International Journal of Industrial Engineering Computations 14 (2023) 645–658

Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Mixed-model assembly line balancing problem in multi-demand scenarios

Kang Wang[a], Qianqian Han[a*] and Zhenping Li[b*]

[a]*School of Management Engineering, Capital University of Economics and Business, Beijing, Republic of China*
[b]*School of Information Beijing Wuzi University, China*

C H R O N I C L E

Article history:
Received June 4 2023
Received in Revised Format July 10 2023
Accepted September 5 2023
Available online September 5 2023

Keywords:
Multi-demand scenarios
Mixed-model assembly line
Mixed-integer programming
Parallel task
Phased algorithm

A B S T R A C T

The mixed-model assembly line balancing problem (MMALBP) in multi-demand scenarios is investigated, which addresses demand fluctuations for each product in each scenario. The objective is to minimize the sum of costs associated with tasks allocation, workstation activation, and penalty costs for unbalanced workloads. A mixed integer programming model is developed to consider the constraint of workstation space capacity. A phased heuristic algorithm is designed to solve the problem. The computational results show that considering demand fluctuations in multiple demand scenarios leads to more balanced workstation loads and improved assembly line production efficiency. Finally, sensitivity analysis of important parameters is conducted to summarize the impact of parameter changes on the results and provide practical management insights.

1. Introduction

The assembly line is a critical production line of manufacturing enterprises, and the layout of the assembly line determines the processing sequence of raw materials, the value-added space of products, and the actual economic benefits generated by the scheduling of various equipment and personnel (Daneshamooz et al., 2022). Therefore, whether the layout of the assembly line is reasonable directly affects the production efficiency and production cost of the enterprise. Manufacturers usually design and layout assembly lines according to actual production needs, to complete the tasks of transforming raw materials and semi-finished products into final delivered products (Zhenping et al., 2023).

Early research on assembly line problems (ALP) typically assumed a fixed production cycle and uniform production ratios across shifts without any changes. However, demand often fluctuates due to market and policy influences, and production tasks vary across shifts, resulting in significant deviations from expected performance and workstation loads. Existing research on demand fluctuations mainly focuses on two aspects: one is to re-optimize existing assembly line systems based on production conditions (rebalancing), and the other is to consider demand fluctuations in the initial design of assembly line layouts and develop a layout scheme that can respond to changes in demand (initial balancing).

Generally, initial balancing methods are more effective than rebalancing methods in terms of balancing the cost and workload of assembly lines when dealing with demand fluctuations. In existing literature that applies initial balancing methods, most introduce flexible methods such as buffer zones, parallel assembly lines, and parallel workstations. A few studies have introduced parallel tasks, which allow the same tasks to be assigned to different workstations for processing, as a flexible method to deal with demand fluctuations. An assembly line with parallel tasks can flexibly handle fluctuating demands while balancing the workload of each workstation. However, the premise of introducing parallel tasks is to configure the corresponding production environment, such as equipment and materials, for the corresponding workstation.

* Corresponding author
E-mail: han577@cueb.edu.cn (Q. Han) lizhenping@bwu.edu.cn (Z. Li)
ISSN 1923-2934 (Online) - ISSN 1923-2926 (Print)
2023 Growing Science Ltd.
doi: 10.5267/j.ijiec.2023.9.002

This paper focuses on the assembly line balancing problem in the face of demand fluctuations. By introducing parallel tasks and enabling different product processing plans in different demand scenarios, the assembly line can maintain a good balance and workload distribution. For the sake of clarity, the problem proposed in this paper is defined as the mixed-model assembly line balancing problem in multi-demand scenarios.

The rest of this paper is organized as follows: Section 2 provides a literature review of relevant studies. Section 3 describes our problem, introduces assumptions and relevant parameters, and presents the mathematical model. Then, in Section 4, a solution method is developed, and in Section 5, the computational results based on the instance of the assembly line problem set are discussed, along with corresponding sensitivity analysis, providing some practical management insights. Finally, Section 6 presents the conclusion and future directions for further research.

2. Literature review

This review is divided into two parts, beginning with a listing and analysis of the two existing balancing types for dealing with demand fluctuations, followed by an evaluation of the main measures, and influencing factors of existing flexible methods.

2.1 Two existing assembly line balancing types

At present, there are two main methods to deal with the assembly line balance problem of demand fluctuations: initial balance and rebalancing.

The rebalancing method mainly considers two aspects: the product launch sequence and the worker processing. Sikora (2021) considers the balancing and sequencing problem of assembly lines under different demand scenarios, where the task allocation remains unchanged but the product launch sequence changes with the demand scenario. Li et al. (2023) address the problem of workload imbalance and product accumulation caused by demand fluctuations, with the objective of minimizing the number of workstations, maximizing workload balance, and reducing product waiting time. They solve the assembly line design and sequencing problem of different scales through hierarchical algorithms and mixed heuristic algorithms. Simaria et al. (2009) consider the U-shaped assembly line production under multiple demand scenarios, where each scenario produces only one product, and workers are allowed to move between multiple workstations to cope with demand fluctuations. Li and Gao (2014) model the assembly line balancing problem by setting the probability of occurrence of different demand scenarios, and allow workers to process the assembly line products in high-load demand scenarios that exceed the timeout. Yang and Gao (2016) train workers to have cross-functional abilities for adjacent workstations to solve the problem of workstation overload under different demand scenarios.

The initial balancing method mainly starts from the actual layout of the assembly line, such as increasing buffer zones, adding parallel assembly lines, parallel workstations, and setting parallel tasks. Zhang (2020) established a finite buffer queue network model. When demand fluctuates, blockages occur on the assembly line. Setting buffer zones can temporarily store unfinished workpieces, ensuring that the production of the assembly line is not affected. Jiao et al. (2022) established a model that combines parallel assembly lines and U-shaped layouts to cope with demand fluctuations and designed a bidirectional priority heuristic algorithm for solving the problem. Danes et al. (2022) considered the problem of parallel assembly of multiple products with high demand, allowing products to be processed at parallel workstations. They established a mixed integer programming model and solved it using an adaptive neighborhood search algorithm. Mosadegh et al. (2022) allowed the allocation of the same task of different products to different workstations. When demand fluctuates, parallel tasks are added to the bottleneck workstation to alleviate its processing pressure. They established a mixed integer programming model with the objectives of minimizing workstation opening costs and tasks duplication rates and designed a branch-and-bound algorithm for solving the problem.

Relatively speaking, when demand fluctuations cause an imbalance in the assembly line, the cost of initial balancing by considering the characteristics of each demand fluctuation scenario is slightly lower than the cost of rebalancing. Thiago et al. (2021) considered different demand scenarios, allowing different processing schemes (allowing equipment movement) for each task in different demand scenarios, with the goal of minimizing the number of workstations and establishing a mixed integer programming model. They also designed a heuristic algorithm for solving the problem and demonstrated the economic feasibility of assembly line layout plans considering different demand scenarios through example results. However, Thiago et al. did not provide a proof of the feasibility of equipment movement (e.g., whether the workstation capacity is sufficient) after relaxing the assumption that equipment cannot move, nor did they consider the time and economic costs of equipment movement.

2.2 The main measures and influencing factors

In addition, assembly lines that only consider assembly line balance often experience unbalanced workstation loads and low production efficiency when facing demand fluctuations. To balance the workload of each workstation on the assembly line, flexible methods such as setting up parallel assembly lines (Aguilar et al., 2020; Özcan, 2019), parallel workstations (Lopes et al., 2019, 2021; Leiber et al., 2022) and parallel tasks (Mosadegh et al., 2012; Anuar & Bukchin, 2006; Guo et al., 2008))can effectively improve bottleneck links and balance workstation loads. Among them, the parallel tasks strategy requires less space and is more universal and easier to operate.

Compared to the parallel layout of multiple assembly lines and multiple work units of parallel workstations, adding parallel tasks requires significantly less space and is more versatile. For a single task, improvements can be made by purchasing equipment needed for task duplication, which is not only cost-effective but also allows for flexible adjustments according to the actual needs of the enterprise.

In the research related to parallel tasks, Mosadegh et al. (2012) considered allowing the same task of a product in a mixed-model assembly line to be processed at different workstations in an MPS, maximizing the utilization of workstation time, reducing the total task time, and balancing the workload of workstations. Anuar et al. (2006) allowed the same task to be processed at multiple workstations, adjusting the assembly tasks between stations dynamically during the operation of the assembly line, reducing the cycle time while achieving greater production efficiency. Guo et al. (2008) introduced parallel tasks with the objective of minimizing the total idle time, first determining the shared tasks and task allocation ratios among different workstations, and then determining the task allocation rules. The above studies balanced the workload between workstations by allowing parallel tasks, but did not consider the equipment configuration cost generated by allowing parallel tasks, nor did they explore the relationship between tasks and workstations.

The introduction of parallel tasks will inevitably increase the cost of equipment configuration, including the cost of equipment purchase and the cost of setting up on the workstation, and the costs generated by the same task configuration on different workstations (such as processing time and raw material transportation time) may vary. Cakir et al. (2011) considered the equipment purchase cost required for task allocation to workstations, with the goal of improving workstation load smoothness and reducing assembly line total cost, and studied the single-product assembly line balancing problem. Niroomand (2021) considered the cost of task allocation and the cost of workstation activation. Salehi et al. (2020) aimed to minimize the sum of equipment configuration costs, workstation activation fees, and worker wages, and designed a hybrid simulated annealing algorithm to solve the assembly line balancing problem. Pearce et al. (2019) considered workstation characteristics, restricted task allocation, and established a multi-constraint integer programming model for complex real-world assembly line balancing problems. Hazır et al. (2013) established a mathematical model for assembly line balancing design problems considering assembly tool constraints and solved it using a solver. These scholars considered task allocation from the perspectives of allocation cost and allocation restrictions, but there are few studies that combine the task-workstation matching relationship, workstation space constraints, and task allocation costs (the equipment configuration cost).

Therefore, this paper investigates the problems of assembly line balancing and load balancing, taking into account different demand scenarios. Parallel tasks are introduced, allowing the allocation of equipment corresponding to the same task to different workstations. Additionally, the equipment configuration cost is considered, as well as space limitations on each workstation. The objective is to minimize the equipment configuration cost, workstation activation cost, and penalty costs for unbalanced load on the assembly line.

The main contributions of this paper are summarized as follows:

Firstly, the MMALBP in multi-demand scenarios is studied, considering parallel tasks, spatial constraints of workstations, and the correspondence between tasks and workstations.

Secondly, a mixed integer programming model is established to decide the tasks allocation plan and product processing strategy. The objective of the model is to minimize the total cost, including workstation activation costs, equipment configuration costs, and penalty costs for uneven workstation loads.

Finally, a phased heuristic method is developed to solve the problem, and the effectiveness of the algorithm is verified through numerical instances and sensitivity analysis providing management insights.

3. Problem definition and mathematical formulation

3.1 Problem Description and Analysis

3.1.1 Problem Description

This paper combines the actual production scenarios in the workshop and considers how to allocate equipment and plan the production workstations for different products in each demand scenario based on the different demand scenarios in multiple production cycles, i.e., the different numbers of products put into production. The goal is to reduce the layout cost of the production line and improve production efficiency, while smoothing the workload of each workstation.

The problem can be described as follows: given a cycle time, the types of products and the demand for each product within each production cycle, the tasks *i* and the priority relationship between the task are predetermined. The set of successor tasks *S(i),* the number of available workstations |*J*|, and the maximum number of tasks that can be assigned to each workstation *N* are also known. Each task requires different equipment, and if a task is assigned to a workstation, the equipment required for that task must be installed at the corresponding workstation. It is assumed that the same task of each product can only be completed at one workstation, while the same task of different products can be completed at different workstations, allowing parallel tasks to be set up at different workstations. The objective is to assign tasks to workstations and determine which workstation will complete each task of each product during the production cycle, to balance the workload of each workstation as much as possible under different demand scenarios and minimize the layout cost of the assembly line.

3.1.2 Problem Analysis

The mixed-model assembly line balancing problem with multiple demand scenarios has a two-stage characteristic. In the first stage, equipment allocation is performed, and in the second stage, the processing plan for each product's various tasks is determined based on the equipment allocation and the production requirements of each demand scenario. Therefore, this problem can be represented as a mixed-integer programming model. The research on the mixed-model assembly line balancing problem with multiple demand scenarios is based on several assumptions:

(1) There is no difference in available workstations, and each task can be assigned to any workstation;
(2) The task time of each task is related to the product, not the assigned workstation;
(3) Each task of the same product must be completed at one workstation within each production cycle, while the same task of different products can be completed by different workstations;
(4) The demand for each product in each scenario is known and remains constant; the task allocation plan for different products in different scenarios may differ.

3.2 Mathematical Model of Mixed-Model Assembly Line Balancing Problem in Multi-Demand Scenarios

3.2.1 Notation

(1) Sets
S : Set of product types
I : Set of tasks
J : Set of workstations
D : Set of demand scenarios
$S(i)$: Set of immediate successor task types for task *i*

(2) Index
s : Index of product category, $s \in S$
i : Index of task index, $i \in I$
j : Index of workstation, $j \in J$
d : Index of requirement scenario, $d \in D$

(3) Parameters:
C : Cycle time
F : Fixed cost of workstation activation
M : Any large real number
N : Maximum number of tasks that can be allocated to a workstation
a_{ij} : The equipment configuration cost required for production of task *i* on workstation *j*
t_{is} : Task time required for task *i* of product *s*
β : Penalty coefficient for unbalanced maximum workload difference
q_{ds} : The proportion of the demand for product *s* in scenario *d* to the total demand for product *s*
Q_d : The total demand for products required in scenario *d*

(4) Decision variables

$$x_{ij} = \begin{cases} 1, & \text{The equipment required for task } i \text{ is allocated to workstation } j \\ 0, & \text{Otherwise} \end{cases}$$

$$w_j = \begin{cases} 1, & \text{Activate Workstation } j \\ 0, & \text{Otherwise} \end{cases}$$

$$y_{dijs} = \begin{cases} 1, & \text{The task } i \text{ of product } s \text{ in scene } d \text{ is assigned to workstation } j. \\ 0, & \text{Otherwise} \end{cases}$$

$T_{\mathrm{d\_min}}$: The minimum workload of the workstation in scenario d

$T_{d\_\max}$: The maximum workload of the workstation in scenario d

3.2.2 Mathematical Model

In this paper, we establish a mixed integer programming model for load balancing in a mixed-model assembly line with multiple demand scenarios of parallel tasks.

$$\min Z = \sum_{i=1}^{I}\sum_{j=1}^{J} a_{ij} x_{ij} + F * \sum_{j=1}^{J} w_j + \beta \sum_{d=1}^{D} Q_d (T_{d\_\max} - T_{d\_\min}) \tag{1}$$

subject to

$$\sum_{j=1}^{J} x_{ij} \geq 1, \ \forall i \tag{2}$$

$$\sum_{i=1}^{I} x_{ij} \leq N, \ \forall j \tag{3}$$

$$y_{dijs} \leq x_{ij}, \ \forall i, j, s, d \tag{4}$$

$$\sum_{j=1}^{J} j y_{dijs} \leq \sum_{j=1}^{J} j y_{dajs}, \ \forall s, d, a \in S(i) \tag{5}$$

$$\sum_{j=1}^{J} y_{dijs} = 1, \ \forall s, i, d \tag{6}$$

$$x_{ij} \leq w_j, \ \forall i, j \tag{7}$$

$$T_{d\_\max} \leq C, \ \forall d \tag{8}$$

$$\sum_{s=1}^{S} q_{ds} \sum_{i=1}^{I} t_{is} y_{dijs} \leq T_{d\_\max}, \ \forall j \tag{9}$$

$$\sum_{s=1}^{S} q_{ds} \sum_{i=1}^{I} t_{is} y_{dijs} \geq T_{d\_\min} - M * (1 - w_j), \ \forall j, d \tag{10}$$

$$T_{d\_\max} \geq 0, T_{d\_\min} \geq 0 \tag{11}$$

$$x_{ij}, y_{dijs}, w_j \in \{0,1\}, \quad \forall i, j, s, d \tag{12}$$

The objective function (1) represents the minimization of total cost, where the first term represents the equipment configuration cost, the second term represents the cost of workstation activation, and the third term represents the sum of penalty costs for unbalanced workload among workstations in different scenarios. The constraints (2)-(12) are as follows: (2) each task must be assigned to at least one workstation; (3) the maximum number of tasks that can be assigned to each workstation cannot exceed N; (4) a workstation can only process a corresponding task if the required equipment is allocated to that workstation; (5) the processing sequence of tasks must be satisfied; (6) a task for the same product can only be processed at the same workstation; (7) if a task is assigned to a workstation, the workstation must be activated; (8) the maximum processing time for any workstation cannot exceed the cycle time; (9)-(10) the average workload of any activated workstation must be within the range of the maximum and minimum values; and (11)-(12) the decision variable values are subject to constraints.

Because the load balancing problem in traditional single-demand scenario mixed-model assembly lines is NP-hard, the load balancing problem in multi-demand scenario mixed-model assembly lines is an extension of the single-demand scenario problem, making the problem even more complex and difficult to solve. In this paper, we propose to use a phased heuristic approach based on the model structure to solve the problem.

2. 3.2.3 Example

This section utilizes an example to illustrate the superiority of considering different demand scenarios compared to not considering demand scenarios in the case of demand fluctuations. Specifically, we consider an assembly line that produces two products and three possible demand scenarios. The product precedence diagram and the processing times for each operation (indicated by the number in the upper right corner of each operation) are shown in Fig 1. The cycle time $C = 7$ and the demand quantities for products A and B under each of the three demand scenarios are presented in Table 1.

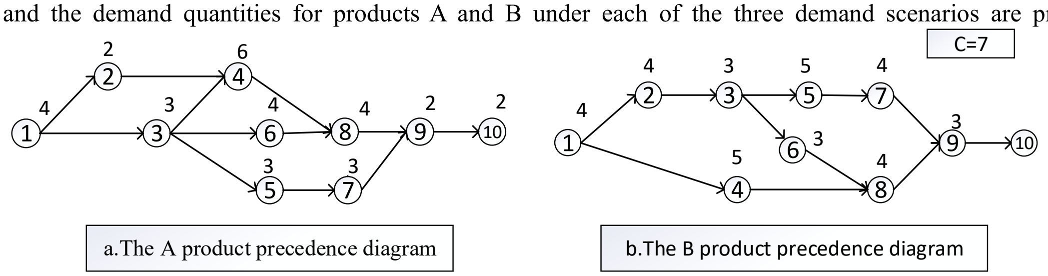

Fig. 1. Product precedence diagram

Table 1
Demand quantities for products under each demand scenario

Product	Scenario 1	Scenario 2	Scenario 3	Average
A	9	6	3	6
B	7	5	9	7

Table 2 presents the optimal task allocation schemes, the number of active workstations, and the average maximum workload difference for four different scenarios. These scenarios include considering demand scenarios and parallel tasks, considering parallel tasks but not demand scenarios (where the demand for each scenario is averaged and treated as a single demand scenario for task allocation), considering demand scenarios only, and not considering either. These results were obtained by using the Gurobi solver to directly solve the model based on the given data.

Table 2 shows that considering demand scenarios and parallel tasks results in the smallest average maximum workload difference, indicating that the workload is most balanced among workstations in this condition. On the other hand, the allocation scheme that does not consider either demand scenarios or parallel tasks has the largest maximum workload difference, indicating poor balance among workstations. When considering a single factor, the effectiveness of considering parallel tasks is better, but it requires additional equipment configuration costs. While considering demand scenarios alone do not require additional equipment costs but requires additional workstations to balance the workload. When both factors are considered, the number of workstations is not increased, and the maximum workload difference is significantly improved, resulting in a balanced workload among workstations. However, this also increases the equipment configuration cost.

Table 2
Allocation results considering different factors

Factors	Sequence number of activated workstations								Average maximum workload difference
	1	2	3	4	5	6	7	8	
considering demand scenarios and parallel tasks	1,2	2,3	4	6,8	5,8	5,7	9,10		1.32
considering only parallel tasks	1,3	4,6	2,3	5,7	4,7	6,8	9,10		1.38
considering only demand scenarios	1	2,3	6	4	5	8	7	9,10	2.71
not considering either	1	2,3	5	4	6，7	8	9,10		3

4. A Phased algorithm

The mathematical model for balancing mixed-model assembly lines with multiple demand scenarios involves many integer variables and complex coupling relationships between variables, making direct solving of the model time-consuming. In this section, we propose a phased heuristic algorithm based on the model structure. In the first stage, we obtain the initial equipment allocation plan for processing tasks based on the product precedence diagram. In the second stage, we decide on the product task allocation plan for each scenario based on the initial equipment allocation plan. In the third stage, we determine the final equipment allocation plan for processing operations based on the product task allocation plan and calculate the objective function value.

In the first stage, we establish a traditional task balancing and allocation model without parallel tasks based on the product precedence diagram. By solving the model, we obtain the initial equipment allocation plan. We solve the task balancing and allocation model $|D|+|S|$ times in total, sequentially solving for $|D|$ demand scenarios and $|S|$ individual products. We then summarize the results of each sub-problem. The specific model is as follows:

$$\min Z=\sum_{i=1}^{I}\sum_{j=1}^{J}a_{ij}x_{ij}+F*\sum_{j=1}^{J}w_j+\frac{\sum_{d=1}^{D}Q_d}{D}(T_{d\_\max}-T_{d\_\min}) \tag{13}$$

subject to

$$\sum_{j=1}^{J}x_{ij}=1,\forall i \tag{14}$$

$$\sum_{i=1}^{I}x_{ij}\le N,\forall j \tag{15}$$

$$\sum_{j=1}^{J}jx_{dj}\le\sum_{j=1}^{J}jx_{aj},\forall d,a\in S(i) \tag{16}$$

$$x_{ij}\le w_j,\forall i,j \tag{17}$$

$$T_{d\_\max}\le C,\forall j,s \tag{18}$$

$$\sum_{s=1}^{S}q_{ds}\sum_{i=1}^{I}t_{is}x_{ij}\le T_{d\_\max},\forall j \tag{20}$$

$$\sum_{s=1}^{S}q_{ds}\sum_{i=1}^{I}t_{is}x_{ij}\ge T_{d\_\min}-M*(1-w_j),\forall j \tag{21}$$

$$T_{d\_\max}\ge 0,T_{d\_\min}\ge 0 \tag{22}$$

$$x_{ij},w_j\in\{0,1\},\ \forall i,j \tag{23}$$

The objective function (13) represents the minimization of total cost, where the first term represents the equipment configuration cost, the second term represents the cost of workstation activation, and the third term represents the sum of penalty costs for unbalanced workload between workstations under average demand during the cycle. Constraint (14) indicates that each task is assigned to a workstation. Constraint (15) specifies that the maximum number of tasks that can be assigned to each workstation cannot exceed N . Constraint (16) ensures that the task processing sequence is satisfied. Constraint (17) states that a workstation can only process a task if the required equipment for that task is assigned to that workstation. Constraint (18) limits the maximum processing time for any workstation to the cycle time. Constraints (19)-(20) ensure that the average workload for any activated workstation falls within the range of the maximum and minimum values. Constraints (21)-(22) specify the value constraints for the decision variables.

In the second stage, we use the results obtained from solving the sub-problems in the first stage to obtain the initial equipment allocation plan for processing operations, i.e., the values of x_{ij} . We then make product task allocation decisions for each cycle with the goal of workload balancing, resulting in the product task allocation plan for each scenario. The specific model is as follows:

$$\min Z=\beta\sum_{d=1}^{D}Q_d(T_{d\_\max}-T_{d\_\min}) \tag{23}$$

subject to

Constraints (2)-(11)

$$y_{dijs},\in\{0,1\},\ \forall i,j,s,d \tag{24}$$

The objective function (23) represents the sum of penalty costs for unbalanced workload between workstations in each demand scenario. The model includes constraints (2)-(11) as described in the previous text. Constraint (24) specifies the value constraints for the decision variables.

In the third stage, we use the product task allocation plan obtained in the second stage as input parameters for the model (1)-(12) and solve the model again to obtain the final equipment allocation plan for processing operations.

The algorithm steps are as follows:

Step 1: Solve for the equipment and product allocation plan without parallel tasks for each demand scenario separately. Combine the equipment allocation results from multiple scenarios to obtain the initial equipment allocation plan. Proceed to Step 2.

Step 2: Solve the sub-problems with parallel tasks allowed for each demand scenario based on the initial equipment allocation plan obtained in Step 1, resulting in the product task allocation plan for the second stage. Proceed to Step 3.

Step 3: Use the product task allocation plan obtained in Step 2 as known parameters and solve the original model again to obtain a new equipment allocation plan and calculate the objective function value.

Repeat Steps 2-3 until the objective function value of the new solution no longer decreases or the specified termination time is reached. The algorithm flow chart is shown in Fig 2:

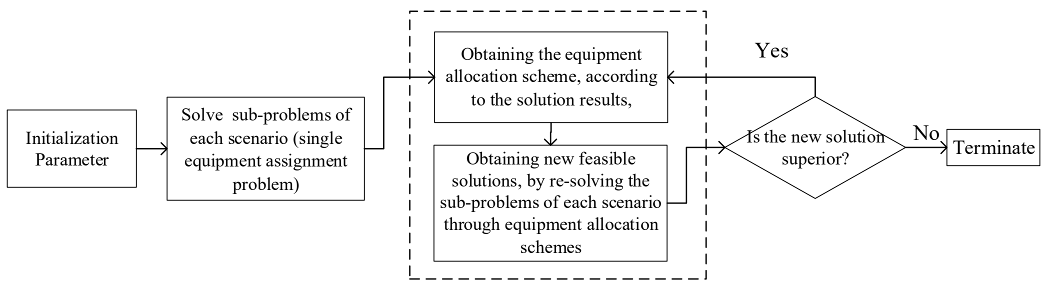

Fig. 2. Algorithm flow chart

5. Numerical experiments

In this section, we generate simulation examples based on the basic data in reference (Otto et al., 2013). We solve the examples using both the commercial solver Gurobi and the phased heuristic algorithm, and then compare and analyze the results of the two methods. Finally, we conduct sensitivity analysis on important parameters to derive some management recommendations.

5.1 Parameter Setting

Based on the data information from literature (Otto et al., 2013), a tasks precedence diagram is generated. The parameters of task number $|I|$, product category number $|S|$, and scene number $|D|$ are set, with task number $|I|$ being 10, 15, and 20, product category number $|S|$ being 2 and 3, and scene number $|D|$ being 2, 3, and 4. The example is generated by using the full combination method for all parameters, resulting in a total of 18 sets of examples. The task time t_i is randomly generated from a uniform discrete distribution $U[0,7]$, and the configuration cost a_{ij} of the task on the workstation is randomly generated from a uniform discrete distribution $U[3,12]$.

The maximum number of tasks that can be scheduled for each workstation is set to $N=3$, with a cycle time of $C=7$. The fixed cost of activating a workstation is $F=20$, and a penalty coefficient is applied for unbalanced workload is $\beta=1$. The demand ratio of demand scenarios in each cycle is randomly generated for different scale instances, with a total sum of 1.

5.2 Analysis of experiment results

For each instance, the Gurobi 9.0.2 solver and a phased heuristic algorithm were used for calculations. The programming was done using Python 3.8 and the calculations were run on a Linux server with an Intel(R) Xeon(R) Gold -6248R CPU clocked at 3.00 GHz and 32 GB RAM. Due to the exponential increase in solver run time with increasing instance size, a maximum run time of CPU3600s was set for the solver in this study.

5.2.1 The comparison of solution time and accuracy

Table 3 shows the results obtained by directly using the Gurobi solver to solve the integer programming model and using the phased algorithm designed in this paper. The table records the solution time, average and maximum objective function values, and the GAP value compared to the solver solution for 18 sets of instances. As the size of the instance increases, the solution time of both methods increases. For instances with the same number of tasks and product types, the solution time also increases with the number of cycles. However, the direct solution time of the solver is much larger than that of the phased heuristic algorithm for all instances. For some instances with 15 tasks, 4 scenarios, and 2 products, the solver cannot obtain the optimal solution within 3600s ('*' in the third column indicates the number of instances in which the solver cannot obtain the optimal solution within 3600s, and '-' indicates that no instance can provide the optimal solution within the given time). However, the maximum solution time of the phased heuristic algorithm for all instances does not exceed 500s, and the GAP value compared to the solver solution is also within 5.7%.

5.2.2 The analysis of the change in objective function values

As the size of the instance increases, the objective function value also increases, but the average GAP value fluctuates between 3% and 6%. The best GAP values obtained during the calculations for each instance are also listed, and in some instances, the optimal solution can be obtained ('*' in the last column indicates the number of case studies in which the optimal solution was obtained).

To investigate the variation in objective function values for different numbers of product types, the mean values of each objective function for each instance are listed in Table 4. The number of workstations, the equipment configuration cost, and penalty cost for unbalanced workload are compared for instances with the same number of tasks and scenarios but different numbers of product types. The results for instances with 2 product types are shown on the left, while those with 3 product types are shown on the right. Fig 3 provides a more intuitive comparison of the optimization objectives for different product types in Table 4. Fig 3(a) shows that, except for the instance with 15 tasks and 2 scenarios, the number of workstations required for 3 product types is lower than that for 2 product types in all other instances.

Table 3
The comparison of solution results

	The Gurobi solver results				The phased algorithm results					
	CPU time(s)		objective value		CPU time(s)		objective value		GAP	
$\lvert I\rvert-\lvert S\rvert-\lvert T\rvert$	Average value	Maximum value	Average value	Maximum value	Average value	Maximum value	Average value	Maximum value		Best GAP
10-2-2	8.69	13.58	225.44	237.80	0.62	1.11	232.17	243.00	3.00	0.00^{3*}
10-2-3	96.65	315.01	238.22	258.20	4.22	2.99	248.14	275.80	4.21	0.00^{2}
10-2-4	455.16	934.57	252.48	264.60	5.33	4.53	262.78	270.20	4.05	0.00^{1}
10-3-2	4.25	11.61	185.17	212.10	1.10	2.87	191.97	253.00	3.40	0.00^{6}
10-3-3	12.57	43.68	199.49	227.20	2.06	4.94	206.28	241.20	3.28	0.00^{3}
10-3-4	292.68	1122.23	212.25	248.20	27.89	124.16	220.03	256.20	3.69	0.00^{2}
15-2-2	43.26	348.50	264.19	304.60	1.44	3.16	275.31	312.30	4.24	0.00^{4}
15-2-3	508.88	2438.40	287.47	336.90	3.31	13.52	302.79	381.90	5.12	0.00^{3}
15-2-4	2238.91	3600^{9*}	311.05	351.50	26.13	88.98	328.60	400.30	5.57	0.00^{1}
15-3-2	352.51	1015.56	256.13	276.30	23.57	59.94	264.39	286.60	3.23	0.00^{4}
15-3-3	1855.52	3600^{5}	274.83	306.00	49.47	114.39	286.46	325.90	4.29	0.00
15-3-4	—	—	294.57	331.90	147.88	289.77	303.79	341.80	3.11	0.00
20-2-2	1203.76	3600^{3}	353.90	390.10	11.98	18.18	366.83	415.00	3.66	0.38
20-2-3	1999.76	3600^{9}	364.67	398.00	32.71	111.41	383.13	429.10	5.03	0.00^{2}
20-2-4	—	—	414.84	479.00	208.05	422.27	427.10	487.50	3.01	0.00
20-3-2	1653.21	3600^{4}	349.19	391.50	32.97	111.67	368.76	422.20	5.65	1.12
20-3-3	—	—	373.88	402.00	159.32	414.45	388.39	402.00	3.88	0.00^{2}
20-3-4	—	—	408.40	463.30	2369.16	899.14	420.94	478.20	3.12	0.03

Table 4
Comparison of different values

$\lvert I\rvert-\lvert D\rvert$	Objective value		The number of workstations		Penalty cost		Equipment configuration cost	
	$\lvert S\rvert$=2	$\lvert S\rvert$=3	$\lvert S\rvert$=2	$\lvert S\rvert$=3	$\lvert S\rvert$=2	$\lvert S\rvert$=3	$\lvert S\rvert$=2	$\lvert S\rvert$=3
10-2	238.25	210.15	5.33	4.93	58.92	28.61	72.67	82.87
10-3	244.35	229.64	5.13	4.93	68.61	47.11	73.07	83.87
10-4	293.40	262.15	5.40	5.07	99.93	65.55	85.47	95.27
15-2	305.99	297.77	7.33	7.53	60.05	42.84	99.27	104.27
15-3	353.69	324.41	8.00	7.80	79.36	48.81	114.33	119.60
15-4	390.92	352.72	7.87	7.53	112.59	76.59	121.00	125.47
20-2	437.89	402.87	10.60	10.53	90.89	46.60	135.00	145.60
20-3	454.79	437.67	10.13	9.87	113.72	84.53	138.40	155.80
20-4	510.83	475.81	10.33	10.13	156.29	114.81	147.87	158.33

In Fig 3(b), the cost of an unbalanced workload for each instance with 2 product types is greater than that for instances with

3 product types. In Fig 3(c), it is observed that the equipment configuration cost increases with the number of product types, but the flexibility of the assembly line is also increased, which reduces the penalty cost for the unbalanced workload. Through comparison, it is found that when the number of tasks remains constant, the number of workstations opened does not change significantly with an increase in the number of cycles, but the cost of unbalanced workload and the equipment configuration cost gradually increase. This is because each scenario incurs penalty costs, and the equipment configuration cost corresponding to reducing the penalty cost also gradually increases.

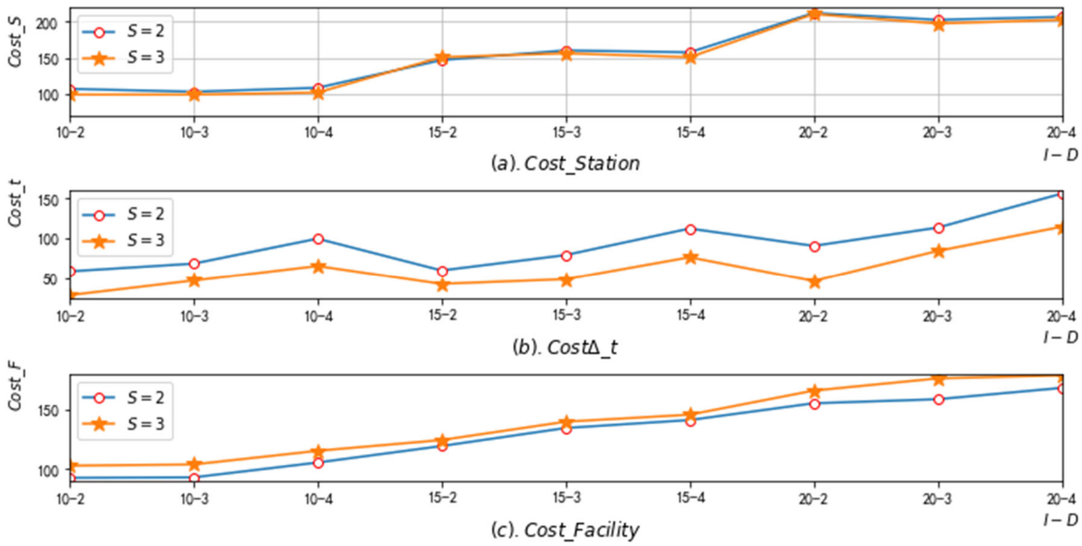

Fig. 3. Cost comparison

5.3 Sensitivity analysis

The calculation results vary with different parameter settings in the same instance, such as equipment allocation results and total cost. This section mainly conducts sensitivity analysis on four parameters: cycle time C, maximum number of devices that can be placed on a workstation N, workstation activation cost F, and penalty coefficient for uneven load cost β. Taking the medium-sized case 15-3-3 (task-scenario-product) as an instance, the impact of different parameter settings on total cost, fixed cost of workstation activation, extreme value of uneven load, and the equipment configuration cost is analyzed.

5.3.1 Analysis of the Impact of Cycle Time Variations

To analyze the impact of cycle time on total cost, workstation activation, load variation, and equipment allocation results, cycle time was sequentially taken from 7 to 12. To avoid the limitation of maximum equipment number of the workstation when the cycle time is large, it was set to 5. Other parameters were fixed, and the calculated results are shown in Fig 4. In Fig 4(a), the bar chart is used to describe the penalty costs for unbalanced workload under different parameters, and the line chart is used to describe the penalty cost generated by load variation. In Fig 4(b), the line chart is used to describe the trend of the objective function value *OBJ* under different parameters, and the changes in the equipment configuration cost *Cost_F* and workstation activation cost *Cost_S* are also shown.

Fig 4(a) shows the penalty costs for unbalanced workload and total load imbalance in different scenarios at different cycle times. It can be observed from the load variation in each scenario that the load variation decreases first and then increases with the increase of cycle time. After the cycle time reaches 12, the load variation remains unchanged. The penalty costs for unbalanced workload follow the same trend, indicating that when the cycle time is sufficiently large, the total cost and solution are no longer affected by the cycle time.

In Fig 4(b), it can be observed that the cost of workstation opening decreases initially, then increases and finally remains constant with an increase in the cycle time. The equipment configuration cost increases initially, then decreases and finally remains constant. The total cost increases initially, then decreases, and finally increases slightly until the cycle time to 12. This indicates that with an increase in the cycle time, the number of workstations decreases, while the amount of equipment that can be placed on each workstation increases. This can reduce the unbalanced load phenomenon and lower the total cost. However, when the cycle time is sufficiently large, the total cost will no longer change with an increase in the cycle time due to the inherent limitations of the problem.

Thus, the cycle time has a direct impact on the total cost of the mixed-model assembly line balancing problem in multi-demand scenarios. It has a significant effect on the number of workstations and equipment configuration cost, while its impact on load balancing is relatively minor.

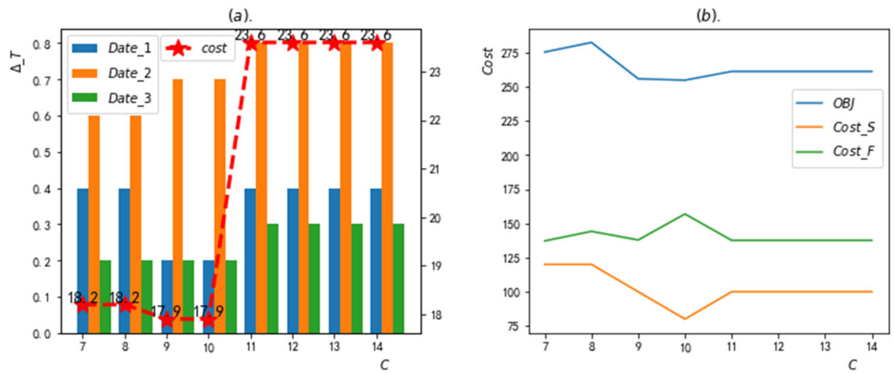

Fig. 4. The Impact of Cycle Time Variation on Cost

5.3.2 Analysis of the impact of changes in the number of equipment that can be installed on workstations

To analyze the effect of the maximum number of equipment that can be accommodated by a workstation on the solution results, the maximum equipment number $|N|$ was sequentially increased from 1 to 7 while keeping the other parameters constant. The solution results are shown in Fig 5. Fig 5(a) uses a bar chart to describe the variation trend of load range under different parameter scenarios and uses a line chart to describe the change in penalty cost caused by load range. Fig 5(b) uses a line chart to represent the objective function value *OBJ,* the equipment configuration cost *Cost_F*, and workstation activation cost *Cost_S* for each scenario corresponding to different parameters.

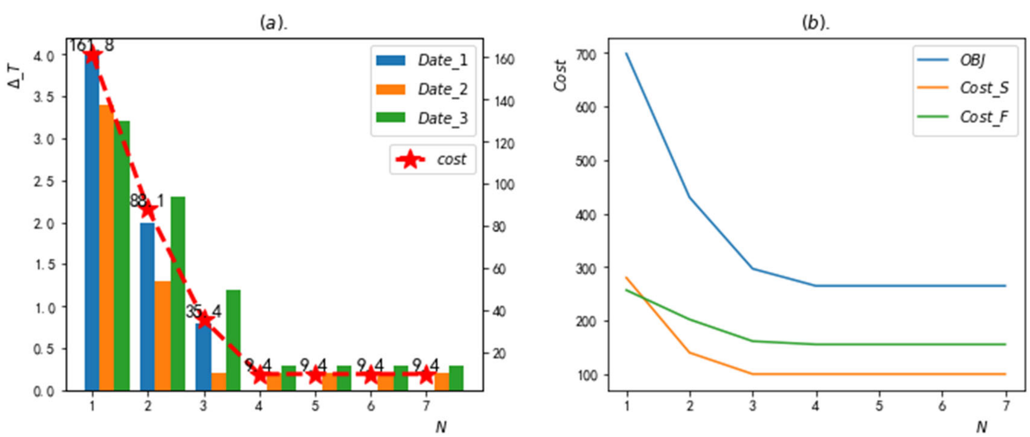

Fig. 5. The Impact of *N* Variation on the Results

From Fig 5(a), it can be observed that as the value of increases, the load imbalance within each scenario gradually decreases. When reaches a value of 4, it remains constant and the load imbalance within Scenario 1 becomes 0. Additionally, the penalty costs for unbalanced workload in each case also gradually decreases and then remains constant. This indicates that the number of devices has a significant impact on load balancing, and when the number of equipment is sufficiently large, the load on the workstation becomes highly balanced.

From Fig 5(b), it can be observed that as *N* increases, the total cost gradually decreases. Both the workstation activation cost and equipment configuration cost decrease initially and then remain constant. The increase in the amount of equipment that can be placed on workstations leads to a reduction in the number of activated workstations, thereby lowering the workstation activation cost. The decrease in total equipment configuration cost is due to the reduction in the amount of equipment that can be placed on workstations, resulting in a decrease in the adjustable space for equipment on the workstation. Moreover, the cost of placing the same equipment on different workstations varies. Therefore, when the amount of equipment that can be placed is smaller, the cost of equipment tends to be relatively higher.

Increasing the number of configurable equipment on workstations is a fundamental method to improve the extreme load imbalance in various scenarios. Moreover, it does not have a significant impact on the total cost when the equipment configuration cost is reasonable. However, due to practical production environment limitations, it is usually not feasible to place too much equipment. Therefore, actual decision-making in enterprises needs to be tailored to the specific circumstances to determine the maximum number of equipment that can be placed.

5.3.3 Analysis of the impact of Fixed Cost Changes in activation workstation on results

To analyze the impact of fixed costs of activation workstations on the solution results, the fixed costs were gradually increased from 0 to 24 while keeping the other parameters constant and solving the problem instance. The solution results are shown in Fig 6. Fig 6(a) uses a bar chart to describe the trend of the average number of workstations activated under different parameter instances and uses a line chart to describe the change in penalty costs generated by load range. Fig 6(b) uses a line chart to represent the objective function value *OBJ*, the equipment configuration cost *Cost_F*, and activation workstation cost *Cost_S* under different parameters and instances.

From Fig 6(a), it can be observed that the number of opened workstations decreases with the increase of fixed cost and then remains constant, while the penalty costs for the unbalanced workload first increases and then decreases while remaining constant. This indicates that the cost of activation workstations has a greater impact on the number of workstations but has a smaller impact on the penalty costs for unbalanced workload. From Fig 6(b), it can be seen that, except for equipment configuration costs, the other two costs increase. When the cost of activation workstations reaches 15, the slope of the change no longer varies, indicating that the solution to the problem remains unchanged. However, the equipment configuration cost decreases as the number of workstations decreases, but when the number of workstations decreases to 6, more equipment is needed to smooth the load to reduce the penalty cost.

Therefore, the variation of *F* mainly affects the number of workstations that are turned on, and has a certain impact on equipment allocation, resulting in some changes in the load.

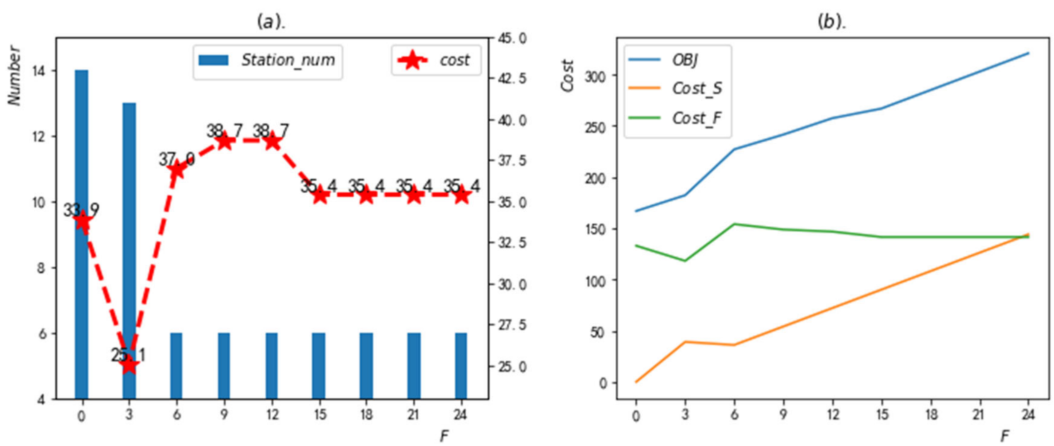

Fig. 6. The Impact of *F* Variation on the Results

5.3.4 Analysis of the impact of penalty coefficient β variation on results

To analyze the impact of penalty coefficient β on the solution results of the problem, β was increased from 0 to 10 while keeping the other parameters constant, and the problem was solved accordingly. The results are shown in Fig 7. Fig 7(a) presents the average number of activation workstations under different parameter settings using a bar chart, and the trend of penalty cost caused by load variation is described using a line chart. Fig 7(b) uses a line chart to show the objective function value *OBJ*, equipment configuration cost *Cost_F*, and workstation opening cost *Cost_S* under different parameter settings.

In Fig 7(a), it can be observed that as the penalty coefficient increases, the load range of workstations within each scenario decreases after reaching the critical point ($\beta = 4{,}7$) of the penalty coefficient change, and the total penalty cost also decreases. However, after the penalty coefficient reaches 7, the load range no longer changes, and the total penalty cost increases according to the rate of change β. From Fig 7(b), as the penalty coefficient increases, to balance the workload of workstations and reduce the load range, the equipment configuration cost of the assembly line also increases, and the workstation cost slightly increases. Moreover, every time the number of workstations increases, the equipment configuration cost also increases accordingly. After the penalty coefficient reaches 7, neither the number of workstations nor the equipment configuration result changes, and only the penalty cost increases at a fixed ratio in the total cost composition.

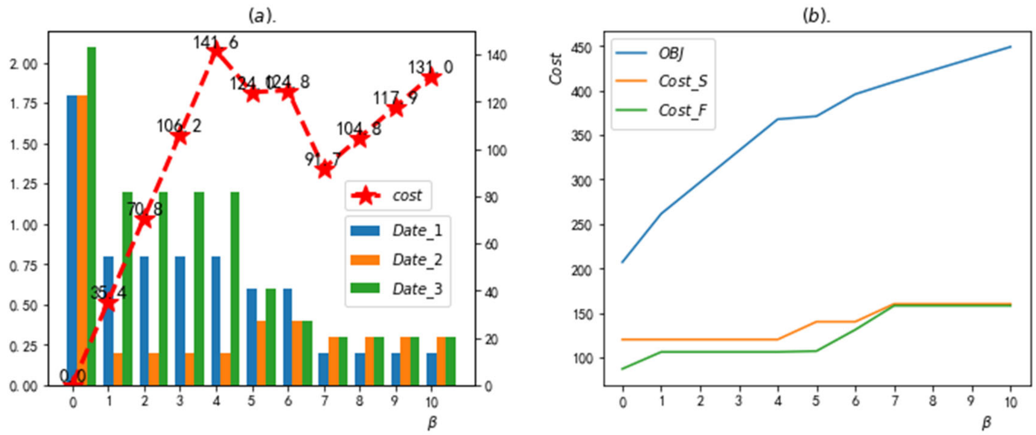

Fig. 7. The Impact of β Variation on the Results

Therefore, the variation of β has a direct impact on load balancing and equipment configuration. When β is too large, it increases the number of workstations activated and reduces the cost of configuring devices, while decreasing the extreme workload difference in each scenario. This indicates that increasing the penalty coefficient can also effectively improve the problem of excessive workload difference, but it will increase the cost of equipment configuration and the cost of opening workstations. Therefore, a trade-off needs to be made in actual decision-making or reasonable weight parameters need to be selected based on different preferences.

6. Conclusions

This paper introduces parallel tasks in the assembly line balancing problem in multi-demand scenarios, taking into account the spatial constraints of workstations and the matching relationship between tasks and workstations. The objective is to balance the assembly line layout cost and the workload between workstations. By considering the equipment configuration cost, workstation activation cost, and penalty cost for workload imbalance, a more realistic mixed integer programming model is established to address the mixed-model assembly line balancing problem.

Decision making for a multi-period problem can be divided into two stages, the first stage of decision-making involves determining the allocation of equipment and the number of workstations to be opened, while the second stage involves deciding on the product allocation plan for each cycle. To tackle the complexity of multi-cycle problems, a phased heuristic algorithm is designed to obtain approximate optimal solutions quickly. For small and medium-sized instances, partial optimal solutions can be obtained, while large-scale instances can be solved rapidly, with GAP values within an acceptable range.

Finally, sensitivity analysis was conducted to analyze the results of the four main parameters after changes. The analysis focused on the cost of equipment configuration, the cost of workstation activation, the maximum workload difference and penalty cost during each cycle. The findings provide insightful conclusions for practical management decisions in the enterprise.

In terms of future research, we could consider many relevant extensions of the assembly line problems. Such as, we assume that each task can be assigned to any workstation, but in subsequent studies, considering the more correspondence between tasks and workstations. Another interesting extension for future research is to consider how to incorporate heuristic rules into the framework of phase algorithms to reduce the difficulty of problem solving and decrease solution time. Alternatively, starting from the structure of the model, designing an exact algorithm for solving the model to improve solution efficiency and accuracy is also an intriguing avenue for further research.

Acknowledgement

This research was jointly supported by the National Natural Science Foundation of China (NSFC) under Project No. 71771028, Beijing Natural Science Foundation under Project No. 9212004 and No. Z180005, and Capital University of Economics and Business Student Academic Newcomer under Project No. 2022XSXR06.

Conflicts of interest/Competing interests

These authors contributed equally to this work and should be considered co-first authors. On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

Aguilar, H., García-Villoria, A., & Pastor, R. (2020). A survey of the parallel assembly lines balancing problem. *Computers & Operations Research, 124*, 105061.
Anuar, R., & Bukchin, Y. (2006). Design and operation of dynamic assembly lines using work-sharing. *International Journal of Production Research, 44*(18-19), 4043-4065.
Cakir, B., Altiparmak, F., & Dengiz, B. (2011). Multi-objective optimization of a stochastic assembly line balancing: A hybrid simulated annealing algorithm. *Computers & Industrial Engineering, 60*(3), 376-384.
Daneshamooz, F., Fattahi, P., & Hosseini, S. M. H. (2022). Scheduling in a flexible job shop followed by some parallel assembly stations considering lot streaming. *Engineering Optimization, 54*(4), 614-633.
Guo, Z. X., Wong, W. K., Leung, S. Y. S., Fan, J. T., & Chan, S. F. (2008). A genetic-algorithm-based optimization model for solving the flexible assembly line balancing problem with work sharing and workstation revisiting. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 38*(2), 218-228.
Hazır, Ö., & Dolgui, A. (2013). Assembly line balancing under uncertainty: Robust optimization models and exact solution method. *Computers & Industrial Engineering, 65*(2), 261-267.
Jiao, Y., Jiao, Y., Deng, X., Li, M., Xing, X., & Xu, B. (2022). Balancing of parallel U-shaped assembly lines with a heuristic algorithm based on bidirectional priority values. *Concurrent Engineering, 30*(1), 80-92.
Leiber, D., Vuong, A. T., & Reinhart, G. (2022). Alternative subgraphs assembly line balancing problem with resource selection and parallel stations. *Engineering Optimization, 54*(11), 1903-1918.
Li, J., & Gao, J. (2014). Balancing manual mixed-model assembly lines using overtime work in a demand variation environment. *International Journal of Production Research, 52*(12), 3552-3567.
Zhenping, L., Ying, S., & Lingyun, W. (2023). Research on Mixed Flow Line Balancing and Scheduling Optimization with Multiple Constraints. *Journal of System Simulation, 35*(1), 27.
Lopes, T. C., Michels, A. S., Sikora, C. G. S., Brauner, N., & Magatão, L. (2021). Assembly line balancing for two cycle times: Anticipating demand fluctuations. *Computers & Industrial Engineering, 162*, 107685.
Lopes, T. C., Michels, A. S., Sikora, C. G. S., & Magatão, L. (2019). Balancing and cyclical scheduling of asynchronous mixed-model assembly lines with parallel stations. *Journal of Manufacturing Systems, 50*, 193-200.
Mosadegh, H., Ghomi, S. F., & Zandieh, M. (2012). Simultaneous solving of balancing and sequencing problems in mixed-model assembly line systems. *International Journal of Production Research, 50*(18), 4994-5016.
Niroomand, S. (2021). Hybrid artificial electric field algorithm for assembly line balancing problem with equipment model selection possibility. *Knowledge-Based Systems, 219*, 106905.
Otto, A., Otto, C., & Scholl, A. (2013). Systematic data generation and test design for solution algorithms on the example of SALBPGen for assembly line balancing. *European Journal of Operational Research, 228*(1), 33-45.
Özcan, U. (2019). Balancing and scheduling tasks in parallel assembly lines with sequence-dependent setup times. *International Journal of Production Economics, 213*, 81-96.
Pearce, B. W., Antani, K., Mears, L., Funk, K., Mayorga, M. E., & Kurz, M. E. (2019). An effective integer program for a general assembly line balancing problem with parallel workers and additional assignment restrictions. *Journal of Manufacturing Systems, 50,* 180-192.
Salehi, M., Maleki, H. R., & Niroomand, S. (2020). Solving a new cost-oriented assembly line balancing problem by classical and hybrid meta-heuristic algorithms. *Neural Computing and Applications, 32*, 8217-8243.
Sikora, C. G. S. (2021). Benders' decomposition for the balancing of assembly lines with stochastic demand. *European Journal of Operational Research, 292*(1), 108-124.
Simaria, A. S., Zanella de Sá, M., & Vilarinho, P. M. (2009). Meeting demand variation using flexible U-shaped assembly lines. *International Journal of Production Research, 47*(14), 3937-3955.
Yang, C., & Gao, J. (2016). Balancing mixed-model assembly lines using adjacent cross-training in a demand variation environment. *Computers & Operations Research, 65*, 139-148.
Zhang, H. Y., Chen, X. Q., & Mao, N. (2020) Queuing network modelling and optimal buffer allocation for reentrant job shops. *Computer Integrated Manufacturing Systems, 26*(9), 2511.