Balancing stochastic two-sided assembly line with multiple constraints using hybrid teaching-learning-based optimization algorithm

Qihua Tang^{1*}, Zixiang Li¹, LiPing Zhang¹ and Chaoyong Zhang²

¹ Industrial Engineering Department, Wuhan University of Science and Technology, Wuhan 430081, Hubei, China

² State Key Laboratory of Digital Manufacturing Equipment & Technology, Huazhong University of Science & Technology, Wuhan, Hubei 430074, China

* Corresponding author: Qihua Tang, E-mail: tangqihua@wust.edu.cn

Abstract: Two-sided assembly lines are usually found in the factories which produce large-sized products. In most literatures, the task times are assumed to be deterministic while these tasks may have varying operation times in real application, causing the reduction of performance or even the infeasibility of the schedule. Moreover, the ignorance of some specific constraints including positional constraints, zoning constraints and synchronism constraints will result in the invalidation of the schedule. To solve this stochastic two-sided assembly line balancing problem with multiple constraints, we propose a hybrid teaching-learning-based optimization (HTLBO) approach which combines both a novel teaching-learning-based optimization algorithm for global search and a variable neighborhood search with seven neighborhood operators for local search. Especially, a new priority-based decoding approach is developed to ensure that the selected tasks satisfy most of the constraints identified by multiple thresholds of the priority value and to reduce the idle times related to sequence-dependence among tasks. Experimental results on benchmark problems demonstrate both remarkable efficiency and universality of the developed decoding approach, and the comparison among 11 algorithms shows the effectiveness of the proposed HTLBO.

Keywords: Stochastic two-sided assembly line balancing; Teaching-learning-based Optimization; Variable neighborhood search; Multiple constraints

1. Introduction

Two-sided assembly lines are usually designed to produce large-sized high-volume products. In this two-sided line, some tasks may be preferred to be operated on exactly one side of this line (called L-type or R-type tasks), while others can be operated on either side of the line (E-type tasks). And, in a typical two-sided assembly line shown in Fig.1, tasks are operated in parallel on both sides. A pair of facing stations (E.g. station 1 and station 2) on this line is called a mated-station (e.g. mated-station 1), and one of them calls the other a companion. Two-sided assembly lines can provide many advantages over the well-known one-sided assembly lines. They are: (1) the length of assembly line can be shortened, (2) throughput and setup time can be reduced, (3) the cost of fixtures and tools can be decreased and (4) the material handling can be lessened (Bartholdi JJ, 1993).

Compared with the one-sided assembly line balancing, the distinguishing feature of the two-sided is the restriction on the operation directions. Due to the combination of the direction constraint and precedence constraint, idle times between two successive tasks are always unavoidable. For example, supposed that task k is an immediate predecessor of task l in Fig. 1, task l cannot be started until task k is completed, resulting in the sequence-dependent waiting time in station 1.

Fig.1 A two-sided assembly line

On the other hand, multiple constraints existing in real application, which did not receive enough attention due to its complexity, should be taken into account. For example, synchronism constraints arise when two operators on both sides of the same mated-station need to collaborate (Simaria and Vilarinho, 2009). Zoning constraints may occur when some tasks are forced or not allowed to be assigned to the same station or mated-station (Baykasoglu and Dereli, 2008). If tasks must be operated on the same station or mated-station, positive zoning constraints often show up. If tasks are prohibited on the same station or mated-station, negative zoning constraints are the consequence. Positional constraints arise when certain tasks need to be allocated to a predetermined station due to heavy or immovable facilities in that station (Kim et al., 2000).

In most literatures concerning two-sided assembly line balancing (TALB) problems, the operation times of tasks are assumed to be deterministic. But in real application, the tasks may have varying

operation times due to machine breakdown, lack of employee training, loss of motivation, non-qualified operators, complex tasks, and unfavorable environment (Özcan, 2010; Chiang et al., 2015). If the stochastic attribute of operation times is ignored, the performing schedule may become less productive. Even worse, the schedule sometimes becomes infeasible and leads to the interruption of production.

To our best knowledge, only two papers deal with the stochastic two-sided assembly line balancing (STALB) problem (Özcan, 2010; Chiang et al., 2015), and no paper focuses on the multiple constraints in stochastic two-sided assembly line. Therefore, this paper concentrates on the STALB with multiple constraints (STALB-MC), and we mainly represent three contributions to this problem as follows. (1) The mathematical model for stochastic two-sided assembly with multiple constraints is presented and a set of new benchmark problems are tested. (2) A new universal priority-based decoding approach is developed to tackle stochastic operation times and multiple constraints, in order to reduce idle times related to sequence-dependence of the tasks and increase the possibility of obtaining feasible solutions. (3) A hybrid teaching-learning-based optimization (TLBO) is utilized to seek for optimal solutions. The TLBO has shown distinguished performance in addressing optimization problems of constrained benchmark functions, constrained mechanical design, and continuous non-linear numerical optimization (Rao et al., 2011, Niknam et al., 2012, Rao et al., 2012 and Rao and Patel, 2013). The main features of the proposed TLBO algorithm include the heuristic initialization, the crossover operator to enhance global search and the variable neighborhood search for a strong local search. Several other algorithms, including a genetic algorithm, a tabu search algorithm, a bee colony intelligence and etc., are extended to solve the STALB problem and we also carry out a comparative evaluation of these meta-heuristics.

This paper is organized as follows: Section 2 presents the literature review on STALB problems; Section 3 deduces the formulation of the STALB problems; Section 4 details implementing the HTLBO for STALB problems, especially the methodology to cope with multiple constraints; Section 5 discusses the experiment results and compares the proposed algorithm with other established ones; Section 6 provides the conclusions and proposes future research.

2. Literature review

Bartholdi (1993) was the first to address and solve the two-sided assembly line balancing (TALB) problem. After that, more approaches are developed, which can be divided into two groups: exact algorithms and heuristic/meta-heuristics algorithms. For exact algorithms, Hu et al. (2008) provided the lower bound of the station number and presented a station-oriented enumerative algorithm to find optimal solutions for small-size problems. Wu et al. (2008) proposed the branch-and-bound algorithms to solve both small-size and certain large-size problems optimally. Hu et al. (2010) developed dominance rules and reduction rules in the branch-and-bound algorithms to solve large-size TALB problems, and most of them were optimal solutions. To speed up the computational process, heuristic/meta-heuristic algorithms are gradually adopted. Kim et al. (2000) proposed a meta-heuristic named genetic algorithm to deal with TALB with positional constraints. Lee et al. (2001) came up with a heuristic-based assignment procedure to maximize the relatedness and work slackness. Özcan and Toklu (2009a) developed a tabu search algorithm, in which two objectives of minimizing the number of stations and the smoothness index were achieved concurrently. Chutima and Chimklai (2012) used the particle swarm optimization algorithm and Özcan and Toklu (2009b) utilized the simulated annealing algorithm to solve the mixed-model TALB problem.

To deal with multiple constraints in TALB problem, Kim et al. (2000) addressed positional constraints. Baykasoglu and Dereli (2008), Özbakir and Tapkan (2011) considered zoning constraints. Simaria and Vilarinho (2009) considered the zoning and synchronism constraints. Then Tapkan et al. (2012a) proposed bee algorithm to deal with the positional constraints, the zoning constraints and the synchronism constraint. Yuan et al. (2015), Li et al. (2014), Wang et al. (2014), and Purnomo et al. (2013) used late acceptance hill-climbing algorithm, teaching-learning-based optimization algorithm, hybrid imperialist competitive algorithm, and genetic algorithm respectively to handle the above three constraints. And Tuncel and Aydin (2014) also utilized teaching-learning-based optimization algorithm to handle the constraints in real application. With respect to the decoding method for TALBP problem with multiple constraints, there are only a few researches including Yuan et al. (2015) and Li et al. (2014). And all of them ignore the positional constraint in the decoding scheme. Moreover, these researches allocate the tasks in positive zoning constraint as a whole, although they only need to be in the same station or mated-station in real application.

To cope with the uncertain factors in real application of the two-sided assembly lines, Özcan (2010) built a chance-constrained mixed integer programming model to approach the TALB problem with stochastic operation times. Chiang et al. (2015) improved this model by considering the probability that

both sides of the mated-station are finished within the cycle time. Özbakir and Tapkan (2010) considered the fuzzy multi-objective two-sided assembly lines and the bee algorithm was proposed to handle the imprecise objectives. Tapkan et al. (2012b) tackled the fuzzy multi-objective two-sided assembly line balancing problem with all the three additional constraints via bee algorithm.

As we can see, multiple constraints and stochastic operation times are two main aspects in two-sided assembly lines. However, no study has been performed to consider these two factors concurrently. Moreover, no research focuses on the decoding methodology of STALB problem with multiple constraints. This study thus concentrates on STALB problem with multiple constraints with new recent algorithm, and develops a new priority-based decoding approach to deal with multiple constraints specially.

3. Stochastic two-sided assembly line balancing

3.1 Problem assumptions

Basic assumptions of the STALB problem with multiple constraints in this study are given as follows:

- 1) The operation times of tasks are stochastic and distributed with normal distribution independently.
- 2) The precedence relationships among tasks and the preferred directions are known.
- 3) All tasks should be operated in parallel on both sides of the line within a given fixed cycle time.
- 4) No buffer is installed on this line.
- 5) A task can be operated only after all its immediate predecessors have been completed.
- 6) Each task must be assigned to only one station and be operated exactly once.
- 7) The tasks with positive zoning constraints must be operated on the same station.
- 8) Each pair of tasks with negative zoning constraints cannot be assigned to the same mated-station.
- 9) The tasks with synchronism constraints must be operated simultaneously on both sides of the same mated-station.

3.2 Notations

The notations used in this mathematical formulation are summarized as follows.

Indices:

i, h, p : A task.

j, g : A mated-station.

k : A side of the line; $k = \begin{cases} 1, & \text{if the side is left} \\ 2, & \text{if the side is right} \end{cases}$

(j, k) : the k side station of the mated-station j .

Parameters:

nt : Number of tasks.

nm : Number of mated-stations.

ns : Number of stations.

I : Set of tasks, $I = \{1, 2, \dots, i, \dots, nt\}$.

J : Set of mated-stations; $J = \{1, 2, \dots, j, \dots, nm\}$.

AL : Set of tasks which should be performed at a left station, $AL \subseteq I$.

AR : Set of tasks which should be performed at a right station, $AR \subseteq I$.

AE : Set of tasks which can be performed on either side of a mated-station, $AE \subseteq I$.

P_0 : Set of tasks that have no immediate predecessors.

$P_a(i)$: Set of all predecessors of the task i .

$P(i)$: Set of immediate predecessors of the task i .

$S(i)$: Set of immediate successors of the task i .

$S_a(i)$: Set of successors of the task i .

$C(i)$: Set of tasks whose operation directions are opposite to the operation direction of task i ;

$$C(i) = \begin{cases} AL & \text{if } i \in AR \\ AR & \text{if } i \in AL \\ \emptyset & \text{if } i \in AE \end{cases}$$

$K(i)$: Set of integers which indicate the preferred operation directions of the task i ;

$$K(i) = \begin{cases} \{1\} & \text{if } i \in \text{AL} \\ \{2\} & \text{if } i \in \text{AR} \\ \{1, 2\} & \text{if } i \in \text{AE} \end{cases}$$

PC : Set of pairs of tasks and predetermined station for positional constraints.

PZ : Set of pairs of tasks for positive zoning constraints.

NZ : Set of pairs of tasks for negative zoning constraints.

SC : Set of pair of tasks for synchronism constraints.

$\tilde{t}_i$: Stochastic operation time of task i .

μ_i : The mean of operation time of task i .

σ_i^2 : The variance of operation time of task i .

α : The pre-determined confidence level and the probability that a task is finished within the cycle time must be above 0.5.

ψ : A large enough positive number.

CT : Cycle time.

ε : A small positive number, and $0 < \varepsilon \leq 1 / (2 \times nt + 1)$.

Decision variables:

$\tilde{t}(M_j)$: Finishing time of mated-station j .

$$x_{ijk} = \begin{cases} 1, & \text{if task } i \text{ is assigned to station } (j, k); \\ 0, & \text{otherwise;} \end{cases} \quad \text{for } \forall i \in I, j \in J, k \in K(i)$$

$$F_j = \begin{cases} 1, & \text{if both sides of mated-station } j \text{ are utilized;} \\ 0, & \text{otherwise;} \end{cases} \quad \text{for } \forall j \in J$$

$$G_j = \begin{cases} 1, & \text{if only one side of mated-station } j \text{ is utilized;} \\ 0, & \text{otherwise;} \end{cases} \quad \text{for } \forall j \in J$$

$$U_{jk} = \begin{cases} 1, & \text{if station } (j, k) \text{ is utilized;} \\ 0, & \text{otherwise;} \end{cases} \quad \text{for } \forall j \in J, k = 1, 2$$

$$z_{ip} = \begin{cases} 1, & \text{if task } i \text{ is assigned earlier than task } p \text{ in the same station;} \\ 0, & \text{if task } p \text{ is assigned earlier than task } i \text{ in the same station;} \end{cases} \quad \text{for } \forall i \in I, p \in \{r \mid r \in I - (P_a(i) \cup S_a(i) \cup C(i)) \text{ and } i < r\}$$

3.3 Mathematical formulation of STALB problem with multiple constraints

Based on the mathematical model for TALB problem by [Özcan \(2010\)](#) and [Chiang et al. \(2015\)](#), the mathematical model for STALB problem with multiple constraints is given as follows. Two objectives of minimizing the number of mated-station and the number of stations are optimized simultaneously. Since it is a multi-objective problem, the linear weighting method is adopted to transfer the two objectives into only one.

$$\text{Min} \sum_{j \in J} (F_j + G_j) + \varepsilon \cdot \sum_{j \in J} \sum_{k=1,2} U_{jk} \quad (1)$$

s.t.

$$\sum_{j \in J} \sum_{k \in K(i)} x_{ijk} = 1 \quad \forall i \in I \quad (2)$$

$$\sum_{g \in J} \sum_{k \in K(h)} g \cdot x_{hgk} \leq \sum_{j \in J} \sum_{k \in K(i)} j \cdot x_{ijk} \quad \forall i \in I - P_0, h \in P(i) \quad (3)$$

$$\tilde{t}_i^f - \tilde{t}_h^f + \psi(1 - \sum_{k \in K(h)} x_{hjk}) + \psi(1 - \sum_{k \in K(i)} x_{ijk}) \geq \tilde{t}_h \quad \forall i \in I - P_0, h \in P(i), j \in J \quad (4)$$

$$\tilde{t}_p^f - \tilde{t}_i^f + \psi(1 - x_{ijk}) + \psi(1 - x_{pjk}) + \psi(1 - z_{ip}) \geq \tilde{t}_p; \quad \forall i \in I, \quad (5)$$

$$p \in \{r \mid r \in I - (P_a(i) \cup S_a(i) \cup C(i)) \text{ and } i < r\}, j \in J, k \in K(i) \cap K(p)$$

$$\tilde{t}_i^f - \tilde{t}_p^f + \psi(1 - x_{ijk}) + \psi(1 - x_{pjk}) + \psi \cdot z_{ip} \geq \tilde{t}_i; \quad \forall i \in I, \quad (6)$$

$$p \in \{r \mid r \in I - (P_a(i) \cup S_a(i) \cup C(i)) \text{ and } i < r\}, j \in J, k \in K(i) \cap K(p)$$

$$x_{ijk} = 1 \quad \forall (i, (j, k)) \in \text{PC} \quad (7)$$

$$x_{ijk} - x_{hjk} = 0 \quad \forall (i, h) \in \text{PZ}, k \in K(i) \cap K(h) \quad (8)$$

$$\sum_{k \in K(i)} x_{ijk} + \sum_{k \in K(h)} x_{hjk} \leq 1 \quad \forall (i, h) \in \text{NZ}, j \in J \quad (9)$$

$$x_{ijf} - x_{hjk} = 0 \quad \forall (i, h) \in \text{SC}, k \in K(h), f \in K(i), k \neq f \quad (10)$$

$$\tilde{t}_i^f - \tilde{t}_i = \tilde{t}_h^f - \tilde{t}_h \quad \forall (i, h) \in \text{SC} \quad (11)$$

$$\sum_{i \in I} x_{ijk} - \psi \cdot U_{jk} \leq 0 \quad \text{for } \forall j \in J, k \in K(i) \quad (12)$$

$$\sum_{k=1,2} U_{jk} - 2 \cdot F_j - G_j = 0 \quad \text{for } \forall j \in J \quad (13)$$

Objective function (1) minimizes the number of mated-station and the number of stations. Constraint (2) is the occurrence constraint and constraints (3-4) are the precedence constraint. Constraints (5-6) guarantee that all the tasks allocated to one station should be completed sequentially. If task i is assigned earlier than task p in the same station, Eq. (5) becomes active and it turns into $\tilde{t}_p^f - \tilde{t}_i^f \geq \tilde{t}_p$.

Otherwise, Eq. (6) becomes $\tilde{t}_i^f - \tilde{t}_p^f \geq \tilde{t}_i$ when task i is assigned later than task p . Constraint (7) deals with positional constraint. Constraints (8-9) handle positive zoning constraint and the negative zoning constraint respectively. Constraints (10-11) guarantee that any two tasks in synchronism constraints are operated at both sides of a mated-station with the same start time. Constraints (12-13) are the station constraint and mated-station constraint and they determine the number of mated-stations and the number of stations.

The completion probability constraint for the stochastic STALB is much more complex than the cycle time constraint for deterministic TALB, which depends on the treatment of uncompleted tasks (Chiang et al., 2015). If uncompleted tasks are completed offline and don't affect the following mated-station, then the completion probability constraint is formulated in expression (14). In expression (14), all the mated-station can be considered separately. If uncompleted tasks are performed online and lead to the breakdown of the latter mated-station, we utilize the expression (15) to deal with the completion probability constraint.

$$\Pr\{\tilde{t}(M_j) \leq CT\} \geq \alpha \quad \forall j \in J \quad (14)$$

$$\prod_j \Pr\{\tilde{t}(M_j) \leq CT\} \geq \alpha \quad (15)$$

Due to the utilization of both sides, the joint completion probability of tasks within a same mated-station is necessary. Based on the proposition 1 in Chiang et al. (2015) expressed with expression (16), it is easily to find that when two tasks i, j are allocated to two sides of a mated-station, the joint probability of these two tasks being finished within cycle time is larger than that when they are allocated to a same station.

$$\Phi\left(\frac{CT - \mu_i}{\sqrt{\sigma_i^2}}\right) \cdot \Phi\left(\frac{CT - \mu_h}{\sqrt{\sigma_h^2}}\right) \geq \Phi\left(\frac{CT - \mu_i - \mu_h}{\sqrt{\sigma_i^2 + \sigma_h^2}}\right) \quad (16)$$

Then, we can further obtain some sufficient conditions and necessary conditions for the existence of a feasible solution. We first consider the situation of that the uncompleted tasks are completed offline. One necessary condition for expression (14) is described with expression (17), in which the tasks within a mated-station are considered independently. One sufficient condition is described with expression (18), in which the tasks are supposed to be allocated to only one side of a mated-station. Since the mated-station times can be regarded as independent, the above necessary and sufficient conditions can be applied to the situation that uncompleted tasks are performed online by utilizing $\prod_j^* (\forall j \in J)$. Furthermore, one sufficient condition for expression (15) can be obtained in expression (19).

$$\prod_{i \in \{h | \sum_k x_{hjk} = 1\}} \Phi\left(\frac{CT - \mu_i}{\sqrt{\sigma_i^2}}\right) \geq \alpha \quad \forall j \in J \quad (17)$$

$$\Phi\left(\frac{CT - \sum_{i \in \{h | \sum_k x_{hjk} = 1\}} \mu_i}{\sqrt{\sum_{i \in \{h | \sum_k x_{hjk} = 1\}} \sigma_i^2}}\right) \geq \alpha \quad \forall j \in J \quad (18)$$

$$\Pr\{\tilde{t}(M_j) \leq CT\} \geq \sqrt[m]{\alpha} \quad \forall j \in J \quad (19)$$

4. The proposed algorithm for the stochastic two-sided assembly line balancing

In this section, a hybrid teaching-learning-based optimization (HTLBO) is presented, in which a priority-based decoding procedure is developed to deal with multiple constraints, a crossover operator is applied to enhance global search and a variable neighborhood search is utilized for strong local search. The components of the proposed HTLBO algorithm are introduced in detail as follows.

4.1 Introduction to TLBO

TLBO is a population-based algorithm and the main idea behind it is the simulation of a classical school learning process. The advantages of TLBO algorithm such as ease of implementation, rapidness of obtaining solutions and robustness are presented in the literature. TLBO seems to be a rising star amongst a number of meta-heuristics with relatively competitive performances. TLBO has showed its superiority for constrained mechanical design optimization problems (Rao et al., 2011), clustering data (Satapathy and Naik, 2011), addressing continuous non-linear large scale problems (Rao et al., 2012) and identifying the location of automatic voltage regulators in distribution systems (Niknam et al., 2012). Consequently, this algorithm is frequently used to solve discrete optimization problems found on the assembly line balancing (Li et al., 2014), and flow shop scheduling and job shop scheduling (Baykasoğlu et al., 2014).

TLBO has two phases: the teacher phase and the learner phase. During the teacher phase, the best solution is selected as the teacher, and the rest of the solutions called students try to improve themselves by learning from the teacher. Therefore, the solutions are improved with the following equation:

$$X_{new,i} = X_{old,i} + r \cdot (X_{teacher} - (T_F \cdot X_{mean})) \quad (20)$$

where parameter r ranges between 0 and 1. T_F is the teaching factor which decides the value of mean to be changed ($T_F = \text{round}(1 + \text{rand}(0,1))$), and X_{mean} is the current mean value of the individuals.

During the learner phase, students improve themselves by interaction with the others. An individual X_i chooses another individual X_{ii} randomly to learn in two ways as shown in the following expression, so that it can involve from X_i into $X_{new,i}$.

$$X_{new,i} = \begin{cases} X_i + r \cdot (X_{ii} - X_i), & X_{ii} \text{ is better than } X_i \\ X_i + r \cdot (X_i - X_{ii}), & \text{Otherwise} \end{cases} \quad (21)$$

4.2 Random and heuristic initialization

Since the nature of STALB problems is discrete, TLBO which is initially designed for continuous optimization problems cannot be applied to STALB problems directly. Therefore, the random-keys method (Bean, 1994) is adopted. First, each solution X_i is described with random real numbers, $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n}]$. Then, the largest order value (LOV) rule is hired to get the task permutation $\pi_i = [\pi_{i,1}, \pi_{i,2}, \dots, \pi_{i,n}]$. This random initialization process is illustrated with the 12-task problem (Kim et al. 2000) in Fig.2 and Table 1. According to LOV rule, $x_{i,10}$ is selected at first and ranks No.1 in task sequence since it has the largest value. Then the task $x_{i,11}$ is selected since it has the second largest value. By repeating this process, we can get the sequence of each task and the final task permutation.

Table 1 Task permutation of individual X_i

Task	1	2	3	4	5	6	7	8	9	10	11	12
Position	0.11	0.03	0.28	0.71	0.42	0.46	0.52	0.23	0.44	0.97	0.84	0.59
Task sequence	11	12	9	3	8	6	5	10	7	1	2	4

Task permutation	10	11	4	12	7	6	9	5	3	8	1	2

Fig.2 Precedence diagram of 12-tasks

The above 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, the operation times t_i by Tonge (1960) and the number of immediate successors IF_i by Helgeson and Birnie (1961), have shown promising results for classical one-sided assembly line balancing problems. Therefore, these two heuristic factors are employed to improve the qualities of initial solutions of STALB problems. Based on the two heuristic factors and their weighted modulus, λ_1 and λ_2 , the synthesis weight of each task can be calculated by Eq.(22), where $\lambda_1 + \lambda_2 = 1$. The task with the largest synthesis weight should be selected at first, and we can also get task permutation just like random initialization. This paper proposes the following initialization procedure: the first $p\%$ individuals among the population are initialized with heuristic factors, and the remained individuals are initialized with random real numbers. This initialization approach makes sure that at least $p\%$ individuals have acceptable quality.

$$w_i = \lambda_1 \cdot t_i / \sum_{j \in I} t_j + \lambda_2 \cdot IF_i / \sum_{j \in I} IF_j \quad \forall i \in I \quad (22)$$

4.3 Priority-based decoding procedure

This part introduces the priority-based decoding procedure to deal with the completion probability constraint, precedence constraint, direction constraint, positional constraint, zoning constraint and synchronism constraints by adjusting the priority value successively. We first introduce the initial priority values of tasks and then propose the updating procedure of priority values of tasks to deal with all the constraints.

In two-sided assembly line balancing problems, each task can be allocated to the given side or either side of the mated-station. Thus, this paper considers the priority value of task i on both sides, (i.e. $PL[i]$ for the left side and $PR[i]$ for the right side). This part only describes the priority value $PL[i]$ of allocating task i to the left side for simplification and the priority value $PR[i]$ can be obtained with the same method. The initial $PL[i]$ is calculated with expression (23), where TS_i is the sequence of task i in the above task permutation of section 4.2. Obviously, the earlier the task is in the permutation, the larger the priority value is. Moreover, the initial priority value of each task is less than 1.

$$PL_i = 1 - \frac{2TS_i - 1}{2nt} \quad TS_i \in \{1, 2, \dots, nt\} \quad (23)$$

After obtaining initial priority values, the decoding procedure can be applied with Step 1-Step 8. After the execution of the priority adjustment procedure, two sets are obtained: (1) $PL[i] > 3$, when completion probability constraint, direction constraints, precedence constraint, negative zoning constraint are satisfied perfectly; (2) $PL[i] < 3$, if any one of the above constraints is violated. The threshold $PL[i] = 3$ is applied to decide whether a task is assignable. It should be noted that the positional constraint, the positive zoning constraint and synchronism constraint can be violated even when $PL[i] > 3$.

Step 1: Update priority value with $PL[i] = PL[i] + M$ ($M = \{-100, 1\}$) for completion probability constraint. If the completion probability constraint (expression (14) or (15)) is satisfied, update priority value with $PL[i] \leftarrow PL[i] + 1$. Otherwise, $PL[i] \leftarrow PL[i] - 100$ and go to **Step 8**.

Step 2: Update priority value with $PL[i] = PL[i] + M$ ($M = \{-100, 1\}$) for precedence constraint. If the predecessors of task i have been allocated, then $PL[i] \leftarrow PL[i] + 1$. Otherwise, $PL[i] \leftarrow PL[i] - 100$ and go to **Step 8**.

Step 3: Update priority value with $PL[i] = PL[i] + TS[i][j][k]$ ($k = 1$) for direction constraint. The increment matrix $TS[i][j][k]$ corresponds the preferred direction of task i . If task i can be allocated to the left side, then $TS[i][j][k] \leftarrow 1$ and thus $PL[i] = PL[i] + 1$. If the direction of task i is right side, then $PL[i] = PL[i] + 0$. After executing the above procedure, execute **Step 4** if $PL[i] > 3$ or go **Step 8** if $PL[i] < 3$.

Step 4: Update priority value with $PL[i] = PL[i] + TS[i][j][k]$ ($k = 1$) for positional constraints. Since the tasks in positional constraints are prohibited on other stations except the predetermined (j^*, k^*), the value of $TS[i][j][k]$ is reset to be 2 for the predetermined station to advocate its priority ($TS[i][j][k] = 2 \quad \forall k = k^*, j = j^*$), 1 for the latter stations to guarantee a solution ($TS[i][j][k] = 1 \quad \forall k = k^*, j = j^*$), and 0 for the former station to reduce search space ($TS[i][j][k] = 0 \quad \forall k = k^*, j > j^*$).

Step 5: Update priority value with $PL[i] = PL[i] + \sum_{k=1}^{NLT} TT[i][LT_k] + \sum_{k=1}^{NTR} \min(0, TT[i][RT_k])$ for zoning constraint. NLT/NTR is the total number of all tasks that have been assigned to left/ right side of the

current mated-station. And, LT_k/RT_k is the k -th task allocated to left/ right side of the mated-station. (1) For positive zoning constraint, the value of increment matrix $TT[i][h]$ is set to 1 ($TT[i][h]=1, TT[h][i]=1 \forall (i,h) \in \text{PZ}$) so as to promote the priority of task i . (2) For negative zoning constraint, the value comes to be -100 ($TT[i][h]=-100, TT[h][i]=-100 \forall (i,h) \in \text{NZ}$). This big negative number is used to make sure that two tasks in negative zoning constraint can never be assigned to the same mated-station. (3) If none of these two constraints are involved, the value is 0 ($TT[i][h]=0, TT[h][i]=0 \forall ((i,h) \notin \text{PZ}) \cap ((i,h) \notin \text{NZ})$). After executing the above procedure, task i cannot be allocated and go to **Step 8** if $PL[i]<0$.

Step 6: Update priority value with $PL[i]=PL[i]+M$ ($M=\{-100, 10\}$) for synchronism constraints. If the other task h in synchronism constraint has been assigned to the right side, $PL[i]=PL[i]+10$. If the other task h in synchronism constraint has not been assigned and it can be allocated to the right side ($PR[h]>3$), $PL[i]=PL[i]+10$. If the other task h in synchronism constraint has not been assigned and it cannot be allocated to the right side ($PR[h]<3$), $L[i]=PL[i]-100$ and **Step 8**.

Step 7: Update priority value with $PL[i]=PL[i]+M$ ($M=\{0,1\}$) for the sequence-dependent circumstance. For the sequence-dependent circumstance, the idle time related to sequence-dependent finishing time of tasks is expected to be reduced. Therefore, $M=1$, provided that task i can be started at the earliest possible time of the left side of the current mated-station, namely $STL[i]=ETL$. Here, $STL[i]$ and ETL is the possible start time of the task i and the station respectively. Otherwise, $M=0$. As for the sequence-dependent circumstance, two scenarios exist. $PL[i]$ is increased by 1, if task i can be started at the earliest possible time of the station; otherwise, $PL[i]$ remains unchanged.

Step 8: Try Step 1-7 for another task or terminate when the priorities of all tasks have been updated.

To get a feasible solution, it is a priority to find out all candidate tasks that can be assigned to the left/right side of the current mated-station, by comparing the priority value of all tasks with the threshold 3. If none of the remained tasks can be allocated to the current mated-station, a new mated-station is opened. As long as one task can be allocated, the preferred task and side should be selected. When all tasks can be assigned to only one side, this side is chosen. When all tasks can be allocated to both sides, the side with earlier start time is chosen. If the start times of both sides are equal, choose one side randomly. Once the side is decided, the task with the highest priority is chosen and assigned. In order to reduce the number of stations, all the tasks on the last mated-station are reassigned under the following conditions: (1) both sides of the last mated-station are used; (2) the operation directions of these tasks are compatible, i.e. L and E, or R and E; (3) the total operation time of these tasks satisfies completion probability constraint; (4) the multiple constraints are not violated if all its tasks are assigned to only one station. The process of decoding procedure is explained in Fig.3, where CTL/ CTR is the number of candidate tasks that can be assigned to the left/right side of the current mated-station and RN is a random number which can be either 0 or 1 ($RN = \text{round}(\text{rand}(0,1))$).

Fig.3 Constraint handling-based decoding

4.4 Enhancement of global search by crossover operator

The original TLBO gradually updates the population based on the random-keys method, which can generate differentiated solutions. But the new solutions may lose some efficient “blocks” of task permutation. Therefore, the crossover in genetic algorithm is integrated with the learner phase in TLBO aiming at preserving efficient “blocks” of task permutation in order to increase the search speed and enhance global search.

The crossover generates new offspring by exchanging the contiguous sections of the chromosomes of parent solutions. The offspring chromosomes inherit partial features from their parents. The one-point crossover and two-point crossover are both applied in this paper. In the one-point crossover, the position is randomly selected to cut each parent into two parts: front (F) and back (B). New offspring are created by swapping the back selection of the parent’s chromosomes. In the two-point crossover, the positions are randomly selected to cut each parent into three parts: front (F), middle (M) and back (B). New offspring are produced by swapping the middle selection of the parent’s chromosomes. But when the diversity of the population is not wide enough, then one-point crossover is not effective enough to generate new offspring embodying great difference with their parents, especially for large-size problems. Thus, a two-point crossover proposed by Akpinar and Bayhan (2011) is also adopted to encourage population diversity and promote the global search.

4.5 Local search by variable neighborhood search

Fig.4 Seven neighborhood operators

Since a systematic change of neighborhood is helpful in increasing the probability of finding a better solution (Mladenovic and Hansen, 1997), the variable neighborhood search (VNS) is employed to enhance the local search ability of the TLBO. 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 Fig. 4. Note that, VNS is hired in each iteration to improve the diversity of solutions and avoid being trapped in a local optimum. And the procedure for utilizing seven neighborhood operators is shown as follows:

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, $x \leftarrow x'$ and set $k \leftarrow 1$; otherwise, set $k \leftarrow k + 1$ when $k < k_{max}$, or $k \leftarrow 1$ when $k = k_{max}$;

Step 4: If the termination criterion is satisfied, stop this process; otherwise, go to Step 2.

4.6 HTLBO procedure

As for a hybrid algorithm, the balance between the exploration and exploitation is a main factor of performance evaluation. The proposed HTLBO includes three parts: the TLBO, the crossover operator and the VNS, as shown in Fig.5. The TLBO and crossover operator cooperate so as to enhance the global search, and VNS based on seven neighborhood operators are employed to enhance 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 it may lose some efficient “blocks” of task permutation. Besides, 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.

Note that the positional constraint, the positive zoning constraint and synchronism constraint can be violated. Thus, corresponding penalty is considered in fitness function with expression (24), where w_p , w_{pz} and w_{sc} are the penalty coefficients of the number of violated positional constraint (np), violated positive zoning constraint (npz), and violated synchronism constraint (nsc) respectively.

$$\sum_{j \in J} (F_j + G_j) + \varepsilon \cdot \sum_{j \in J} \sum_{k=1,2} U_{jk} + w_p \cdot np + w_{pz} \cdot npz + w_{sc} \cdot nsc \quad (24)$$

Fig.5 Main body of the hybrid TLBO

5. Experimental results

5.1 Experimental design

All the benchmark problems from literatures about the two-sided assembly line balancing are solved in this paper. Among them, P9, P12, and P24 are from Kim et al. (2000), P16, P65 and P205 are from Lee et al. (2001), and P148 are taken from Bartholdi (1993) and modified by Lee et al. (2001). The additional constraints are from Yuan et al. (2015). For STALB problems, the means of tasks and the task variances are from Özcan (2010). For STALB problems with multiple constraints, the means of tasks and the task variances are from Özcan (2010) and the additional constraints from Yuan et al. (2015). All the algorithms are programmed in C++ language of Microsoft Visual Studio 2012 and all the consequent tests are conducted on an Intel(R) Core2 personal computer with 2.33GHZ CPU and 3.036 GB RAM.

For STALB problems, the task variance is divided into two parts: low task variance when it is between zero and $(t_i/4)^2$, and high task variance when between zero and $(t_i/2)^2$ (Özcan, 2010; Chiang et al., 2015). The deterministic TALB problem is also considered as a specific stochastic condition with variance of 0.0. And three confidence levels are taken into account similar to Özcan, (2010) ($\alpha = \{0.9, 0.95, 0.975\}$). The Clark's approximation method is applied to approximate the completion probability when assigning tasks to mated-stations (Clark, 1961; Chiang et al., 2015). Due to the inaccuracy of the approximation (Chiang et al., 2015), we run the simulation 10,000 times for each new solution to calculate the completion probability and its fitness function takes a large penalty if the completion probability is larger than the confidence level.

The coefficients w_p , w_{pz} and w_{sc} in fitness function are set to 100.0, 100.0 and 100.0, respectively.

The searching process terminates when the computational time reaches $nt \times nt \times 10$ milliseconds for deterministic TALB problems and $nt \times nt \times 100$ milliseconds for STALB problems with initial experiments. The above computational time for STALB problems is much larger due to the simulation experiments. This expression ensures that the computational time increases with the increase of the problem size. In order to calibrate the hybrid algorithm, all the possible combinations of the following factors are tested.

- Population size: 4 sizes (40, 80, 120 and 160).
- Heuristic initialization percentage: 8 percentages (0%, 10%, 20%, 30%, 40%, 50%, 60% and 70%).
- Crossover type: 4 types (No crossover, OP, TP, OP_TP), implying no crossover, one-point crossover,

two-point crossover, single or two-point crossover operator chosen randomly respectively.

▪ VNS type: 2 types (No VNS and VNS).

All the factors lead to $4 \times 8 \times 4 \times 2 = 256$ different combinations and thus, 256 different algorithms. Each algorithm is tested on P205 (CT=1322, 1699) for ten times since the two cases with variance 0.0 (deterministic TALB problem) are the largest cases in all known benchmarks. After all the experiments are carried out, relative percentage deviation (RPD) is calculated with $RPD(\%) = 100 \times (Sol - LB) / LB$.

Where, Sol is the number of stations obtained via each algorithm and LB is the lower bound (Wu et al., 2008). The analysis of variance (ANOVA) is employed as a statistical method to analyze the experimental results. Figs.6 (a-b) depicts the effect of the crossover type and VNS.

Based on the results of ANOVA, the size of the population is set to 40. 50% of individuals are initialized by heuristic initialization and another 50% by random initialization. The two-point crossover operator is utilized to enhance the global search. Table 2 describes the variability of RPD considering all the above factors. Only one P -value is less than a desired α ($\alpha = 0.05$), namely the crossover type, which demonstrates its significant effect on RPD. However, if the no-crossover condition is removed, then the P -value of crossover is updated to 0.693, which proves the superiority of the proposed crossover and the robustness of the hybrid algorithm.

Fig.6 Means plots and 95 percent Tukey HSD confidence intervals for crossover operators and VNS

Table 2 Analysis of variance for RPD

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
Population size	2.32225	3	0.774084	0.42	0.7366
Initialization	1.4521	7	0.207443	0.11	0.9975
Crossover	28.6253	3	9.54176	5.21	0.0014
VNS	0.171096	1	0.171096	0.09	0.7598
RESIDUAL	4658.14	2545	1.83031		
TOTAL	4690.71	2559			

5.2 Comparison of the decoding schemes

To test the effectiveness of the proposed decoding procedure, the following experiments are performed. The proposed decoding is compared with three other reported decoding approaches: Decode-Özcan (Özcan and Toklu, 2009a), Decode-Baykasoglu (Baykasoglu and Dereli, 2008) and Decode-Yuan (Yuan et al., 2015). And these decoding procedures are tested with STALB problem with variance 0.0 (deterministic TALB problem) and STALB problem with low task variance and $\alpha = 0.9$ while satisfying constraints (14). All these decoding schemes are embedded into simulated annealing algorithm (SA) (Özcan, 2010), and the average RPD values for three large-size problems, P65, P148 and P205, are utilized for comparison.

Initial analysis of these values shows strong deviation from normality, which is necessary for parametric ANOVA technique, and thus we employ a non-parametric Friedman rank-based test (Friedman, 1937). The results for each problem are transformed and the best result is marked with a rank of 1 and the worst is marked with a rank of 4 for TALB problem. The analysis shows that the p -value is close to zero, which indicates that there is a statistically significant difference in the average ranks of different decoding schemes. Fig.7 (a) depicts the mean ranks for TALB problem and Fig.7 (b) depicts the mean ranks for STALB problems.

Both Fig. 7(a) and Fig. 7(b) show that the proposed priority-based decoding approach obtains distinguished results for STALB problems. In fact, the proposed decoding scheme outperforms the others in two aspects: (1) it reduces the idle times related to sequence-dependence of the tasks by

allocating the tasks which do not generate idle times. (2) It selects the side with larger capacity at first, as shown in Fig.3, which balances the workload on the two sides of a same mated-station.

(a) Average rank for TALB problem (b) Average rank for STALB problem with low task variance

Fig.7 Means plot of average rank for TALB problem with 95% confidence intervals

For STALB problems with multiple constraints, the proposed decoding is compared with decoding (Yuan et al., 2015), marked with Decode-Yuan. We also employ non-parametric test, and the analysis shows that the p-value is close to zero, which indicates that there is a statistically significant difference in the average ranks of different decoding schemes. Fig.8 (a) and Fig.8 (b) depict the means plot of average RPD values for TALB problem with multiple constraints and STALB problem with multiple constraints respectively. Fig. 8b shows that the proposed priority-based decoding approach obtains distinguished results on STALB problems with multiple constraints.

(a) Average rank for TALB problem (b) Average rank for STALB problem with low task variance

Fig.8 Means plot of average rank for TALB problem with multiple constraints with 95% confidence intervals

In fact, the proposed decoding inherits the above advantages and also adopts new techniques to tackle positional constraints and positive zoning constraints. As for handling positional constraints, Yuan et al. (2015) ignored the positional constraint in decoding scheme and only employed a cost function. In this research, the task is prohibited on the former mated-station and the priority of the task is boosted when the station is just the station in the positional constraint. This improvement provides more chance to find a solution satisfying the positional constraints. As for positive zoning constraints, while Baykasoglu and Dereli (2008) and Yuan et al. (2015) assigned tasks with positive zoning constraints together to one station, this study assigns tasks one by one and permits each task to be operated at the earliest possible time. This modification reduces the idle time related to sequence-dependent finishing time of the tasks. All experimental results demonstrate remarkable effectiveness and versatility of the priority-based decoding approach by reducing idle times, balancing workloads, employing new techniques to tackle positional constraints and positive zoning constraints.

5.3 Comparison of the algorithms on STALB problems with multiple constraints

To test the performance of the proposed HTLBO, it is compared with other algorithms: genetic algorithm (GA) (Kim et al., 2000), tabu search algorithm (TS) (Özcan and Toklu, 2009a), ant colony-based heuristic (ACO) algorithm (Baykasoglu and Dereli.,2008), ant colony optimization

algorithm (2-ANTBAL) (Simaria and Vilarinho, 2009), bee colony intelligence (BA) (Özbakir et al., 2011), late acceptance hill-climbing algorithm (LAHC) (Yuan et al., 2015), simulated annealing algorithm (SA) (Khorasanian et al., 2013), TLBO (Li et al., 2014), variable neighborhood search (VNS) (Mladenovic and Hansen, 1997) and improved TLBO (ITLBO) (Li et al., 2014). All the algorithms are recoded with the priority-based decoding approach, and all the codes and all the parameters of the proposed algorithms are available upon request.

We test the performances of these algorithms on STALB problem with variance 0.0 and STALB problems with low or high variance and three confidence levels. Each case is running for twenty times and the average results of the number of stations are utilized for comparison. Notice that these algorithms always find similar results for small-size problems, and thus we focus on the average number of stations for large-size problems. All the experimental results are transferred into relative percentage deviation and we utilize both ANOVA technique and non-parametric Friedman rank-based test (Friedman, 1937) since initial analysis shows small deviation from normality. We first show the comparison of different algorithms for STALB problem with variance 0.0 plotted in Fig.9 (a-b). The average RPD values with 95.0% LSD intervals are depicted in Fig.9 (a) while the average ranks with 95% confidence intervals are plotted in Fig.9 (b). In Fig.9 (a), the average RPD value for HTLBO is less than 1.5, while the average RPD values for GA, TS, BA, LAHC, SA and VNS are larger than 2.0. In Fig. 9 (b), the average rank for HTLBO is next to 4.5, while average ranks for GA, TS, BA, LAHC, SA and VNS are larger than 6.0. Based on the above analysis, it is reasonable to say that HTLBO obtains distinguished performance for both the average RPD value and average rank.

Fig.9 Comparison of average numbers of stations for large-size problems

As for STALB problems with low or high deviation and three confidence levels, similar results are obtained for the average number of stations and the HTLBO obtains the peak performance. In fact, only the completion probability constraint in the decoding procedure is modified and thus the algorithms can obtain similar results for different cases. For simplification, we only present the best results of HTLBO and SA for the STALB problems while satisfying the constraint (14). Note that the SA algorithm is only one among the above method which has been applied to STALB problems and the results by Özcan (2010) needs to be adjusted due to the joint probability of tasks within a same mated-station.

Tables 3 and 4 present the best number of stations by the two algorithms on the large-size problems since the same results can be obtained for small-size problems. The low bound of the number of stations is calculated similar to that in Özcan (2010) and Chiang et al., 2015. And the SA is recoded with the mathematical model in Section 3.3. It is obvious that the proposed HTLBO algorithm outperforms SA regarding to the best solution. Under the condition of low task variance in Table 3, 4 cases for P148 and 7 cases for P205 are outperformed by the proposed HTLBO. Under the condition of high task variance in Table 4, 2 cases for P65, 10 cases for P148, and 2 cases for P205 are outperformed by the proposed HTLBO. Meanwhile, the number of stations with high task variance is larger than that of those with low task variance when comparing the data from Tables 3 and 4, which further proves the cost of handling stochastic task times.

Table 3 Computational results for STALB problems-low task variance

Problem	Cycle time	LB			SA			Hybrid TLBO		
		$\Phi^{-1}(\alpha)$			$\Phi^{-1}(\alpha)$			$\Phi^{-1}(\alpha)$		
		1.28	1.645	1.96	1.28	1.645	1.96	1.28	1.645	1.96
P65	326	17	*	*	19	*	*	19	*	*
	381	14	14	15	17	17	17	17	17	17
	435	13	13	13	14	15	15	14	15	15
	490	11	11	11	13	13	13	13	13	13
P148	544	10	10	10	11	11	12	11	11	12
	255	21	21	21	24	25	26	24	24	25
	306	18	18	18	20	21	21	20	20	21
	357	15	15	15	17	17	18	17	17	18
	408	13	13	14	15	15	15	15	15	15
	459	12	12	12	13	14	14	13	13	14
P205	510	11	11	11	12	12	12	12	12	12
	1133	22	22	*	26	26	*	25	26	*
	1322	19	19	19	22	22	23	21	22	22
	1510	16	16	16	18	20	20	18	19	19
	1699	15	15	15	16	17	18	16	17	17
	1888	13	13	13	15	16	16	15	15	16
	2077	12	12	12	13	14	14	13	14	14
	2266	11	11	11	12	12	13	12	12	13
	2454	10	10	10	12	12	12	12	12	12
	2643	10	10	10	10	11	11	10	11	11
	2832	9	9	9	10	10	10	10	10	10

Table 4 Computational results for STALB problems-high task variance

Problem	Cycle time	LB			SA			Hybrid TLBO		
		$\Phi^{-1}(\alpha)$			$\Phi^{-1}(\alpha)$			$\Phi^{-1}(\alpha)$		
		1.28	1.645	1.96	1.28	1.645	1.96	1.28	1.645	1.96
P65	326	*	*	*	*	*	*	*	*	*
	381	15	*	*	19	*	*	19	*	*
	435	13	13	13	17	18	19	16	17	19
	490	12	12	12	15	15	16	15	15	16
	544	11	11	11	13	13	14	13	13	14
P148	255	22	22	*	28	29	*	27	28	*
	306	18	18	18	23	24	25	22	23	24
	357	16	16	16	19	20	21	19	19	20
	408	14	14	14	16	17	18	16	17	17
	459	12	12	12	14	15	16	14	15	15
	510	11	11	11	13	14	14	13	13	14
P205	1133	*	*	*	*	*	*	*	*	*
	1322	*	*	*	*	*	*	*	*	*
	1510	17	*	*	21	*	*	21	*	*
	1699	15	15	*	18	20	*	18	19	*
	1888	13	14	14	16	18	18	16	18	18
	2077	12	12	13	15	16	16	15	16	16
	2266	11	11	12	14	14	15	14	14	15
	2454	10	11	11	13	13	14	13	13	14
	2643	10	10	10	12	12	13	12	12	12
	2832	9	9	9	11	12	12	11	12	12

As for STALB problems with multiple constraints (STALB-MC), we mainly present the results for STALB problem with low variance and confidence level 0.9 while satisfying the constraint (14). The best results by HTLBO are reported, and the results of the STALB and the deterministic TALB with multiple constraints (TALB-MC) are also reported to show the influence of multiple constraints and stochastic task times respectively. Table 5 shows that in most cases the cost for dealing with the stochastic task times is greater than that for multiple constraints, although the task variance is low, which explains why the methodology of handling uncertainties is urgent in workplace nowadays. Meanwhile, 19 out of 26 cases get the same results as those of STALB or TALB with multiple constraints.

Table 5 Computational results for STALB problems with multiple constraints

Problem	Cycle time	STALB	TALB-MC	STALB -MC
		nm[ns]	nm[ns]	nm[ns]
P9	5	3[5]	2[4]	3[5]
	6	2[4]	2[3]	2[4]
P12	7	3[5]	3[5]	3[5]
	8	3[5]	2[4]	3[5]
P24	25	4[7]	3[6]	4[7]
	30	3[6]	3[5]	3[6]
	35	3[5]	3[5]	3[5]
	40	2[4]	2[4]	3[5]
P65	381	9[17]	7[14]	9[17]
	435	7[14]	7[13]	7[14]
	490	7[13]	6[11]	6[13]
	544	6[11]	6[11]	6[11]
P148	255	12[24]	11[21]	12[24]
	306	10[20]	9[18]	10[20]
	357	8[17]	8[15]	9[17]
	408	7[15]	7[13]	8[15]
	459	7[13]	6[12]	7[13]
	510	6[12]	6[11]	6[12]
P205	1510	9[18]	9[17]	10[19]
	1699	8[16]	8[16]	9[17]
	1888	7[15]	7[14]	8[16]
	2077	7[13]	7[14]	7[14]
	2266	6[12]	6[12]	7[13]
	2454	6[12]	6[12]	6[12]
	2643	5[10]	6[11]	6[12]
	2832	5[10]	5[10]	6[11]

The above computational results exhibit outstanding performance of the proposed HTLBO algorithm for both STALB problems and STALB problems with multiple constraints. For the STALB problem, the HTLBO also finds many new upper bounds. Moreover, comparison among the STALB problems with multiple constraints proves that multiple constraints and stochastic task times are both significant factors in production, and neither of them can be ignored while balancing workload along two-sided assembly lines. In summary, the HTLBO shows promising results for solving STALB with multiple constraints, especially for large-size cases.

6. Conclusions and future research

This study addresses the two-sided assembly line balancing problem with stochastic task times and multiple constraints. Considering the NP-hardness of this problem, a hybrid TLBO algorithm is proposed, which has the following advantages: (1) the teacher phase in TLBO guarantees the fast convergence to the best solution; (2) the learner phase based on random-keys method and crossover operator aiming at preserving some “blocks” of task permutation generates a differentiated and high-quality population; and (3) VNS based on seven different neighborhood operators provides a strong local search. Consequently, the intensification and diversification of the proposed algorithm are well balanced.

Especially, this study develops a novel priority-based decoding approach. This approach adopts four improvements: (1) it selects the side with larger capacity which balances the workload on the two sides of a same mated-station; (2) it allocates the tasks which do not generate idle times at first and hence eliminates the idle times related to sequence-dependence of the tasks; (3) it prohibits the tasks from being allocated to the former mated-station in the positional constraint to increase the possibility of finding feasible solutions; (4) it assigns the tasks with positive zoning constraints one by one and permits each task to be operated at the earliest possible time to further reduce idle times related to sequence-dependence of the tasks. Thus, this decoding approach shows the advantages of reducing idle times related to sequence-dependence of the tasks and hence improving the balance of the workload.

Series of experiments demonstrates the excellent performance of the proposed HTLBO algorithm. Experimental comparisons of different decoding schemes prove the effectiveness and versatility of the priority-based decoding approach for all the tested problems. And, comparisons among 11 algorithms demonstrate the outstanding performance from the proposed HTLBO. Additionally, the HTLBO also finds some new upper bounds for STALB problems.

Since the proposed priority-based decoding approach shows distinguished results, it can be applied

to other two-sided assembly line balancing problems, such as mixed-model two-sided assembly line balancing problem and two-sided assembly line balancing problem type II. Since stochastic factors show a great impact on the production on two-sided assembly lines, other uncertain factors such as unstable material supply, variable customer demands and machine breakdown should be further considered.

Acknowledgment

We acknowledge the support from the National Science Foundation of China under grants 50875190 and 51275366/E051005. The authors also thank the anonymous referees for their suggestions and comments to improve this paper.

References

- [1] Akpınar S., Bayhan G. M. A hybrid genetic algorithm for mixed-model assembly line balancing problem with parallel workstations and zoning constraints. *Engineering Applications of Artificial Intelligence* 2011; 24(3): 449-457.
- [2] Bartholdi J.J. Balancing two-sided assembly lines: a case study. *International Journal of Production Research* 1993; 31(10):2447-2461.
- [3] Baykasoglu A., Dereli T. Two-sided assembly line balancing using an ant-colony-based heuristic. *International Journal of Advanced Manufacturing Technology* 2006; 36:582–588.
- [4] Baykasoglu A., Hamzadayi A., Köse S.Y. Testing the performance of teaching-learning-based optimization (TLBO) algorithm on combinatorial problems: Flow shop and job shop scheduling cases. *Information Sciences* 2014; 276: 204-218.
- [5] Bean J. Genetics and random keys for sequencing and optimization. *ORSA Journal on Computing* 1994; 6 (2):154-160.
- [6] Chiang W-C, Urban T. L., Luo C. Balancing stochastic two-sided assembly lines. *International Journal of Production Research* 2015; doi:10.1080/00207543.2015.1029084.
- [7] Chutima P., Chinklai P. Multi-objective two-sided mixed-model assembly line balancing using particle swarm optimization with negative knowledge. *Computers & Industrial Engineering* 2012; 62(1):39-55.
- [8] Clark Charles E. The greatest of a finite set of random variables. *Operations Research* 1961; 9(2): 145–162.
- [9] Friedman M.. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association* 1937; 32:675-701.
- [10] Helgeson W. B., Birnie D. P. Assembly line balancing using the ranked positional weight technique. *Journal of Industrial Engineering* 1961;12(6): 394-398.
- [11] Hu X-F, W E-F, Jin Ye. A station-oriented enumerative algorithm for two-sided assembly line balancing. *European Journal of Operational Research* 2008; 186:435-440.
- [12] Hu X-F, Wu E-F, Bao J-S, Jin Ye. A branch-and-bound algorithm to minimize the line length of a two-sided assembly line. *European Journal of Operational Research* 2010; 206(3):703-707.
- [13] Khorasanian D., Hejazi S.R., Moslehi G. Two-sided assembly line balancing considering the relationships between tasks. *Computers & Industrial Engineering* 2013; 66(4): 1096-1105.
- [14] Kim Y.K., Kim Y., Kim Y.J. Two-sided assembly line balancing: a genetic algorithm approach, *Production Planning & Control* 2000; 11(1):44-53.
- [15] Kim Y. K., Song W. S., Kim J. H. A mathematical model and a genetic algorithm for two-sided assembly line balancing. *Computers & Operations Research* 2009; 36(3):853-865.
- [16] Lee T.O., Kim Y, Kim Y.K. Two-sided assembly line balancing to maximize work relatedness and slackness. *Computer & Industrial Engineering* 2001; 40(3):273-292.
- [17] Li D-S, Zhang C-Y, Shao X-Y , Lin W-W. A multi-objective TLBO algorithm for balancing two-sided assembly line with multiple constraints. *Journal of Intelligent Manufacturing* 2014; DOI 10.1007/s10845-014-0919-2.
- [18] Mladenovic N., Hansen P. Variable neighborhood search. *Computers & Operations Research* 1997; 24:1097-1100.
- [19] Niknam T., Azizipanah-Abarghooee R., Narimani M. R. A new multi-objective optimization approach based on TLBO for location of automatic voltage regulators in distribution systems. *Engineering Applications of Artificial Intelligence* 2012; 25(8), 1577-1588.
- [20] Özbakir L., Tapkan P. Bee colony intelligence in zone constrained two-sided assembly line balancing problem. *Expert Systems with Applications* 2011; 38(9), 11947-11957.
- [21] Özbakir L, Tapkan P. Balancing fuzzy multi-objective two-sided assembly lines via Bees Algorithm. *Journal of Intelligent and Fuzzy Systems* 2010; 21(5): 317-329.
- [22] Özcan U., Toklu B. A tabu search algorithm for two-sided assembly line balancing, *International Journal of Advanced Manufacturing Technology* 2009a; 43:822-829.
- [23] Özcan U., Toklu B. Balancing of mixed-model two-sided assembly lines. *Computers & Industrial Engineering* 2009b; 57:217-227.
- [24] Özcan, U. Balancing stochastic two-sided assembly lines: A chance-constrained, piecewise-linear, mixed integer program and a simulated annealing algorithm. *European Journal of Operational Research* 2010; 205(1):81-97.
- [25] Purnomo H. D., Wee, H-M, Rau H. Two-sided assembly lines balancing with assignment restrictions. *Mathematical and Computer Modeling* 2013; 57(1), 189-199.

- [26] Rao R.V., Savsani V.J., Vakharia D.P. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design* 2011; 43 (3):303-315.
- [27] Rao R.V., Savsani V.J., Vakharia D.P. Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. *Information Sciences* 2012; 183 (1):1-15.
- [28] Rao R.V., Patel V. Multi-objective optimization of heat exchangers using a modified teaching-learning-based optimization algorithm. *Applied Mathematical Modeling* 2013; 37(3), 1147-1162.
- [29] Satapathy S. C., Naik A. Data clustering based on teaching-learning-based optimization. Springer 2011; In *Swarm, Evolutionary, and Memetic Computing* (pp. 148-156).
- [30] Simaria A. S., Vilarinho P. M. 2-ANTBAL: An ant colony optimization algorithm for balancing two-sided assembly lines. *Computers & Industrial Engineering* 2009; 56(2), 489-506.
- [31] Tapkan P., Ozbakir L., Baykasoglu A. Modeling and solving constrained two-sided assembly line balancing problem via bee algorithms. *Applied Soft Computing* 2012a; 12(11): 3343-3355.
- [32] Tapkan P., Ozbakir L., Baykasoglu A. Bees algorithm for constrained fuzzy multi-objective two-sided assembly line balancing problem. *Optimization Letters* 2012b; 6(6):1039–1049.
- [33] Tuncel G., Aydin D. Two-sided assembly line balancing using teaching-learning-based optimization algorithm. *Computers & Industrial Engineering* 2014; 74: 291-299.
- [34] Tonge, F. M.. Summary of a heuristic line balancing procedure. *Management Science* 1960; 7(1): 21-42.
- [35] Wang B-X, Guan Z-L, Li D-S, Zhang C-Y, Chen L. Two-sided assembly line balancing with operator number and task constraints: a hybrid imperialist competitive algorithm. *The International Journal of Advanced Manufacturing Technology* 2014; 74(5-8): 791-805.
- [36] Wu E- F, Ye Jin, Bao J-S, & Hu X-F. A branch-and-bound algorithm for two-sided assembly line balancing. *International Journal of Advanced Manufacturing Technology* 2008; 39:1009-1015.
- [37] Yuan B., Zhang C-Y, Shao X-Y. A late acceptance hill-climbing algorithm for balancing two-sided assembly lines with multiple constraints. *Journal of Intelligent Manufacturing* 2015; 26(1):159-168.