

RESEARCH

Open Access



# An integrated scheduling algorithm for multi-device-processes with the strategy of exchanging adjacent parallel processes of the same device

Zhen Wang<sup>1</sup>, Xiaohuan Zhang<sup>1\*</sup>, Yuxiao Cao<sup>2</sup> and Guoming Lai<sup>1</sup>

\*Correspondence:  
zhangxiaohuan@hzu.edu.cn

<sup>1</sup> School of Computer  
Science and Engineering,  
Huizhou University,  
Huizhou 516007, People's  
Republic of China  
Full list of author information  
is available at the end of the  
article

## Abstract

At present, Multi-Devices-Process Integrated Scheduling Algorithm with Time-Selective Strategy for Process Sequence (MISATPS) is an advanced algorithm in the field of integrated scheduling with multi-devices-process problems. This algorithm ignores the influence of the pre-process on the post-process when solving the multi-devices-process integrated scheduling problem, which leads to the problem of poor closeness between serial processes and poor parallelism between parallel processes. This paper points out that there is no restriction of scheduling sequence between parallel processes on the same processing device. It can be scheduled flexibly of the sequence between parallel processes of the same device. Therefore, based on the scheduling scheme of MISATPS, the algorithm is improved by applying the interchange strategy and the interchange adjustment strategy of multi-device adjacent parallel process. In this way, the influence of the pre-process on the post-process is avoided, the compactness of the serial process and the parallelism of the parallel process are improved, and the scheduling result is optimized.

**Keywords:** Multi-Device-Process, Integrated scheduling algorithm, Adjacent parallel processes, Parallel processes, Serial processes, Interchange adjustment

## 1 Introduction

The traditional production and manufacturing process is to decompose the products according to the workpiece and make them by the manufacturing-before-assembly method [1–3]. While flow-shop is adopted for large quantities of the same products, Job-shop is adopted for multi-variety and small-batch products. Therefore, the corresponding pure machining scheduling and pure assembly scheduling have been deeply studied. Sya et al. [4] solved scheduling problem with genetic algorithm, Huang et al. [5] and Shen et al. [6] applied neural network to solve scheduling problem, Swlab et al. [7] solved scheduling problem with simulated annealing method, Mathlouthi et al. [8] applied Tabu search method to solve scheduling problem, and Wu et al. [9] and Anghinolfi et al. [10] solved scheduling problem with heuristic algorithm.

With the increase in the social demand for personalized products, more and more attention is paid to the production mode of multi-variety and small batches. For multi-variety and small batch products, especially complex single products with tree structure, if the mode of manufacturing-before-assembly is adopted, the parallel processing relationship between product processing and assembly will be separated inevitably and will affect the manufacturing efficiency of the product. Therefore, Xie et al. [11, 12] proposed a complex single-product comprehensive scheduling algorithm for processing and assembly at the same time.

In the actual production of assembly manufacturing enterprises, there is a situation that a single process needs multiple devices to process together. For example, it needs two devices to hold and one welding device to weld when welding two large parts together. That means the welding process needs three devices to complete one process. Therefore, it is necessary to solve the integrated scheduling problem that a single complex product needs multiple equipment to complete a single special process.

The rest of this paper is organized as follows. Second section shows the problem description and analysis. Third section discusses the strategy design. The optimization algorithm is proposed in fourth section. Fifth section shows the case analysis. Sixth section shows experimental comparison and analysis. Seventh section concludes the paper with summary.

## 2 Problem description and analysis

The process tree of complex products is a tree-like structure established according to the restriction relationship between processes of the product. The nodes in the process tree represent the process of the product, the directed edges in the process tree represent the partial order relation of the process order, and the root node in the process tree denotes the final step of the product. It indicates that the product is finished when the root process comes to an end. When multi-device processes exist in the process set, the complex single-product scheduling problem must satisfy the following six constraints simultaneously and minimize the total processing time of the product.

- (1) Each process must strictly comply with the partial order relationship agreed in the process tree;
- (2) Each device can only process one procedure at any time, and the processing process cannot be interrupted;
- (3) Devices with the same functionality do not exist on the device set;
- (4) In the process set, there are multi-device-processes which are jointly processed by multiple related devices;
- (5) Each process must wait for the completion of all the process tree pre-process and device pre-process before processing;
- (6) The difference between the finishing time of the latest processing procedure and the beginning time of the earliest processing procedure is the total processing time of the product.

### 3 Strategy design

The strategy design in this paper is based on the scheduling results of MISATPS. On this basis, several new product scheduling schemes are generated by using the interchange strategy and the interchange adjustment strategy of multi-device adjacent parallel process, and select the scheduling scheme with the best scheduling result.

The interchange strategy and the interchange adjustment strategy of multi-device adjacent parallel process are introduced, respectively.

#### 3.1 The interchange strategy of multi-device adjacent parallel process

This strategy will look for adjacent parallel processes on each processing device, since the problem studied in this paper is the integrated scheduling problem involving multi-device-process, the parallel process will be divided into two kinds, namely common process and multi-device-process.

Therefore, the following four situations need to be discussed in the adjacent parallel processes:

- (1) Situation 1: Both the pre-tightening and post-tightening processes are common processes

Assume that in the current parallel adjacent processes in the same device, the pre-tightening process is process *A*, and the post-tightening process is process *B*. They are all common processes.

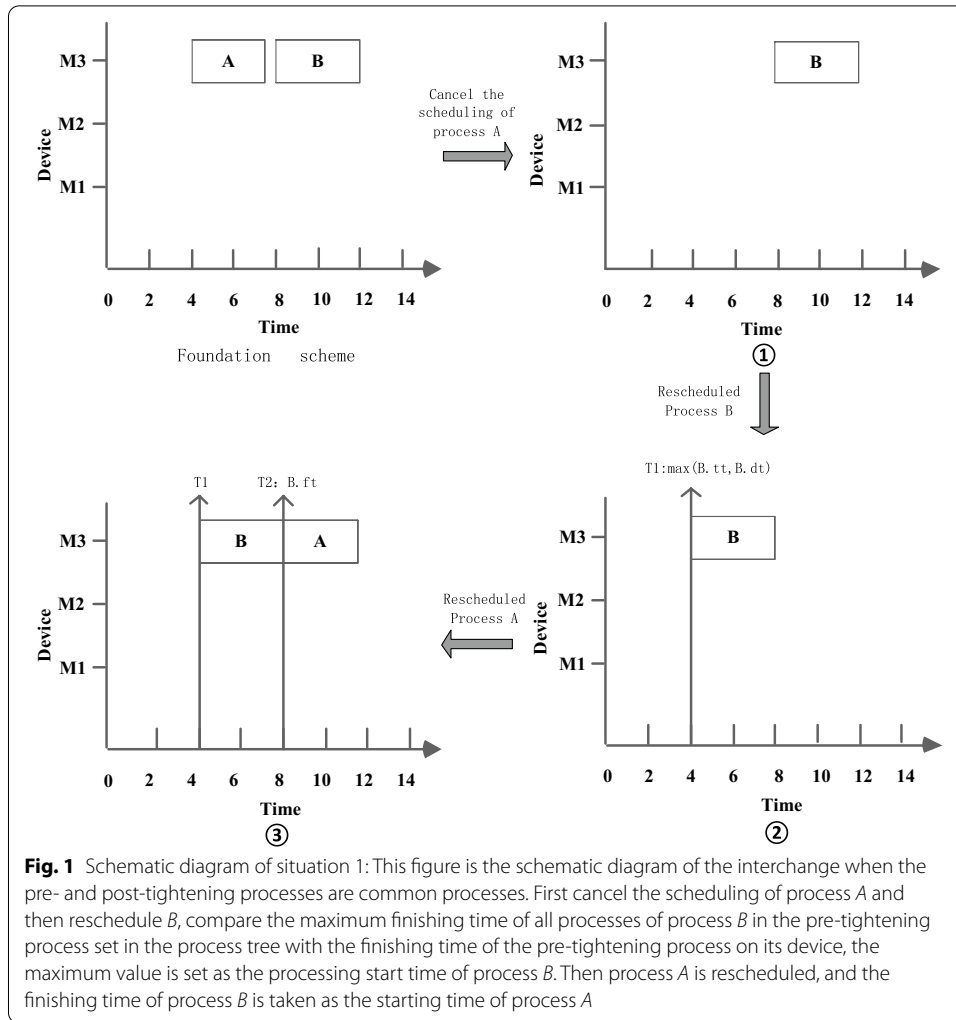
As shown in Fig. 1, first cancel the scheduling of process *A* and then reschedule *B*, compare the maximum finishing time of all processes of process *B* in the pre-tightening process set in the process tree with the finishing time of the pre-tightening process on its device, the maximum value is set as the processing start time of process *B*. Then process *A* is rescheduled, and the finishing time of process *B* is taken as the starting time of process *A*.

- (2) Situation 2: Pre-tightening processes are common processes, and post-tightening processes are multi-device-processes

Assume that in the current parallel adjacent processes in the same device, the pre-tightening process is process *A* which is common process, and the post-tightening process is process *B* which is multi-device-process and contains several parallel sub-processes.

As shown in Fig. 2, first cancel the scheduling of process *A* and then reschedule *B*, compare the maximum finishing time of all processes in the pre-tightening process set of each parallel sub-process included in process *B* in its process tree and the finishing time of the pre-tightening process on its device, respectively. The maximum of all the above times is set as the processing start time of process *B*. Then process *A* is rescheduled, and the finishing time of process *B* is taken as the starting time of process *A*.

- (3) Situation 3: The pre-tightening process is multi-device process, and the post-tightening process is common processes

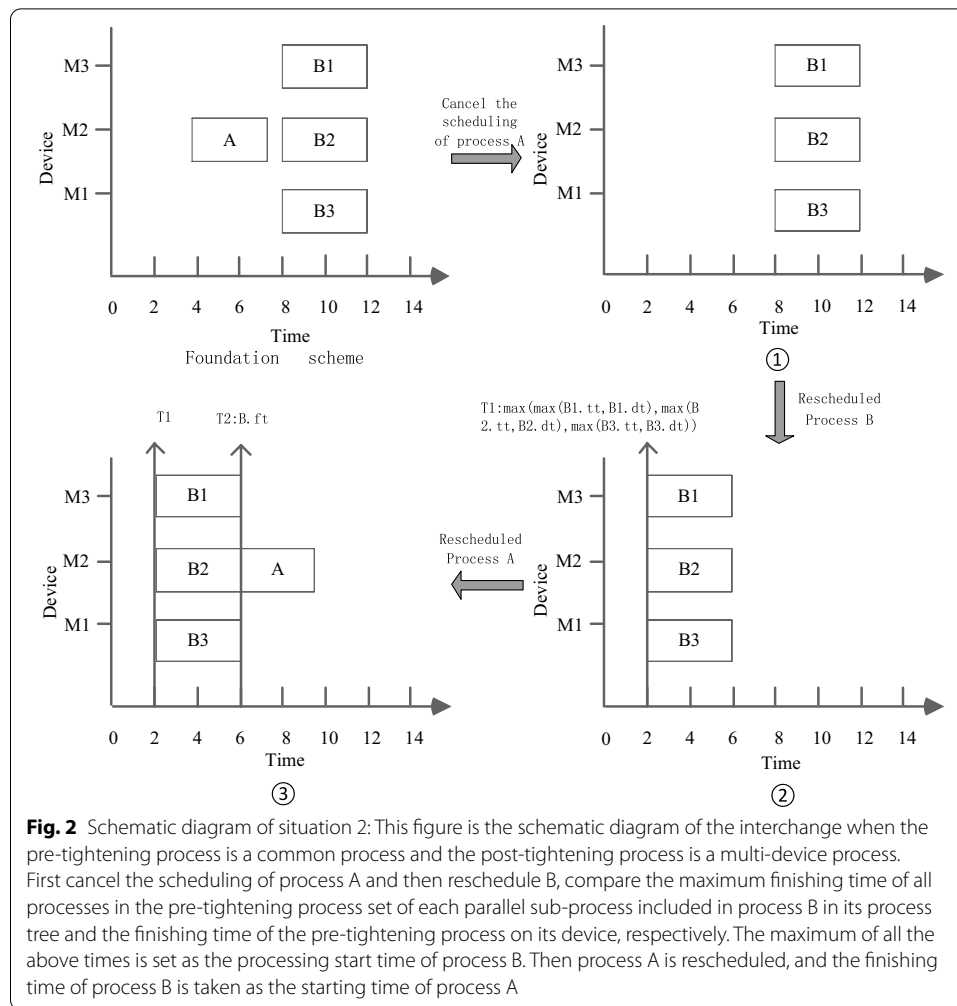


Assume that in the current parallel adjacent processes in the same device, the pre-tightening process is process B, it is multi-device-process and contains several parallel sub-processes, the post-tightening process is process A, it is common process.

As shown in Fig. 3, first of all, cancel the scheduling of all parallel sub-processes contained in process B. Then process A is rescheduled, compare the maximum finishing time of all processes in the pre-tightening process set of process A in the processing process tree with the finishing time of the pre-tightening process set of process A on its processing device.

The maximum of all the above times is set as the processing start time of process A. Then process B is rescheduled, the finishing time of process A is, respectively, taken as the starting time of each parallel sub-process in process B.

(4) Situation 4: Both the pre-tightening and post-tightening processes are multi-device-processes



Assume that in the current parallel adjacent processes in the same device, the pre-tightening process is process B, and the post-tightening process is process A. They are all multi-device-processes and contain several parallel sub-processes.

As shown in Fig. 4, cancel the scheduling of all parallel sub-processes contained in process B. Then process A is rescheduled, compare the maximum finishing time of all processes in the pre-tightening process set of each parallel sub-process included in process A in its process tree and the finishing time of the pre-tightening process on its device, respectively. The maximum of all the above times is set as the processing start time of process A. Then process B is rescheduled, the finishing time of process A is, respectively, taken as the starting time of each parallel sub-process in process B.

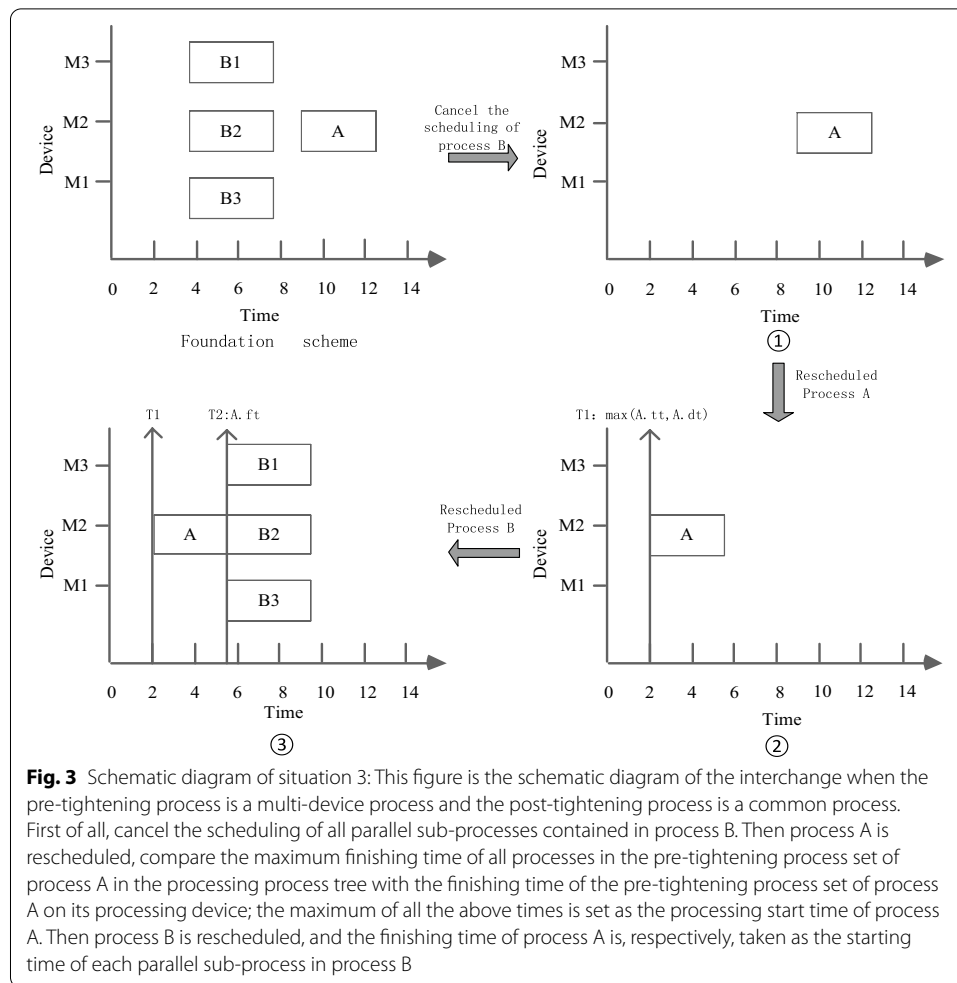
The steps of this strategy are as follows:

Step 1. Assume that in the current parallel adjacent processes in the same device, the pre-tightening process is process A, and the post-tightening process is process B;

Step 2. Determine whether the pre-tightening process A is a common process; if true, go to step 3, otherwise go to step 4;

Step 3. Cancel the scheduling of process A, then go to Step 5;

Step 4. Cancels the scheduling of all parallel sub-processes contained in process A;



**Fig. 3** Schematic diagram of situation 3: This figure is the schematic diagram of the interchange when the pre-tightening process is a multi-device process and the post-tightening process is a common process. First of all, cancel the scheduling of all parallel sub-processes contained in process B. Then process A is rescheduled, compare the maximum finishing time of all processes in the pre-tightening process set of process A in the processing process tree with the finishing time of the pre-tightening process set of process A on its processing device; the maximum of all the above times is set as the processing start time of process A. Then process B is rescheduled, and the finishing time of process A is, respectively, taken as the starting time of each parallel sub-process in process B

Step 5. Determine whether the post-tightening process  $B$  is a common process, go to step 6, otherwise go to step 7;

Step 6. Rescheduling Process  $B$ , compare the maximum finishing time of all processes in the pre-tightening process set of process  $B$  in the process tree with the finishing time of the pre-tightening process set of process  $B$  on its device, The maximum of all the above times is set as the processing start time of process  $B$ ;

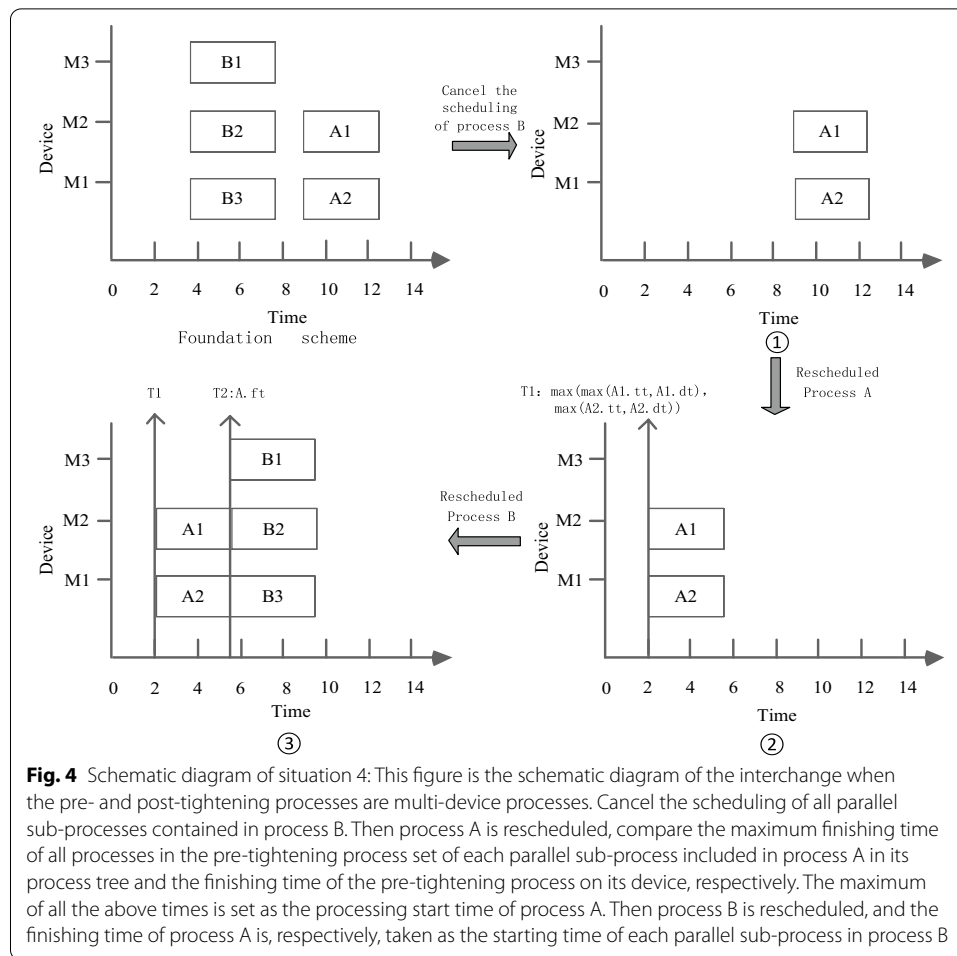
Step 7. Rescheduling Process  $B$ , compare the maximum finishing time of all processes in the pre-process set of process  $B$  in the process tree with the finishing time of the pre-tightening process set of each parallel sub-process included in the process  $B$  on their device, the maximum of all the above times is set as the processing start time of process  $B$ ;

Step 8. Determine whether the pre-tightening process  $A$  is a common process; if true, go to step 9, otherwise go to step 10;

Step 9. Take the finishing time of process  $B$  as the starting time of process  $A$ , go to step 11;

Step 10. The finishing time of process  $B$  is, respectively, taken as the starting time of each parallel sub-process in process  $A$ ;

Step 11. End.



The implementation process of Multi-device adjacent parallel process interchange strategy is shown in Fig. 5.

### 3.2 Multi-device adjacent parallel process interchange adjustment strategy

According to the analysis in Sect. 3.1, the interchange process is divided into two kinds: common process and multi-device process.

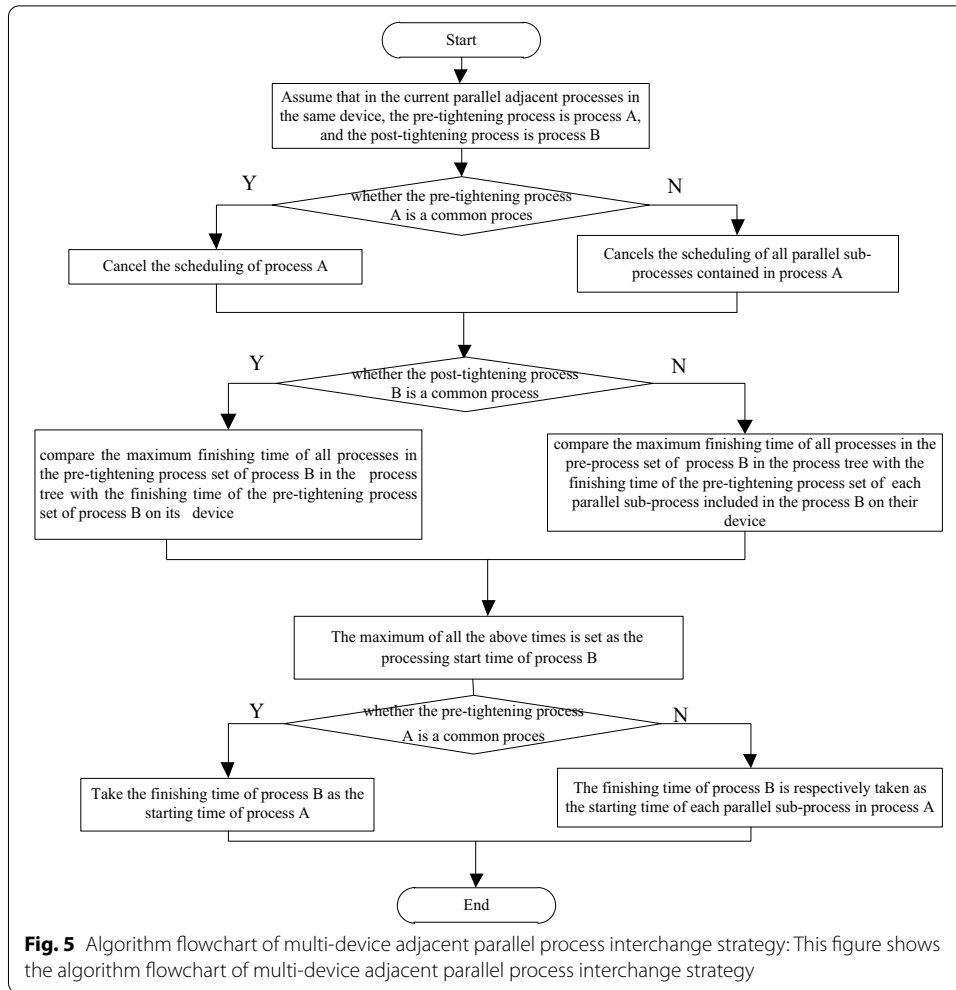
In the interchange process, the post-tightint process  $P$  will be changed to the front, the processing start time is earlier,

Therefore, the scheduling of post-tightening process  $P$  may advance the processing time of post-tightening process in its process tree and post-tightening process of device.

At the same time, the tight pre-process  $F$  will shift to the back and the he processing start time will be delayed, which may affect other processes. The influence mentioned here means that the rescheduling of the previous process will result in overlapping time with the scheduling of the subsequent processes and making the processing scheme unfeasible. It is necessary to adjust the affected subsequent processes.

Rescheduling the post-tightint process  $P$  can be divided into two situations:

- (1) Process  $P$  is common process



If the adjustment process is a common process, the processes affected are divided into two categories: One is the post-tightening process in the process tree; the other is the post-tightening process in the processing device. It is necessary to check whether the processing time of these two types of processes needs to be advanced.

(2) Process  $P$  is multi-device-process

If the process is adjusted to a multi-device process, the number of parallel sub-processes contained in the process is first determined, then check the post-tightening process in the process tree and the post-tightening process of processing device of each parallel sub-process, determine whether the processing time of these two kinds of processes needs to be advanced.

The methods to advance the processing time of the above processes are as follows:

(1) For the common process



Starting from the rescheduling end time of process  $P$ , the first available time point was found to be the starting time of the post-tightening of process  $P$  in the process tree, taking the end time of process  $P$  rescheduling as the starting time of the post-tightening of process  $P$  in its device.

(2) For the multi-device-process

Starting from the rescheduling end time of process  $P$ , the first available time point on each device is found as the starting time of the post-tightening process in process tree of process  $P$ . The end time of process  $P$  rescheduling is taken as the start time of post-tightening process on the device.

Rescheduling the pre-tightint process  $F$  can be divided into two situations:

(1) Process  $F$  is common process

If the adjustment process is a common process, there are two kinds of processes affected. One is the post-tightening process in the process tree; the other is the post-tightening process in the processing device. The two kinds of processes need to be inspected, respectively, to determine whether they are affected.

(2) Process  $F$  is multi-device-process

When the adjustment process is a multi-device process, the parallel sub processes should be determined first. Then, check the next process of  $F$  in the process tree and the post-process of each parallel sub process in its processing device to determine whether it is affected. If it is affected, reschedule it.

The above-mentioned processes involved in the adjustment are divided into two situations:

(1) For the common process

For the post-processing process in the process tree of  $F$ , set its processing start time as the maximum of the processing end time of the pre-tightening process on the device and the processing end time of process  $F$  in the current scheme, for the post-tightening process on the device, its processing start time is set as the processing end time of the process  $F$ .

(2) For the multi-device-process

Due to the multi-device-process includes several parallel sub-processes, adjust the post-tightening process of process  $F$  in the process tree to ensure that the starting time of each sub-process is consistent. Calculate the finishing time of process  $F$  and the finishing time of the pre-processing process of each sub-process on its processing device in the current scheme, respectively. Then, the maximum value is set as the processing start time of the multi-device process. For the post-tightening process on

the device, the finishing time of process  $F$  is set as the starting time of each parallel sub-process contained in it.

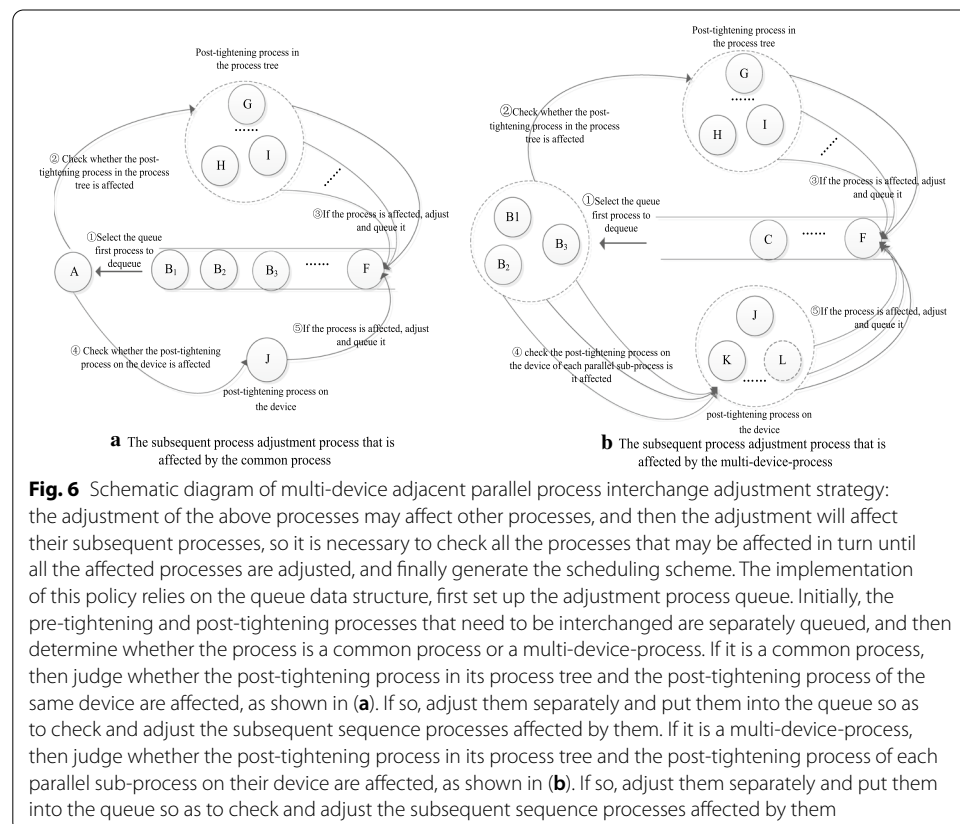
In addition, the adjustment of the above processes may affect other processes, and then the adjustment will affect their subsequent processes. It is necessary to check all the processes that may be affected in turn until all the affected processes are adjusted, and finally generate the scheduling scheme.

The implementation of this policy relies on the queue data structure, first set up the adjustment process queue. Initially, the pre-tightening and post-tightening processes that need to be interchanged are separately queued, and then determine whether the process is a common process or a multi-device-process. If it is a common process, then judge whether the post-tightening process in its process tree and the post-tightening process of the same device are affected, as shown in Fig. 6a. If so, adjust them separately and put them into the queue so as to check and adjust the subsequent sequence processes affected by them;

If it is a multi-device-process, then judge whether the post-tightening process in its process tree and the post-tightening process of each parallel sub-process on their device are affected, as shown in Fig. 6b. If so, adjust them separately and put them into the queue so as to check and adjust the subsequent sequence processes affected by them.

The steps of this strategy are as follows:

Step 1. Adjacent interchange processes on the same device are, respectively, added to the adjustment queue;



- Step 2. The adjustment queue is queued and the result is stored in  $P$ ;
- Step 3. Judge whether  $P$  is empty; if not, go to step 4, otherwise go to step 32;
- Step 4. Determine whether  $P$  is a common process; if it is, go to step 5; otherwise, go to step 18;
- Step 5. Judge whether  $P$  is processed in advance; if it goes to step 6, otherwise, go to step 12;
- Step 6. Look for the post-tightening process  $FT$  of  $P$  in the process tree; if any, go to step 7, otherwise go to step 9;
- Step 7. Taking the end time of process  $P$  rescheduling as the starting point, the first available time point was found as the starting time of process  $FT$ ;
- Step 8. Add process  $FT$  into the adjustment queue; if it is a multi-device-process, then add each parallel sub-procedure into the adjustment queue in turn;
- Step 9. Look for the post-tightening process  $TD$  of  $P$  on the device; if any, go to step 10, otherwise go to step 12;
- Step 10. Taking the end time of process  $P$  rescheduling as the starting point, the first available time point was found as the starting time of process  $TD$ ;
- Step 11. Add process  $TD$  into the adjustment queue; if it is a multi-device-process, then add each parallel sub-process into the adjustment queue in turn, go to step 2;
- Step 12. Look for the post-tightening process  $AT$  of  $P$  in the process tree; if any, go to step 13, otherwise go to step 15;
- Step 13. The maximum value of the finishing time of pre-tightening process of  $AT$  and the finishing time of process  $P$  is set as the starting time of process  $AT$ ;
- Step 14. Add process  $AT$  into the adjustment queue; if it is a multi-device-process, then add each parallel sub-process into the adjustment queue in turn;
- Step 15. Look for the post-tightening process  $AD$  of  $P$  on the device; if any, go to step 16, otherwise go to step 18;
- Step 16. The finishing time of the current process is set as the processing start time of  $AD$  process;
- Step 17. Add process  $AD$  into the adjustment queue; if it is a multi-device-process, then add each parallel sub-process into the adjustment queue in turn, go to step 2;
- Step 18. Judge whether  $P$  is processed in advance; if it is, go to step 19; otherwise, go to step 25;
- Step 19. Look for the post-tightening process  $MFT$  of  $P$  in the process tree; if any, go to step 20, otherwise go to step 22;
- Step 20. Taking the end time of process  $P$  rescheduling as the starting point, the first available time point was found as the starting time of process  $MFT$ ;
- Step 21. Add process  $MFT$  into the adjustment queue; if it is a multi-device-process, then add each parallel sub-process into the adjustment queue in turn;
- Step 22. Look for the post-tightening processes on the device of each parallel sub-process included in the process  $P$ , and store it in  $MFD[]$ ; if any, go to step 23, otherwise go to step 25;
- Step 23. The finishing time of each parallel sub-process in procedure  $P$  is taken as the starting time of each process in its corresponding process  $MFD[]$ ;

Step 24. Add each process in  $MFD[]$  to the adjustment queue; if one of the process is a multi-device procedure, it is necessary to add each parallel sub-process into the adjustment queue successively, go to step 2;

Step 25. Look for the post-tightening process  $MAT$  of  $P$  in the process tree; if any, go to step 26, otherwise go to step 28;

Step 26. Calculate the finishing time of the pre-processing process on the device and the finishing time of the pre-processing process in the process tree of each parallel sub-process, respectively. Then, the maximum value is set as the start time of  $MAT$ ;

Step 27. Add process  $MAT$  into the adjustment queue; if it is a multi-device-process, then add each parallel sub-process into the adjustment queue in turn;

Step 28. Look for the post-tightening processes on the device of each parallel sub-process included in the process  $P$ , and store it in  $MAD[]$ ; if any, go to step 29, otherwise go to step 31;

Step 29. The finishing time of each parallel sub-process in procedure  $P$  is taken as the starting time of each process in its corresponding process  $MAD[]$ ;

Step 30. Add each process in  $MAD[]$  to the adjustment queue; if one of the process is a multi-device procedure, it is necessary to add each parallel sub-process into the adjustment queue successively;

Step 31. Go to step 2;

Step 32. End the adjustment, calculate the total processing time of the current scheme, exit.

The implementation process of multi-device adjacent parallel process interchange adjustment strategy is shown in Fig. 7.

#### 4 Algorithm design

The steps of the proposed algorithm are as follows:

Step 1. The basic scheduling scheme is developed by using multi-devices-process integrated scheduling algorithm with time-selective strategy for process sequence;

Step 2. Assume that the number of processing device is  $M$ ;

Step 3.  $k = 1$ ;

Step 4. Determine  $k \leq M$ ; if it is established, go to step 5; if it is not established, go to step 14;

Step 5. Assume that the number of device  $k$  is  $Nk$ ;

Step 6.  $i = 1$ ;

Step 7. Determine  $i < Nk$ ; if it is established, go to step 8; if it is not established, go to step 13;

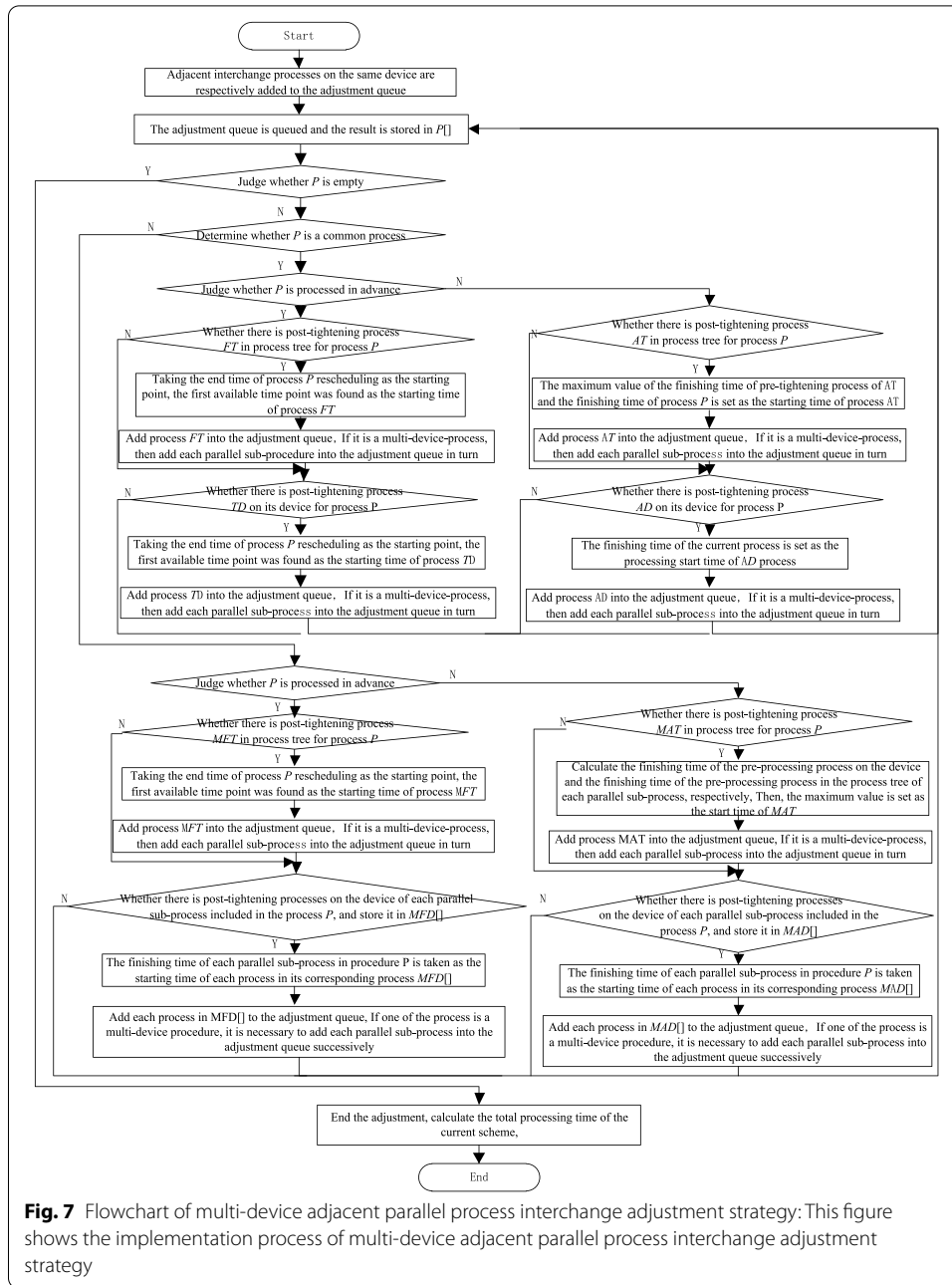
Step 8. Determine whether the  $i$  process and  $i + 1$  process are parallel processes; if so, go to 9, not go to step 12;

Step 9. Start the multi-device adjacent parallel process interchange strategy, and interchange the  $i$  process with the  $i + 1$  process;

Step 10. Start the multi-device adjacent parallel process interchange adjustment strategy, and adjust the process affected by the step 9;

Step 11. Generate a new product scheduling scheme and add it to the product scheduling set;

Step 12.  $i++$ , go to step 7;



**Fig. 7** Flowchart of multi-device adjacent parallel process interchange adjustment strategy: This figure shows the implementation process of multi-device adjacent parallel process interchange adjustment strategy

Step 13.  $k++$ , go to step 4;

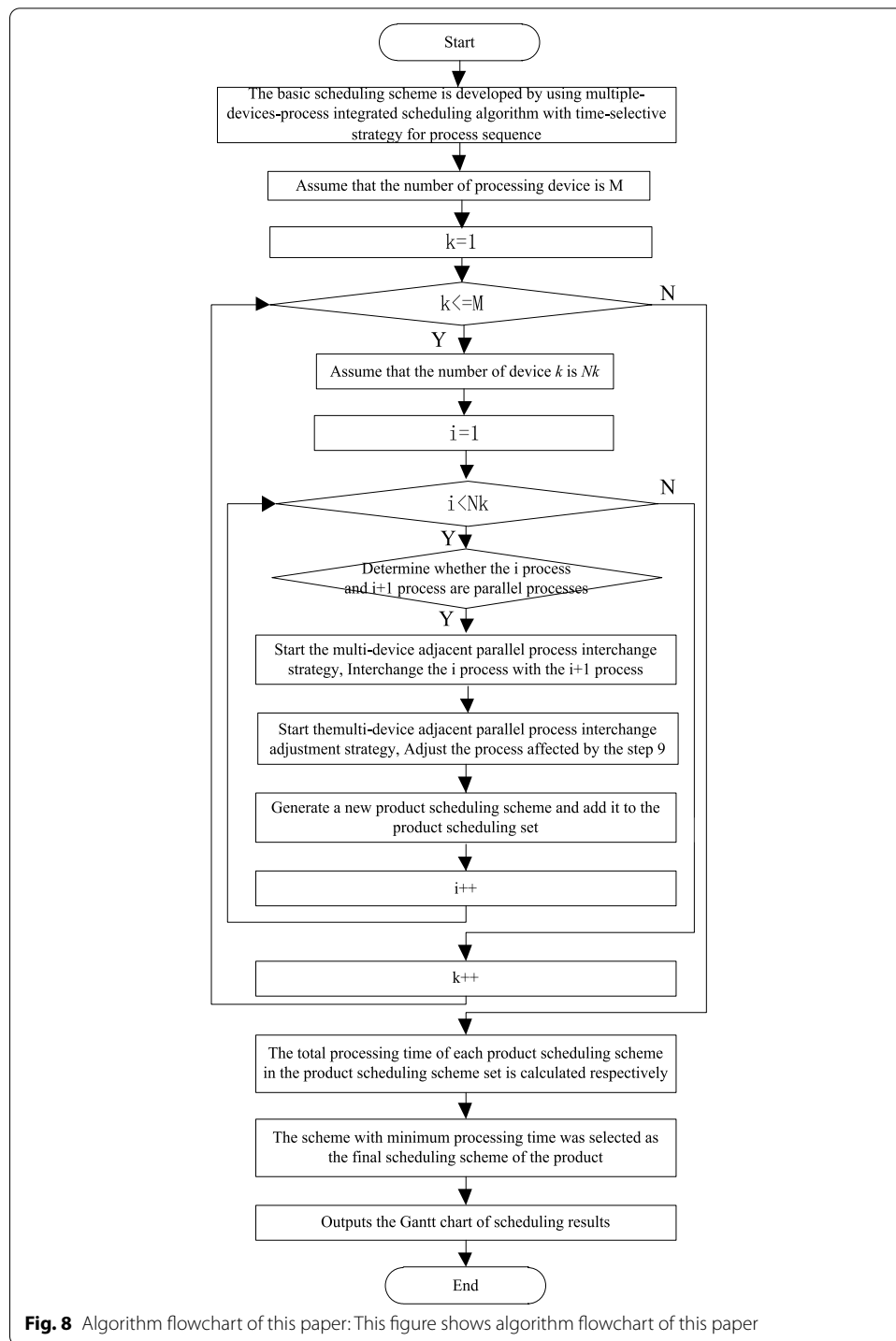
Step 14. The total processing time of each product scheduling scheme in the product scheduling scheme set is calculated, respectively;

Step 15. The scheme with minimum processing time was selected as the final scheduling scheme of the product;

Step 16. Outputs the Gantt chart of scheduling results;

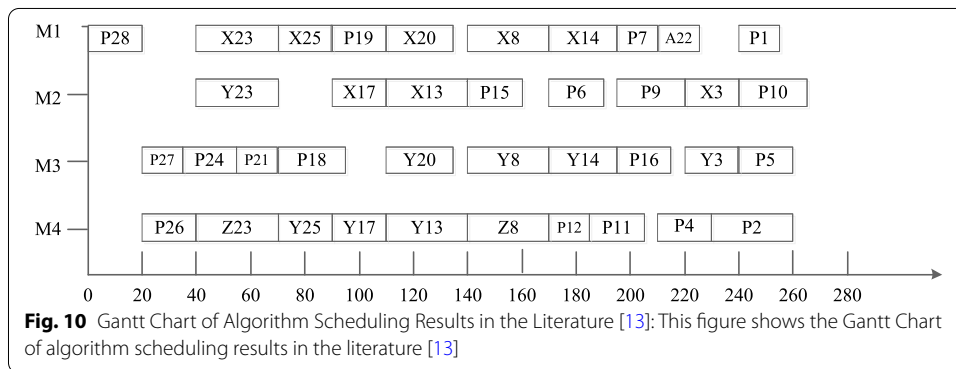
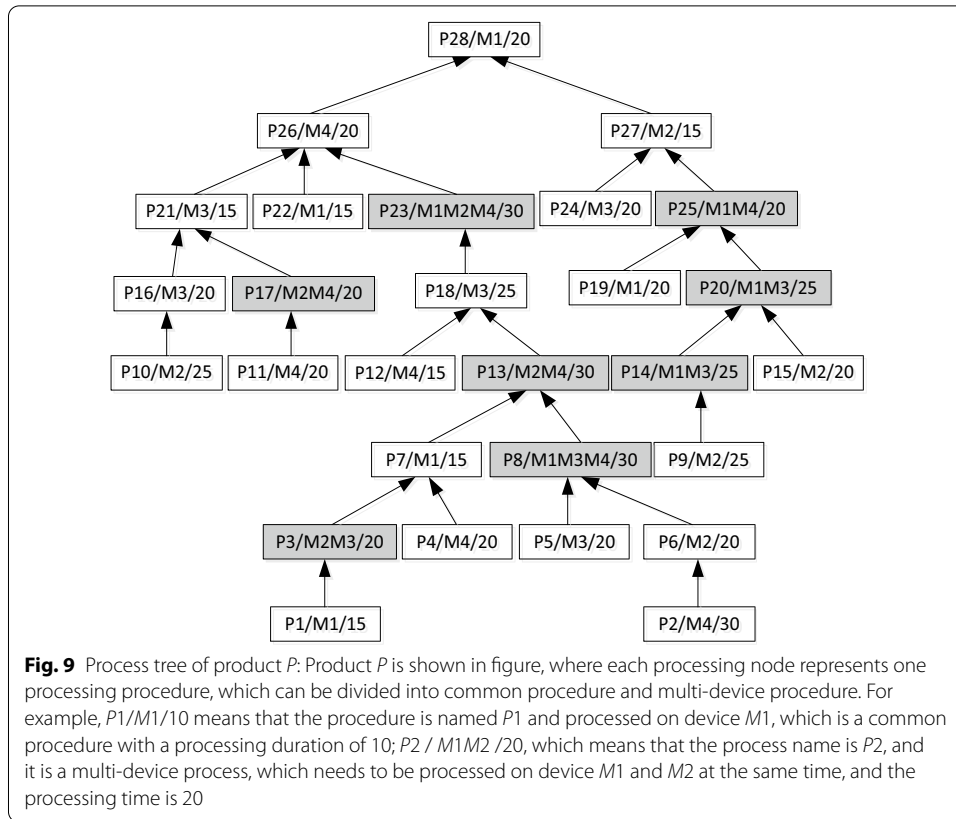
Step 17. End.

The implementation process of the proposed algorithm is shown in Fig. 8.



## 5 The case analysis

In order to facilitate readers to understand the algorithm, the following through the example analysis. Product  $P$  is shown in Fig. 9, where each processing node represents one processing procedure, which can be divided into common procedure and



multi-device procedure. For example,  $P1/M1/10$  means that the procedure is named  $P1$  and processed on device  $M1$ , which is a common procedure with a processing duration of 10;  $P2/M1M2/20$ , which means that the process name is  $P2$ , and it is a multi-device process, which needs to be processed on device  $M1$  and  $M2$  at the same time, and the processing time is 20. According to the algorithm proposed in the literature [13], the total processing time of product  $P$  is 265, and its scheduling Gantt chart is shown in Fig. 10. Meanwhile, according to the algorithm proposed in this paper, the scheduling process of product  $P$  is shown in Table 1, and the total processing time of product  $P$  is 260, and the final product  $P$  scheduling Gantt chart is shown in Fig. 11.

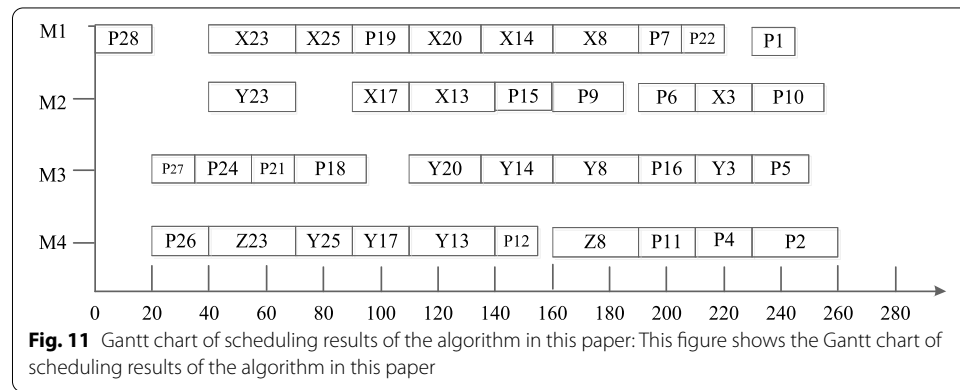
**Table 1** The scheduling process of product *P* scheduled by the proposed algorithm

Device id	Exchange process	Each process scheduling time point in the current scheme	Total time
M1	P23, P25	P28:0,P26:20,P23:60,P18:90,P13:115,P8:145,P6:175,P2:235,P27:20,P25:40,P20:115,P14:175,P9:200,P21:55,P16:200,P10:245,P7:200,P3:225,P1:245,P17:90,P11:190,P4:215,P24:35,P19:90,P15:145,P5:245,P22:215,P12:175	270
M1	P19, P20	P28:0,P26:20,P23:40,P18:70,P13:110,P8:140,P6:170,P2:230,P27:20,P25:70,P20:95,P14:170,P9:195,P21:55,P16:195,P10:240,P7:195,P3:220,P1:240,P17:90,P11:185,P4:210,P24:35,P19:120,P15:140,P5:240,P22:210,P12:170	265
M1	P20, P8	P28:0,P26:20,P23:40,P18:70,P13:110,P8:140,P6:170,P2:255,P27:20,P25:70,P20:170,P14:195,P9:220,P21:55,P16:220,P10:265,P7:220,P3:245,P1:265,P17:90,P11:185,P4:235,P24:35,P19:90,P15:140,P5:265,P22:235,P12:170	290
M1	P14, P8	P28:0,P26:20,P23:40,P18:70,P13:110,P8:160,P6:190,P2:230,P27:20,P25:70,P20:110,P14:135,P9:160,P21:55,P16:190,P10:230,P7:190,P3:210,P1:230,P17:90,P11:190,P4:210,P24:35,P19:90,P15:140,P5:230,P22:205,P12:140	260
M1	P14, P7	P28:0,P26:20,P23:40,P18:70,P13:110,P8:140,P6:170,P2:230,P27:20,P25:70,P20:110,P14:185,P9:210,P21:55,P16:210,P10:255,P7:170,P3:235,P1:255,P17:90,P11:185,P4:210,P24:35,P19:90,P15:140,P5:255,P22:210,P12:170	280
M1	P7, P22	P28:0,P26:20,P23:40,P18:70,P13:110,P8:140,P6:170,P2:235,P27:20,P25:70,P20:110,P14:170,P9:195,P21:55,P16:195,P10:245,P7:210,P3:225,P1:245,P17:90,P11:185,P4:210,P24:35,P19:90,P15:140,P5:245,P22:195,P12:170	270
M1	P22, P1	P28:0,P26:20,P23:40,P18:70,P13:110,P8:140,P6:170,P2:230,P27:20,P25:70,P20:110,P14:170,P9:195,P21:55,P16:195,P10:240,P7:195,P3:220,P1:240,P17:90,P11:185,P4:210,P24:35,P19:90,P15:140,P5:240,P22:255,P12:170	270
M2	P23, P17	P28:0,P26:20,P23:90,P18:120,P13:145,P8:185,P6:215,P2:275,P27:20,P25:120,P20:160,P14:215,P9:240,P21:55,P16:240,P10:285,P7:240,P3:265,P1:285,P17:70,P11:230,P4:255,P24:35,P19:140,P15:185,P5:285,P22:255,P12:215	310
M2	P17, P13	P28:0,P26:20,P23:40,P18:70,P13:95,P8:145,P6:175,P2:235,P27:20,P25:70,P20:110,P14:175,P9:200,P21:55,P16:200,P10:245,P7:200,P3:225,P1:245,P17:125,P11:190,P4:215,P24:35,P19:90,P15:145,P5:245,P22:215,P12:175	270
M2	P13, P15	P28:0,P26:20,P23:40,P18:70,P13:155,P8:185,P6:215,P2:230,P27:20,P25:70,P20:110,P14:135,P9:185,P21:55,P16:160,P10:255,P7:215,P3:235,P1:255,P17:90,P11:125,P4:230,P24:35,P19:90,P15:135,P5:215,P22:230,P12:110	280
M2	P15, P6	P28:0,P26:20,P23:40,P18:70,P13:110,P8:140,P6:170,P2:230,P27:20,P25:70,P20:110,P14:170,P9:210,P21:55,P16:195,P10:255,P7:195,P3:235,P1:255,P17:90,P11:185,P4:210,P24:35,P19:90,P15:190,P5:255,P22:210,P12:170	280
M2	P9, P6	P28:0,P26:20,P23:40,P18:70,P13:110,P8:140,P6:220,P2:240,P27:20,P25:70,P20:110,P14:170,P9:195,P21:55,P16:195,P10:260,P7:195,P3:240,P1:260,P17:90,P11:185,P4:210,P24:35,P19:90,P15:140,P5:260,P22:210,P12:170	285
M2	P9, P3	P28:0,P26:20,P23:40,P18:70,P13:110,P8:140,P6:170,P2:230,P27:20,P25:70,P20:110,P14:170,P9:235,P21:55,P16:195,P10:260,P7:195,P3:215,P1:235,P17:90,P11:185,P4:210,P24:35,P19:90,P15:140,P5:235,P22:210,P12:170	285
M2	P3, P10	P28:0,P26:20,P23:40,P18:70,P13:110,P8:140,P6:170,P2:230,P27:20,P25:70,P20:110,P14:170,P9:195,P21:55,P16:195,P10:220,P7:195,P3:245,P1:265,P17:90,P11:185,P4:210,P24:35,P19:90,P15:140,P5:215,P22:210,P12:170	280
M3	P24, P21	P28:0,P26:20,P23:40,P18:70,P13:110,P8:140,P6:170,P2:230,P27:20,P25:70,P20:110,P14:170,P9:195,P21:35,P16:195,P10:240,P7:195,P3:220,P1:240,P17:90,P11:185,P4:210,P24:50,P19:90,P15:140,P5:240,P22:210,P12:170	265
M3	P21, P18	P28:0,P26:20,P23:40,P18:70,P13:130,P8:160,P6:190,P2:250,P27:20,P25:70,P20:110,P14:190,P9:195,P21:95,P16:135,P10:260,P7:215,P3:240,P1:260,P17:110,P11:205,P4:230,P24:35,P19:90,P15:160,P5:215,P22:230,P12:190	285
M3	P18, P20	P28:0,P26:20,P23:40,P18:115,P13:140,P8:170,P6:200,P2:255,P27:20,P25:70,P20:90,P14:140,P9:220,P21:55,P16:70,P10:110,P7:200,P3:245,P1:265,P17:90,P11:215,P4:235,P24:35,P19:115,P15:170,P5:200,P22:215,P12:200	285
M3	P20, P8	same as M1	290
M3	P8, P14	same as M1	260
M3	P14, P16	P28:0,P26:20,P23:40,P18:70,P13:110,P8:140,P6:170,P2:250,P27:20,P25:70,P20:110,P14:190,P9:215,P21:55,P16:170,P10:260,P7:215,P3:240,P1:260,P17:90,P11:185,P4:230,P24:35,P19:90,P15:140,P5:215,P22:230,P12:170	285



**Table 1** (continued)

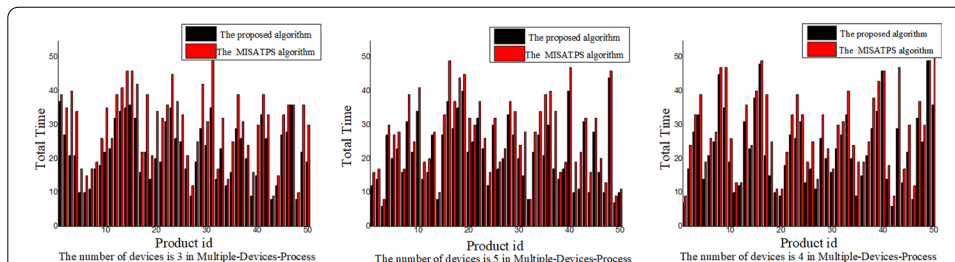
Device id	Exchange process	Each process scheduling time point in the current scheme	Total time
M3	P16, P3	P28:0,P26:20,P23:40,P18:70,P13:110,P8:140,P6:170,P2:230,P27:20,P25:70,P20:110,P14:170,P9:230,P21:55,P16:230,P10:255,P7:195,P3:210,P1:230,P17:90,P11:185,P4:210,P24:35,P19:90,P15:140,P5:250,P22:210,P12:170	280
M3	P3, P5	P28:0,P26:20,P23:40,P18:70,P13:110,P8:140,P6:170,P2:230,P27:20,P25:70,P20:110,P14:170,P9:195,P21:55,P16:195,P10:255,P7:195,P3:235,P1:255,P17:90,P11:185,P4:210,P24:35,P19:90,P15:140,P5:215,P22:210,P12:170	280
M3	P23, P25	same as M1	270
M4	P25, P17	P28:0,P26:20,P23:40,P18:70,P13:110,P8:155,P6:185,P2:245,P27:20,P25:90,P20:130,P14:185,P9:210,P21:55,P16:210,P10:255,P7:210,P3:235,P1:255,P17:70,P11:200,P4:225,P24:35,P19:110,P15:155,P5:255,P22:225,P12:185	280
M4	P17, P13	same as M2	270
M4	P8, P12	P28:0,P26:20,P23:40,P18:70,P13:110,P8:155,P6:185,P2:245,P27:20,P25:70,P20:110,P14:185,P9:210,P21:55,P16:210,P10:255,P7:210,P3:235,P1:255,P17:90,P11:185,P4:225,P24:35,P19:90,P15:140,P5:255,P22:225,P12:140	280
M4	P12, P11	P28:0,P26:20,P23:40,P18:70,P13:110,P8:140,P6:170,P2:230,P27:20,P25:70,P20:110,P14:170,P9:195,P21:55,P16:195,P10:240,P7:195,P3:220,P1:240,P17:90,P11:170,P4:210,P24:35,P19:90,P15:140,P5:240,P22:210,P12:190	265
M4	P11, P4	P28:0,P26:20,P23:40,P18:70,P13:110,P8:140,P6:170,P2:250,P27:20,P25:70,P20:110,P14:170,P9:195,P21:55,P16:195,P10:240,P7:195,P3:220,P1:240,P17:90,P11:230,P4:210,P24:35,P19:90,P15:140,P5:240,P22:210,P12:170	280
M4	P4, P2	P28:0,P26:20,P23:40,P18:70,P13:110,P8:140,P6:170,P2:205,P27:20,P25:70,P20:110,P14:170,P9:195,P21:55,P16:195,P10:240,P7:195,P3:220,P1:240,P17:90,P11:185,P4:235,P24:35,P19:90,P15:140,P5:240,P22:210,P12:170	265



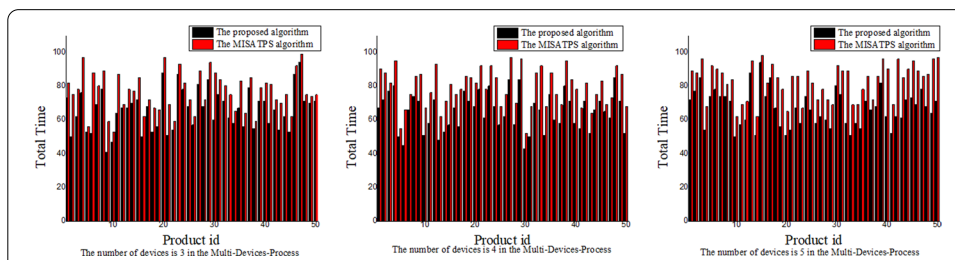
## 6 Experimental

In order to verify the performance of the algorithm, a group of experiments are designed for comparison. The algorithm performance is affected by the structure of product process tree, the number of processing equipment and the processing time. Therefore, the proposed algorithm is compared with the MISATPS algorithm [13] from the perspective of different scale parameters. First, each experiment will randomly generate 50 products. The parameters in the process tree are randomly generated. The parameters are: the structure of the process tree (including: the total number of layers of the process tree, the number of processes in each layer, the relationship between the front and back of the process), the processing time and equipment number of the process in the process tree, and the total number of processing equipment of the process. Each group of

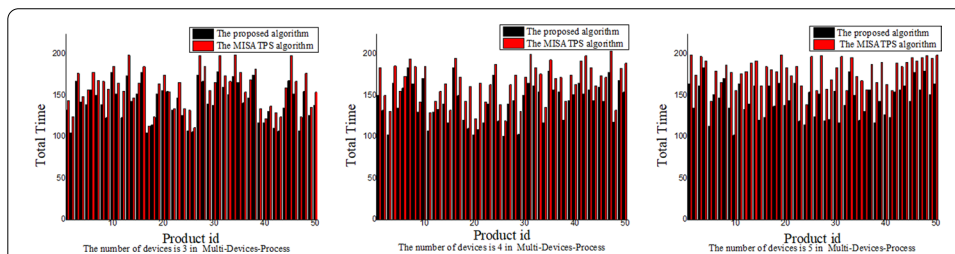
experiments will randomly generate 50 products, because the product structure is random, which can effectively prove the effectiveness of the algorithm in different cases. Five groups of experiments were designed as follows: Four groups of experiments were designed as follows: As shown in Fig. 12, the randomly generated data in Experiment 1 were that the number of processes was less than 20, the total processing time of the workpiece was less than 50 and the number of devices is 3, 4 and 5 in Multi-Devices-Process, respectively. As shown in Fig. 13, the randomly generated data in Experiment 2 were that the number of processes was between 20 and 30, the total processing time of the workpiece was between 50 and 100 and the number of devices is 3, 4 and 5 in Multi-Devices-Process, respectively. As shown in Fig. 14, the randomly generated data



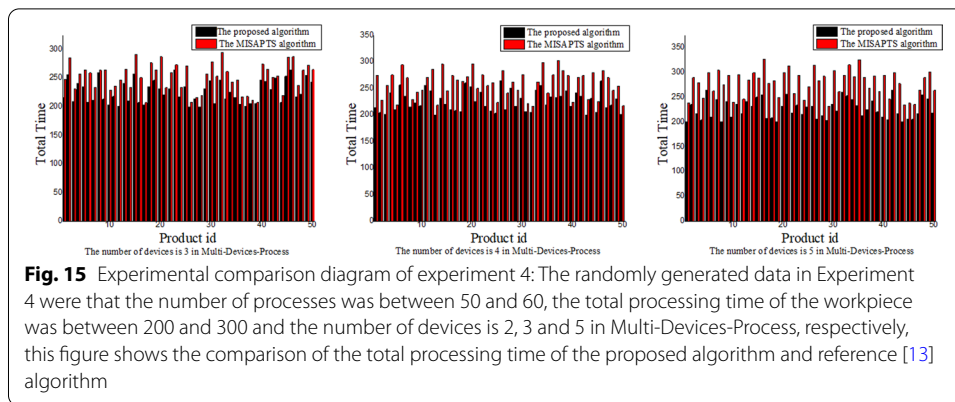
**Fig. 12** Experimental comparison diagram of experiment 1: The randomly generated data in Experiment 1 were that the number of processes was less than 20, the total processing time of the workpiece was less than 50 and the number of devices is 2, 3 and 5 in Multi-Devices-Process, respectively, this figure shows the comparison of the total processing time of the proposed algorithm and reference [13] algorithm



**Fig. 13** Experimental comparison diagram of experiment 2: The randomly generated data in Experiment 2 were that the number of processes was between 20 and 30, the total processing time of the workpiece was between 50 and 100 and the number of devices is 2, 3 and 5 in Multi-Devices-Process, respectively, this figure shows the comparison of the total processing time of the proposed algorithm and reference [13] algorithm



**Fig. 14** Experimental comparison diagram of experiment 3: The randomly generated data in Experiment 3 were that the number of processes was between 30 and 50, the total processing time of the workpiece was between 100 and 200 and the number of devices is 2, 3 and 5 in Multi-Devices-Process, respectively, this figure shows the comparison of the total processing time of the proposed algorithm and reference [13] algorithm



in Experiment 3 were that the number of processes was between 30 and 50, the total processing time of the workpiece was between 100 and 200 and the number of devices is 3, 4 and 5 in Multi-Devices-Process, respectively. As shown in Fig. 15, the randomly generated data in Experiment 4 were that the number of processes was between 50 and 60, the total processing time of the workpiece was between 200 and 300 and the number of devices is 3, 4 and 5 in Multi-Devices-Process, respectively.

## 7 Results and discussion

Through the above experiments, it can be seen that when the number of randomly generated processes in the process tree is in the interval of (1, 20) (20, 30) (30, 50) and (50, 60), the corresponding total processing time of all processes in the process tree is (0, 50), (50, 10), (100, 200) and (200, 300), under the above premise, when the number of device is 3, 4 and 5, The results show that the performance of the proposed algorithm is obviously better than that of MISATPS algorithm, and the proposed algorithm is an effective improvement on the basis of MISATPS algorithm.

## 8 Conclusion

Based on MDPISAWTSFPS algorithm, multi-device adjacent parallel process interchange strategy and multi-device adjacent parallel process interchange adjustment strategy are proposed. They improve the compactness of serial processes and the parallelism of parallel processes, and optimize the scheduling results. It may be the next step to apply the proposed algorithm to different integrated scheduling fields, such as batch scheduling problem, distributed scheduling problem and flexible scheduling problem.

### Acknowledgements

The authors acknowledged the anonymous reviewers and editors for their efforts in valuable comments and suggestions.

### Authors' contributions

ZW proposes the innovation ideas and theoretical analysis, and XZ carries out experiments and data analysis. YC conceived of the study, and participated in its design and coordination and helped to draft the manuscript. All authors read and approved the final manuscript.

### Funding

This work was supported by the Project of Educational Commission of Guangdong (Grant No. 2019KTSCX177), Science and Technology Planning Project of Guangdong (Grant No. 2020A1414010235), Science and Technology Planning

Project of Huizhou (Grant No. 2020SC0306023) and Professorial and Doctoral Scientific Research Foundation of Huizhou University (Grant Nos. 2019JB014, 2018JB007).

#### Availability of data and materials

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

#### Declarations

##### Competing interests

The authors declare that they have no competing interests.

##### Author details

<sup>1</sup>School of Computer Science and Engineering, Huizhou University, Huizhou 516007, People's Republic of China. <sup>2</sup>School of Computer Engineering, Nanjing Institute of Technology, Nanjing 211167, People's Republic of China.

Received: 15 February 2021 Accepted: 14 April 2021

Published online: 23 April 2021

#### References

1. Z. Liu, L. Lang, B. Hu, L. Shi, B. Huang, Y. Zhao, Emission reduction decision of agricultural supply chain considering carbon tax and investment cooperation. *J Clean Prod* **294**(4), 126305 (2021)
2. C. Jiang, R. Li, T. Chen, C. Xu, L. Li, S. Li, A two-lane mixed traffic flow model with drivers' intention to change lane based on cellular automata. *Int J Bio-Inspired Comput* **6**(4), 229–240 (2020)
3. X. Dong, M. Qian, R. Jiang, Packet classification based on decision tree with information entropy. *J Supercomput* **76**(6), 4117–4131 (2020)
4. B. Sya, B. Tla, B. Bwa, A co-evolutionary genetic algorithm for the two-machine flow shop group scheduling problem with job-related blocking and transportation times. *Expert SystAppl* **152**, 113360 (2020)
5. W. Huang, L. Gao, A time wave neural network framework for solving time-dependent project scheduling problems. *IEEE Trans Neural Netw Learn Syst* **31**, 1–10 (2019)
6. H. Shen, S. Huo, H. Yan et al., Distributed dissipative state estimation for Markov jump genetic regulatory networks subject to round-robin scheduling. *IEEE Trans Neural Netw Learn Syst* **99**, 1–10 (2019)
7. C. Swlab, C.Y. Cheng, P. Pourhejazy et al., Multi-temperature simulated annealing for optimizing mixed-blocking permutation flowshop scheduling problems. *Expert SystAppl* **165**, 113837 (2021)
8. I. Mathlouthi, M. Gendreau, J. Potvin, A metaheuristic based on Tabu search for solving a technician routing and scheduling problem. *ComputOper Res* **125**, 105079 (2020)
9. P. Wu, J. Cheng, F. Chu, Large-scale energy-conscious bi-objective single-machine batch scheduling under time-of-use electricity tariffs via effective iterative heuristics. *Ann Oper Res* **296**, 471–494 (2019)
10. D. Anghinolfi, M. Paolucci, R. Ronco, A bi-objective heuristic approach for green identical parallel machine scheduling. *Eur J Oper Res* **289**, 416–434 (2020)
11. Z. Xie, S. Liu, P. Qiao, Dynamic job-shop scheduling algorithm based on ACPM and BFSM. *J Comput Res Dev* **40**, 977–983 (2003)
12. Z. Xie, J. Yang, Y. Zhou et al., Dynamic critical paths multi-product manufacturing scheduling algorithm based on operation set. *Chin J Comput.* **34**, 406–412 (2011)
13. X. Zhang, D. Zhang, Z. Wang, Y. Xin, Multi-devices-process integrated scheduling algorithm with time-selective strategy for process sequence. *Complexity* **2020**, 1–10 (2020)

#### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)

---