

Article

Integrated Scheduling Algorithm of Two Workshops Based on Process End Time Driven and Processing Area Priority

Xiaojuan Zhan ^{1,2}, Zhiqiang Xie ^{1,*} and Dengju Yao ¹ 

¹ School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China

² College of Computer Science and Technology, Heilongjiang Institute of Technology, Harbin 150050, China

* Correspondence: xiezhiqiang@hrbust.edu.cn; Tel.: +86-186-8685-1167

Abstract: In order to effectively allocate processes of a single complex product to two workshops with the same equipment, an integrated scheduling algorithm of two workshops based on process end time driven and processing area priority is proposed. To optimize the total processing time, a process scheduling strategy based on process end time driven is proposed. By scheduling multiple devices at the same time, the parallelism of process execution is improved, in order to reduce the total processing time and improve the utilization rate of equipment. In order to optimize the process migration times between two workshops, a workshop assignment strategy based on process processing area priority is proposed, which comprehensively considers the influence of the immediate preceding process, neighbor process and friend process of the schedulable process on workshop assignment and makes the process in the same processing area be processed in the same workshop as much as possible, so as to reduce the process migration times. The example analysis shows that the proposed algorithm is superior to the existing two workshops integrated scheduling algorithms in terms of reducing the total processing time, reducing the process migration times, and improving the utilization rate of equipment.



Citation: Zhan, X.; Xie, Z.; Yao, D. Integrated Scheduling Algorithm of Two Workshops Based on Process End Time Driven and Processing Area Priority. *Electronics* **2022**, *11*, 2594. <https://doi.org/10.3390/electronics11162594>

Academic Editor: Dah-Jye Lee

Received: 26 July 2022

Accepted: 17 August 2022

Published: 18 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: integrated scheduling algorithm; tree-structured; machining and assembling; two workshops; process end time driven; processing area priority; process sorting; single product

1. Introduction

The world has entered the stage of “Industry 4.0,” and the traditional product manufacturing mode has undergone revolutionary changes. On the one hand, with the development of information technology, intelligent manufacturing has been widely used in the manufacturing process of various products. On the other hand, with the change of social demand, personalized and small batch complex products are gradually becoming an important part of the manufacturing industry. In order to cope with the new industrial production situation, the major countries in the world have put forward their own coping strategies. As the largest developing country in the world, China has also put forward the development strategy of “Made in China 2025”.

As an important part of the “Made in China 2025” development strategy, intelligent manufacturing has received extensive attention in all fields of manufacturing. In order to improve the production efficiency of manufacturing enterprises and reduce production costs, the production scheduling problem of products, especially small batch complex products, has become a research hotspot in academic and industrial circles. As one of the important research directions in product production scheduling, the job shop scheduling problem (JSSP) is a kind of combinatorial optimization problem that has been widely studied and is also a typical NP-hard problem [1]. For decades, a large number of job shop scheduling problem optimization algorithms have been proposed. For the two-stage no-wait job shop scheduling problem, Hadi Mokhtari proposed a new optimization method by

combining an enhanced variable neighborhood search and an artificial neural network [2]. For the flexible job shop scheduling problem, Zhao proposed a hybrid algorithm mixed with bi-level neighborhood search and genetic algorithms to optimize the maximum completion time [3]. For the multi-objective flexible job shop scheduling problem with machine capacity constraints, Mohamad Rohanineja et al., proposed a new nonlinear integer programming model to formulate the problem and designed a hybrid meta-heuristic algorithm to overcome its complexity. For assigning and sequencing processes, they proposed a multi-objective genetic algorithm and designed a heuristic approach to tradeoff the objective functions [4]. To reduce the total energy consumption in the manufacturing workshop with degradation effects and imperfect maintenance, An et al. proposed a novel integrated optimization model including flexible job-shop scheduling, forklift transportation and imperfect cutting tool maintenance [5]. Although the relevant research has achieved fruitful results, there is still a room for optimizing the results of the job shop scheduling.

With the increasing demand for personalized products, there are more and more orders for single complex products. For a complex product with a tree process structure, if the processing and assembly are scheduled by traditional job shop scheduling algorithms, it will split the internal parallel relationship between processing and assembly and result in a longer production cycle. Therefore, Xie et al. proposed the concept of integrated scheduling which deals with the processing and assembly of a complex product at the same time [6]. In order to solve the problem that the integrated scheduling algorithm based on the allied critical path method (ACPM) may cause the machine idle time to be too long, Xie et al. proposed an integrated scheduling algorithm driven by machine idle events, which regards the end of each process as a machine idle event that drives the idle machine to search for schedulable processes [6]. Aiming at the problem that the compactness of serial processes was neglected, Xie et al. proposed a time-selective integrated scheduling algorithm which used the process sequence sorting strategy to sort processes and employed the two workshops time-selective strategy to search for the start time and processing workshop for processes [7]. Recently, Guo et al. proposed an integrated scheduling algorithm with a no-wait constraint based on virtual components [8]. Wang et al. proposed an integrated scheduling algorithm for multi-device-processes that adopted the strategy of exchanging adjacent parallel processes of the same device [9].

As mentioned above, there have been dozens of single workshop integrated scheduling algorithms, but there are few studies on two workshops integrated scheduling problem. Because the scheduling objective of two workshops integrated scheduling problem is not only to minimize the product completion time but also to consider the additional time cost caused by the movement of processes between different processing workshops and the balanced utilization of multiple processing machines. In order to minimize the time cost and realize the balanced utilization of processing machines in two workshops, this paper proposes an integrated scheduling algorithm of two workshops based on process end time driven and processing area priority (PETD-PAP-TWISA). Firstly, a process scheduling strategy based on the process end time driven was proposed to optimize the total processing time. By scheduling multiple machines at the same time, the parallelism of process execution was improved. Secondly, a workshop assignment strategy based on process processing area priority was proposed to optimize the process migration times between two workshops. The proposed workshop assignment strategy comprehensively considered the influence of the immediate preceding process, neighbor process, and friend process of the schedulable process on workshop assignment and made the processes in the same processing area be processed in the same workshop as much as possible, so as to reduce the process migration times. The example analysis results show that the performance of the PETD-PAP-TWISA is better than that of the four existing two workshops integrated scheduling algorithms in terms of the total processing time, the process migration time, and the utilization rate of equipment.

The following is a brief introduction to the structure of this paper. In the second section, we will introduce the related works of two workshops on integrated scheduling

problem and the research ideas of this paper. In the third section, we will present the research objectives of two workshops integrated scheduling problem, the definition of related concepts and the detailed design of the two workshops integrated scheduling algorithm proposed in this paper. The fourth part of the paper will introduce the specific process of product scheduling of the PETD-PAP-TWISA via an example and analyze and compare the performance of the PETD-PAP-TWISA with the existing algorithms. In the fifth part, we will discuss the characteristics of the proposed algorithm. Finally, we will briefly summarize the whole work.

2. Related Works

This paper mainly studies the integrated scheduling algorithm of two workshops. Recently, with the increase of product complexity and the change of production mode, the research on integrated scheduling of a single complex product is developing from single workshop scheduling to two or multi-workshops scheduling. Compared with the single workshop scheduling, the two workshop scheduling problem is more complex. In addition, to minimize the total processing time pursued by the single workshop scheduling, the two workshop scheduling also needs to consider the balanced utilization of the same equipment resources and the process migration time between the two workshops. Xie et al. first proposed the integrated scheduling problem of two workshops and then designed an integrated scheduling algorithm of two workshops based on the allied critical path method (ACPM-TWISA) [10]. In the ACPM-TWISA, processes are first sorted based on the quasi-critical path method, and then the pre-scheduling strategy of two workshops with close processing end times are used for scheduling. However, the ACPM-TWISA pays too much attention to the serial processing of the process tree and neglects the parallelism between processes. Later, Xie et al. proposed an integrated scheduling algorithm of two workshops based on schedulable process balancing (BP-TWISA) [11]. In the BP-TWISA, the schedulable process workshop balancing strategy is used to group the processes, and the processes are assigned and scheduled according to the grouped process workshop determine strategy. The BP-TWISA makes the product completion time shorter and reduces the number of process movements between two workshops. However, using the grouped process workshop to determine the strategy to process processes in batches will increase the migration times between leaf nodes and their subsequent processes. At the same time, it gives priority to scheduling leaf nodes, pays too much attention to the parallelism between processes, and ignores the vertical scheduling of process trees. In 2016, Xie et al. proposed an integrated scheduling algorithm for two workshops based on neighborhood rendering (NR-TWISA) [12]. In the NR-TWISA, a key equipment balancing strategy is proposed to allocate all the processes on the key equipment to the two workshops according to the longest parallel processing time in advance. A neighborhood rendering strategy and a workshop selection strategy for the same equipment processes are proposed. By calculating the rendering factor and influence factor of the neighboring processes of the unassigned workshop processes, the processing workshop of the process is determined, and the dynamic critical path strategy and short time strategy are used to determine the process scheduling order. The NR-TWISA algorithm has shorter product processing time and fewer process migration times. However, it does not fully consider the impact of the scheduling of the first process on the subsequent processes, resulting in poor parallelism between processes. In addition, it still cannot solve the problem of unnecessary process migration and processing equipment preemption caused by the balance strategy of key equipment.

Taking into account the above introduction and analysis, the existing two workshops' integrated scheduling algorithms based on the allied critical path method or process sequence sorting method ignore the parallelism between processes, while the algorithms based on process grouping overemphasize the parallelism between processes and ignore their seriality. All these factors affect the total processing time of products. In addition, the existing two workshops integrated scheduling algorithms pay too much attention to the balance between two workshops, so it will increase the process migration times and

affect the final scheduling results. To solve the above problems, this paper proposed an integrated scheduling algorithm of two workshops based on process end time driven and processing area priority. By setting an independent candidate schedulable process queue for each kind of equipment, dynamically determining the schedulable process and scheduling processes based on long path and long-time strategy, the parallelism of process execution was improved, and the total processing time was reduced. Moreover, the concept of process processing area was proposed. Based on the principle of processing area priority, workshops and equipment were allocated to the scheduled processes to reduce the process migration times between the two workshops. In the end, the optimization goal of reducing the total processing time of product and the process migration times was taken into account by adopting the strategy of scheduling the process and allocating the workshop at the same time, which simplified the algorithm implementation and achieved higher equipment utilization.

3. Materials and Methods

3.1. Problem Description

The integrated scheduling problem of two workshops considers the unified scheduling of the processing processes of a single complex product in two workshops with the same equipment. The two workshop integrated scheduling problem is more complex than the single workshop integrated scheduling problem. In addition to the minimum total processing time and the earliest product processing time pursued by the single workshop integrated scheduling, the two workshop integrated scheduling also needs to consider the process migration times and the balanced utilization of equipment.

The preconditions of the two workshops integrated scheduling problem include:

- (1) The different device in each workshop is unique, but the two workshops have the same device;
- (2) A process can only be processed on one device, but it can be allocated to any one device of the two workshops;
- (3) A processing device can only process one process at a certain time;
- (4) A process can only be started when all the immediate preceding processes are finished or there is no immediate preceding process;
- (5) The processing time of each process is known and has nothing to do with the processing sequence;
- (6) The process can wait, and the equipment can be idle before the process arrives.
- (7) Each process and its immediate preceding or immediate following processes can be processed on the device of different workshops;
- (8) If a process is not processed in the same workshop as its immediate preceding or immediate following processes, a process migration will occur;
- (9) No process can be interrupted;
- (10) The completion time of the last process is the total processing time of the product.

Suppose that a product is composed of N processes and is processed in two identical workshops, a and b . Each workshop has m different equipment. In order to complete the product processing as soon as possible, the objective function and constraints of integrated scheduling are expressed as follows:

$$T = \min[\max(ST_{ij} + PT_{ij})] \quad (1)$$

$$s. t. \min(ST_{ij}) \quad (2)$$

$$ST_{wi(j+1)} \geq ST_{wij} + PT_{wij} \quad (3)$$

$$ST_{xy} \geq \max(ST_{ij} + PT_{ij}) \quad (4)$$

$$V = \min\left(\sum_{j=1}^n V_j\right) \quad (5)$$

Here, ST_{ij} is the start processing time of process j on equipment i ; PT_{ij} represents the continuous processing time of process j on equipment i ; w stands for workshop a or b ; ST_{wij} is the start time of the j -th process on equipment i in workshop w ; PT_{wij} is the continuous processing time of the j -th process on equipment i in workshop w ; ST_{ij} is the start time of the immediate preceding process of ST_{xy} ; PT_{ij} is the continuous processing time of the immediate preceding process of ST_{xy} ; V_j refers to the number of migration after the processing workshop is determined for each process; V is the total number of process migration after the product has been processed; $i = 1, 2, \dots, m, j = 1, 2, \dots, n$; Formula (1) represents the goal of minimizing the product's total processing time; Formula (2) indicates that each process starts processing as early as possible; Formula (3) indicates that the process is processed after its immediate preceding processes on the same equipment in the same workshop; Formula (4) indicates that the immediate following processes can only be started after its immediate preceding processes have finished; Formula (5) indicates that the total number of process migration in the process of product processing should be as small as possible.

3.2. Scheduling Strategy Design

3.2.1. Problem Analysis and Concept Definition

The main objective of the integrated scheduling of two workshops is to minimize the product completion time and the process migration times between two workshops. To reduce the total time for product processing, the process must be executed as parallel as possible while meeting the product processing constraints. Most of the existing two workshops' integrated scheduling algorithms sort all processes according to the long path, short time and other strategies to determine the scheduling order, and then determine the start processing time according to the equipment conditions. These methods consider the serial nature of the processes, but it is difficult to realize the full parallel between the processes. Therefore, this paper proposes an integrated scheduling strategy based on the process end time driven. By setting an independent candidate process queue for each device and scheduling processes simultaneously on multiple devices, the parallel processing between processes is strengthened to optimize the total time of product processing. In order to reduce the process migration times between workshops, the distribution of the immediate preceding or immediate following processes and other processes within the same sub-tree range should be considered comprehensively. Therefore, this paper proposes the concept of process processing area, and divides the processes into several process processing areas according to the structure of the process tree. When allocating a workshop for a process, the processes belonging to the same processing area are allocated to the same workshop as possible, to reduce the process migration times between workshops.

In order to facilitate the subsequent description of the algorithm, the following definitions are given first.

Definition 1. *Equipment set: the set of all processing equipment in the scheduling system, which is expressed as*

$$DS = \{D_{a1}, D_{a2}, \dots, D_{aM}, D_{b1}, D_{b2}, \dots, D_{bM}\} \quad (6)$$

Here, a and b refer to the workshop where the equipment is located.

Definition 2. *Equipment status matrix: $a2 \times M$ matrix representing the status of all processing equipment in the scheduling system, which is expressed as*

$$DC = \{DC_{a1}, DC_{a2}, \dots, DC_{aM}, DC_{b1}, DC_{b2}, \dots, DC_{bM}\} \quad (7)$$

Here, DC_{ij} represents the status of the j -th equipment in the i -th workshop, and the value is BUSY or IDLE, $i = a, b, j = 1, 2, \dots, M$.

Definition 3. *Process set: the set of all processing processes in the scheduling system, which is expressed as*

$$PS = \{P_1, P_2, \dots, P_N\} \quad (8)$$

Definition 4. *Constraint relation set: the set of constraint relations among all processes in the scheduling system, which is represented as*

$$CS = \{P_x < P_y\}, 1 \leq x \leq N, 1 \leq y \leq N, x \neq y \quad (9)$$

Here, P_x is the immediate following processes of P_y .

Definition 5. *Equipment candidate process queue: the queue of processes waiting to be scheduled on a device D_m at a certain time t_i , which is represented as*

$$PQ_{D_m t_i} = \{P_{D_m 1}, P_{D_m 2}, \dots, P_{D_m k}\} \quad (10)$$

Definition 6. *Schedulable process: the process that has no immediate preceding process or its immediate preceding process has been completed.*

Definition 7. *Non-schedulable process: the process whose preceding process has not been finished.*

Definition 8. *Schedulable process queue: a queue composed of all schedulable processes on a device D_m at a certain time t_i , which is represented as*

$$SPQ_{D_m t_i} = \{P_{D_m 1}, P_{D_m 2}, \dots, P_{D_m l}\} \quad (11)$$

Definition 9. *Process end time queue: a queue composed of all the process end time determined in the scheduling process, which is represented as*

$$TQ = \{ET_{p_k}\}, 1 \leq k \leq N \quad (12)$$

Definition 10. *Process path length: the sum of the processing time of all processes from the root node (including) to the current process (including) in the product process tree.*

3.2.2. Scheduling Strategy Based on Process End Time Driven

Except for the initial time, there are two events at the end of each process. One is that the state of the equipment where the end process is located becomes idle, so it is necessary to make scheduling decisions on the equipment; the other is that the immediate following process of the end process may change from non-schedulable to schedulable, so it is necessary to make scheduling decisions on its corresponding equipment. By setting a candidate process queue for each device, it is convenient to judge the status of multiple devices and their schedulable processes at the same time, so as to make scheduling decisions on multiple devices.

Specifically, at the initial time of system scheduling, all devices are idle, and all leaf node processes are schedulable. There may be multiple schedulable processes in each device candidate process queue. At this time, the execution order of the schedulable processes can be determined according to the scheduling strategy of long path and long time, and the process with long path and long time is scheduled first. It should be noted that in the existing integrated scheduling algorithms, the scheduling strategy of long path and short time is used to determine the execution order of schedulable processes, that is, when there are multiple schedulable processes, the process with the largest path length is first scheduled; if the process with the largest path length is not unique, the process with

the shortest execution time is first scheduled. The reason is that if the short-time process is executed first, the subsequent process can be executed as soon as possible to reduce the total processing time, which is based on the serial characteristics of the process. However, when multiple equipment are executed at the same time, especially in multiple workshops, the degree of process parallelism will have a greater impact on the total processing time. At this time, the long path and long-time processes may have a greater impact on the subsequent process processing than the short-time processes. Therefore, if the long-time process is given priority over the short-time process, the parallelism of process execution can be increased, and the total processing time can be reduced.

When the processing process is determined for each device according to the long path and long-time strategy, the end time of the scheduled process can be calculated, and then it can be added to the process end time queue in the order of time from small to large. It should be noted that the time in the process end time queue is not necessarily continuous, and there may be multiple times with the same process end time. At this time, the time only needs to be saved once in the process end time queue. Because the time in the process end time queue is sorted in order from small to large, it is easy to insert a new process end time into the queue.

When the process end time is calculated, it is considered that the process can be finished when the end time arrives. At this point, it is necessary to update the set of process constraints, that is, to delete the scheduled process from the set of process constraints and merge the constraints between processes. At the same time, modify the equipment status matrix and set the equipment status of the ending process to idle. In each scheduling process, multiple devices can be scheduled at the same time to reduce the time complexity of the scheduling algorithm.

When a scheduling decision is completed, the process end time queue is checked, and the next scheduling decision is made. Firstly, according to the new process constraints, the schedulable processes in each equipment candidate process queue are determined to form an equipment schedulable process queue; then check the device status matrix to determine the current idle device; next, according to the scheduling strategy of long path and long time, the scheduling order of schedulable processes on each idle device is determined. The specific method is: at each scheduling time, in the schedulable process queue of each equipment, if the schedulable process of a certain equipment is not empty, but the current equipment is not idle, the schedulable process scheduling decision is delayed; if a device is idle, but there is no schedulable process on the device, the device will continue to be idle; if only one equipment in two workshops is idle and there is only one schedulable process on the equipment, the schedulable process is assigned to the corresponding idle equipment; if the current schedulable process is not unique, but there is a unique process with the longest path, the schedulable process with the longest path will be scheduled first; if the current schedulable process is not unique and the process with the longest path is not unique, but the process with the longest execution time in the processes with the longest path is unique, the process with the longest execution time will be scheduled first; if the current schedulable process is not unique and the process with the longest execution time among the processes with the longest path is not unique, any process with the longest path and execution time is selected for scheduling. Finally, if the same kind of equipment in two workshops is idle, the processing workshop and processing equipment are determined for the schedulable process according to the workshop allocation strategy based on process processing area priority described below.

3.2.3. Workshop Allocation Strategy Based on Process Processing Area Priority

In order to reduce the process migration times between two workshops, this paper proposes the concept of process processing area and a workshop allocation strategy based on process processing area priority and adopts the method of scheduling a process and assigning a workshop at the same time, which simplifies the Z algorithm implementation.

Definition 11. Process processing area: in the process tree of a product, the process set formed by a sub-tree of the root node is called a process processing area, which is represented as

$$PS = PA_1 \cup PA_2 \cup \dots \cup PA_t \cup P_1 \quad (13)$$

$$PA_i = \{P_{i1}, P_{i2}, \dots, P_{ik}\} \quad (14)$$

Here, PS refers to the process set in the scheduling system, P_1 represents the root node of the product process tree, PA_i represents the i -th process processing area, and P_{ik} is the node of the i -th sub-tree of the root node.

Definition 12. Neighbor process: in a product process tree, the processes with the same immediate following process are called neighbor processes.

Definition 13. Friend process: in a product process tree, the processes belonging to the same process processing area other than the immediate preceding processes, the immediate following processes and the neighbor processes are called friend processes.

As shown in Figure 1, in the process tree of product P , the three sub-trees of the root node are divided into process processing areas a , b , and c respectively. $P5$ and $P6$ are neighbor processes in the process processing area a , $P29$ and $P30$ are neighbor processes, and $P18$, $P19$, $P20$, and $P21$ are neighbor processes. The friend processes of $P26$ include $P5$, $P18$, $P19$, $P20$, and $P25$. It can be seen from the product process tree that there is a strong dependence between processes in the same process processing area. In order to reduce the number of process migrations between workshops, when assigning processing workshops to schedulable processes, it should try to ensure that processes in the same process processing area are processed in the same workshop.

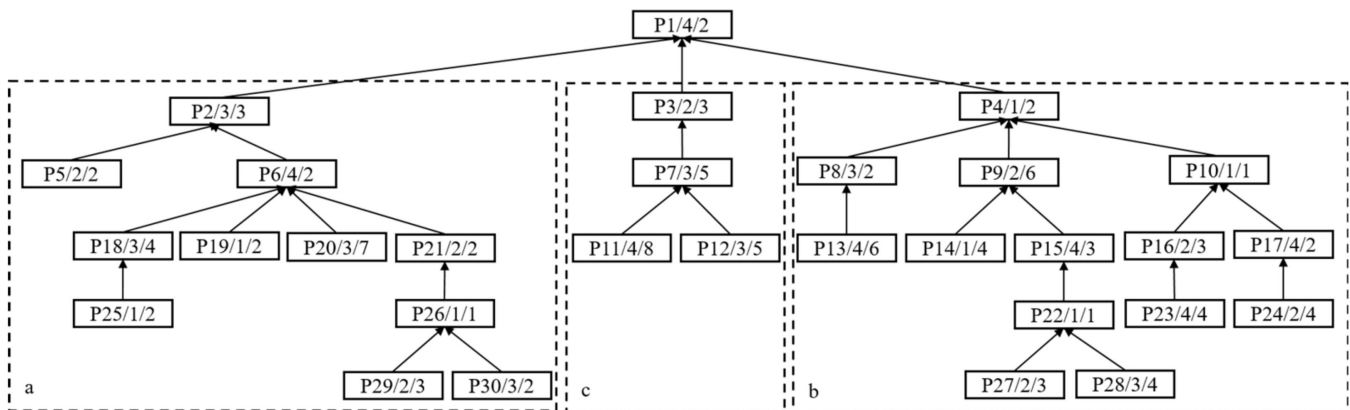


Figure 1. The process tree of product P . a , b and c respectively represent three different processing areas.

The specific design of the workshop allocation strategy based on process processing area priority is as follows: suppose the schedulable process on device D_m is P_i at some time, $1 \leq i \leq N$, the number of its immediate preceding process is recorded as $PreNum_{P_i}$, in which the number of its immediate preceding process in workshop a is recorded as $PreNum_{P_i}^a$, and the number of its immediate preceding process in workshop b is recorded as $PreNum_{P_i}^b$, where $PreNum_{P_i} = PreNum_{P_i}^a + PreNum_{P_i}^b$; the number of its neighbor processes which have been assigned workshop is recorded as $NeiNum_{P_i}$, in which the number of neighbor processes in workshop a is recorded as $NeiNum_{P_i}^a$, and the number of neighbor processes in workshop b is recorded as $NeiNum_{P_i}^b$, where $NeiNum_{P_i} = NeiNum_{P_i}^a + NeiNum_{P_i}^b$; the number of its friend processes which have been assigned workshop is recorded as $FriNum_{P_i}$, in which the number of friend processes in

workshop a is recorded as $FriNum_{P_i}^a$, and the number of friend processes in workshop b is recorded as $FriNum_{P_i}^b$, where $FriNum_{P_i} = FriNum_{P_i}^a + FriNum_{P_i}^b$.

At each scheduled time, according to the scheduling strategy based on process end time driven, the scheduling decision of multiple devices is made at the same time, and the processing workshop and processing equipment are allocated for the scheduled process as follows:

Allocation strategy (1): For each processing device D_m , if there is only one schedulable process P_i in the schedulable process queue at a certain scheduling time and device D_m is only idle in one workshop a or b , if P_i is assigned to the idle device D_m and the number of process migration caused by this assignment does not exceed a certain value (such as 2), the schedulable process P_i is assigned to the workshop where the idle device D_m is located; otherwise, if the number of process migration is greater than the specified value, in order to reduce the number of process migration, the scheduling time of the schedulable process is delayed by one time.

Allocation strategy (2): For each processing device D_m , if there is only one schedulable process P_i in the schedulable process queue at a certain scheduling time and device D_m is idle in both workshops a and b , if $PreNum_{P_i}^a > PreNum_{P_i}^b$, then the schedulable process P_i is assigned to workshop a ; on the contrary, if $PreNum_{P_i}^a < PreNum_{P_i}^b$, then the schedulable process P_i is assigned to workshop b ; if $PreNum_{P_i}^a = PreNum_{P_i}^b$, then further compare $NeiNum_{P_i}^a$ and $NeiNum_{P_i}^b$. If $NeiNum_{P_i}^a > NeiNum_{P_i}^b$, then the schedulable process P_i is assigned to workshop a ; on the contrary, if $NeiNum_{P_i}^a < NeiNum_{P_i}^b$, then the schedulable process P_i is assigned to workshop b ; if $NeiNum_{P_i}^a = NeiNum_{P_i}^b$, then further compare $FriNum_{P_i}^a$ and $FriNum_{P_i}^b$. If $FriNum_{P_i}^a > FriNum_{P_i}^b$, then the schedulable process P_i is assigned to workshop a ; on the contrary, if $FriNum_{P_i}^a < FriNum_{P_i}^b$, then the schedulable process P_i is assigned to workshop b ; if $FriNum_{P_i}^a = FriNum_{P_i}^b$, then the schedulable P_i can be assigned to any workshop a or b .

Allocation strategy (3): For each processing device D_m , if there are more than two schedulable processes in the schedulable process queue at a certain scheduling time, $P_{k1}, P_{k2}, \dots, P_{kl}$, $l \geq 2$, and D_m is only idle in one workshop a or b , then a schedulable process P_{kx} whose process migration times do not exceed a specific value (such as 2) is selected according to the scheduling strategy of long path and long time and assigned to the workshop where the idle device D_m is located; If the process migration times of all schedulable processes are greater than the specified value, then the P_{kx} with minimum process migration times is assigned to the workshop where the idle device D_m is located.

Allocation strategy (4): For each processing device D_m , if there are more than two schedulable processes in the schedulable process queue at a certain scheduling time, $P_{k1}, P_{k2}, \dots, P_{kl}$, $l \geq 2$, and D_m is idle in both workshops a and b , then two schedulable processes P_{kx}^1 and P_{kx}^2 are selected according to the scheduling strategy of long path and long time. If $PreNum_{P_{kx}^1}^a > PreNum_{P_{kx}^2}^a$, then the schedulable process P_{kx}^1 is assigned to workshop a and the schedulable process P_{kx}^2 is assigned to workshop b ; on the contrary, if $PreNum_{P_{kx}^1}^a < PreNum_{P_{kx}^2}^a$, then the schedulable process P_{kx}^1 is assigned to workshop b and the schedulable process P_{kx}^2 is assigned to workshop a ; if $PreNum_{P_{kx}^1}^a = PreNum_{P_{kx}^2}^a$, then further compare $NeiNum_{P_{kx}^1}^a$ and $NeiNum_{P_{kx}^2}^a$. If $NeiNum_{P_{kx}^1}^a > NeiNum_{P_{kx}^2}^a$, then the schedulable process P_{kx}^1 is assigned to workshop a and the schedulable process P_{kx}^2 is assigned to workshop b ; on the contrary, if $NeiNum_{P_{kx}^1}^a < NeiNum_{P_{kx}^2}^a$, then the schedulable process P_{kx}^1 is assigned to workshop b and the schedulable process P_{kx}^2 is assigned to workshop a ; if $NeiNum_{P_{kx}^1}^a = NeiNum_{P_{kx}^2}^a$, then further compare $FriNum_{P_{kx}^1}^a$ and $FriNum_{P_{kx}^2}^a$. If $FriNum_{P_{kx}^1}^a > FriNum_{P_{kx}^2}^a$, then the schedulable process P_{kx}^1 is assigned to workshop a and the schedulable process P_{kx}^2 is assigned to workshop b ; on the contrary, if $FriNum_{P_{kx}^1}^a < FriNum_{P_{kx}^2}^a$, then the schedulable process P_{kx}^1 is assigned to workshop b and

the schedulable process P_{kx}^2 is assigned to workshop a ; if $FriNum_{P_{kx}^1}^a == FriNum_{P_{kx}^2}^a$, then the schedulable process P_{kx}^1 and P_{kx}^2 are randomly assigned to workshop a or b .

3.3. Algorithm Design and Analysis

3.3.1. Algorithm Design

Based on the scheduling strategy and the workshop allocation strategy described above, we proposed a new two workshops integrated scheduling algorithm, named PETD-PAP-TWISA. Next, we introduce the design of the PETD-PAP-TWISA in detail. Firstly, we perform initialization, including three aspects: (1) to initialize the process end time queue (TQ) and the equipment status matrix (DC); (2) to calculate the path length of each process (PL) and to initialize the process constraint relation set (CS) based on the product process tree; and (3) to initialize the candidate process queue (PQ) for each device. Secondly, when PQ is not empty for all devices, we determine the schedulable process queue (SPQ) for each device according to CS . Then, according to the status of each device in the two workshops, DC , we use the corresponding workshop allocation strategies introduced in Section 3.2.3 to allocate processing workshops for the processes. Thirdly, we change the device status of the schedulable process to *BUSY* and compute the start time and the end time of the schedulable process. Fourthly, we add the end time of the schedulable process into TQ and delete the scheduled process from PQ and SPQ . Fifthly, we select the next scheduling time from TQ , change the status of the device whose process ends at that time, and delete the constraint relations of the ended process from CS . Next, we jump to the second step and execute the loop until the candidate process queue for each device is empty. Finally, we output the scheduling Gantt chart according to the order of process execution and total processing time of the product. The flow chart of the PETD-PAP-TWISA is shown in Figure 2.

According to the previous introduction and algorithm flowchart, the pseudocode of the PETD-PAP-TWISA is shown as Algorithm 1:

Algorithm 1 PETD-PAP-TWISA

Input: process data of product

Output: process scheduling scheme and total processing time of product

- Step 1. procedure PETD-PAP-TWISA (process data of product)
- Step 2. Set the process end time queue $TQ = \{t_0 = 0\}$.
- Step 3. Set the equipment status matrix $DC = \{IDLE\}$.
- Step 4. Calculate the path length of each process and store it as $PL = \{PL_1, PL_2, \dots, PL_N\}$.
- Step 5. Set process constraint relation set $CS = CS_{t_0}$.
- Step 6. Set candidate process queue for each device D_m ,
 $PQ_{D_m} = \{P_{D_m1}, P_{D_m2}, \dots, P_{D_mk}\}, 1 \leq m \leq M$.
- Step 7. Set schedulable process queue for D_m , $SPQ_{D_m} = \emptyset, 1 \leq m \leq M$.
- Step 8. **for** ($t = t_i \in TQ, PQ_{D_m} \neq \emptyset, 1 \leq m \leq M$)
- Step 9. Determine schedulable process queue SPQ_{D_mt} on device D_m according to the process constraint relation set CS_t in PQ_{D_mt} , $SPQ_{D_mt} = \{P_{k1}, P_{k2}, \dots, P_{kl}\}, 1 \leq m \leq M$.
- Step 10. For each device D_m , make scheduling decisions respectively: for a certain device D_m , if $SPQ_{D_mt} == \emptyset$, then the device does not need to carry out process scheduling at this time, and then check the next device. For all the devices, if $SPQ_{D_mt} == \emptyset$, jump to Step 22; Otherwise, proceed to the next Step.
- Step 11. For a certain device D_m , if $SPQ_{D_mt} \neq \emptyset$, but its status $DC_{wm} = BUSY, w = a, b$, then device D_m cannot be scheduled at this time, and then check next device. For all devices, if $DC_{wm} = BUSY, w = a, b$, go to Step 22; Otherwise, proceed to the next Step.
-

Algorithm 1 *Cont.*

- Step 12. If there is only one schedulable process P_{k1} in $SPQ_{D_m t}$, and the device status $DC_{am} = IDLE$ or $DC_{bm} = IDLE$, then allocates the processing workshop for P_{k1} according to the workshop allocation strategy (1) in Section 3.2.3, and then jumps to Step 16; Otherwise, proceed to the next Step.
- Step 13. If there is only one schedulable process P_{k1} in $SPQ_{D_m t}$, and the device status $DC_{am} = IDLE$ and $DC_{bm} = IDLE$, then calculate $PreNum_{P_{k1}}^a, PreNum_{P_{k1}}^b, NeiNum_{P_{k1}}^a, NeiNum_{P_{k1}}^b, FriNum_{P_{k1}}^a, FriNum_{P_{k1}}^b$ and allocates the processing workshop for P_{k1} according to the workshop allocation strategy (2) in Section 3.2.3, and then jumps to Step 16; Otherwise, proceed to the next Step.
- Step 14. If there are more than one schedulable process in the $SPQ_{D_m t}, P_{k1}, P_{k2}, \dots, P_{kl}, l \geq 2$, and the device status $DC_{am} = IDLE$ or $DC_{bm} = IDLE$, then selects a schedulable process P_{kx} according to the scheduling strategy of long path and long time, and then allocates the processing workshop for P_{kx} according to the workshop allocation strategy (3) in Section 3.2.3, and then jumps to Step 16; Otherwise, proceed to the next Step.
- Step 15. If there are more than one schedulable process in the $SPQ_{D_m t}, P_{k1}, P_{k2}, \dots, P_{kl}, l \geq 2$, and the device status $DC_{am} = IDLE$ and $DC_{bm} = IDLE$, then selects two schedulable process P_{kx}^1 and P_{kx}^2 according to the scheduling strategy of long path and long time and calculate $PreNum_{P_{kx}^1}^a, PreNum_{P_{kx}^2}^a, NeiNum_{P_{kx}^1}^a, NeiNum_{P_{kx}^2}^a, FriNum_{P_{kx}^1}^a, FriNum_{P_{kx}^2}^a$, and then allocates the processing workshop for P_{kx}^1 and P_{kx}^2 according to the workshop allocation strategy (4) in Section 3.2.3, and then proceed to the next Step.
- Step 16. Set the status of processing device D_m of the scheduled process P_{kx} , $DC_{wm} = BUSY$.
- Step 17. Set the start time of the scheduled process P_{kx} , $ST_{P_{kx}} = t_i$.
- Step 18. Calculate the end time of the scheduled process P_{kx} , $ET_{P_{kx}} = t_i + PT_{P_{kx}}$.
Check the end time of the scheduled process P_{kx} , $ET_{P_{kx}}$, whether it exists in the process end time queue TQ . If it does not exist, then adds $ET_{P_{kx}}$ to the TQ in descending order, $TQ = TQ + \{ET_{P_{kx}}\}$; Otherwise, proceed to the next Step.
- Step 19. Delete the scheduled process P_{kx} from equipment candidate process queue PQ_{D_m} , $PQ_{D_m} = PQ_{D_m} - \{P_{kx}\}$.
- Step 20. Delete the scheduled process P_{kx} from equipment schedulable process queue SPQ_{D_m} , $SPQ_{D_m} = SPQ_{D_m} - \{P_{kx}\}$.
- Step 21. Select the next time $t = t_{i+1}$ from the process end time queue TQ .
- Step 22. At time t_{i+1} , find process P_{kx} whose process end time $ET_{P_{kx}} = t_{i+1}$, set the status of device D_m where P_{kx} is located, $DC_{wm} = IDLE$.
- Step 23. Delete the constraint relation of the process P_{kx} from constraint relation set CS_t , $CS_t = CS_{t_{i+1}}$.
- Step 24. **end for**
- Step 25. Output scheduling Gantt chart according to the order of process execution, which is the process scheduling scheme of product.
- Step 26. Output the maximum process end time in the process end time queue TQ , $T_{max} = \max(TQ)$, which is the total processing time of product.
- Step 27. **end procedure**

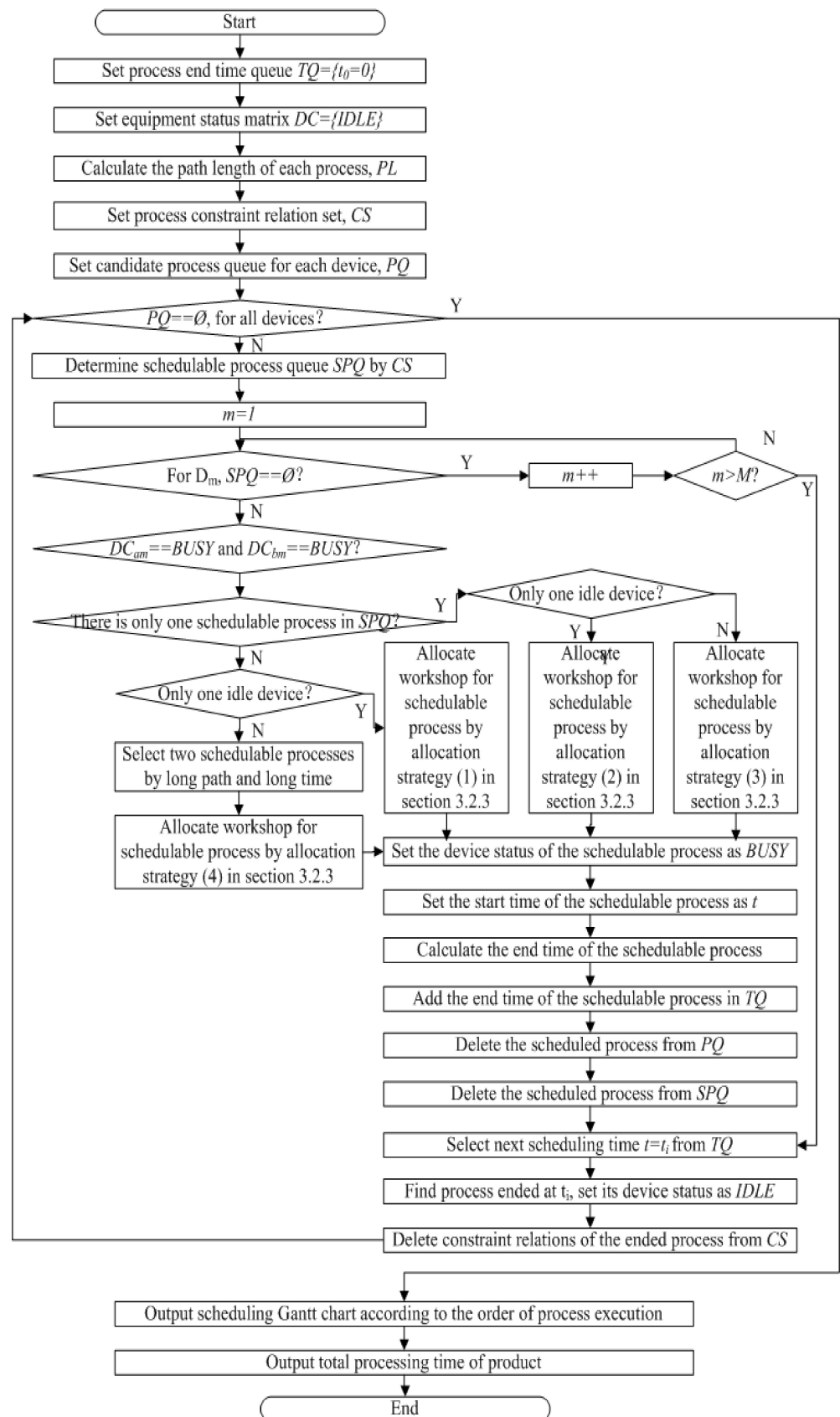


Figure 2. The flow chart of the PETD-PAP-TWIS.

3.3.2. Algorithm Complexity Analysis

Suppose that the product consists of N processes and is processed on M sets of equipment. The complexity of the PETD-PAP-TWISA algorithm is analyzed as follows. The algorithm steps 1–7 are initialization operations, which need to traverse the product process tree, and its time complexity is $O(N)$; Steps 8–25 schedule the processes and allocate the workshops: At step 9, on average, the number of candidate scheduling processes on each kind of equipment is N/M ; for each kind of equipment, to determine the schedulable processes according to the process constraints needs to be compared N/M times at most, and the M kinds of equipment need to be compared N times at most, so the time complexity is $O(N)$; In step 10, it takes M times for each device to determine whether the schedulable process queue is empty, so the time complexity is $O(M)$; In step 11, it takes two steps to determine the state of each device, so the time complexity is $O(M)$; In step 12, according to the workshop allocation strategy (1), a total of M comparisons are needed to allocate workshops for schedulable processes, so the time complexity is $O(M)$; In step 13, according to the workshop allocation strategy (2), to allocate a workshop for the schedulable process, one needs to compute the number of immediate preceding processes, neighbor processes and friend processes that have been allocated a workshop. In the worst-case scenario, each kind of equipment must be calculated no more than $N-1$ times, and M kinds of equipment must be calculated $MM \times (N-1)$ times, resulting in a time complexity of $O(MM \times (N-1))$; In step 14, the schedulable process is allocated to the workshop according to the workshop allocation strategy (3). In the worst case, each kind of equipment needs to be compared N/M times, and M kinds of equipment processes need to be compared N times, so the time complexity is $O(N)$; In step 15, according to the workshop allocation strategy (4), to allocate a workshop for the schedulable process, one need's to compute the number of its immediate preceding process, neighbor process, and friend process which have been allocated workshop. In the worst-case scenario, we need to calculate no more than $2 \times (N-1)$ for each type of equipment, while M types of equipment need to calculate $M \times 2 \times (N-1)$ times, so the time complexity is $O(M \times (N-1))$. In steps 16–21, the number of calculation times required for each kind of equipment to carry out various calculations for the scheduled process is constant, so the total computational complexity of M kinds of equipment is $O(M)$; In steps 22–25, in the worst case, it is necessary to find the process ended at a certain time, set its equipment status, and delete its corresponding constraint relationship need to execute $MM \times 2 \times N/MM$ times, so the time complexity is $O(N)$. In general, at each scheduling time in steps 8–25, the maximum time complexity is $O(MM \times N)$; moreover, there are at most N times scheduled in the system scheduling process, so the total time complexity of steps 7–24 is $O(MM \times N^2)$. Finally, the time complexity of step 26 is $O(N)$, and the time complexity of step 27 is constant. To sum up, the total time complexity of the PETD-PAP-TWISA algorithm is less than $O(MM \times N^2)$. Because of $M \ll N$ in general, the time complexity of the PETD-PAP-TWISA algorithm is not more than a quadratic polynomial.

4. Comparison and Analysis

4.1. Case Analysis

The PETD-PAP-TWISA proposed in this paper is general and does not take concrete examples as objects. In order to help readers better understand the algorithm, the product P shown in Figure 1 is taken as an example to illustrate the execution process of the integrated scheduling algorithm of two workshops based on the process end time driven and processing area priority. Product P has a total of 30 processes, which are processed in two workshops with the same resources, and each workshop has four different devices.

At time T_0 , the candidate process queue on device M_1 , M_2 , M_3 , and M_4 is $\{P_4, P_{10}, P_{14}, P_{19}, P_{22}, P_{25}, P_{26}\}$, $\{P_3, P_5, P_9, P_{16}, P_{21}, P_{24}, P_{27}, P_{29}\}$, $\{P_2, P_7, P_8, P_{12}, P_{18}, P_{20}, P_{28}, P_{30}\}$ and $\{P_1, P_6, P_{11}, P_{13}, P_{15}, P_{17}, P_{23}\}$ respectively. The process constraint relation set CS_{T_0} is $\{P_1 < P_2 < P_5, P_1 < P_2 < P_6 < P_{18} < P_{25}, P_1 < P_2 < P_6 < P_{19}, P_1 < P_2 < P_6 < P_{20}, P_1 < P_2 < P_6 < P_{21} < P_{26} < P_{29}, P_1 < P_2 < P_6 < P_{21} < P_{26} < P_{30}, P_1 < P_3 < P_7 < P_{11},$

$P1 < P3 < P7 < P12$, $P1 < P4 < P8 < P13$, $P1 < P4 < P9 < P14$, $P1 < P4 < P9 < P15 < P22 < P27$,
 $P1 < P4 < P9 < P15 < P22 < P28$, $P1 < P4 < P10 < P16 < P23$, $P1 < P4 < P10 < P17 < P24$.
 The schedulable process queue on device M1, M2, M3 and M4 is {P14, P19, P25}, {P5, P24, P27, P29}, {P12, P20, P28, P30} and {P11, P13, P23} respectively. Currently, the status of M1, M2, M3, and M4 is IDLE in the two workshops. According to the long path and long-time scheduling strategy, two processes P14 and P25 are selected from the schedulable process queue of M1 for scheduling. Since all processes in the scheduling system are not assigned to workshops at this time, process P14 is randomly assigned to workshop a, according to the workshop allocation strategy (4), and process P25 is assigned to workshop b. Next, according to the long path and long-time scheduling strategy, two processes, P27 and P29, are selected from the schedulable process queue of M2 for scheduling. At this time, both P27 and P29 have no immediate preceding processes, and their neighbor processes have not been assigned to workshops. However, P27 and P14 belong to the same process processing area b, which are friend processes, and P14 has been assigned to workshop a; P29 and P25 belong to the same process processing area b, which is a friend process. P25 has been assigned to workshop b; Therefore, according to the workshop allocation strategy (4), process P27 is assigned to workshop a, and P29 to workshop b. Next, according to the long path and long-time scheduling strategies, two processes, P28 and P12, are selected from the scheduling process queue of M3 for scheduling. At this time, P28 and P12 have no immediate preceding processes. P28's neighbor process P27 has been assigned to workshop a, and the neighbor processes of P12 have not been assigned to the workshop. Therefore, according to the workshop allocation strategy (4), process P28 is assigned to workshop a, and P12 to workshop b. Next, according to the long path and long-time scheduling strategies, two processes, P11 and P13, are selected from the scheduling process queue of M4 to schedule. At this time, neither P11 nor P13 has an immediate preceding process. P11's neighbor process P12 has been assigned to workshop b, and P13 has no neighbor process, but P13's friend processes P14, P27, and P28 have been assigned to workshop a, so according to the workshop allocation strategy (4), assign process P13 to workshop a, and P11 to workshop b. At this time, the process scheduling and workshop assignment on all equipment are completed, and all equipment status of all workshops is set to BUSY. Because $t_0 = 0$, the end time of P14 is 4, recorded as T_4 ; the end time of P25 is 2, recorded as T_2 ; the end time of P27 is 3, recorded as T_3 ; the end time of P29 is 3, recorded as T_3 ; the end time of P28 is 4, recorded as T_4 ; the end time of P12 is 5, recorded as T_5 ; the end time of P13 is 6, recorded as T_6 ; the end time of P11 is 8, recorded as T_8 . Next, add the process end time $T_2, T_3, T_4, T_5, T_6, T_8$ to the process end time queue TQ in order, and $TQ = \{T_0, T_2, T_3, T_4, T_5, T_6, T_8\}$. Next, after deleting the scheduled process from the candidate process queue of M1, M2, M3, and M4, the candidate process queue of M1 is {P4, P10, P19, P22, P26}, the candidate process queue of M2 is {P3, P5, P9, P16, P21, P24}, the candidate process queue of M3 is {P2, P7, P8, P18, P20, P30}, and the candidate process queue of M4 is {P1, P6, P15, P17, P23}. Next, after deleting the scheduled process from the schedulable process queue of M1, M2, M3, and M4, the schedulable process queue of M1 is {P19}, the schedulable process queue of M2 is {P5, P24}, the schedulable process queue of M3 is {P20, P30}, and the schedulable process queue of M4 is {P23}. Next, at time T_2 , find the process P25 that ends at that time, set the status of M1 in workshop a where P25 is located to IDLE, and modify the constraint relationship $P1 < P2 < P5$ in the process constraint relation set to $P1 < P2$. Because $P1 < P2$ is a subset of other constraint relation such as $P1 < P2 < P6 < P18 < P25$, so it can be merged into the constraint relation $P1 < P2 < P6 < P18 < P25$ to get a new constraint relation CS_{T_2} . At this time, at T_2 , the next round of process scheduling and workshop allocation starts. This continues, until the candidate process set on all devices is empty and the algorithm ends. The complete scheduling process of product P is shown in Table 1. The Gantt chart of system scheduling can be drawn according to the order of the processes to be scheduled, their start time, and processing time. As shown in Figure 3, the maximum time in the process end time queue is recorded as the total processing time of the product. In this case, the total processing time of product P is 21.

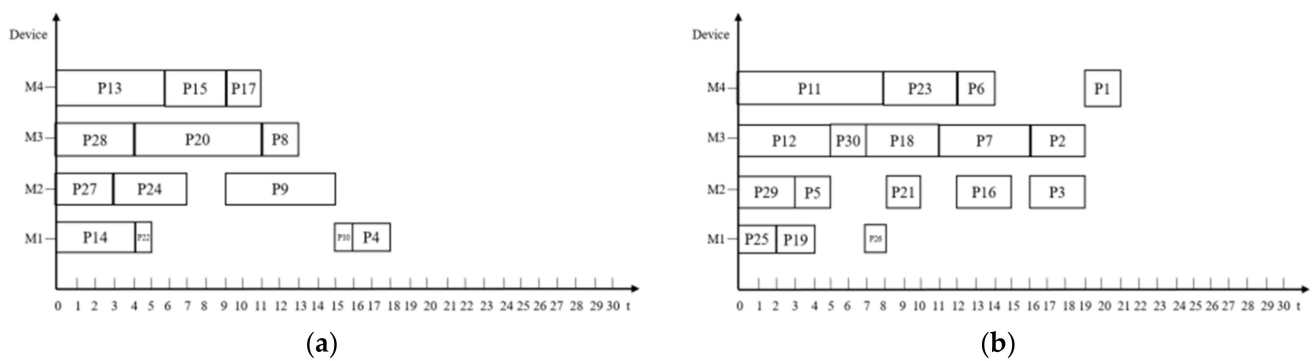


Figure 3. Gantt chart of workshops, a and b, for product P scheduling based on PETD-PAP-TWISA. (a) workshop a (b) workshop b.

Table 1. The processing scheduling process of product P by the PETD-PAP-TWISA.

Time	Device	Candidate Process	Schedulable Process	Scheduled Process	Workshop	End Time	Process Migration
T0	M1	P4(2)-4, P10(1)-5, P14(4)-14, P19(2)-9, P22(1)-14, P25(2)-13, P26(1)-10	P14, P19, P25	P14	a	T4	-
				P25	b	T2	-
	M2	P3(3)-5, P5(2)-7, P9(6)-10, P16(3)-8, P21(2)-9, P24(4)-11, P27(3)-17, P29(3)-13	P5, P24, P27, P29	P27	a	T3	-
				P29	b	T3	-
T2	M3	P2(3)-5, P7(5)-10, P8(2)-6, P12(5)-15, P18(4)-11, P20(7)-14, P28(4)-18, P30(2)-12	P12, P20, P28, P30	P28	a	T4	-
				P12	b	T5	-
T3	M4	P1(2)-2, P6(2)-7, P11(8)-18, P13(6)-12, P15(3)-13, P17(2)-7, P23(4)-12	P11, P13, P23	P13	a	T6	-
				P11	b	T8	-
T4	M1	P4(2)-4, P10(1)-5, P19(2)-9, P22(1)-14, P26(1)-10	P19	P19	b	T4	-
T5	M2	P3(3)-5, P5(2)-7, P9(6)-10, P16(3)-8, P21(2)-9, P24(4)-11	P5, P24	P24	a	T7	-
				P5	b	T5	-
T6	M1	P4(2)-4, P10(1)-5, P22(1)-14, P26(1)-10	P22	P22	a	T5	-
				-	b	-	-
T7	M3	P2(3)-5, P7(5)-10, P8(2)-6, P18(4)-11, P20(7)-14, P30(2)-12	P20, P30	P20	a	T11	-
				-	a	-	-
T8	M1	P4(2)-4, P10(1)-5, P26(1)-10	-	-	b	-	-
				-	a	-	-
T9	M2	P3(3)-5, P9(6)-10, P16(3)-8, P21(2)-9	-	-	b	-	-
				-	a	-	-
T10	M3	P2(3)-5, P7(5)-10, P8(2)-6, P18(4)-11, P30(2)-12	P18, P30	P30	b	T7	-
				-	a	-	-
T11	M1	P4(2)-4, P10(1)-5, P26(1)-10	-	-	b	-	-
				-	a	-	-
T12	M2	P3(3)-5, P9(6)-10, P16(3)-8, P21(2)-9	-	-	a	-	-
				-	b	-	-
T13	M4	P1(2)-2, P6(2)-7, P15(3)-13, P17(2)-7, P23(4)-12	P15, P23	P15	a	T9	-
				-	b	-	-
T14	M1	P4(2)-4, P10(1)-5, P26(1)-10	P26	P26	a	-	-
				-	b	-	-
T15	M2	P3(3)-5, P9(6)-10, P16(3)-8, P21(2)-9	-	-	a	-	-
				-	b	-	-
T16	M3	P2(3)-5, P7(5)-10, P8(2)-6, P18(4)-11	P8, P18	P18	b	T11	-
				-	a	-	-

Table 1. Cont.

Time	Device	Candidate Process	Schedulable Process	Scheduled Process	Workshop	End Time	Process Migration
T8	M1	P4(2)-4, P10(1)-5	-	-	a	-	-
				-	b	-	-
	M2	P3(3)-5, P9(6)-10, P16(3)-8, P21(2)-9	P21	P21	a	-	-
	M4	P1(2)-2, P6(2)-7, P17(2)-7, P23(4)-12	P17, P23	P23	b	T10	-
T9	M1	P4(2)-4, P10(1)-5	-	-	a	-	-
				-	b	-	-
	M2	P3(3)-5, P9(6)-10, P16(3)-8	P9	P9	a	T15	-
	M4	P1(2)-2, P6(2)-7, P17(2)-7	P17	P17	a	T11	-
T10	M1	P4(2)-4, P10(1)-5	-	-	a	-	-
				-	b	-	-
	M2	P3(3)-5, P16(3)-8	-	-	b	-	-
				-	a	-	-
T11	M1	P4(2)-4, P10(1)-5	-	-	b	-	-
				-	a	-	-
	M2	P3(3)-5, P16(3)-8	-	-	b	-	-
				-	a	-	-
T12	M3	P2(3)-5, P7(5)-10, P8(2)-6	P7, P8	P8	a	T13	-
				P7	b	T16	-
	M4	P1(2)-2, P6(2)-7	-	-	a	-	-
				-	a	-	-
T13	M1	P4(2)-4, P10(1)-5	-	-	b	-	-
				-	a	-	-
	M2	P3(3)-5, P16(3)-8	P16	P16	b	T15	-
				-	a	-	-
T14	M4	P1(2)-2, P6(2)-7	P6	P6	b	T14	P20:a->b
				-	a	-	-
	M1	P4(2)-4, P10(1)-5	-	-	b	-	-
				-	a	-	-
T15	M3	P2(3)-5	-	-	a	-	-
				-	a	-	-
	M4	P1(2)-2	-	-	a	-	-
				-	a	-	-
T16	M1	P4(2)-4, P10(1)-5	-	-	a	-	-
				-	b	-	-
	M2	P3(3)-5	-	-	a	-	-
				-	a	-	-
T17	M3	P2(3)-5	-	-	a	-	-
				-	a	-	-
	M4	P1(2)-2	-	-	a	-	-
				-	a	-	-
T18	M1	P4(2)-4	P4	P4	a	T18	-
				-	b	-	-
	M2	P3(3)-5	P3	P3	a	-	-
				-	b	-	-
T19	M3	P2(3)-5	P2	P2	a	T19	-
				-	b	-	-
	M4	P1(2)-2	-	-	a	-	-
				-	b	-	-
T20	M1	-	-	-	a	-	-
				-	b	-	-
	M2	-	-	-	a	-	-
				-	a	-	-
T21	M3	-	-	-	a	-	-
				-	a	-	-
	M4	P1(2)-2	-	-	a	-	-
				-	b	-	-

Table 1. Cont.

Time	Device	Candidate Process	Schedulable Process	Scheduled Process	Workshop	End Time	Process Migration
T19	M1	-	-	-	a	-	-
					b	-	-
	M2	-	-	-	a	-	-
					b	-	-
T21	M3	-	-	-	a	-	-
					b	-	-
T21	M4	P1(2)-2	P1	P1	a	-	-
					b	T21	P4:a->b
	M1	-	-	-	a	-	-
					b	-	-
T21	M2	-	-	-	a	-	-
					b	-	-
	M3	-	-	-	a	-	-
					b	-	-
T21	M4	-	-	-	a	-	-
					b	-	-

Note: P4 (2)–4 in the table indicates that the execution time of process P4 is 2, and the path length in the process tree is 4.

4.2. Performance Evaluation and Comparison

To assess the PETD-PAP-TWISA's performance, we compare it to the four existing two workshops integrated scheduling algorithms using the processing example of product P shown in Figure 1. The algorithms involved in the comparison are briefly described below.

PETD-PAP-TWISA: the integrated scheduling algorithm of two workshops based on the process end time driven and processing area priority, proposed in this paper. The scheduling results of PETD-PAP-TWISA are shown in Figure 3.

NR-TWISA: the integrated scheduling algorithm of two workshops based on neighborhood rendering [12]. The NR-TWISA proposed the critical-equipment equilibrium strategy, which prioritized allocation of processes of key-equipment balanced to two workshops based on the longest parallel processing scheme. It also adopted a neighborhood rendering strategy. By calculating the rendering factor and the influence factor of the neighborhood process, other unallocated workshops are determined. The scheduling results of NR-TWISA are shown in Figure 4.

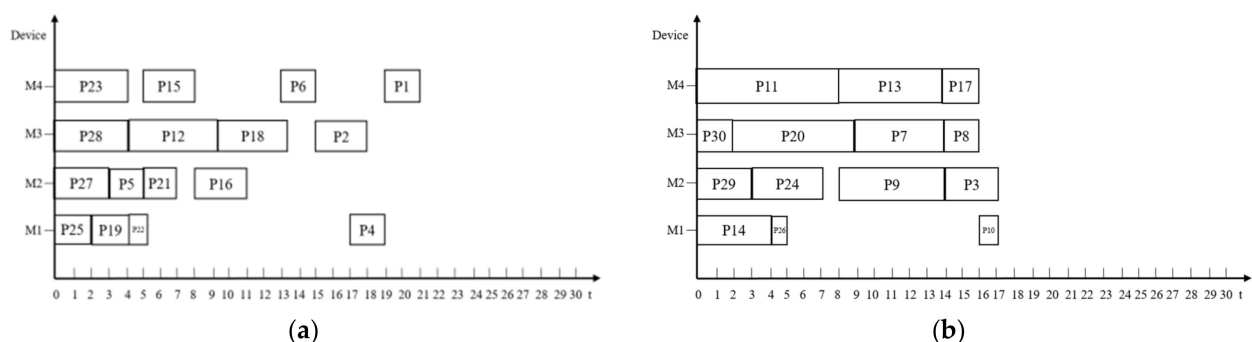


Figure 4. Gantt chart of workshops, a and b, for product P scheduling based on NR-TWISA. (a) workshop a (b) workshop b.

ED-TWISA: the integrated scheduling algorithm of two workshops based on equipment driving [12]. The ED-TWISA is driven by the event that the equipment is idle, and each schedule is for the purpose of busy equipment. The scheduling results of ED-TWISA are shown in Figure 5.

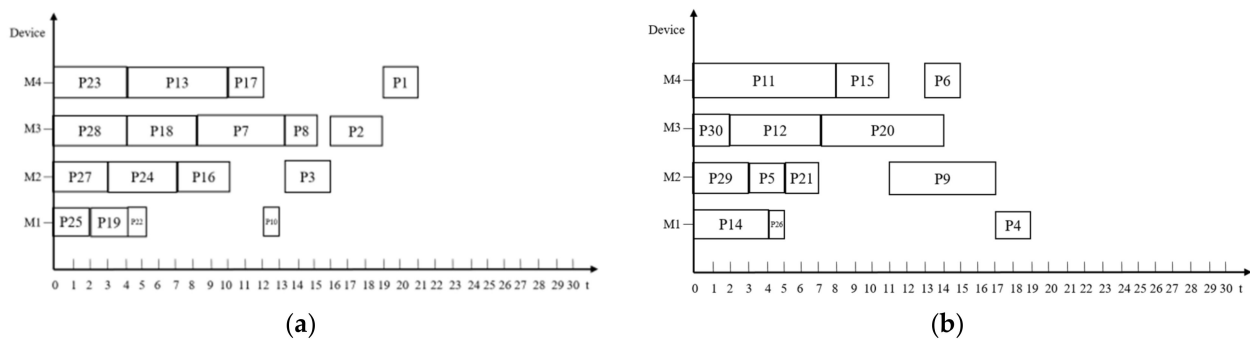


Figure 5. Gantt chart of workshop, a and b, for product P scheduling based on ED-TWISA. (a) workshop a (b) workshop b.

BP-TWISA: the integrated scheduling algorithm for balancing schedulable processes of two workshops [11]. The BP-TWISA considered the flexibility and parallelism of schedulable processes and the condition of two workshops having the same equipment; used workshop equilibrium strategies to group the schedulable processes; and assigned the processes to the workshop based on the proposed workshop determination strategy. The scheduling results of BP-TWISA are shown in Figure 6.

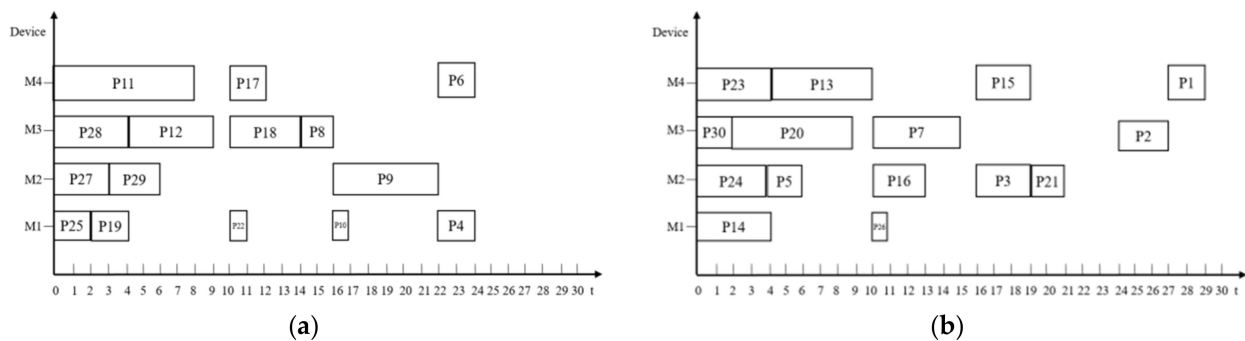


Figure 6. Gantt chart of workshop, a and b, for product P scheduling based on BP-TWISA. (a) workshop a (b) workshop b.

ACPM-TWISA: the two workshops integrated scheduling algorithm based on the allied critical path method [10]. In order to balance the two workshops' load and complete products processing as soon as possible, the ACPM-TWISA sorted the processes according to ACPM and took the pre-scheduled strategy with a close processing end time of the two workshops for scheduling. The scheduling results of ACPM-TWISA are shown in Figure 7.

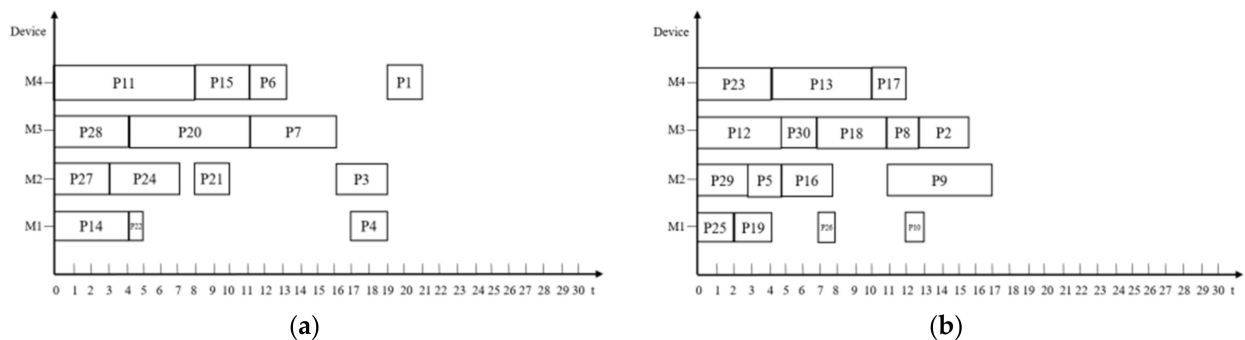


Figure 7. Gantt chart of workshop, a and b, for product P scheduling based on ACPM-TWISA. (a) workshop a (b) workshop b.

According to Figures 3–7, the completion time of the product P by five algorithms can be calculated, which is shown in Table 2. By the same token, the equipment utilization rate of the two workshops by five algorithms is shown in Table 3, the average equipment utilization rate of the two workshops by five algorithms is shown in Table 4, and the migration times of processes by five algorithms are shown in Table 5.

Table 2. Completion time of product P by five algorithms.

Algorithm	Completion Time of Devices in Workshop a/h				Completion Time of Devices in Workshop b/h			
	M1	M2	M3	M4	M1	M2	M3	M4
NR-TWISA	19	11	18	21	17	17	16	16
ED-TWISA	13	16	19	21	19	17	14	15
BP-TWISA	24	22	16	24	11	21	27	29
ACPM-TWISA	19	19	16	21	13	17	16	12
PETD-PAP-TWISA	18	15	13	11	8	19	19	21

Table 3. Equipment utilization rate of two workshops by five algorithms.

Algorithm	Equipment Utilization Rate of Workshop a				Equipment Utilization Rate of Workshop b			
	M1	M2	M3	M4	M1	M2	M3	M4
NR-TWISA	0.37	0.91	0.89	0.52	0.35	0.94	1.0	1.0
ED-TWISA	0.46	0.81	0.95	0.67	0.37	0.76	1.0	0.87
BP-TWISA	0.33	0.55	0.94	0.5	0.45	0.67	0.63	0.52
ACPM-TWISA	0.37	0.63	1.0	0.71	0.46	0.82	1.0	1.0
PETD-PAP-TWISA	0.44	0.87	1.0	1.0	0.63	0.68	1.0	0.76

Table 4. Average equipment utilization rate of two workshops by five algorithms.

Algorithm	NR-TWISA	ED-TWISA	BP-TWISA	ACPM-TWISA	PETD-PAP-TWISA
Workshop a	0.67	0.72	0.58	0.68	0.83
Workshop b	0.82	0.75	0.57	0.82	0.77
Mean	0.75	0.74	0.58	0.75	0.8

Table 5. Migration times of processes by five algorithms.

Algorithm	NR-TWISA	ED-TWISA	BP-TWISA	ACPM-TWISA	PETD-PAP-TWISA
Migration time	6	9	12	10	3

As one can see from Figures 4–7 and Table 2, except for the BP-TWISA, the completion time of product P obtained by the NR-TWISA, the ED-TWISA, the ACPM-TWISA, and the PETD-PAP-TWISA is 21, which is the optimal result for product P integrated scheduling, indicating that these four algorithms are equally matched in optimizing the total processing time of product. The completion time of the BP-TWISA is 29, which is much longer than that of the other four algorithms.

It can be seen from Tables 3 and 4 that the average equipment utilization rate of the PETD-PAP-TWISA is significantly higher than that of the other four algorithms. Specifically, the average equipment utilization rate of the two workshops by the NR-TWISA, the ED-TWISA, the BP-TWISA, the ACPM-TWISA, and the PETD-PAP-TWISA is 0.75, 0.74, 0.58, 0.75, and 0.8, respectively.

It can be seen from Table 5 that the migration times of processes by the NR-TWISA, the ED-TWISA, the BP-TWISA, the ACPM-TWISA, and the PETD-PAP-TWISA are 6, 9, 12, 10, and 3, respectively. Obviously, the migration time of processes by the PETD-PAP-TWISA is the smallest among all the algorithms involved in the comparison.

5. Discussion

According to the previous performance evaluation and comparison, it can be easily seen that the PETD-PAP-TWISA, proposed in this paper, has better performance in terms of the total processing time, the migration time of processes, and equipment utilization rate. The reasons for this result are briefly analyzed and discussed below.

The NR-TWISA comprehensively considers completion time and migration times. By adopting the balance strategy of key equipment, the product can be completed as early as possible. By employing the rendering strategy, the processing arrangement of processes in the two workshops is balanced to avoid unnecessary migration. In terms of completion time, both the NR-TWISA and the PETD-PAP-TWISA have reached the minimum value. However, the NR-TWISA does not fully optimize the migration times of adjacent processes between two workshops. Therefore, the migration time of the NR-TWISA is obviously greater than that of the PETD-PAP-TWISA. In addition, the average equipment utilization rate of the NR-TWISA is lower than the PETD-PAP-TWISA.

The ED-TWISA is driven by the event that the equipment is idle, and the equipment is busy for the purpose of each schedule, so the completion time is necessarily shorter. In fact, the ED-TWISA has achieved the minimum value of completion time. However, the impact of different combination schemes between processes on the parallel time and the migration between processes is not considered in the processing process. Therefore, the migration between processes is more frequent than the NR-TWISA and the PETD-PAP-TWISA.

The BP-TWISA processes the leaf node processes in batches to balance the total processing time of the equipment, but to unify the start time of each batch of processes results in a long idle waiting time for the equipment and a corresponding extension of the completion time. Moreover, the influence of the parent node on process migration cannot be taken into account, so there are too many process migrations. In fact, the competing time of the BP-TWISA is 29, which is much longer than that of the other four algorithms. In addition, the average equipment utilization rate of the BP-TWISA is the minimum, and the process migration time is the maximum among all algorithms involved in the comparison.

The ACPM-TWISA is mainly aimed at products with serial processes in the processing process tree, and its application scope has great limitations. Non-serial processes can only be processed by the allied critical path method, so a large number of migration times are caused.

The PETD-PAP-TWISA employs the scheduling strategy based on the process end time driven and processing area priority, which can make better use of the idle time of equipment and achieve the purpose of keeping the equipment as busy as possible. It can be seen from Tables 3 and 4 that the average equipment utilization rate of the PETD-PAP-TWISA is significantly higher than that of the other four algorithms. Moreover, the proposed workshop allocation strategy based on the processing area priority fully considers the influence of the immediate preceding process, neighbor process, and friend process of the schedulable process on the workshop allocation, and arranges the closely related processes in the same workshop as far as possible. Therefore, the PETD-PAP-TWISA obtains the minimum process migration time among all algorithms involved in the comparison. Finally, the PETD-PAP-TWISA adopts the strategy based on the process end time driven to make scheduling decisions on multiple devices at the same time, which improves the parallelism of two workshops' process scheduling and simplifies the implementation of the algorithm.

6. Conclusions

The integrated scheduling algorithm of two workshops based on process end time driven and processing area priority proposed the parallel scheduling strategy driven by

process end time and the workshop allocation strategy based on process processing area priority. The main conclusions are as follows:

- (1) The process scheduling strategy based on the process end time driven can improve the parallelism of processes by scheduling multiple devices at the same time, which can achieve the optimization goal of reducing the total processing time. In addition, the process scheduling strategy based on the process end time driven can make the processing equipment as busy as possible, to improve the utilization of the equipment. Moreover, the algorithm is simple and easy to implement;
- (2) The workshop allocation strategy based on the processing area priority takes into account the influence of the immediate preceding process, neighbor process, and friend process of the schedulable process on the workshop allocation and tries to allocate the processes in the same processing area to the same workshop, so as to achieve the optimization goal of reducing the number of process migrations between two workshops.

To sum up, the integrated scheduling algorithm based on process end time driven and processing area priority provides a new idea for solving the two workshops integrated scheduling problem. Further research on the multi-workshops integrated scheduling problem based on process end time driven and processing area priority is a direction with important theoretical and practical significance.

Author Contributions: Conceptualization, X.Z.; methodology, X.Z. and Z.X.; project administration, Z.X.; software, X.Z. and D.Y.; supervision, Z.X.; writing—original draft, X.Z.; writing—review and editing, X.Z. and D.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No. 61772160) and Postdoctoral Science-Research Developmental Foundation of Heilongjiang Province of China (No. LBHQ13092).

Acknowledgments: We would like to thank the reviewers for their comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xiong, H.G.; Ren, S.D.; Hu, J.J. A survey of job shop scheduling problem: The types and models. *Comput. Oper. Res.* **2022**, *142*, 105731. [\[CrossRef\]](#)
2. Mokhtari, H. A two-stage no-wait job shop scheduling problem by using a neuro-evolutionary variable neighborhood search. *Int. J. Adv. Manuf. Technol.* **2014**, *74*, 1595–1610. [\[CrossRef\]](#)
3. Zhao, S.K. Bilevel neighborhood search hybrid algorithm for the flexible job shop scheduling problem. *J. Mech. Eng.* **2015**, *51*, 175–184. [\[CrossRef\]](#)
4. Mohamad, R.; Amirsaman, K.; Parviz, F.; Behdin, V.N. A hybrid multi-objective genetic algorithm based on the ELECTRE method for a capacitated flexible job shop scheduling problem. *Int. J. Adv. Manuf. Technol.* **2015**, *77*, 51–66.
5. An, Y.J.; Chen, X.H.; Zhang, J.; Li, Y.H. A hybrid multi-objective evolutionary algorithm to integrate optimization of the production scheduling and imperfect cutting tool maintenance considering total energy consumption. *J. Clean. Prod.* **2020**, *268*, 121540. [\[CrossRef\]](#)
6. Xie, Z.Q.; Xin, Y.; Yang, J. Integrated scheduling algorithm based on event driven by machines idle. *J. Mech. Eng.* **2011**, *47*, 139–147. [\[CrossRef\]](#)
7. Xie, Z.Q.; Zhang, X.H.; Xin, Y.; Yang, J. Time-selective integrated scheduling algorithm considering posterior processes. *Acta Autom. Sin.* **2018**, *44*, 345–362.
8. Guo, W.F.; Lei, Q.; Song, Y.C.; Lü, X.F.; Li, L. Integrated Scheduling Algorithm of Complex Product with No-wait Constraint Based on Virtual Component. *J. Mech. Eng.* **2020**, *56*, 246–257.
9. Wang, Z.; Zhang, X.H.; Cao, Y.X.; Lai, G.M. An integrated scheduling algorithm for multi-device-processes with the strategy of exchanging adjacent parallel processes of the same device. *EURASIP J. Wirel. Commun. Netw.* **2021**, *2021*, 104. [\[CrossRef\]](#)
10. Xie, Z.Q.; Zhou, H.X.; Gui, Z.Y.; Zheng, F.P. Integrated scheduling algorithm of two workshops based on ACPM. *Comput. Sci.* **2013**, *40*, 193–198.
11. Xie, Z.Q.; Zheng, F.P.; Zhu, T.H.; Zhou, H.X. Integrated scheduling algorithm with equalization processing of schedulable processes in two workshops. *Comput. Eng.* **2014**, *40*, 295–300.
12. Xie, Z.Q.; Yu, J.; Chen, D.Y.; Yang, J. Integrated scheduling algorithm of two workshops based on the principle of the neighbourhood rendering. *J. Mech. Eng.* **2016**, *52*, 149–159. [\[CrossRef\]](#)