Journal of Engineering Management and Systems Engineering
https://www.acadlore.com/journals/JEMSE

An Investigation into Multi-Stage, Variable-Batch Scheduling Across Multiple Production Units

Yi Du[1*], T. K. Satish Kumar[2], Yongqiang Wang[3], Jialin Wang[4]

[1] School of Electronic Information and Electrical Engineering, Anyang Institute of Technology, 455000 Anyang, China
[2] University of Southern California, 90007 Los Angeles, California, USA
[3] Anyang Branch of China Mobile Tietong Co., Ltd., 455001 Anyang, China
[4] Kharkiv Institute at Hangzhou Normal University, 311121 Hangzhou, China

[*] Correspondence: Yi Du (20170020@ayit.edu.cn)

Received: 11-12-2023 **Revised:** 12-20-2023 **Accepted:** 12-31-2023

Citation: Y. Du, T. K. Satish Kumar, Y. Q. Wang, and J. L. Wang, "An investigation into multi-stage, variable-batch scheduling across multiple production units," *J. Eng. Manag. Syst. Eng.*, vol. 3, no. 1, pp. 1–20, 2024. https://doi.org/10.56578/jemse030101.

Abstract: Against the backdrop of current market demands for a variety of products in small batches, traditional single-variety assembly lines are transitioning to variable production lines to accommodate the manufacturing of multiple similar products. This paper discusses the production unit as a microcosm of the variable production line, which boasts advantages such as smaller line scale, short setup times for changeovers, and ease of product scheduling. A mathematical model for splitting variable production lines into production units is established, with solutions at two levels: resource allocation and product scheduling. The upper-level model focuses on determining the number of production units and the distribution scheme of operators and equipment across multiple channels; the lower-level model addresses the product allocation problem, which is characterized by multiple stages, divisibility, variable batch sizes, and minimum batch size constraints. The solution approaches include a branch and bound method for small-scale problems to obtain optimal solutions, and an improved particle swarm optimization (PSO) algorithm for medium to large-scale problems to find near-optimal solutions. The innovation of the paper lies in the construction of the variable production line splitting model and the optimization algorithms for resource allocation and product scheduling.

Keywords: Variable production lines; Multi-channel; Improved particle swarm optimization (PSO); Branch and bound

1 Introduction

The manufacturing industry is an important pillar of China's national economy, especially in driving economic growth and improving industrial levels, it is playing a key role. The Made in China 2025 strategy clearly states that the manufacturing industry will be the core area to push China towards the transformation into a manufacturing powerhouse. This strategy emphasizes the shift from scale expansion to improvements in quality, efficiency, and innovation-driven change, as well as implementing innovation, green, and intelligent development as the main theme. This not only involves key technologies and product development in informatization, automation, and intelligence but also concerns the fundamental transformation towards green and sustainable development of the manufacturing industry, aimed at enhancing supply quality and market competitiveness.

In this context, this paper focuses on the problem of variable production line splitting in the production of electronic products, a common issue in the manufacturing industry, especially in assembly production lines. Studying this problem has significant theoretical significance and practical application value. From a practical perspective, solving the joint decision-making problem in production line restructuring is crucial for improving the management level of enterprises. An efficient production line can not only shorten the product production cycle but also reduce the number of product delays, thereby enhancing the overall productivity of the enterprise.

Moreover, from a theoretical perspective, this paper proposes a "cellular" construction method for multi-channel production lines and an improved PSO for the scheduling problem of parallel machines. These innovative algorithms provide new references for the field and have proven their effectiveness through a large number of simulation examples. This in-depth study from theory to practice not only promotes the scientific management of production lines but also provides solid technical support for the sustainable development of the manufacturing industry.

1.1 Multi-Stage Assembly Line Reconstruction Methods

To ensure the effective operation of assembly lines in a dynamic and fluctuating market demand environment, many scholars choose to divide a planning period into multiple small production planning periods, each with different varieties of components and demand volumes. They continuously adjust the internal configuration of the assembly production line dynamically according to the component varieties and demand volumes of each production planning period to adapt to the changing market demands, promptly providing customers with high-quality products and services. This method of continuously adjusting the internal configuration of the assembly production line according to dynamic changes in market demand is known as dynamic assembly line reconstruction. Construction methods for multistage assembly line as shown in Table 1.

Table 1. Construction methods for multistage assembly line

Literature	Model Considerations	Dynamic Solving Algorithm
Literature [1]	Processing path and stochastic production demand	Neural-Network-Method
Literature [2]	Variable-demand, but fixed component mix ratio	PSO
Literature [3]	Framework of dynamic cell manufacturing system	Static Solving and Dynamic Recursion
Literature [4]	Minimum cost	Branch and Boun
Literature [5]	Selectable process routes, operation sequences, machine-capabilities, workload, operation-costs, production-setup-costs	Two-Stage-Genetic Algorithm
Literature [6]	Minimum cost	Genetic-Algorithm
Literature [7]	Minimum cost	Hybrid Genetic-Algorithm
Literature [8]	Selectable process routes and nonlinear mixed-integer programming model for equipment processing capabilities	Genetic Algorithm

1.2 Parallel Machine Scheduling Problem

The parallel machine scheduling problem is one of the most typical scheduling problems in the manufacturing process of manufacturing enterprises. This problem involves processing n tasks on m processing machines and determining the processing order of each task to optimize a target indicator. Although this scheduling problem belongs to a category within the study of mixed-flow workshop scheduling problems, its representativeness has led to this type of parallel machine scheduling problem being studied as a distinct category [9]. In these scheduling problems, based on whether the equipment is identical—that is, whether all processing equipment has the same processing capacity for the same task—it can be classified into homogeneous parallel machine scheduling problems and unrelated parallel machine scheduling problems. Furthermore, they can also be divided into single-objective parallel machine scheduling problems and multi-objective parallel machine scheduling problems according to the number of study objectives.

1.2.1 Current state of single-objective parallel machine scheduling research

In previous studies, the majority of scholars have primarily focused on single-objective parallel machine scheduling problems. Wang and Alidaee [10] studied the scheduling problem of large-scale unrelated parallel machines (UPM) with the goal of minimizing the weighted maximum completion time. Pfund et al. [11] aimed to minimize the total weighted tardiness, establishing a parallel machine scheduling mathematical model that considered scheduling job preparation times and processing sequence constraints, and designed a composite scheduling rule grid method called the Apparent Tardiness Cost with Setups (ATCS) approach to solve this scheduling problem. Palacios et al. [12] and Palacios et al. [13], considering the size of processing tasks and their different arrival times, aimed to minimize the maximum completion time, established a corresponding mathematical model, and proposed a batch-then-schedule algorithm to solve the model. Vallada et al. [14], with the goal of minimizing the maximum completion time, developed a model for the parallel machine scheduling problem with resource constraints and proposed a new discrete search algorithm by combining discrete search and iterative greedy algorithms. Fanjul-Peyro et al. [15] targeted the UPM problem with the aim of minimizing the maximum completion time, established a scheduling model with setup and resource constraints, and proposed a three-phase algorithm based on exact mathematical methods to solve the problem.

Yepes-Borrero et al. [16] aimed to minimize the maximum completion time in their study of UPM scheduling problems with setup times and additional limited resources. They proposed three metaheuristic algorithms for comparative solution analysis. Arnaout et al. [17] focused on minimizing the maximum completion time, investigating UPM scheduling problems with equipment-related and sequence-related considerations, and introduced an Ant Colony Optimization (ACO) algorithm to solve the problem. Ozturk et al. [18] aimed to minimize the maximum completion time as their optimization goal. Under the constraint of preventive maintenance for processing equipment, they developed a distributed unrelated parallel machine scheduling model and designed a new type of Artificial Bee Colony (ABC) algorithm to address the issue.

1.2.2 Current state of multi-objective parallel machine scheduling research

As the manufacturing industry continues to evolve, production methods are increasingly diversifying, leading to the need to consider multiple performance indicators in actual production scheduling. Therefore, some scholars have begun to study multi-objective parallel machine scheduling problems. Ho and Tay [19] aimed to minimize both the maximum completion time and the total weighted tardiness by establishing an UPM scheduling model with deterioration and learning effects. They improved the simulated annealing algorithm to obtain better approximate solutions for this problem. Nouiri et al. [20] aimed to minimize the total delay time and reduce the total energy consumption as multi-objectives, establishing a low-carbon parallel machine scheduling model considering the relative importance of objectives. They proposed a new ICA algorithm for solving the model by combining the lexicographic method. Zambrano et al. [21], considering the sequence of task processing and preventive maintenance time, aimed to minimize maintenance costs and shorten the flow time of workpieces as multi-objectives, constructing a corresponding production scheduling model and optimizing it using an improved multi-objective genetic algorithm. Mahmoodjanloo et al. [22], focusing on UPM scheduling, aimed to minimize the total cost of earliness/delays and the maximum completion time as multi-objectives, established a mathematical model, and proposed a multi-objective optimization algorithm to solve the model. Kongsri and Buddhakulsomsiri [3], based on setup times with dependencies, established a mixed integer linear programming model with the objectives of minimizing the maximum completion time and total delay. Afzalirad and Rezaeian [**?**], considering constraints such as sequence-dependent setup times, different arrival times, and equipment qualifications, aimed to minimize the average weighted flow time and the average weighted tardiness time as multi-objectives, and established a mixed integer programming model. Shahidi-Zadeh et al. [23] studied the production scheduling problem of UPM considering task release times, preparation times, and batch capacity constraints, aiming to minimize the maximum completion time, minimize equipment procurement costs, and minimize delay/earliness penalties as multi-objectives, and established a corresponding mathematical model.

According to the research and literature reviewed, the focus is mostly on constructing equipment costs. There are fewer studies that consider operators and equipment as joint optimization targets. However, given the fluctuation in demand, equipment represents a fixed asset, often entailing a one-time investment. In contrast, the investment in operators compared to equipment is much more flexible. This investment fluctuates with demand changes. Therefore, it is crucial to consider human resource investment as a very important optimization objective.

2 Modelling

This paper studies the scheduling of production to meet the demand for i types of products ($i = 1, \dots, I$) over a planning period ($t = 1, \dots, T$). These products, having similar manufacturing processes, pass through several of J workstations (1,2,. . . ,J), where each workstation may contain multiple work positions. The process route for these I types of products, the production time per product at each workstation, and the demand quantities are known. Each workstation has a limit on the number of positions, and there's also a limit on the number of operators on the production line. Due to the high frequency of changeovers required by the variety of products and the large number of operators in a variable production line, preparation times for changeovers are long, leading to delayed costs. Splitting into parallel production lines reduces the scale of production lines, the variety of products, preparation times for changeovers, and hence, delayed costs. The goal is to construct several parallel production lines (1,. . . ,m,. . . ,K), where l(1,2) indicates the type of production line, with 1 representing multi-channel and 2 representing product units, aiming to minimize the number of delays. The constraints include:

Limit on the number of operators.

Limit on the types of workstations and the number of devices at each workstation.

Production time limits for each time period.

Restrictions on the minimum batch size that each product type can be split into.

Delivery time constraints.

Differences in operator skills are not considered.

2.1 Assumptions of the Model

The variable production line is divisible.

The demand for each product varies across different periods but is known.

Equipment and operators are available and in good condition in each period; machine failures and personnel absences are not considered.

After the division of the variable production line, the composition of the production line remains relatively stable (i.e., the physical structure of each production line and the number of operators remain unchanged).

Inventory costs between production lines are not considered.

All operators are skilled in multiple tasks.

2.2 Known Parameters

Q_{it} : The demand quantity of product type i in period t.
B_i : The minimum batch size for product type i.

2.3 Decision Variables

$P_{it}^{k_l}$: The production quantity of product type i in period t on production line k.
$\Delta Q_{iT''}$: The quantity of product type i orders produced ahead of schedule.
$\Delta Q_{iT''}$: The quantity of product type i orders produced behind schedule.
r_{k_l} : The number of operators on production line k.
z_{tk_l} : The product type manufactured on production line k in period t.
Q'_{it} : The adjusted demand quantity for product type i in period t.
s_{k_lj} : The number of devices at workstation j on production line k.
K_l : The number of production lines of type l.

2.4 The Uniform Mathematical Model

The objective function focuses on the quantity of delays, reflecting the production line's capability to meet delivery deadlines. If, for any period and any production line, the adjusted order quantity after scheduling exceeds the actual production quantity, a delay is counted as $Q'_{it} - P_{it}^{k_l}$; otherwise, it is considered as zero. This can be represented as:

$$\sum_{t=1}^{T} \sum_{k_l=1}^{K_l} \sum_{i=1}^{I} \max \left\{0, \left(Q'_{it} - P_{it}^{k_l}\right)\right\} \tag{1}$$

$$\begin{cases} Q'_{it} = Q_{it} \quad \forall i, t \\ Q'_{it} = Q_{it} + \Delta Q_{iT} - \Delta Q_{iT''} \quad T'' > t, T''' < t; \forall i, t \end{cases} \tag{2}$$

$$P_{it}^{k_l} \geq B_i \quad \forall i, k, t, l = 1, 2 \tag{3}$$

$$\Delta Q_{iT''}, \Delta Q_{iT'''} \geq B_i \quad \forall i, T'', T'' < t \tag{4}$$

$$\sum_{it}^{I} P_{it}^{k_l} = f\left(r_{k_l}, z_{tk_l}, s_{kj}\right) \quad \forall k, t, l = 1, 2 \tag{5}$$

$$\Delta Q_{iT''}, B_i, P_{it}^{k_l}, K_l \text{ are non-negative integers.} \tag{6}$$

Constraints (2) to (6) define the limitations within which the model operates: Constraint (2) deals with segment constraints, indicating that for any product i, if the number of orders in stage t equals the actual demand (i.e., there are no orders for producing T'' and T'''), then the Q'_{it} in the objective function is equal to Q_{it}; otherwise it is equal to $Q_{it} + \Delta Q_{iT''} - \Delta Q_{iT'''}$, wherein $\Delta Q_{iT''}$ is the number of product i orders advanced to this stage from the next, and $\Delta Q_{iT'''}$ is the number of orders delayed to this stage from the previous one. If customer does not allow late delivery, $\Delta Q_{iT'''} = 0$; if advance delivery is allowed, then $\Delta Q_{iT''} = 0$. Constraint (3) limits the batch size of the actual product quantity $P_{it}^{k_l}$, for any i, under the condition of $k_l, l = 1, 2$, ensuring the scale is not less than the minimum batch size B_i. Constraint (4) ensures that both $\Delta Q_{iT''}$ and $\Delta Q_{iT'''}$ meet the minimum batch size B_i requirement for product i. Constraint (5) establishes that for any $k_l, l = 1, 2, t$, the actual product quantity is a function of the number of operators on that production line, the number of workstations, and the type of product being produced in that stage. Constraint (6) states that $\Delta Q_{iT''}, B_i, P_{it}^{k_l}$, and K_l are all non-negative integers.

3 Improved PSO

In the improved PSO algorithm, each solution to the optimization problem is visualized as a bird, referred to as a "particle." All particles search within a D-dimensional space. The fitness function determines the suitability of each particle's current position. Each particle is endowed with a memory function to recall the locations it has discovered. Additionally, each particle possesses a velocity determining its flying distance and direction, dynamically adjusted based on its own flying experience and that of its companions.

3.1 Encoding Structure and Generating Initial Solutions

Each particle is represented by a $I \times T$ matrix indicating the production quantity matrix of products over T stages. Rows in this matrix represent product types, and columns represent time periods. Each element in the matrix denotes the quantity of product type i allocated to channel k in stage t. Each column of the encoding structure indicates the types of products to be produced and their quantities in multiple channels k during that period. Every element of the matrix is a non-negative integer. This encoding structure, which matches the order structure, allows for the direct calculation of the delay quantity for the particle.

$$\underbrace{\overbrace{\begin{bmatrix} {m^k}_{11} & {m^k}_{12} & \ldots\ldots & {m^k}_{1T} \\ {m^k}_{21} & {m^k}_{22} & \ldots\ldots & {m^k}_{2T} \\ \ldots\ldots & & & \\ {m^k}_{I1} & {m^k}_{I2} & \ldots\ldots & {m^k}_{IT} \end{bmatrix}}^{I\times T}}_{\text{The encoding of any particle } k\text{, every element within its matrix is a non-negative integer.}} \tag{7}$$

The improved PSO relies heavily on the initial solution; a good initial solution can significantly enhance search efficiency. This is particularly important for multi-stage life cycle orders where delivery delays are not permitted, but early production is allowed. Therefore, the likelihood that randomly generated initial values meet the constraints of no delivery delays and allow for early production is relatively low. A well-constructed initial value lays a solid foundation for subsequent steps. The setting of initial values in the improved PSO, based on known resource configurations (the number of channels, the number of operators per channel, workstation, and equipment allocation), involves producing K matrices of I rows and T columns in a certain proportion. Each element within these product matrices must be a non-negative integer, and the algebraic sum of these K product matrices equals the product order matrix. Each multi-channel product matrix represents a particle.

The selection of initial values is based on the known number of channels, following a specific pattern to produce the product distribution matrices for the channels, where all elements of these matrices are non-negative integers. The sum of all these channel product distribution matrices equals the order matrix. Each product distribution matrix for a channel represents a particle.

The steps for establishing the initial solution of the algorithm are as follows:

Step 1: Calculate the proportion of operators in each channel relative to the total number of operators $c_{1k} = r_k/A, 0 < c_{1k} < 1$.

Step 2: The initial value of the particle swarm is set to be $c_{1k} * Q_{it}$, where c_{1k} are k constants obtained from Step 1, and Q_{it} is the input order matrix.

Step 3: The k matrices satisfy a certain relation.

$$\begin{aligned} &\underbrace{\begin{bmatrix} {m^1}_{11} & {m^1}_{12} & \ldots\ldots & {m^1}_{1T} \\ {m^1}_{21} & {m^1}_{22} & \ldots\ldots & {m^1}_{2T} \\ \ldots\ldots & & & \\ {m^1}_{I1} & {m^1}_{I2} & \ldots\ldots & {m^1}_{IT} \end{bmatrix}}_{\text{Initial value for Multi-Channel 1 (Particle 1)}} + \underbrace{\begin{bmatrix} {m^2}_{11} & {m^2}_{12} & \ldots\ldots & {m^2}_{1T} \\ {m^2}_{21} & {m^2}_{22} & \ldots\ldots & {m^2}_{2T} \\ \ldots\ldots & & & \\ {m^2}_{I1} & {m^2}_{I2} & \ldots\ldots & {m^2}_{IT} \end{bmatrix}}_{\text{Initial value for Multi-Channel 2 (Particle 2)}} + \ldots\ldots \\ &+ \underbrace{\begin{bmatrix} {m^K}_{11} & {m^K}_{12} & \ldots\ldots & {m^K}_{1T} \\ {m^K}_{21} & {m^K}_{22} & \ldots\ldots & {m^K}_{2T} \\ \ldots\ldots & & & \\ {m^K}_{I1} & {m^K}_{I2} & \ldots\ldots & {m^K}_{IT} \end{bmatrix}}_{\text{Initial value for Multi-Channel } K \text{ (Particle } K\text{)}} = \underbrace{\begin{bmatrix} Q_{11} & Q_{12} & \ldots\ldots & Q_{1T} \\ Q_{21} & Q_{22} & \ldots\ldots & Q_{2T} \\ \ldots\ldots & & & \\ Q_{I1} & Q_{I2} & \ldots\ldots & Q_{IT} \end{bmatrix}}_{\text{Orders for } I \text{ types of products over } T \text{ stages}} \end{aligned} \tag{8}$$

Step 4: Knowing the number of operators per channel allows for determining the pace of each product in each channel. The initial values of the improved PSO particles obtained from Step 2 are then inputted into the scheduling mathematical model's formula (5) to calculate the delay quantity (objective function) for each channel, serving as the initial fitness value for each particle.

3.2 Improved PSO with Crossover Operation

By integrating the crossover operation from genetic algorithms with the PSO algorithm, a hybrid approach is created, allowing for crossover among different particles. This enhances the ability of particles to explore new positions in the search space, preventing the swarm from prematurely converging on local optima.

If optimization is only conducted within each multi-channel (particle), it may easily fall into local optima. To broaden the search range, adjustments are made among multiple particles. For example, the m_{12}^1 of the first particle could be adjusted to m_{11}^k in the k-th particle, namely to use $m_{12}^1 + m_{11}^k$ to replace the original m_{11}^k, and the original position of m_{12}^1 is set to 0 . Partial adjustments are also possible; the process involves several steps: Step 1, generate a random number c_2; Step 2, multiply c_2 by an element from the second column onwards of the first particle, such as $c_2 \times m_{12}^1$; Step 3, judge if the result from Step 2 meets the minimum batch requirement, i.e., whether $c_2 \times m_{12}^1$ and $(1 - c_2) \times m_{12}^1$ are not less than the minimum batch size B_1. If it is less than B_1, return to Step 1 and select a new value; otherwise replace the original m_{11}^k with $c_2 \times m_{12}^1 + m_{11}^k$, and the original position of m_{12}^1 is set to $(1 - c_2) \times m_{12}^1$.

$$\overbrace{\underbrace{\begin{bmatrix} m^1{}_{11} & m^1{}_{12} & ...... & m^1{}_{1T} \\ m^1{}_{21} & m^1{}_{22} & ...... & m^1{}_{2T} \\ ...... & & & \\ m^1{}_{I1} & m^1{}_{I2} & ...... & m^1{}_{IT} \end{bmatrix}}}^{I\times T} and \overbrace{\underbrace{\begin{bmatrix} m^k{}_{11} & m^k{}_{12} & ...... & m^k{}_{1T} \\ m^k{}_{21} & m^k{}_{22} & ...... & m^k{}_{2T} \\ ...... & & & \\ m^k{}_{I1} & m^k{}_{I2} & ...... & m^k{}_{IT} \end{bmatrix}}}^{I\times T} \Rightarrow \begin{bmatrix} m^1{}_{11} & 0 & ...... & m^1{}_{1T} \\ m^1{}_{21} & m^1{}_{22} & ...... & m^1{}_{2T} \\ ...... & & & \\ m^1{}_{I1} & m^1{}_{I2} & ...... & m^1{}_{IT} \end{bmatrix}$$
$$and \overbrace{\begin{bmatrix} m^k{}_{11} + m^1{}_{12} & m^k{}_{12} & ...... & m^k{}_{1T} \\ m^k{}_{21} & m^k{}_{22} & ...... & m^k{}_{2T} \\ ...... & & & \\ m^k{}_{I1} & m^k{}_{I2} & ...... & m^k{}_{IT} \end{bmatrix}}^{I\times T} \tag{9}$$

3.3 Improved PSO: Mutation Operation for a Single Particle

Due to the constraint of no delivery delays, when adjusting the initial value of a particle, one should start from the second column of the particle's initial value matrix. Subsequently, elements in the same row of each column can be adjusted, either partially or entirely, to the columns with smaller indices than the current one, following the idea of binary search.

For an entire adjustment, only one step is required, such as replacing the original m_{11}^k with $m_{12}^k + m_{11}^k$, and setting the original position of m_{12}^k to 0 . For a partial adjustment, three steps are needed: Step 1 , generate a random number, $c_1 = 0.5$; Step 2 , multiply c_1 by an element from the second column onwards, such as $c_1 \times m_{12}^k$; Step 3, judge if the result from Step 2 meets the minimum batch requirement, i.e., whether $c_1 \times m_{12}^k$ and $(1 - c_1) \times m_{12}^k$ are not less than the minimum batch size B_1. If it is less than B_1, return to Step 1 and reselect a value; otherwise replace the original m_{11}^k with $c_1 \times m_{12}^k + m_{11}^k$, and the original position of m_{12}^k is set to $(1 - c_1) \times m_{12}^k$.

$$\overbrace{\underbrace{\begin{bmatrix} m^k{}_{11} & m^k{}_{12} & ...... & m^k{}_{1T} \\ m^k{}_{21} & m^k{}_{22} & ...... & m^k{}_{2T} \\ ...... & & & \\ m^k{}_{I1} & m^k{}_{I2} & ...... & m^k{}_{IT} \end{bmatrix}}}^{I\times T} \Rightarrow \overbrace{\begin{bmatrix} m^k{}_{11} + m^k{}_{12} & 0 & ...... & m^k{}_{1T} \\ m^k{}_{21} & m^k{}_{22} & ...... & m^k{}_{2T} \\ ...... & & & \\ m^k{}_{I1} & m^k{}_{I2} & ...... & m^k{}_{IT} \end{bmatrix}}^{I\times T} \tag{10}$$

3.4 Particle Swarm Position Update

Set a velocity v, upon obtaining a feasible solution, i.e., its current fitness value is compared with the fitness value corresponding to its personal best position (pbest). If the current fitness value is higher, the current position is used to update the personal best position (pbest). The pbest values of all particles are aggregated to serve as the global fitness value, which is then compared with the global best (gbest). If the current fitness value is more optimal, the position of the current particle is updated to the global best position (gbest).

The position update of the particle swarm is represented by:

$$X_i^{k+1} = c_2 \otimes f\left\{\left[c_1 \otimes q\left(w \otimes h\left(X_i^k\right), pB_i^k\right), pB_i^k\right], gB^k\right\} \tag{11}$$

where, X_i^k represents the particle's position, w is the inertia weight; c_1 is the cognitive coefficient, and c_2 is the social coefficient; $w, c_1, c_2 \in [0, 1], pB_i^k$ and gB_i^k represent the individual best value and global best value of the k-th generation particles, respectively; h, q, g, and f are operators. The Formula (10) consists of three parts.

The first part is formula (11), with $r \in \left[\begin{matrix} 0 & 1 \end{matrix} \right]$.

$$E_i^k = w \otimes h\left(X_i^k\right) = \begin{cases} h\left(X_i^k\right) & r < w \\ X_i^k & r \geq w \end{cases} \tag{12}$$

See the mutation operation of single particles in the improved PSO.

The second part is Formula (12):

$$F_i^k = c_1 \otimes q\left(E_i^k, pB_i^k\right) = \begin{cases} q\left(E_i^k, pB_i^k\right) & r < c_1 \\ E_i^k & r \geq c_1 \end{cases} \tag{13}$$

See the crossover operation among particles.

$$X_i^k = c_2 \otimes f\left(F_i^k, gB^k\right) = \begin{cases} f\left(F_i^k, gB^k\right) & r < c_2 \\ F_i^k & r \geq c_2 \end{cases} \tag{14}$$

Formula (13) indicates the adjustment of the particle swarm according to the global best position.

The termination condition for the algorithm is reaching a pre-set number of iterations, at which point the algorithm terminates (Figure 1).

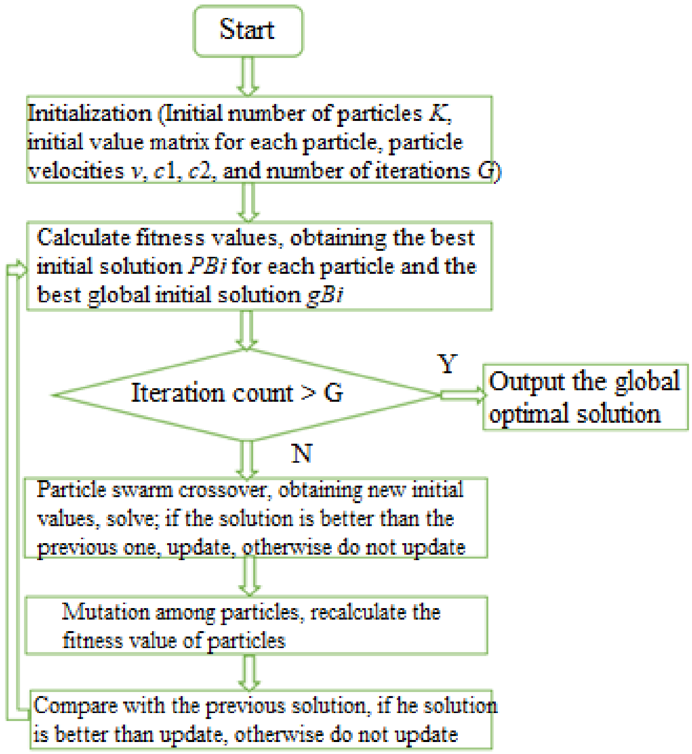


Figure 1. Program flowchart of the improved PSO

Step 1: Input the number of channels K, the order matrix of products, the number of iterations G, the inertia constant w, and the learning factors $c1$ and $c2$.

Step 2: Generate the product distribution matrix for each channel according to the proportion of operators in the channels, with all elements of this matrix being non-negative integers. The sum of all channel product distribution matrices equals the order matrix. Each product distribution matrix for a channel represents a particle. Calculate the local optimal solution pB_i and the global optimal solution gB_i.

Step 3: Update the position of particles based on the mutation operation of a single particle in the improved PSO, and update the local optimal solution pB_i and the global optimal solution gB_i.

Step 4: Further update the position of particles based on the crossover operation among particles, and update the local optimal solution pB_i and the global optimal solution gB_i.

Step 5: Check if the termination condition (e.g., number of iterations $> G$) is met. If satisfied, output the optimal solution; otherwise, go to Step 3 .

4 Example

The specific parameters used in the experiment are shown in Table 2, Table 3, Table 4, Table 5. In Table 2 and Table 3, the first column vertically lists a total of ten (five) product types, P1-P10 (5). Columns 3-12 detail the ten (five) workstations, J1-J10 (5), included in the production line, along with the product types each workstation can process and their production times. For instance, J1 can process the first operation for products P3 and P4, with each product requiring 850s for production on a single machine. The first row horizontally lists the names of the workstations (processes), J1-J10. Rows 2-11 describe the process routes for products P1-P10 and the production time at each individual workstation. For example, product P1 goes through five workstations, J2-J3-J6-J8-J9, with a production time of 860s at a single position at J2. In Table 4 and Table 5, the first column lists ten product types, P1-P10 (5), and columns 3-12 show the ten (five) workstations, J1-J10 (5), included in the production line, along with the number of machines required at each workstation to process the product types. For example, producing P3 and P4 during their first operation at J1 requires four machines each. The first row horizontally lists the workstation (process) names, J1-J10. Rows 2-11 detail the process routes for products P1-P10 (5) and the number of positions required at each workstation. For instance, product P1 needs eight machines at workstation J2 as it passes through five workstations, J2-J3-J6-J8-J9. The twelfth row horizontally displays the maximum number of positions (machines) available at each workstation, like up to 8 positions (machines) at J1. The cycle times for the ten products are respectively 120s, 120s, 120s, 120s, 90s, 90s, 60s, 60s, 60s, 60s. The current number of operators available is 32. The task is to determine the range of multi-channel quantities that can be constructed and the method for allocating operators and equipment.

Table 2. 5 kinds of products process

		U	**J2**	**J3**	**J4**
P1	s_{ij}	0	960s	830s	0
P2	s_{ij}	480s	0	0	450s
P3	s_{ij}	850 s	450 s	480 s	0
P4	s_{ij}	850 s	0	0	960 s
P5	s_{ij}	0	720 s	640 s	0

Table 3. 10 kinds of products process

	J1	**J2**	**J3**	**J4**	**J5**	**J6**	**J7**	**J8**	**J9**	**J10**
P1	0	860 s	830 s	0	0	960 s	0	0	850 s	0
P2	0	0	0	0	0	960 s	1332 s	368 s	850 s	0
P3	850 s	0	0	0	848 s	960 s	0	0	850 s	0
P4	850 s	0	0	0	480 s	1328 s	0	0	850 s	0
P5	0	636 s	640 s	0	650 s	0	0	0	720 s	0
P6	0	636 s	640 s	0	650 s	0	0	0	0	720 s
P7	0	900 s	0	0	0	0	0	0	0	960 s
P8	0	0	428 s	430 s	0	0	0	188 s	0	720 s
P9	0	0	460 s	0	0	0	0	428 s	0	960 s
P10	0	0	920 s	0	0	0	0	0	0	960 s

Table 4. Five kinds of products location distribution

		J1	**J2**	**J3**	**J4**
P1	s_{ij}	0	4	4	0
P2	s_{ij}	4	0	0	4
P3	s_{ij}	4	2	2	0
P4	s_{ij}	4	0	0	4

Table 5. Ten kinds of products location distribution

	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	
P1	0	8	8	0	0	8	0	0	8	0	32
P2	0	0	0	0	0	8	12	4	8	0	32
P3	8	0	0	0	8	8	0	0	8	0	32
P4	8	0	0	0	4	12	0	0	8	0	32
P5	0	8	8	0	8	0	0	0	8	0	32
P6	0	8	8	0	8	0	0	0	0	8	32
P7	0	16	0	0	0	0	0	0	0	16	32
P8	0	0	8	8	0	0	0	4	0	12	32
P9	0	0	8	0	0	0	0	8	0	16	32
P10	0	0	16	0	0	0	0	0	0	16	32
Number of positions at the workstation	8	16	16	8	8	12	12	8	8	16	

The actual quantities of orders for five products across six periods (each period lasting 154,000s), the normalized quantities of the orders over the product lifecycle, and the quantities of non-lifecycle orders are shown in Table 6, Table 7 and Table 8. The actual quantities of orders for ten products across twelve periods (each period lasting 154,000s), the normalized quantities of the orders over the product lifecycle, and the quantities of non-lifecycle orders are displayed in Table 9, Table 10 and Table 11. The lifecycle consideration includes the cost of hiring and departing operators, whereas the non-lifecycle does not consider the cost of acquiring and departing operators.

Table 6. The product orders of five kinds in six phases

	T1	T2	T3	T4	T5	T6
P1	330	350	180	50	230	300
P2	0	30	80	180	100	10
P3	180	160	240	300	160	160
P4	0	0	160	160	160	160
P5	180	180	180	180	180	180

Table 7. The normalized product orders of five kinds in six phases

	T1	T2	T3	T4	T5	T6
P1	330	350	180	50	230	300
P2	0	15	40	90	50	50
P3	180	160	240	300	160	160
P4	0	0	160	160	160	160
P5	135	135	135	135	135	135
Order Number	645	660	755	735	735	805

Table 8. The inanimate cycle product orders of five kinds in six phases

	T1	T2	T3	T4	T5	T6
P1	240	240	240	240	240	240
P2	35	35	35	35	35	35
P3	200	200	200	200	200	200
P4	110	110	110	110	110	110
P5	60	75	170	150	150	220
Order Number	645	660	755	735	735	805

Table 9. The product orders of ten kinds in twelve phases

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
P1	330	350	180	180	180	180	180	180	0	0	0	0
P2	0	180	180	180	180	180	180	180	0	0	0	0
P3	330	160	160	160	160	160	160	160	350	350	350	350
P4	0	0	160	160	160	160	160	160	350	350	350	350
P5	180	180	180	180	180	180	180	180	180	180	180	180
P6	260	260	260	260	260	260	260	260	280	280	280	280
P7	350	220	160	160	160	160	160	160	0	0	0	0
P8	0	120	180	180	180	180	180	180	320	320	340	340
P9	330	320	160	160	160	160	160	160	120	120	0	0
P10	0	0	180	180	180	180	180	180	220	220	340	340

Table 10. The normalized product orders of ten kinds in twelve phases

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
P1	330	350	180	180	180	180	180	180	0	0	0	0
P2	0	180	180	180	180	180	180	180	0	0	0	0
P3	330	160	160	160	160	160	160	160	350	350	350	350
P4	0	0	160	160	160	160	160	160	350	350	350	350
P5	135	135	135	135	135	135	135	135	135	135	135	135
P6	195	195	195	195	195	195	195	195	210	210	210	210
P7	175	110	80	80	80	80	80	80	0	0	0	0
P8	0	60	90	90	90	90	90	90	160	160	170	170
P9	165	160	80	80	80	80	80	80	60	60	0	0
P10	0	0	90	90	90	90	90	90	110	110	170	170
Order Number	1330	1350	1350	1350	1350	1350	1350	1350	1375	1375	1385	1385

Table 11. The Inanimate cycle product orders of ten kinds in twelve phases

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
P1	170	180	180	180	180	180	180	180	180	180	180	180
P2	160	180	180	180	180	180	180	180	180	180	180	180
P3	160	160	160	160	160	160	160	160	70	70	170	170
P4	180	170	160	160	160	160	160	160	350	350	170	170
P5	135	135	135	135	135	135	135	135	55	55	135	135
P6	195	195	195	195	195	195	195	195	210	210	210	210
P7	115	110	80	80	80	80	80	80	80	80	80	80
P8	60	60	90	90	90	90	90	90	80	80	90	90
P9	75	70	80	80	80	80	80	80	60	60	80	80
P10	90	90	90	90	90	90	90	90	110	110	90	90
Order Number	1330	1350	1350	1350	1350	1350	1350	1350	1375	1375	1385	1385

4.1 Example Simulation and Result Analysis

The test dataset for the scheduling algorithm should be generated under the premise of known numbers of product types, multi-channel quantities, operator and equipment allocation results, and a certain total normalized number of products. What can vary is the quantity distribution relationship among different products. Based on Table 7, Table 8, Table 10 and Table 11, using MATLAB, ten 10*12 matrices for lifecycle orders and non-lifecycle orders, and ten 5*6 matrices for lifecycle orders and non-lifecycle orders, were randomly generated. The delay quantity refers to the average value of delays for all matrices that meet the criteria as orders.

4.2 Solution for Traditional Scheduling Rules

From Table 12, Table 13, Table 14 and Table 15, it is observable that for the same five(ten) products and the same order matrix, different scheduling methods result in significantly different quantities of delays.

Table 12. The delay quantity of the five product life cycle orders

Setup Time(s)		EDD+FCFS	EDD+LPT	EDD+SPT	EDD+RW
60	8 operators, 2 multi-channels, each multi-channel staffed with 4 onerators.	746	746	1045	746
120		760	780	1080	750
180		774	774	1087	774
240		788	788	1088	788

Table 13. The delay quantity of the five product inanimate cycle orders

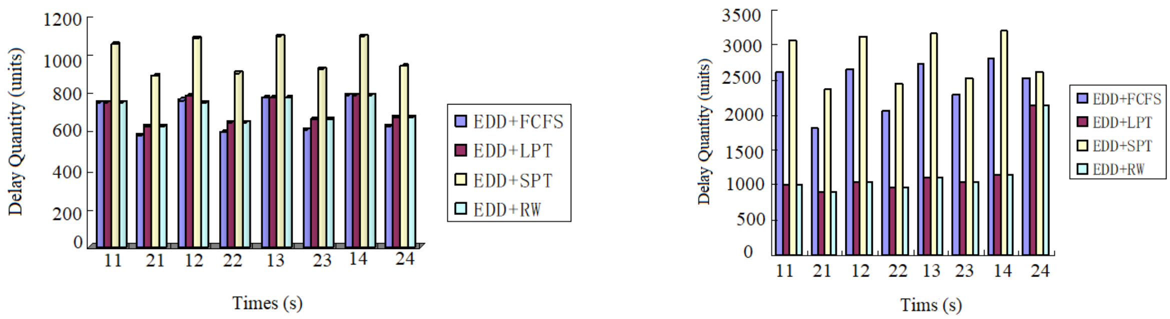
Setup Time(s)		EDD+FCFS	EDD+LPT	EDD+SPT	EDD+RW
60	8 operators, 2 multi-channels, each multi-channel staffed with 4 operators.	578	628	884	628
120		593	643	900	643
180		609	657	916	657
240		625	671	932	671

Table 14. The delay quantity of the ten product life cycle orders

Setup Time(s)		EDD+FCFS	EDD+LPT	EDD+SPT	EDD+RW
60	32 operators, 4 multi-channels, each multi-channel staffed with 8 operators.	1824	900	2375	900
80		2058	978	2453	978
100		2292	1056	2531	1056
120		2526	2134	2609	2134

Table 15. The delay quantity of the ten product inanimate cycle orders

Setup Time(s)		EDD+FCFS	EDD+LPT	EDD+SPT	EDD+RW
60	32 operators, 4 multi-channels, each multi-channel staffed with 8 operators.	2605	1005	3067	1005
80		2648	1053	3115	1053
100		2726	1101	3163	1101
120		2804	1149	3211	1149



(a) (b)

Figure 2. Comparison of four priority scheduling methods for five and ten products

In subgraph (a) of Figure 2, the x-axis labels 11, 21, etc., where the tens digit 1 represents lifecycle orders, and 2 represents non-lifecycle orders. The units digit 1, 2, 3, 4 correspond to individual operator setup times of 60s, 120s, 180s, and 240s, respectively. From subgraph (a) of Figure 2, it is observable that for the same five products and the same order matrix, different scheduling methods result in significantly different quantities of delays. For this type of input, the EDD+SPT (Earliest Due Date + Shortest Processing Time) method results in the highest number of delays, while EDD+LPT (Earliest Due Date + Longest Processing Time) results in the lowest number of delays. Subgraph (b) of Figure 2, concerning ten products, illustrates the same issue. The number of delays is not only related to the scheduling method but also to the setup time of individual operators. The longer the setup time, the greater the number of delays under the same scheduling method.

4.3 Solution with Improved Scheduling Rules

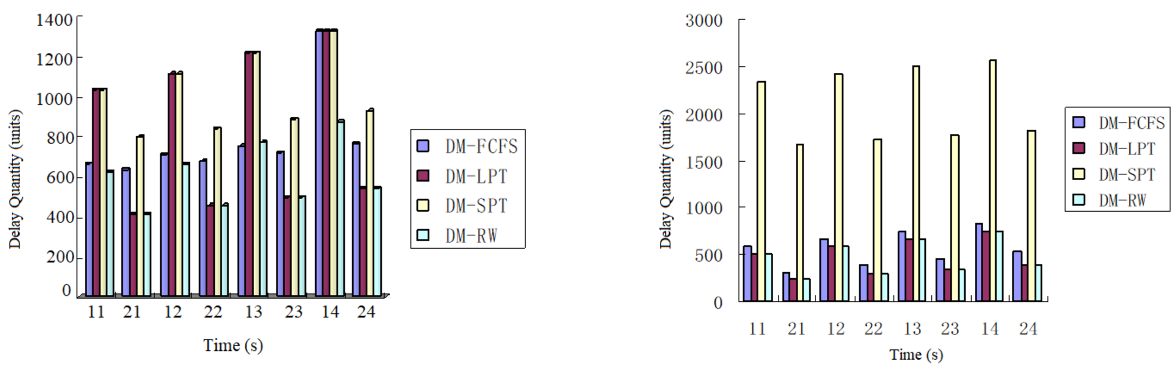
The setup time for changeovers is related to the scale of the production line (the number of operators in multi-channels). Each period T has a production time of 157,200s, with minimum batch sizes of 1, 5, 10, and 15 products. For one changeover, a single operator requires 60s, 120s, 180s, or 240s, making the total changeover time for multi-channel equal to the number of operators in the channel multiplied by 60s (120s, 180s, 300s). For lifecycle orders, delivery delays are permissible. For example, if the production task is not completed in the first period, the remaining orders will occupy the production time of the second period for processing until completion. However, the calculation of delay quantities takes into account that the actual production time for the second period is reduced due to the first period's occupancy, leading to a tendency towards more delays.

From Table 16, Table 17, Table 18, and Table 19, under the same conditions of ten product types, order quantities, scheduling methods, and preparation times, the quantity of delays allowed for batching is smaller than the quantity of delays not allowed for batching. Under the same scheduling methods, the delay quantity for non-lifecycle product orders is smaller than for lifecycle product orders.

From Figure 3, it is observed that the improved priority scheduling algorithm, which schedules multi-period orders with minimum batch size restrictions and allows for early production of orders that can be delayed in batches, is not suitable for lifecycle order types. The scheduling algorithm with the smallest delay quantity for non-cyclical orders is DM-LPT, and the one with the largest delay quantity is DM-SPT. The difference becomes more pronounced as the scale of the production line and the variety of products increase.

Table 16. Average delay quantity per batch for lifecycle orders of five products with permitted delivery delays

Minimum Batch Size (units)		DM-FCFS	DM-LPT	DM-SPT	DM-RW
1	8 operators, 2 multi-channels,	660	1030	1030	620
5	each channel staffed with 4	660	1035	1035	620
10	operators, with a single operator's	660	1030	1035	620
15	setup time being 60 s.	660	1030	1030	620
1	8 operators, 2 multi-channels,	705	1110	1110	660
5	each channel staffed with 4	705	1110	1115	665
10	operators, with a single operator's	705	1130	1110	660
15	setup time being 120 s.	705	1125	1135	660
1	8 operators, 2 multi-channels,	750	1214	1214	765
5	each channel staffed with 4	750	1214	1214	765
10	operators, with a single operator's	750	1229	1229	765
15	setup time being 180 s.	750	1229	1229	770
1	8 operators, 2 multi-channels,	1319	1319	1319	870
5	each channel staffed with 4	1324	1324	1319	870
10	operators, with a single operator's	1334	1334	1324	875
15	setup time being 240 s.	1334	1334	1319	875



(a) (b)

Figure 3. Comparison of four improved scheduling methods for ten products

Table 17. Average delay quantity per batch for non-lifecycle orders of five products with permitted delivery delays

Minimum Batch Size (units)		DM-FCFS	DM-LPT	DM-SPT	DM-RW
0	8 operators, 2 multi-channels,	633	413	796	413
5	each channel staffed with 4	633	413	796	413
10	operators, with a single operator's	633	581	796	581
15	setup time being 60 s.	633	581	796	581
0	8 operators, 2 multi-channels,	675	455	838	455
5	each channel staffed with 4	675	455	838	455
10	operators, with a single operator's	675	612	838	612
15	setup time being 120 s.	675	612	838	612
0	8 operators, 2 multi-channels,	718	497	880	497
5	each channel staffed with 4	718	497	880	497
10	operators, with a single operator's	718	648	880	648
15	setup time being 180 s.	718	648	880	648
0	8 operators, 2 multi-channels,	760	539	922	539
5	each channel staffed with 4	760	539	922	539
10	operators, with a single operator's	760	684	922	684
15	setup time being 240 s.	760	684	922	684

Table 18. Average delay quantity per batch for lifecycle orders of ten products with permitted delivery delays

Minimum Batch Size (units)		DM-FCFS	DM-LPT	DM-SPT	DM-RW
0	32 operators, 4 multi-channels,	587	510	2335	510
20	each channel staffed with 8	1791	510	2335	510
40	operators, with a single operator's	1791	900	2375	900
60	setup time being 60 s.	1824	900	2375	900
0	32 operators, 4 multi-channels,	665	588	2413	588
20	each channel staffed with 8	1869	588	2413	588
40	operators, with a single operator's	1947	978	2453	978
60	setup time being 80s.	2058	978	2453	978
0	32 operators, 4 multi-channels,	743	666	2491	666
20	each channel staffed with 8	1947	666	2491	666
40	operators, with a single operator's	2025	1056	2531	1056
60	setup time being 100 s.	2292	1056	2531	1056
0	32 operators, 4 multi-channels,	821	744	2569	744
20	each channel staffed with 8	2025	2144	2569	2213
40	operators, with a single operator's	2103	2334	2609	2351
60	setup time being 120 s.	2526	2734	2609	2478

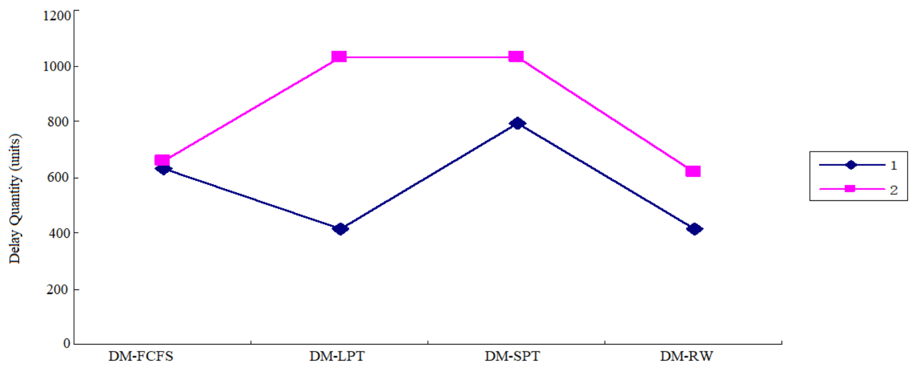
Table 19. Average delay quantity per batch for non-lifecycle orders of ten products with permitted delivery delays

Minimum Batch Size (units)		DM-FCFS	DM-LPT	DM-SPT	DM-RW
0	32 operators, 4 multi-channels, each channel staffed with 8 operators, with a single operator's setup time being 60 s.	303	240	1677	240
20		2245	905	1677	905
40		2425	1005	1987	1005
60		2605	1005	3067	1005
0	32 operators, 4 multi-channels, each channel staffed with 8 operators, with a single operator's setup time being 80 s.	382	288	1725	288
20		2323	953	1725	953
40		2401	1053	2035	1053
60		2648	1053	3115	1053
0	32 operators, 4 multi-channels, each channel staffed with 8 operators, with a single operator's setup time being 100 s.	460	336	1773	336
20		2401	1001	1773	1001
60		2726	1101	3163	1101
0	32 operators, 4 multi-channels, each channel staffed with 8 operators, with a single operator's setup time being 120 s.	538	384	1821	384
20		2479	1049	1821	1049
40		2557	1149	2131	1149
60		2804	1149	3211	1149

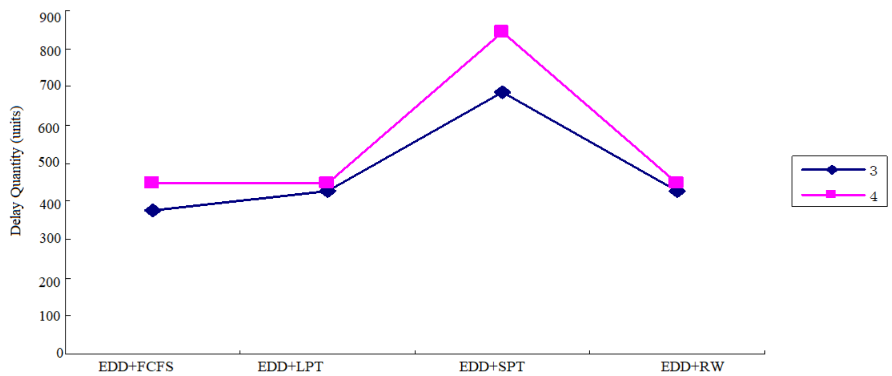
Table 20. Average delay quantity per batch for PSO without allowed delivery delays

		Lifecycle Orders		Non-Lifecycle Orders	
Minimum Batch Size (units)		**Allowed Delivery Delays**	**Not Allowed Delivery Delays**	**Allowed Delivery Delays**	**Not Allowed Delivery Delays**
0	32 operators, 4 multi-channels,	0	0	0	0
20	each channel staffed with 8	0	0	0	0
40	operators, with a single	0	0	0	0
60	operator's setup time being 10 s.	0	0	0	0
0	32 operators, 4 multi-channels,	128	0	365	331
20	each channel staffed with 8	144	0	380	342
40	operators, with a single	158	0	390	349
60	operator's setup time being 80 s.	162	0	410	354
0	32 operators, 4 multi-channels,	603	0	413	360
20	each channel staffed with 8	625	0	423	365
40	operators, with a single	649	0	441	368
60	operator's setup time being 100 s.	701	0	457	375
0	32 operators, 4 multi-channels,	814	60	424	400
20	each channel staffed with 8	867	72	1005	465
40	operators, with a single	889	83	1099	500
60	operator's setup time being 120 s	932	95	1112	792

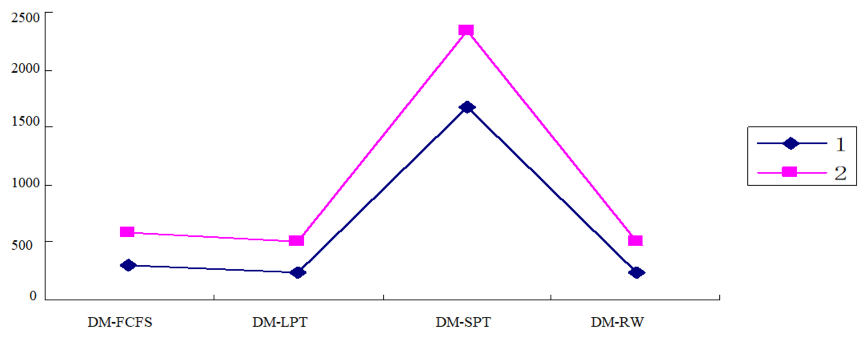
It can be seen from Table 20 that the delay quantities for non-lifecycle orders are smaller than those for lifecycle orders when using rule-based scheduling methods.



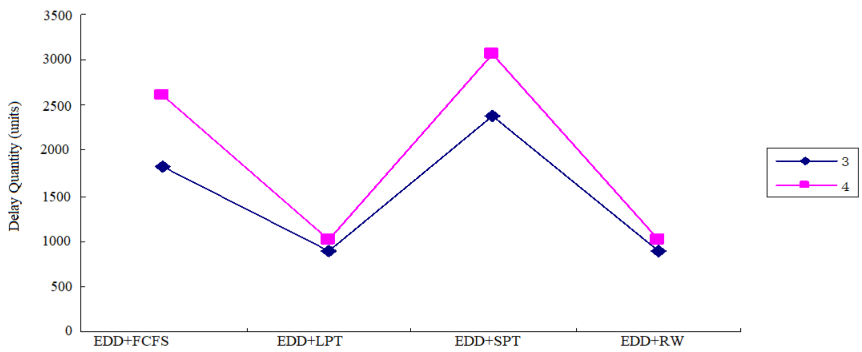
(a)



(b)



(c)



(d)

Figure 4. Comparison of delay quantities for different types of orders under heuristic scheduling algorithms

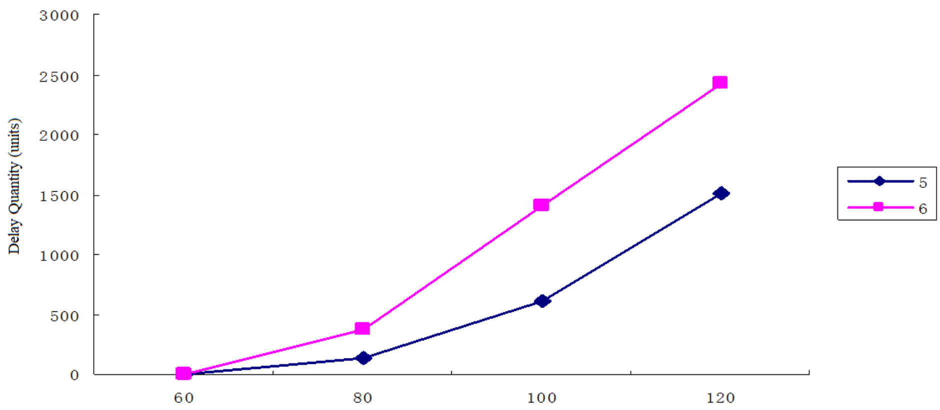


Figure 5. Comparison of delay quantities for ten products using the improved PSO

In Figure 4, the lower curve represents the delay quantities allowed for non-lifecycle orders with a minimum batch size of 0 under four heuristic scheduling methods. The upper curve represents the delay quantities allowed for lifecycle orders with a minimum batch size of 0 under four improved scheduling methods. It can be seen from Figure 4 that the delay quantities for non-lifecycle orders are smaller than those for lifecycle orders when using rule-based scheduling methods.

In Figure 5, the lower curve represents the delay quantities for lifecycle orders with a minimum batch size of 0 allowed delivery delays under the improved PSO. The upper curve shows the delay quantities for non-lifecycle orders with a minimum batch size of 0 allowed delivery delays under the improved PSO. The x-axis represents the delay quantities for an operator's preparation times of 60s, 80s, 100s, and 120s. It can be seen from Figure 5 that the delay quantities for non-lifecycle orders are greater than those for lifecycle orders under the improved PSO.

According to the results in Table 21, subgraphs (a) and (b) of Figure 6 correspond to lifecycle orders with ten products under conditions of allowed and not allowed delivery delays, respectively, showing the objective function values under different scheduling methods. The improved PSO results in the smallest delay quantities, followed by rule-based scheduling method 2. Subgraphs (c) and (d) of Figure 6 correspond to conditions of allowed and not allowed delivery delays for five products, respectively, with the smallest delay quantities also achieved by the improved PSO. The advantage of the improved PSO is more apparent for ten products compared to five products. Allowing delivery delays is more sensitive to the improved PSO than not allowing them.

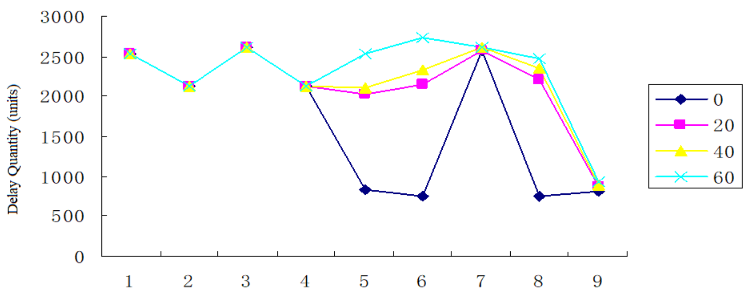
According to the results in Table 22, subgraphs (a), (b), (c), and (d) of Figure 7 correspond to non-lifecycle orders for ten and five products under conditions of allowed and not allowed delivery delays, respectively, showing the objective function values under different scheduling methods. Unlike lifecycle orders, the smallest objective function value occurs with scheduling method 2.

Table 21. Comparison of delay quantities among different algorithms

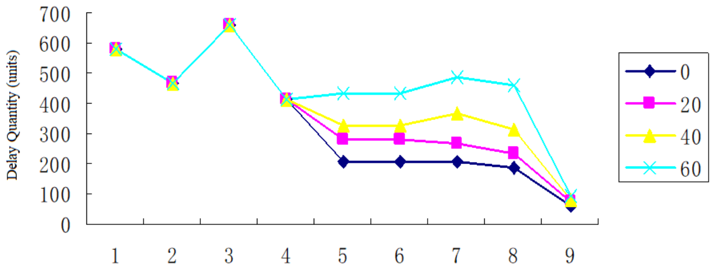
	Lifecycle Orders, Setup Time of 120s															
	Ten Products								Five Products							
	Allowed Delivery Delays				Not Allowed Delivery Delays				Allowed Delivery Delays				Not Allowed Delivery Delays			
	0	20	40	60	0	20	40	60	0	5	10	15	0	5	10	15
EDD+FCFS	2526	2526	2526	2526	578	578	578	578	760	760	760	760	592	592	592	592
EDD+LPT	2134	2134	2134	2134	467	467	467	467	780	780	780	780	574	574	574	574
EDD+SPT	2609	2609	2609	2609	662	662	662	662	1080	1080	1080	1080	608	608	608	608
EDD+RW	2134	2135	2136	2137	412	412	412	412	750	750	750	750	569	569	569	569
DM+FCFS	821	2025	2103	2526	204	280	326	432	705	705	705	705	521	521	524	529
DM + LPT	744	2144	2334	2734	204	280	326	432	1110	1110	1130	1125	513	521	531	501
DM + SPT	2569	2569	2609	2609	204	265	368	488	1110	1130	1110	1135	520	520	523	520
DM + RW	744	2213	2351	2478	185	235	316	461	660	660	665	660	520	520	523	520
Improved PSO	814	867	889	932	60	72	83	95	583	606	613	645	544	534	543	540

Table 22. Comparison of delay quantities for non-lifecycle orders among different algorithms

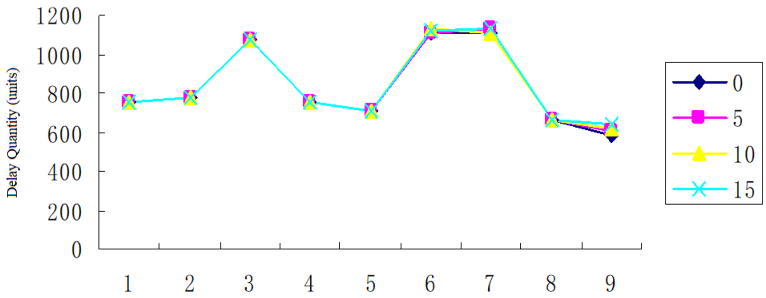
	Non-Lifecycle Orders, Setup Time of 120s															
	Ten Products								Five Products							
	Allowed Delivery Delays				Not Allowed Delivery Delays				Allowed Delivery Delays				Not Allowed Delivery Delays			
	0	20	40	60	0	20	40	60	0	5	10	15	0	5	10	15
EDD + FCFS	2804	2804	2804	2804	598	598	598	598	593	593	593	593	592	592	592	592
EDD + LPT	1149	1149	1149	1149	546	546	546	546	643	643	643	643	74	574	74	574
EDD + SPT	3211	3211	3211	3211	646	646	646	646	900	900	900	900	608	608	608	608
EDD + RW	1149	1150	1151	1152	560	560	560	560	643	643	643	3	569	569	569	569
DM + FCFS	538	2479	2557	2804	428	484	553	795	675	675	675	675	546	546	546	546
DM + LPT	384	1049	1149	1149	428	484	653	895	455	455	612	612	520	520	523	520
DM + SPT	1821	1821	2231	3221	398	404	522	630	838	838	838	838	550	550	553	550
DM + RW	384	1049	1149	1149	360	435	543	628	455	455	612	612	520	520	523	520
Improved PSO	424	1005	1099	1112	400	465	550	792	684	680	673	682	540	545	548	549



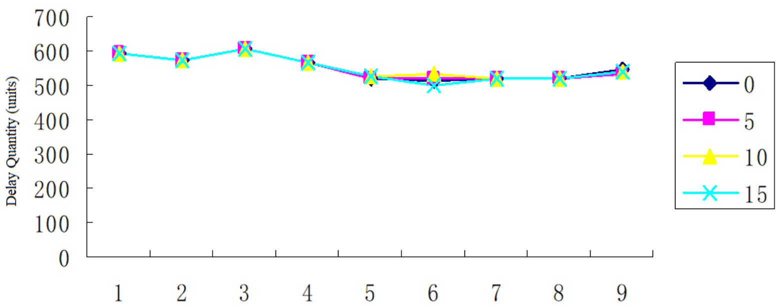
(a)



(b)



(c)

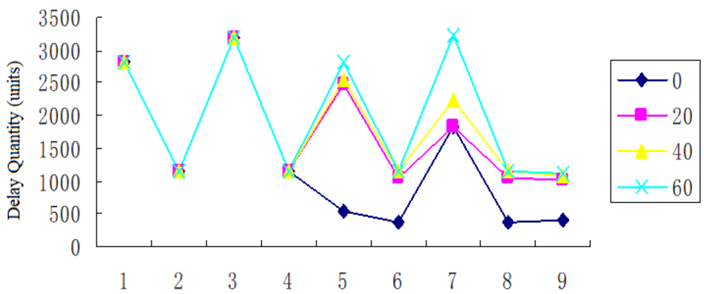


(d)

Figure 6. Comparison of delay quantities for lifecycle orders among different algorithms

To analyze the stability of the improved PSO, it was run 10 times, with an average delay quantity of 463 units, and the fluctuation is shown in Figure 8. During the 10 runs, the fluctuation of the objective function was within a certain range, indicating that the improved PSO has good stability.

Regarding the convergence of the algorithm, Figure 9 shows the convergence process of the average delay quantity of the objective function as the number of iterations increases in the improved PSO. The results indicate that as the number of iterations increases, the delay quantity shows a decreasing trend and tends towards a stable value after more than 300 iterations. This demonstrates that the algorithm has good convergence.



(a)

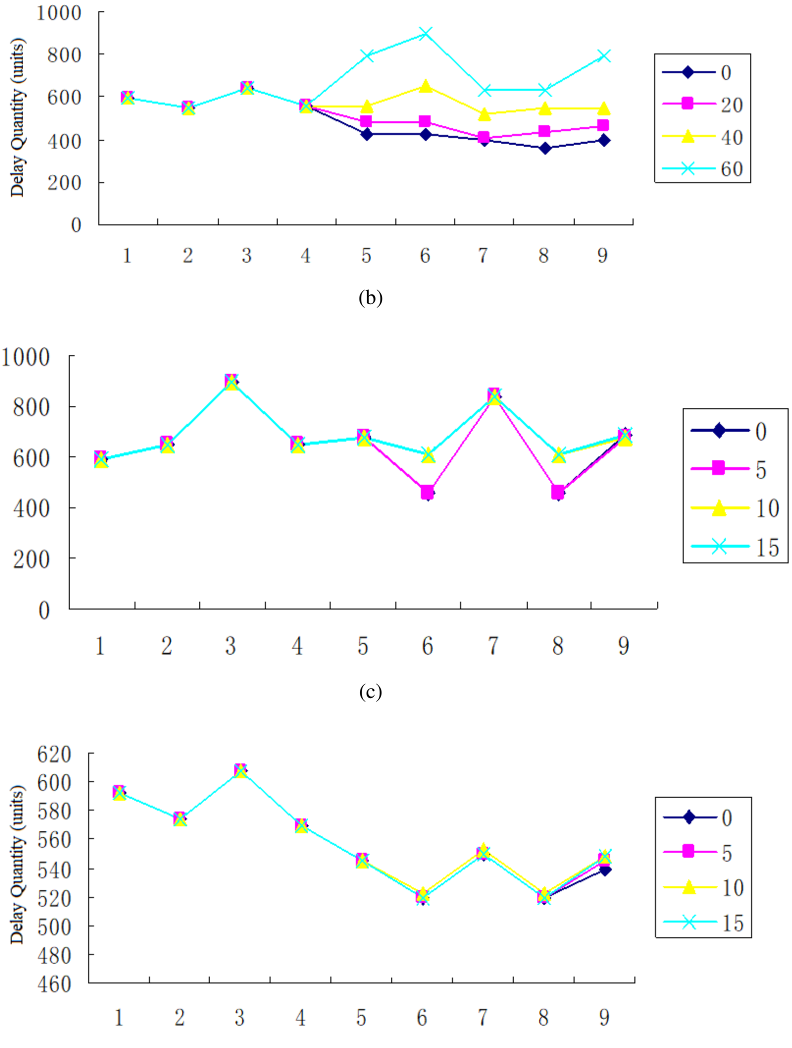


Figure 7. Comparison of delay quantities for non-lifecycle orders among different algorithms

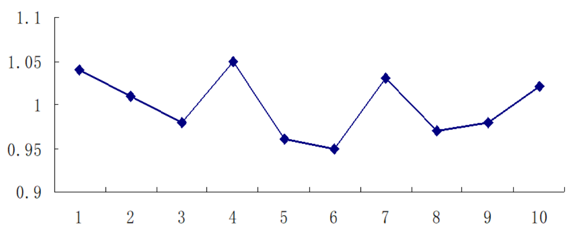


Figure 8. Fluctuation of objective function values over 10 runs of the algorithm

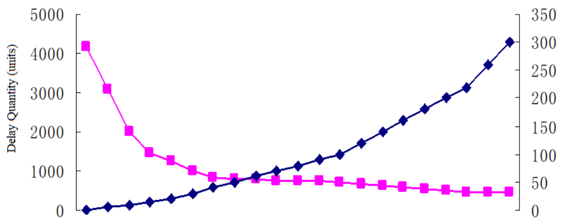


Figure 9. Change in objective function values over 300 iterations of the algorithm

5 Conclusions

The problem of splitting variable production lines is prevalent in electronic product assembly enterprises. An effective variable production line splitting scheme can not only ensure the effective utilization of equipment resources, especially human resources, shorten the production cycle of products but also enhance the flexibility and robustness of the production line. In this work, the solution approach for the variable production line reconstruction model was analyzed. To find a unified model solution, the unified model was first solved in two steps and then jointly. The first step obtains all possible resource allocation results that could lead to the optimal solution, including the number of production lines and the resource allocation plan, using the "cell" multi-channel approach to exclude those multi-channel resource allocation methods that are unlikely to achieve the optimal solution. The second step product scheduling uses traditional optimization algorithms such as EDD+FCFS, EDD+LPT, EDD+SPT, and EDD+RW, four improved optimization algorithms DM-FCFS, DM-LPT, DM-SPT, and DM-RW, and the improved PSO to solve the objective function of lifecycle and non-lifecycle orders under the known multi-channel production line resource allocation. Through numerous examples, it is proven that the algorithms proposed in this paper are superior to those in the literature in terms of production time and objective function values.

Researching the problem of splitting variable production lines plays an important role in reducing production costs, shortening production cycles, and enhancing the competitiveness of electronic product manufacturing enterprises in the market. There is still much research content and technical means involved. With the deepening of research, the next steps of research work can be approached from the following aspects:

(1) Given the complexity of the actual production process, the objective function of the mathematical model for the splitting problem should not be singular; hence, how to convert multiple objectives into a single objective, and how to construct effective algorithms for the multi-objective electronic production enterprise assembly line splitting problem based on the dimensionality reduction algorithm proposed in this article, require further study.

(2) In actual manufacturing systems, operators are graded; thus, the next research focus of this article can be placed on the problem of splitting variable production lines where operators have different capabilities.

Data Availability

The data used to support the research findings are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] F. Zhao, X. Hu, L. Wang, T. Xu, N. Zhu, and Jonrinaldi, "A brain storm optimization algorithm with feature information knowledge and learning mechanism," *Appl. Intell.*, vol. 53, no. 5, pp. 5199–5223, 2023. https://doi.org/10.1007/s10489-022-03762-3

[2] L. Dan, S. Thevenin, and A. Dolgui, "A state-of-the-art on production planning in industry 4.0." *Int. J. Prod. Res.*, vol. 61, no. 7, pp. 1–31, 2022. http://doi.org/10.1080/00207543.2022.2122622

[3] P. Kongsri and J. Buddhakulsomsiri, "A mixed integer programming model for unrelated parallel machine scheduling problem with sequence dependent setup time to minimize makespan and total tardiness," in *2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA), Bangkok, Thailand*, 2020, pp. 605–609. https://doi.org/10.1109/ICIEA49774.2020.9102086

[4] J. C. Yepes-Borrero, F. Perea, R. Ruiz, and F. Villa, "Bi-objective parallel machine scheduling with additional resources during setups," *Eur. J. Oper. Res.*, vol. 292, no. 2, pp. 443–455, 2021. https://doi.org/10.1016/j.ejor.2020.10.052

[5] P. Fattahi, N. Bagheri Rad, F. Daneshamooz, and S. Ahmadi, "A new hybrid particle swarm optimization and parallel variable neighborhood search algorithm for flexible job shop scheduling with assembly process," *Assem. Autom.*, vol. 40, no. 3, pp. 419–432, 2020. https://doi.org/10.1108/AA-11-2018-0178

[6] J. Li, X. Gu, Y. Zhang, and X. Zhou, "Distributed flexible job-shop scheduling problem based on hybrid chemical reaction optimization algorithm," *Complex Syst. Model. Simul.*, vol. 2, no. 2, pp. 156–173, 2022. https://doi.org/10.23919/CSMS.2022.0010

[7] T. S. Lee and Y. T. Loong, "A review of scheduling problem and resolution methods in flexible flow shop," *Int. J. Ind. Eng. Comput.*, vol. 10, no. 1, pp. 67–88, 2019. http://doi.org/10.5267/j.ijiec.2018.4.001

[8] X. Wu, C. H. Chu, Y. Wang, and D. Yue, "Genetic algorithms for integrating cell formation with machine layout and scheduling," *Comput. Ind. Eng.*, vol. 53, no. 2, pp. 277–289, 2007. https://doi.org/10.1016/j.cie.2007.06.021

[9] K. Gao, P. N. Suganthan, Q.-K. Pan, T. J. Chua, C. S. Chong, and T. Cai, "An improved artificial bee colony algorithm for flexible job-shop scheduling problem with fuzzy processing time," *Expert Syst. Appl.*, vol. 65, pp. 52–67, 2016. https://doi.org/10.1016/j.eswa.2016.07.046

[10] H. Wang and B. Alidaee, "Effective heuristic for large-scale unrelated parallel machines scheduling problems," *Omega*, vol. 83, pp. 261–274, 2019. https://doi.org/10.1016/j.omega.2018.07.005

[11] M. Pfund, J. W. Fowler, A. Gadkari, and Y. Chen, "Scheduling jobs on parallel machines with setup times and ready times," *Comput. Ind. Eng.*, vol. 54, no. 4, pp. 764–782, 2008. https://doi.org/10.1016/j.cie.2007.08.011

[12] J. J. Palacios, M. A. González, C. R. Vela, I. González-Rodríguez, and J. Puente, "Genetic tabu search for the fuzzy flexible job shop problem," *Comput. Oper. Res.*, vol. 54, pp. 74–89, 2015. https://doi.org/10.1016/j.cor.2014.08.023

[13] J. J. Palacios, I. Gonzalez-Rodriguez, C. R. Vela, and J. Puente, "Coevolutionary makespan optimisation through different ranking methods for the fuzzy flexible job shop," *Fuzzy Sets Syst.*, vol. 278, pp. 81–97, 2015. https://doi.org/10.1016/j.fss.2014.12.003

[14] E. Vallada, F. Villa, and L. Fanjul-Peyro, "Enriched metaheuristics for the resource constrained unrelated parallel machine scheduling problem," *Comput. Oper. Res.*, vol. 111, pp. 415–424, 2019. https://doi.org/10.1016/j.cor.2019.07.016

[15] L. Fanjul-Peyro, "Models and an exact method for the unrelated parallel machine scheduling problem with setups and resources," *Expert Syst. Appl. X*, vol. 5, p. 100022, 2020. https://doi.org/10.1016/j.eswax.2020.100022

[16] J. C. Yepes-Borrero, F. Villa, F. Perea, and J. P. Caballero-Villalobos, "GRASP algorithm for the unrelated parallel machine scheduling problem with setup times and additional resources," *Expert Syst. Appl.*, vol. 141, p. 112959, 2020. https://doi.org/10.1016/j.eswa.2019.112959

[17] J. P. Arnaout, G. Rabadi, and R. Musa, "A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times," *J. Intell. Manuf.*, vol. 21, pp. 693–701, 2010. https://doi.org/10.1007/s10845-009-0246-1

[18] G. Ozturk, O. Bahadir, and A. Teymourifar, "Extracting priority rules for dynamic multi-objective flexible job shop scheduling problems using gene expression programming," *Int. J. Prod. Res.*, vol. 57, no. 10, pp. 3121–3137, 2019. https://doi.org/10.1080/00207543.2018.1543964

[19] N. B. Ho and J. C. Tay, "Solving multiple-objective flexible job shop problems by evolution and local search," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 38, no. 5, pp. 674–685, 2008. https://doi.org/10.1109/TSMCC.2008.923888

[20] M. Nouiri, A. Bekrar, A. Jemai, D. Trentesaux, A. C. Ammari, and S. Niar, "Two stage particle swarm optimization to solve the flexible job shop predictive scheduling problem considering possible machine breakdowns," *Comput. Ind. Eng.*, vol. 112, pp. 595–606, 2017. https://doi.org/10.1016/j.cie.2017.03.006

[21] G. Z. Rey, A. Bekrar, V. Prabhu, and D. Trentesaux, "Coupling a genetic algorithm with the distributed arrival-time control for the JIT dynamic scheduling of flexible job-shops," *Int. J. Prod. Res.*, vol. 52, no. 12, pp. 3688–3709, 2014. https://doi.org/10.1080/00207543.2014.881575

[22] M. Mahmoodjanloo, R. Tavakkoli-Moghaddam, A. Baboli, and A. Bozorgi-Amiri, "Flexible job shop scheduling problem with reconfigurable machine tools: An improved differential evolution algorithm," *Appl. Soft Comput.*, vol. 94, p. 106416, 2020. https://doi.org/10.1016/j.asoc.2020.106416

[23] M. Afzalirad and J. Rezaeian, "A realistic variant of bi-objective unrelated parallel machine scheduling problem: NSGA-II and MOACO approaches," *Appl. Soft Comput.*, vol. 50, pp. 109–123, 2017. https://doi.org/10.1016/j.asoc.2016.10.039

[24] B. Shahidi-Zadeh, R. Tavakkoli-Moghaddam, A. Taheri-Moghadam, and I. Rastgar, "Solving a bi-objective unrelated parallel batch processing machines scheduling problem: A comparison study," *Comput. Oper. Res.*, vol. 88, pp. 71–90, 2017. https://doi.org/10.1016/j.cor.2017.06.019