

Hybrid genetic algorithm to solve resource constrained assembly line balancing problem in footwear manufacturing

Nguyen Thi Phuong Quyen¹ · James C. Chen² · Chao-Lung Yang¹

Published online: 11 May 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract This paper aims to develop a hybrid genetic algorithm (HGA) to solve the resource constrained assembly line balancing problem (RCALBP) in the sewing line of a footwear manufacturing plant. Sewing, which is the most critical process in footwear manufacturing, has a series of processes, such as punching, trimming, attaching shoelaces. RCALBP in the sewing line considers not only the precedence constraints of product assembly but also the resource constraints, such as operators and equipment. A novel HGA that includes two stages is proposed to optimize the resources in the sewing line. The first stage uses the priority rule-based method (PRBM) to determine the feasible solutions of assigning tasks and machines to workstations. The solutions of PRBM are used to construct the initial population of genetic algorithm (GA) in the second stage. To ensure that the solution of GA is feasible, a two-point-order crossover with the new technique of searching feasible solution patterns is proposed. Moreover, the mutation procedure of GA is modified to avoid the building block from breaking, which may cause unfeasible solutions in RCALBP. A self-tuning method is also applied recursively to exclude unfeasible solutions. The proposed HGA is compared with the man-

ual procedure adopted practically in factories, the existing heuristic model in the literature, and the traditional GA. Based on actual data from a footwear factory, computational results demonstrate that the proposed HGA can achieve better results than the other algorithms.

Keywords Footwear manufacturing · Genetic algorithm · Priority rule-based method · Resource constrained assembly line balancing

1 Introduction

The footwear industry is rapidly expanding. According to the 2014 Global Footwear Market Analysis published by Research and Markets, the total revenue of the world footwear market in 2013 was US\$ 258.5 billion, which represented a compound annual growth rate of 4.4% between 2009 and 2013. The value of the footwear market is expected to reach US\$ 329.8 billion in 2016 (Marketline 2014).

Similar to other industries, the footwear industry faces various challenges because of the market trend of low-volume orders and design variations. Nowadays, footwear is considered as a fashion item and its design style changes frequently. The footwear industry must adapt to these fast-paced changes. Thus, their assembly lines need to quickly respond to new footwear designs. Moreover, major footwear brands, such as Nike and Adidas, shortened their product life cycles to reflect market trends. As a result, footwear manufacturers have faced relatively low-volume orders.

Footwear manufacturing is known to depend mostly on considerable labor operations. Thus, the factories of major footwear manufacturers are located in countries with low labor cost, such as China, Vietnam, and Indonesia. However, the labor cost in these countries increased recently,

Communicated by V. Loia.

✉ Nguyen Thi Phuong Quyen
quyen.ntp@gmail.com

¹ Department of Industrial Management, National Taiwan University of Science and Technology, No. 43, Sec. 4, Keelung Rd., Taipei, Taiwan, ROC

² Department of Industrial Engineering and Engineering Management, National Tsing Hua University, No. 101, Sec. 2, Kuang-Fu Rd., Hsinchu, Taiwan, ROC

and robot-assisted manufacturing is not yet widely applied in the footwear industry. Thus, cost reduction is critical in this industry, particularly in the manufacturing sector in the supply chain.

The footwear manufacturing process includes several stages, and each stage specifies a product attribute, such as gender, size, color, sole type, and material. In all these stages, the sewing stage accounts for a high proportion of labor cost. This stage consists of a series of processes, such as punching, trimming, attaching shoelaces. The tasks are assigned to workstations (WSs) depending not only on precedence relations but also on resource constraints such as varying machine types. If workloads among the WS in the sewing line are unbalanced, the work-in-process and waiting time tend to increase. This process increases cost and production time. Therefore, balancing task loadings of WS in the sewing line is necessary. A factory that is more capable of balancing its assembly line in the sewing process has a more efficient sewing process and lower labor cost.

The mathematical formalization of the assembly line balancing problem (ALBP) was first introduced by [Salveson \(1955\)](#). ALBP assigns various tasks to WS and achieves specific objectives without violating certain constraints. Many methods have been developed to solve various forms of ALBPs. ALBP has numerous assumptions. Thus, [Baybars \(1986\)](#) proposed a classification for ALBP under the basic problems called simple assembly line balancing problem (SALBP). [Becker and Scholl \(2006\)](#) conducted a survey that focused on generalizing ALBP. In their study, all the characteristics of assembly line systems with both simple and generalized problems were surveyed to further exploit realistic problems. [Boysen et al. \(2007\)](#) proposed a classification of ALBP under the basic problem of SALBP with more complex assumptions, such as subsequent works, u-shaped line, parallel stations, and processing alternatives, which were classified to the general assembly line balancing problems. [Boysen et al. \(2008\)](#) summarized many methods, which can be categorized into exact and heuristic methods, to solve the various forms of ALBP.

Over the past decades, many researchers have developed exact solution approaches to determine the optimal solution to the task arrangement in ALBP. The two most popular approaches are the algorithm-based branch-and-bound method and dynamic programming, such as the FABLE algorithm ([Johnson 1988](#)), adapted general sequencing algorithm ([Sprecher 1999](#)), bidirectional branch-and-bound algorithm ([Scholl and Klein 1997](#)), and graph-based dynamic programming ([Bautista and Pereira 2009](#)). However, in most cases, the exact solution approaches cannot provide an efficient solution to large-scale problems because of computational complexity.

Aside from exact methods, heuristic approaches are also used to solve ALBP, which is considered as an NP-hard prob-

lem. Heuristic approaches cannot produce optimal solutions, but their procedures are applicable to general problems. Several evolutionary computation and meta-heuristic methodologies, including genetic algorithm (GA) ([Chen et al. 2002](#); [Ghosh and Gagnon 1989](#); [Kim et al. 1996](#); [Levitin et al. 1995](#); [Rekiek et al. 2001](#); [Sabuncuoglu et al. 2000](#); [Triki et al. 2014](#)), tabu search, ant colony algorithm, particle swarm optimization, simulated annealing, and scatter search ([Azadeh et al. 2015](#); [Cano-Belmán et al. 2010](#); [Chica et al. 2015](#); [Delice et al. 2014](#); [Özcan and Toklu 2009](#); [Rada-Vilela et al. 2013](#); [Rahimi-Vahed et al. 2007](#); [Roshani et al. 2012, 2013](#); [Zha and Yu 2014](#)), were proposed. According to [Tasan and Tunali \(2008\)](#), among the meta-heuristic approaches, GA gained research attention because of its directed random searches to determine optimal solutions in complicated contexts as a new alternative to traditional optimization techniques. Many studies employed the GA method to solve manufacturing optimization problems not only in ALBP but also in various areas, such as facility layout and location design, production planning and control, supply chain management ([Tasan and Tunali 2008](#)).

Thus, the present study focuses on GA as the main technique to solve the resource constrained ALBP (RCALBP). RCALBP was first proposed by [Ağpak and Gökçen \(2005\)](#) to consider not only the precedence relationships among tasks but also the resource constraints that limit the assigned tasks in the WS. [Chen et al. \(2012\)](#) solved RCALBP in the garment industry with the workers' skill level as a resource constraint. In the present study, we consider the type of machines as the resource constraint that limits only one type of machine in one WS. [Kao et al. \(2011\)](#) also addressed this problem using a bidirectional heuristic with the ranked positional weight (RPW) rule to assign tasks into WS (Kao's heuristic model). However, the solution of Kao's heuristic model is ineffective particularly to large-sized problems because the RPW rule can generate only an initial balance for the assembly line. Thus, Kao et al. only provided feasible solutions for large problems, and the results may be in the local optimum.

In this study, the proposed hybrid genetic algorithm (HGA) consists of the heuristic approach-based priority rule as the first stage based on the work of Kao et al. to establish a feasible initial solution. GA with a self-tuning method is used in the second stage to identify effective solutions. The novelty of the proposed HGA lies in the combination of two heuristic stages with new crossover and mutation techniques, and this algorithm is expected to improve the solutions of RCALBA in footwear factories.

The rest of this paper is organized as follows: Sect. 2 discusses the characteristics of the production line, particularly the sewing line, in footwear manufacturing. Section 3 presents a mathematical model of RCALBP. Section 4 introduces the proposed HGA, which is used to solve the problem. Section 5 presents the computational results vis-à-vis those of

the manual process, Kao's heuristic model, and the traditional GA. Section 6 summarizes the conclusions and provides directions for future research.

2 Characteristics of production line in footwear manufacturing

This section introduces the footwear production process. A shoe consists of two main components: the upper part and the sole. The upper part contains all the parts above the sole, such as toe box, vamp, foxing, lining, tongue, and quarter. These components are highly diverse in terms of styles and designs. The upper parts are produced through cutting, preparation, and sewing processes. Sole manufacturing consists of the outsole and midsole processes. The upper and sole parts are assembled in the final assembly process. The sewing line under this study belongs to the manufacturing process of the upper part as a case study, which exhibits the following specific characteristics:

2.1 Workstation arrangement

The completion of each task requires a specific piece of equipment that needs an operator with a particular skill level to operate. A WS is assigned to process a task after the equipment and operator are arranged to perform the assigned tasks. WSs are lined up near and adjacent to each other on the work floor. Workstations may have different sewing machines and operators with varying skill levels. One task cannot be assigned to a WS that does not have the appropriate resources. Only when the required task on a workpiece is completed on the current WS the workpiece can be transferred to the next station. Thus, a sewing line is considered an "un-paced asynchronous line." These requirements are fundamental in designing a sewing line.

2.2 Resource assignment

Different kinds of machines can be arranged in one sewing line, such as shaping or needle stitching machines. An operator with a certain skill level is required to operate a particular machine. Different processes can be assigned to a WS, such as gluing and inserting shoelaces. One station can be assigned to operate only one machine type, and one machine can be assigned to only one WS. This arrangement builds up a one-to-one relationship between WS and machine type.

2.3 Processing times and parallel station

At present, the sewing process still needs human operators because of its limits in automation, although machines are

provided and used. Thus, sewing operating time is regarded as stochastic deviations. The sources of deviations may be one of thousands of possible factors, such as machine setting, environment, and complexity of tasks. In practice, the smoothness of a sewing line is maintained by frequent adjustments when a possible risk of imbalance occurs. Thus, the assumption that processing time is static and deterministic is acceptable. A few tasks have a longer processing time than the production cycle time. These tasks are divided into short tasks with similar precedence constraints and sewing characteristics. Thus, more than one WS is required for the mentioned task.

2.4 Structure of precedence graph

The process of a sewing line typically contains one main branch where the entire task is to assemble components into the main component. All the processes end at one final point, which is usually shoelace insertion (except when the shoe is lace-less).

The sewing process is illustrated by the precedence graph relationship. One task can have more than one precedence task but only one follower. This arrangement ensures that the shape has a convergent look. The shape is usually classified into three categories: sequence, middle, and assembly. The assembly shape has several subgroups, each standing for a set of tasks required to produce the component.

The detailed task description of the sewing line of a simple model is shown in Table 1.

Based on a review of the production line characteristics in footwear manufacturing, the specific sewing line under investigation can be formulated by RCALBP. The next section introduces the mathematical model of RCALBP, which was proposed by Ağpak and Gökçen (2005).

3 Mathematical model

The aforementioned problem is known as RCALBP. We identify the problem in a sewing assembly line based on a decent classification system proposed by a recent study by Boysen et al. (2007), which is presented in Table 2. The problem notation is presented in the three elements $[\alpha|\beta|\gamma]$, where α represents the characteristics of the precedence graph, β denotes the station and line characteristics, and γ indicates the objective function.

The following notations are used for the mathematical model, which is based on the study by Ağpak and Gökçen (2005):

Index: i (task), j (workstation), r (type of resources)

N : number of tasks ($i = 1, \dots, N$)

Table 1 Detailed task description of a sewing line

No.	Task name	Task time	Machine ID	Machine type	Predecessors
1	Press and shape tip	40	2	Shaping machine	
2	Stitch quarter in and quarter out	56	3	Zigzag stitching machine	
3	Stitch collar logo	35	4	Computer stitching machine (small)	
4	Stitch zigzag quarter with collar	47	3	Zigzag stitching machine	2, 3
5	Stitch collar with heel	102	10	Needle stitching machine 2	4
6	Stitch collar lining	38	5	Needle stitching machine 1	5
7	Spray glue and paste vamp	15	7	Spray glue machine	
8	Stitch quarter and vamp	89	5	Needle stitching machine 1	6, 7
9	Stitch quarter and tip	174	5	Needle stitching machine 1	1, 8
10	Edge stitching	42	5	Needle stitching machine 1	9
11	Insert shoelace	137	101	Manual	10

Table 2 Classification of RCALBP

Attribute	Characteristics of problem	Tuple
<i>Precedence graph characteristics</i>		
Product specific	Single product	$\alpha 1 = 0$
Structure of precedence graph	Converging graph	$\alpha 2 = \text{spec}$
Processing time	Static and deterministic	$\alpha 3 = 0$
Sequence-dependent task time increments	Disregarded	$\alpha 4 = 0$
Assignment restrictions	No restriction	$\alpha 5 = 0$
Processing alternatives	Disregarded	$\alpha 6 = 0$
<i>Station and line characteristics</i>		
Movement of workpiece	Un-paced asynchronous	$\beta 1 = \text{unpac}\lambda = 0$
Line layout	Production flow line, not specific	$\beta 2 = 0$
Parallelization	Parallel workstations and tasks One workstation—many tasks	$\beta 3 = \text{pstat}\lambda, \text{ptask}\lambda$
Resource assignment	One workstation—one/no machine	$\beta 4 = \text{res}\lambda$
Station-dependent time increments	Disregarded	$\beta 5 = 0$
Additional aspects of line configuration	Disregarded	$\beta 6 = 0$
<i>Objective</i>		
Objective	Minimized number of workstations	$\gamma = m$

m_{\max} : maximum number of workstations

($j = 1, \dots, m_{\max}$)

R : number of types of resources ($r = 1, \dots, R$)

W_j : set of tasks that can be assigned to station j

C : cycle time

t_i : processing time of task i

E_i : earliest station that task i can be assigned to, given the precedence graph

L_i : latest station that task i can be assigned to, given the precedence graph

P : set of tasks that precedes a task

K_{jr} : set of tasks that can be processed in station j using resource r

$\|K_{jr}\|$: number of tasks in set K_{jr}

Decision variables

$$x_{ij} = \begin{cases} 1: & \text{if task } i \text{ is assigned to workstation } j \\ 0: & \text{otherwise} \end{cases}$$

$$z_j = \begin{cases} 1: & \text{if any task is assigned to workstation } j \\ 0: & \text{otherwise} \end{cases}$$

$$M_{jr} = \begin{cases} 1: & \text{if resource } r \text{ is assigned to workstation } j \\ 0: & \text{otherwise} \end{cases}$$

Objective function

$$\text{Minimum} \quad \sum_{r=1}^R \sum_{j=1}^{m_{\max}} M_{jr} \quad (1)$$

Subject to

$$\sum_{j=E_i}^{L_i} (x_{ij}) = 1, \quad i = 1, \dots, N, \quad (2)$$

$$\sum_{i \in W_j} t_i(x_{ij}) \leq C, \quad j = 1, \dots, m_{\max}, \quad (3)$$

$$\sum_{j=E_a}^{L_a} j(x_{aj}) - \sum_{j=E_b}^{L_b} j(x_{bj}) \leq 0, \quad \forall (a, b) \in P, \quad (4)$$

$$\sum_{i \in K_{jr}} x_{ij} - K_{jr} M_{jr} \leq 0, \quad r = 1, \dots, R, \quad (5)$$

$$\sum_{j=1}^{m_{\max}} M_{jr} = 1, \quad r = 1, \dots, R, \quad (6)$$

$$\sum_{j=1}^{m_{\max}} z_j \leq m_{\max} \quad (7)$$

$$x_{ij}, z_j, M_{jr} \in \{0, 1\} \quad \forall i, j, r$$

The original objective of ALBP is to minimize the number of WS. However, in RCALBP, the objective can be stated as minimizing the number of resources assigned to the WS (Ağpak and Gökçen 2005). The constraints are described as follows: Equation (2) is a constraint of decision variables for the task assignment. The model guarantees that each task is assigned to only one WS. Equation (3) ensures that the station time of WS m does not exceed the cycle time. Equation (4) is a precedence constraint, given that the processing sequence of tasks is subject to precedence restriction. This constraint guarantees that task a cannot be assigned later than task b if task a is a precedence of task b . Equation (5) is subject to the type of resource assignment. This model ensures that if one task is performed in WS j with resource r , the M_{jr} value obtains 1. Equation (6) guarantees that only one type of resource is assigned to each WS. Equation (7) indicates that all tasks have to be assigned to the WS.

As mentioned, RCALBP is an NP-hard problem. Thus, this study develops a heuristic algorithm to solve RCALBP. More information on the proposed algorithm is presented in Sect. 4.

4 Proposed hybrid genetic algorithm

The proposed HGA includes the priority rule-based method (PRBM) in the first stage and the proposed GA in the second stage. PRBM is one of the heuristic approaches to identify the feasible solutions of ALBP. However, PRBM is insufficient in solving large-scale problems. By contrast, GA is a very popular approach for solving relatively large-scale optimization problems. The initial solution in GA is highly

critical because of its effect on the performance of GA. Thus, the integration of PRBM and GA aims to enhance the performance of the proposed method. The first stage employs PRBM to obtain the initial balance, and the second stage uses GA, which adopts the result of the first stage as the initial population. In the GA procedure, the two-point-order crossover with varied crossover length and feasible pattern is designed to ensure good inheritance among the offspring and to prevent unfeasible solutions in a set of offspring. The self-tuning method is applied to remedy the unfeasible solutions caused by the precedence and resource constraints. The mutation process is developed to prevent the building block, which may cause unfeasible solutions. Figure 1 presents the general procedure for the proposed HGA. Essentially, the first stage initially uses PRBM to identify a feasible initial solution to the second stage where GA is used to determine the optimal solution of RCALBP.

4.1 Stage 1: PRBM

PRBM is proposed to identify the feasible solutions of NP-hard optimization problems. Several studies were conducted on the priority rules of ALBP (Helgeson and Birnie 1961; Scholl 1999; Scholl et al. 2010). The priority rules are divided into two groups: minimization rules (high rank for small priority value) and maximization rules (high rank for great priority value). The factors employed to identify the priority rules, including processing time, cycle time, and precedence diagram. The most popular priority rule in ALBP is ranked positional weight (RPW), which was introduced by Helgeson and Birnie (1961). Kao et al. (2011) constructed the RPW rule as an efficient method to solve RCALBP. In their work, the task time divided by latest workstation (TLW) rule is chosen to conduct feasible solutions in PRBM based on a study by Otto and Otto (2014). This priority rule is selected for two reasons. First, the TLW rule is in a group that depends on the information on both task time and precedence relations. The TLW rule is already aggregated in the latest workstation rule. Second, TLW is in the group of five priority rules that were proven to potentially perform in terms of computation efficiency.

Aside from the notations used in the mathematical model, a few more notations are introduced to describe the heuristic approach in PRBM, including the total time of a WS ($t(S_m)$), set of feasible tasks assigned to (FS), set of assigned tasks (A), set of unassigned tasks (U), set of feasible tasks operated by a similar type of resource in WS m (WR_m), set of feasible tasks with predecessors assigned to previous WS $m - 1$ and using similar types of resource as those of predecessors (WR_{ir}), and set of feasible tasks with predecessors assigned to previous WS $m - 1$ but not using a similar type of resource as those of predecessors (PUS_{ir}). The procedure is shown in Fig. 2.

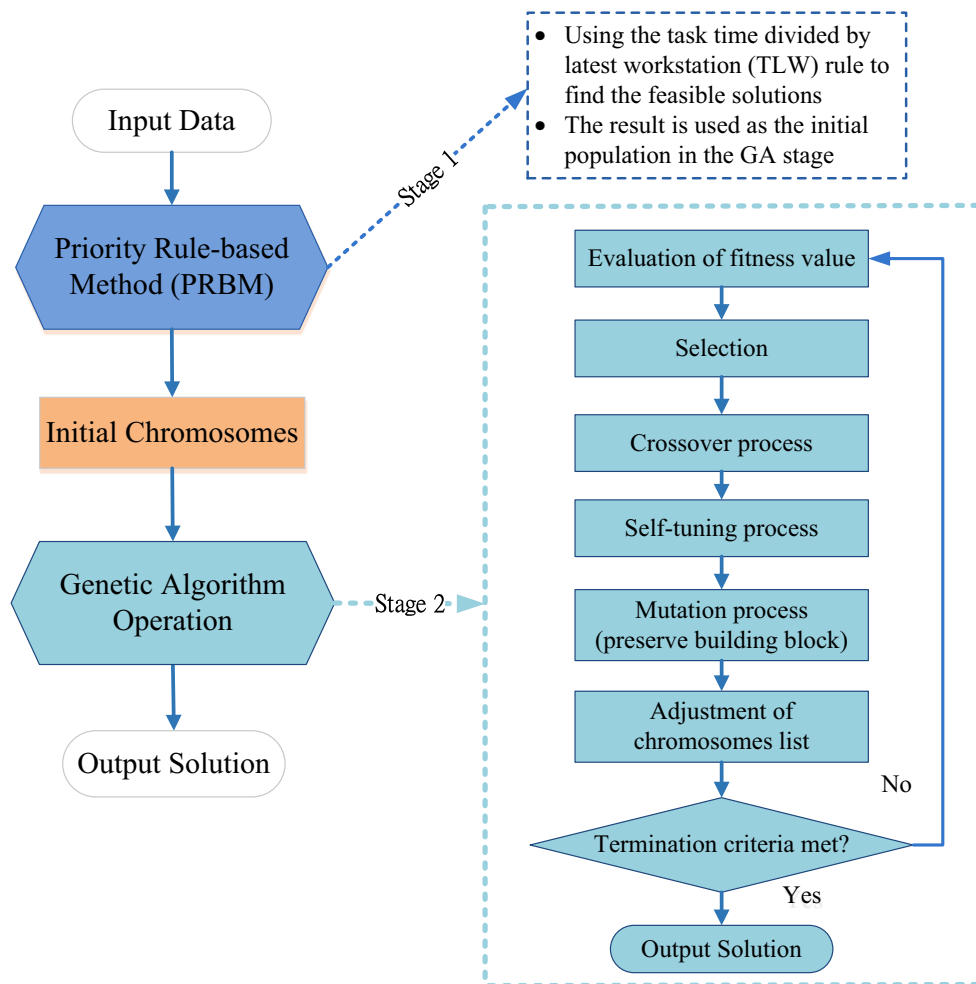


Fig. 1 HGA procedure

4.2 Stage 2: Proposed GA

4.2.1 Encoding and decoding

Choosing the chromosome representation is the first and most important step in forming the GA. Chromosome representation clearly describes the real problem and affects the diversity of the quality of the solutions. Goldberg (1989) introduced several methods to construct chromosome representations that were applicable in ALBP, such as WS-oriented, sequence-oriented, and partition-oriented representation. In the present study, sequence-oriented representation is selected because this method offers a number of advantages. First, this representation is adequate for ALBP, whose number of WS is not pre-delimited. Moreover, this representation is more tractable in the genetic operator because all tasks are arranged in the order of their allocations to the WS. The number of genes in the chromosome is defined by a number of tasks.

Figure 3 illustrates the encoding and decoding of the chromosome expression. We suppose that 10 tasks are assigned to seven WSs. The chromosome representation is {1 1 3 2 7 4 5 5 6 4}, which means that tasks {1, 2} are assigned to WS 1, task {4} to WS 2, task {3} to WS 3, tasks {6, 10} to WS 4, tasks {7, 8} to WS 5, task {9} to WS 6, and task {5} to WS 7.

4.2.2 Initial population

Initial solutions significantly affect the quality of the solution in heuristic approaches. In the proposed GA, the result from the first stage is used to construct the initial population. When the produced initial solutions are more diverse, heuristic approaches that perform better can be obtained. In this study, the proposed PRBM heuristic with TLW rule can enhance the initial population diversity by increasing the number of feasible solutions. First, the heuristic PRBM con-

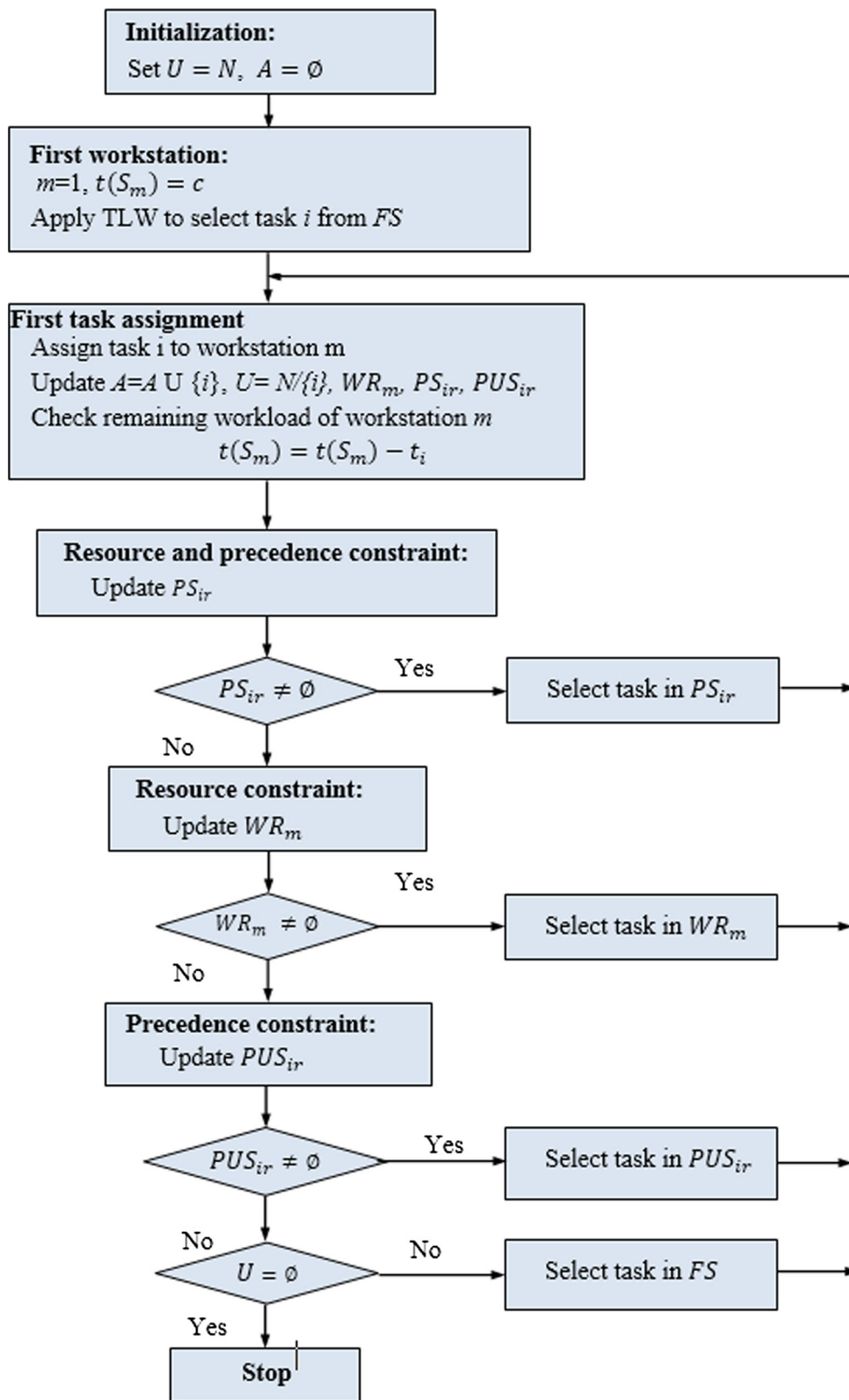


Fig. 2 PRBM procedure

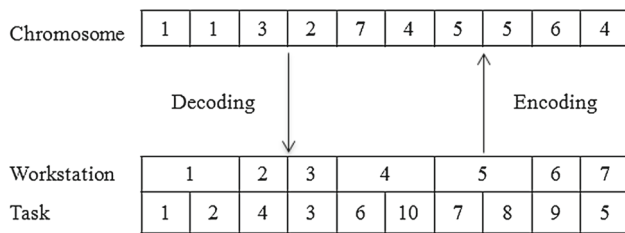


Fig. 3 Encoding and decoding of chromosome expression in HGA

siders the tasks without a predecessor task in the precedence relation. Each task from the set of tasks without a predecessor task is sequentially selected as the first task to assign to WS. The remaining tasks are assigned based on the TLW rule. The number of feasible solutions can be equal to the number of tasks in the set of tasks without a predecessor task.

All the feasible solutions generated are employed to construct the initial population. The quality of feasible solutions that are evaluated based on the fitness value is considered to identify the number of initial elements that should be generated from each feasible solution. The initial elements are constructed according to the probability distribution of fitness values. The initial elements from each feasible solution are calculated by multiplying the number of population by the corresponding probability of each feasible solution. For example, PRBM found three feasible solutions at the first stage. The fitness values of the three feasible solutions are 8, 10, and 11, and the corresponding probabilities are 27.6, 34.5, and 37.9%, respectively. If the population size is 150, the initial elements from each feasible solution are 41, 52, and 57, correspondingly.

4.2.3 Fitness function

A fitness value is a function that identifies how well an allocation performs. Only one machine can be assigned to one WS. Thus, minimizing the number of resources is equivalent to minimizing the number of WS. With the given cycle time, a smaller number of WS produces a better result than a larger one. Thus, chromosomes with a small fitness value have a high probability of being chosen in the next generation. However, chromosomes with recessive genes can produce a particular feature in a child in a few cases. If we choose only the best solution from an initial population, we can omit the chromosome with recessive genes. By defining the adjusted fitness value (AFV) to adjust the number of WS, we can increase the probability of an offspring being chosen for the next generation.

$$\text{AFV} = m + \eta, \quad (8)$$

where m , number of workstations; η , line balance efficiency

$$\eta = \frac{\text{Total task time}}{\text{No. of stations} \times \text{Cycle time}} = \left(\sum_{i=1}^N t_i \right) / (m \times C) \quad (9)$$

The line balance efficiency measures the percentage of WS time, which is used effectively to assign the appropriate tasks to the WS. According to the formulation of AFV, η becomes larger as m becomes smaller and vice versa. Thus, the chromosomes with a large fitness value may not be chosen in the next generation, which can be selected after performing the AFV.

4.2.4 Selection

Selection in GA is a procedure of selecting individuals for reproduction. The two most popular methods are roulette wheel and tournament selection. In the roulette wheel method, the selection of parents is based on the probability distribution of the AFV. Tournament selection includes running a few tournaments among a few individuals randomly chosen from the population and selecting the winner. The roulette wheel method is chosen, and its procedure is based on AFV.

4.2.5 Crossover operator: two-point-order crossover

The crossover phase is the main genetic operator in GA. This phase mates two chromosomes to produce the next generation using the information of the parents. The parent chromosomes are selected in the selection procedure. This step encounters difficulties in designing a favorable crossover operator to mix the fitting characteristics of the parent solutions into valid offspring solutions.

The traditional one-point or two-point crossover can generate an unfeasible solution because it disregards the precedence and resource constraints. To expend the possibility of feasible patterns inherited from the parent chromosomes, the two-point crossover is used and a new method of choosing crossover points is proposed in this study. The new crossover procedure includes two steps. The first step is searching the two crossover points from two parents. Second, the pattern of genes within crossover points is reordered to create diverse solutions. The resources and precedence constraints are checked, and the self-tuning method is applied to guarantee that the solutions of the new chromosome are feasible.

In the proposed method, all possible crossover points are evaluated. The chromosomes that violate the precedence and resource constraints are excluded. The procedure of

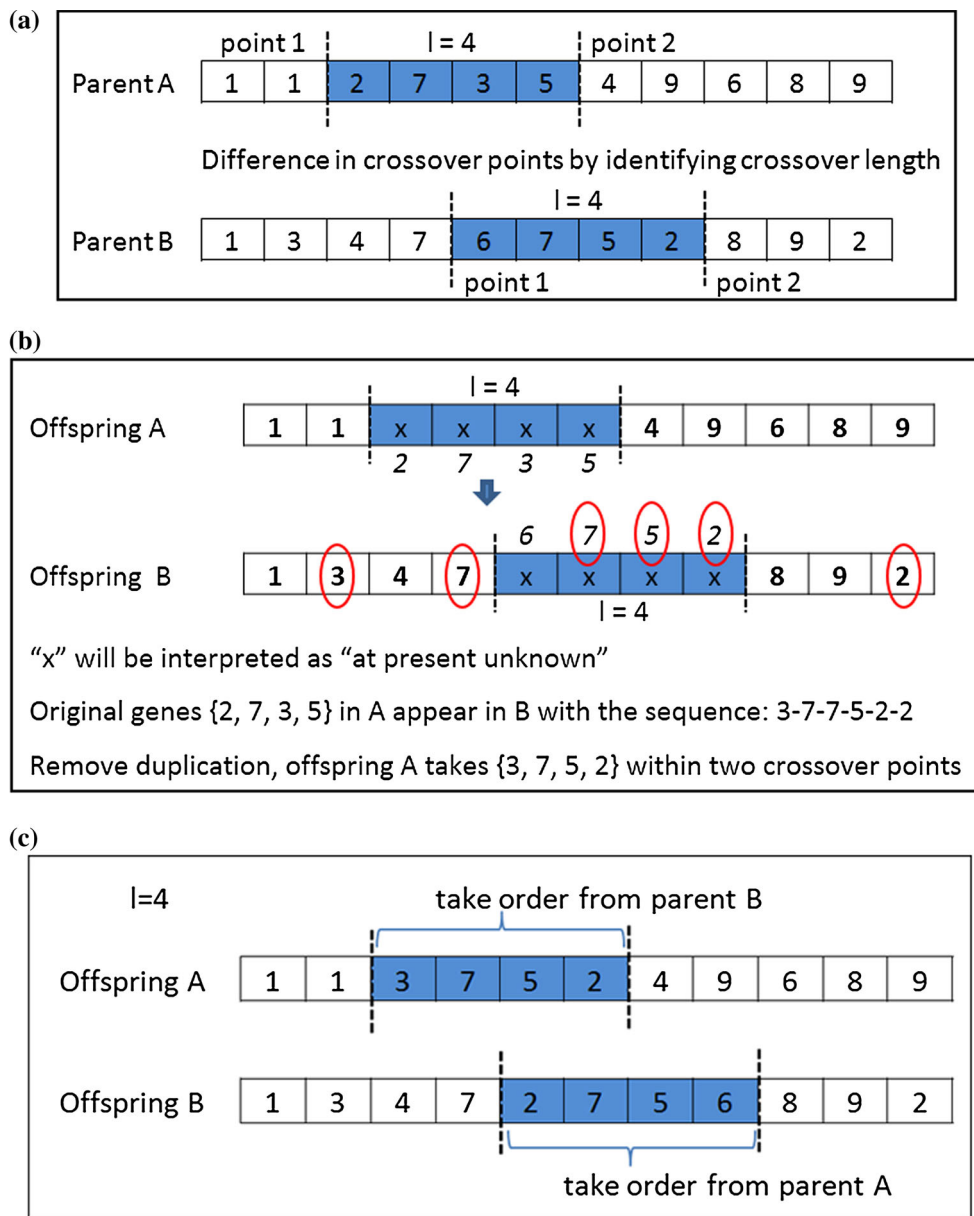


Fig. 4 Crossover procedure in proposed HGA

choosing the crossover points is demonstrated in Fig. 4a. A few definitions for the crossover procedure are listed as follows:

- u : index of gene in parent A;
- v : index of gene in parent B;
- l : crossover length; $l = 2 : [g - 2]$ with g : number of genes.

The process starts with the first gene in parent A ($u = 1$) and crossover length $l = 2$. All the feasible crossover lengths in parent B are checked and placed in the feasible crossover set. The crossover length is increased, and the process in parent B is repeated. The loop is repeated until the length

reaches the last gene in parent A, and the next gene in A is chosen.

For $u = 1$ (start with the first point in parent A)

$l = 2$ (the shortest length (initial value))

$v = l, \dots, g$

Reject all the chromosomes that violate the precedence and resource constraints

The remaining solutions are in a set of feasible pattern crossover

$l = l + 1$ (increase crossover length)

$u = u + 1$

End

The crossover procedure includes the following steps:

- Step 1* Select two chromosomes using the roulette wheel method.
- Step 2* Select two crossover points from a set of feasible pattern crossovers. Thus, all cases are placed in the feasible pattern. The procedure will scan all and pick the best one.
- Step 3* Do crossover.
- The offspring maintain the part on the left and right sides of the cross points.
 - For offspring A, choose the genes within two crossover points of parent B, and preserve the relative order. Offspring A inherits all the genes within two crossover points of parent A with the order derived from parent B.
 - Repeat the same procedure for offspring B.

Figure 4a illustrates the two crossover points in different locations with the crossover length that is selected from the feasible pattern crossover equal to 4 ($l = 4$) as an example. The procedure of taking the gene's order is described in Fig. 4b. Two selected parents are denoted as {1 1|2 7 3 5|4 9 6 8 9} and {1 3 4 7|6 7 5 2|8 9 2} with the cut-point for $l = 4$. For offspring A and B, "x" indicates the genes waiting for the crossover process. Thus, offspring A and B are temporarily presented as {1 1|x x x x|4 9 6 8 9} and {1 3 4 7|x x x x|8 9 2}, respectively. Based on the genes within two crossover points, the waiting genes of offspring A are determined by identifying the sequence of these genes in parent B and vice versa. In this case, the genes {2}, {7}, {3}, and {5} appear as sequentially at the sequence 3-7-7-5-2-2 in parent B ({1 3 4 7|6 7 5 2|8 9 2}). Note that the sequence 3-7-7-5-2-2 has some repeated numbers because two tasks are assigned in the same WS in parent B. After the duplication in 3-7-7-5-2-2 is removed, the sequence is 3-7-5-2. Offspring A takes this sequence and fills in the "x" spots as {1 1|3 7 5 2|4 9 6 8 9} in Fig. 4c. Based on the same procedure, the "x" spots in offspring B can be filled with 2-7-5-6, which is the presenting sequence of 6-7-5-2 in parent A. The offspring after crossover are presented in Fig. 4c.

4.2.6 Self-tuning method

The self-tuning method is applied to correct the unfeasible solutions caused by the constraints. Chen et al. (2002) used the self-tuning method in multi-objective assembly planning. In their study, the self-tuning method was employed after generating the initial solution to fix the unfeasible solutions. However, in the present study, the initial solutions are already feasible because they are derived from PRBM. Thus, the self-tuning method is used to check the unfeasible solutions after

conducting crossover. The procedure of this method is shown in Fig. 5. To validate the chromosome, the genes are checked whether they satisfy the constraints or not. If they satisfy the constraints and the corresponding tasks have not been assigned to WS, the corresponding tasks are assigned to WS and recorded in the correct gene record. The genes that do not satisfy the constraints are placed in the backlog record for further checking. After considering all the genes that satisfy the constraints, the genes in the backlog record are checked. If no gene is listed in the backlog record, the correct gene record returns the feasible chromosomes. If the backlog record is not empty, the algorithm checks the constraints of genes in the backlog record. The corresponding tasks of the genes that satisfy the constraints are assigned to the WS. The assigned genes are removed from the backlog record, and the result is stored in the correct gene record. This process is repeated recursively until all the chromosomes are finished.

4.2.7 Mutation operator

The mutation operation can create new feasible solutions and reproduce offspring. This operation can be used to increase the diversity of the population and prevent local optimal solutions. The mutation procedure is modified to avoid breaking building blocks and to prevent unfeasible solutions.

The mutation procedure includes the following steps, as shown in Fig. 6:

- Identify a set of available elements to conduct the mutation (SA).
- Randomly select the first element from a set of available elements according to the selection procedure.
- Identify a set of building block elements (SBB) based on the constraints.
- Set of mutation feasible elements = {set of available tasks}/(set of building block tasks) (SMF = {SA}/{SBB}).
- Randomly select the second element from a set of mutation feasible elements.
- Generate a random number $r \sim U(0, 1)$.
- If $r < p_m$ (mutation rate): swap the two elements.

To enhance the exploitation of local optima, 10% of the population is replaced by very good solutions when the iteration exceeds 90% of the total iterations, allowing a dense search around the very best solutions of the population.

The proposed HGA method was performed to test real-world data from a footwear manufacturer in China. The experimental result of HGA was compared with the results of the manual method applied on site, Kao's bidirectional heuristic method, which is the representative method for solving RCALBP in the footwear assembly line, and the traditional GA with conventional heuristic procedure. In the

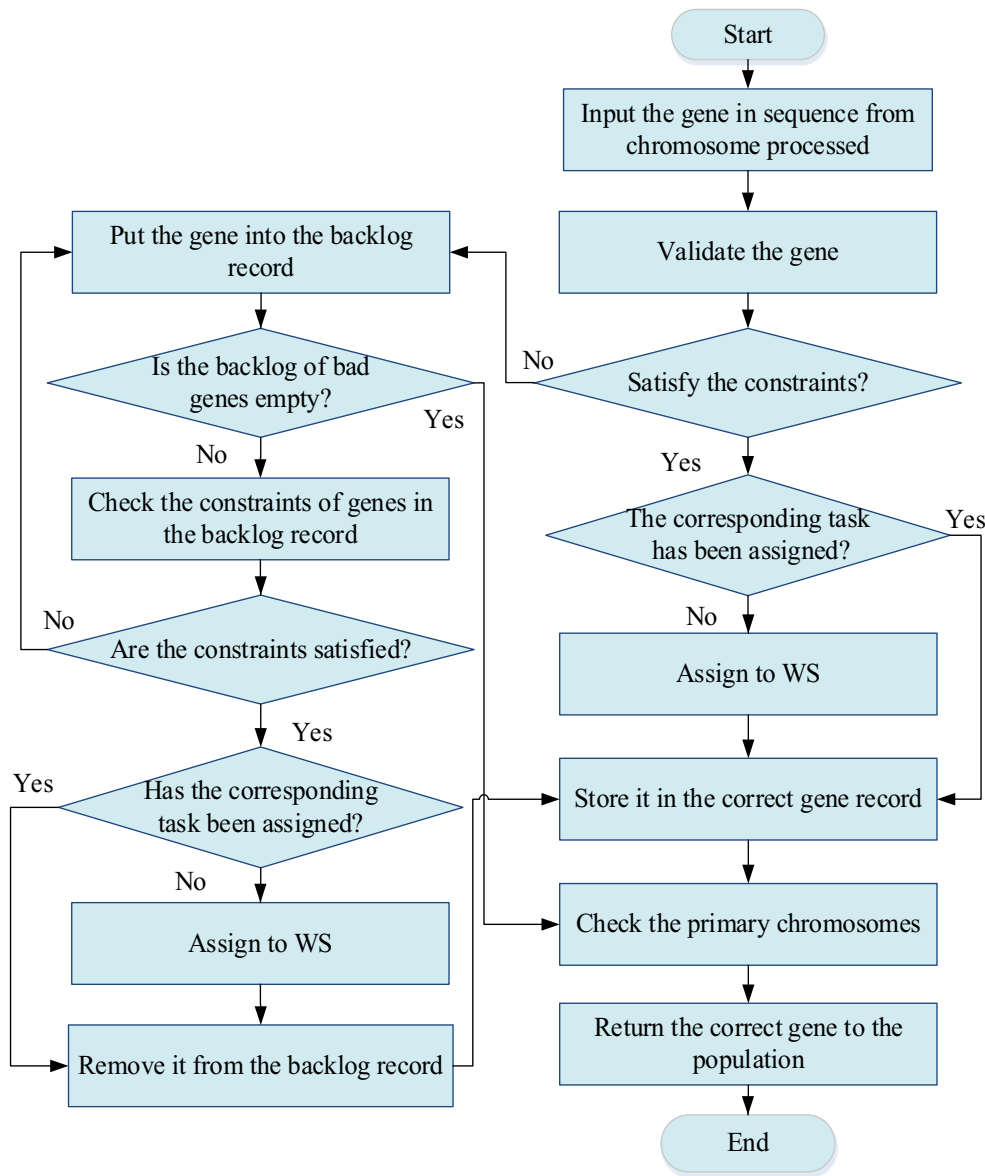


Fig. 5 Self-tuning method to correct the unfeasible solutions

following section, additional details on the results of the comparison are discussed.

5 Result and discussion

In this study, all the methods were coded in the C# programming language and run on a computer with Intel Core i7 and a 2.93-GHz processor with 4GB RAM. The initial parameter setting for the proposed HGA was set as follows: generation = 100, population = 100, crossover rate = 0.8, and mutation rate = 0.1.

Two factors, namely problem size and shape of the precedence graph, were adopted in the experiments to compare

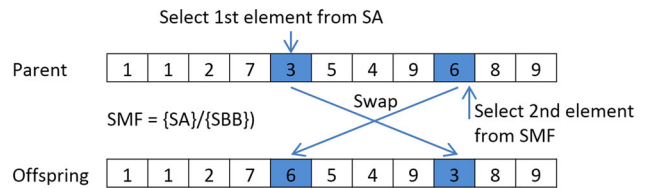


Fig. 6 Mutation procedure in proposed HGA

the quality of algorithm and computational efficiency. These two factors were categorized according to manufacturing processes in actual situations. The problem size in terms of the number of tasks can be divided into three main categories: small, medium, and large scale. The small-scale problems deal with operations involving children’s shoes or slippers

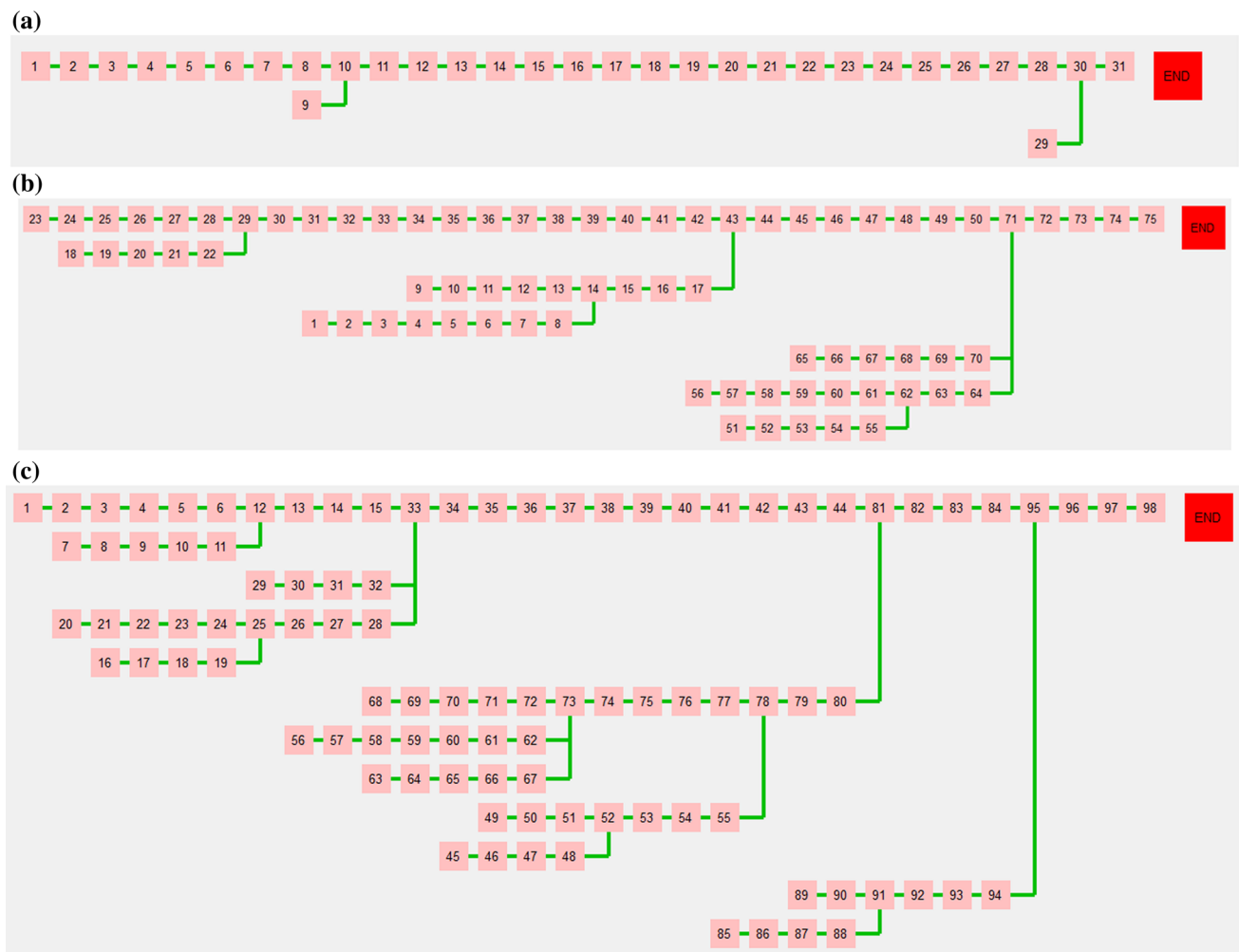


Fig. 7 Precedence graph of assembly line

with approximately 30 tasks. The medium-scale problems are for running shoes or outdoor-activity shoes with approximately 70 tasks. The large-scale problems include functional shoes with up to 100 complicated design tasks. The precedence graphs were divided into three categories: sequence, middle, and assembly. The shapes of the precedence graphs demonstrate the differences and complexity in the assembly processes (Fig. 7).

The optimal solutions of the problems are unknown. To evaluate the efficiency of the proposed HGA, a comparison was conducted between the proposed HGA and (1) the current manual procedure in the footwear factory, (2) the existing Kao's heuristic model, which was adjusted to meet the resource constraint requirement of the mentioned problems, and (3) the traditional GA without considering feasible solution remedy. The proposed HGA was compared with the traditional GA based on the work of [Chen et al. \(2002\)](#). The experiments were first tested in the traditional GA where all the procedures were processed conventionally without the

self-tuning method. The improvement in the quality of solution using the proposed HGA was addressed.

A total of 18 models, which were derived from the real-world sewing line of an athletic and casual footwear manufacturer in southern China, were used in the experiments. The experiment was repeated 30 times for each model, and the average of the best solution was considered to represent the result. The initial parameter settings for GA were mentioned at the beginning of this section. The number of generations was adopted as the termination criteria of both traditional GA and HGA. Table 3 indicates the representative results of the experiments in terms of WS of these methods.

The number of WS and the relative deviation (RD) were used to measure the improvement of the proposed HGA relative to that of the manual process, Kao's heuristic model, and the traditional GA. RD was calculated based on the differences between the two methods to demonstrate the improvement in percentage points. The higher RD value

Table 3 Experimental results of number of WS

Problem size	No. of task	Shape	Workstation (WS)					HGA versus manual		HGA versus Kao's heuristic		HGA versus traditional GA	
			Manual process (a)	Kao's heuristic (b)	Traditional GA (c)	HGA (d)	WS (a-d)	RD (%)	WS (b-d)	RD (%)	WS (c-d)	RD (%)	
Small	28	Sequence	34	33	33	33	1	2.9	0	0.0	0	0.0	
Small	31	Sequence	35	35	35	35	0	0.0	0	0.0	0	0.0	
Small	33	Middle	35	35	35	34	1	2.9	1	2.9	1	2.9	
Small	27	Middle	33	31	31	31	2	6.1	0	0.0	0	0.0	
Small	32	Assembly	42	42	42	41	1	2.4	1	2.4	1	2.4	
Small	29	Assembly	32	30	30	29	3	9.4	1	3.3	1	3.3	
Medium	67	Sequence	82	80	82	80	2	2.4	0	0.0	2	2.4	
Medium	71	Sequence	81	81	81	80	1	1.2	1	1.2	1	1.2	
Medium	65	Middle	77	76	79	68	9	11.7	8	10.5	11	13.9	
Medium	75	Middle	87	87	85	76	11	12.6	11	12.6	9	10.6	
Medium	74	Assembly	78	78	84	68	10	12.8	10	12.8	16	19.0	
Medium	68	Assembly	75	74	77	66	9	12.0	8	10.8	11	14.3	
Large	101	Sequence	125	125	128	124	1	0.8	1	0.8	4	3.1	
Large	98	Sequence	112	112	110	108	4	3.6	4	3.6	2	1.8	
Large	106	Middle	122	121	125	108	14	11.5	13	10.7	17	13.6	
Large	108	Middle	134	127	135	119	15	11.2	8	6.3	16	11.9	
Large	98	Assembly	95	94	111	86	9	9.5	8	8.5	25	22.5	
Large	100	Assembly	110	103	114	100	10	9.1	3	2.9	14	12.3	

results in greater improvement of the proposed HGA. The formula for calculating RD is as follows:

$$RD = \frac{(m_X - m_Y)}{m_X} \times 100\%, \quad (10)$$

where m_X is the number of WS of the existing method and m_Y is the number of WS of the proposed method.

According to the results in Table 3, HGA outperforms other methods in terms of WS and RD. However, the improvements vary among the comparisons of HGA with different methods. Thus, the comparisons between HGA and other methods are analyzed as follows:

First, based on the comparison of the manual process with the proposed HGA, HGA yields improvements on 17 models with a total of 103 WS improvements in 18 tested models. The improvements are more significant in more complicated models, such as the medium-scale and large-scale problem with a complex precedence graph. In terms of RD, the proposed HGA achieves the best performance at 12.8% improvement for the medium-scale problem and assembly-shaped model on average.

Second, the result shows that the HGA performs better than Kao's heuristic model in terms of WS and RD values. HGA achieves an improvement of 78% of the total models (14/18 models) with 78 WS improvement in total. Similar to the preceding comparison, the improvements obtained by HGA are more significant on the tested models that combine the medium and large scales with middle and assembly shape. The best performance is at 12.82% improvement in terms of RD in one model, which combines the medium scale and assembly graph. In terms of saving WS, the proposed HGA achieves the best performance at the large-scale problem and assembly-shaped model where 13 WSs are saved in total.

Finally, the results of the HGA and traditional GA were compared. As mentioned, this comparison aims to determine the improvement of the proposed HGA compared with the conventional heuristic GA. According to the results shown in Table 3, the proposed HGA is enhanced in 15 models, saving a total of 131 WSs compared with the traditional GA. This improvement by HGA is significant in the models that combine medium-scale and large-scale problems with middle-shaped and assembly-shaped models.

According to the experimental results, the proposed HGA exhibited significant improvement on medium- and large-scale problems. Table 3 indicates that among the six small-scale models, HGA can obtain better solutions in only three small-scale models with the best RD at 3.33%. By contrast, a significant improvement can be achieved by the proposed HGA in terms of WS and RD for medium- and large-scale problems. The HGA exhibits the best performance in the model with large scale and assembly shape at 22.5% improvement in terms of RD and 25 WSs.

Figure 8 shows the differences of HGA and the three methods in terms of WS based on the two mentioned factors. A larger difference implies greater improvement. HGA exhibits better improvement for the medium- and large-scale problem than for the small problem. This outcome is reasonable because small problems are almost optimal, and achieving significant improvement is more difficult than that in medium- and large-scale problems. The HGA can achieve the best performance with the large-scale problems. Similarly, in terms of the shapes of precedence graphs, the HGA works more effectively on the middle and assembly graphs. The best improvement can be obtained using a model that has a precedence graph with a middle shape.

The computational time is also compared in Table 4. The manual model and Kao's heuristic model can produce the feasible solutions in seconds, whereas the traditional GA and HGA take a longer time to obtain the solutions. This result is reasonable because Kao's heuristic model is a simple heuristic procedure based on the priority rule. Although Kao's method is faster, the solution quality is compromised. HGA and the traditional GA require the longest time to obtain the results because of the conventional heuristic procedure. However, the quality of improvement of the conventional heuristic procedure is significantly better than that using Kao's method. HGA has significantly better computational time than the traditional GA because generating initial solutions, designing crossover and mutation, and correcting unfeasible solutions are improved. The traditional GA searches the entire feasible space. Thus, it may generate unfeasible solutions without any remedy, such as applying self-tuning method to correct them. These unfeasible solutions can entail additional time in reproducing the chromosomes for the next generation. In general, HGA takes around 1 min to produce solutions for the medium-scale problem. The two most complicated large-scale problems can also be solved at around 600s (10 min) by HGA. This computational time is acceptable on site because the current manual method takes more than 30 min for the large-scale problems without any guarantee in terms of the quality of the solutions.

The experimental results demonstrate that the proposed HGA outperforms the manual procedure, Kao's heuristic model, and the traditional GA in terms of the quality of the solutions. The manual procedure is currently used by factories to balance the workload on the sewing line. Although the rule of the manual method is simple, this method takes a longer time to obtain the results without guaranteeing the quality of the results. Kao's heuristic model seems to exhibit a slight improvement compared with the manual procedure. The traditional GA with the conventional heuristic procedure does not seem to provide better solutions. The proposed HGA with the novelty outperforms other methods with an acceptable running time in generating the initial solu-

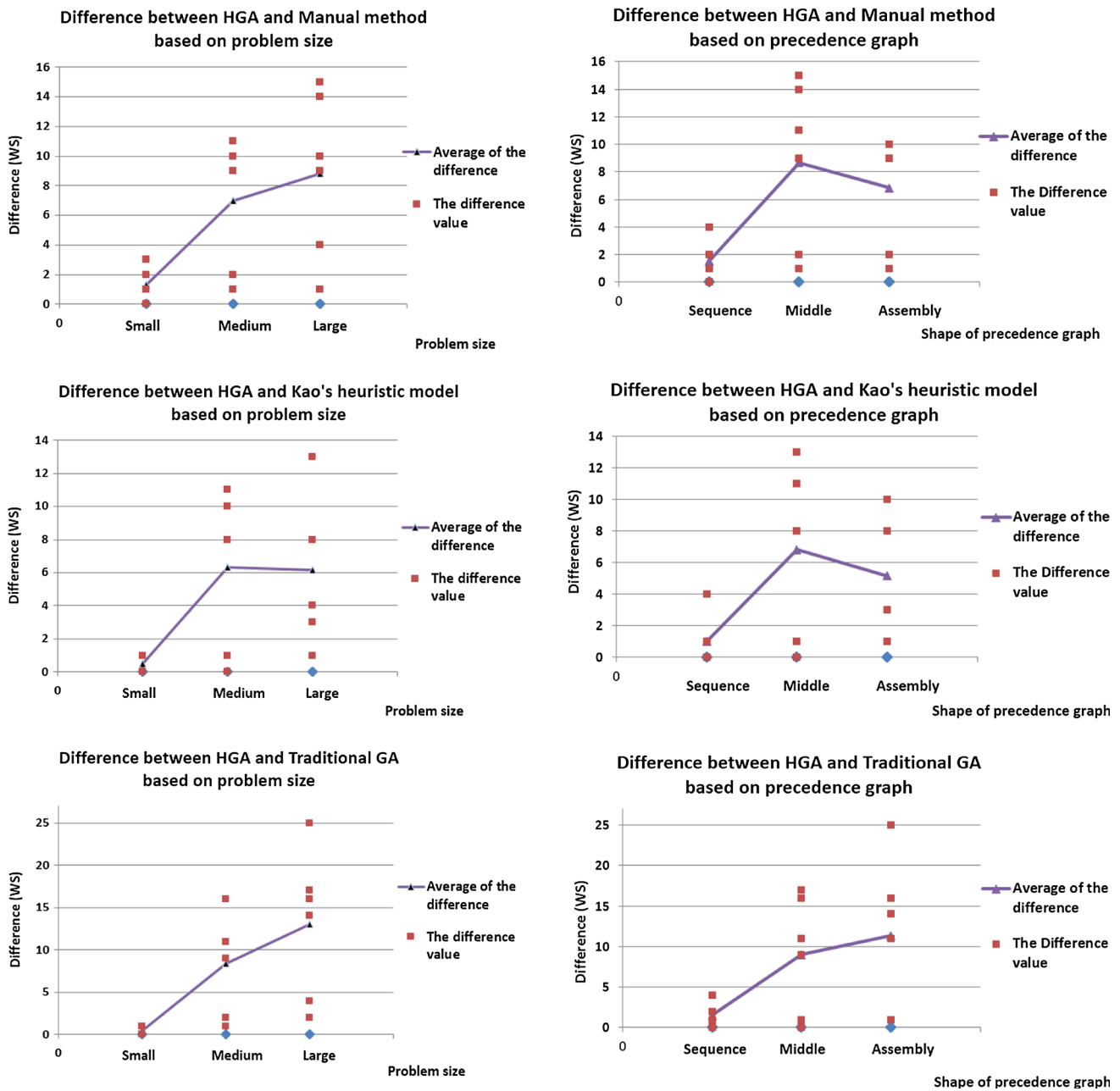


Fig. 8 Difference between HGA and other models based on size of problem and precedence graph

tions, designing the crossover and mutation procedure, and correcting the unfeasible solutions by using the self-tuning method.

The contribution to the manufacturing process using the proposed HGA is critical. With a given cycle time, reducing the number of WS implies that the WSs perform the tasks with the better-balanced loading. Labor time is also maximized on the field. Saving a few seconds for each item in the process can definitely reduce labor requirements because of the large-scale throughput with thousands of shoes produced per day in the factory. Reduced labor caused by the improved

load balance eventually leads to significant cost reduction by accumulating the total reduction of manufacturing time.

6 Conclusion and future research directions

This study investigates RCALBP in the sewing line in footwear manufacturing. The problem considers not only the constraints of WS capacity and precedence graph but also the resources. The main innovation of this study is that the proposed HGA combines the PRBM and GA with

Table 4 Experimental result of computational time

Size category	No. of task	Shape	Computation time (in s)		
			Kao's heuristic	Traditional GA	HGA
Small	28	Sequence	0.8	21.6	0.5
Small	31	Sequence	0.7	24.1	0.5
Small	33	Middle	0.8	24.5	0.5
Small	27	Middle	0.9	19.0	0.6
Small	32	Assembly	0.7	22.7	1.2
Small	29	Assembly	0.7	20.8	1.2
Medium	67	Sequence	1.9	83.5	32.3
Medium	71	Sequence	1.9	99.3	51.8
Medium	65	Middle	1.3	80.1	36.1
Medium	75	Middle	1.6	99.6	45.5
Medium	74	Assembly	1.6	178.6	68.9
Medium	68	Assembly	2.1	187.0	55.5
Large	101	Sequence	4.4	308.3	111.0
Large	98	Sequence	2.8	299.4	203.6
Large	106	Middle	4.4	346.6	256.3
Large	108	Middle	5.4	355.7	268.9
Large	98	Assembly	5.5	620.4	587.4
Large	100	Assembly	6.1	689.5	600.2

redesigned crossover and mutation techniques. The results proved that this proposed HGA could provide improved solutions for RCALBP in the sewing line of an athletic and casual footwear manufacturer. The PRBM result is used to generate the initial population that can determine better starting points for further GA. Moreover, the proposed GA develops a new technique to search the two crossover points at different gene locations on two chromosomes. All feasible crossover cases are placed in the feasible pattern, and the selection for crossover can choose the best pattern. Besides, the self-tuning method is applied to correct unfeasible solutions quickly. The mutation operator is redesigned to avoid breaking a building block that may cause unfeasible solutions. Finally, this study was applied on the sewing line of the athletic and casual footwear factory of Pou Chen International Group, which is one of major footwear manufacturers in the world. The experimental results show that the proposed HGA outperforms the current manual procedure in the factory, the existing Kao's heuristic model, and the traditional GA in terms of quality of solution (WS and RD) with acceptable computation time. HGA could provide significantly improved results, particularly for a combination of medium-scale or large-scale problems with middle type or assembly type of precedence graph.

Some suggestions for future research are as follows: First, the better priority rules in the first stage should be further studied because the initial solutions have an important role in identifying good solutions in heuristic approaches. To diversify the initial population, the initialization stage can consider

including the solutions generated by other heuristics. Moreover, the objective function can be extended using multiple objective functions with the stochastic model in terms of processing time. In addition, RCALBP can be enlarged with additional resource constraints, such as types of tools and labor skills.

Acknowledgements This study was supported by the National Science Council of Taiwan, ROC (Contract No. NSC 102-2221-E-007-123-MY3), and Pou Chen International Group.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

- Ağpak K, Gökçen H (2005) Assembly line balancing: two resource constrained cases. *Int J Prod Econ* 96:129–140
- Azadeh A, Sangari MS, Sangari E, Fatehi S (2015) A particle swarm algorithm for optimising inspection policies in serial multistage production processes with uncertain inspection costs. *Int J Comput Integr Manuf* 28:766–780
- Bautista J, Pereira J (2009) A dynamic programming based heuristic for the assembly line balancing problem. *Eur J Oper Res* 194:787–794
- Baybars İ (1986) A survey of exact algorithms for the simple assembly line balancing problem. *Manag Sci* 32:909–932
- Becker C, Scholl A (2006) A survey on problems and methods in generalized assembly line balancing. *Eur J Oper Res* 168:694–715
- Boysen N, Fließner M, Scholl A (2007) A classification of assembly line balancing problems. *Eur J Oper Res* 183:674–693

- Boysen N, Fliedner M, Scholl A (2008) Assembly line balancing: which model to use when? *Int J Prod Econ* 111:509–528
- Cano-Belmán J, Ríos-Mercado R, Bautista J (2010) A scatter search based hyper-heuristic for sequencing a mixed-model assembly line. *J Heuristics* 16:749–770
- Chen JC, Chen C-C, Su L-H, Wu H-B, Sun C-J (2012) Assembly line balancing in garment industry. *Expert Syst Appl* 39:10073–10081
- 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* 15:447–457
- Chica M, Cordon Ó, Damas S, Bautista J (2015) Interactive preferences in multiobjective ant colony optimisation for assembly line balancing. *Soft Comput* 19:2891–2903
- Delice Y, Kızılkaya Aydoğan E, Özcan U, İlkay M (2014) A modified particle swarm optimization algorithm to mixed-model two-sided assembly line balancing. *J Intell Manuf*. doi:10.1007/s10845-014-0959-7
- Ghosh S, Gagnon RJ (1989) A comprehensive review and analysis of the design, balancing and scheduling of assembly systems. *Int J Prod Res* 27:637–670
- Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Publishing Company Inc, Boston
- Helgeson WB, Birnie DP (1961) Assembly line balancing using the ranked positional weight technique. *J Ind Eng* 12:394–398
- Johnson RV (1988) Optimally balancing large assembly lines with FABLE. *Manag Sci* 34:240–253
- Kao H-H, Yeh D-H, Wang Y-H (2011) Resource constrained assembly line balancing problem solved with ranked positional weight rule. *Rev Econ Finance* 1:71–80
- Kim YK, Kim YJ, Kim Y (1996) Genetic algorithms for assembly line balancing with various objectives. *Comput Ind Eng* 30:397–409
- Levitin G, Rubinovitz J, Shnits B (1995) Genetic algorithm for assembly line balancing. *Int J Prod Econ* 41:343–354
- Marketline (2014) Footwear: global industry guide. Research and Markets, Dublin
- Otto A, Otto C (2014) How to design effective priority rules: example of simple assembly line balancing. *Comput Ind Eng* 69:43–52
- Özcan U, Toklu B (2009) A tabu search algorithm for two-sided assembly line balancing. *Int J Adv Manuf Technol* 43:822–829
- Rada-Vilela J, Chica M, Cordon Ó, Damas S (2013) A comparative study of multi-objective ant colony optimization algorithms for the time and space assembly line balancing problem. *Appl Soft Comput* 13:4370–4382
- Rahimi-Vahed AR, Mirghorbani SM, Rabbani M (2007) A new particle swarm algorithm for a multi-objective mixed-model assembly line sequencing problem. *Soft Comput* 11:997–1012
- Rekiek B, De Lit P, Pellichero F, L'Eglise T, Fouda P, Falkenauer E, Delchambre A (2001) A multiple objective grouping genetic algorithm for assembly line design. *J Intell Manuf* 12:467–485
- Roshani A, Fattahi P, Roshani A, Salehi M, Roshani A (2012) Cost-oriented two-sided assembly line balancing problem: a simulated annealing approach. *Int J Comput Integr Manuf* 25:689–715
- Roshani A, Roshani A, Roshani A, Salehi M, Esfandyari A (2013) A simulated annealing algorithm for multi-manned assembly line balancing problem. *J Manuf Syst* 32:238–247
- Sabuncuoglu I, Erel E, Tanyer M (2000) Assembly line balancing using genetic algorithms. *J Intell Manuf* 11:295–310
- Salveson ME (1955) The assembly line balancing problem. *J Ind Eng* 6:18–25
- Scholl A (1999) Balancing and sequencing of assembly lines. Publications of Darmstadt Technical University, Institute for Business Studies (BWL), Darmstadt
- Scholl A, Fliedner M, Boysen N (2010) Absalom: Balancing assembly lines with assignment restrictions. *Eur J Oper Res* 200:688–701
- Scholl A, Klein R (1997) SALOME: a bidirectional branch and bound procedure for assembly line balancing. *Inform J Comput* 9:319–334
- Sprecher A (1999) Competitive branch-and-bound algorithm for the simple assembly line balancing problem. *Int J Prod Res* 37:1787–1816
- Tasan SO, Tunali S (2008) A review of the current applications of genetic algorithms in assembly line balancing. *J Intell Manuf* 19:49–69
- Triki H, Mellouli A, Masmoudi F (2014) A multi-objective genetic algorithm for assembly line resource assignment and balancing problem of type 2 (ALRABP-2). *J Intell Manuf*. doi:10.1007/s10845-014-0984-6
- Zha J, Yu J-J (2014) A hybrid ant colony algorithm for U-line balancing and rebalancing in just-in-time production environment. *J Manuf Syst* 33:93–102