

Using an intelligent algorithm for performance improvement of two-sided assembly line balancing problem considering learning effect and allocation of multi-skilled operators

Hassan Gharoun¹, Mahdi Hamid^{1*}, Hossein Iranmanesh¹, Reza Yazdanparast¹

¹School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran
h.gharoun@ut.ac.ir, m.hamid31400@ut.ac.ir, shiranmanesh@ut.ac.ir, r.yazdanparast@ut.ac.ir

Abstract

Two-sided assembly lines have been extensively studied due to their application in various auto industries. This paper investigates balancing problem type-II, which serves to minimize cycle time and consider learning effect based on a predefined workstation and costs pertaining to the assignment of operators with various skills. To this end, an integrated approach based on discrete event simulation (DES), artificial neural network (ANN), and data envelopment analysis (DEA) is utilized to optimize the performance of two-sided assembly line balancing (2S-ALB) problem type-II. The developed approach is applied to a real case study. Since many scenarios (suggestions for production line improvement) are needed for the simulation, the 2^k Factorial design of experiment (DOE) is used to reduce their number. ANN and DEA were then used to select the best scenarios. It has been shown that incorporating learning effect and multi-skilled operators can improve the performance of 2S-ALB problem type-II better than does the conventional approach.

Keywords: Two-sided assembly line, discrete event simulation, neural network, data envelopment analysis (DEA), learning effect

Motivation and Significance

Two-sided assembly lines are a type of assembly line in which work is done on both sides of the line. In two stations facing each other in two-sided assembly lines, if the operation on one side is finished sooner, it creates an idle time. Also, studies examining the production line have concluded that work process times are compressible. Considering the learning effect is a common approach to decreasing work process time. Accordingly, this study was conducted with the aim of minimizing cycle time and the operators' idle time by examining the learning effect in balancing a two-sided multi-model assembly line problem. Previous studies did not take into account the impact of learning effect on the operator assignment problem in balancing the two-sided assembly problem. The modeling of a two-sided assembly line considering the simultaneous learning effect and multi-skilled operators' allocation can be a complex and non-linear system that consists of different parameters. In this paper, an integrated simulation-DOE-ANN-DEA approach was proposed for the first time to find an optimum solution for EPALP.

1- Introduction

Assembly line is a series of sequential workstations that operate a set of tasks. Some of these assembly lines are one-sided and some others are two-sided. In the former, the assembling operation is performed

*Corresponding author

only on one side of the line, to which the operators and equipment are deployed. In the latter, however, tasks are operated on both left and right sides. This type of line is a major assembly line for putting together large-size products such as trucks and automobiles. Two-sided assembly lines have several advantages, including the reduction in the number of operators, a decrease in equipment costs, and lowering the length of the line as well as handling costs (Bartholdi, 1993). In these lines, any two stations facing each other are known as mated stations. In some mated stations, one side may wait for the other to finish its task before the next task is started by both sides. This type of restriction (known as interference) only happens on two-sided assembly lines (Simaria and Vilarinho (2009). Thus, in a mated station, if the operation on one side is finished earlier, idle time will emerge. Therefore, in order to decrease the idle time as much as possible in a mated station, it is preferable to conduct the operations on both sides simultaneously. The balancing problem refers to finding the optimal allocation of tasks to the workstation while considering the restrictions applied to the line. Unlike other balancing problems which aim at minimizing cycle times, the so-called equal piles for assembly lines problem (EPALP) equalizes the workstation's workload in order to achieve a certain number of stations.

Job-processing times are usually assumed as a constant factor. In practice, in some industries, workers can gain better skills over times to increase efficiency. As a result, various tasks are performed more quickly, and the production processing time is continuously reduced (Chutima and Naruemitwong, 2014). Learning effect (LE) was first studied by Wright (1936) and examined on balancing lines by Chakravarty (1988). Two approaches are considered for applying this effect to production lines: in the first case, learning is affected by the number of works processed by operators; in the second case, learning is affected by the sum of times processed by operators. In this study, we used the first approach. The learning effect formulation applied here was developed by Biskup (1999) as follows:

$$P_r = P \cdot r^{-a}, r = 1, 2, 3, \dots, n$$

Where P_r is the processing time of the r^{th} station on the cumulative production quantity r , p represents the processing time of the first item, and $a < 0$ denotes the constant learning index.

In this paper, considering the LE, we attempted to allocate the operators with different skill levels to the workstations in order to achieve three objectives:

- Reducing the operator's idle time;
- Reducing the cycle time;
- Reducing costs.

The rest of this paper is organized as follows: In section 2, the literature review is presented; The solution methodology is discussed in section 3; Section 4 introduces the problem along with the proposed assumptions; In section 5, the numerical results are discussed; and section 6 concludes the study and provides suggestions for future research.

2- Literature review

This section can be divided into two main sections; first papers related to 2S-ALP are reviewed then in the final part of this section application of hybrid ANN-DEA in the literature is discussed.

2-1- Two-Sided assembly line balancing problem

The literature of 2S-ALP can be classified into several categories; the first category consists of researches generally concentrated on minimizing cycle time. Kim et al. (2000) along with examining the features of an assembly line, with aim of minimizing cycle time developed a genetic algorithm to solve the two-sided assembly line balancing (2S-ALB) problem. Purnomo et al. (2013) formulated a mathematical model that considers the assignment constraint so as to minimize cycle time for a certain number of stations. They employed genetic algorithm and the iterative first-fit rule to solve the model and compared the performance of the two methods. Polat, Mutlu et al. (2018) studied 2S-ALB problem with aim of minimizing cycle time as well as physical workload of stations. Accordingly, they considered balancing problem type-II and utilized goal programming to solve the presented model. They concluded that considering workload not only led to a more balanced line but also decreased work-related diseases.

Duan et al. (2019) introduced task priority and presented artificial bee colony optimization with the aim of minimizing cycle time in 2S-ALP.

The second category of researches is dedicated to those attempted to minimize the number of stations and mated stations in 2S-ALP. Özcan and Toklu (2009a) developed a simulated annealing (SA) approach to minimize the number of mated stations (or the number of operators) so as to solve the proposed mathematical model. Two criteria were considered to measure the performance of the SA algorithm: maximizing weighted line efficiency while minimizing the weighted smoothness index.

Simaria and Vilarinho (2009) advanced a mathematical model for 2S-ALB problems and applied ant colony optimization to solve the model. Accordingly, they minimized the number of stations and balanced the workloads between and within the stations for a variety of models. Özcan and Toklu (2010) presented a mixed-integer program (MIP) and a heuristic approach to consider the setup time and minimize the number of mated stations (line length) and stations (operators) for specific cycle time. Özcan (2010) presented a chance-constraint, piecewise-linear, mixed-integer program that considers stochastic task time for a 2S-ALB problem. In this model, the main objective was to minimize the number of mated stations and stations, and simulated annealing was applied to solve the proposed algorithm. Özbakır and Tapkan (2011) utilized the bee colony algorithm to solve a 2S-ALB problem while considering a zoning constraint to minimize the number of stations with given cycle time. Chutima and Chimklai (2012) introduced particle swarm optimization with negative knowledge (PSONK) to solve multi-objective mixed-model 2S-ALB problems. They tried to minimize the number of mated stations and workstations. In addition, the authors attempted to simultaneously maximize work-relatedness while minimizing workload smoothness as two conflicting sub-targets. Tapkan, Ozbakir et al. (2012) presented a mathematical model to minimize the number of workstations in a 2S-ALB problem. They used the bee algorithm and artificial bee colony to solve the model. Tuncel and Aydin (2014) adopted the teaching-learning-based optimization algorithm in order to minimize the number of workstations and ensure workload smoothness. Yuan et al. (2015) developed a honey bee mating optimization algorithm (HBMO) to minimize the number of mated stations and the total number of stations for given cycle time. They applied simulated annealing in HBMO algorithm to achieve a better balance between intensification and diversification during the search, with respect to the fact that tasks were affected by machine breakdowns, loss of motivation, environmental factors.

The literature has also explored the minimization of the number of mated stations and cycle time simultaneously. Kim et al. (2009) presented a mathematical model and developed a genetic algorithm to solve a 2S-ALB problem. Their goal was to minimize cycle time and the number of mated stations. Özcan and Toklu (2009b) formulated a mixed-integer programming for a 2S-ALB problem. They applied preemptive goal programming for precise goals and fuzzy goal programming for imprecise ones. They considered three objectives: minimizing the number of mated stations, minimizing cycle time, and minimizing the number of tasks assigned to each station while considering the zoning constraint.

The next category of researches conducted in the field of 2S-ALP is focused on introducing new constraints for this problem. Toksarı et al. (2008) explored LE in U-type assembly lines in order to assign tasks in given cycle time. Furthermore, Toksarı et al. (2010) studied the minimization of the number of stations while considering LE in a simple assembly line balancing model. LE was investigated in a single-model assembly line balancing model by Hamta et al. (2013) to minimize cycle time. Khorasanian et al. (2013) applied simulated annealing to solve a 2S-ALB problem. They proposed an index that calculated the relationship between every two tasks. They also suggested a performance criterion, namely assembly line tasks consistency, to determine the average relationship among tasks designated to the stations of each solution. To assess the performance of the presented algorithm, they considered three criteria: the number of mated stations, the number of stations, and assembly line tasks consistency. Hu and Wu (2018) aimed at smoothing workload in the 2S-ALB problem. They introduced a new index to measure workload in a two-sided assembly line and developed a heuristic algorithm. Their results supported the effectiveness of their approach in distributing the idle time and workload among the workstations.

Kucukkoc et al. (2018) presented a mixed-integer linear model for the 2S-ALB problem, which takes account of underground stations in addition to the left and right stations placed on both sides of the

assembly line. They developed an ant colony optimization algorithm to solve real-life problems and investigated its performance on 78 problems. Delice et al. (2018) introduced sequence-dependent-setup time in U-type 2S-ALP and developed the ant colony optimization algorithm to solve the proposed problem.

Researches proposing a new solution methodology for the 2S-ALP formed the other category. Lee et al. (2001) presented an assignment procedure to maximize work-relatedness and work slackness in a 2S-ALB problem. Hu et al. (2008) presented a station oriented enumerative algorithm to deal with task assignment in a 2S-ALB problem. Aiming at minimizing the length of assembly lines, Xiaofeng et al. (2010) developed a new branch and bound algorithm to solve a 2S-ALB problem. In this study, the two stations facing each other were regarded as a single position and a one-sided assembly line. The authors calculated a number of lower bounds on the positions and developed dominance and reduction rules on the one-sided assembly line. Taha et al. (2011) developed a genetic algorithm to assign tasks to mated stations in a 2S-ALB problem. They utilized a hybrid crossover and modified scramble mutation as a new method for creating an initial population. Chutima and Naruemitwong (2014) applied Pareto biogeography-based optimization (BBO) to examine a two-sided assembly line sequencing problem that takes the learning effect into account. The purpose of this study was to minimize production volume variance, total utility, and total sequence-dependent setup time. Purnomo and Wee (2014) developed a bi-objective mathematical model in an attempt to maximize the production rate and distribute the workload in a two sided-assembly line. They addressed the zoning constraint in their model and applied a harmony search to solve the model. The performance of the presented algorithm was compared with NSGA-II, and it was demonstrated that the presented algorithms yielded a better convergence for small- and medium-sized problems and also provided a better solution for large-sized ones. Exact solutions are also proposed for 2S-ALP by (Yadav et al., 2019, Yadav and Agrawal, 2019).

2-2- Hybrid ANN-DEA

The combination of DEA and neural networks can be used to predict problem variables and rank them in various contexts. For example, Bashiri et al. (2013) developed a neuro-data envelopment analysis approach to optimize a multi-response optimization problem based on the Taguchi method. This approach was suggested for processes in which controllable factors are the smaller-the-better (STB)-type variables and the purpose is to reach an optimal solution with a smaller number of controllable factors. Rabbani et al. (2018) presented a simulation optimization approach for allocation of resources in an emergency department which considers laboratories, radiology departments, and pharmacies. To optimize the system, they proposed a new approach that makes use of discrete simulation, design of experiments (DOE), DEA, multi-layer perceptron ANN, and radial basis function. Yazdanparast et al. (2018) adopted an intelligent algorithm to optimize an emergency unit by emphasizing human error. The algorithm consisted of simulation, ANN, DOE, and DEA. They used simulation, ANN, and DEA to evaluate several scenarios, predict response variables, and identify the optimal scenario, respectively. In another research, Nasiri et al. (2017) employed a similar approach to assess several dispatching rules in an open shop scheduling problem. Similarly, Yazdanparast et al. (2016) applied ANN-DEA to obtain optimize operator allocation settings in a simulation of the multi-stage injection process. In another research, Yazdanparast et al. (2017) employed an intelligent algorithm composed of ANN, DOE, and DEA for the operator allocation and the job dispatching rule in a cellular manufacturing system by considering the DMSs of the operators.

Meanwhile, LE and assigning multi-skilled operators in balancing two-sided assembly lines have received little attention so far. Therefore, the present study proposes an integrated approach consisting of simulation, ANN, and the Taguchi method to solve the problem in order to minimize both idle time and cycle time.

3- Solution methodology

Modeling two-sided assembly lines with stochastic task time incorporating LE and multi-skilled operators' allocation can lead to a complex and nonlinear model. Therefore, it requires methods with high

flexibility. Since numerous scenarios must be simulated, 2^k factorial design of the experiment was used to reduce the number of these scenarios. Consequently, a DOE-discrete event simulation-DEA was proposed.

Simulation network is launched for these scenarios, and the following measurements are recorded:

- Cycle time
- Operators' idle time
- Cost of operator allocation

The flowchart of the research methods is shown in figure 1.

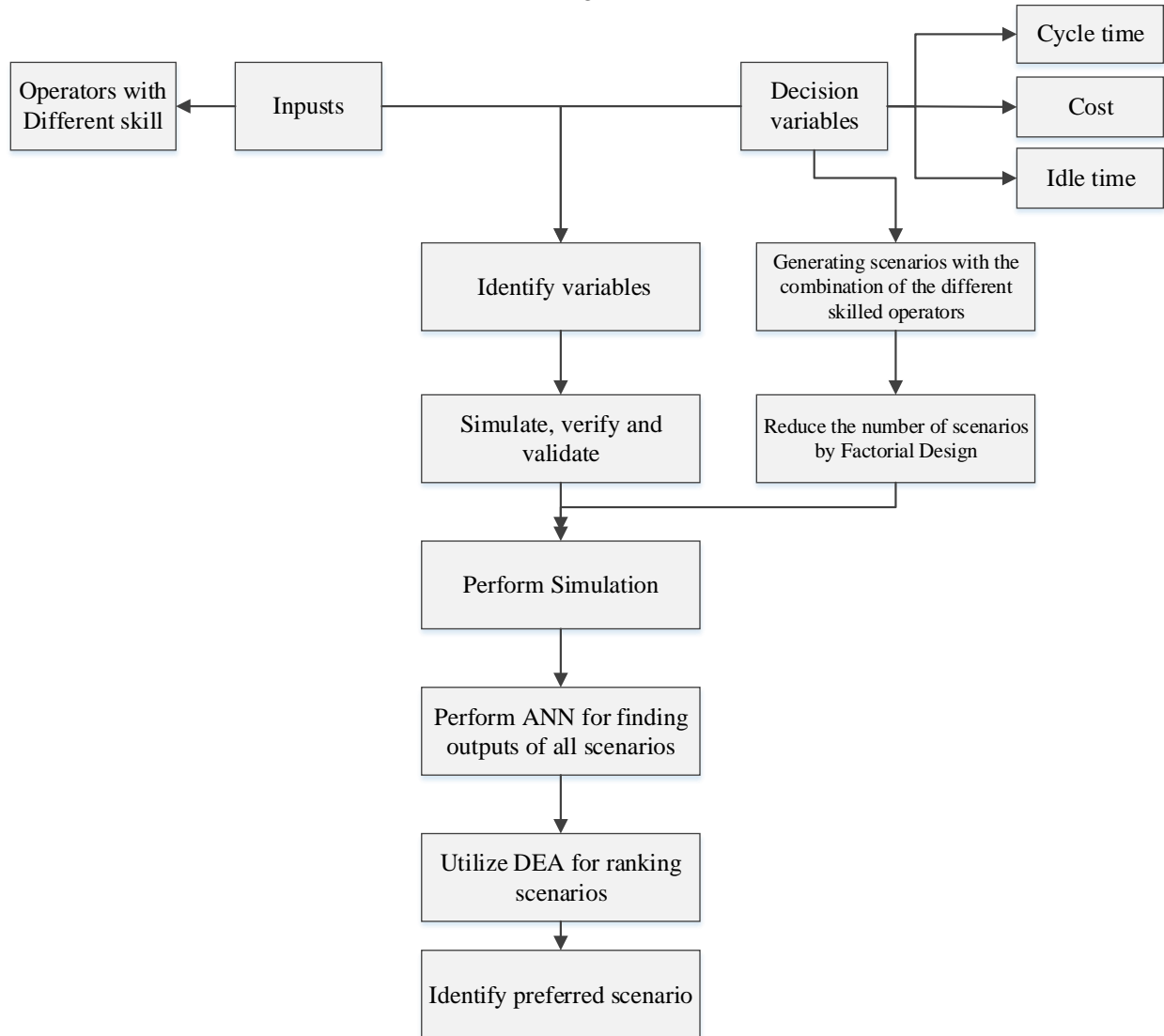


Fig 1. Flowchart of the proposed approach

3-1- Discrete event simulation

Systems that collect data under the proposed conditions may cause several changes in actual environments. Thus, simulation can be used to assess the behavior of the system in which the changes are applied to the simulation model instead of an actual environment. This is performed because the actual time and cost perspective and any changes in the actual system are not economically justifiable.

In fact, the simulation is regarded as the art and science of building a model or modeling of a process or system to test and evaluate the strategies, as well as a method to find out the results of the proposed ideas

before implementation. In manufacturing processes, simulation is frequently employed. In the real world, any variation in the configuration of a/an production/assembly line is costly. Thus, using a simulation model with DOE seems imperative.

The validation process is the most important step in designing the model (Hamid et al., 2018b, Davoudkhani et al., 2019). It determines whether the simulation model represents the actual system.

In this study, Arena was used for simulation. Arena simulation is a practical software program used in discrete event systems. It is a complete software for simulation studies and supports all the steps of the simulation process. The probability distribution of task times is obtained using the Input Analyser tool of Arena 14. Arena also represents the simulation logic by passing the entities through the model in the form of animation.

3-2- Factorial design

By searching for the behavior of response and factors, the design of experiments tries to obtain the shortest and least costly way to reach the target. Factorial design includes all the combinations of various levels considered for each factor (Nasiri et al., 2017). In a factorial design, if the number of factors is “n” and the number of levels is “m”, the total number of experiments required will be indicated by m^n . The two-level factorial design is one of the most important designs in the factorial design of the experiment. In 2^k factorial design, each factor has two levels: upper and lower. The 2^k full factorial design and 2^k fractional factorial design are two types of 2^k full factorial design of experiments. The former defines all the possible combinations of factors and levels to determine the optimum setting of inputs to obtain outputs. The latter that was used in this study, selects a fraction of all the possible combinations ($\frac{1}{2}, \frac{1}{4}, \dots$) and implements the experiment in order to save time and reduce costs through fewer runs of experiments (Nasiri et al., 2017).

In this study, two levels of skill were considered for each operator and an integrated approach based on DOE-DES-DEA was presented to optimize the performance of the two-sided multi-model assembly line.

3-3- Artificial neural network

Artificial neural networks were inspired by biological neural networks. The main component of biological neural networks is the neurons and the neuronal component which includes the main body, dendrites, synapses, and axons. In these networks, the inputs are imported from dendrites, pass through the main body, and exit through the axon. The information signal goes to the second part dendrites by the relationship between axons. The relationship between two dendrites is called a synapse. Biological neural networks have a memory in communication between neurons, known as synaptic weights (Zhalechian et al., 2017). The artificial neural network is widely used in the literature in different contexts (for example see Kharola and Patil, 2017, Hussein et al., 2017, El-said, 2013, Abdel et al., 2016).

General artificial neural networks include an input layer, a hidden layer, and an output layer (Habibifaar et al., 2019, Gharoun et al., 2019). Neurons in network layers are connected by synaptic weights. The network learning algorithm updates these weights for discovering and modeling the relationship between inputs and outputs. The total weighted inputs are processed and applied to the activation function which produces the output. By adjusting the weights and constant values (bias), this algorithm tries to minimize the difference between generated output and actual output.

The multilayer perceptron (MLP) neural network is the most popular and most widely used neural network model. In MLP, every neuron of each layer is connected to all the neurons of adjacent layers. MLP has three parts: the input layer, the hidden layer(s), and the output layer. The number of neurons in the input layer equals to a number of independent variables, while the number of neurons in the output layer equals the number of dependent variables. Training function, number of hidden layers, transfer function, number of neurons in each hidden layer, and output layer transfer function are cases which must be defined to determine the structure of MLP (Nasiri et al., 2017, Yazdanparast et al., 2018).

In this study, we employed a MLP network with one hidden layer, the Levenberg-Marquardt function as the training function, a tangent sigmoid function as the transfer function from the input layer to the hidden layer, and a linear function as the transfer function from the hidden layer to the output layer.

3-4- Data Envelopment Analysis (DEA)

Several approaches are employed in the literature to evaluate alternatives, such as TOPSIS (e.g., see Jamili et al., 2018, Amalnick et al., 2019), PROMETHEE-II (e.g., see Hamid et al., 2019, Bastan et al., 2019), DEA (e.g., see Azadeh et al., 2016, Babajani et al., 2019, Gharoun et al., 2018, Habibifar et al., 2019, Hamid et al., 2018c, Yazdanparast et al., 2018, Hamid et al., 2018a, Hamid et al., 2017, Mirzamohammadi and Hamid, 2019). One of the most important and powerful decision-making methods is DEA that consists of many inputs and outputs. DEA is based on a series of optimization methods utilizing linear programming, also known as a nonparametric method (Azadeh et al., 2016). In this method, an efficient border curve is made by a series of points determined by linear programming. After a series of optimizations, linear programming determines whether the decision-making unit is located on an efficient border curve or not. Then, efficient and inefficient units are separated. There are several types in the DEA model: the input-oriented model looks for a portion in which the inputs are reduced while the outputs remain unchanged. However, the output-oriented model looks for a portion in which the outputs are increased without any changes in the inputs. In both cases, the unit remains on the efficient border curve. In some cases, management has no control over the outputs, and the value of the outputs is fixed in advance. On the contrary, in some cases, the amount of inputs is fixed, and the value of the outputs is a decision variable. In the above circumstances, input- and output-oriented models are respectively the appropriate ones.

The capability of the DEA is to utilize output/input-oriented models with different returns to the scale pattern. The return to scale represents the link between the changes in the inputs and outputs of the system. The constant return to scale means that any input coefficient produces the same coefficient of outputs, while the variable return to scale means that any coefficient of inputs can produce equally, less, or more coefficient outputs. The basic model of DEA, which is also called CCR was developed by (Charnes et al., 1978). The CCR model is a constant return to scale in which input-and output-oriented models are the same. The following notations will be used for describing the linear programming of CCR:

Sets:

S Set of outputs

M Set of inputs

n Set of DMUs

index:

o Index of DMU under investigation

j Index of DMUs

r Index of outputs

i Index of inputs

Parameters:

u_r Weight of output r

v_i Weight of input i

y_{ro} Value of output r of unit o (DMU under investigation)

x_{io} Value of input i of unit o (DMU under investigation)

y_{rj} Value of output r of unit j

x_{ij} Value of input i of unit j

(1)

$$\text{Max } z_o = \sum_{r=1}^s u_r \cdot y_{ro}$$

Subject to:

$$\sum_{i=1}^m v_i \cdot x_{io} = 1 \quad (2)$$

$$\sum_{r=1}^s u_r \cdot y_{rj} - \sum_{i=1}^m v_i \cdot x_{ij} \leq 0 \quad j=1, \dots, n \quad (3)$$

$$u_r \geq 0, \quad r=1, \dots, s \quad (4)$$

$$v_i \geq 0, \quad i=1, \dots, m \quad (5)$$

Equation 1 attempts to maximize the weighted outputs, while the weighted sum of inputs equals 1 due to Constraint 2. Equation 3 ensures that each DMU should be located on or inside the efficient border.

The BCC model developed by Banker et al. (1984) is a variable return to scale model and, contrary to CCR, input- and output-oriented BCC models are different. The linear programming of BCC is presented below, by adding an unconstrained variable u_0 in comparison with CCR. This variable ensures that the efficient border should have some convexity linear combinations of the best practice and contain increasing and decreasing returns to scale areas.

$$\text{max } z_o = \sum_{r=1}^s u_r \cdot y_{ro} - u_0 \quad (6)$$

Subject to:

$$\sum_{i=1}^m v_i \cdot x_{io} = 1 \quad (7)$$

$$\sum_{r=1}^s u_r \cdot y_{rj} - \sum_{i=1}^m v_i \cdot x_{ij} - u_0 \leq 0 \quad j=1, \dots, n \quad (8)$$

$$v_i \geq 0, \quad i=1, \dots, m \quad (9)$$

$$u_r \geq 0, \quad r=1, \dots, s \quad (10)$$

$$u_0 \text{ free} \quad (11)$$

In this study, since we can control the inputs of the problem (i.e., assigning workers with different skill levels and different learning factors to the workstations in the production line), the input-oriented variable returns to the scale of BCC model was applied for selecting the best scenario.

3-5- Experimental model

In this paper, we attempted to develop a discrete event simulation model to examine the assembly line problem in a two-sided assembly line. In a real two-sided assembly line, there are certain constraints, e.g. the zoning constraint and stochastic task time. In this paper, the real-world configuration of the two-sided assembly line was examined, and the necessary data were collected. In our simulation method, firstly, the data needed for the simulation was acquired. The experiments are based on 10054 records from the assembly line. To collect the data, we designed a form and employed it to collect assembly line data for 2 months (from 1st March 2017 to 31st April 2017). Equal piles for Assembly Line Problem (EPALP) were considered, i.e., the objective was to equalize the stations' workload with respect to a certain number of stations. The other assumptions of the problem were defined as follows:

- Two models of one product are produced on a two-sided assembly line.
- Operators perform tasks in parallel on both sides of the line.
- Some tasks may be required to be performed on one side, while others may be performed on either side of the line.
- Precedence diagrams are known.
- Task times are stochastic.
- Each station has only one operator, and the change of operators is not allowed.
- LEs are considered.
- No work in progress is allowed.

3-6- Implementing the simulation model

The flowchart of the simulation model utilized in this paper is depicted in figure 2.

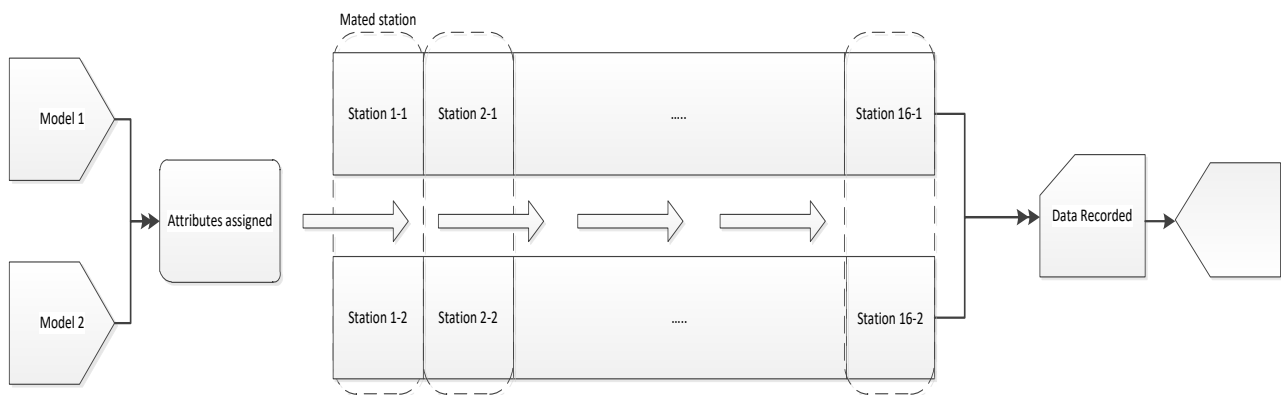


Fig 2. Flowchart of the simulation model

In the present study, two different products were considered as entities. One of the attributes assigned to each entity is the serial number starting from 1 which is used in calculating the processing time with respect to the LE. In the simulation model, each product model had its own process time at different stations. The duration of processing time for each model at different stations is presented in table 1.

Table 1. Process time at stations

Product 1	Process time $N(\mu, \sigma)$ second	Product 2	Process time $N(\mu, \sigma)$ second
Station 1-1	(52 , 2.5)	Station 1-1	(54 , 3.5)
Station 1-2	(54 , 3.5)	Station 1-2	(53 , 2.1)
Station 2-1	(57 , 3)	Station 2-1	(58 , 3.5)
Station 2-2	(59 , 4)	Station 2-2	(61 , 4)
Station 3-1	(64 , 3.5)	Station 3-1	(60 , 4)
Station 3-2	(62 , 4)	Station 3-2	(58 , 4)
Station 4-1	(60 , 3)	Station 4-1	(64 , 6)
Station 4-2	(64 , 4.5)	Station 4-2	(61 , 2.3)
Station 5-1	(67 , 4.2)	Station 5-1	(64 , 3)
Station 5-2	(65 , 5)	Station 5-2	(67 , 5)

Two different skill levels were considered for the operators: expert and semi-skilled. The processing time mentioned in table 3 was the operation time carried out by expert operators. To calculate process time for semi-skilled operators, we multiplied the values given in Table 3 by the coefficient 1.75. Also, based on historical data, the constant learning index for expert and semi-skilled operators equaled (-0.1) and (-0.05), respectively. Figure 3 illustrates the design of the simulation model by Arena software.

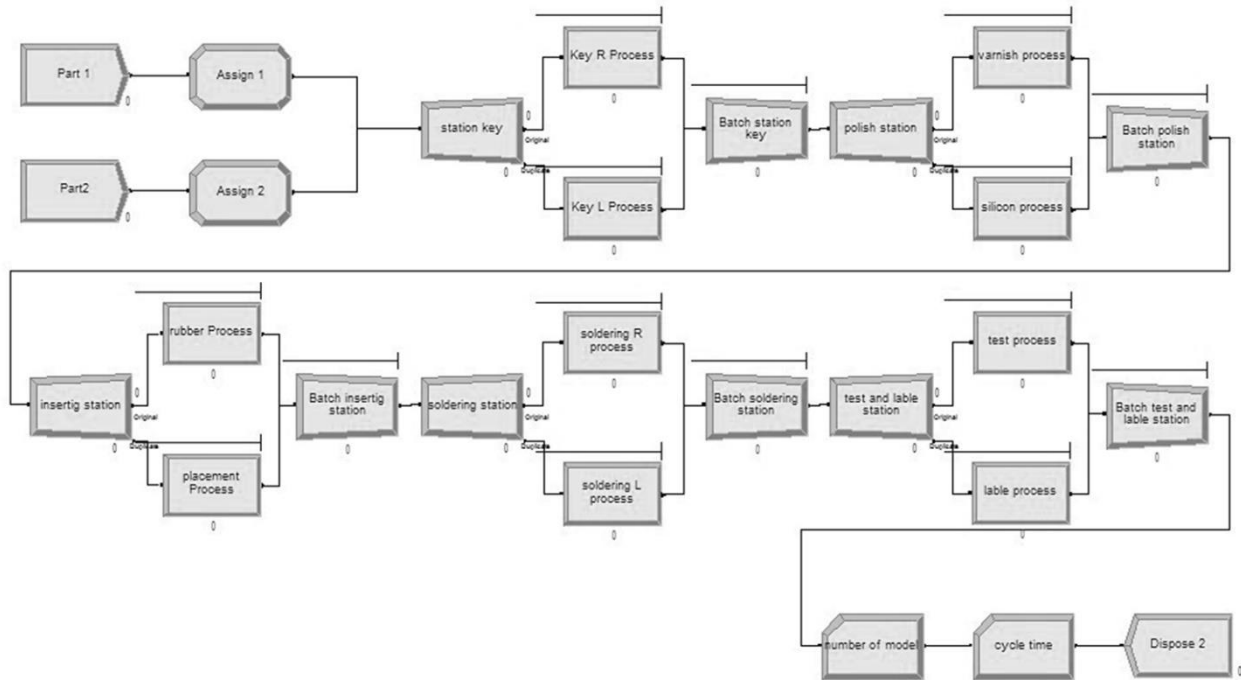


Figure 3: Arena simulation model

In Arena software, modeling components were employed to build the simulation model. To simulate the arrival of entities to the assembly line and assign attributes to them, create and assign modules have been applied.

In the two-sided assembly line, the simulation logic must be defined such that each entity should not be allowed to be sent to the next station after entering a mated station, while each side of the assembly line performs its tasks. Thus, to simulate the model, we applied separate, batch, and process modules. The separate module was used to duplicate the entity. Any entity entering a separate module was duplicated to two entities with the same specification and sent to two process modules representing every two facing stations in the mated station. The operator allocation and task timing for each station are determined in the process module. Afterwards, the entities were sent to the batch module. The application of the batch module was to pack the entities into a single one. Therefore, both entities arisen by one entity through the separate module were joined by the batch module. Entities could be joined if they had identical serial numbers. As already mentioned, the entities duplicated by the separate module had the same specifications, including serial numbers. Any entity which entered the batch module would wait for its pair entity until it entered the same batch module. Then, they would be sent to the next station. The interference constraint was simulated accordingly. Finally, necessary statistical data would be collected by the record module, and the entities would leave the system by the dispose module. Any information about the operators, including the wage rates, was defined in the resource module. The wage rate for the expert and semi-skilled operators were respectively considered as 20 and 12 from the view of the monetary unit. Parameters of the model had to be defined in each run of the simulation model, including process time at each station with respect to entity type, learning effect, and wage rate.

One of the requirements in the simulation is setting an appropriate value for starting and stopping the simulation model. In this case, simulation run time was set to produce 1000 entities equal to 2500 hours with 180 hours as the warm-up period. The number of replications was set at 4. The system initialized when an entity entered the model.

3-7- Model verification and validation

The evaluation of the model behaviour for its proper operation is called validation. The validation of the models is critically important. Thus, in this study, the animation tool of the software was used for validation. The animation model was run several times and examined by the experts who confirmed the accuracy of the simulation model.

Validation demonstrates that a computerized model in its application scope has a satisfactory range of accuracy complied with the intended use for that model (Sargent, 2005, Yatimi and Aroudam, 2018). The most important state variable in this system is the operators with different skill levels. This parameter can simply be considered as an actual value. As a result, if all model parameters are set equal to their actual values, the cycle time will be the most important criterion for assessing model validity. For this purpose, the actual data about the cycle time were compared with cycle time results obtained through the simulation using paired-samples t-test. In this case, a t-test was used for each model at the $p < 0.05$ significance level to test the following hypotheses.

$$1) \begin{cases} H_0 : \mu_{simulation} = \mu_{real} \\ H_1 : \mu_{simulation} \neq \mu_{real} \end{cases}$$

In the case of Model 1, the calculated confidence interval for this test was [-0.641, 0.619], the t-statistic equaled (-0.03), and the p-value was 0.972. Furthermore, for Model 2, the calculated confidence interval for this test was [-0.641, 0.619], the t-statistic equaled (-0.05), and the p-value was 0.958. According to the results, there is no statistically significant difference between the means of the two values (real data and simulation result). Thus, the hypothesis is not rejected.

4- Numerical results

Regarding the number of factors and considering 2 levels for each factor, the total number of possible scenarios for implementing simulation will equal 1024. A 2^k fractional factorial design was employed based on the number of possible scenarios. Then, 128 experiments (i.e. $\frac{1}{8}$ of total experiments) were selected by factorial design in Minitab. Using the selected scenarios, the simulated two-sided assembly line was implemented for both modes of considering and ignoring LE. The results of simulation by considering LE and simulation without considering LE are shown in Appendices 1 and 2, respectively. A close look at the results reveals that, as expected, cycle times and operators' idle times decrease if LE is taken into account.

In the next step, to predict the value of other possible scenarios for which simulation was not implemented, we used ANN to determine the impact of considering and ignoring LE in balancing the two-sided assembly line for all possible scenarios.

After the simulation outputs were collected (appendices 1 and 2), the multilayer perceptron (MLP) artificial neural network was applied. Inputs and outputs of the neural network were the defined scenarios and the 4 defined criteria of simulation outputs, respectively. Since the inputs (scenarios) were nominal variables, pre-processing was applied and the nominal variables were converted to numerical ones. Each variable was divided into two variables (tables 2 and 3).

Table 2. Inputs of the network before pre-processing

Variable	Example Scenario
Variable 1 : operator Type of Station 1	Semi-skilled
Variable 2 : operator Type of Station 2	Semi-skilled
Variable 3: operator Type of Station 3	Expert
Variable 4 : operator Type of Station 4	Semi-skilled
Variable 5: operator Type of Station 5	Expert
Variable 6: operator Type of Station 6	Expert
Variable 7: operator Type of Station 7	Expert
Variable 8 : operator Type of Station 8	Semi-skilled
Variable 9: operator Type of Station 9	Expert
Variable 10 : operator Type of Station 10	Semi-skilled

Table 3. Inputs of the network after pre-processing

Variable	Example Scenario
Variable 1: operator Type of Station 1: Semi-skilled	1
Variable 2: operator Type of Station 1: Expert	0
Variable 3: operator Type of Station 2: Semi-skilled	1
Variable 4: operator Type of Station 2: Expert	0
Variable 5: operator Type of Station 3: Semi-skilled	0
Variable 6: operator Type of Station 3: Expert	1
Variable 7: operator Type of Station 4: Semi-skilled	1
Variable 8: operator Type of Station 4: Expert	0
Variable 9: operator Type of Station 5: Semi-skilled	0
Variable 10: operator Type of Station 5: Expert	1
Variable 11: operator Type of Station 6: Semi-skilled	0
Variable 12: operator Type of Station 6: Expert	1
Variable 13: operator Type of Station 7: Semi-skilled	0
Variable 14: operator Type of Station 7: Expert	1
Variable 15: operator Type of Station 8: Semi-skilled	1
Variable 16: operator Type of Station 8: Expert	0
Variable 17: operator Type of Station 9: Semi-skilled	0

To implement MLP, 75%, 15%, and 15% of the data were respectively selected for training, validating, and testing the network. The decision on the number of neurons in the hidden layer was conducted based on the mean squared error (MSE). In this way, the number of neurons was changed in each network execution. The number of neurons to which the MSE was minimized was selected as the optimal number of neurons in the hidden layer. Tables 4 depicts the minimum MSE in this study for 10 neurons in the single hidden layer for both modes of considering and ignoring the LE.

Table 4. ANN-MLP obtained results

	Performance	Total cost	Cycle time 1	Cycle time 2	idle time
Ignoring LE	MSE	0.003117434	0.002786378	0.001757459	0.000122346
	R	0.949126875	0.976518769	0.981895864	0.997976942
Considering LE	MSE	0.003676412	0.002721414	0.001515365	0.00012449
	R	0.937666572	0.976615526	0.987079883	0.997960145

To verify the accuracy of the network, 60%, 40%, and 20% of data were selected for testing the network, and the sensitivity of the above test dataset was measured. In so doing, at each stage, the number of training data was increased compared to test data. In this case, we expected to reduce errors because the network would receive more information by increasing the volume of training data and reducing the test data. The sensitivity analysis result is presented in table 5.

Table 5. ANN-MLP obtained sensitivity result

		MSE			
	% of Test Data	<i>Cycle time model 1</i>	<i>Cycle time model 2</i>	<i>idle time</i>	<i>Total cost</i>
Considering LE	60%	0.013174	0.00818	0.027351	0.000407
	40%	0.002842	0.003822	0.007829	0.000694
	20%	0.001926	0.002097	0.001554	0.000149
Ignoring LE	60%	0.00767	0.006711	0.019688	0.000275
	40%	0.002156	0.002294	0.005355	0.000152
	20%	0.001889	0.001947	0.001473	0.000113

After training ANN, the trained network’s estimation for other scenarios was selected to find the best operator allocation scenario with minimum cost, minimum operators’ idle times, and minimum cycle time. All scenarios were produced by Minitab software. After using the trained network for all the scenarios and collecting their results, the variable return to scale input-oriented DEA model was applied to select the best scenario. For this purpose, the cost, cycle times for both models as well as the operators’ idle times are regarded as inputs. Also, we create another virtual indicator which the values of all arrays are “1” in all considered scenarios. This virtual indicator is used as output in the DEA model. All scenarios for both modes of considering and ignoring the LE at one time were entered in the DEA model. The total number of DMUs equaled 2048; the DMUs numbers 1-1024 were related to scenarios ignoring LE, and DMU numbers 1025-2048 were related to scenarios considering LE. Because the outputs obtained from the simulation were different in range, each data set was normalized. The efficiency of each DMU was calculated by the Auto-Assess software. The average efficiency equaled 0.513401, and 4 DMUs were selected and nominated as strongly efficient. The nominated DMUs expressed considering LE in the 2S-ALB problem help achieve higher performance. The nominated DMUs are shown in table 6.

Table 6. Selected scenarios

DMU	Scenario										Efficiency	Total Cost	Cycle time model 1	Cycle time model 2	Operators idle time
	Station1	Station2	Station3	Station4	Station5	Station6	Station7	Station8	Station9	Station10					
1403	2	1	2	2	2	2	2	1	1	2	1	373,717.47	76.50	79.75	96,774.15
1482	1	1	1	2	2	1	2	1	1	1	1	373,788.15	80.27	75.71	80,302.35
1703	1	2	1	2	2	1	1	1	2	2	1	373,987.12	72.91	70.17	68,508.32
1937	2	1	2	2	1	2	2	1	2	2	1	364,497.05	78.49	80.58	69,052.43

1= Operator Type:Expert ; 2= Operator Type:Semi skilled

5- Conclusion

The two-sided assembly line is a type of line on which both sides are used and operational tasks are performed on the left and right sides. In this study, the two-sided assembly line was first simulated considering the issue of LE. We proposed the DOE-discrete event simulation and ANN-DEA approach for optimizing the two-sided assembly line by adding LE as a major assumption. An actual case of the two-sided assembly line was simulated by the Arena 13.5 software. We looked for the combination of different operator skills to minimize the cost of operator allocation, cycle time, and operators' idle time. All the scenarios were defined based on different levels of operators. Then, by applying 2^k factorial design, 1/8 of all the scenarios were selected. The issues of cost, cycle time, and operators' idle time were defined as decision variables. The outputs of all scenarios were obtained by applying the ANN. Finally, for all scenarios (or DMUs), the efficiency was calculated using the DEA model, and the best scenario was selected.

Two-sided assembly lines are widely used in large-size production such as automobiles and trucks. According to its wide application in varied industries, this type of production line configuration has been attracted the attention of many researchers as reviewed in Section 2. The integrated approach of simulation-ANN-DEA could be applied in problems where there are a set of scenarios to implement on a system and problem deals with choosing the most preferable scenario. While there is large number of scenarios, it is not conceivable to run simulated model for all scenarios. Accordingly, a practical number of scenarios would have selected by utilization of DOE methods and results of simulated model collected. Since, in order to select the most preferable scenarios, it is needed to compare all scenarios, ANN could utilize to estimate the results of simulated model for those scenarios which the model did not run. Eventually, with having results of all scenarios, the most preferable scenario could be selected by applying varied methods such as MADM techniques. DEA is one of well-known method which could be utilized efficiently to determine the most preferable. For future research, we can consider the sequencing problem and fatigue effect.

References

- Abdel, M. S. E.-D. A., El Samahy, M., Hassan, M. A. M. & El Bendary, F. 2016. Applications of ANFIS in Loss of excitation faults detection in hydro-generators. *International Journal of System Dynamics Applications (IJSDA)*, 5, 63-79.
- Amalnick, M. S., Habibifar, N., Hamid, M. & Bastan, M. 2019. An intelligent algorithm for final product demand forecasting in pharmaceutical units. *International Journal of System Assurance Engineering and Management*, 1-13.
- Azadeh, A., Zarrin, M. & Hamid, M. 2016. A novel framework for improvement of road accidents considering decision-making styles of drivers in a large metropolitan area. *Accident Analysis & Prevention*, 87, 17-33.
- Babajani, R., Abbasi, M., Taher Azar, A., Bastan, M., Yazdanparast, R. & Hamid, M. 2019. Integrated safety and economic factors in a sand mine industry: a multivariate algorithm. *International Journal of Computer Applications in Technology*, In press.
- Banker, R. D., Charnes, A. & Cooper, W. W. 1984. Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management science*, 30, 1078-1092.
- Bartholdi, J. 1993. Balancing two-sided assembly lines: a case study. *International Journal of Production Research*, 31, 2447-2461.
- Bashiri, M., Farshbaf-Geranmayeh, A. & Mogouie, H. 2013. A neuro-data envelopment analysis approach for optimization of uncorrelated multiple response problems with smaller the better type controllable factors. *Journal of Industrial Engineering International*, 9, 30.
- Bastan, M., Kiani, P., Korlou, S. & Mahdi, H. 2019. Selection of a Biomass Product using a Hybrid Approach of BW-PROMETHEE. *International Conference on Industrial Engineering and Operations Management*. Pilsen, Czech Republic.
- Biskup, D. 1999. Single-machine scheduling with learning considerations. *European Journal of Operational Research*, 115, 173-178.
- Chakravarty, A. K. 1988. Line Balancing With Task Learning Effects. *IIE Transactions*, 20, 186-193.
- Charnes, A., Cooper, W. W. & Rhodes, E. 1978. Measuring the efficiency of decision making units. *European Journal of Operational Research*, 2, 429-444.
- Chutima, P. & Chimklai, P. 2012. Multi-objective two-sided mixed-model assembly line balancing using particle swarm optimisation with negative knowledge. *Computers & Industrial Engineering*, 62, 39-55.
- Chutima, P. & Naruemitwong, W. 2014. A Pareto biogeography-based optimisation for multi-objective two-sided assembly line sequencing problems with a learning effect. *Computers & Industrial Engineering*, 69, 89-104.
- Davoudkhani, M., Hamid, M., Tavakkoli-Moghaddam, R., Bastan, M., Nasiri, M. M. & Shakouri Ganjavi, H. 2019. A Two-Step Stochastic Optimization and Simulation Approach for Scheduling Operating Rooms in an Ophthalmology Surgery Department. *the International Conference on Industrial Engineering and Operations Management*. Pilsen, Czech Republic.

- Delice, Y., AYDOĞAN, E. K., SÖYLEMEZ, İ. & ÖZCAN, U. 2018. An ant colony optimisation algorithm for balancing two-sided U-type assembly lines with sequence-dependent set-up times. *Sādhanā*, 43, 199.
- Duan, X., Wu, B., Hu, Y., Liu, J. & Xiong, J. 2019. An improved artificial bee colony algorithm with MaxTF heuristic rule for two-sided assembly line balancing problem. *Frontiers of Mechanical Engineering*, 14, 241-253.
- El-said, S. A. 2013. Reliable Face Recognition Using Artificial Neural Network. *International Journal of System Dynamics Applications (IJSDA)*, 2, 14-42.
- Gharoun, H., Hamid, M., Iranmanesh, H. & Yazdanparast, R. 2018. Performance Optimization of Two-Sided Assembly Line Balancing Problem Considering Learning Effect and Multi-Skilled Operators' Allocation. *the 2ND Gelisim-UWE International Conference on Economics, Finance , Istanbul, Turkey*.
- Gharoun, H., Keramati, A., Nasiri, M. M. & Azadeh, A. 2019. An integrated approach for aircraft turbofan engine fault detection based on data mining techniques. *Expert Systems*, 36, e12370.
- Habibifaar, N., Hamid, M. & Nasiri, M. M. 2019. Concurrent Optimization of Integrated Macro-Ergonomics and Resilience Engineering in a Pharmaceutical Manufacturer. *Journal of Industrial and Systems Engineering*, In press.
- Habibifar, N., Hamid, M., Bastan, M. & Taher Azar, A. 2019. Performance Optimization of a Pharmaceutical Production Line by Integrated Simulation and Data Envelopment Analysis. *International Journal of Simulation and Process Modelling*, In press.
- Hamid, M., Barzinpour, F., Hamid, M. & Mirzamohammadi, S. 2018a. A multi-objective mathematical model for nurse scheduling problem with hybrid DEA and augmented ϵ -constraint method: a case study. *Journal of Industrial and Systems Engineering*, 11, 98-108.
- Hamid, M., Hamid, M., Musavi, M. & Azadeh, A. 2018b. Scheduling elective patients based on sequence-dependent setup times in an open-heart surgical department using an optimization and simulation approach. *Simulation: Transactions of the Society for Modeling and Simulation International*, 1-24.
- Hamid, M., Hamid, M., Nasiri, M. M. & Ebrahimnia, M. 2017. Improving surgical theater performance using an integrated and multi-objective mathematical model and data envelopment analysis: A case study. *3th International Conference on Industrial Engineering, Babolsar, Iran*.
- Hamid, M., Hamid, M., Nasiri, M. M. & Ebrahimnia, M. 2018c. Improvement of operating room performance using a multi-objective mathematical model and data envelopment analysis: A case study. *International Journal of Industrial Engineering & Production Research*, 29, 117-132.
- Hamid, M., Nasiri, M. M., Werner, F., Sheikahmadi, F. & Zhalechian, M. 2019. Operating room scheduling by considering the decision-making styles of surgical team members: a comprehensive approach. *Computers & Operations Research*, 108, 166-181.
- Hamta, N., Fatemi Ghomi, S. M. T., Jolai, F. & Akbarpour Shirazi, M. 2013. A hybrid PSO algorithm for a multi-objective assembly line balancing problem with flexible operation times, sequence-dependent setup times and learning effect. *International Journal of Production Economics*, 141, 99-111.

- Hu, X., Wu, E. & Jin, Y. 2008. A station-oriented enumerative algorithm for two-sided assembly line balancing. *European Journal of Operational Research*, 186, 435-440.
- Hussein, H. T., Ammar, M. & Hassan, M. M. 2017. Three Phase Induction Motor's Stator Turns Fault Analysis Based on Artificial Intelligence. *International Journal of System Dynamics Applications (IJSDA)*, 6, 1-19.
- Jamili, A., Hamid, M., Gharoun, H. & khoshnoudi, R. Developing a Comprehensive and Multi-Objective Mathematical Model for University Course Timetabling Problem: A Real Case Study. Conference: Proceedings of the International Conference on Industrial Engineering and Operations Management, Paris, France, 2018. 2108,2119.
- Kharola, A. & Patil, P. P. 2017. Position and Tilt Control of Two-Wheeled Robot (TWR): A Neuro-Fuzzy Approach. *International Journal of System Dynamics Applications (IJSDA)*, 6, 17-33.
- Khorasanian, D., Hejazi, S. R. & Moslehi, G. 2013. Two-sided assembly line balancing considering the relationships between tasks. *Computers & Industrial Engineering*, 66, 1096-1105.
- Kim, Y. K., Kim, Y. & Kim, Y. J. 2000. Two-sided assembly line balancing: a genetic algorithm approach. *Production Planning & Control*, 11, 44-53.
- Kim, Y. K., Song, W. S. & Kim, J. H. 2009. A mathematical model and a genetic algorithm for two-sided assembly line balancing. *Computers & Operations Research*, 36, 853-865.
- Kucukkoc, I., Li, Z., Karaoglan, A. D. & Zhang, D. Z. 2018. Balancing of mixed-model two-sided assembly lines with underground workstations: A mathematical model and ant colony optimization algorithm. *International Journal of Production Economics*.
- Lee, T. O., Kim, Y. & Kim, Y. K. 2001. Two-sided assembly line balancing to maximize work relatedness and slackness. *Computers & Industrial Engineering*, 40, 273-292.
- Mirzamohammadi, S. & Hamid, M. 2019. Optimization of customers' trust in the insurance industry by data envelopment analysis: An actual case study. *Proceedings of the International Conference on Industrial Engineering and Operations Management Pilsen, Czech Republic*.
- Nasiri, M. M., Yazdanparast, R. & Jolai, F. 2017. A simulation optimisation approach for real-time scheduling in an open shop environment using a composite dispatching rule. *International Journal of Computer Integrated Manufacturing*, 1-14.
- Özbakır, L. & Tapkan, P. 2011. Bee colony intelligence in zone constrained two-sided assembly line balancing problem. *Expert Systems with Applications*, 38, 11947-11957.
- Özcan, U. 2010. Balancing stochastic two-sided assembly lines: A chance-constrained, piecewise-linear, mixed integer program and a simulated annealing algorithm. *European Journal of Operational Research*, 205, 81-97.
- Özcan, U. & Toklu, B. 2009a. Balancing of mixed-model two-sided assembly lines. *Computers & Industrial Engineering*, 57, 217-227.
- Özcan, U. & Toklu, B. 2009b. Multiple-criteria decision-making in two-sided assembly line balancing: A goal programming and a fuzzy goal programming models. *Computers & Operations Research*, 36, 1955-1965.

- Özcan, U. & Toklu, B. 2010. Balancing two-sided assembly lines with sequence-dependent setup times. *International Journal of Production Research*, 48, 5363-5383.
- Purnomo, H. D. & Wee, H.-M. 2014. Maximizing production rate and workload balancing in a two-sided assembly line using harmony search. *Computers & Industrial Engineering*, 76, 222-230.
- Purnomo, H. D., Wee, H.-M. & Rau, H. 2013. Two-sided assembly lines balancing with assignment restrictions. *Mathematical and Computer Modelling*, 57, 189-199.
- Rabbani, M., Farshbaf-Geranmayeh, A. & Yazdanparast, R. 2018. A simulation optimization approach for integrated resource allocation in an emergency department, pharmacy, and lab. *Intelligent Decision Technologies*, 12, 187-212.
- Sargent, R. G. Verification and validation of simulation models. Proceedings of the 37th conference on Winter simulation, 2005. winter simulation conference, 130-143.
- Simaria, A. S. & Vilarinho, P. M. 2009. 2-ANTBAL: An ant colony optimisation algorithm for balancing two-sided assembly lines. *Computers & Industrial Engineering*, 56, 489-506.
- Taha, R. B., El-Kharbotly, A. K., Sadek, Y. M. & Afia, N. H. 2011. A Genetic Algorithm for solving two-sided assembly line balancing problems. *Ain Shams Engineering Journal*, 2, 227-240.
- Toksarı, M. D., İşleyen, S. K., Güner, E. & Baykoç, Ö. F. 2008. Simple and U-type assembly line balancing problems with a learning effect. *Applied Mathematical Modelling*, 32, 2954-2961.
- Toksarı, M. D., İşleyen, S. K., Güner, E. & Baykoç, Ö. F. 2010. Assembly line balancing problem with deterioration tasks and learning effect. *Expert Systems with Applications*, 37, 1223-1228.
- Wright, T. P. 1936. Factors affecting the cost of airplanes. *Journal of aeronautical sciences*, 3, 122-128.
- Xiaofeng, H., Erfei, W., Jinsong, B. & Ye, J. 2010. A branch-and-bound algorithm to minimize the line length of a two-sided assembly line. *European Journal of Operational Research*, 206, 703-707.
- Yadav, A. & Agrawal, S. Minimize Idle Time in Two Sided Assembly Line Balancing Using Exact Search Approach. Proceedings of the 2019 International Conference on Management Science and Industrial Engineering, 2019. ACM, 220-227.
- Yadav, A., Verma, P. & Agrawal, S. 2019. A Mathematical Formulation for Mixed Model Two Sided Assembly Line Balancing Problem to Consider Boundary Conditions. *Global Journal of Research In Engineering*.
- Yatimi, H. & Aroudam, E. 2018. Standalone Photovoltaic System with Maximum Power Point Tracking: Modeling and Simulation. *International Journal of System Dynamics Applications (IJSDA)*, 7, 94-111.
- Yazdanparast, R., Hamid, M., Azadeh, M. A. & Keramati, A. 2018. An Intelligent Algorithm for Optimization of Resource Allocation Problem by Considering Human Error in an Emergency Department. *Journal of Industrial and Systems Engineering*, 11, 0-0.
- Yazdanparast, R., Heydari, M. H. & Jolai, F. 2016. A multi response optimization approach toward optimizing operator allocation and machine dispatching rule in a multi stage injection process.

Yazdanparast, R. , Tavakkoli-Moghaddam, R. , Rezaie-Malek, M. , Zare-Akandeh, Z. , 2017. A simulation optimization approach for operator allocation and machines dispatching rule in a cellular manufacturing system with an operators' decision-making style. In: GASaM, Gen (Ed.), Cellular Manufacturing Systems: Recent Developments, Analysis and Case Studies. Nova Science Publishers, New York, United States, pp. 381–410 .

Yuan, B., Zhang, C., Shao, X. & Jiang, Z. 2015. An effective hybrid honey bee mating optimization algorithm for balancing mixed-model two-sided assembly lines. *Computers & Operations Research*, 53, 32-41.

Zhalechian, M., Tavakkoli-Moghaddam, R. & Rahimi, Y. 2017. A self-adaptive evolutionary algorithm for a fuzzy multi-objective hub location problem: An integration of responsiveness and social responsibility. *Engineering Applications of Artificial Intelligence*, 62, 1-16.