

Balancing of mixed-model two-sided assembly lines using teaching-learning based optimization algorithm

Öğretme-Öğrenme algoritmasını kullanarak iki yönlü karışık modelli montaj hattı dengeleme

Alper HAMZADAYI¹ 

¹Department of Industrial Engineering, Engineering Faculty, Van Yuzuncu Yil University, Van, Turkey.
alperhamzadayi@yyu.edu.tr

Received/Geliş Tarihi: 25.09.2017, Accepted/Kabul Tarihi: 15.11.2017

* Corresponding author/Yazışılan Yazar

doi: 10.5505/pajes.2017.14227

Research Article/Araştırma Makalesi

Abstract

The Teaching-Learning Based Optimization (TLBO) algorithm is a population-based optimization technique that has been shown to be competitive against other population-based algorithms. The main purpose of this paper is to solve the balancing problem of mixed-model two-sided assembly lines by using TLBO algorithm first time in the literature. Most recently, hybrid teaching-learning-based optimization (HTLBO) algorithm is proposed by [1] for solving the balancing of stochastic simple two-sided assembly line problem. The HTLBO algorithm is compared with the well-known 10 different meta-heuristic algorithms in the literature in [1]. The tests performed underlined that HTLBO algorithm presented more outstanding performance when compared to other algorithms. In this paper, HTLBO algorithm is also adapted for solving the problem of balancing mixed-model two-sided assembly line and its performance is analysed. The objective function of this study is to minimize the number of mated-stations and total number of stations for a predefined cycle time. A comprehensive computational study is conducted on a set of test problems that are taken from the literature and the performance of the algorithms are compared with existing approaches. Experimental results show that TLBO algorithm has a noticeable potential against to the best-known heuristic algorithms and HTLBO algorithm results show that it performs well as far as the best-known heuristic algorithms for the problem in the literature.

Keywords: Assembly line balancing, Teaching-learning based optimization algorithm, Hybrid teaching-learning based optimization algorithm, Two-sided assembly lines, Mixed-model assembly lines

Öz

Öğretme-Öğrenme-Tabanlı Eniyileme (ÖÖTE) algoritması, diğer popülasyon-tabanlı algoritmalar kadar etkin olduğu ortaya konmuş, popülasyon-tabanlı bir eniyileme algoritmasıdır. Bu makalenin temel amacı, ÖÖTE algoritmasını kullanarak iki yönlü karışık modelli montaj hattı dengeleme problemini ilk defa çözmektir. Yakın zamanda, stokastik iki yönlü tek modelli montaj hattı dengeleme problemini çözmek için [1]'de melez öğretme-öğrenme-tabanlı eniyileme (MÖÖTE) algoritması önerilmiştir. [1]'de MÖÖTE algoritması en iyi bilinen 10 farklı meta-sezgisel algoritma ile karşılaştırılmıştır. Yapılan testler MÖÖTE algoritmasının diğer algoritmalarla göre daha üstün bir performans sergilediğini ortaya koymuştur. Bu makalede ayrıca, MÖÖTE algoritması iki yönlü karışık modelli montaj hattı dengeleme problemini çözmek için adapte edilmiş ve algoritmanın performansı test edilmiştir. Bu çalışmanın amacı önceden tanımlanmış çevrim süresinde karşılıklı eşleşen istasyon sayısını ve toplam istasyon sayısını en aza indirmektir. Literatürden alınan test problem grupları üzerinden kapsamlı bir deneysel çalışma gerçekleştirilmiştir ve algoritmaların performansları var olan yaklaşımlarla karşılaştırılmıştır. Deneysel çalışmalar ÖÖTE algoritmasının karşılaştırılan diğer en iyi bilinen sezgisel algoritmalarla karşı göze çarpan bir potansiyele sahip olduğunu ve problemin çözümünde MÖÖTE algoritmasının bilinen en iyi sezgisel algoritmalar kadar iyi performans sergilediğini ortaya koymuştur.

Anahtar kelimeler: Montaj hattı dengeleme, Öğretme-öğrenme-tabanlı eniyileme algoritması, Melez öğretme-öğrenme-tabanlı eniyileme algoritması, İki yönlü montaj hatları, Karışık modelli montaj hatları

1 Introduction

An assembly line is a production process in which a number of tasks are assigned to stations based on the previously defined precedence relationship among the tasks. Tasks in the assembly lines are consecutively assembled on a series of stations. The stations are interconnected by a material handling system for producing a final product. Tasks on the stations are performed in a certain time (called as the task time). Task times are independent of station assignment and they are independent of the preceding task. Each station operates the allocated tasks within a pre-determined and fixed time. The period required to complete the tasks at each station called as cycle time. The problem of assembly line balancing (ALBP) is the problem of determining the amount and order of tasks assigned to stations taking into account one or more optimization criteria [2].

The ALBP was first formulated in [3] and has attracted great interest over the years. ALBP is a problem of NP-hard combinatorial optimization [4]. For this reason, it is difficult to solve problem due to the complex mathematical structure [5].

Assembly lines can be generally categorized into three classes in terms of the variety of products and the number of products assembled in the line:

- (i) Single-model assembly lines where only one product's high volume production is performed,
- (ii) Mixed-model assembly lines where a set of different models of the same basic product is produced, and
- (iii) (iii) multi-model assembly lines where the batches of similar models with intermediate setup operations is produced [6]-[9].

Three versions of the ALBP can be identified by taken into account the used performance measure [10]: Minimizing the station number by taking the given cycle time into

consideration (Type I), Reducing the cycle time by taking the given the station number into consideration (Type II), and maximizing the line efficiency by taking the given cycle time and the station number into consideration (Type E).

Assembly lines can generally be classified as two-sided assembly lines (TALs) and one-sided assembly lines. In one-sided assembly lines, only one side of the line (either right-side or left-side) is used to assembly the tasks to get the final product. In TALs, both of the sides of the line (right side (R) and the left side (L)) are used in parallel to get the final product [11].

TAL structures are often preferred by manufacturers to assembly high volume products such as buses, automobiles and trucks. In a TAL, some tasks may be preferred to be assembled on only one side of the assembly line, while others may be assembled on both sides (E) without side restrictions. ALBPs are combinatorial optimization problem. Therefore, getting an optimal solution becomes too difficult as the size of the problem solved increases. Similarly, TAL balancing problem (TALB) is also a member of NP-hard combinatorial optimization problems [12]. The TALB problems consist of two classes [13],[14]; Type-I: When the cycle time is given, the number of mated- stations are minimized; and Type-II: when the number of mated-stations are given, the cycle time is minimized. However, consider that there are two different solutions with the same number of mated-stations in the Type-I problem. One of these solutions may be more balanced than the other because one of them may have less number of stations than the other. For this reason, the number of stations and the number of mated-stations should be considered when balancing the TAL Type-I problem [15].

There are many studies on different types of ALBPs in the literature. A detailed literature reviews about the different types of ALBPs can be found by [6],[16]-[18] and more recently by [19]. Although many researchers studied single-model TAL balancing problem (STALBP), the studies on mixed-model TAL balancing problem (MTALBP) are very limited in the literature [19]. A mathematical model for MTALBP is proposed by [11]. However, it seemed impossible to solve the model optimally because of high complexity. Ant colony algorithm employing two ants on both sides to simultaneously build a solution for MTALBP, named 2-ANTBAL, is proposed for solving this problem [11]. A mathematical model and a heuristic algorithm (simulated annealing) for MTALBP are developed in [15]. A mathematical programming model is formulated for solving MTALBP with multiple U-shaped layouts and a meta-heuristic algorithm (based on genetic algorithm) is also developed to solve this problem in [20]. A modified particle swarm algorithm with negative knowledge is developed for tackling the multi-objective MTALBP in [21]. A hybrid honey bee mating algorithm is developed to solve MTALBP in [22]. Multi-objective imperialist competitive algorithm is adopted to solve MTALBP in [23]. More recently, a new modified meta-heuristic algorithm (based on particle swarm algorithm using negative knowledge) is proposed for solving MTALBP in [24].

TLBO algorithm proposed by [25],[26] is a new stochastic optimization algorithm that simulates the teaching and learning behaviour in a classroom. TLBO algorithm benefits from the collective intelligence of the learners in the whole class. TLBO algorithm has shown a distinguished performance in addressing optimization problems of continuous non-linear numerical optimization, constrained mechanical design, and constrained benchmark functions [25]-[29].

There are only two papers using TLBO algorithm for solving STALBP in the current literature. TLBO algorithm is utilized to handle the constraints in real application to solve STALBP in [30]. A comparison of TLBO algorithm with any algorithm in the literature is not available in [30]. More recently, a hybrid teaching-learning based optimization (HTLBO) algorithm is proposed to solve the stochastic STALBP with multiple constraints in [1]. A comparison of the HTLBO algorithm with the following algorithms is made in [1]: genetic algorithm [31], tabu search algorithm [32], ant colony-based heuristic algorithm [33], ant colony optimization algorithm [11], bee colony intelligence [34], late acceptance hill-climbing algorithm [35], simulated annealing algorithm [36], TLBO algorithm and improved TLBO algorithm [37], variable neighborhood search (VNS) [38]. Series of experiments demonstrated the excellent performance of HTLBO algorithm and, comparisons among 11 algorithms demonstrated the outstanding performance from HTLBO algorithm. Additionally, HTLBO algorithm also found some new upper bounds for STALBPs.

To the best knowledge of the authors, any study in order to solve the type-I problem of balancing mixed-model TALs (MTALBP-I) by using both TLBO and HTLBO algorithms is not available in the literature. The main direction of this paper is to analyse the performance of original TLBO and HTLBO algorithms on MTALBP-I for the first time in the literature.

The rest of the paper is divided into four sections. Problem characteristics are given in Section 2. This is followed by the structures of the algorithms in Section 3. The performances of the algorithms are tested in Section 4. Finally, conclusions are given in Section 5.

2 Characteristics of the problem

Production of a set of similar models is carried out on a design of mixed-model TAL. The produced models may be in mix model order or any model order. The operators assemble the tasks to each other on a set of mated-stations. The mated-stations are consisted of a pair of stations that are directly standing to each other (right and left side) [11]. A precedence graph is used to show task priorities in each production model. A combined priority diagram is used to combine the precedence diagrams of the models [15]. The combined precedence graph (c) of two models (a), (b) and the related task times of the models are shown in Figure 1. Each task is represented by using a cycle, and the different tasks are connected by using an arrow that denotes the precedence relationship between these tasks. Figure 2 shows an example of TAL. The tasks of models are assembled on a mated-stations set by using the precedence relations on the combined precedence diagram. Each of mated-stations has a stations pairs that are directly opposite to each other (right-side and left-side stations). The tasks of models are assembled in a certain time. A pre-determined planning horizon is used to assembly the product models. The demand request for the model m over the planning horizon is D_m . The cycle time (C) is calculated by $C = \frac{P}{\sum_{m \in \{1,2,\dots,M\}} D_m}$, and the overall proportion of the number of units of model m (q_m) is calculated by $q_m = \frac{D_m}{\sum_{m \in \{1,2,\dots,M\}} D_m}$, $\forall m \in \{1, 2, \dots, M\}$ [11].

The following assumptions are made in this study (MTALBP-I) [15]:

- Models having similar characteristics are assembled on the same TAL,
- Operators use both sides of the lines to assembly the tasks,
- Some tasks may be assembled at only one-side of the line (due to the side restriction), and some tasks may be assembled at either side of the line.
- The concept of combined precedence diagram in [39] is used to handle the precedence diagrams of different models,
- Task times are known beforehand (deterministic task times) and task times are independent from the station where they will be assembled to each other,
- Different models may have the common tasks. A task time may be equal to zero and it may differ from one model to another,
- The travel times of operators for passing one station to another of are ignored,
- Parallel stations are not taken into consideration,
- Work-in-process inventories are not taken into consideration.

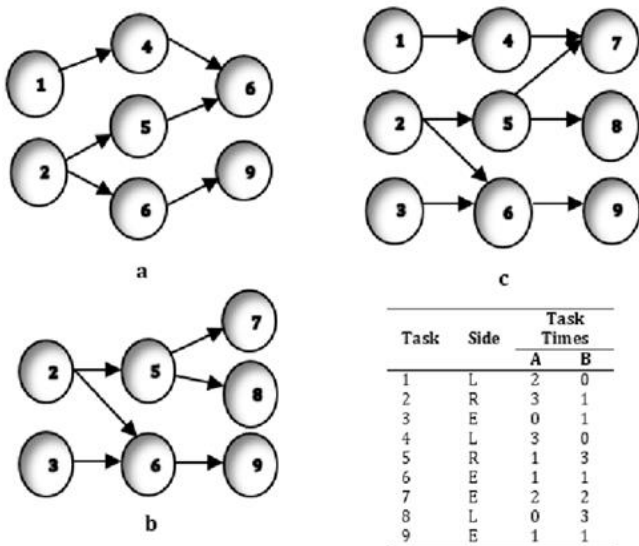


Figure 1: Model precedence graphs, combined precedence graph, directions of two models and task times.

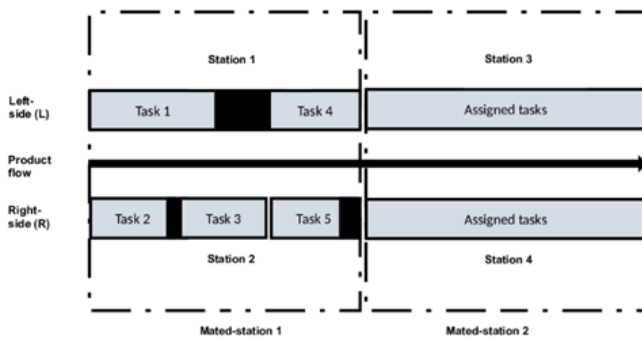


Figure 2: A TAL configuration.

3 TLBO and HTLBO algorithms for MTALBP-I

In this section, a detailed explanation of TLBO and HTLBO algorithms for MTALBP-I and proposed solution methodology are given.

3.1 TLBO algorithm

TLBO algorithm is developed by [25]. The main idea of this algorithm is that the algorithm imitates the teaching and learning process in a classroom. The abilities of teaching and learning for teachers and students in a class are principally imitated by using that algorithm. The TLBO algorithm is divided into two phases:

- Teaching phase and,
- Learning phase. The population of TLBO algorithm includes a group of learners. The best solution in the TLBO algorithm population represents the teacher. Important features of the phases (Teacher and Learner phases) of TLBO algorithm are given in the following subsections.

3.1.1 Teacher phase

In this part of TLBO algorithm the teacher teaches the learners for increasing the mean level of learners from M_1 to level of teacher M_0 . But in practice it is impossible to move from the mean level of the learners M_1 to the level of teacher M_0 . Depending on the capacity of the teacher the mean level of the learners M_1 can be moved to any other value M_2 that is much better than M_1 . Teaching phenomenon is mathematically given as follows [25]. If we consider that T_i is the teacher and M_j is the mean level of the learners at any iteration i , the current level of the mean M_j will be tried to be improved the level T_i by the teacher. The new level of the mean will be M_{new} . The difference between the current level of the mean and the new level of the mean will be given in [25] as follows (see Equation 1);

$$\text{Difference_Mean}_i = r_i(M_{new} - T_F M_j) \quad (1)$$

In Equation 1, T_F decides the mean value to be changed. r_i is the uniform random number between 0 and 1. T_F can take the values of 1 or 2. T_F value is decided randomly as shown follows (see Equation 2):

$$T_F = \text{round}[1 + \text{rand}(0,1)\{2 - 1\}] \quad (2)$$

The new solution is generated by using Difference_Mean as shown in Equation 3.

$$X_{new,i} = X_{old,i} + \text{Difference_Mean}_i \quad (3)$$

3.1.2 Learner phase

In the learning phase of TLBO algorithm, the interaction among the learners is effective to increase the level of knowledge that learners have. In order to improve his/her level of knowledge, the learners interact randomly with each other. Learning phase of TLBO algorithm is mathematically expressed as given below [25].

If we consider two different learners X_i and X_j at each iteration i so that $i \neq j$, the new calculated level of X_i will be M_{new} . If X_{new} gives better function value than X_i , it will be accepted and replaced by X_i (see Equations 4 and 5).

$$X_{new,i} = X_{old,i} + r_i(X_i - X_j) \quad \text{if } F(X_i) < F(X_j) \quad (4)$$

$$X_{new,i} = X_{old,i} + r_i (X_j - X_i) \quad \text{if } F(X_j) < F(X_i) \quad (5)$$

$$x_i = LB + \text{rand}[0,1](UB - LB) \quad (6)$$

Detailed information about TLBO algorithm is given in [26].

3.2 Implementation of TLBO algorithm to solve MTALBP-I

Detailed features of the proposed TLBO algorithm for handling MTALBP-I is given in the following sections. Figure 3 shows the flow chart of the developed TLBO algorithm.

3.2.1 Representation of solution and fitness calculation

A learner in MTALBP-I is represented as an n-dimensional real number vector, $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n}]$. In this vector, the priority of task i is denoted using x_i . Equation (6) is used for generating the task priorities.

In Equation 6, LB represents the minimum number of tasks that the problem has, UB represents the maximum number of tasks that the problem has, and $\text{rand}[0, 1]$ represents the randomly generated value that ranges from zero to one. For being able to evaluate the objective function value, the priority vector must be converted to a task permutation. So, the largest order value (LOV) rule is used for obtaining the task permutation $\pi_i = [\pi_{i,1}, \pi_{i,2}, \dots, \pi_{i,n}]$. This random initialization process is demonstrated with the 9-task problem (see Figure 1) as shown in Table 1. According to LOV rule, $x_{i,3}$ is selected at first and ranks No. 1 in the task sequence since it has the largest value. Then the task $x_{i,9}$ is selected since it has the second largest value.

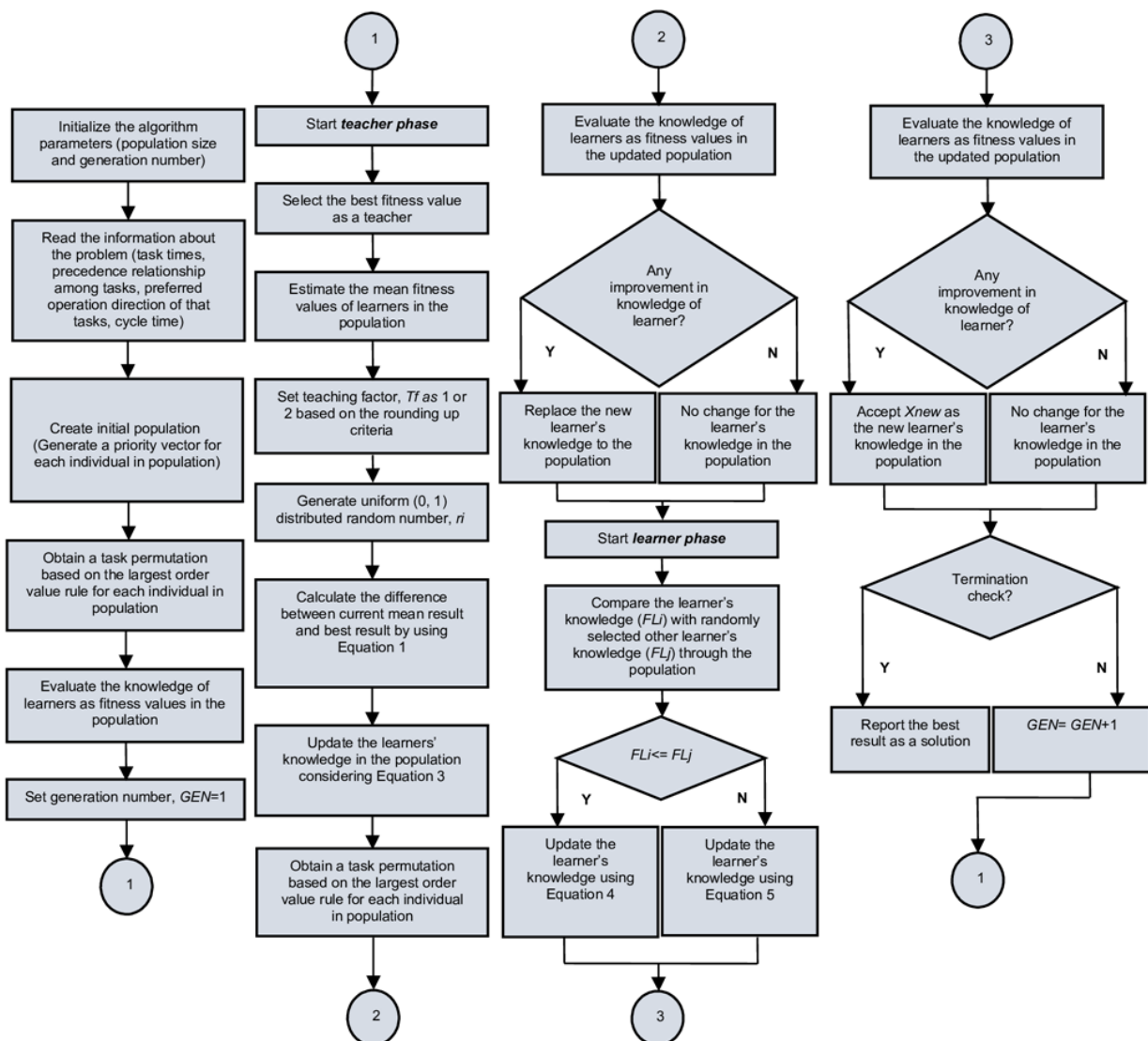


Figure 3: The control logic of TLBO algorithm.

Table 1: Task permutation of individual Xi.

Task	1	2	3	4	5	6	7	8	9
Position	2.36	5.16	8.32	4.45	3.12	1.98	6.68	3.05	7.75
Task sequence	8	4	1	5	6	9	3	7	2
Task permutation	3	9	7	2	4	5	8	1	6

3.2.2 Calculation of the objective function

A priority based performance measure is used to minimize the mated-station number (NM) for a pre-determined cycle time (C) as primary goal and the station number $NS = (NL + NR)$ as secondary goal [24]. For this reason, Equation 7 is used to calculate the objective function value of each solution string.

$$\text{Min } Z = 1,000 \times NM + 100 \times NS \quad (7)$$

Calculation of the objective function procedure begins with the opening the first mated-station (NM) station. Then, a set of assignable tasks list (SAT) that are providing the precedence relations are determined using the combined precedence diagram. The task that has the highest priority in the task list SAT is selected. Then, the selected task is assigned to the first mated-station according to the direction of that task. The assignment of the tasks to the first mated-station is made as much as possible. Then, the next mated station is opened and the first mated-station is closed. These assignments are repeated until all tasks are assigned to the stations. The implementation steps of building a solution for MTALBP-I are given in the following procedure [15]:

- Step 1 : Set $NM = 1, NL = 0, NR = 0, mWL_1^1 = 0$ and $mWL_1^2 = 0$ for all $m \in M$,
- Step 2 : Determine SAT. If SAT has not any assignable task ($\{\emptyset\}$), then go to Step 6,
- Step 3 : Sort the in SAT in increasing order,
- Step 4 : Identify the task h having the highest priority in SAT and assign this task h for which,
- Step 4.1 : If task $h \in A_L$ then.

Step 4.1.1: If $t_{hm} + mWL_{NM}^1 \leq C$ and $t_{hm} + t_{rm}^f \leq C$ ($t_{rm}^f = \max\{t_{pm}^f \mid \text{Task } p \text{ is assigned to the right side of the current mated-station}\}$) for all $m \in M$, then assign task h to the left-side station; $TL_{NM}^1 = TL_{NM}^1 + \{h\}$, and set $t_{hm}^f = \max\{(t_{hm} + mWL_{NM}^1), (t_{hm} + t_{rm}^f)\}$ for all $m \in M$. Set $mWL_{NM}^1 = t_{hm}^f$ for all $m \in M$ and then, go to Step 2. Otherwise go to Step 5.

Step 4.2: If task $h \in A_R$ then,

Step 4.2.1: If $t_{hm} + mWL_{NM}^2 \leq C$ and $t_{hm} + t_{rm}^f \leq C$ ($t_{rm}^f = \max\{t_{pm}^f \mid \text{Task } p \text{ is assigned to the left-side station of the current mated-station}\}$) for all $m \in M$, then assign task h to the right-side station; $TL_{NM}^2 = TL_{NM}^2 + \{h\}$, and set $t_{hm}^f = \max\{(t_{hm} + mWL_{NM}^2), (t_{hm} + t_{rm}^f)\}$ for all $m \in M$. Set $mWL_{NM}^2 = t_{hm}^f$ for all $m \in M$ and then, go to Step 2. Otherwise go to Step 5.

Step 4.3: If task $h \in A_E$ then,

Step 4.3.1: Generate a uniform random number between 0 and 1, p_2 . If p_2 is less than or equal 0.5, then go to Step 4.1.1. Otherwise go to Step 4.2.1.

Step 5: If the tasks in SAT are not assigned to any side of the current mated-station, then open a new mated-station. If $TL_{NM}^1 \neq \{\emptyset\}$ then $NL = NL + 1$. If $TL_{NM}^2 \neq \{\emptyset\}$ then $NR = NR + 1$. Set $NM = NM + 1, mWL_{NM}^1 = 0$ and $mWL_{NM}^2 = 0$ for all $m \in M$, and then, go to Step 2.

Step 6: Stop.

3.3 HTLBO algorithm

HTLBO algorithm consists of three parts [1]: TLBO algorithm, the crossover operator, and the variable neighborhood search (VNS) [38], as shown in Figure 4. TLBO algorithm and the

crossover operator cooperate for enhancing the global search; and VNS based on seven neighborhood operators are used for enhancing the improvements on the individual itself. The crossover operator is applied by preserving these "blocks" of task permutation since the learner phase based on random-keys can generate differentiated solutions and may lose some efficient task permutation "blocks". Besides, the VNS works as a strong local search method and seven neighborhood operators increase the probability of finding a better solution. The combination achieves the balance between intensification and diversification within the population.

The details of the proposed HTLBO algorithm for solving MTALBP-I are given in the following sections.

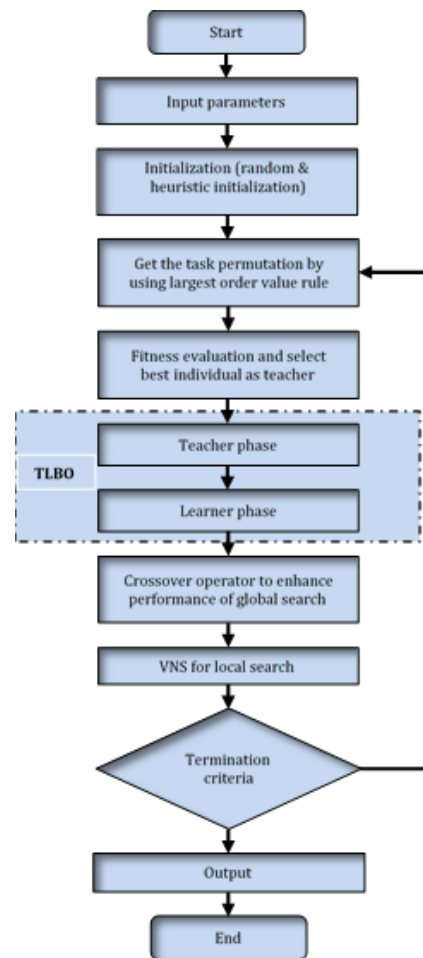


Figure: 4 Main body of HTLBO algorithm.

3.3.1 Initialization of the initial population

The random initialization can promise the diversity of the initial population, but the population may lack high-quality individuals. In order to speed up the process of evolution, a heuristic initialization is also applied together with the random initialization. Two heuristic factors, namely the operation times t_i [40] and the number of immediate successors IF_i [41], have presented promising results for classical one-sided assembly line balancing problems. Therefore, these two heuristic factors are employed for improving the qualities of initial solutions [1]. Based on the two factors and their weighted modulus, λ_1 and λ_2 , the related weights of the tasks can be calculated by Equation (8), where $\lambda_1 + \lambda_2 = 1$.

$$w_i = \lambda_1 \times \left(\frac{t_i}{\sum_{j \in I} t_j} \right) + \lambda_2 \times \left(\frac{IF_i}{\sum_{j \in I} IF_j} \right) \quad (8)$$

Where; nt = number of tasks, I = Set of tasks, and $\forall i \in I$. The task with the largest synthesis weight should be selected at first, and a task permutation is obtained just like random initialization. As a result of the tests carried out in [1], the best solutions are obtained by generating 50% of the individuals in the initial population randomly and %50 of them heuristically. In the same way, 50% of the individuals in the initial population are generated randomly and %50 of them are generated heuristically for solving MATALBP-I.

3.3.2 Enhancement of global search by crossover operator

The original TLBO algorithm updates gradually the population based on the random-keys method, which can generate differentiated solutions. However, the new solutions may lose some efficient task permutation "blocks". For this reason, the crossover operator in the genetic algorithm is integrated into the learning phase of TLBO algorithm, which aims at preserving the efficient "blocks" of task permutation for increasing the search speed and enhancing the global search. The crossover operator exchanges the contiguous sections of the parent solutions in order to produce a new offspring. The offspring chromosomes carry partial features of their parents. As a result of the tests carried out in [1], the most effective solutions are obtained in cases where the tow-point crossover is used. Similarly, the two-point crossover is used in HTLBO algorithm for solving MTALBP-I. In two point crossover, two points are randomly generated that cut each of the parents into three parts. Two fragments, called as head and tail of the parent, are copied into the offspring. After which, the empty positions in the offspring (the middle sections of the offspring) are sequentially filled in according to the elements of the other parent in order, but skipping over all elements already present in the offspring [42].

3.3.3 Local search by variable neighborhood search

Since a systematic change of neighborhood is helpful in increasing the probability of finding a better solution [38], the variable neighborhood search (VNS) is employed for enhancing the local search ability of the TLBO algorithm [1]. Seven neighborhood operators ($N_k, k = 1, 2, \dots, 7$) are used, including backward-insert, forward-insert, neighbor-swap, swap, inverse, multi- insert and multi-swap. These neighborhood operators are depicted in Figure 5. Note that, VNS is hired in each of iterations of HTLBO algorithm to improve the diversity of solutions and avoid being trapped in a local optimum. The procedure for utilizing seven neighborhood operators is shown as follows [1]:

- Step 1 : Generate an initial solution x ,
- Step 2 : Obtain local optimum x' with the k th neighborhood operator (N_k),
- Step 3 : If this local optimum is better than the incumbent, $x = x'$ and set $k = 1$; otherwise, set $k = k + 1$ when $k < k_{max}$, or $k = 1$ when $k = k_{max}$,
- Step 4 : If the termination criterion is satisfied, stop this process; otherwise, go to Step 2.

4 Numerical study

TLBO and HTLBO codes are written using Matlab 7.8.0 and the generated codes are executed on a 3.00 GHz Pentium 4 computer. In order to compare the efficiency of the algorithms with the existing methods, the test problems called as P9, P12, P16, P24, P65, P148 and P205 are taken from the current literature. Test problems are divided into two groups called as the small-sized test problems (P9, P12, P16 and P24) and the big-sized test problems (P65, P148 and P205). P9, P12 and P24 test problems are generated in [31]. P16 and P65 test problems are generated in [13]. P148 test problem is generated in [12]. For the P205 test problem, the operation directions of the tasks and the precedence relationship are taken from [13]. Task times for the P205 test problem are taken from [15]. The numbers of units of all models for the overall proportions are the same ($q_A = q_B = \dots = q_m$).

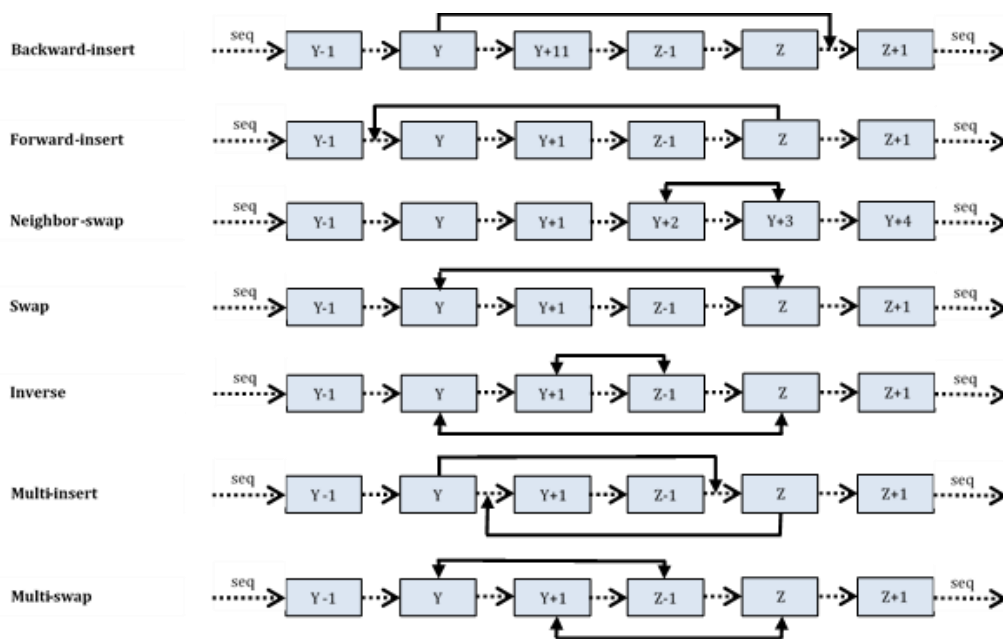


Figure 5: Seven neighborhood operator.

4.1 Configuration of parameters

The proper parameters design has an important effect on the efficiency of the proposed algorithms. The attitudes of the proposed algorithms with different parameters are examined in this subsection. The proposed algorithms can be defined by the control parameter set $\Pi = \{NP, GEN\}$ where;

- NP is the size of the population,
- GEN is the number of the generation.

The approach of the statistical design of experiments (DOE) is employed to optimize this parameter set [43]. The DOE is an investigative method that is extremely important in terms of effectiveness and efficiency in assessing the impact of multiple factors on a process. Levels of the parameters are determined through preliminary experiments and different levels of parameters are considered to be analyzed for two different problem sizes as shown in Table 2. As shown in Table 2, each group of problems has two parameters with four levels. P16 that has 16 tasks and 2 models is chosen to calibrate the parameters of the small-sized test problems; P148 that has 148 tasks and 4 models is chosen to calibrate the parameters of the big-sized test problems. The selected problems have the medium complexity in their groups. 20 independent runs at each design point and a total of 640 runs ($16 \times 20 \times 2$) are made. A statistical analysis of variance (ANOVA) is performed for each test problem group to determine which control parameter effects are significant. According to the ANOVA results in Table 3 and Table 4, each of the main effects and interactions are found to be significant at the 5% level. The main effects plot of the obtained mean values at each level for the small-sized test problems and the big-sized test problems are shown in Figures 6 and 7, respectively. It is possible to recommend that the values of NP=20 and GEN=20 and the values of NP=80 and GEN=150 can be used as efficient control parameters for the small and big-sized test problems, respectively. The same parameter sets are used to test the efficiency of the TLBO algorithm.

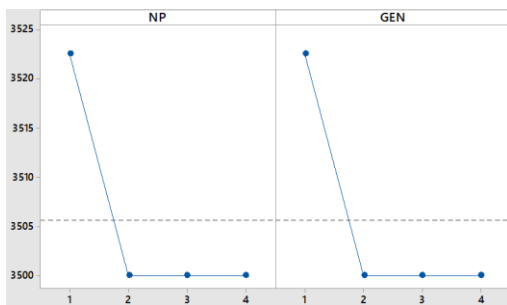


Figure 6: Main effects plot (data means) for the small-sized test problems.

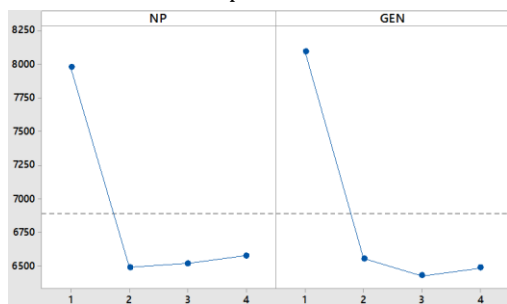


Figure 7: Main effects plot (data means) for the big-sized test problems.

Table 2: Considered levels of HTLBO algorithm parameters for each group problems.

TLBO algorithm parameters	Small-sized test problems	Big-sized test problems
NP	10	40
	20	80
	30	120
	40	160
GEN	10	50
	20	100
	30	150
	40	200

Table 3: ANOVA for P16 with cycle time 19 (small-sized test problems).

Source of variation	Degree of freedom	Sum of squares	Mean of squares	F_{calc}	Prob($F > F_{calc}$)
NP	3	15187	5062,5	81	0
GEN	3	15188	5062,5	81	0
NP*GEN	9	45562	5062,5	81	0
Error	144	9000	62,5		
Total	159	84937			

Table 4: ANOVA for P148 with cycle time 357 (big-sized test problems).

Source of variation	Degree of freedom	Sum of squares	Mean of squares	F_{calc}	Prob($F > F_{calc}$)
NP	3	62808750	20936250	55,84	0
GEN	3	77273250	25757750	68,70	0
NP*GEN	9	7415750	823972	2,20	0,025
Error	144	53990000	374931		
Total	159	201487750			

4.2 Computational results

TLBO and HTLBO algorithms are run ten times for each benchmark problem. TLBO and HTLBO algorithms are compared with the algorithms that yield the best solutions for the benchmark instances in the current literature. To the best of our knowledge, the best known upper bounds for the test problems were found by the following algorithms:

- SA: Simulated Annealing algorithm [15],
- PSO: Particle Swarm Optimization [24].

Therefore, the performance of TLBO and HTLBO algorithms for solving the MTALBP-I is compared against these algorithms reported in the literature.

Table 5 shows the computational results of TLBO and HTLBO algorithms for the small-sized test problems. As it can be seen from Table 5, both of TLBO and HTLBO algorithms produce the best solutions in small CPU times for all of the small-sized test problems, as is the case in the SA and the PSO. The computational results of TLBO and HTLBO algorithms for the big-sized test problems are shown in Table 6.

As it can be seen from Table 6, HTLBO algorithm produces the best solutions for all of the big-sized test problems, as is the case in the PSO. Once the minimum solution values obtained by SA and TLBO in Table 6 are compared, both algorithms have found the same solution values for the big-sized test problems no. 1, 2, 3, 4, 5, 6, 13, 14, 15, 16, 17, 18, 19, 20, 21 and 22.

Table 5: Experiment results of the small-sized test problems.

Problem	M	Instance Number	Cycle time	LB	SA		CPU (s)	PSO		CPU (s)	TLBO		CPU (s)	HTLBO		CPU (s)
					NM	NS		NM	NS		NM	NS		NM	NS	
P9	2	1	4	4	3	4	1,07	3	4	<1	3	4	<1	3	4	<1
		2	5	3	2	3	0,96	2	3	<1	2	3	<1	2	3	<1
		3	6	3	2	3	0,91	2	3	<1	2	3	<1	2	3	<1
P12	2	4	5	5	3	5	2,17	3	5	<1	3	5	<1	3	5	<1
		5	6	4	2	4	2,11	2	4	<1	2	4	<1	2	4	<1
		6	7	3	2	4	2,05	2	4	<1	2	4	<1	2	4	<1
		7	8	3	2	3	1,93	2	3	<1	2	3	<1	2	3	<1
P16	2	8	15	5	4	6	5,18	4	6	<1	4	6	<1	4	6	<1
		9	16	5	4	6	5,08	4	6	<1	4	6	<1	4	6	<1
		10	18	4	3	5	5,05	3	5	<1	3	5	<1	3	5	<1
		11	19	4	3	5	4,82	3	5	<1	3	5	<1	3	5	<1
		12	21	4	2	4	4,71	2	4	<1	2	4	<1	2	4	<1
		13	22	4	2	4	4,65	2	4	<1	2	4	<1	2	4	<1
P24	2	14	20	7	4	7	17,70	4	7	<1	4	7	<1	4	7	<1
		15	24	6	3	6	18,71	3	6	<1	3	6	<1	3	6	<1
		16	25	5	3	6	17,46	3	6	<1	3	6	<1	3	6	<1
		17	30	5	3	5	16,23	3	5	<1	3	5	<1	3	5	<1
		18	35	4	2	4	15,09	2	4	<1	2	4	<1	2	4	<1
		19	40	4	2	4	15,08	2	4	<1	2	4	<1	2	4	<1

Table 6: Experiment results of the big-sized test problems.

Problem	M	Instance Number	Cycle time	LB	SA		CPU (s)	PSO		CPU (s)	TLBO		CPU (s)	HTLBO		CPU (s)
					NM	NS		NM	NS		NM	NS		NM	NS	
P65	3	1	326	8	5	9	528	5	9	110	5	9	165	5	9	278
		2	381	7	4	8	523	4	8	<10	4	8	163	4	8	281
		3	435	6	4	7	515	4	7	<10	4	7	161	4	7	279
		4	490	6	3	6	505	3	6	<10	3	6	162	3	6	277
		5	544	5	3	6	504	3	6	<1	3	6	161	3	6	279
P148	4	6	204	13	9	17	4025	9	17	<10	9	17	190	9	17	605
		7	255	11	7	14	3859	7	13	<10	7	13	192	7	13	636
		8	306	9	6	12	3845	6	11	<10	6	11	191	6	11	644
		9	357	8	5	10	3835	5	9	<10	5	9	190	5	9	615
		10	408	7	5	10	3826	4	8	<10	4	8	199	4	8	657
		11	459	6	4	8	3810	4	7	<10	4	7	198	4	7	634
		12	510	6	4	8	3829	3	6	<15	3	6	193	3	6	612
P205	5	13	1133	11	8	15	NA	7	14	420	8	15	547	7	14	1282
		14	1322	9	7	13	NA	6	12	<10	7	13	553	6	12	1285
		15	1510	8	6	12	NA	6	11	<60	6	12	548	6	11	1278
		16	1699	7	5	10	NA	5	10	<10	5	10	548	5	10	1298
		17	1888	7	5	10	NA	4	8	<20	5	10	554	4	8	1299
		18	2077	6	5	9	NA	4	8	<10	5	9	550	4	8	1322
		19	2266	6	4	8	NA	4	8	<10	4	8	554	4	8	1289
		20	2454	5	4	8	NA	3	6	<25	4	8	551	3	6	1320
		21	2643	5	4	7	NA	3	6	<10	4	7	549	3	6	1322
		22	2832	5	3	6	NA	3	6	<10	3	6	552	3	6	1321

For all the remaining problems, TLBO approach has found higher quality solutions than the SA algorithm. From these computational results, it can be concluded that the TLBO algorithm is able to solve P9, P12, P16, P24, P65 and P148 in higher quality when compared to HTLBO and PSO algorithms. In the problem handled, the search space grows with the problem size. As the problem size increases, HTLBO algorithm finds better solutions. It can be seen from the experimental results that TLBO algorithm is a sufficient and powerful algorithm to obtain good solutions for solving the MTALBP-I.

It seems again that the HTLBO is a very effective and comparable algorithm for solving the MTALBP-I. Due to the existing different influencing factors such as hardware, software, and coding, the computational times of the algorithms cannot be compared fairly. However, the experiments reveal that TLBO and HTLBO algorithms have solved all the problem

instances in a fairly small computation time. This result shows that the HTLBO algorithm has a high potential in solving the MTALBP-1 in an acceptable computation time.

5 Conclusions

The TLBO algorithm has recently been developed based on a source of inspiration from the teaching and learning process in a classroom. The algorithm imitates the teaching and learning abilities of teachers and students in a classroom. TLBO algorithm is successively applied by its developers and several other researchers to solve some design optimization problems and some constrained and unconstrained nonlinear programming problems. TLBO algorithm has been tested for the first time in this study on the balancing of the mixed model two-sided assembly line type-I problem (MTALBP-I). In order to understand whether the TLBO algorithm has an effective

potential as well as the best-known algorithms developed to solve the MTALBP-I, it is solved using a series of test problems and compared with the best-known algorithms. More recently, a hybrid teaching-learning based optimization (HTLBO) algorithm is proposed by [1] for solving the balancing of stochastic simple two-sided assembly lines problem and in this study, the HTLBO algorithm is compared with the best known 10 different heuristic algorithms known in the literature. HTLBO algorithm has presented more outstanding performance when compared to other algorithms as a result of the tests performed in [1]. In this paper, HTLBO is also adapted algorithm for solving the MTALBP-I and its performance is analyzed. According to the experimental results, it can be concluded that TLBO algorithm has a significant potential as important as some of the best known heuristic algorithms for the problem and HTLBO algorithm performs as good as the best-known heuristic algorithms in the literature. It can be concluded that the HTLBO is an effective algorithm to solve MTALBP-I.

6 References

- [1] Tang Q, Li Z, Zhang L, Zhang C. "Balancing stochastic two-sided assembly line with multiple constraints using hybrid teaching-learning-based optimization algorithm". *Computers & Operations Research*, 82, 102-113, 2017.
- [2] Scholl A, Becker C. "State-of-the-art exact and heuristic solution procedures for simple assembly line balancing". *European Journal of Operational Research*, 168(3), 666-693, 2006.
- [3] Salvesson ME. "The assembly line balancing problem". *Journal of Industrial Engineering*, 6, 18-25, 1955.
- [4] Karp RM. *Reducibility Among Combinatorial Problems*. Editors: Miller RE, Thatcher JW. Complexity of computer applications, 85-104, New York, NY, USA, Plenum Press, 1972.
- [5] Erel E, Gokcen H. "Shortest-route formulation of mixed-model assembly line balancing problem". *European Journal of Operational Research*, 116(1), 194-204, 1999.
- [6] Becker C, Scholl A. "A survey on problems and methods in generalized assembly line balancing". *European Journal of Operational Research*, 168, 694-715, 2006.
- [7] Boysen N, Fliedner M, Scholl A. "Assembly line balancing: Which model to use when?". *International Journal of Production Economics*, 111, 509-528, 2008.
- [8] Rekiek B, Delchambre A. *Assembly Line Design: The Balancing of Mixed Model Hybrid Assembly Lines with Genetic Algorithms*. London, England, Springer, 2006.
- [9] Scholl A. *Balancing and Sequencing of Assembly Lines*. Heidelberg, Germany, Physica-Verlag, 1999.
- [10] Scholl A, Klein R. "ULINO: Optimally balancing U-shaped JIT assembly lines". *International Journal of Production Research*, 7(4), 721-736, 1999.
- [11] Simaria AS, Vilarinho PM. "2-ANTBAL: An ant colony optimisation algorithm for balancing two-sided assembly lines". *Computers & Industrial Engineering*, 56(2), 489-506, 2009.
- [12] Bartholdi JJ. "Balancing two-sided assembly lines: A case study". *International Journal of Production Research*, 31, 2447-2461, 1993.
- [13] Lee TO, Kim Y, Kim YK. "Two-sided assembly line balancing to maximize work relatedness and slackness". *Computers & Industrial Engineering*, 40, 273-292, 2001.
- [14] Wu EF, Jin Y, Bao JS, Hu XF. "A branch-and-bound algorithm for two-sided assembly line balancing". *International Journal of Advanced Manufacturing Technology*, 39, 1009-1015, 2008.
- [15] Özcan U, Toklu B. "Balancing of mixed-model two-sided assembly lines". *Computers & Industrial Engineering*, 57(1), 217-227, 2009.
- [16] Ghosh S, Gagnon RJ. "A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems". *International Journal of Production Research*, 27, 637-670, 1989.
- [17] Erel E, Sarin SC. "A survey of the assembly line balancing procedures". *Production Planning and Control*, 9, 414-434, 1998.
- [18] Batini D, Faccio M, Ferrari E, Persona A, Sgarbossa F. "Design configuration for a mixed-model assembly system in case of low product demand". *The International Journal of Advanced Manufacturing Technology*, 34, 188-200, 2007.
- [19] Li Z, Kucukkoc I, Nilakantan JM. "Comprehensive review and evaluation of heuristics and meta-heuristics for two-sided assembly line balancing problem". *Computers & Operations Research*, 84, 146-161, 2017.
- [20] Rabbani M, Moghaddam M, Manavizadeh N. "Balancing of mixed-model two-sided assembly lines with multiple U-shaped layout". *The International Journal of Advanced Manufacturing Technology*, 59(9-12), 1191-1210, 2012.
- [21] Chutima P, Chimklai P. "Multi-objective two-sided mixed model assembly line balancing using particle swarm optimisation with negative knowledge". *Computers & Industrial Engineering*, 62, 39-55, 2012.
- [22] Yuan B, Zhang C, Shao X, Jiang Z. "An effective hybrid honey bee mating optimization algorithm for balancing mixed-model two-sided assembly lines". *Computers & Operations Research*, 53, 32-41, 2015.
- [23] Li D, Zhang C, Tian G, Shao X, Li Z. "Multiobjective program and hybrid imperialist competitive algorithm for the mixed-model two-sided assembly lines subject to multiple constraints". *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 99, 1-11, 2016.
- [24] Delice Y, Aydoğan EK, Özcan U, İlkay MS. "A modified particle swarm optimization algorithm to mixed-model two-sided assembly line balancing". *Journal of Intelligent Manufacturing*, 28, 23-36, 2017.
- [25] Rao RV, Vakharia DP, Savsani VJ. "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems". *Computer-Aided Design*, 43(3), 303-315, 2011.
- [26] Rao RV, Savsani VJ, Vakharia DP. "Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems". *Information Sciences*, 183, 1-15, 2012.
- [27] Niknam T, Azizipanah-Abarghooee R, Narimani MR. "A new multi-objective optimization approach based on TLBO for location of automatic voltage regulators in distribution systems". *Engineering Applications of Artificial Intelligence*, 25(8), 1577-1588, 2012.
- [28] Rao RV, Savsani VJ. *Mechanical Design Optimization using Advanced Optimization Techniques*. London, England, Springer-Verlag, 2012.
- [29] Rao RV, Patel V. "Multi-objective optimization of heat exchangers using a modified teaching-learning-based optimization algorithm". *Applied Mathematical Modelling*, 37(3), 1147-1162, 2013.

- [30] Tuncel G, Aydin D. "Two-sided assembly line balancing using teaching-learning based optimization algorithm". *Computers & Industrial Engineering*, 74, 291-299, 2014.
- [31] Kim YK, Kim Y, Kim YJ. "Two-sided assembly line balancing: A genetic algorithm approach". *Production Planning and Control*, 11(1), 44-53, 2000.
- [32] Özcan U, Toklu B. "A tabu search algorithm for two-sided assembly line balancing". *The International Journal of Advanced Manufacturing Technology*, 43, 822-829, 2009.
- [33] Baykasoglu A, Dereli T. "Two-sided assembly line balancing using an ant-colony-based heuristic". *The International Journal of Advanced Manufacturing Technology*, 36, 582-588, 2008.
- [34] Özbakir L, Tapkan P. "Bee colony intelligence in zone constrained two-sided assembly line balancing problem". *Expert Systems with Applications*, 38(9), 11947-11957, 2011.
- [35] Yuan B, Zhang C, Shao X. "A late acceptance hill-climbing algorithm for balancing two-sided assembly lines with multiple constraints". *Journal of Intelligent Manufacturing*, 26(1), 159-168, 2015.
- [36] Khorasanian D, Hejazi SR, Moslehi G. "Two-sided assembly line balancing considering the relationships between tasks". *Computers & Industrial Engineering*, 66(4), 1096-1105, 2013.
- [37] Li D, Zhang C, Shao X, Lin W. "A multi-objective TLBO algorithm for balancing two-sided assembly line with multiple constraints". *Journal of Intelligent Manufacturing*, 27, 725-739, 2016.
- [38] Mladenovic N, Hansen P. "Variable neighborhood search". *Computers & Operations Research*, 24, 1097-1100, 1997.
- [39] Macaskill JLC. "Production-line balances for mixed model lines". *Management Science*, 19, 423-434, 1972.
- [40] Tonge FM. "Summary of a heuristic line balancing procedure". *Management Science*, 7(1), 21-42, 1960.
- [41] Helgeson WB, Birnie DP. "Assembly line balancing using the ranked positional weight technique". *Journal of Industrial Engineering*, 12(6), 394-398, 1961.
- [42] Hamzadayı A, Yildiz G. "Modeling and solving static m identical parallel machines scheduling problem with a common server and sequence dependent setup times". *Computers & Industrial Engineering*, 106, 287-298, 2017.
- [43] Montgomery DC. *Design and Analysis of Experiments*. New York, NY, USA, John Wiley & Sons, 2000.