Int J Adv Manuf Technol (2014) 70:1775–1786
DOI 10.1007/s00170-013-5425-8

ORIGINAL ARTICLE

Product family assembly line balancing based on an improved genetic algorithm

Liang Hou · Yong-ming Wu · Rong-shen Lai · Chi-Tay Tsai

Received: 20 April 2013 / Accepted: 15 October 2013 / Published online: 31 October 2013

Abstract Product family assembly line (PFAL) is a mixed-model assembly line on which a family of similar products can be assembled at the same time. Aiming at the balance problem of PFAL, a balancing model for PFAL is established, and simultaneously an improved dual-population genetic algorithm is proposed. Firstly, through the characteristic analysis of PFAL, the tasks on PFAL are divided into three categories, namely the common, optional, and personality tasks. In addition, the correlation between the tasks is mainly considered. In the improved genetic algorithm, minimizing the number of stations, minimizing the load indexes between stations and within each station, and maximizing task-related degree are used as optimization objectives. In the initialization process, a method based on a TOP sorting algorithm is adopted for generating chromosomes. Furthermore, a new decoding algorithm is proposed to make up for the lack of the traditional decoding method, and individuals in the two populations are exchanged. Therefore, the search speed of the algorithm is accelerated, which shows good performance through classic tested problems. Finally, the effectiveness and feasibility of the method were validated by optimizing assembly line balancing of loaders.

Keywords Product family · Assembly line · Optimal balancing · Genetic algorithm

L. Hou · Y.-m. Wu (✉) · R.-s. Lai
Department of Mechanical and Electrical Engineering,
Xiamen University, Xiamen 361005, China
e-mail: wu20811055@163.com

C.-T. Tsai
Department of Ocean & Mechanical Engineering,
Florida Atlantic University, Boca Raton, FL 33431, USA

1 Introduction

Assembly lines, which are special flow-line production systems for the industrial assembly of high-volume standardized commodities, are utilized increasingly in today's manufacturing systems [1]. During the assembly process, workpieces traverse the assembly line by some kind of transportation systems, station by station, while in each station a fixed predetermined set of tasks is performed within a given cycle time.

Single-model assembly line (SMAL), designed to carry out a single homogenous product, is the most suitable choice for low variety of demand scenarios. Aiming at the balancing problem of SMAL, the main goal is to maximize the efficiency of the assembly line by minimizing the required capacity per unit of throughput (e.g. minimizing the number of stations in a given cycle time or by minimizing the cycle time in the required number of stations). Many researches on SMAL have been done to promote the development of the manufacturing industry. However, in today's market environment, faced with the challenge of cost-effectively supplying high variety within short product development times, SMAL is not able to respond the requirements of this new type of manufacturing strategies anymore. Specifically, changing market demands are leading more and more industries to diversify their product mix, with more models and optional features being offered.

As an effective means for mass customization (MC), product family (PF) strategy [2] is not the same with a single product development, aiming to maximize the overall performance by adjusting the balance between performance and commonality within a PF, in which products have some similar characteristics. In order to perform the models in a PF on the same assembly line, mixed-model assembly line (MMAL), which can produce several models of a standardized commodity simultaneously, is the best choice for manufacturers. The MMAL is a more complex situation in which several variants of a PF, referred to as models, are assembled simultaneously on the assembly line. In terms of the models, attributes, and prices of a PF, the structure of MMAL not only directly affect customers' purchasing decisions but also have a large impact upon the efficiency of product fulfillment. Therefore, the research on MMAL has henceforth attracted enormous attention in both marketing and product manufacturing [3].

In this paper, we address the mixed-model assembly line balancing problems (MALBPs) of PF. The balancing problem concerns how to assign assembly tasks and operators to candidate stations under the constraint of a given cycle time. The objectives are to minimize the number of stations, workload balancing at each station for different models in a PF, and design costs of the Product family assembly line (PFAL).

The rest of the paper is organized as follows. In the next section, we provide a detailed review of the related literature. In Section 3, the model description for PFAL is discussed in detail. Section 4 proposes an improved genetic algorithm for PFAL balancing, and then the effectiveness and feasibility of the method is proved by classic test problems. In Section 5, a case study of a loader assembly line balancing based on IGA is introduced. Finally, Section 6 concludes the paper with a summary and further study issues.

2 Literature review

Nowadays, MMALs are becoming popular in modern industry as an integral part of just-in-time (JIT) production systems [4, 5], especially for the assembly of products that are demanded in variety of models with comparatively low prices. However, the MALBPs are known to be NP-hard [1, 6, 7], in which there are the additional considerations of interactions between the assembled models. Therefore, finding an optimal solution is hardly possible, and the process of finding a feasible solution consists in the assignment of assembly tasks to each station with precedence constraints among these tasks. Considering the objective to be optimized, there are three well-known types for MALBPs [8, 9]:

Type I: Finding an assignment of tasks to stations such that the number of required stations is minimized in a pre-specified cycle time.

Type II: Aiming to minimize the cycle time for allocating tasks to a given number of stations.

Type III: Improving the assembly line efficiency by means of minimizing the cycle time and the number of stations simultaneously.

In the balancing process of MMAL, there are mainly two kinds of situations in traditional single-model assembly lines[10]:(1) design and balancing of a new assembly line and (2) redesign of an existing assembly line when changes in the assembly process or in the product range occurs. In terms of design and balancing of new assembly lines, Wang et al. [11] proposed a multi-objective optimization approach to balance the production process when a PF and the MMAL are designed. Also Zhu et al. [12] developed a model in which optimal sequences based on complexity induced by product variety in a mixed-model assembly line are defined. Yossi Bukchin et al. [13] developed an optimal solution procedure based on a backtracking branch-and-bound algorithm for MALBPs to minimize the total costs of the stations and the task duplication, and finally its performance was evaluated by a large set of experiments. Sener et al.[14] presented an effective method, addressing some particular features of the problem such as parallel stations and zoning constraints, to solve a mixed-model assembly line balancing problem of Type I. Zeng et al. [15] studied balance control of a complicated hybrid assembly line which appeared in the apparel sewing manufacturing system and solved the operator allocation problems. Ozcan et al. [16] presented a new mathematical model and a simulated annealing algorithm for the mixed-model two-sided assembly line balancing problem, and finally the experimental results shown that the proposed approach performs well. Considering simultaneously minimizing the cycle time (CT) and the number of stations for MMAL, Manavizadeh et al. [17] presented a multi-objective genetic algorithm to solve the balancing problem and the decision maker was provided with the subsequent answers to pick one based on the specific situation. Yagmahan [9] considered the mixed-model assembly line balancing problem to minimize the balance delay, the smoothness index between stations and the smoothness index within each station for a given cycle time. For redesign of an existing assembly line, what should be considered is productivity and adjustment cost. Yang et al. [18] proposed a multi-objective genetic algorithm to solve the rebalancing problem for a MMAL specifically with seasonal demands. Corominas et al. [19] considered the rebalancing problem by introducing assignment restrictions treated in the motorcycle-assembly process. Gamberini et al. [20] also solved rebalancing problem by using different heuristic algorithms and obtain satisfactory results. Grassi et al. [21] dealt with rebalancing problems in SMAL by retraining for new tasks, and two separate objective functions were used for concerning expected completion costs and the degree of similarity between initial and new task assignments.

Furthermore, in the balancing process (e.g. minimizing work load diversity between stations), there are different balancing objectives for MMAL. Therefore, the objectives of the previous studies could be categorized as: minimizing number of stations [22], minimizing cycle time [15, 23], and minimizing total costs. Besides, MALBPs are investigated with multiple objectives [24] which are optimized by heuristic procedures [25], linear programming [26], and dynamic programming [27].

In summary, the above researches are the basis of assembly line planning, which is also the key to assembly line optimal balance. What has received little attention in assembly line planning thus far, however, is the PFAL balancing, especially the lack of task correlation studies and the relationship analysis between the product demand and the assembly line balancing, so the optimization goal is only the load balancing for assembly line. In order to design the assembly line better for a product family, and support the implementation of innovation strategy for business managers, in this paper, on the basis of the product modular theory, we study the relationship between product modules and the assembly process, and analyze the correlation among the tasks on the PFAL. In the optimization process, we take the correlation between tasks into consideration mainly. Therefore, an improved genetic algorithm (IGA) with the objectives of minimizing the number of stations and maximize the workload smoothness between stations and within each station for different models, for solving the problems of traditional genetic algorithm (GA) in the assembly line optimization process, is proposed for better efficiency. In particular, dual population is adopted in IGA, and a new decoding algorithm is proposed for shortening evolutionary time. After a series of classic testing problems, IGA is used to optimize the assembly line for loaders.

3 Model descriptions for PFAL

A family of similar products can be assembled at the same time on PFAL, the balance of which is directly related to the facility utilization, production efficiency and product quality. PFAL based on modularity is the research object in this paper. Also, there is an intimate relationship between product modularity and the assembly line design for PFs. Furthermore, the analysis of relevant indicators between them is the key to PFAL optimal balance. Therefore, after the analysis of their relationships, we determine optimization objectives in this section, and the balancing optimization model is established for PFAL.

3.1 Modular product family and assembly line

Meeting the individual needs of customers, high efficiency and low cost are the main features of MC, and product family is an effective way to achieve MC. The product structure has similarity in PF, and the product process also has a lot of similarities. In order to improve production efficiency, MMAL is adopted to produce a family of products. However, PFAL, unlike ordinary MMAL, is a special kind of production line for a family of similar products, usually having dynamic changes and more task elements for different customer needs. A family based on modularity, adding, deleting, and replacing the personality module, is assembled on the same production line. The relationship between product family and the assembly line is described in Fig. 1. Theoretically, PFAL planning can be divided into the module, product, task and join layer from top to bottom. Composed of common, optional, and personality modules, products in the same product family have similarities and differences, which eventually led to common, optional, and personality tasks in PFAL. In optimization process for balancing, common tasks are the main body of PFAL planning, and simultaneously, optional and personality tasks as the supplement. The division of these tasks also provides a theoretical basis for the PFAL stability. In this paper, for load balancing, reducing the idle time and the highest efficiency in a specific constraint condition (e.g., the dynamic needs of customers), and how to assign assembly elements to different stations are the main objectives of PFAL balancing.

3.2 Model formulation

Considering a PF with J models, we assume that the demand for model J is D_j. In a PFAL, each model has its own precedence diagram, and a task may have different processing times because of the different assembled model. So the PFAL balancing problem is usually solved by converting them into so-called single-model line which can be described by a directed acyclic graph, namely $G=(V,P,t)$,where V,P,t represents the set of task elements, precedence relations between tasks, and processing times for different tasks, respectively in PFAL. Let the total number of tasks be equal to N, each task has an operation time t_i, and the precedence relations between tasks can be described by the matrix $P_{mn}=[x_{mn}]_{N\times N}$,where $x_{mn}=1$ means that task m has a higher priority than task n ,otherwise $x_{mn}=0$. The balancing goal is to find an optimal task allocation for the stations in PFAL planning, which can achieves the load balancing goals between stations and within each station, minimizing the idle time of stations in the cycle time,and realizing cost minimization and output maximization in the case of order constraints.

In this paper, we study the Type I balancing problem for PFAL, and the model notations were described as follows:

T_{total}: The total processing time
S: The total number of stations where $k=1,2,..,S$
N: The total number of tasks where $i=1,2,..,N$
J: The total number of models where $j=1,2,\ldots,J$
D_j: The demand for the model j
CT: The cycle time where $CT=\frac{T_{total}}{\sum_{j=1}^{J} D_j}$

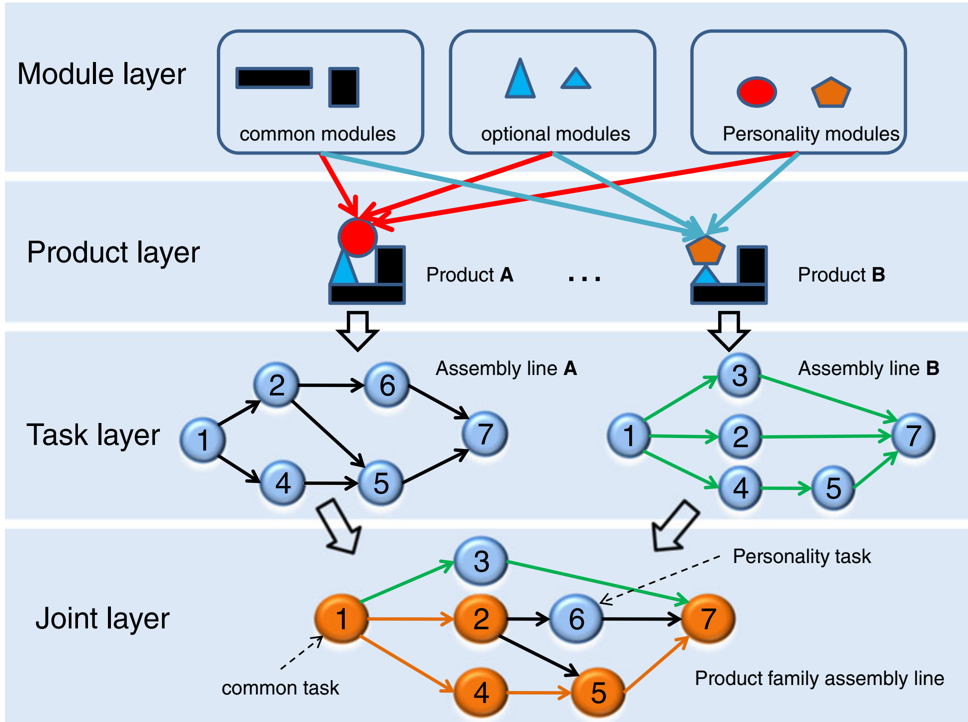


Fig. 1 Product family and assembly line

q_j: The proportion of mode j where $q_j = \frac{D_j}{\sum_{j=1}^{J} D_j}$

t_{ij}: The processing time of task i for model j

a_{mn}: The correlation between task m and n (when the task m,n are assigned to the same station, there is an important influence for PFAL planning, e.g. operation, transportation is more convenient), so $a_{mn}=a_{nm}$

$$Q_{ij} = \begin{cases} 0 & \text{if model } j \text{ has the task } i; \\ 1 & \text{otherwise.} \end{cases}$$

$$T_{ki} = \begin{cases} 0 & \text{if task } i \text{ assigned to station } k; \\ 1 & \text{otherwise.} \end{cases}$$

$$C_{kmn} = \begin{cases} 0 & \text{if task } m, n \text{ assigned to station } k \\ & \text{simultaneously;} \\ 1 & \text{otherwise.} \end{cases}$$

In the joint precedence diagram of MMAL, the process time for each task is calculated by the weighted average method [14, 18]. Constrained by customer needs, in order to obtain the actual required time for each task of PFAL, the process time (t_i) is calculated for task i as follows:

$$t_i = \sum_{j=1}^{J} \left(t_{ij} \cdot q_j \right) / \sum_{j=1}^{J} \left(Q_{ij} \cdot q_j \right) \tag{1}$$

In order to maintain a sustainable and stable process, the balancing planning for PFAL is a multi-objective optimization problem. So determining one or more indicators is needed to evaluate efficiency and planning of an assembly line. In our study, four objective functions for balancing are given as follows:

$$\mathrm{Obj}_1 = \sum_{k=1}^{S} k.T_{kN} \tag{2}$$

$$\mathrm{Obj}_2 = \left(\frac{1}{S-1} \sum_{k=1}^{S} \left(\sum_{j=1}^{J} D_j . \sum_{i=1}^{N} Q_{ij}.T_{ki} t_i - \overline{T}_s \right)^2 \right)^{\frac{1}{2}} \tag{3}$$

Where $\overline{T_s} = \sum_{k=1}^{S} \sum_{j=1}^{J} \sum_{i=1}^{N} q_j.T_{ki} t_{ij}$ represents the average time between stations

$$\mathrm{Obj}_3 = \frac{1}{S} \sum_{k=1}^{S} \left(\frac{1}{J-1} \cdot \sum_{j=1}^{J} \left(D_j \sum_{i=1}^{N} Q_{ij}.T_{ki} t_i - \overline{T}_k \right)^2 \right)^{\frac{1}{2}} \tag{4}$$

Where $\overline{T_s} = \sum_{j=1}^{J} \sum_{i=1}^{N} q_j.T_{ki} t_{ij}$ represents the average time within each station

$$\mathrm{Obj}_4 = \sum_{k=1}^{S} \sum_{m=1}^{N} \sum_{n=1}^{N} \frac{C_{kmn}.a_{mn}}{2} \tag{5}$$

In the above optimization objectives, the constraint between parameters is described in [14]. The optimization

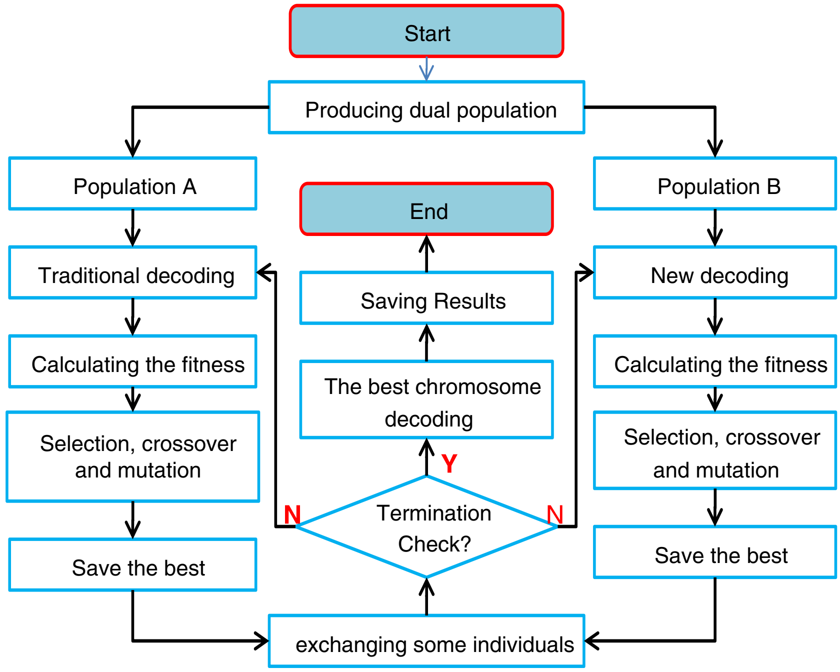


Fig. 2 Flow diagram of the IGA

objectives in Eqs. 2, 3, and 4 are minimizing the number of stations, and the load index between stations and within each station respectively. Equation 5 represents the task correlation of an assembly line. In order to simplify the calculation, we convert these objectives into a single objective as follows:

$$Z_{\min} = \alpha \mathrm{Obj}_1 + \beta \mathrm{Obj}_2 + \gamma \mathrm{Obj}_3 + \delta (1/\mathrm{Obj}_4) \tag{6}$$

For keeping dimensionless consistent for each sub-objectives, dimensionless coefficients $(\alpha, \beta, \gamma, \delta)$ are added in the optimization process for PFAL.

4 IGA for PFAL balancing

As an intelligent algorithm which simulates the process of biological evolution, GA is a powerful tool for solving complex problems. Also, needing only a few parameters, GA is easy to implement with powerful global search capability [28], which is most widely employed in complex nonlinear problems such as manufacturing systems [29, 30], engineering structure optimization and computer science [31]. In order to maintain the diversity of individuals and obtain the global optimal solution, all kinds of IGA, coming from the simple genetic algorithm, is exploited to solve practical problems [30–32]. Furthermore, the balance of PFAL affected by a variety of external factors is a typical NP-hard problem [1, 6, 7], which is very difficult to be solved using traditional mathematical methods, and the combination and operation between tasks is subjected to the constraints of various relationships. Additionally, the number of combinations between tasks grows sharply with the increase of the number of tasks, and GA is the best choice for solving this kind of

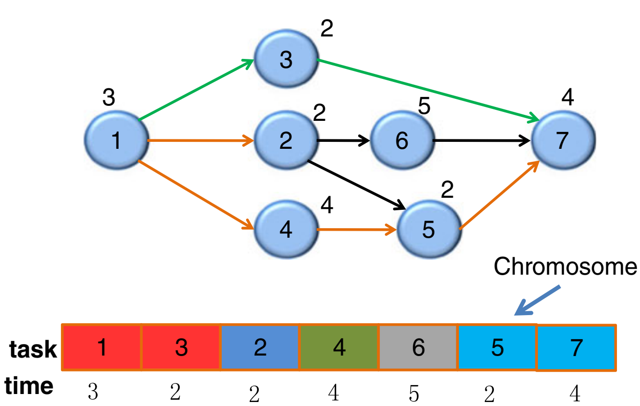


Fig. 3 The precedence diagram and encoding

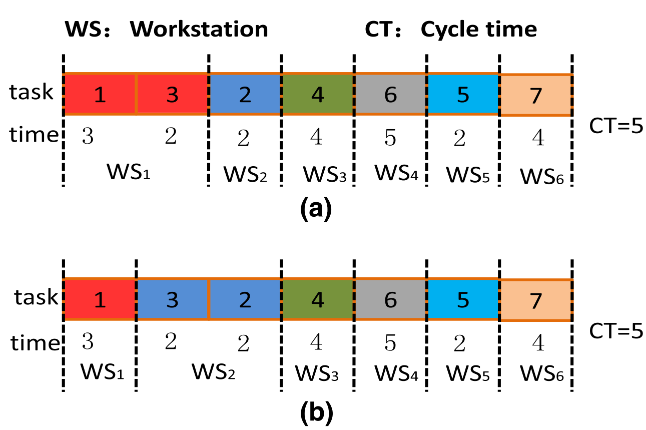


Fig. 4 Two different decoding methods

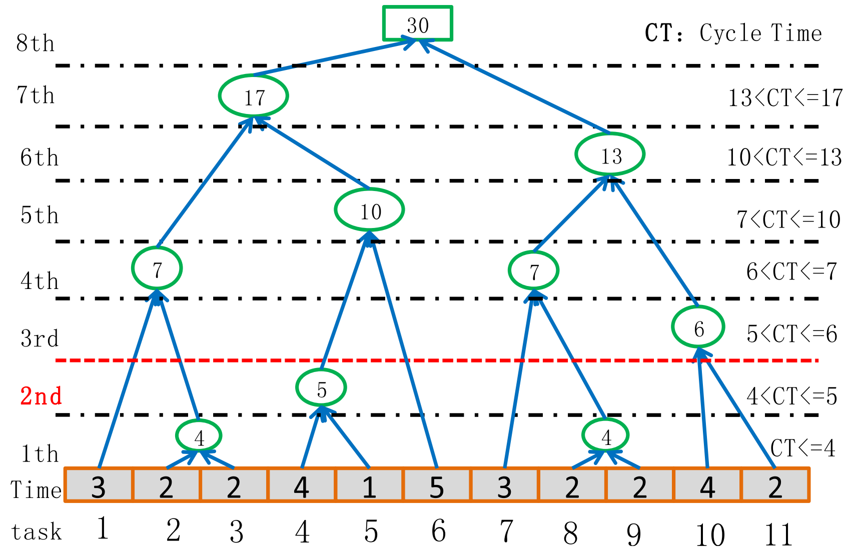


Fig. 5 The decoding method in IGA

problem. Therefore, the IGA is proposed for PFAL balancing problem, and the detailed process is shown in Fig. 2.

4.1 Encoding and decoding

Integer encoding method is adopted in IGA, in which each gene of a chromosome represents a task number, so the encoding length is equal to the number of tasks. However, task sequence is generated randomly in the traditional encoding method [14, 18], and the stations are divided according to the constraints and cycle time of an assembly line, for which the initial chromosome may not meet the task constraints and many tasks are divided into an independent station. Under the precedence constraint between tasks, TOP sort algorithm is used for producing reasonable chromosomes, which can reduce the search time to speed up the pace of optimization algorithm. The algorithm for initial chromosomes is described as follows:

Step 1: Initializing precedence relations graph G, let the task set $S=\{1,2,\ldots,N\}$

Step 2: Selecting the task i which in-degree value is 0 to the sorting set I

Step 3: Deleting task i from set S and the in-degree value is decreased by 1 for the direct successor of the task i

Step 4: Judging whether the number of elements in S is 0, if the value is false, returning Step 2, otherwise, the algorithm ends

According to a precedence diagram with seven tasks and their operating time, a chromosome is generated by the above algorithm and encoding results are described in Fig. 3.

In the Type I balancing process for PFAL, CT can be calculated according to customer needs, and the traditional decoding method calculates the fitness value for a chromosome from left to right [10, 14, 18, 33], namely as long as the sum time of a station and the next task does not exceed the CT, the next task is included in the station. Taking the chromosome shown in Fig. 3 as example, when the value of CT is 5, the tasks are divided into six workstations as shown in Fig. 4a while the other result, as shown in Fig. 4b, is obtained by a new decoding method. Since the time variance between stations shown in Fig. 4b is smaller than that in Fig. 4a, we can see that the station planning in Fig. 4a has a

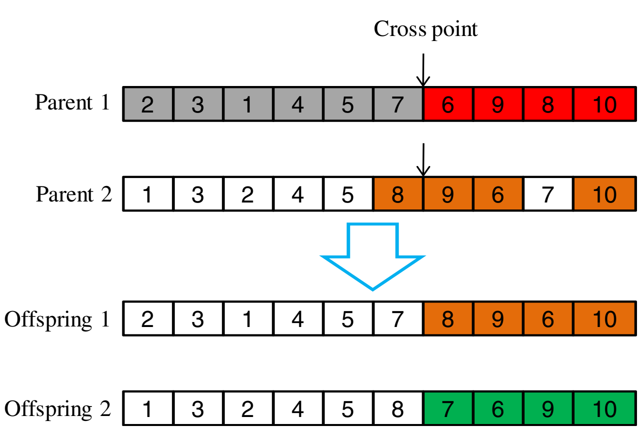


Fig. 6 The single-point crossover for chromosomes

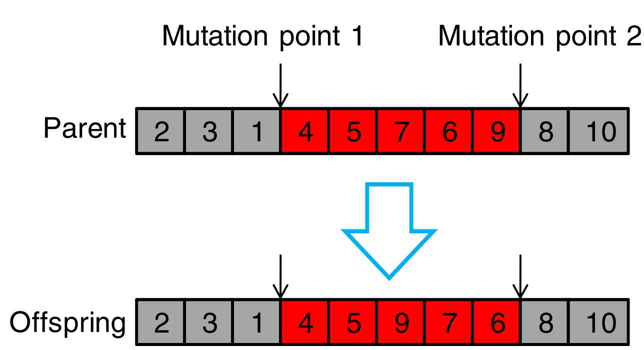


Fig. 7 The mutation process for chromosome

Table 1 The parameters of test problems and optimization results

Problems	Tasks (N)	Models (J)	Proportion	Noorul		IGA	
				Evolution times	Obj1	Evolution times	Obj1
1 Bowman	8	2	0.5:0.5	33	4	35	4
2 Thomopoulos	19	3	0.55:0.27:0.18	45	3	44	3
3 Mitchell	21	2	0.2:0.8	47	6	40	6
4 Heskia	28	2	0.7:0.3	52	7	45	7
5 Sawyer	30	2	0.75:0.25	55	10	47	9
6 Lutz	32	3	0.5:0.2:0.3	62	9	50	8
7 Gunther	35	2	0.8:0.2	70	9	55	9
8 Kim	61	4	0.4:02:0.3:0.1	75	19	60	18
9 Tonge	70	2	0.6:0.4	80	16	65	16
10 Arcus	111	5	0.25:0.2:0.2:0.15:0.2	92	18	67	16

lack of considering the balance between stations. Therefore, in order to increase the search speed and improve the efficiency of GA in PFAL planning for balancing, a new decoding method is proposed for considering load balancing between stations from the beginning. Let CT be equal to 5, and the encoded chromosome shown in Fig. 5 needs only 2 times to complete station planning: $\{1\},\{2,3\},\{4,5\},\{6\},\{7\},\{8,9\}$, $\{10\}$and$\{11\}$. The algorithm is described in detail as follows:

Step 1: Obtaining the task time sequence $Seq=\{T_1,T_2,\ldots T_i,\ldots,T_N\}$ from a chromosome

Step 2: Selecting two adjacent tasks T_i,T_j, the sum time of which is the smallest. If T_{sum}<CT, go to Step3, otherwise, the algorithm ends

Step 3: writing T_{sum} before T_i in set Seq, and then deleting T_i,T_j from Seq

Step 4: checking whether the number of elements in Seq is 1, if the value is false, returning Step2, otherwise, the algorithm ends

To compensate for lack of the traditional method, dual populations are generated in the initialization process, and the two decoding methods, namely the traditional method and the new method proposed in this paper, are used for the two populations respectively. For the expansion of solution space and improving search speed, we exchange chromosomes between two populations in IGA.

4.2 Population initialization and fitness evaluation

According to the precedence diagram of tasks in a PF and the initialization method described in 4.1, we generate two initial

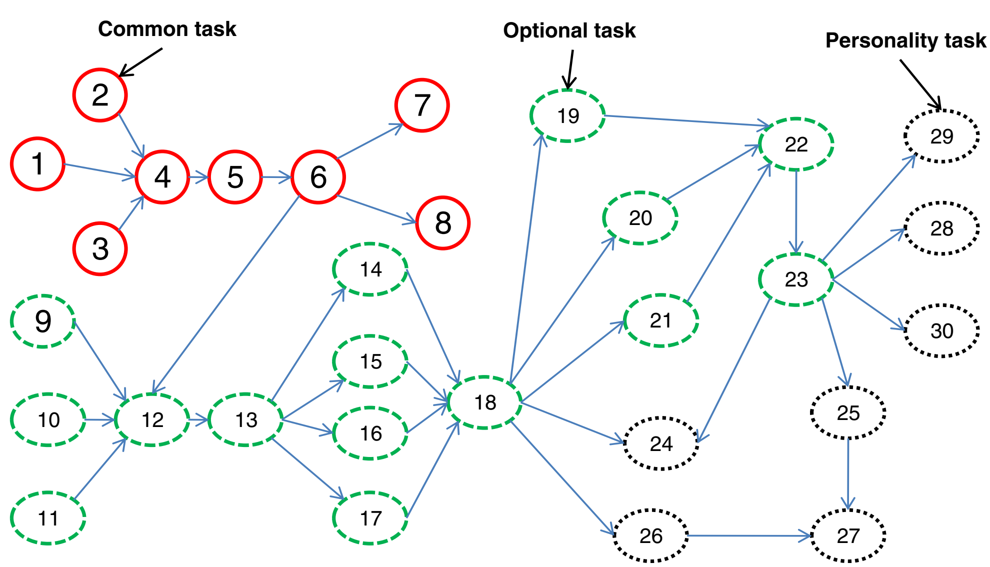


Fig. 8 The precedence graph for SWLs

Table 2 Tasks and time for the family of SWLs

Task	Time			t_i	Number of worker	T_{fin}
	t_{916}	t_{918}	t_{920T}			
1	200	195	240	202.9	5	40.6
2	245.2	256.2	262.1	253.9	6	42.3
3	150.6	171.7	140.6	161.2	4	40.3
4	347.2	400.8	439.3	391.0	10	39.1
5	410	620.5	560.2	551.7	14	39.4
6	103.8	83.9	132.4	96.5	2	48.3
7	93.5	115.3	113.7	108.8	3	36.3
8	25	43	33	36.4	1	36.4
9	28	0	0	28	1	28
10	0	30	0	30	1	30
11	0	0	35	35	1	35
12	14	14	14	14	1	14
13	13	13	13	13	1	13
14	12	12	0	12	1	12
15	15	16	13	15.3	1	15.3
16	0	0	18	18	1	18
17	0	0	15	15	1	15
18	5	5	5	5	1	5
19	8	8	8	8	1	8
20	7	7	7	7	1	7
21	7	7	7	7	1	7
22	7	7	7	7	1	7
23	11	13	8	11	1	11
24	23	23	0	23	1	23
25	0	0	17	17	1	17
26	0	0	4	4	1	4
27	0	0	8	8	1	8
28	24	0	0	24	1	24
29	0	20	0	20	1	20
30	0	0	31	31	1	31

Table 3 The correlation of partial tasks

Task A	Task B	Correlation
1	2	0.8
1	3	0.5
1	4	0.1
1	5	0.1
2	3	0.7
2	4	0.7
3	4	0.8
4	5	0.8
4	6	0.4
5	6	0.5
6	7	0.1
6	8	0.1
7	8	0.5
9	12	0.8
10	12	0.8
11	12	0.8
14	15	0.7
15	18	0.6
16	17	0.7
17	18	0.6
19	20	0.5
19	21	0.5
19	22	0.5
19	23	0.5
20	21	0.5
21	22	0.5
21	23	0.5
22	23	0.7
25	26	0.7
25	27	0.7
26	27	0.4
28	29	0.4
28	30	0.4
29	30	0.4

populations, namely population A and B. For chromosome decoding, the traditional method and the new method proposed in this paper are used for the two populations, respectively.

The fitness function is used to determine that the chromosomes are good or bad in GA, which is also the only standard and the key to algorithm implementation simultaneously. In this study, Eq. 6 is used as the optimization objective for PFAL planning for balancing.

4.3 Selection

Individuals with high fitness value are selected from the population, which is the basis of crossover and mutation for chromosomes, also the key to optimization. In IGA, roulette wheel selection strategy [3] is used to select individuals with high fitness values. Additionally, the best individual of each generation is always retained to the next generation.

4.4 Crossover

Crossover plays a key role in the evolution process, and the excellent traits of parental individuals can be preserved in the offspring as much as possible. In the crossover process, single-point crossover is adopted in IGA, and two individuals selected randomly from the parental generation exchange genetic information after the cross-point, new chromosomes followed. However, new chromosomes may not be reasonable by the direct exchange for PFAL constraints. To solve this

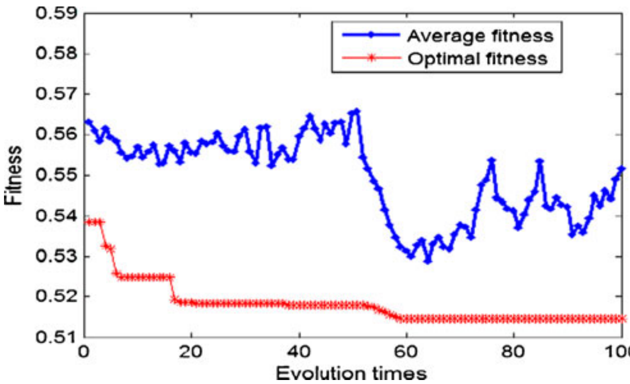


Fig. 9 Fitness tracking for optimization process

problem, taking the exchange process shown in Fig. 6 as example, the exchanged sequence of the parent 1 is 6, 9, 8, and 10, and then looking for the ordering of this sequence in parent 2, namely 8, 9, 6, and 10, which is replaced to the original sequence in parent 1. Finally, a new chromosome is generated, which is offspring 1. Similarly, parent 2 is converted to offspring 2.

4.5 Mutation

In mutation operation for IGA, firstly, two mutation points are produced randomly, between which the gene fragment is restructured, namely according to the precedence graph of a PF, a new reasonable fragment is generated as the method of initializing chromosomes. The detailed process is shown in Fig. 7.

4.6 Generating new populations

In order to increase the diversity of two populations and improve search performance of IGA, several chromosomes, excepting the best individuals, are exchanged between the two populations, and the number of exchanged chromosomes is generated randomly. Finally, new populations are generated after individual exchange.

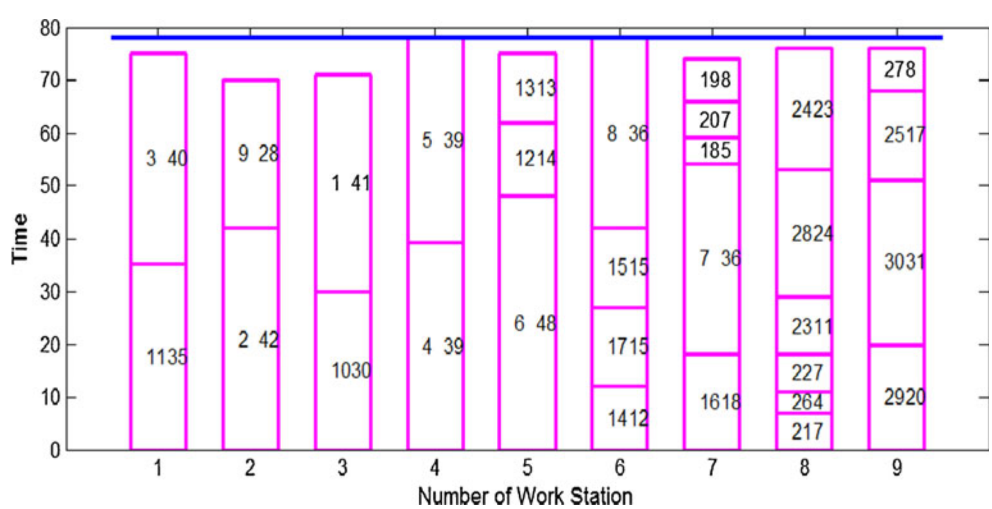
4.7 Performance evaluation for the IGA

Ten classic problems [14, 18] of MMAL for balancing test are selected for evaluating the IGA proposed in this paper. In the case of the same fitness function, compare with the algorithm proposed by Noorul [34], the IGA obtains better optimization values, specifically a fewer number of evolution times, namely IGA has a higher efficiency in the optimization process, which is exhibited in Table 1.

5 Case study

Wheel loader is an engineering mechanical device with many advantages such as high efficiency, strong maneuverability, reliable performance, and convenient operation. However, having some uncertain factors, the assembly process of wheel loaders is extremely complex, and the close coordination between workstations is necessary for assembly operations. For assembling variations for Small Wheel Loaders (SWLs) on the same assembly line, the mode of PFAL is exploited by most manufacturers. Therefore, in order to verify the effectiveness and feasibility of IGA proposed in this paper, the assembly line for SWLs, namely a family of SWLs, is optimized by IGA in this section. Additionally, IGA is compared with other optimization methods and shows its excellent efficiency and performance in the optimization process.

Fig. 10 Task allocation for stations by IGA



5.1 PFAL balancing

The characteristics and parameters of SWL assembly line are described as follows:

1. There are three models included in the product family of SWLs, which are 916, 918, and 920 T, and they are produced on the same assembly line. The working time in a day is 560 time units. In additional, the demands for 916, 918, and 920 T are 2, 4 and 1. Therefore, the CT= 560/(2+4+1)=80.
2. Thirty tasks are included in SWL assembly line, and the precedence graph, where each task has its own ID and the common, optional, and personality tasks are numbered 1–8, 9–23, and 24–30, respectively, is also displayed clearly for the family of SWLs as shown in Fig. 8.
3. In the planning process for PFML, if the actual operating time of some task is greater than the CT, more workers are assigned for the task. So the actual operating time: t_{916}, t_{918} and t_{920T}, the number of assigned workers and the time T_{fin} used in the optimization process for each task in SWLs are shown in Table 2. According to Eq. 1, the time t_i can be calculated, and then t_i divided by the number of workers is T_{fin}.
4. Considering the correlation between tasks for PFAL, the entire assembly line may be more balanced for convenience and operability, the correlation degree of PFAL is calculated by Eq. 5. According to the company survey and experience knowledge of experts in this field, the correlation of partial tasks in the family of SWLs is given in Table 3, and the other task correlation is 0.

According to the method of population initialization described in section 4, population A and B are generated. Except for taking different decoding methods, the two populations have the same optimized operations, e.g., Eq. 6 is used as the objective function, population size, termination times, crossover, and mutation probability are 60, 100, 0.8, and 0.2, respectively. In the optimization process, calculated jointly for two populations, the best and average fitness is tracked in Fig. 9. In addition, Fig. 10 describes the optimization results for the assembly line of the SWLs family and the final planning is shown in Table 4.

Table 4 IGA optimization results for SWLs

Stations	Task portfolio	Optimization objectives
1	1,10	
2	2,9	
3	3,11	
4	4,5	1. Obj_1=9
5	6,12,13	2. Obj_2=18
6	7,16,18,19,20	3. Obj_3=4.5
7	8,14,15,17	4. Obj_4=4.8
8	21,22,23,24,26,28	
9	25,27,29,30	

Table 5 Balancing algorithm parameters

Balancing algorithm	Population size	Maximal generation	P_c	P_m	Evolution times	Solution
IGA	60	100	0.6	0.15	59	Table 4
GA	100	100	0.7	0.2	90	Table 6
hGA	100	100	0.5	0.15	78	Table 6
MoGA	100	100	0.9	0.1	70	Table 6

Table 6 Other optimization results for SWLs

Stations	Task portfolio	Optimization objectives
GA		
1	1,11	
2	2,9	
3	3,10	
4	4,5	1. Obj_1=9
5	6,12,13	2. Obj_2=25
6	7,14,15,17	3. Obj_3=6.3
7	8,16,18,20,21,26	4. Obj_4=4.1
8	19,22,23,29,30	
9	24,25,27,28	
hGA		
1	3,11	
2	1,10	
3	2,9	
4	4,5	1. Obj1=9
5	6,12,13	2. Obj_2=23
6	8,14,15,17	3. Obj_3=5.8
7	7,16,18,20,21	4. Obj_4=4.8
8	19,22,23,26,28,29	
9	24,25,27,30	
MoGA		
1	3,11	
2	1,10	
3	2,9	
4	4,5	1. Obj_1=9
5	6,12,13	2. Obj_2=19
6	8,14,15,17	3. Obj_3=4.9
7	7,16,18,20,21,26	4. Obj_4=4.3
8	19,22,23,24,28	
9	25,27,29,30	

5.2 Verifying the validity of IGA

In order to further verify the IGA effectiveness, the pure GA, hGA [14], and MoGA (NSGA-II) are selected as the comparison experiments, whose parameters are given in Table 5 and the optimization results are shown in Table 6. As can be seen from Tables 4 and 6, IGA proposed in this paper, in the case of smaller populations, has less termination times which is 50, namely having higher efficiency. In optimization results, although the number of stations is nine for all the algorithms, the planning of SWLs assembly line, optimized by IGA, has a smaller load index for between stations and within station and a greater task correlation. Therefore, not only does IGA have higher efficiency but the optimization result is also the best.

6 Conclusions

Characteristics of PFAL are analyzed in this paper. From the point of view of PF modularity, PFAL tasks are divided into common, optional, and personality tasks, by which PFAL can be easily rebalanced for adjustment costs. Characteristics and correlation between tasks are mainly analyzed in PFAL planning process.

For characteristics of PFAL, the TOP sort algorithm is used for population initialization meeting assembly constraints. The IGA with two populations is proposed, in which a new decoding method can make up for the deficiencies of the traditional decoding. And, also the performance of IGA is tested for 20 classic balancing problems of assembly lines, and finally the better results, with higher efficiency and search speed, are obtained in the optimization process for balancing.

IGA, as used in the assembly line balancing for SWLs, is compared with the other three methods and the results show that the smallest load indexes and the highest correlation degrees for tasks, between stations and within each station, are obtained in the optimization process. Therefore, the IGA proposed in this paper has achieved good results in PFAL, which, furthermore, can provide some theoretical support for business managers in the planning process for assembly line.

The PFAL balancing problem is, however, still an NP-hard problem. It is difficult to fully consider all of the production factors in the optimization model of PFAL for balancing. More importantly, product family changes dynamically as customer demands, how to establish an objective model more precisely to achieve the actual optimization goals and the PFAL rebalancing, namely dynamic balance planning for PFAL, in which further research is needed.

Acknowledgments This work was supported by National Natural Science Foundation of China (Grant No. 71172055) and The Special Program for Innovation method of the Ministry of Science and Technology, China (Grant No. 2011IM020400)

References

1. Rashid MFF, Hutabarat W, Tiwari A (2012) A review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches. Int J Adv Manuf Technol 59:335–349
2. Battini D, Faccio M, Ferrari E, Persona A, Sgarbossa F (2007) Design configuration for a mixed-model assembly system in case of low product demand. Int J Adv Manuf Technol 34:188–200
3. Zhu X, Jack Hu S, Koren Y, Huang N (2012) A complexity model for sequence planning in mixed-model assembly lines. J Manuf Syst 31: 121–130
4. Sandanayake YG, Oduoza CF (2009) Dynamic simulation for performance optimization in just-in-time-enabled manufacturing processes. Int J Adv Manuf Technol 42:372–380
5. Hadi M, Zandieh M (2013) An imperialist competitive algorithm for a mixed-model assembly line sequencing problem. J Manuf Syst 32(1):46–54
6. Yolmeh A, Kianfar F (2012) An efficient hybrid genetic algorithm to solve assembly line balancing problem with sequence-dependent setup times. Comput Ind Eng 62:936–945
7. Guo ZX, Wong WK, Leung SYS, Fan JT, Chan SF (2008) A genetic-algorithm-based optimization model for scheduling flexible assembly lines. Int J Adv Manuf Technol 36:156–168
8. Liu SB, Ng KM, Ong HL (2008) Branch-and-bound algorithms for simple assembly line balancing problem. Branch-and-bound algorithms for simple assembly line balancing problem. Int J Adv Manuf Technol 36:169–177
9. Yagmahan B (2011) Mixed-model assembly line balancing using a multi-objective ant colony optimization approach. Expert Syst Appl 38:12453–12461
10. Sener A, Mirac Bayhan G, Adil B (2013) Hybridizing ant colony optimization via genetic algorithm for mixed-model assembly line balancing problem with sequence dependent setup times between tasks. Appl Soft Comput 13:574–589
11. Bard JF, Shtub A, Joshi SB (1994) Sequencing mixed-model assembly lines to level parts usage and minimize line length. Int J Prod Res 32:2431–2454
12. Zhu X, Hu SJ, Koren Y, Huang N (2012) A complexity model for sequence planning in mixed-model assembly lines. J Manuf Syst 31: 121–130
13. Bukchin Y, Rabinowitch I (2006) A branch-and-bound based solution approach for the mixed-model assembly line-balancing problem for minimizing stations and task duplication costs. Eur J Oper Res 174:492–508
14. Sener A, Mirac Bayhan G (2011) A hybrid genetic algorithm for mixed model assembly line balancing problem with parallel workstations and zoning constraints. Eng Appl Artif Intell 24:449–457
15. Xianhui Z, Wai-Keung W, Sunney Yung-Sun L (2012) An operator allocation optimization model for balancing control of the hybrid assembly lines using Pareto utility discrete differential evolution algorithm. Comput Oper Res 39:1145–1159
16. Ozcan U, Toklu B (2009) Balancing of mixed-model two-sided assembly lines. Comput Ind Eng 57:217–227
17. Manavizadeh N, Rabbani M, Moshtaghi D, Jolai F (2012) Mixed-model assembly line balancing in the make-to-order and stochastic environment using multi-objective evolutionary algorithms. Expert Syst Appl 39:12026–12031
18. Caijun Y, Jie G, Linyan S (2012) A multi-objective genetic algorithm for mixed-model assembly line rebalancing. Comput Ind Eng. doi:10.1016/j.cie.2011.11.033
19. Corominas A, Pastor R, Plans J (2008) Balancing assembly line with skilled and unskilled workers. Omega 36(6):1126–1132
20. Gamberini R, Gamberini E, Grassi A, Regattieri A (2009) A multiple singlepass heuristic algorithm solving the stochastic assembly line rebalancing problem. Int J Prod Res 47(8):2141–2164

21. Gamberini R, Grassi A, Rimini R (2006) A new multi-objective heuristic algorithm for solving the stochastic assembly line rebalancing problem. Int J Prod Econ 102(2):226–243
22. Andres C, Miralles C, Pastor R (2008) Balancing and scheduling tasks in assembly lines with sequence-dependent setup times. Eur J Oper Res 187:1212–1223
23. Levitin G, Rubinovitz J, Shnits B (2006) A genetic algorithm for robotic assembly line balancing. Eur J Oper Res 168:811–825
24. Kara Y, Paksoy T, Chang CT (2009) Binary fuzzy goal programming approach to single model straight and U-shaped assembly line balancing. Eur J Oper Res 195:335–347
25. Dimitriadis SG (2006) Assembly line balancing and group working: a heuristic procedure for workers' groups operating on the same product and workstation. Comput Oper Res 33(9):2757–2774
26. Peeters M, Degraeve Z (2006) An linear programming based lower bound for the simple assembly line balancing problem. Eur J Oper Res 168(3):716–731
27. Bautista J, Pereira J (2009) A dynamic programming based heuristic for the assembly line balancing problem. Eur J Oper Res 194(3):787–794
28. Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning [J]. Mach Learn 3(2):95–99
29. Moslemipour G, Lee TS, Rilling D (2012) A review of intelligent approaches for designing dynamic and robust layouts in flexible manufacturing systems [J]. Int J Adv Manuf Technol 60:11–27
30. Sun Y, Zhang C, Gao L, Wang X (2011) Multi-objective optimization algorithms for flow shop scheduling problem: a review and prospects. Int J Adv Manuf Technol 55:723–739
31. Chandrasekaran M, Muralidhar M, Krishna CM, Dixit US (2010) Application of soft computing techniques in machining performance prediction and optimization: a literature review. Int J Adv Manuf Technol 46:445–464
32. Wang L, Keshavarzmanesh S, Feng HY, Buchal RO (2009) Assembly process planning and its future in collaborative manufacturing: a review. Int J Adv Manuf Technol 41:132–144
33. Chica M, Cordon O, Damas S (2011) An advanced multiobjective genetic algorithm design for the time and space assembly line balancing problem. Comput Ind Eng 61:103–117
34. Noorul HA, Jayaprakash J, Rengarajan K (2006) A hybrid genetic algorithm approach to mixed-model assembly line balancing. Int J Adv Manuf Technol 28:337–34