Enhancing the labor division in the balancing of apparel assembly lines with parallel workstation through an improved ant colony algorithm

Ziang Xie¹, Jinsong Du¹,², Qingting Chen¹ and Xiaotong Wang¹

Abstract
In order to overcome the inefficiency of labor division and management confusion in garment manufacturing assembly lines when introducing parallel workstations, an improved ant colony optimization (IACO) algorithm with multi-pheromones is proposed in this paper, which provides a new strategy to improve the match degree of operators and tasks in garment assembly line with parallel workstations. In proposed IACO, the first pheromone is set for ant to improve the efficiency of lines efficiency by balancing the workload allocation among workstations, and the second pheromone determines whether to create parallel workstations by valuing the complexity distribution of tasks. Existing real data of a garment manufacturing factory is used to verify the effective of IACO and the result shows that IACO contributes to a smooth work flow, and simultaneously reduce the efficiency loss of labor divisions (ELLD). Moreover, experiments are conduct to explore the trade-off relation between labor division and line arrangement efficiency when introducing parallel workstations. And based on the relation, strategies to create parallel workstations in the assembly line is provided for garment factories with different staff qualities for different apparel products.

Keywords
Apparel assembly line, parallel workstation, labor division, ant colony optimization

Introduction
An apparel assembly line is a flow-oriented production system with a series of workstations arranged along a transportation system,¹ usually a conveyor belt or hanging system.²,³ Specific tasks are assigned to each workstation and workpieces visit workstations successively,⁴ so that a product can be completed within a given cycle time.⁵ The assembly line system was originally designed to exploit a high specialization of labor, increase efficiency, and reduce the cost of batch production.⁶,⁷ However, a trend in the apparel industry is mass-customized production with unpredictable demand fluctuations. Consequently, flexibility, and quick response abilities have become increasingly crucial attributes of the assembly line for apparel producers.

The aim of the assembly line balance problem (ALBP) is to assign tasks to workstations while optimizing one or more objectives.⁸ The fundamental assumption of a

¹College of Fashion Design, Donghua University, Shanghai, China
²Key Laboratory of Clothing Design and Technology, Donghua University, Ministry of Education, Shanghai, China

Corresponding author:
Jinsong Du, Key Laboratory of Clothing Design and Technology, Donghua University, Ministry of Education, 1882 West Yan’an Road, Changning District, Shanghai 200051, China.
Email: ducccp@dhu.edu.cn
traditional ALBP is usually a linear layout with no parallel workstations in which configuration is inflexible as each task is only assigned to one station or operator.\(^9\) In order to address the restrictions of the traditional ALBP, researchers have developed several alternative practical configurations that can provide additional flexibility in the face of changing demands. Ünal et al.\(^10\) set up the production line as a U-type line and proposed a heuristic line balancing algorithm to balance the line, concluding that the U-type line configuration is more advantageous than the linear configuration based on the average throughput per worker and average workstation utilization values. Kucukkoc and Zhang\(^11\) first proposed an assembly line design with a mixture of parallel and U-type lines, and the proposed heuristic algorithm. Ozbakir et al.\(^12\) developed a new multiple swarm ant colony algorithm to balance a dual-objective parallel assembly line, which effectively solved the balance problem of the assembly line. Lusa\(^13\) focused on a production system consisting of multiple parallel assembly lines and discussed the advantages and disadvantages of using multiple assembly lines, and described the multiple assembly line balancing problem and its related characteristics. In summary, assembly lines can be classified as U-type lines (U-ALBP), parallel lines (P-ALBP), and lines with parallel workstations (PW-ALBP),\(^14,15\) depending on the line arrangement and the parallelism of the lines or workstations.

Figure 1 provides examples of workforce planning among these configurations.\(^4\)

![Figure 1. Examples of different configurations for assembly lines.](image)

Apparel production is characterized by low automation and high specialization.\(^16\) Most tasks (such as sewing, overlocking, blind stitching, etc.) require operators to work in a sitting position and makes it inconvenient for operators to turn around as they work in the U-type or parallel line configurations. In addition, some tasks are integrated, and operators working cooperatively in a multi-manned workstation can barely improve their efficiency. By introducing parallel workstations to apparel assembly lines, operators in two or more replicas of a workstation can perform the same set of tasks.\(^17\) One benefit of this configuration is the reduction of the non-value-added activities, as each operator works at only one station. By allowing tasks on the same workpiece to be performed simultaneously in parallel, the cycle time and the delivery time can be reduced. Moreover, this configuration can increase the flexibility of the assembly line design as practitioners can reproduce or remove a parallel station according to the requirements of various products.\(^18\) As a result, an assembly line with multiple parallel stations is a more practical configuration for apparel production.

Numerous research efforts have attempted to solve the PW-ALBP, usually by applying the methods of Mathematical Programming,\(^14,17\) heuristics and meta-heuristics. Compared with Mathematical Programming techniques meta-heuristics show strong ability to solve large-scale problem\(^19,20\) and it is widely used in the area of ALBP. For instance, Askin and Zhou\(^21\) developed a heuristic to create parallel workstations and using a threshold variable to evaluate the utilization of parallel stations at the target level, which is applicable to solving problems with mixed models. Aydemir-Karadag and Turkbey\(^22\) proposed a genetic algorithm to solve a multi-objective optimization problem for assembly lines. Akpinar and Mirac Bayhan\(^23\) proposed a hybrid genetic algorithm to improve the workload smoothing between workstations, and optimize the workload smoothing within workstations at the same time for mixed model ALBP. Vilarinho and Simaria\(^24\) developed an ant colony optimization algorithm (ACO) to deal with the PW-ALBP with partitioning constraints for minimizing workstations and workload variations among parallel workstations. However, when introducing parallel workstations, the number of tasks performed by each worker increases, which may result in inefficiencies of the labor division.\(^25\) Schofield\(^26\) and Sarker and Shanthikumar\(^27\) used the way of setting the number of parallel workstations to solve the problem of inefficient division of labor. In contrast, Buxey\(^18\) used a way to limit the number of tasks...
per workstation to improve the labor division efficiency. McMullen and Frazier allowed to create a parallel workstation as long as its utilization increases; Vilarinho and Simaria allowed to create a parallel workstation only if process time exceeds the predefined limit. The trigger mechanism for the decision to create a parallel workstation, however, depends mainly on experience, and the indicators to evaluate the efficiency of labor division is insufficient, especially for garment production.

In this research, we intended to develop an improved ACO algorithm (IACO) to optimize the workload balance and the labor division for PW-ALBP of apparel production. The main idea of the algorithm is to set an addition pheromone to decide parallel workstations establishment based on the matching degree between the tasks complexity and the operator skill level (OSL). Furthermore, based on the existing real data, we conduct a series of experiments to explore feasible policies for creating parallel workstations for different type of clothes and for operators of different proficiency levels.

**Factors that influence apparel assembly line arrangement**

A piece of clothing usually consists of several components, like the collar, cuff and sleeve of a shirt, for example. In apparel production, the components of an item are manufactured before they are assembled into complete apparel products. This process can be divided into a sequence of tasks. Task attributes including standard time, complexity level, and the precedence relation with other tasks determine the assignment of the task to suitable workstations and operators, and the assembly line layout follows these relations. Several factors that affect the apparel assembly line arrangement are relevant, as follows:

**Labor factors**

The two key dimensions of the labor factor are OSL and operator proficiency (OP). As apparel manufacturing is a labor-intensive industry with high specialization, there are various kinds of tasks ranging from simple (ironing, trimming, etc.), to complex (sleeve attachment, topstitching, etc.) and each task requires an operator with a specific skill level. In practical production, OSL corresponds to the task complexity level, which means an operator with higher OSL can perform all tasks of equal or lower complexity. When a simple task is assigned to an operator with a high OSL, the efficiency of labor division decreases.

As automation levels in apparel manufacturing are relatively low, non-standard movements are inevitable as operators take, place and hold work pieces or perform other operations which result in the floating time in production. This floating time is typically associated with OP, and an operator with a higher OP has less floating time when performing tasks.

**Precedence relation of tasks**

The process of apparel production consists of one main branch (trunk) and several branches leading into the trunk. Each branch refers to the component manufacturing process, while the trunk is the final assembly process. The precedence relation of the branch structure illustrates the sequence of task assignments. According to the precedence relation, one task can be assigned to a workstation only when its preceding tasks have been completed in the same workstation or former workstations.

Because a complicated apparel product contains more branch tasks than a simple one, the scale of the problem extends as the complexity of the product increases.

**Parallel station arrangement**

In an assembly line with parallel workstations, a workstation group consists of one ordinary workstation or a set of parallel workstations. Efficiency losses of the labor division in an assembly line can be reduced by assigning fewer, more similar tasks to a workstation group with parallel workstations. As a result, operations with more similar tasks and longer process times are more likely to be assigned to parallel workstations. In addition, a limit on the number of the parallel workstations and the replicas of a workstation should be set to maintain the advantage of the labor division in assembly lines.

**Problem definition**

In apparel production, the cycle time is usually designed to satisfy the planned output. As a result, the aim of ALBP for apparel production is typically to balance the workload (process time for total tasks in the workstation group) among workstations and to allocate suitable operators to perform the required tasks within the given cycle time. A common index to evaluate the workload balance is the smooth index (SI), and many researches for apparel ALBP have focused on minimizing the SI to improve the efficiency of the apparel assembly line. This study introduces an index to evaluate the efficiency loss of labor division (ELLD) for apparel production. Based on the SI and ELLD, we set the dual objectives of (a) smoothing the workload of PW-ALBP apparel assembly lines and (b) avoiding efficiency losses of the labor division when introducing parallel workstations.

The assumptions associated with the problem definition are:

1. The cycle time of the assembly line is fixed;
2. A workstation group consists of one ordinary workstation or a set of parallel workstations. Assigning a task to a workstation group means to assign the task to either an ordinary workstation or to all parallel workstations simultaneously;
3. Each workstation (parallel or ordinary) is assigned to only one operator;
4. The maximum workload (process time that allows for total tasks in the workstation group) of a workstation group is determined by the parallel workstations it contains. For example, the maximum workload of a workstation group with two parallel workstations is double the cycle time;
5. The precedence relation and process time of tasks are known;
6. The OSL and OP of an operator is known;
7. A task cannot be performed unless its predecessor task is completed.

The notation used for the mathematical formulation is described in Table 1.

Decision variables are $x_{ig}$, $y_{gu}$, and $z_{pk}$, representing the decision of task assignment, parallel workstation arrangement and operator assignment, respectively. The detail is described as follows.

$$
x_{ig} = \begin{cases} 
1 & \text{if task } i \text{ is assigned to group } g \\
0 & \text{otherwise}
\end{cases}$$

$$
y_{gu} = \begin{cases} 
1 & \text{g contains u workstation(s)} \\
0 & \text{otherwise}
\end{cases}$$

$$
z_{pk} = \begin{cases} 
1 & \text{if operator } p \text{ is assigned to workstation } k \\
0 & \text{otherwise}
\end{cases}$$

The objective is to minimize both the SI and the ELLD.

As the SI directly relates to the efficiency of the assembly line, this study sets SI as the key objective and ELLD as the non-key objective. The objective functions are described as equations (1) and (2) and the constraints are given in equations (3)–(8).

Minimize $SI = \sqrt{\sum_{g=1}^{G} C - \sum_{u=1}^{w} \sum_{i=1}^{N} t_i \cdot x_{ig} / \sum_{w} W \cdot y_{gu}}$ (1)

Minimize $ELLD = \sqrt{\frac{1}{K} \sum_{k=1}^{K} (1 - \frac{\sum_{i=1}^{N} d_i \cdot x_{ik}}{\text{num}_k \cdot \max(k_d)})}$ (2)

st.

$$
\sum_{g=1}^{G} \sum_{u=1}^{U} w y_{gu} = K
$$ (3)

$$
\sum_{g=1}^{G} x_{ig} = 1 \quad i = 1, 2, ..., N
$$ (4)

$$
\sum_{k=1}^{K} z_{pk} = 1 \quad k = 1, 2, ..., K
$$ (5)

$$
OSL_p = \max \{ CL_i \} \quad i \in S_k
$$ (6)

$$
\sum_{i=1}^{K} t_i x_{ig} \leq C \cdot \sum_{u=1}^{U} u y_{gu} \quad g = 1, 2, ..., G
$$ (7)

$$
\sum_{g=1}^{G} (k \cdot x_{ig} - k \cdot x_{ig}) \geq 0 \quad j = 2, 3, ..., N; i \in P(j)
$$ (8)
The objective function (1) is set to try and eliminate differences of workload among workstations. The objective function (2) is set to reduce the mismatch between OSL and the complexity level in a workstation. Constraint (3) ensures that the total number of workstations on the assembly line is the sum of the parallel workstation and ordinary workstation. Constraint (4) ensures that each task is only assigned to an ordinary workstation or to a set of parallel workstations. Constraint (5) ensures that each workstation is assigned to only one operator. Constraint (6) ensures that the assigned operator is able to do the most complex task in the workstation. Constraint (7) is the cycle time constraint which ensures that the total time of a workstation group cannot exceed its capacity (the upper limit of total process time for workstation group). Constraint (8) ensures that all precedence relations amongst tasks are satisfied.

**Proposed new ant colony optimization algorithm**

The ACO algorithm was first presented by Dorigo et al.\cite{1} for solving nondeterministic polynomial problems. In an ACO algorithm, each ant systematically builds a single solution based on the transition rule, which is the combination of heuristic information and the pheromone trail (a previous solution that the ant obtained). When the ant colony generates solutions for the problem, the pheromone trail is updated and guides the ants to find a better solution in the next iteration.

In PW-ALBP, a solution must contain both task assignment and parallel workstation creating strategy. However, in traditional ACO an ant only accounts for task assignment process,\cite{2,3} while it depends mainly on experience when decide whether to create a parallel workstation. In this study, we set two transition rules for an ant to arrive at a feasible solution; accounting for the task assignment and the parallel workstation arrangement process, respectively. An outline of the proposed algorithm is presented in Figure 2.

**Construction of the solution using the proposed new ACO algorithm**

The procedure for an ant to build a solution is as follows:

**Start: Initializing**

Set the initial parameters, including the cycle time, the admitted maximum number of parallel workstations in each workstation group (U), the workstation counter \( k = 0 \), the workstation group counter \( g = 0 \), and parallel workstation counter \( u = 0 \) for the workstation group.

**Step 1 Open a new workstation group**

Open a new workstation group. \( g \leftarrow g + 1 \); go to Step 2

**Step 2 Open a new workstation**

Open a new workstation. \( k \leftarrow k + 1 \); \( u \leftarrow u + 1 \); go to Step 3

**Step 3 Determine a candidate task list**

Set a candidate task list for the current workstation. In the proposed approach, a task is a candidate if:

1. the task is unassigned;
2. Its predecessor tasks are assigned to the same workstation group or to a former workstation group, according to Constraint (8).

In this approach, the candidate task list is empty only when all tasks are assigned, which means that a solution is built and the solution construction procedure of the ant has ended. If the candidate task list is not empty, go to Step 4.

**Step 4. Select a task for the current workstation group**
In this step, an ant will select one task from the candidate task list according to a transition rule-I:

\[
\text{task } i = \begin{cases} 
\arg\max_{\text{scallowed}}(\tau_{ig}^1)^\alpha (\eta_{ig}^1)^\beta & (q \leq q_0) \\
\frac{(\tau_{ig}^1)^\alpha (\eta_{ig}^1)^\beta}{\sum_{\text{scallowed}}(\tau_{ig}^1)^\alpha (\eta_{ig}^1)^\beta} & (q > q_0)
\end{cases}
\] (9)

Where \(\tau_{ig}^1\) is the intensity of the pheromone trail for selecting a task; \(\alpha\) and \(\beta\) represent the importance of the pheromone intensity and heuristic information, respectively; \(q\) is a uniformly distributed random number in the range of 0–1 and \(q_0\) is a predefined parameter. The heuristic information of task \(i\) is \(\eta_i\). In this study, the heuristic information used by ants is the Ranked Positional Weights Method introduced by Helgeson and Birnie:36

\[
\eta_i = \frac{P_{W_i}}{\sum_{\text{scallowed}} P_{W_i}}
\] (10)

\[
P_{W_i} = t_i + \sum_{j \in S(i)} f_{ij}
\] (11)

In this method, a task takes precedence if it and its successor tasks have longer process times.

According to Constraint (7), if the total task time does not exceed the capacity of the current workstation group when task \(i\) is assigned, assign task \(i\) to the \(u\)th workstation of workstation group \(g\). This workstation is also the workstation \(k\) in assembly lines. Then, repeat this step. Otherwise, go to Step 5.

**Step 5 Assign an operator to the workstation**

Assign an operator to workstation \(k\), subject to Constraint (6). Go to Step 6.

**Step 6 decide whether to create a parallel workstation**

A parallel workstation will be created if:

1. \(u < U\), which means that the number of parallel workstations in the current workstation group is less than the permitted maximum.
2. if the ant chooses to create a parallel workstation according to transition rule-I:

\[
P_{II}^{go} = \frac{(\tau_{go}^I)^\alpha (\eta_{go}^I)^\beta}{\sum_{\text{go}(0,1)}(\tau_{go}^I)^\alpha (\eta_{go}^I)^\beta}
\] (12)

\[
\eta_{go}^I = \begin{cases} 
\frac{T_p + t_i}{(w + o) \cdot C} & o = 1 \\
\frac{T_p}{(w + o) \cdot C} & o = 0
\end{cases}
\] (13)

\(T_p\) is the process time of the current workstation; \(o\) is the binary variable in \(\{0,1\}\), in which \(o = 1\) will open a parallel workstation and \(o = 0\) will not open one. \(P_{II}^{go}\) is the probability of workstation group \(g\) opening a parallel workstation. \(\tau_{ig}^1\) and \(\eta_{ig}^1\) represent the intensities of the pheromone trail and the heuristic information, respectively. Equation (13) determines that the probability of opening a parallel workstation increases if the gap between the workload and the capacity of workstation group \(g\) narrows when opening a new parallel workstation for task \(i\).

In this step, if the ant chooses to create a parallel workstation, assign the selected task to workstation group \(g\), then go back to Step 2. Otherwise, if all tasks are assigned, loop terminates, or else go back to Step 1.

**Pheromone trail update**

The proposed strategy for pheromone updating is based on the Ant Colony System.37 There are two different strategies to update \(\tau_{ig}^1\) and \(\tau_{go}^I\).

**Strategy 1.** Each ant performs the local pheromone update after each solution construct. The local pheromone for task \(i\) and task \(j\) updates in accordance with the following functions:

\[
\tau_{ig}^1(n) = (1 - \varphi) \cdot \tau_{ig}^1(n - 1) + \varphi \cdot \tau_{ig}^1_0
\] (14)

\[
\tau_{go}^I(n) = (1 - \varphi) \cdot \tau_{go}^I(n - 1) + \varphi \cdot \tau_{go}^I_0
\] (15)

where \(\tau_{ig}^1_0\) and \(\tau_{go}^I_0\) are the initial pheromone.

**Strategy 2.** Iteration-best ants perform a global pheromone update once all ants have completed their tours. An iteration-best solution is predicted to deliver good performance for both SI and ELLD. The procedure to find the iteration-best solution includes two steps:

Step 1: Find the solutions with minimum SI in all solutions;

Step 2: Find the solutions with minimum ELS among the solutions identified in Step 1.

The global pheromone between task \(i\) and task \(j\) updates according to the following functions:

\[
\tau_{ig}^1(n) = (1 - \rho) \cdot \tau_{ig}^1(n - 1) + \rho \cdot \Delta \tau_{ig}^1(n)
\] (16)

\[
\tau_{go}^I(n) = (1 - \rho) \cdot \tau_{go}^I(n - 1) + \rho \cdot \Delta \tau_{go}^I(n)
\] (17)

\[
\tau_{ig}^1 = \begin{cases} 
\frac{1}{SI_{\min}} (i, j) \in \text{iteration-best} \\
0 & \text{otherwise}
\end{cases}
\] (18)
Data collection

Data were collected from the sewing department of a garment factory in south-east China. This factory produces three kinds of apparel products, namely shirts, jackets, and suits. A description of the production parameters of each product is provided in Table 2.

Shirt products consist of several components, including: two front-pieces, a front placket with buttons, one back-piece, one patch pocket, two shirt sleeves and one shirt collar. Jacket components include: two front-pieces, a zipper, two back-pieces, two side-pieces, two insert pockets, two set-in sleeves with rib-knit cuffs, one stand collar, and the corresponding lining part. Suit components include: two front-pieces, two back-pieces, two side pieces, one breast pocket, two insert pockets with flap, two set-in sleeves, one lapel collar, and the corresponding lining part. In the actual production environment, components are produced simultaneously and then assembled as a completed product. Figures 3 to 5 is the structure of precedence graph for these three apparel products, respectively. The process time and the coefficient level of each task are also shown.

Experimental model settings

An experiment was conducted to evaluate the performance of the IACO algorithm for constructing solutions based on different apparel products, the attendant OP’s, and the policies of parallel workstation assignment.

Operator proficiency

Operators can be ranked according to their operator proficiency coefficient. The OP determines the process time of an operator according to the following function:

$$T_{iN} = \frac{t_i}{\alpha}$$  \hspace{1cm} (19)

where $\alpha$ is the operator proficiency coefficient.

Parallel workstation assignments

In a practical production environment, there are limits to the number of parallel workstations in a workstation group and the total number of parallel workstations on an assembly line. Typically, production disorders will occur if more than three operators process the same workpiece simultaneously. Thus, it is appropriate to set an upper limit of three parallel workstations for a workstation group. On this basis, the workstations can be divided into three types: ordinary workstation (OW), parallel workstations of type 1 (PW-1), and parallel workstations of type 2 (PW-2), according to the workstation group they belong to. OW is the workstation in a workstation group with an ordinary workstation. PW-1 and PW-2 are the workstation in a workstation group with two and three parallel workstations, respectively.

Additionally, to enhance the labor division, an upper limit should be set for the number of parallel workstations in an assembly line. In this study, the maximum number of parallel workstations ($PW_{max}$) is determined by the following functions:

---

### Table 2. Description of the production parameters.

| Type of product | Number of tasks | Total process time (s) | Theoretical cycle time (s) | Number of workers on the line |
|-----------------|-----------------|------------------------|---------------------------|------------------------------|
| Shirt           | 41              | 1227                   | 155                       | 8–10                         |
| Jacket          | 65              | 2408                   | 185                       | 13–15                        |
| Suit            | 111             | 5286                   | 260                       | 21–23                        |

---

Figure 3. Precedence graph for shirt.
Figure 4. Precedence graph for jacket.

Figure 5. Precedence graph for suit.
\[ PW_{\text{max}} = THW \cdot \gamma \]  

\[ THW = \left[ \frac{\sum_{i \in N} t_i}{C} \right] \]  

where \( THW \) is the theoretical minimum number of workstations on the assembly line; \( \gamma \) is the parameter in the range of 0–1 to control the number of parallel workstations on the assembly line, and \( \lceil x \rceil \) denotes the smallest integer larger than or equal to \( x \).

**Experimental design**

An 3 \( \times \) 3 \( \times \) 3 experiment was designed including three factors: types of apparel products (factor A with three levels: shirt, jacket, suit); OP (factor B with three levels: \( \alpha = 0.95 \), 1, 1.05) and the proportion of parallel workstations on the assembly line (factor C with three levels: \( \gamma = 0\% \), 25\%, 50\%). There were 27 experiment sets and each set of experiments was repeated 10 times. The parameters of \( \alpha \), \( \beta \), \( \phi \), \( \rho \), and \( q_0 \) were set to 0.5, 1, 0.1, 0.2, and 0.5 respectively, according to Lale Ozbakir et al.\(^\text{12}\) and Sabuncuoglu I et al.\(^\text{39}\) The proposed ACO algorithm was implemented in Python 3.7.

**Data analysis**

The results of IACO optimized solutions and the company’s solutions are shown in Table 3. company’s solutions are historical data provided by sewing line supervisor. Supervisors decide the proportion of parallel workstations and assign tasks to workstations based on experience. As a result there is no constraint of \( \gamma \) for company’s solutions.

The optimal value and the mean value of the SI, ELLD, the line arrangement efficiency (LAE) and the number of workstations (K) are shown in this table. The LAE is computed by function (22):

| Types (A) | \( \gamma \) (B) | OP (C) | SI | ELLD | LAS | K |
|-----------|-----------------|--------|----|------|-----|---|
| Shirt     | 0               | 0.95   | 8.83 | 8.83 | 9.79 | 22.72 |
| Shirt     | 0.25            | 0.95   | 6.78 | 7.01 | 24.42 | 26.81 |
| Shirt     | 0.5             | 0.95   | 5.39 | 5.76 | 24.63 | 29.79 |
| Shirt     | 0               | 1      | 9.22 | 9.22 | 22.93 | 23.46 |
| Shirt     | 0.25            | 1      | 7.00 | 7.24 | 24.73 | 25.64 |
| Shirt     | 0.5             | 1      | 5.74 | 6.60 | 25.21 | 27.89 |
| Shirt     | 0               | 1.05   | 6.08 | 6.70 | 16.83 | 18.37 |
| Shirt     | 0.25            | 1.05   | 6.71 | 6.77 | 14.70 | 17.63 |
| Shirt     | 0.5             | 1.05   | 4.90 | 5.45 | 20.22 | 25.51 |
| Jacket    | 0               | 0.95   | 18.87 | 18.87 | 20.81 | 18.23 |
| Jacket    | 0.25            | 0.95   | 12.85 | 13.21 | 21.79 | 26.64 |
| Jacket    | 0.5             | 0.95   | 9.00 | 11.10 | 19.72 | 26.66 |
| Jacket    | 0               | 1      | 11.83 | 12.01 | 17.12 | 22.28 |
| Jacket    | 0.25            | 1      | 10.86 | 11.21 | 22.07 | 24.41 |
| Jacket    | 0.5             | 1      | 9.22 | 9.67 | 22.75 | 25.06 |
| Jacket    | 0               | 1.05   | 10.10 | 10.84 | 11.14 | 19.44 |
| Jacket    | 0.25            | 1.05   | 8.94 | 9.00 | 27.26 | 31.56 |
| Jacket    | 0.5             | 1.05   | 6.56 | 7.32 | 29.33 | 32.53 |
| Suit      | 0               | 0.95   | 20.42 | 20.42 | 25.53 | 25.94 |
| Suit      | 0.25            | 0.95   | 17.35 | 18.15 | 25.57 | 30.03 |
| Suit      | 0.5             | 0.95   | 12.29 | 13.67 | 31.70 | 35.04 |
| Suit      | 0               | 1      | 13.19 | 18.20 | 20.52 | 27.17 |
| Suit      | 0.25            | 1      | 11.66 | 12.20 | 30.89 | 32.59 |
| Suit      | 0.5             | 1      | 10.3 | 10.64 | 32.91 | 35.78 |
| Suit      | 0               | 1.05   | 13.67 | 16.10 | 19.87 | 27.43 |
| Suit      | 0.25            | 1.05   | 13.30 | 14.35 | 30.61 | 32.20 |
| Suit      | 0.5             | 1.05   | 11.47 | 12.37 | 33.82 | 35.86 |

\[ \text{Table 3. The results of the proposed algorithm for different OP and } \gamma. \]
The results in Table 3 demonstrate that there is little difference between the optimal and mean values of each index for IACO, indicating the applicability and stability of the algorithm.

The comparison between IACO and company’s solutions are shown in Figures 6 to 8. The best and the worst solutions of IACO are the solution with optimal and worst SI in all level of \( \gamma \) at each OP and apparel type, respectively.

\[
LA_E = \sum_{g=1}^{G} \left( \frac{t_i}{g_w} \right) \times 100\% \tag{22}
\]

Figures 6 to 8 show the superiority of the IACO in SI compared with supervisor’s experience. The superiority of IACO is more significant as apparel complexity increases. When \( OP = 1.05 \), the best solution that IACO gives only achieves 37.26% SI improvement and has no improvement in \( K \) compared with the supervisor’s experience, however, the SI improve 41.22% and 42.27%, respectively in jacket and suit assembly lines with 1–2 reduce in \( K \). When \( OP \) decrease, the performance disparity between IACO and supervisor’s experience will increase.

The ELLD of best solutions by IACO are relatively low compared with supervisor’s experience because the substitution effect between the labor division and workload balanced of each workstation in assembly line.

Data were analyzed by the multivariate analysis of variance (MANOVA) with a significance level of 0.05. Table 4 presents a summary of the MANOVA result.

| Factors | SI | ELLD | LAE | K |
|---------|----|------|-----|---|
| A       | Y  | Y    | Y   | Y |
| B       | Y  | N    | Y   | Y |
| C       | Y  | Y    | Y   | Y |
| A \times B | Y | Y    | Y   | Y |
| A \times C | Y | Y    | Y   | Y |
| B \times C | Y | Y    | Y   | Y |
| A \times B \times C | Y | Y    | Y   | Y |

Y: significant to performance index; N: insignificant to performance index. A: apparel type; B: operator proficiency; C: number of parallel workstations.
The interactions among apparel type, operator proficiency and proportion of parallel workstation is significant for SI, ELLD, LAE, and K. This demonstrates that the optimal strategy of parallel workstation assignment can differ, depending on to the apparel type and personnel quality of the factory.

Further comparisons were made to explore the relationship between SI and ELLD. Figures 9 to 11 show the solutions with minimum SI of each experiment set for shirt, jacket, and suit, respectively. The SI, ELLD, LAE, and number of workstations, including OW, PW-1, and PW-2, of these solutions are also shown.

Figures 9 to 11 show the number of workstations and workers required for the assembly line by increasing OP. With the increase of OP from 0.95 to 1.05, the number of workstations required for the shirt assembly line is reduced from 9 to 8, while for the jacket, and suit assembly lines it reduces by 2–3.

The introduction of parallel workstations increases the line arrangement efficiency, but results in a decrease of labor division efficiency. The substitution effect is significant when workers have lower operator proficiency (OP). For example, for the jacket assembly line with 0.95 OP, as γ increases from 0% to 50%, the SI decreases by more than half (52.31%) from 18.87 to 9. LAE simultaneously increases by 10.62% from 87.57% to 96.87%. Meanwhile, ELLD also increases from 18.23% to 22.17%, while performance decreases by 21.61%. This substitution effect is also significant for the suit assembly line at 0.95 OP and for the shirt assembly line when the OP is 1 or less. However, as the OP reaches a certain level, the efficiency of the assembly line can only be improved marginally with the introduction of parallel workstations, but the labor division efficiency decreases significantly. On the jacket assembly line with 1 OP, for example, when γ increases from 0% to 50%, the SI decreases from 11.83 to 9.22, a
decrease of 22.06%, while LAE increases from 94.51% to 96.59%, only 2.2%. In this process, however, ELLD increases from 19.44% to 23.16%, and the overall performance decreases by 19.13%. This also applies to the shirt and jacket assembly lines when OP reaches 1 and 1.05, respectively. Furthermore, for shirt assembly line when OP is 1.05 and \( \gamma \) is 0.25, introducing parallel workstation cannot help to improve the workload balance and the line efficiency and the algorithm chooses to build a solution with no parallel workstation.

Based on above analysis, the proper number of parallel workstations on an assembly line can be determined according to the substitution effect between SI and ELLD. A strategy matrix of parallel workstation assignments is presented in Figure 12.

Figure 12 shows the recommend proportion (high: more than 25% but less than 50%; low: less than 25%) of parallel workstations on an assembly line for different apparel types and OP. The proportion is determined by the rate of increase in SI (Significant: more than 5%) and
Results and discussion

In this work, we take into account the real two-objectives and multi-constraints of the PWALBP problem to optimize the workload allocation of workstations, minimize the loss of labor-division. As the labor division inefficient might occur when establish a parallel workstation, an improved IACO algorithm is designed to determine the number of ordinary and parallel workstations automatically. In IACO, two transition strategies are set for an ant to assign tasks and deciding whether to create a parallel workstation, respectively. Moreover, considered the application for apparel production in the real world, a strategy for parallel workstation assignment was concluded for apparel factories with different personnel quality in a production line to product different apparel products.

Acknowledgements
The authors are grateful to the company who provided the production data and insights for this research.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs
Ziang Xie https://orcid.org/0000-0002-7037-0598
Jinsong Du https://orcid.org/0000-0002-5806-0810

References
1. Chan KCC, Hui PCL, Yeung KW, et al. Handling the assembly line balancing problem in the clothing industry using a genetic algorithm. Int J Clothing Sci Technol 1998; 10(1): 21–37.
2. Cui H and Yan Y. The optimization design of uniform’s hanging production line. Int J Clothing Sci Technol 2015; 27(3): 370–389.
3. Hu LY, Huo YR, Jin CQ, et al. One-Piece flow implementation research in the garment enterprises. Adv Mater Res 2013; 655–657: 2352–2357.
4. Boysen N, Fiedler M and Scholl A. A classification of assembly line balancing problems. Eur J Oper Res 2007; 183(2): 674–693.
5. Kellegöz T and Toklu B. An efficient branch and bound algorithm for assembly line balancing problems with parallel multi-manned workstations. Comput Oper Res 2012; 39(12): 3344–3360.
6. Scholl A and Klein R. ULINO: optimally balancing U-shaped JIT assembly lines. Int J Prod Res 1999; 37(4): 721–736.
7. Shtub A and Dar-el EM. A methodology for the selection of assembly systems. Int J Prod Res 1989; 27(1): 175–186.
8. Kim YK, Song WS and Kim JH. A mathematical model and a genetic algorithm for two-sided assembly line balancing. Comput Oper Res 2009; 36(3): 853–865.
9. Miltenburg GJ and Wijnbergaard J. The U-line line balancing problem. Manage Sci 1994; 40(10): 1378–1388.
10. Ünal C, Tunali S and Güner M. Evaluation of alternative line configurations in apparel industry using simulation. Text Res J 2009; 79: 908–916.
11. Kucukkoc I and Zhang DZ. Balancing of parallel U-shaped assembly lines. Comput Oper Res 2015; 64: 234–244.
12. Ozbakir L, Baykasoglu A, Gorkemli B, et al. Multiple-colony ant algorithm for parallel assembly line balancing problem. Appl Soft Comput 2011; 11(3): 3186–3198.
13. Luca A. A survey of the literature on the multiple or parallel assembly line balancing problem. Eur J Ind Eng 2008; 2(1): 50–72.
14. Ege Y, Azizoglu M and Ozdemirel NE. Assembly line balancing with station paralleling. Comput Ind Eng 2009; 57(4): 1218–1225.
15. Becker C and Scholl A. Balancing assembly lines with variable parallel work-places: problem definition and effective solution procedure. Eur J Oper Res 2009; 199: 359–374.
16. Stojanov T and Ding X-M. Operators skill level evaluation method for balancing of an apparel assembly line. Int J Productivity Manag Assess Technol 2015; 3(1): 1–12.
17. Bakchin J and Rubinovitz J. A weighted approach for assembly line design with station paralleling and equipment selection. IIE Trans 2003; 35(1): 73–85.
18. Buxey GM. Assembly line balancing with multiple stations. Manage Sci 1974; 20: 1010–1021.
19. Deng W, Xu J, Zhao H, et al. A novel gate resource allocation method using improved PSO-Based QEA. IEEE Trans Intell Transp Syst 2020; 9: 1–9. DOI: 10.1109/TITS.2020.3025796
20. Cai X, Zhao H, Shang S, et al. An improved quantum-inspired cooperative co-evolution algorithm with multi-strategy and its application. Expert Syst Appl 2021; 171: 114629.
21. Askin RG and Zhou M. A parallel station heuristic for the mixed-model production line balancing problem. Int J Prod Res 1997; 35: 3095–3106.
22. Aydemir-Karadag A and Turkbey O. Multi-objective optimization of stochastic disassembly line balancing with station paralleling. *Comput Ind Eng* 2013; 65(3): 413–425.

23. Akpınar S and Mirac Bayhan G. A hybrid genetic algorithm for mixed model assembly line balancing problem with parallel workstations and zoning constraints. *Eng Appl Artif Intell* 2011; 24(3): 449–457.

24. Vilarinho PM and Simaria AS. ANTBAL: an ant colony optimization algorithm for balancing mixed-model assembly lines with parallel workstations. *Int J Prod Res* 2006; 44: 291–303.

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

26. Schofield NA. Assembly line balancing and the application of computer techniques. *Comput Ind Eng* 1979; 3(1): 53–69.

27. Sarker BR and Shanthikumari JG. A generalized approach for serial or parallel line balancing. *Int J Prod Res* 1983; 21(1): 109–133.

28. McMullen PR and Frazier GV. Using simulated annealing to solve a multiobjective assembly line balancing problem with parallel workstations. *Int J Prod Res* 1998; 36: 2717–2741.

29. Tasan SO and Tunali S. A review of the current applications of genetic algorithms in assembly line balancing. *J Intell Manuf* 2008; 19: 49–69.

30. Hou L, Wu YM, Lai R-S, et al. Product family assembly line balancing based on an improved genetic algorithm. *Int J Adv Manuf Technol* 2014; 70(9-12): 1775–1786.

31. Xu H, Xu B and Yan J. Balancing apparel assembly lines through adaptive ant colony optimization. *Text Res J* 2019; 89(18): 3677–3691.

32. Xu YN, Zhou HM, Wang LC, et al. Simulation of clothing sewing line and evaluation indexes. *J Text Res* 2015; 36(12): 146–151.

33. Dorigo M, Maniezzo V and Colorni A. Ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern B Cybern*, 1996; 26: 29–41.

34. Baykasoğlu A and Dereli T. Simple and U-type assembly line balancing by using an ant colony based algorithm. *Math Comput Appl* 2009; 14: 1–12.

35. Yagmahan B. Mixed-model assembly line balancing using a multi-objective ant colony optimization approach. *Expert Syst Appl* 2011; 38: 12453–12461.

36. Helgeson WB and Brine DP. Assembly line balancing using ranked positional weight technique. *J Ind Eng* 1961; 12: 394–398.

37. Mullen RJ, Monekosso D, Barman S, et al. A review of ant algorithms. *Expert Syst Appl* 2009; 36(6): 9608–9617.

38. Sun YH and DU Js. Simulation of one-piece flow garment assembly line based on Flexsim software. *J Text Res* 2018; 39(06): 155–161.

39. Sabuncuoglu I, Erel E and Alp A. Ant colony optimization for the single model U-type assembly line balancing problem. *Int J Prod Econ* 2009; 120: 287–300.