [30]:

$$h = \frac{i+o}{2} + \sqrt{N}, \qquad (14)$$

where i is the number of input nodes, o is the number

Table 3. Travel path distance between locations (meter).

From/to	1	2	3	4	5	6
1	0	10	20	15	20	30
2	10	0	10	20	15	20
3	20	10	0	25	20	15
4	15	20	25	0	10	20
5	20	15	20	10	0	10
6	30	20	15	20	10	0

Table 2. Characteristics of the transporters.

	Capacity	Speed (m/s)	Fixed cost ($)	No of available transporters
Transporter 1	10	Normal (3,0.5)	500	4
Transporter 2	15	Normal (7,2)	700	3
Transporter 3	10	Normal (3,1)	450	5
Transporter 4	10	Normal (8,2.5)	600	2

Table 4. Results of real system and simulation model.

Days no.	**Obtained makespan (minute)**		**% Error**
	Real system	**Simulation**	
1	387	389	0.52
2	340	347	2.06
3	390	401	2.82
4	395	386	2.28
5	384	394	2.60
6	364	383	5.22
7	400	397	0.75
8	400	407	1.75
9	386	393	1.81
10	342	348	1.75
11	399	390	2.26
12	358	355	0.84
13	352	348	1.14
14	396	391	1.26
15	388	394	1.55
16	382	378	1.05
17	340	351	3.24
18	383	391	2.09
19	375	379	1.07
20	342	347	1.46
21	391	380	2.81
22	363	372	2.48
23	388	393	1.29
24	356	370	3.93
25	367	371	1.09
26	399	411	3.01
27	368	361	1.90
28	372	374	0.54
29	385	382	0.78
30	378	387	2.38
Average	375	379	1.92

Table 5. The optimal ANN parameters.

Parameter	**value**
Hidden layers	1
Nodes in hidden layer	14
Learning rate	0.015

of output nodes, N is the number experiments, and h is the number of nodes in the hidden layer. Consequently, $h = \frac{16+1}{2} + \sqrt{24} \approx 14$. The optimal ANN configuration, which was found experimentally, is summarized in Table 5.

After determining the structure of ANN, back-propagation algorithm is carried out to train the network. The back-propagation algorithm has powerful approximation capacity and is applicable to both binary and continuous inputs. The type of transfer function employed in this work is a sigmoid function (Eq. (15)) at hidden layer and a linear transfer function at output layer. Neural Network Toolbox V4.0 of MATLAB mathematical software was used for makespan prediction:

$$f(x) = \frac{1}{1 + e^{-x}}. \tag{15}$$

For inter-comparisons between the simulated and measured makespans using the ANN model, two performance measures, i.e. the Root Mean Squared Error (RMSE) and coefficient of determination (R^2), are used as follows [31]:

$$\text{RMSE} = \frac{1}{m}\sqrt{\sum_{i=1}^{M}\left(\frac{Y_i - Y_i}{Y_i}\right)^2} \tag{16}$$

where m is the number of samples, Y_i is the actual response of sample i, and $\hat{Y}_i$ is the predicted response of sample i. According to RMSE $= 0.01545$ and $R^2 = 0.9771$, the ANN model has been properly trained and has good quality predictions.

Usually, it is necessary to check the fitted model to ensure that it provides an adequate approximation to the new input data. To this aim, the ANN results are compared with respect to their deviations from the simulation results for 15 new trials, which do not belong to the training data set. We propose the confidence intervals to evaluate the overall performance of the neural network, because interval estimates are much more useful than point estimates for decision-making. Each trial is simulated for 10 replications and $E(Y_{sim})$ and $VAR(Y_{sim})$ are combined to form confidence intervals for each trial (Eq. (17)):

$$\text{Interval} = E(Y_{sim}) \pm t_{\frac{\alpha}{2},r-r}\sqrt{VAR(Y_{sim})}. \tag{17}$$

As show in Figure 11, in each experiment trail, the predicted result by ANN (red points) falls within the interval obtained by the simulation, and so the capabilities of structured ANN will be proven.

5.3. Evaluation of the proposed metaheuristic algorithm

We tested the optimization framework on numerous random data sets that differ with respect to input parameters such as products demand, transporters capacity, material handling costs, etc. The main challenge in comparing the two multi-objective algorithms is that they do not try to find one optimal solution, but a set of Pareto solutions. A good

Table 6. The parameters of the algorithms and their levels.

Algorithm	Parameter	Description	Low level	High level	Optimum level
H-NSGA-II	Pc	Percent of cross over	0.6	0.9	0.85
	Pm	Percent of mutation	0.05	0.15	0.08
	$N-pop$	Initial size of pop	50	150	130
	$N-iter$	Number of iteration	50	300	230
	α	Temperature decrement rate	0.96	0.99	0.985
	m	Number of iterations inside each inner loop	5	30	20
	$Out-L$	Number of outer loop iterations	50	200	127
NSGA-II	Pc	Percent of cross over	0.6	0.9	0.80
	Pm	Percent of mutation	0.05	0.15	0.14
	$N-pop$	Initial size of pop	50	150	140
	$N-iter$	Number of iterations	50	300	260

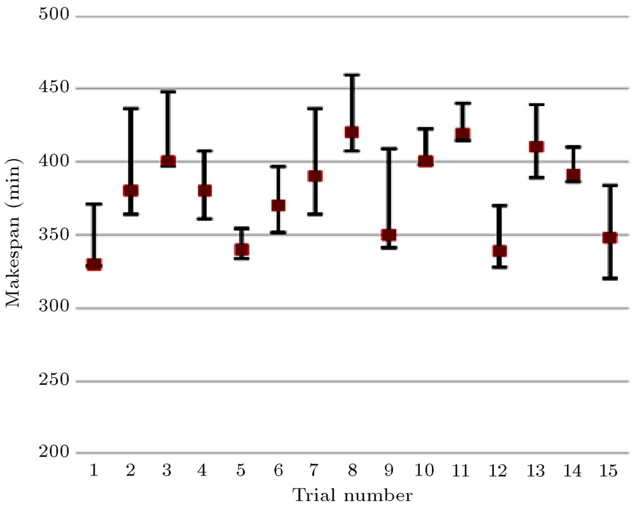


Figure 11. Comparison of the results obtained by simulation and ANN model.

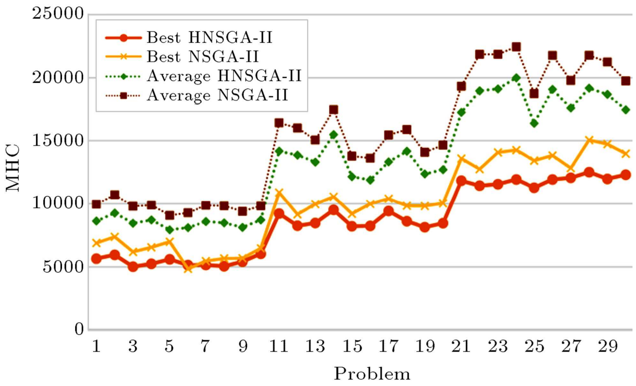


Figure 12. Performance of the algorithms in terms of MHC.

Pareto front is characterized by (i) converging to the real Pareto-optimal front, and (ii) maximizing the diversity of the Pareto solutions. Consequently, the quality of multi-objective optimization algorithms is often difficult to define precisely by any single performance metric. A list of indicators has been introduced over the past few decades. To analyze the performance of the H-NSGA-II, we compare its Pareto fronts with those obtained by common NSGA-II in terms of metrics Pareto Ratio (PR), Spacing (S), Overall Pareto Spread (OPS), and Computational time (CPU time). PR is the ability of a multi-objective optimization method to produce non-dominated solutions. S measures the standard deviation of the distances among solutions of the Pareto front. The smaller the value of the spacing metric, the better are the solutions spread along the obtained front. OPS quantifies how widely the non-dominated solutions spread over the objectives pace considering all the objectives.

Before evaluating the algorithms, both algorithms are tuned by response surface methodology. The explored bound and optimum level of each parameter are presented in Table 6. Then, the parameters of the algorithms are fixed to their optimum level and test problems are solved by the algorithms. Figures 12 and 13 present the results of the proposed H-NSGA-II and normal NSGA-II in the context of MHC and makespan. As shown in Figure 12, in terms of the best MHC index, H-NSGA-II outperforms NSGA-II in all the test problems, except for P6. Also, in terms of the best makespan, H-NSGA-II finds better results than NSGA-II in all problems, except for P5 (Figure 13). Furthermore, the average values for both objectives considerably improve with the introduction of the adaptive scheme. We also analyze the performances of the algorithms using statistical tests.

Appendix B highlights the values of PR, S, OPS, and CPU time performance measures. As discussed above, these scales are used to measure convergence

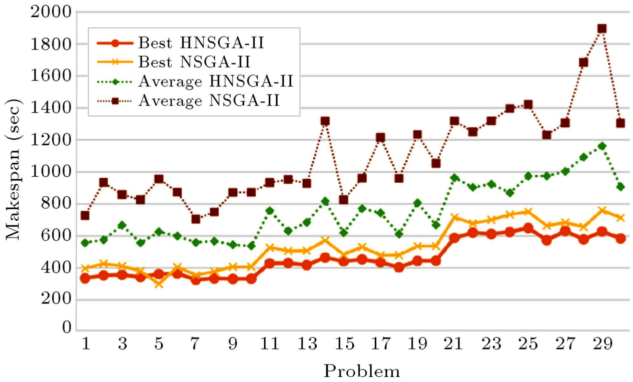


Figure 13. Performance of the algorithms in terms of makespan.

and diversity of Pareto-front for handling multiple objectives. As presented in Appendix B, the H-NSGA-II approach with adaptive local search shows better behavior than the approach without local search. In order to statistically analyze the results shown in Appendix B, we use Student's t-test. The hypothesis is:

$$H_0: \ \mu_1 - \mu_2 = 0,$$

$$H_1: \ \mu_1 - \mu_2 \neq 0,$$

in which μ_1 is the obtained average value by H-NSGA-II and μ_2 is the obtained value by NSGA-II algorithm. The results are shown in Table 7. According to Table 7, it can be seen that in terms of PR and CPU time index, there is no meaningful difference between algorithms. But, H-NSGA-II outperforms NSGA-II with respect to spacing and OPS metrics. The last column of Table B.1 in Appendix B shows the computational times of the algorithms for each test problem. Although there was no significant CPU time difference between algorithms, according to "Mean" statistics, it can be claimed that H-NSGA-II is relatively faster than NSGA-II. This statement implies that although embedding the SA in some generations of NSGA-II may lead to increase in computational time, large increase in solution quality and speed of convergence can compensate it.

6. Conclusion and remarks

This paper presents a new model and a novel solving approach to solve facility layout optimization problems for manufacturing systems with dynamic characteristics. The proposed approach integrates computer simulation, ANN, and H-NSGA-II techniques to overcome the limitations of traditional layout optimization methods. The main motivation behind choosing this problem is the necessity for integrating decisions for the layout of machines and material handling vehicles. The results show that the application of an ANN model can predict the makespan in a complicated manufacturing system, efficiently. Despite the previous models, this paper considers real aspects of material handling such as random breakdowns, random processing times, and waiting times during the handling process. Since these aspects are stochastic variables and theoretically difficult to obtain, the ANN model, which has been trained by the simulation model results, computes the problem makespan.

In the experimental results, there were 6^6 alternatives for the layout of machines and 10^4 alternatives for selecting the transporters; therefore, there were a total number of $6^6 * 10^4$ designs. However, in the experiment, only 24 simulation configurations were run to train the data set needed for ANN model. A case study was presented and the simulation results were given for validation of the ANN results. In all experiments, the ANN precision was 95%. Finally, a hybrid non-dominated sorting genetic algorithm was proposed to search the solution space. The performance of the proposed algorithm was compared with that of the normal non-dominated sorting genetic algorithm. In terms of PR and CPU time criteria, both algorithms

Table 7. Results of two-sample T-test.

	Source	**No**	**Mean**	**St. Dev**	**95% CI for difference**	***T*-Value**	***P*-Value**
PR	H-NSGA-II	30	0.676	0.206	(-0.0697, 0.1217)	0.54	0.588
	NSGA-II	30	0.650	0.168			
OPS	H-NSGA-II	30	0.557	0.177	(0.0282, 0.1812)	2.75	0.008
	NSGA-II	30	0.452	0.110			
Spacing	H-NSGA-II	30	6.02	5.33	(-9.29, -0.27)	-2.14	0.038
	NSGA-II	30	10.8	11.0			
CPU time	H-NSGA-II	30	1116	511	(-405, 169)	-0.82	0.413
	NSGA-II	30	1234	579			

performed similar. But, in terms of OPS and spacing metric criteria, the performance of the proposed hybrid algorithm was statistically better. For future research, one may extend unequal area constraint of departments, which demonstrates a more realistic representation of real-world manufacturing facilities. Also, other factors affecting the makespan, such as human factors, can be investigated.

References

1. Singh, S.P. and Sharma, R.R.K. "A review of different approaches to the facility layout problems", *Int. J. Adv. Manuf. Technol.*, **30**, pp. 425-433 (2006).

2. Tompkins, J.A., White, J.A., Bozer, Y.A. and Tanchoco, J.M.A., *Facilities Planning*, Wiley, New York (2003).

3. Meller, R.D. and Gau, K.Y. "The facility layout problem: Recent and emerging trends and perspectives", *J. Manuf. Syst.*, **15**(5), pp. 351-366 (1996).

4. Azadivar, F. and Wang, J. "Facility layout optimization using simulation and genetic algorithms", *Int. J. Prod. Res.*, **38**(17), pp. 4369-4383 (2000).

5. Egbelu, P.J. and Tanchoco, J.M.A. "Potentials for bidirectional guide-path for automated guided vehicle based systems", *Int. J. Prod. Res.*, **24**(5), pp. 1075-1097 (1986).

6. Rosenblatt, M.J. "The facilities layout problem: A multi-goal approach", *Int. J. Prod. Res.*, **17**(4), pp. 323-332 (1979).

7. Savsar, M. "Flexible facility layout by simulation", *Comput. Ind. Eng.*, **20**(1), pp. 155-165 (1991).

8. Morris, J.S. and Tersine, R.J. "A simulation comparison of process and cellular lay-outs in a dual resource constrained environment", *Comput. Ind. Eng.,* **26**(4), pp. 733-41 (1994).

9. Pagell, M. and Melnyk, S.A. "Assessing the impact of alternative manufacturing layouts in a service setting", *J. Oper. Manage.*, **22**(4), pp. 413-29 (2004).

10. Zhou, F., AbouRizk, S.M. and AL-Battaineh, H. "Optimization of construction site layout using a hybrid simulation-based system", *Simul. Modell. Pract. Theory*, **17**, pp. 348-363 (2009).

11. Jithavech, I. and Krishnan, K.K. "A simulation-based approach for risk assessment of facility layout designs under stochastic product demands", *Int. J. Adv. Manuf. Technol.*, **49**(1), pp. 27-40 (2009).

12. Gupta, R.M. "Flexibility in layouts: A simulation approach", *Material Flow.*, **3**, pp. 243-250 (1986).

13. Grajo, E.S. "Strategic layout planning and simulation for lean manufacturing a layOpt tutorial", *Proceedings of the 1996 Winter Simulation Conference*, pp. 564-568 (1996).

14. Azadivar, F. and Tompkins, G. "Simulation optimization with quantitative variables and structural model changes: a genetic algorithm approach", *Eur. J. Oper. Res.*, **113**, pp. 169-182 (1999).

15. Kulturel-Konak, S., Smith, A.E. and Norman, B.A. "Layout optimization considering production uncertainty and routing flexibility", *Int. J. Prod. Res.*, **42**(21), pp. 4475-4493 (2004).

16. Altuntas, S. and Selim, H. "Facility layout using weighted association rule-based data mining algorithms: Evaluation with simulation", *Expert Syst. Appl.*, **39**, pp. 3-13 (2012).

17. Dombrowski, U. and Ernst, S. "Scenario-based simulation approach for layout planning", *8th CIRP Conf. on Intell. Comput in Manuf. Eng.*, **12**, pp. 354-359 (2013).

18. Karpe, A.B., Kulkarni, C.N. and Jahagirdar, R.S. "Simulation methodology for facility layout problems", *Int. J. Res. Eng. Sci. Technol.*, **1**, pp. 1-10 (2011).

19. Azadeh, A., Nazari, T. and Charkhand, H. "Optimization of facility layout design problem with safety and environmental factors by stochastic DEA and simulation approach", *Int. J. Prod. Res.*, **1**(1), pp. 12-17 (2014).

20. Chiang, W., Kouvelis, P. and Urban, T. "Single and multiobjective facility layout with workflow interference considerations", *Eur. J. Oper. Res.*, **174**(3), pp. 1414-1426 (2006).

21. Fu, M.C. "Optimization for simulation: Theory vs. practice", *Inf J. Comput.*, **14**, pp. 192-215 (2002).

22. Wang, G., Yan, Y., Zhang, X., Shangguan, J. and Xiao, Y. "A simulation optimization approach for facility layout problem", *Proc. of the IEEE Int'l. Conf. on Ind. Eng. Eng. Manage.*, pp. 734-738 (2008).

23. Swisher, J.R., Hyden, P.D., Jacobson, S.H. and Schruben, L.W. "A survey of recent advances in discrete input parameter discrete-event simulation optimization", *IIE Trans.*, **36**, pp. 591-600 (2004).

24. Fu, M.C., Glover, F.W. and April, J. "Simulation optimization: a review, new developments, and applications", In: Kuhl ME, Steiger NM, Joines JA, Eds., *Proceedings of the 2005 Winter Simulation Conference* (2005).

25. Ashtiani, A., Mirzaei, P.A. and Haghighat, F. "Indoor thermal condition in urban heat island: Comparison of the artificial neural network and regression methods prediction", *Energy Build.*, **76**, pp. 597-604 (2014).

26. Funahashi, K. "On the approximate realization of continuous mappings by neural networks", *Neural Netw.*, **2**(1), pp. 83-92 (1989).

27. Hornik, K., Stinchcombe, M. and White, H. "Multilayer feed forward networks are universal approximators", *Neural Netw.*, **2**, pp. 359-66 (1989).

28. Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Trans. Evol. Comput.*, **6**, pp. 182-197 (2002).

29. Fang, K.T. "The uniform design: application of number-theoretic methods in experimental design", *Acta Math. Appl. Sin.*, **3**, pp. 363-372 (1980).

30. Chen, M.C. and Yang, T. "Design of manufacturing systems by a hybrid approach with neural network meta modelling and stochastic local search", *Int. J. Prod. Res.*, **40**, pp. 71-92 (2002).

31. Geyikci, F., Kilic, E., Coruh, S. and Elevli, S. "Modeling of lead adsorption from industrial sludge leachate on red mud by using RSM and ANN", *Chem. Eng. J.*, **183**, pp. 53-59 (2012).

Appendix A

The uniform design of experiments is shown in Table A.1.

Appendix B

The multi objective performance measures obtained for the algorithms is shown in Table B.1.

Biographies

Parham Azimi was born in 1974. He received his BS, MS, and PhD degrees in Industrial Engineering from Sharif University of Technology in 1996, 1998, and 2005, respectively. Currently, he is the Assistant Professor of Industrial Engineering at Islamic Azad University (Qazvin Branch). His research interests include optimization via simulation techniques, facility layout problems, location allocation problems, graph theory, and data mining techniques.

Parham Soofi Parham Soofi is a PhD candidate at the Faculty of Industrial and Mechanical engineering, Azad university, Qazvin branch. His research fields are MCDM and Simulation optimization. He has been also a project manager at the Iranian Fuel Conservation Organization for designing and production of 500,000 pcs. of CNG valves and also exporting of LPG valves to the European markets including Germany and the Netherlands.

Table A.1. Uniform design of experiments.

Exp.	**Factors**																**Makespan**
no.	X_1	X_2	X_3	X_4	X_5	X_6	Y_{12}	Y_{13}	Y_{23}	Y_{24}	Y_{26}	Y_{35}	Y_{36}	Y_{46}	Y_{52}	Y_{56}	**(min)**
1	24	2	5	16	10	19	13	9	1	8	4	8	3	10	21	10	429
2	8	24	20	14	1	12	15	8	2	15	24	15	12	11	2	5	672
3	9	4	17	23	16	19	7	12	12	20	21	20	7	4	22	20	570
4	2	17	11	6	15	22	4	11	3	21	11	4	20	9	19	21	414
5	4	10	5	20	23	12	23	23	6	14	14	14	11	6	15	4	712
6	14	7	22	8	6	24	10	16	15	13	1	12	24	7	20	1	514
7	1	8	21	12	14	20	12	10	24	10	16	21	17	24	6	6	957
8	11	23	4	15	11	8	20	13	17	1	5	17	22	5	8	22	408
9	18	13	3	11	2	16	5	24	19	17	12	16	10	19	23	24	737
10	7	3	14	17	23	11	17	4	8	7	7	22	13	20	17	16	868
11	21	12	19	2	22	17	2	5	9	4	13	18	18	3	14	9	344
12	16	17	6	3	7	23	22	2	7	22	20	11	23	17	11	14	1003
13	23	18	12	3	9	7	21	14	22	19	18	24	6	12	16	3	1028
14	19	1	23	9	13	8	24	15	5	3	15	3	8	21	7	23	431
15	13	4	18	21	12	6	1	17	10	23	2	19	1	23	9	7	739
16	17	22	15	13	21	1	15	18	13	9	23	7	21	22	24	11	667
17	15	6	3	11	20	24	6	3	18	12	22	13	4	8	3	19	457
18	22	19	13	2	6	12	9	20	4	11	6	23	14	13	4	18	692
19	12	1	6	15	18	24	18	22	14	24	8	6	16	15	1	8	834
20	3	21	21	10	24	18	19	6	20	16	3	10	5	16	13	15	794
21	20	11	16	21	5	2	14	7	23	18	10	1	15	2	5	13	529
22	5	3	20	13	9	15	3	21	21	5	19	9	19	14	12	17	449
23	10	16	2	24	17	7	7	1	16	2	9	2	9	18	18	2	290
24	6	20	8	3	12	21	11	19	11	6	17	5	2	1	10	12	317

Table B.1. Multi-objective performance measures obtained for the algorithms.

No. Problem	Pareto Ratio (PR)		Overall Pareto Spread (OPS)		Spacing (S)		CPU time (sec)	
	H-NSGA-II	NSGA-II	H-NSGA-II	NSGA-II	H-NSGA-II	NSGA-II	H-NSGA-II	NSGA-II
P1	0.808	0.593	0.520	0.399	2.57	8.2	571	610
P2	0.818	0.607	0.652	0.628	3.26	5.93	489	525
P3	0.898	0.714	0.531	0.576	1.66	2.17	505	544
P4	0.880	0.696	0.695	0.600	6.73	9.55	436	441
P5	0.870	1.352	0.236	0.435	3.8	11.51	642	652
P6	0.748	0.787	0.283	0.241	4.6	2.56	566	610
P7	0.499	0.737	0.598	0.512	5.33	8.99	664	685
P8	0.834	0.431	0.195	0.416	2.14	11.11	521	562
P9	0.981	0.806	0.398	0.458	8.38	0.92	609	638
P10	0.959	0.663	0.748	0.418	2.19	18.95	400	394
P11	0.662	0.717	0.842	0.531	5.05	10.83	1038	1182
P12	0.895	0.660	0.566	0.669	7.96	0.81	1113	1220
P13	0.520	0.635	0.471	0.418	2.55	8.55	975	1083
P14	0.980	0.735	0.780	0.478	6.15	2.08	964	1012
P15	0.942	0.729	0.613	0.361	0.77	1.99	913	954
P16	0.662	0.648	0.458	0.354	3.04	3.66	963	1038
P17	0.465	0.690	0.856	0.239	18.86	39.54	1226	1384
P18	0.566	0.727	0.457	0.456	10.84	19.36	1294	1369
P19	0.674	0.632	0.516	0.510	1.69	3.55	1096	1231
P20	0.426	0.640	0.748	0.400	4.93	1.71	1146	1264
P21	0.500	0.433	0.433	0.306	10.22	18.66	1996	2181
P22	0.556	0.606	0.489	0.394	0.9	1.3	1767	1910
P23	0.485	0.586	0.462	0.546	2.87	6.25	1791	2078
P24	0.660	0.480	0.438	0.507	1.66	2.68	1933	2235
P25	0.590	0.438	0.749	0.392	9.92	17.18	1612	1863
P26	0.680	0.674	0.408	0.643	10.75	17.29	1749	1993
P27	0.360	0.550	0.856	0.529	23.52	36.98	1568	1798
P28	0.318	0.472	0.547	0.445	2.33	5.36	1752	1968
P29	0.243	0.618	0.695	0.390	3.36	7.54	1648	1837
P30	0.793	0.432	0.471	0.319	12.58	38.73	1538	1769