J Ind Eng Int (2016) 12:485–497
DOI 10.1007/s40092-016-0158-6

ORIGINAL RESEARCH

A multi-objective genetic algorithm for a mixed-model assembly U-line balancing type-I problem considering human-related issues, training, and learning

Masoud Rabbani[1] · **Mona Montazeri**[1] · **Hamed Farrokhi-Asl**[2] · **Hamed Rafiei**[1]

Received: 29 March 2015 / Accepted: 7 June 2016 / Published online: 1 July 2016

Abstract Mixed-model assembly lines are increasingly accepted in many industrial environments to meet the growing trend of greater product variability, diversification of customer demands, and shorter life cycles. In this research, a new mathematical model is presented considering balancing a mixed-model U-line and human-related issues, simultaneously. The objective function consists of two separate components. The first part of the objective function is related to balance problem. In this part, objective functions are minimizing the cycle time, minimizing the number of workstations, and maximizing the line efficiencies. The second part is related to human issues and consists of hiring cost, firing cost, training cost, and salary. To solve the presented model, two well-known multi-objective evolutionary algorithms, namely non-dominated sorting genetic algorithm and multi-objective particle swarm optimization, have been used. A simple solution representation is provided in this paper to encode the solutions. Finally, the computational results are compared and analyzed.

Keywords Mixed-model assembly lines · U-shaped assembly lines · Learning and training effect · Human-related issues · Multi-objective

✉ Masoud Rabbani
mrabanu@ut.ac.ir

[1] School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran

[2] School of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

Introduction

An assembly line is a group of successive workstations, joined by a material handling system. In each workstation, a set of tasks are carried out using a predefined assembly process, in which the time required to carry out each task and a set of priority relations which determines the order of the tasks are defined. The current market is severely competitive and consumer-centric with high variety in demands. As a result of high cost to establish and maintain an assembly line, the manufacturers produce one model with various features or several different models on a single assembly line. In situations like this, the mixed-model assembly line balancing problem arises to smooth the production and decreases the cost. Mixed-model Assembly Line (MMAL) is a kind of production line, where a set of similar models of a product are assembled to respond to the diversity of customer's demands. There are two types of assembly line balancing problems. The purpose of type-I problems are minimizing the number of workstations. In this problem, the required production rate, assembly tasks, tasks times, and precedence requirements will be given. In type-II problems, the goal is to minimize the cycle time and maximize the production rate with fixed number of workstations or production employees. This study is mainly focused on the type-I problem, which wants to minimize the number of workstations.

U-type line balancing was first invented by Miltenburg and Wijngaard (1994). The U-type assembly line is an attractive substitute for assembly production systems from the time operators became multi-skilled by performing tasks defined on different parts of assembly line (Gökçen et al. 2005). The advantage of the U-type assembly line is the flexibility that it offers to choose an appropriate number of operators to satisfy demand changes (Aigbedo and Monden 1997).

Learning effect is another important factor at assembly lines in the time of the new product lunch, or start of production (Baloff 1971). The length of the learning stage has become an important performance indicator for a firm because of some common topics, such as shortened product life cycles, high innovation rates, and, therefore, more frequent product launches. Learning effect has to be considered in firms, because shorter learning stages enable firms to increase sales and, as a result, achieve more profits with the highest revenues, by the time, the new product reaches the market. Learning effects may occur by a highly repetitive execution of certain tasks. “A worker learns as he works; and the more often, he repeats an operation”. Andress (1954) mentioned, learning effects at assembly lines and overall for repetitive operations. According to aircraft construction, Wright (2012) described learning effects at assembly lines and overall for repetitive operations. He figured out that by making the cumulated output double, average construction costs per unit sunk by about 20 %. This observation was formalized as an inversely proportional relationship between unit costs and cumulated output called learning curve. After that, for assembly lines in different industries, the presence of significant learning effects was confirmed. Basically, in mixed-model U-shaped assembly lines, workers are capable of operating several tasks. As Park (1991) said, training, the process by which workers become multi-skilled, has been recognized as a tool for boosting production flexibility. The minimum introduction of worker cross-training has the most significant improvement from no cross-training, and the subsequent increase of the cross-training has a diminishing return. In this research for the first time, a new model is presented considering both line balancing and worker assignment simultaneously, considering human-related issues. Two meta-heuristic algorithms [i.e., multi-objective particle swarm optimization (MOPSO) and non-dominated sorting genetic algorithm (NSGA-II)] are used to solve the proposed bi-objective problem, and a simple method is applied to represent solutions.

The rest of the paper is organized as follows: in “Literature review”, the relevant literature is reviewed. In “Problem description”, the bi-objective problem, the objective function, and a mathematical model are presented. The methodology is described in “Methodology”, and the illustrative examples are presented in this section. In “Parameters tuning”, comparisons and discussion are brought. The study is finally ended by conclusions and future research in “Conclusion”.

Literature review

The existing competitive and consumer-centric market and the observed trend of diversification of customer demands and high fluctuations is an important subject that is worth studying. Firms should improve their performance for dealing with these pressures to meet the customers demand within a short delivery time and with the lowest possible cost. Mixed-model assembly lines are one of the most relevant production environments that deal with these problem. The assembly line balancing problem encompasses assigning tasks to an ordered sequence of stations, such that precedence relations among tasks should not be violated (Erel and Sarin 1998). A mixed-model assembly line is assembly line, in which some similar product type with some insignificant difference is assembled. Many attempts have been made to solve the assembly line balancing (ALB) problems using the exact solution methods, heuristics, and meta-heuristic approaches. Some comprehensive reviews of such studies have been done (Becker and Scholl 2006; Erel and Sarin 1998). Some researches solved the assembly line balancing problem using a ranked positional weight technique (Helgeson and Birnie 1961). Monden (2011) was concerned with the sequencing of assembly lines, such as considering the stability of parts usage rates. Kim et al. (2009) presented a mathematical formulation and a genetic algorithm for the ALB-II problem. Some practitioners presented a formal ALB-I problem, and they also developed a branch-and-bound algorithm to solve the problem (Wu et al. 2008). Erel and Gokcen (1999) proposed a study that was concerned with minimizing the task time for different models considering precedence constraints using shortest route formulation. A binary integer formulation for the mixed-model assembly line balancing problem is developed by Gökcen and Erel (1998). In another work, Gokcen and Erel (1997) extended a goal programming approach which was previously developed by Thomopoulos (1967), using a combined precedence diagram. Vilarinho and Simaria (2002) develop a two-stage heuristic method for balancing mixed-model assembly lines. The application of genetic algorithms (GA) for assembly line balancing has widely been considered in many studies. A genetic algorithm for type-II problems was presented by Anderson and Ferris (1994), and Leu et al. (1994) presented a GA-based approach to solve type-I problems with multiple objectives. Kim et al. (1996) presented a genetic algorithm for work load smoothing. In another study, a hybrid genetic algorithm approach to the assembly line planning problem was developed (Chen et al. 2002). There are only a few studies which use more than one meta-heuristic approach to solve their problem, but in this study, two meta-heuristic algorithms (i.e., MOPSO and NSGA-II) are used to solve the proposed bi-objective problem.

Many practitioners studied the mixed-model straight line assembly line balancing problem which has been reported in the literature (Erel and Gokcen 1999; McMullen and Frazier 1998; Simaria and Vilarinho 2004, 2009;

Thomopoulos 1967; Vilarinho and Simaria 2002). Simaria and Vilarinho (2009) proposed a mathematical programming model to formally describe the MMALB problem presenting an ant colony optimization algorithm. One of the effective factors for realizing the objectives of lean manufacturing is workforce planning. Several options of alternative production planning that can be applied for dealing with changing demand patterns, considering use of variable workforce, overtime, seasonal inventory, and planned backlogs have been developed by Hax and Candea (1984). Several classical LP models combining the production, manpower, and inventory-related trade-offs in each of the options mentioned above have been presented (Bhatnagar et al. 2003). Just-in-time (JIT) is able to adjust to changes in the external environment of the firm, because of several reasons, including efficient facility layouts and multi-functional workers (Monden 2011; Schonberger 1983). Japanese companies are operating with very low level of inventory and recognizing a high level of productivity using the just-in-time (JIT) manufacturing system which has the goal of continuously reducing and ultimately removing all forms of wastes (Ōno 1988). The replacement of the traditional straight lines with U-shaped production lines is one of the most important changes resulting from JIT implementation (Chiang and Urban 2006). Reducing the work in process inventory and wasted operator's movement, labor productivity improvement, material handling improvement, zero-defects campaign's implementation, and higher flexibility in workforce planning in the face of changing demand patterns (Monden 2011) are the main benefits of the U-line as compared to a straight line.

(In some reference, it is shown that one of the best applicable types of line is U-shape line and they illustrate that the benefits are impressive. The main characteristics in a U-shaped line are (Miltenburg and Wijngaard 1994): the U-line arranges machines around a U-shaped line in an operators work inside the U-line; U-lines are rebalanced periodically when production requirements change; the operators must be multi-skilled and versatile to do several different processes; it requires operators to walking, when setup times are negligible; U-lines are operated as mixed-model lines, where each station is able to produce any product in any cycle; when setup times are larger, multiple U-lines are formed and dedicated to different products. Miltenburg and Wijngaard (1994) have a comprehensive article in the subject of U-shaped production line. In his article, the benefits of U-shape line were mentioned, and by some statistic information, they are proved for all). There are several studies on line balancing problems. Most of them assumed that the time of tasks for repetition tasks is independent from learning of workers. A few researchers have examined the learning effect on assembly line balancing problems (Chakravarty and Shtub 1988; Cohen and Ezey Dar-El 1998; Cohen et al. 2006). Learning can play a considerable role in manufacturing environments and there are many empirical studies that have proven learning effects (Cochran 1960; Yelle 1979). Learning occurs on the part of workers directly involved into manufacturing of the product (Andress 1954).

The first model of Wright (2012) describes the learning rate as a relative decrease in average costs per product unit over the whole history of production. The second learning curve model, called Crawford or Stanford model (Yelle 1979), introduces the learning rate as a relative decline in the marginal costs, i.e., costs required to produce the last product unit. It is being observed that learning is present only in the initial production state, i.e., after a while task times converge to steady-state task (Table 1). A brief review of the related literature and contributions of this study is presented in Table 2.

Problem description

In this study, the focus is on minimizing the number of stations to achieve an optimum balance; therefore, the idle time should be minimized and the efficiency of the line should be enhanced. These goals may be achieved by smoothing the amount of workload and maximizing the equalization of the workload among stations. It was assumed that training, which is done to promote workers to upper levels, is performed between periods and it takes zero time. Workers are classified into four types based on their skill levels. The level of each work station indicates types of workers allowed to work at that station. Each worker has exactly one skill and exactly belongs to one skill level. Workers with skill level 4 can work on task levels 1, 2, 3, and 4. Workers with skill level 3 can work on task levels 1, 2, 3, and so on. In each period, workers can be trained to improve their working abilities to operate

Table 1 Worker skills promotion possibility

	Skill level 1	Skill level 2	Skill level 3	Skill level 4
Skill level 1	–	*	*	*
Skill level 2	–	–	*	*
Skill level 3	–	–	–	*
Skill level 4	–	–	–	–

Table 2 Overview of the related literature and contributions of this study

Study	Human-related issues (hiring, firing, and salary)	Training and learning	ALB-I	ALB-II	Parallel stations assembly line balancing	U-I type assembly line balancing	Simple assembly line balancing	Two-sided assembly line balancing	Mixed-model assembly line balancing	Method
This study	*	*	*			*			*	NSGA-II MOPSO
Yuan et al. (2015)						*			*	HBMO and SA
Kucukkoc and Zhang (2014)					*			*	*	Agent-based ACO enhanced heuristics and model sequencing agent
Manavizadeh et al. (2013)		*	*			*			*	SA
Özcan and Toklu (2010)							*		*	GA
Simaria and Vilarinho (2009)		*	*						*	SA
Kim et al. (2009)				*				*		GA
Wu et al. (2008)			*					*		Branch-and-bound
Aryanezhad et al. (2009)	*	*								Model numerical examples
Simaria and Vilarinho (2004)		*		*		*	*			Genetic algorithm

other task levels. The initial number of workers with skill level O in the beginning of the planning horizon is known. Levels of tasks are known, and the level of each station is equal to the maximum level of tasks which are assigned to it.

Assumptions

- Parallel stations are not allowed.
- Operator walking time is ignored.
- All parameters in the model are assumed to be deterministic.
- There is no uncertainty.
- Each task must be assigned to exactly one station.
- All predecessors or successors of a task have already been assigned to a station (the precedence constraint.
- The total time of the tasks assigned to each station, (i.e., the station time), may not exceed the cycle time (the cycle time constraint).
- Salary is merely dependent on worker's skill level and not depending on machine levels.
- All of the machine types which need the same skill levels assumed to be similar in worker assignment.
- Cost of hiring and firing are given, and they merely depend on skill levels.
- Each task needs just one worker.
- Training, which is done to promote workers to upper levels, is performed between periods and it takes zero time.
- The productivity of experienced workers is assumed to be equal to 100 %.
- The productivity of newly trained workers is assumed to be fewer than that of experienced ones, and it depends on the skill level to which they are trained.
- Productivity of newly hired workers is assumed to be fewer than that of experienced ones, and it depends on the skill level for which they are hired.
- Cost of training from one skill level to another is given, and it depends on both skill levels.

Objective functions

Minimizing the number of stations which is equivalent to the minimization of the idle time related to the line is one of the most important objectives in this article. Each model's idle time is multiplied by the corresponding proportion ($q'j$). Computation of total weighted idle time (WIT) is shown below (Manavizadeh et al. 2013; Simaria and Vilarinho 2009).

$$\text{Minimize WIT} = \sum_{j=1}^{M} q'_j \sum_{r=1}^{R} \left(C - \sum_{i=1}^{I} x_{ir} \times t_{ir} \right). \quad (1)$$

By balancing the workloads between stations, the idle time will be distributed across the workstations as equally as possible for each model. The workload balance between workstations will be computed by function B_b. Therefore, the objective would be minimization of B_{b}, as shown below (Simaria and Vilarinho 2009):

$$\text{Minimize } B_{\mathrm{b}} = \frac{v}{v-1}\sum_{r=1}^{R}\left(\frac{id_r}{\mathrm{WIT}} - \frac{1}{v}\right) \tag{2}$$

where id_r is the idle time of workstation r:

$$id_r = \sum_{j=1}^{M} q_j' id_{rj}. \tag{3}$$

The value of function B_{b} is within the value range of [0,1]. In worst case, where the average idle time of the line is equal to the idle time of one of the workstations, the value equals 1, and in optimal case, equals zero when it is equally distributed among all workstations in the line actually it. By minimizing B_{w}, the optimal value for B_{w} is calculated as shown below (Simaria and Vilarinho 2009).

$$\text{Minimization } B_{\mathrm{w}} = \frac{M^2}{v(M^2-1)}\sum\sum\left(\frac{q_j' id_{rj}}{id_{rj}} - \frac{1}{M^2}\right). \tag{4}$$

The value of B_{w} is within the value range of [0,1]. In worst case, when only model attributes to the idle time of each workstation, it equals 1, and when all models attribute equally to the idle time at each workstation it equals zero (Simaria and Vilarinho 2009).

The value of WIT is different from one problem to another due to their dependence on the cycle time and task processing of each specific problem, whereas the function B_{b} and B_{w} are always within the value range of [0,1]. An alternative measurement, which is always within a fix range of values, is the weighted line efficiency (WE) (Simaria and Vilarinho 2009).

This value varies between 0 and 1 is a direct indication of the efficiency of the line; 1 being the optimal value which indicates no idle time is found. The WE in an objective function computed as follows (Simaria and Vilarinho 2009):

$$\mathrm{WE} = \sum q_j' \times \left(\frac{\sum_{i=1}^{I} t_{ij}}{v \times c}\right). \tag{5}$$

Another important objective is ti distribute tasks among workstations in a balanced fashion based on the job processing time. To achieve this goal, the difference between processing time of each model in each station and the average processing time for each model should be minimized. The formula is given below:

$$\mathrm{WI} = \sum_{r=1}^{R}\left|\sum_{j=1}^{M}\sum_{i=1}^{I} t_{ij} \times x_{ij} - \mathrm{mean}_j\right| \tag{6}$$

where t_{ij} is the processing time of task i related to model j, and x_{ir} is equal 1 if task i assigned to station r and $mean_j$ is the average processing time workload needed for model j (Simaria and Vilarinho 2009).

$$\mathrm{mean}_j = \frac{\sum_{i=1}^{I} t_{ij}}{I}. \tag{7}$$

In Z_2, we want to minimize all costs related to operators considering:

$$\text{Hiring cost:} \quad \sum_{s=1}^{T}\sum_{o=1}^{4}\sum_{k=1}^{\mathrm{MS}} h_{o,s} \times u_{o,s,k} \tag{8}$$

$$\text{Training cost:} \quad \sum_{s=1}^{T}\sum_{o=1}^{4}\sum_{o'}\sum_{k=1}^{\mathrm{MS}}\sum_{k'=1}^{\mathrm{MS}} c_{o,o',s} \times UX_{o',o,s,k',k} \tag{9}$$

$$\text{Salary:} \quad \sum_{s=1}^{T}\sum_{o=1}^{4}\sum_{k=1}^{\mathrm{MS}} s_{o,s} \times E_{o,s,k} \tag{10}$$

$$\text{Firing cost:} \quad \sum_{s=1}^{T}\sum_{o=1}^{4}\sum_{k=1}^{\mathrm{MS}} f_{o,s} \times W_{o,s,k}. \tag{11}$$

Mathematical model

Parameters

i, b	Index of task
R	Maximum number of stations
r, r'	Index of station
J	Model (product) $\{1,\ldots, M\}$
s	Index of period
O	Work skills category {1, 2, 3, 4}
k,k'	Index for station levels $\{1, \ldots, \mathrm{MS} = 4\}$
M	Number of models
V	Number of operators
I	Total number of tasks in the combined precedence diagram, $(i = 1, 2, 3, \ldots, I)$
MS	Number of station levels
D	The vector presenting the total demand for each model, $D = \{D_1, D_2, \ldots, Dm\}$
q'	The overall proportion of the number of units of model j
P_{ib}	Showing the precedence relationship between task b and i. Equal 1 if task b is the precedence for task i
su_{ib}	Showing the succeeding relationship between task b and i. Equal 1 if task b is a successor for task i

o_{ib} A zero–one variable which determines whether or not constraints 2 or constraint 3 is satisfied
C Cycle time
P Total time in the planning horizon
id_r Idle time of station r
D_{js} Demand of model j in period s
t_{ij} Processing time of task i of model j
w_o Number of workers of skill category o
w_s^o Number of workers of skill category o working in period s
pt_s Regular time rate for workers during period s
ot_s Overtime rate for workers during period s
h' Total working hours in a period
h' Minimum overtime work for operators
$h_{o,s}$ Cost of hiring of a worker with skill level o in period s
$s_{o,s}$ Salary of each o-level worker in period s
$f_{o,s}$ Firing cost of each o-level worker fired in period s
Co,o'',s Training cost of each o-level worker trained for skill level o' in period s
α_o Productivity of each newly o-level worker hired in period s $0 < \alpha_o < 1$
$\beta_{o,o}'$ Training productivity of o-level worker trained for skill level o' $0 < \beta_{o,o'} < 1$
a_{ro} Equals 1 if workers of skill category o can work at processing stage r and zero

Decision variables

x_{ir} Equals 1 if task i is assigned to station r and equal 0 otherwise
y'_r Equals 1 if workstation r is used for assembly and 0 otherwise
x'_{rs} Total number of overtime hours done by workers at station r in period s
x_{rs}^o Equals 1 if worker from skills category o is allocated to station r in period s
$U_{o,s,k}$ Number of o-level workers who are hired and assigned to station level k in period s
$E_{o,s,k}$ Number of existing o-level workers who are assigned to station level k in period s
$UX_{o',o,s,k,}$ Number of o'–level workers who were assigned to task level k in period $s-1$ and now are trained to skill level o and assigned to task level k' in period s
$UG_{o',o}$ Equals 1 if training from skill level o' to skill level o is possible and 0 otherwise

$$\text{mean}_j = \frac{\sum_{i=1}^{I} t_{ij}}{I}$$

$$z_1 = \hat{c} \times \left(\frac{1}{\sum_{j-1}^{M} q'_j \left(\frac{\sum_{i=1}^{I} t_{ij}}{v \times c} \right)} \right) + \Phi \times \text{WIT} + \gamma \times \left(\sum_{r=1}^{R} \left| \sum_{j=1}^{M} \sum_{i=1}^{I} t_{ij} \times x_{ij} - \text{mean}_j \right| \right) + \in \times V.$$

Subject to:

$$\sum_{r=1}^{R} x_{ir} = 1 \tag{12}$$

$$\sum_{r1} x_{ir1} - x_{br} \le M \times o_{ib} \quad \forall i, b, r, \quad r1 \le 1, \quad p_{ib} = 1 \tag{13}$$

$$\sum_{r1} x_{ir1} - x_{ib} \le M \times (1 - o_{ib}) \quad \forall i, b, r \quad r1 \le 1 \quad \text{su}_{ib} = 1 \tag{14}$$

$$t_{jr} = \sum_{i=1}^{I} x_{ir} \times \text{Max}\{t_{ij}\} \quad \forall j, r, s \tag{15}$$

$$\sum_{i=1}^{I} \sum_{j=1}^{M} x_{ir} \times \text{Max}\{t_{ij}\} \le C \quad \forall r \tag{16}$$

$$C = \frac{p}{\sum_{j=1}^{M} D_j} \tag{17}$$

$$q'_j = \frac{D_j}{\sum_{j=1}^{M} D_j} \tag{18}$$

$$v = \sum_{r=1}^{R} y'_r \tag{19}$$

$$\text{WIT} = \sum_{j=1}^{M} q'_j \sum_{r=1}^{R} \left(C - \sum_{i=1}^{I} x_{ir} \times t_{ir} \right) \tag{20}$$

$$\text{id}_{rj} = C - \sum_{i=1}^{I} x_{ir} \times t_{ir} \quad \forall r, j \tag{21}$$

$$\text{id}_r = \sum_{j=1}^{M} q'_j id_{rj} \quad \forall r \tag{22}$$

$$\sum_{i=1}^{I} x_{ir} \ge y'_r \quad \forall r. \tag{23}$$

Minimizing

$$Z_2 = \sum_{r=1}^{R}\sum_{o=1}^{4}\sum_{s=1}^{T}(x_{rs}^{o} \times pt_s) + (x'_{rs} \times ot_s) + \sum_{s=1}^{T}\sum_{o=1}^{4}\sum_{k=1}^{MS} h_{o,s} \times u_{o,s,k} + \sum_{s=1}^{T}\sum_{o=1}^{4}\sum_{o'}\sum_{k=1}^{MS}\sum_{k'=1}^{MS} c_{o,o',s} \times UX_{o',o,s,k',k} + \sum_{s=1}^{T}\sum_{o=1}^{4}\sum_{k=1}^{MS} s_{o,s} \times E_{o,s,k} + \sum_{s=1}^{T}\sum_{o=1}^{4}\sum_{k=1}^{MS} f_{o,s} \times W_{o,s,k}.$$

Subject to:

$$\sum_{o=1}^{4} x_{rs}^{o} \times a_{ro} = 1 \quad \forall r, s \tag{24}$$

$$\sum_{r=1}^{R} x_{rs}^{o} \times a_{ro} = w_s^o \quad \forall o, s \tag{25}$$

$$w_s^o \le w_o \quad \forall o, s \tag{26}$$

$$\mathrm{t_j} = \sum_{i=1}^{\mathrm{I}} t_{ij} \quad \forall j \tag{27}$$

$$\sum_{s=1}^{T}\left[(x_{rs}^{o}) \times h' + (x'_{rs})\right] \ge \sum_{s=1}^{T}\sum_{j=1}^{M}\frac{D_{js}}{t_j} \quad \forall r \tag{28}$$

$$v = \sum_{r=1}^{R}\sum_{o=1}^{4} x_{rs}^{o} \tag{29}$$

$$E_{o,s,k} = E_{o,s-1,k} + U_{o,s,k} - W_{o,s,k} + \sum_{s=1}^{4}\sum_{k'=1}^{MS}(UX_{o',o,s,k',k} - UX_{o,o',s,k',k}) \quad \forall o, s, k \tag{30}$$

$$A \times \sum_{o=1}^{4}\left[E_{o,s-1,k} + \alpha_o \times U_{o,s,k} - W_{o,s,k} + \sum_{e=1}^{4}\sum_{k'=1}^{MS}(\beta_{o',o}UX_{o',o,s,k',k} - UX_{o,o',s,k',k})\right] \ge \sum_{r=1}^{R}\sum_{o=1}^{4} \mathrm{x}_{rs}^{o} \tag{31}$$

$$W_{o,s,k} \le R \times a_{ro} \quad \forall o, s, k \tag{32}$$

$$U_{o,s,k} \le R \times a_{ro} \quad \forall o, s, k \tag{33}$$

$$UX_{o,o\prime,s,k',k} \le R \times a_{ro} \quad \forall o', o, s, k, k' \tag{34}$$

$$UX_{o\prime,o,s,k',k} \le R \times a_{ro} \quad \forall o', o, s, k, k' \tag{35}$$

$$UX_{o\prime,o,s,k',k} \le R \times \mathrm{UG_{o\prime,o}} \tag{36}$$

$$\sum_{o\prime}^{O=4}\sum_{k'}^{\mathrm{MS}} UX_{o,o\prime,s,k',k} + W_{o,s,k} \le E_{o,s-1,k} \quad \forall o, s, k \tag{37}$$

$$\sum_{o'=1}^{O=4}\sum_{k=1}^{\mathrm{MS}} UX_{o',o,s,k',k} \le R \times y_{o,s,k'} \quad \forall o, s, k' \quad y_{o,s,k'} = [0, 1] \tag{38}$$

$$W_{o,s,k'} \le R \times (1 - y_{o,s,k'}) \quad \forall o, s, k. \tag{39}$$

Methodology

Proposed model in this paper is multi-objective, so the methods for solving the problem are NSGA-II and MOPSO. Rabbani et al. (2016a, b) applied these two algorithms for solving a mixed-model assembly line problem, and the results obtained by these two algorithms were compared to each other. NSGA-II is a popular non-domination-based genetic algorithm for multi-objective optimization. It is a very effective algorithm but has been generally criticized for its computational complexity, lack of elitism, and for choosing the optimal parameter value for sharing parameter (Rabbani et al. 2016a, b). Kusiak and Wei (2012) introduced MOPSO for optimizing continuous non-linear functions, Particle Swarm Optimization (PSO) defined a new era in Swarm Intelligence (SI). PSO is a population-based method for optimization. The population of the potential solution is called as swarm and each individual in the swarm is defined as particle. PSO is motivated by social behavior of birds flocking or fish schooling Solutions are represented by particles in the search space. The particles fly in the swarm to search their best solution based on experience of their own and the other particles of the same swarm. PSO started to hold the grip amongst many researchers and became the most popular SI technique soon after getting introduced, but due to its limitation of optimization only of single objective, a new concept Multi-Objective PSO (MOPSO) was introduced, by which optimization can be performed for more than one conflicting objectives, simultaneously. Coello et al. (2002) described the advantages of using MOPSO in solving multi-objective optimization problem rather than the single objective version of the algorithm.

Representation of solutions

The chromosome is a string of length I which shows the task numbers, where each element represents a task and the value of each element represents the workstations to which the corresponding task is assigned. The maximum number of stations is equal to total number of tasks. For example, for 16 tasks, 9 workstations will be created.

In this research, individuals in the initial population are all randomly generated. While a heuristic procedure can provide good initial solutions, it can cause the solutions to be biased.

Illustrative example

In this section, 5 small-size and 5 large-scale problems are implemented to compare the performance of algorithms with each other in various size problems. Parameters of problems were generated based on Table 3. In this paper, the workers assignment is based on their skill level. Workers with skill level 4 can work on task levels 1, 2, 3, and 4. Workers with skill level 3 can work on task levels 1, 2, 3, and so on. The problem with five tasks is as follows:

The precedence diagram of five task problems is shown in Tables 4, 5, 6, 7, 8, 9.

The results from NSGA-II algorithm are shown below:

This table shows that task number 1 is assigned to workstation number 4, task 2 and task 3 are assigned to workstation number 3, task number 4 is assigned to workstation number 2, and task number 5 is assigned to workstation number 1. Training should happen according to Table 10:

Table 3 Test problem generation

Parameters	Value	Parameters	Value
Demand	U(5, 10)	Hiring cost	U(1500, 2000)
Processing time	(2, 5)	Firing cost	U(1500, 2000)
Training cost	U(50, 150)	Salary	U(100, 500)

Table 4 Initial number of workers with skill level 1 in the beginning of the planning horizon

Skill level	1	2	3	4
Initial number of workers	5	1	1	0

Table 5 Level of tasks

	Task 1	Task 2	Task 3	Task 4	Task 5
Skill level 1					
Skill level 2					
Skill level 3		*			*
Skill level 4	*		*	*	

Signed cells means that worker with skill level o can work the task number j considering tasks level, and in addition, the training cost from skill level O to skill level O' in period s is shown in Table 6

Table 6 Cost of training from skill level O to skill level O' in periods

From skill	Period/ to skill	Skill level 2	Skill level 3	Skill level 4
Skill level 1	1	118	64	127
	2	125	69	112
	3	118	83	117
Skill level 2	1		97	130
	2		95	65
	3		106	66
Skill level 3	1			124
	2			86
	3			146

Parameters tuning

The efficiency of the meta-heuristic algorithms in finding better solutions in less run time is considerably dependent on their parameters. To setting the MOPSO and NSGA-II parameters, design of experiment (DOE) using Taguchi approach is used in the paper. The performance of NSGA-II is influenced by four parameters, including population size (N_p), maximum number of generations (Max_Iteration), mutation rate (P_m), and crossover rate (P_c). MOPSO parameters consist of population size (N_p), maximum number of iterations (Max_Iteration), inertia weight (w), repository size (N_r), personal learning coefficient (c_1), and global learning coefficient (c_2). After specifying levels for each parameter (factor), design of experiment is performed using the Minitab software to set these two groups of parameters (Figs. 1, 2, 3). Parameters tuning for both algorithms are done according to the results of large-sized problem (Table 11). The consequences of Taguchi method in tuning of parameters are shown in Figs. 4 and 5. In addition, the results are summarized in Table 12.

Comparative results

Comparison metrics: It is common to compare the performance of the multi-objective algorithms' performance by means of some specific comparison metrics; to compare proposed algorithms with each other, three comparison metrics are employed (Rabbani et al. 2016a, b).

1. Number of Pareto solutions (NPS): The quantity of non-dominated solutions that every algorithm can discover.

2. Spacing metrics (SM): This kind of metric provides us details about the uniformity of the distribution of the solutions obtained by the way of each algorithm. This metrics are computed as follows:

$$\mathrm{SM} = \sqrt{\frac{1}{N-1} \times \sum_{i=1}^{n} \left(d_i - \bar{d}\right)^2} \tag{40}$$

Table 7 Cost of hiring, firing, and salary of each *O*-level worker in each period are generated randomly

Skill level/period	Hiring			Firing			Salary		
	1	2	3	1	2	3	1	2	3
Skill level 1	1823	1604	1667	1823	1604	1667	1800	1600	1500
Skill level 2	1813	1530	1959	1813	1530	1959	1800	1500	1900
Skill level 3	1910	1677	1509	1910	1677	1509	1900	1677	1500
Skill level 4	1878	1628	1651	1878	1628	1651	1800	1700	1600

Table 8 Processing time related to five task problems

Model	1	2	3	4
Task				
1	5	2	5	4
2	4	3	2	3
3	4	4	4	3
4	5	3	3	4
5	5	3	4	3

Table 9 Task assignment

4	3	3	2	1

where d_i is the Euclidean distance between solution i and the nearest solution belonged to Pareto sets of solutions. $\overline{d}$ is the average value of all d_i.

3. Diversification metrics (DM): This metric specifies the spread of solution set and determined as follows:

$$\mathrm{DM} = \sqrt{\sum_{i=1}^{n} \max\left(\left\|x_t^i - y_t^i\right\|\right)} \qquad (41)$$

where $\max(\|x_t^i - y_t^i\|)$ is the Euclidean distance between the non-dominated solutions x_i^t and y_i^t

Small-size problem

NSGA-II and MOPSO algorithms are used for solving the test problems. Each test problem operates five times, and the outcomes are summarized in Table 13. The average values for all mentioned metrics are shown in Table 12, and the average run time for each test problem is demonstrated in Table 14. Generally, we can say that in small-size problems, NSGA-II could achieve greater number of Pareto solutions than MOPSO. Spacing metrics obtained by mentioned formula show that NSGA-II provides non-dominated solutions that have less average value of spacing metrics. These results show that the non-dominated set obtained by NSGA-II is more uniformly distributed in comparison with the MOPSO algorithm. Diversification metric in NSGA-II and MOPSO does not show superiority of none of them, but average value for diversification metric obtained by NSGA-II for test problems is greater than MOPSO. In small-size problems, computational time for MOPSO algorithm is considerably less than the NSGA-

Table 10 Training

	Period 1				Period 2				Period 3			
	Skill level 1	Skill level 2	Skill level 3	Skill level 4	Skill level 1	Skill level 2	Skill level 3	Skill level 4	Skill level 1	Skill level 2	Skill level 3	Skill level 4
Skill level 1	0	0	0	2	0	0	0	2	0	0	0	2
Skill level 2	0	0	0	1	0	0	0	0	0	0	0	0
Skill level 3	0	0	0	0	0	0	0	1	0	0	0	1
Skill level 4	0	0	0	0	0	0	0	0	0	0	0	0

14	14	16	3	4	4	4	2	2	1	2	6	7	6	9	9

Encoding Decoding

Workstation	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Assigned tasks	10	8,9	4	5,6,7	-	12,14	13	-	15,16	-	-	-	-	1,2	-	3

Fig. 1 One task assignment chromosome

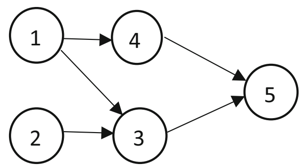

Fig. 2 Combined diagram for five task problems

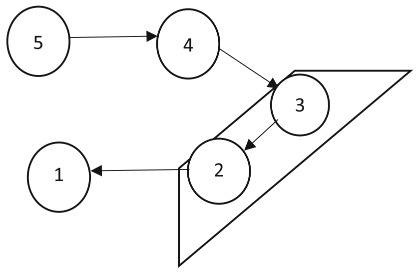

Fig. 3 NSGA-II solution for five task problems

Table 11 Total number of hiring and firings

	Hire				Fire			
Skill level	1	2	3	4	1	2	3	4
Number of workers	0	0	0	6	9	2	1	0

II algorithm. Table 14 shows that the average computational times for both the algorithms.

Large-size problem

In this Sect. "Parameters tuning", various size problems are implemented to compare the performance of algorithms with each other in large-scale problems. The comparisons metrics are similar to small-sized problems, and we employ number of Pareto solutions (NPS), spacing metrics (SM), and diversification metric (DM) for comparison of algorithms. In large-size test problems, number of Pareto solutions in the NSGA-II and MOPSO algorithms does not show superiority of none of them (Tables 15, 16). Spacing metrics obtained by mentioned formula show that NSGA-II provides non-dominated solutions that have less average value of spacing metrics. These results show that the non-dominated set obtained by NSGA-II is more uniformly distributed in comparison with the MOPSO algorithm. Diversification metric in NSGA-II and MOPSO does not show superiority of none of them, but average value for diversification metric obtained by MOPSO for test problems is greater than NSGA-II. In large-size problems, the average computational time for MOPSO algorithm is greater than NSGA-II.

Conclusion

This research deals with balancing a mixed-model assembly U-line considering human-related issues. The objective function consists of two separate components. The first part of the objective function is related to balance problem. In this part, objective functions are minimizing the cycle time, minimizing the number of workstations, and maximizing the line efficiencies. The second part is related to human issues and consists of hiring cost, firing cost, training cost, and salary, and the labor assignment policy was defined. In this research, workers are classified into four types based on their skill levels. The level of each work station indicates types of workers allowed to work at that station. Two

Fig. 4 Obtained results for NSGA-II parameters tuning

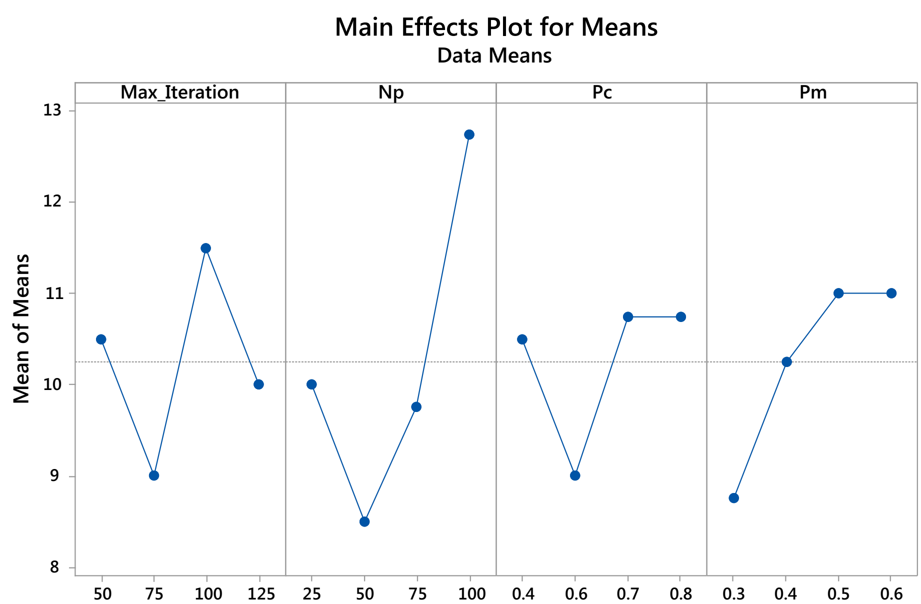

Fig. 5 Obtained results for MOPSO parameters tuning

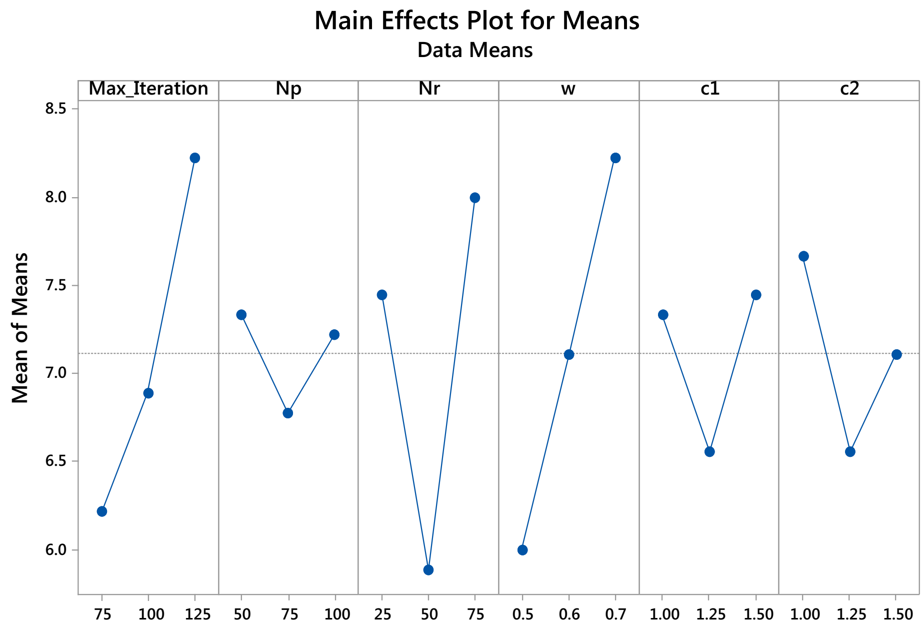

Table 12 Tuned parameters for NSGA-II and MOPSO algorithms

Algorithm	Parameter							
	Max_iteration	N_p	P_c	P_m	N_r	W	c_1	c_2
NSGA-II	100	100	0.7	0.5	…	…	…	…
MOPSO	125	50	…	…	75	0.7	1.5	1

meta-heuristic algorithms (NSGA-II and MOPSO) are used for solving a bi-objective problem presented in this paper. In small-sized problem, MOPSO outperforms NSGA-II with respect to computational time, but in large-scale problem in all problems except the problem with 16 tasks, the operation of NSGA-II is better than MOPSO with regard to computational time. In most problems, including small- and large-sized problems, the number of Pareto solutions (NPS) generated with NSGA-II is more than MOPSO. Spacing metrics obtained by the NSGA-II provide non-dominated solutions that have a less average value of the spacing metrics. These data reveal that the non-dominated set obtained by the NSGA-II is more uniformly distributed in comparison with the MOPSO algorithm. In two other comparison metrics, the obtained results do not show any superiority of each algorithm with comparison another one. The algorithms provided approximated Pareto solutions for decision maker to choose from them, but in some real cases, especially in critical industries, where any error has catastrophic results, finding approximated solutions cannot be helpful for decision makers.

Future developments will be devoted to investigate the effects of human resource planning policies on balancing of a mixed-model assembly U-line in uncertainty conditions,

Table 14 Average computational times for small-size problems (in seconds)

Number of tasks	NSGA-II	MOPSO
5	7.812923	2.629137
6	4.666816	2.084193
7	5.151678	2.364705
8	41.968083	2.780
10	47.039556	6.261575

Table 13 Computational results for small-size problem

Number of tasks	NPS		SM		MID		Diversity	
	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II
5	3	3.66	0.939	0.029	12,104.313	23,473.333	26.697	233.388
6	2.88	3.8	1.923	1.280	14,280.774	19,886.250	255.612	337.352
7	6	8.9	1.145	1.4219	8657.736	18,860.888	255.668	480.552
8	6.3	10.5	1.519	0.612	11,554.265	17,592.728	297.687	442.358
10	10.4	12	1.103	1.265	27,919.017	23,332.4	502.774	430.670

Table 15 Computational results for large-size problems

Number of tasks	NPS		SM		MID		Diversity	
	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II
16	11.5	12.5	1.307	1.244	61,031.23	54,649.58	513.695	508.78
17	10	7.4	1.598	1.013	64,565.99	58,431.00	543.438	257.06
18	10	5	1.914	1.224	68,312.35	66,500.80	472.476	255.95
19	11.3	13	1.068	1.024	71,492.21	67,707.84	452.856	498.853
20	13	13	1.31991	1.23546	75,269.613	73,343.53	673.46195	560.282

Table 16 Average computational times for large-size problems (in seconds)

Number of tasks	NSGA-II	MOPSO
16	460.982041	143.715486
17	2620.811549	2972.869897
18	1067.034342	1200.195
19	1518.914096	6041.283304
20	13,680.837762	16,997.6252

given the fact that human activities are not deterministic. In addition, solving a problem by exact methods, such as goal programming and goal attainment, can have great managerial insights to make decisions more precisely.

References

Aigbedo H, Monden Y (1997) A parametric procedure for multicriterion sequence scheduling for JustIn-Time mixed-model assembly lines. Int J Prod Res 35(9):2543–2564

Anderson EJ, Ferris MC (1994) Genetic algorithms for combinatorial optimization: the assemble line balancing problem. ORSA J Comput 6(2):161–173

Andress FJ (1954) The learning curve as a production tool. Harvard Bus Rev 32(1):87–97

Aryanezhad M, Deljoo V, Mirzapour Al-e-hashem S (2009) Dynamic cell formation and the worker assignment problem: a new model. Int J Adv Manuf Technol 41(3–4):329–342

Baloff N (1971) Extension of the learning curve–some empirical results. Oper Res Q 329–340

Becker C, Scholl A (2006) A survey on problems and methods in generalized assembly line balancing. Eur J Oper Res 168(3):694–715

Bhatnagar R, Venkataramanaiah S, Rajagopalan A (2003) A model for contingent manpower planning: insights from a high clock speed industry

Chakravarty AK, Shtub A (1988) Modelling the effects of learning and job enlargement on assembly systems with parallel lines. Int J Prod Res 26(2):267–281

Chen R-S, Lu K-Y, Yu S-C (2002) A hybrid genetic algorithm approach on multi-objective of assembly planning problem. Eng Appl Artif Intell 15(5):447–457

Chiang W-C, Urban TL (2006) The stochastic U-line balancing problem: a heuristic procedure. Eur J Oper Res 175(3):1767–1781

Cochran E (1960) New concepts of the learning curve. J Ind Eng 11(4):317–327

Coello CAC, Van Veldhuizen DA, Lamont GB (2002) Evolutionary algorithms for solving multi-objective problems, vol 242. Kluwer Academic, New York

Cohen Y, Ezey Dar-El M (1998) Optimizing the number of stations in assembly lines under learning for limited production. Prod Plan Control 9(3):230–240

Cohen Y, Vitner G, Sarin SC (2006) Optimal allocation of work in assembly lines for lots with homogenous learning. Eur J Oper Res 168(3):922–931

Erel E, Gokcen H (1999) Shortest-route formulation of mixed-model assembly line balancing problem. Eur J Oper Res 116(1):194–204

Erel E, Sarin SC (1998) A survey of the assembly line balancing procedures. Prod Plan Control 9(5):414–434

Gokcen H, Erel E (1997) A goal programming approach to mixed-model assembly line balancing problem. Int J Prod Econ 48(2):177–185

Gökcen H, Erel E (1998) Binary integer formulation for mixed-model assembly line balancing problem. Comput Ind Eng 34(2):451–461

Gökçen H, Ağpak K, Gencer C, Kizilkaya E (2005) A shortest route formulation of simple U-type assembly line balancing problem. Appl Math Model 29(4):373–380

Hax A, Candea D (1984) Production and inventory management. Prentice-Hall, Englewood Cliffs, NJ

Helgeson W, Birnie D (1961) Assembly line balancing using the ranked positional weight technique. J Ind Eng 12(6):394–398

Kim YK, Hyun CJ, Kim Y (1996) Sequencing in mixed model assembly lines: a genetic algorithm approach. Comput Oper Res 23(12):1131–1145

Kim YK, Song WS, Kim JH (2009) A mathematical model and a genetic algorithm for two-sided assembly line balancing. Comput Oper Res 36(3):853–865

Kucukkoc I, Zhang DZ (2014) Mathematical model and agent based solution approach for the simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines. Int J Prod Econ 158:314–333

Kusiak A, Wei X (2012) Optimization of the activated sludge process. J Energy Eng 139(1):12–17

Leu YY, Matheson LA, Rees LP (1994) Assembly line balancing using genetic algorithms with heuristic-generated initial populations and multiple evaluation criteria*. Decis Sci 25(4):581–605

Manavizadeh N, Hosseini N-S, Rabbani M, Jolai F (2013) A Simulated Annealing algorithm for a mixed model assembly

U-line balancing type-I problem considering human efficiency and just-in-time approach. Comput Ind Eng 64(2):669–685

McMullen PR, Frazier G (1998) Using simulated annealing to solve a multiobjective assembly line balancing problem with parallel workstations. Int J Prod Res 36(10):2717–2741

Miltenburg G, Wijngaard J (1994) The U-line line balancing problem. Manag Sci 40(10):1378–1388

Monden Y (2011) Toyota production system: an integrated approach to just-in-time. CRC Press, New York

Ōno T (1988) Toyota production system: beyond large-scale production. Productivity Press, New York

Özcan U, Toklu B (2010) Balancing two-sided assembly lines with sequence-dependent setup times. Int J Prod Res 48(18): 5363–5383

Park PS (1991) The examination of worker cross-training in a dual resource constrained job shop. Eur J Oper Res 52(3):291–299

Rabbani M, Mousavi Z, Farrokhi-Asl H (2016) Multi-objective metaheuristics for solving a type II robotic mixed-model assembly line balancing problem. J Ind Prod Eng 1–13

Rabbani M, Siadatian R, Farrokhi-Asl H, Manavizadeh N (2016b) Multi-objective optimization algorithms for mixed model assembly line balancing problem with parallel workstations. Cogent Eng 3(1):1158903

Schonberger RJ (1983) Japanese manufacturing techniques. The Free Press, New York

Simaria AS, Vilarinho PM (2004) A genetic algorithm based approach to the mixed-model assembly line balancing problem of type II. Comput Ind Eng 47(4):391–407

Simaria AS, Vilarinho PM (2009) 2-ANTBAL: an ant colony optimisation algorithm for balancing two-sided assembly lines. Comput Ind Eng 56(2):489–506

Thomopoulos NT (1967) Line balancing-sequencing for mixed-model assembly. Manag Sci 14(2):B-59–B-75

Vilarinho PM, Simaria AS (2002) A two-stage heuristic method for balancing mixed-model assembly lines with parallel workstations. Int J Prod Res 40(6):1405–1420

Wright TP (2012) Factors affecting the cost of airplanes. J Aeronaut Sci Inst Aeronaut Sci 3(4)

Wu E-F, Jin Y, Bao J-S, Hu X-F (2008) A branch-and-bound algorithm for two-sided assembly line balancing. Int J Adv Manuf Technol 39(9–10):1009–1015

Yelle LE (1979) The learning curve: historical review and comprehensive survey. Decis Sci 10(2):302–328

Yuan B, Zhang C, Shao X, Jiang Z (2015) An effective hybrid honey bee mating optimization algorithm for balancing mixed-model two-sided assembly lines. Comput Oper Res 53:32–41