

Retraction

Retracted: Multiple-Devices-Process Integrated Scheduling Algorithm with Time-Selective Strategy for Process Sequence

Complexity

Received 23 January 2024; Accepted 23 January 2024; Published 24 January 2024

Copyright © 2024 Complexity. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] X. Zhang, D. Zhang, Z. Wang, and Y. Xin, "Multiple-Devices-Process Integrated Scheduling Algorithm with Time-Selective Strategy for Process Sequence," *Complexity*, vol. 2020, Article ID 8898536, 12 pages, 2020.

Research Article

Multiple-Devices-Process Integrated Scheduling Algorithm with Time-Selective Strategy for Process Sequence

Xiaohuan Zhang ¹, Dan Zhang,¹ Zhen Wang,¹ and Yu Xin²

¹College of Computer Science and Technology, Huizhou University, Huizhou 516007, China

²Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China

Correspondence should be addressed to Xiaohuan Zhang; zhangxiaohuan@hzu.edu.cn

Received 7 August 2020; Revised 6 September 2020; Accepted 3 October 2020; Published 17 October 2020

Academic Editor: Zhihan Lv

Copyright © 2020 Xiaohuan Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper is in view of the current algorithm to solve Multiple-Devices-Process integrated scheduling problems without considering parallel process between the parallel processing. Meanwhile, the influence of the first processing procedure on the second processing procedure is ignored, resulting in poor tightness between serial processes and poor parallelism between parallel processes, ultimately affecting the product scheduling results. We proposed Multiple-Devices-Process integrated scheduling algorithm with time-selective for process sequence. The proposed Multiple-Devices-Process sequencing strategy determines the process of scheduling order and improves the tightness between serial process. This paper presents a method to determine the quasischeduling time point of the multiequipment process, the time-selective strategy of Multiple-Devices-Process, and the time-selective adjustment strategy of Multiple-Devices-Process so that the first and the second processing processes cooperate with each other, and the purpose of improving the tightness of the serial process and the parallelism of the parallel processes is achieved, so as to shorten the product processing time.

1. Introduction

Scheduling, as a key factor affecting the production efficiency of enterprises, has always been a hot issue studied by scholars. Efficient scheduling optimization algorithm can maximize production efficiency and help enterprises achieve higher benefits [1]. At present, there are two main research directions in this field, namely, single processing (assembly) scheduling and integrated scheduling. Typical representatives of the former scheduling are job-shop and flow-shop scheduling [2]. This type of processing and assembly is common in mass production. The reason is the large quantity of the ordered products. The centralized production of disassembling them into work pieces will generate a large amount of inventory, and then the inventory will be assembled. In this way, synchronous processing and assembly can shorten the production cycle.

Currently, there are many research achievements in this direction, and more advanced methods include genetic

algorithm [3], tabu search [4], neural network [5], heuristic algorithm [6], particle swarm optimization [7], bionics algorithm [8], and various hybrid algorithms [9]. With the improvement of people's living standards, today's consumers increasingly pursue personalized products. Personalized product orders are often single small batch orders, which only produce a small amount of inventory or even zero inventories in the production process.

As a result, the above production methods will no longer be dominant but will separate the internal relationship between processing activities and assembly activities. The integrated scheduling problem makes up for this defect and provides practical solutions for single small batch production [10]. The main scheduling idea is that during the scheduling process, the assembly can be carried out as long as the work piece is finished and the assembly conditions are satisfied. The assembly activity is carried out synchronously with the ongoing processing activity, so as to effectively improve the production efficiency and reduce internal

consumption [11]. The most advanced research achievements in this field include the integrated scheduling algorithm for multiple-devices-process.

The content of this paper is the scheduling problem of Multiple-Devices-Process in the integrated scheduling problem. In the actual manufacturing process, some processes need to be processed by multiple equipment. This paper will study this problem. At present, the research on this problem is in its infancy, puts forward this problem, and gives solutions for the first time. In this method, the critical path method and priority scheduling the Multiple-Devices-Process method are adopted to determine the process scheduling sequence, and the first adaptive strategy is applied to determine the scheduling time of the process. The disadvantages are as follows: (1) Paying too much attention to the serial processing of the process, not considering the parallel processing between parallel processes. (2) The influence of the first processing step on the second processing step is ignored, resulting in poor tightness between serial processes and poor parallelism between parallel processes, which ultimately affects the product scheduling result.

In order to solve the above problems, a Multiple-Devices-Process Integrated Scheduling Algorithm with Selecting Time for Process Sequence is proposed (MDOI-SAWSTFPS). The Multiple-Devices-Process sequencing strategy is proposed. It can determine the scheduling sequence of the process and improve the tightness between serial processes. This paper presents a method to determine the quasischeduling time point of the Multiple-Devices-Process, the time-selective strategy of Multiple-Devices-Process, and the time-selective adjustment strategy of Multiple-Devices-Process so that the first and the second processing processes cooperate with each other, and the purpose of improving the tightness of the serial process and the parallelism of the parallel processes is achieved, so as to shorten the product processing time.

2. Problem Description and Analysis

The product processing technology of the multidevices scheduling problem is shown as a tree structure. The node in the tree represents the working procedure of the product, the directed edge represents the partial order relation of the working procedure processing order, the root node represents the last working procedure in the product, and the finished processing of the root node indicates the finished processing of the product. The following constraints must be met for the scheduling problem of Multiple-Devices-Process:

- (1) Each process must strictly comply with the agreed partial order relationship in the processing tree.
- (2) Each equipment can only process one process at any time, and the processing process can not be interrupted.
- (3) There are no devices with the same function in the device set.

- (4) There are multiequipment processes in the process set in which there are several related types of equipment working together. The multidevices process refers to the process that requires multiple equipment to cooperate with each other in a process. The processing time of multiple equipment is the same, and the processing start time and the processing end time are the same.
- (5) A process can only be processed if and only if all the preceding processes are in the finished state (or no prior process).
- (6) The difference between the end time of the latest processing step and the beginning time of the earliest processing step is the total processing time of the product.

3. Strategy Design

3.1. Multiple-Devices-Process Sequencing Strategy. The design of this strategy is similar to that of the common equipment process sequence sequencing strategy. The difference between the two strategies is that the multidevices process sequencing strategy contains multidevices processes and needs to be discussed separately. Since the start time, end time, and processing time of these processes are the same, they can be regarded as the same process, and they are arranged together in no particular order [12]. For example, the product process tree is shown in Figure 1, which contains 7 processes. First, the path length of 3 leaf nodes is calculated, and the results are A2:35, A6:70, and A7:80. Therefore, the process sequence 1 is A1, A3, A5, and A7. Remove the process in the first process sequence from the process tree, and the remaining leaf node is A2, A6. The path length is calculated, respectively, and the results are A2:20 and A6:35. Therefore, the second process sequence is A4, A6. Remove the process in the second process sequence from the process tree, and the remaining leaf node is A2. Since it is the last process sequence, there is no need to calculate its path length, and the upper process of the third process sequence is directly determined as A2.

A simplified version of the proposed algorithm can be described as follows:

Step 1: $i = 0$.

Step 2: calculate the path length of existing leaf nodes in the reverse process tree, respectively.

Step 3: $i++$.

Step 4: select the leaf node W with the longest path.

Step 5: if process W is not unique, select the process Q with the highest number of sequences on its path.

Step 6: if process Q is not unique, select the process O with the smallest difference between the number of layers of the process and the number of layers of the root node in the original processing tree.

Step 7: if process O is not unique, select the process P with the maximum total processing time of all preceding processes of each process on its path.

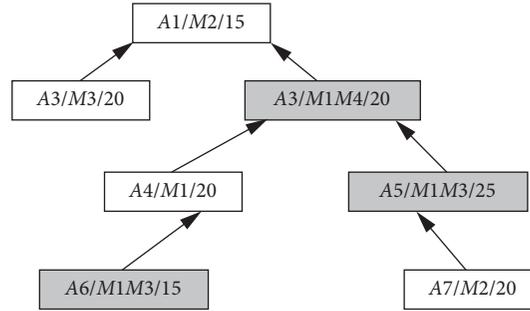


FIGURE 1: A product process tree with Multiple-Devices-Process.

Step 8: the sequence formed by all the processes on the path of process P is denoted as process sequence I .

Step 9: starting from process P , all the processes on process sequence I are successively pushed into S , followed by bomb stack S , and the resulting processes are successively stored in queue Qu .

Step 10: remove the process in queue Qu from the process tree.

Step 11: judge whether the current process tree is empty or not. Turn step 2 and turn step 12.

Step 12: exit.

3.2. Determination of Quasischeduling Time Points of Multiple-Devices-Process. Set the current scheduling process as process A , the finishing time of its pretightening process is T . Starting from point T on its processing equipment, the processing end time point of each scheduled process is found as the “quasischeduling time point” of process A and added to the “quasischeduling time point” set.

The determination of quasischeduling time point of Multiple-Devices-Process needs to consider the processing equipment of each parallel subprocess separately. The set of quasischeduled time points of a Multiple-Devices-Process is the union of all parallel subprocesses on its processing equipment. For example, there are two parallel subprocesses in processes A , $A1$, and $A2$, which are processed on different processing equipment, respectively. The “quasischeduling time point” set of $A1$ is $Ta1 = \{T1, T2, T3\}$. The “quasischeduling time points” set of $A2$ is $Ta2 = \{T4, T5\}$. The set of “quasischeduling time points” of procedure A is $tA = ta1 \cup ta2$.

3.3. Time-Selective Strategy of Multiple-Devices-Process. The time-selective strategy of Multiple-Devices-Process schedules each parallel subprocess separately at each time point in the process’s “quasischeduled time points” set. In the scheduling process, it should be noted that the first quasischeduling time point is the end time of the pretight process in the processing process tree of this process. There are two situations at this time point, as illustrated below.

As shown in Figure 2, Wi is a Multiple-Devices-Process, which contains two parallel subprocesses $Wi1$ and $Wi2$.

According to the partial order relationship in the process tree, the start time of process $Wi1$ and $Wi2$ should be consistent and greater than or equal to the end time $T1$ of the process which tightening front of process A in the process tree. The processing equipment of process $Wi1$ is $M2$, and the processing equipment of process $Wi2$ is $M1$. At this time, there are the following two conditions on the processing equipment of each parallel subprocess Wi at time $T1$.

Situation 1: $T1$ in Figure 2(a) is the end time of the scheduled process B or blank time. At this point, $T1$ is taken as the first “quasischeduling time point” of process Wi , so that $Wi1$ starts processing on $M2$ at time $T1$, and $Wi2$ starts processing on $M1$ at time $T1$.

Situation 2: $T1$ in Figure 2(b) is the processing time of the scheduled procedures B and C . In order to avoid conflict, there are two solutions. One is processes B and C do not move. Select $T3$, the maximum end time of steps B and C , as the first “quasischeduling time point” of step Wi . Second, $T1$ is selected as the first “quasischeduling time point” of process Wi , and process B is adjusted to the back of process $Wi1$, and process C is adjusted to the back of process $Wi2$. As shown in Figure 3, it is obvious that Wi is scheduled according to the second processing mode, and the resulting Multiple-Devices-Process trial scheduling scheme set will contain the case of Wi scheduled according to the first processing mode. To enlarge the problem solution space, the second processing method is selected to schedule Wi .

A simplified version of the proposed time-selective strategy of the Multiple-Devices-Process algorithm can be described as follows:

Step 1: queue a process from queue Qu and set to process A .

Step 2: suppose the end time of the tight preprocess of procedure A is T .

Step 3: determine if the process is an ordinary process or a Multiple-Devices-Process. If the ordinary process, go to 4. If the Multiple-Devices-Process, go to 5.

Step 4: start from point T on the processing equipment of process A , find the finishing time point of each

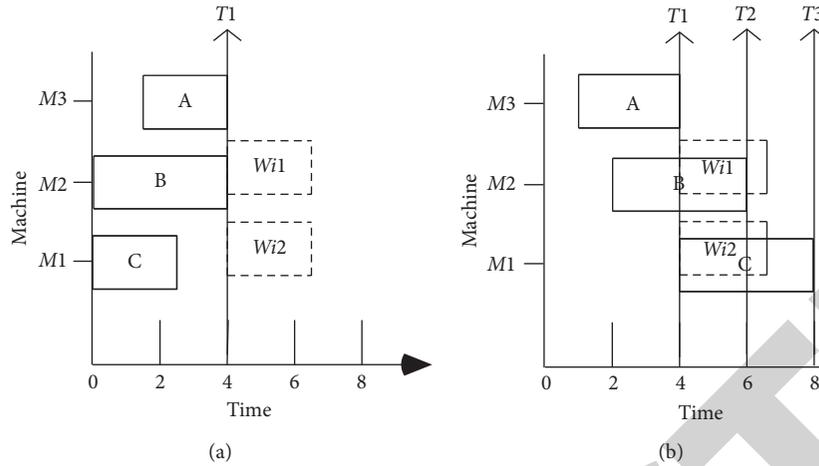


FIGURE 2: Scheduling analysis at the first quasischeduling time point.

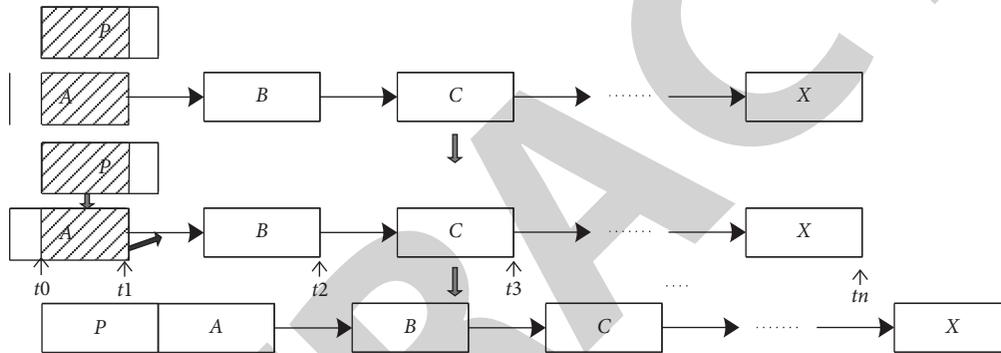


FIGURE 3: A schematic diagram of the backshift of the process affected by scheduling at the first "quasischeduling time point."

scheduled process as the "quasischeduling time point" of process A , and add it into the "quasischeduling time point" set.

Step 5: starting from point T on several parallel processing equipment of process A , the processing end time point of each scheduled process on each parallel processing equipment is found to be the "quasischeduling time point" of process A and added to the "quasischeduling time point" set.

Step 6: judge whether the set of "quasischeduling time points" is empty, not for idling to 7, for idling to 15.

Step 7: take a time point T from the set of "quasischeduled time points," and judge whether T point is the first "quasischeduled time point," if it goes to 8, if not to 11.

Step 8: if process A is an ordinary process, go to 9. If process A is a Multiple-Devices-Process, go to 10.

Step 9: determine whether process A has affected the scheduled process on the current equipment. If the affected process is added to the list of scheduled processes of the equipment, the position is after process A and goes to 12.

Step 10: judge whether each parallel subprocess of Multiple-Devices-Process A has affected the scheduled process on the current device. If the affected procedure is added to the scheduled process chain list of each parallel subprocess equipment, the position is after process A and then goes to 13.

Step 11: if process A is an ordinary process, go to 12. If process A is a multiple-devices-process, go to 13.

Step 12: take T as the start time of process A to conduct trial scheduling of process A and adjust the processes affected by the scheduling of process A , so as to produce the trial scheduling scheme generated by the scheduling of process A at the "quasischeduling time point," and add the trial scheduling scheme into the set of trial scheduling schemes of process A and turn it to 10.

Step 13: take T as the start time of each parallel subprocess in process A , and then conduct trial scheduling for each subprocess, and adjust the processes affected by each parallel subprocess of the scheduling process A (in accordance with section 3.4). The trial scheduling scheme of process A at the quasischeduling time point is generated; the trial scheduling scheme is added to the trial scheduling scheme set of Process A .

Step 14: delete this time point in the quasischeduling time point collection and go to 6.

Step 15: exit.

3.4. Time-Selective Adjustment Strategy of Multiple-Devices-Process. The selection of “quasischeduling time point” only considers the processing order between processes, without considering whether the scheduling of the process will affect the scheduled process on the equipment. Therefore, after the scheduling process, it is necessary to check the scheduled process that may be affected by it. When the processing completion time of the scheduling process is greater than the processing start time of the subsequent process on the same equipment or the processing start time of the process in the process tree, the time-selective adjustment strategy shall be started. For Multiple-Devices-Process, all parallel subprocesses need to be checked separately. Considering the situation of the first quasischeduling time point, as shown in Figure 4(a), when scheduling process W_{i1} , the processes affected are: Process B , Process D , Process E , and Process F . Wherein, when process B determines the first quasischeduling time point, its equipment is processing process, which has been processed in the design of time-selective scheduling strategy. The method is to arrange it in the chain list of processed processes of the equipment, and the position is after process W_{i1} . To adjust process B will affect the work process D on the device after tight; to adjust process D will affect the work process E on the device after tight, and work process F after tight in the process tree, by the same token, the scheduling process W_{i1} affected by the process are process C , working process, G/I and H . Consider the case that it is not the first quasischeduling time point, as shown in Figure 4(b), when scheduling process W_{i1} ; the processes affected are Process D , Process E , and Process F . When scheduling process W_{i1} , the processes affected are Process C , Process I , and Process H . To sum up, the time-selective adjustment strategy considers two situations when adjusting a process: one is the posttightening process on the processing equipment of the scheduling process. The other is the posttightening process on the equipment of the adjusted process and the middle-tightening process in the processing technology tree.

A simplified version of the proposed Multiple-Devices-Process timing adjustment strategy algorithm can be described as follows:

Step 1: queue all parallel subprocesses of the current Multiple-Devices-Process in sequence into Aqu .

Step 2: queue Aqu and store the results in W .

Step 3: judge whether W is empty, do not idle to 4, otherwise go to 10.

Step 4: suppose there are k post-tight processes in w 's process tree, $n = 1$.

Step 5: judge whether $n > k$ is true. If it is true, go to 8. If not, go to 6.

Step 6: judge whether the processing completion time of W is greater than the processing start time of process wn after the tight process in the process tree (the starting time of the unscheduled process is $+\infty$ by default), if to 7, otherwise to 8.

Step 7: take the machining end time of w as the start time of wn , and put wn into the team Aqu , $n++$.

Step 8: determine whether w has tight postprocess WM on the chain list of scheduled processes of its processing equipment and the completion time of W is greater than the start time of WM . If it is 9, otherwise it is 2.

Step 9: take the end time of W as the start time of WM , enter WM into Aqu , and turn it to 2.

Step 10: after the adjustment, calculate the total processing time of the current scheme, exit.

A simplified version of the proposed multiple-devices-process integrated scheduling algorithm with time-selective for process sequence can be described as follows:

Step 1: invert the processing partial order relationship in the process tree.

Step 2: determine the process scheduling sequence in the process tree by using the Multiple- Devices-Process sequencing strategy.

Step 3: process sequence 1 is queued out of queue Qu and an initial scheduling scheme is formed.

Step 4: determine whether the process queue Qu is empty, not for idling to 5, for idling to 10.

Step 5: queue a process from process queue Qu and set to process A .

Step 6: judge that process A is an ordinary process and multiple-devices-process. If it is an ordinary process, switch to 7. If it is a multidevices process, switch to 8.

Step 7: apply time-selective strategy and the time-selective adjustment strategy to schedule process A , generate process A trial scheme set, and go to 9.

Step 8: apply the time-selective strategy of Multiple-Devices-Process and the time-selective adjustment strategy of Multiple-Devices-Process to schedule process A , generate process A trial scheme set.

Step 9: the scheduling scheme with the minimum processing time is found out from the trial scheduling scheme set of process A , which is taken as the scheduling scheme of process A , go to 4.

Step 10: process A scheduling scheme is the product scheduling scheme.

Step 11: exit.

4. Example Analysis

In order to facilitate to the reader to understand the algorithm, the following is an example analysis. Product A is shown in Figure 5, and its reverse process tree is shown in

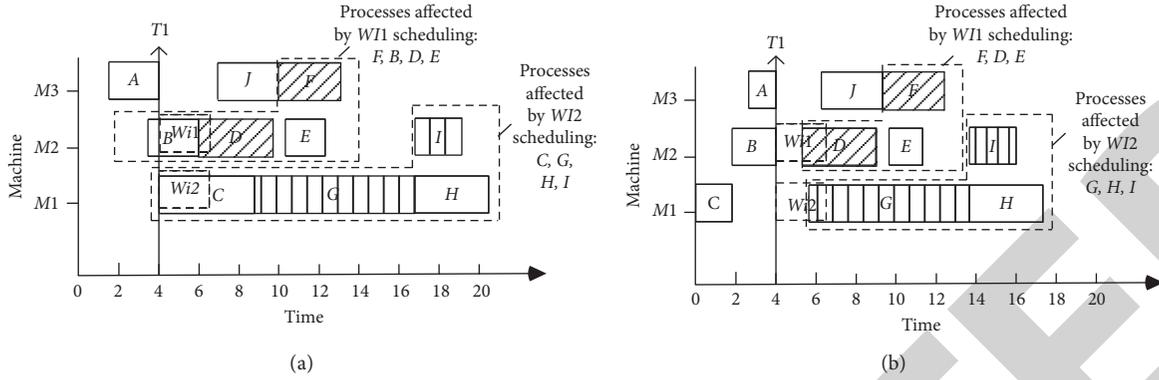


FIGURE 4: Scheduling analysis at the first quasischeduling time point.

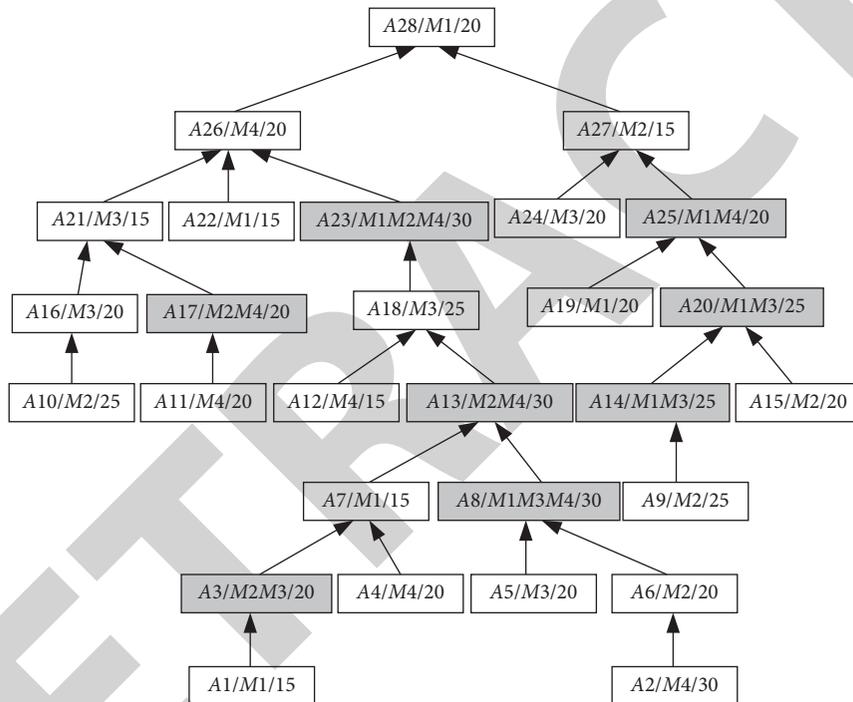


FIGURE 5: Product processing tree.

Figure 6 where each processing node represents A processing process, which is divided into ordinary processes and Multiple-Devices-Process. Such as $A28/M1/20$, which means that the process name is $A28$, and it is processed on equipment $M1$, which is an ordinary process and the processing time is 20. $A17/M2M4/20$ means that the process name is $A17$, which needs to be processed on the equipment $M2$ and $M4$ at the same time, and it is a Multiple-Devices-Process. According to the algorithm proposed in this paper, the product processing process tree is shown in Figure 6. The scheduling process is shown in Table 1, and the total processing time of product A is 265. The result of scheduling product A is shown in Figures 7 and 8. Meanwhile, the total processing time of scheduling product A according to the algorithm proposed in [12] is 275, and the result of scheduling final product A is shown in Figure 9.

5. Experimental Results and Analysis

To verify the effectiveness of the proposed algorithm, four sets of data were randomly generated, with 50 products in each set. The parameters of the products were randomly generated. The software used on the experimental platform is Windows 10, 64 bit, GCC5.5, and the hardware of the experimental platform is an Intel Core I7-860 processor with 32 GB of memory. Five groups of experiments were designed as follows: Figure 10 is a comparison of the total elapsed processing time when the number of multidevices is 2 for the two algorithms, Figure 11 is a comparison of the total elapsed processing time when the number of multidevices is 3 for the two algorithms, Figure 12 is a comparison of the total elapsed processing time when the number of multidevices is 5 for the two

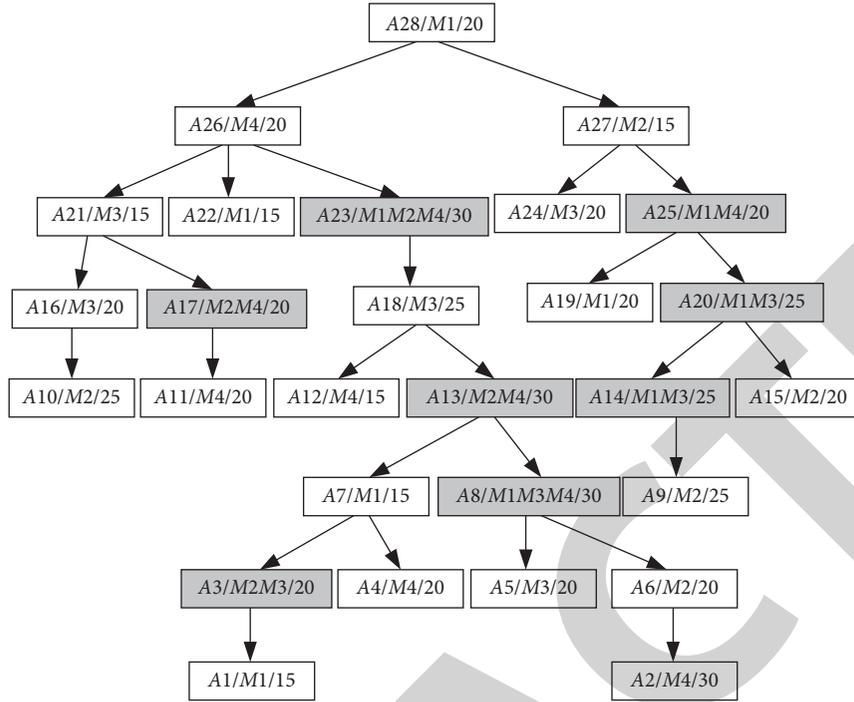


FIGURE 6: The reverse product processing tree.

TABLE 1: The scheduling process of the algorithm scheduling product A in this paper.

Id	Device	Quasischeduling time points	Total processing time of trial scheduling scheme	Determine the scheduling time point	Scheduling time points for each process in the current scheme
A28	M1	---	---	0	A28:0
A26	M4	---	---	20	A28:0, A26:20
A23	M124	---	---	40	A28:0, A26:20, A23:40
A18	M3	---	---	70	A28:0, A26:20, A23:40, A18:70
A13	M4	---	---	95	A28:0, A26:20, A23:40, A18:70, A13:95
A8	M134	---	---	125	A28:0, A26:20, A23:40, A18:70, A13:95, A8:125
A6	M2	---	---	155	A28:0, A26:20, A23:40, A18:70, A13:95, A8:125, A6:155
A2	M4	---	---	175	A28:0, A26:20, A23:40, A18:70, A13:95, A8:125, A6:155, A2:175
A27	M2	20 70 125 175	205 205 205 205	20	A28:0, A26:20, A23:40, A18:70, A13:95, A8:125, A6:155, A2:175, A27:20
A25	M1M4	35 70 95 125 155 205	220 205 220 220 205 225	70	A28:0, A26:20, A23:40, A18:70, A13:95, A8:125, A6:155, A2:175, A27:20, A25:70
A20	M1M3	90 95 155	250 205 205	95	A28:0, A26:20, A23:40, A18:70, A13:95, A8:125, A6:155, A2:175, A27:20, A25:70, A20:95
A14	M1M3	120 155	225 205	155	A28:0, A26:20, A23:40, A18:70, A13:95, A8:125, A6:155, A2:175, A27:20, A25:70, A20:95, A14:155
A9	M2	180	205	180	A28:0, A26:20, A23:40, A18:70, A13:95, A8:125, A6:155, A2:175, A27:20, A25:70, A20:95, A14:155, A9:180
A21	M3	40 95 120 155 180	205 215 215 220 205	40	A28:0, A26:20, A23:40, A18:70, A13:95, A8:125, A6:155, A2:175, A27:20, A25:70, A20:95, A14:155, A9:180, A21:40
A16	M3	55 95 120 155 180	210 220 220 225 205	180	A28:0, A26:20, A23:40, A18:70, A13:95, A8:125, A6:155, A2:175, A27:20, A25:70, A20:95, A14:155, A9:180, A21:40, A16:180

TABLE 1: Continued.

Id	Device	Quasischeduling time points	Total processing time of trial scheduling scheme	Determine the scheduling time point	Scheduling time points for each process in the current scheme
A10	M2	200 205	250 230	205	A28:0, A26:20, A23:40, A18:70, A13:95, A8:125, A6:155, A2:175, A27:20, A25:70, A20:95, A14:155, A9:180, A21:40, A16:180, A10:205
A7	M1	125 155 180	245 245 230	180	A28:0, A26:20, A23:40, A18:70, A13:95, A8:125, A6:155, A2:175, A27:20, A25:70, A20:95, A14:155, A9:180, A21:40, A16:180, A10:205, A7:180
A3	M2M3	205 230	250 250	205	A28:0, A26:20, A23:40, A18:70, A13:95, A8:125, A6:155, A2:175, A27:20, A25:70, A20:95, A14:155, A9:180, A21:40, A16:180, A10:205, A7:180, A3:205
A1	M1	225	250	225	A28:0, A26:20, A23:40, A18:70, A13:95, A8:125, A6:155, A2:175, A27:20, A25:70, A20:95, A14:155, A9:180, A21:40, A16:180, A10:205, A7:180, A3:205, A1:225
A17	M2M4	55 70 90 125 155 175 205	285 270 265 270 270 265 270	90	A28:0, A26:20, A23:40, A18:70, A13:110, A8:140, A6:170, A2:190, A27:20, A25:70, A20:95, A14:170, A9:195, A21:40, A16:195, A10:240, A7:195, A3:220, A1:240, A17:90
A11	M4	110 140 170 220	285 285 265 265	170	A28:0, A26:20, A23:40, A18:70, A13:110, A8:140, A6:170, A2:190, A27:20, A25:70, A20:95, A14:170, A9:195, A21:40, A16:195, A10:240, A7:195, A3:220, A1:240, A17:90, A11:170
A4	M4	210 220	265 265	210	A28:0, A26:20, A23:40, A18:70, A13:110, A8:140, A6:170, A2:230, A27:20, A25:70, A20:95, A14:170, A9:195, A21:40, A16:195, A10:240, A7:195, A3:220, A1:240, A17:90, A11:170, A4:210
A24	M3	35 55 95 120 170 195 215 240	265 265 265 265 285 280 280 265	35	A28:0, A26:20, A23:40, A18:70, A13:110, A8:140, A6:170, A2:230, A27:20, A25:70, A20:95, A14:170, A9:195, A21:55, A16:195, A10:240, A7:195, A3:220, A1:240, A17:90, A11:170, A4:210, A24:35
A19	M1	90 120 170 195 210 255	265 265 285 275 265 275	90	A28:0, A26:20, A23:40, A18:70, A13:110, A8:140, A6:170, A2:230, A27:20, A25:70, A20:110, A14:170, A9:195, A21:55, A16:195, A10:240, A7:195, A3:220, A1:240, A17:90, A11:170, A4:210, A24:35, A19:90
A15	M2	135 140 190 220 240 265	310 265 280 285 285 285	140	A28:0, A26:20, A23:40, A18:70, A13:110, A8:140, A6:170, A2:230, A27:20, A25:70, A20:110, A14:170, A9:195, A21:55, A16:195, A10:240, A7:195, A3:220, A1:240, A17:90, A11:170, A4:210, A24:35, A19:90, A15:140
A5	M3	170 195 215 240	285 280 280 265	240	A28:0, A26:20, A23:40, A18:70, A13:110, A8:140, A6:170, A2:230, A27:20, A25:70, A20:110, A14:170, A9:195, A21:55, A16:195, A10:240, A7:195, A3:220, A1:240, A17:90, A11:170, A4:210, A24:35, A19:90, A15:140, A5:240
A22	M1	40 70 90 110 135 170 195 210 255	280 280 275 275 275 280 270 265 270	210	A28:0, A26:20, A23:40, A18:70, A13:110, A8:140, A6:170, A2:230, A27:20, A25:70, A20:110, A14:170, A9:195, A21:55, A16:195, A10:240, A7:195, A3:220, A1:240, A17:90, A11:170, A4:210, A24:35, A19:90, A15:140, A5:240, A22:210
A12	M4	95 110 140 170 190 230 260	285 280 280 265 265 275 275	170	A28:0, A26:20, A23:40, A18:70, A13:110, A8:140, A6:170, A2:230, A27:20, A25:70, A20:110, A14:170, A9:195, A21:55, A16:195, A10:240, A7:195, A3:220, A1:240, A17:90, A11:185, A4:210, A24:35, A19:90, A15:140, A5:240, A22:210, A12:170

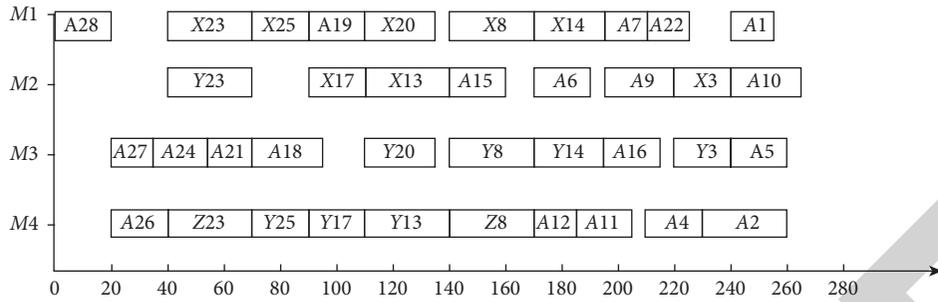


FIGURE 7: Scheduling results of the time-selective strategy of Multiple-Devices-Process.

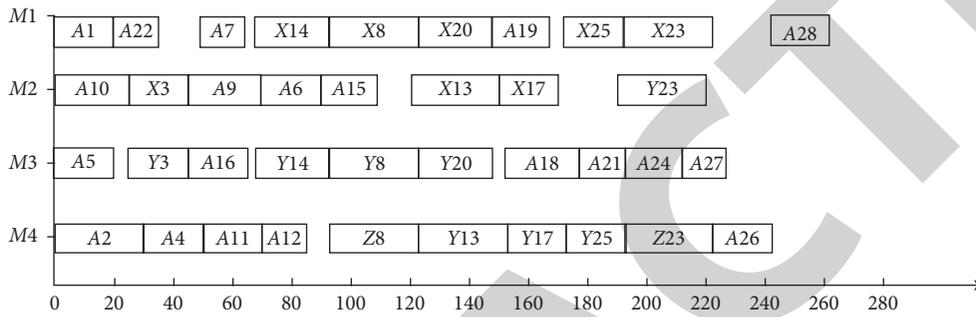


FIGURE 8: Reverse scheduling results of the time-selective strategy of Multiple-Devices-Process.

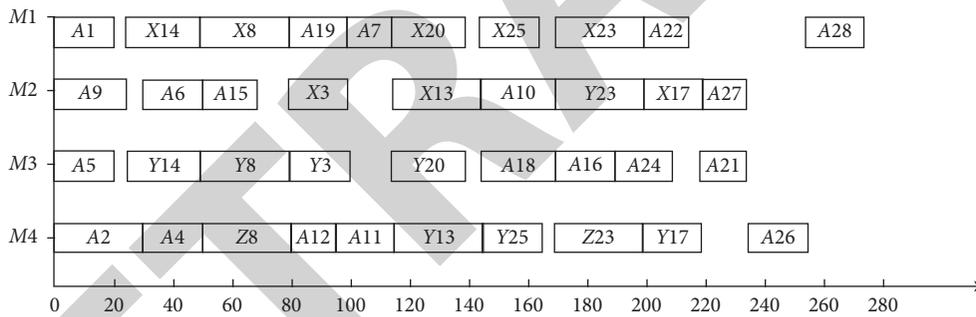


FIGURE 9: Scheduling results of the reverse multidevice first adaptation algorithm.

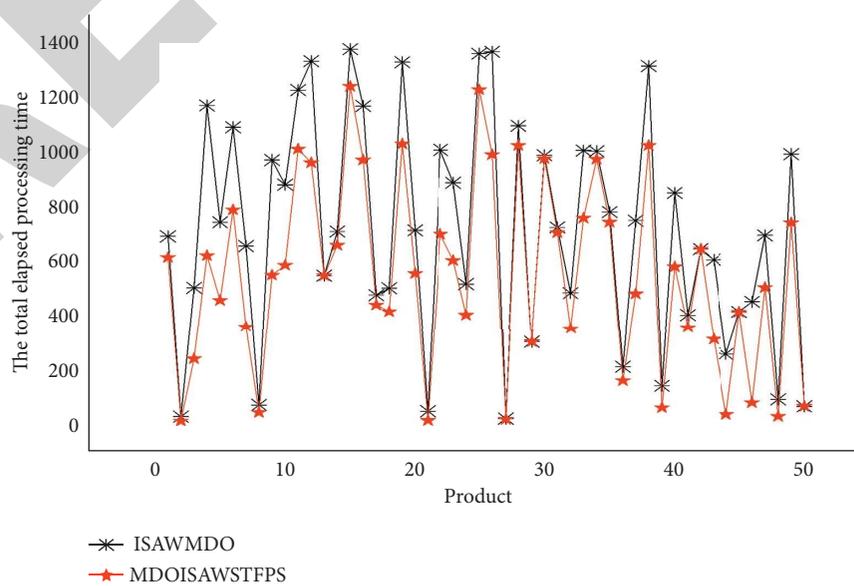


FIGURE 10: Comparison of the total elapsed processing time when the number of multiequipments is 2 for the two algorithms.

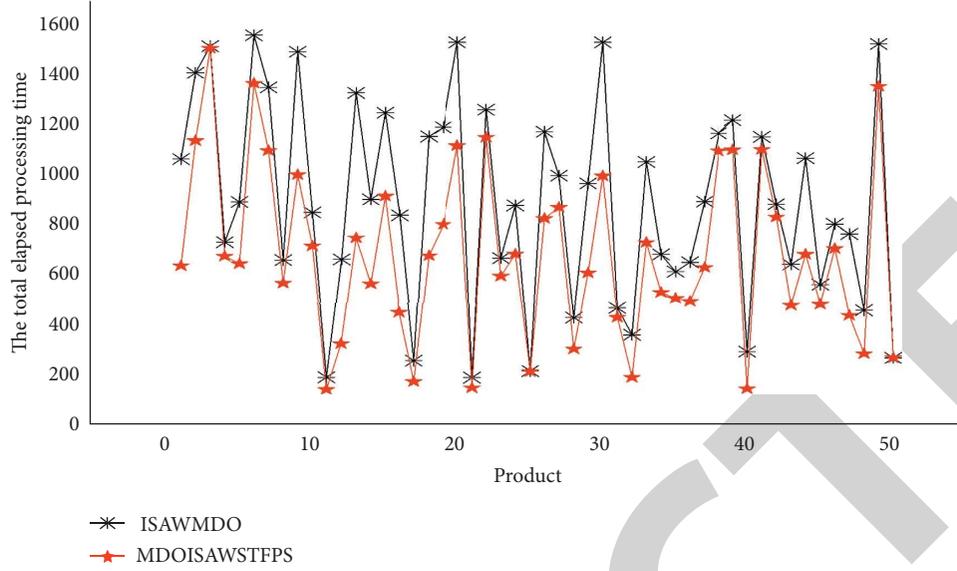


FIGURE 11: Comparison of the total elapsed processing time when the number of multiequipments is 3 for the two algorithms.

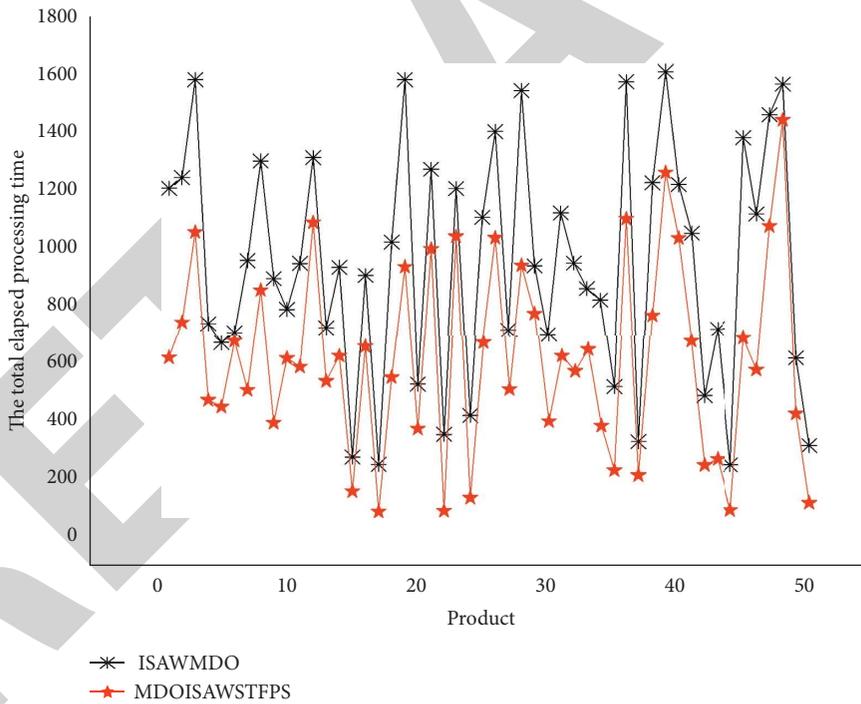


FIGURE 12: Comparison of the total elapsed processing time when the number of multiequipments is 5 for the two algorithms.

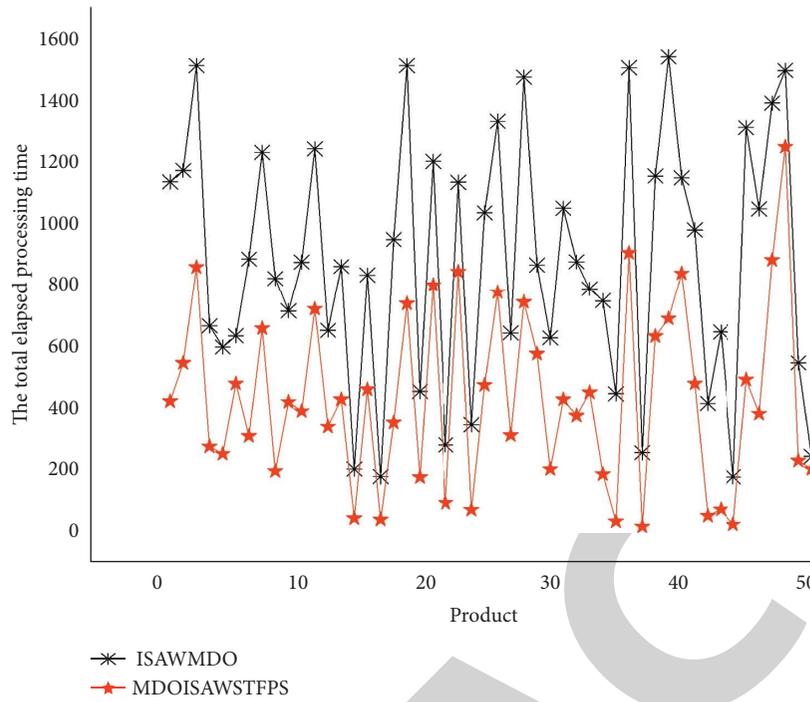


FIGURE 13: Comparison of the total elapsed processing time when the number of multiequipments is 10 for the two algorithms.

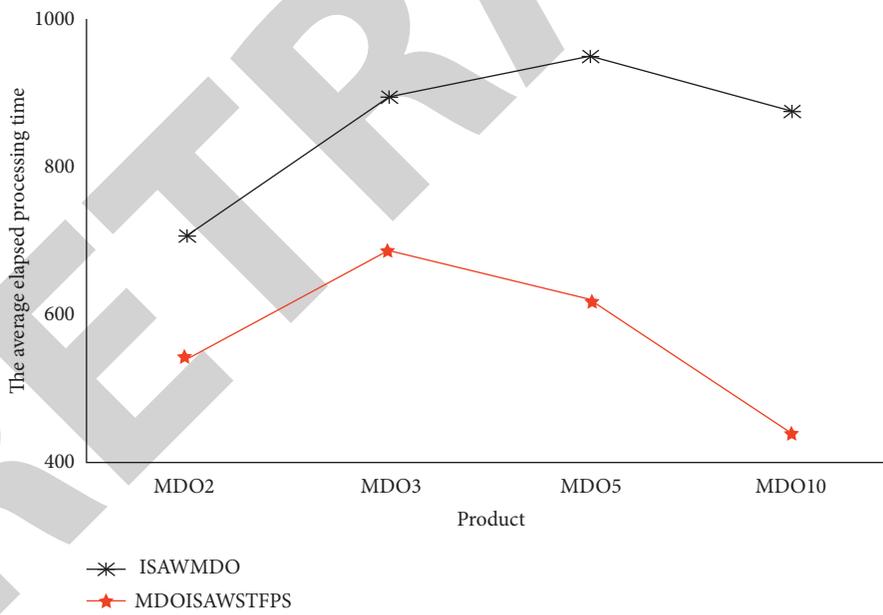


FIGURE 14: Comparison of the average elapsed processing time for the two algorithms.

algorithms, Figure 13 is a comparison of the total elapsed processing time when the number of multidevices is 10 for the two algorithms, and Figure 14 is a comparison of the average elapsed processing time for the two algorithms.

6. Conclusion

On the basis of ensuring parallel processing, the Multiple-Devices-Process sequencing strategy improves the tightness between serial processes and shortens the product completion time. The time-selective strategy of Multiple-Devices-Process and the time-selective adjustment strategy of Multiple-Devices-Process to make the current scheduling process has to cooperate with each other, to make the current partial product processing always take the smallest scheduling scheme, further shortening the product completion time, and when the scheme of minimum total available not only chooses the plan of process over time are the first, to further improve the possibility of parallel processing in the sequence after work. The scheduling results are superior to the existing Multiple-Devices-Process integrated scheduling algorithms. The introduction of a backtracking strategy to improve the accuracy of the algorithm may be the next direction of work, and it is intended to solve the problem of the multidevices process in distributed manufacturing.

Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also form part of an ongoing study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

Acknowledgments

This work was supported by the Project of Educational Commission of Guang dong (No. 2019KTSCX177), PhD Research Startup Foundation of Huizhou University (No. 2019JB014), Project of Ningbo Natural Science Foundation (No. 2019A610093), and the special research funding from the Marine Biotechnology and Marine Engineering Discipline Group in Ningbo University (No. 422004582).

References

- [1] Q. Yuan, H. Tang, W. You, X. Wand, and Y. Zhao, "Virtual network function scheduling via multilayer encoding genetic algorithm with distributed bandwidth allocation," *Science China (Information Sciences)*, vol. 61, no. 9, pp. 93–111, 2018.
- [2] Y. Zhang, F. He, G. Lu, and H. Xioing, "An impossibility message scheduling based on modified genetic algorithm for time-triggered Ethernet," *Science China (Information Sciences)*, vol. 61, no. 1, pp. 279–281, 2018.
- [3] B. Zhou and X. Li, "Heuristic tabu search scheduling algorithm for wet-etching systems in semiconductor wafer fabrications," *High Technology Letters*, vol. 19, no. 2, pp. 111–116, 2013.
- [4] G. Mejía and J. Pereira, "Multiobjective scheduling algorithm for flexible manufacturing systems with Petri nets," *Journal of Manufacturing Systems*, vol. 1, no. 1, pp. 1–10, 2020.
- [5] X. Zhang and Y. Wang, "New mixed broadcast scheduling approach using neural networks and graph coloring in wireless sensor network," *Journal of Systems Engineering and Electronics*, vol. 20, no. 1, pp. 185–191, 2009.
- [6] X. Zhang, X. Cui, and S. Zheng, "Heuristic task scheduling algorithm based on rational ant colony optimization," *Chinese Journal of Electronics*, vol. 23, no. 2, pp. 311–314, 2014.
- [7] H. Ding and X. Gu, "Improved particle swarm optimization algorithm based novel encoding and decoding schemes for flexible job shop scheduling problem," *Computers & Process Research*, vol. 121, no. 1, 5 pages, Article ID 104951, 2020.
- [8] K. Kumar and E. Das, "Genetic algorithm and particle swarm optimization in minimizing make span time in job shop scheduling," *Proceedings of ICDMC-2019*, vol. 1, no. 1, 5 pages, 2020.
- [9] Y. . Jiao, Y. E. Chunming, and S. . Business, "Novel firefly algorithm for solving job shop scheduling problem," *Computer Engineering and Applications*, vol. 49, no. 11, pp. 213–203, 2013.
- [10] H. Zhang, L. Yang, and B. Liu, "Registration of multi-resolution medical images using a modified artificial fish-swarm algorithm combined with powell's method," *Journal of Xi'an Jiaotong University*, vol. 45, no. 4, pp. 46–52, 2011.
- [11] F. Naifar, M. Gzara, and M. T. Loukil, "A new leader guided optimization for the flexible job shop problem," *Journal of Combinatorial Optimization*, vol. 39, no. 10, pp. 1–5, 2020.
- [12] Z. Xie, Y. Qi, and J. Yang, "Integrated scheduling algorithm with multiple-devices- process," *Journal of Mechanical Engineering*, vol. 49, no. 6, pp. 89–98, 2014.