

Article

# Multi-Objective Workflow Optimization Algorithm Based on a Dynamic Virtual Staged Pruning Strategy

Zhiyong Luo \*, Shanxin Tan, Xintong Liu, Haifeng Xu and Jiahui Liu

School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China; tanshanxinemail@163.com (S.T.); liuxintong2542@163.com (X.L.); xuhaifeng0325@163.com (H.X.)

\* Correspondence: luozhiyong@hrbust.edu.cn

**Abstract:** Time, cost, and quality are critical factors that impact the production of intelligent manufacturing enterprises. Achieving optimal values of production parameters is a complex problem known as an NP-hard problem, involving balancing various constraints. To address this issue, a workflow multi-objective optimization algorithm, based on the dynamic virtual staged pruning (DVSP) strategy, was proposed to optimize multi-stage nonlinear production processes. The algorithm establishes a virtual workflow model based on the actual production process and proposes a pruning strategy to eliminate the indirect constraint relationship between tasks. A virtual hierarchical strategy is employed to divide the task node set, and the Pareto optimal service set is calculated through backward iteration in stages. The optimal path is generated through forward scheduling, and the global optimal solution is obtained. The algorithm was compared with the minimum critical path algorithm (MCP) and the partial critical path budget balance scheduling algorithm (PCP-B2). The experimental results demonstrated that the DVSP can improve product quality, reduce production costs, and ensure production stability while completing production tasks. This paper used a pruning strategy and virtual workflow modeling methods to achieve dynamic multi-objective optimization scheduling for nonlinear feedback manufacturing processes.

**Keywords:** optimize scheduling; pruning strategy; workflow; virtual node; production quality



**Citation:** Luo, Z.; Tan, S.; Liu, X.; Xu, H.; Liu, J. Multi-Objective Workflow Optimization Algorithm Based on a Dynamic Virtual Staged Pruning Strategy. *Processes* **2023**, *11*, 1160. <https://doi.org/10.3390/pr11041160>

Academic Editor: Olympia Roeva

Received: 11 March 2023

Revised: 31 March 2023

Accepted: 4 April 2023

Published: 10 April 2023



**Copyright:** © 2023 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/>).

## 1. Introduction

In recent years, the intelligent manufacturing industry has experienced rapid development, and as a result, production quality and efficiency have become the primary focus of many enterprises. Workflow scheduling technology, as an advanced technology in intelligent manufacturing, is widely adopted for enterprise manufacturing equipment scheduling and scientific production. It plays a positive role in improving the quality of production processes and reducing production costs. With the rapid progress of intelligent manufacturing technology, the production mode of enterprises is moving towards complexity, synergy, and refinement. Traditional scheduling methods for single linear workflows mainly optimize a single linear objective. However, this approach does not significantly improve the efficiency and production quality of more complex production activities. Therefore, finding the optimal solution of multi-objective comprehensive performance by weighing multiple objectives is considered an NP-hard problem [1].

As production technology continues to progress and innovate, an increasing number of workflow technologies are being applied to actual production processes. This trend further promotes the continuous advancement of workflow scheduling technology [2]. Time and cost optimization are the primary optimization objectives of scheduling algorithms, and these objectives have been widely studied. For instance, particle swarm optimization, genetic algorithms (GAS), and simulated annealing algorithms have been proposed and applied to solve scheduling problems [3–8].

With the intensification of competition among manufacturers, the production qualification rate of products has become increasingly important. Improving the production

qualification rate and reducing production energy consumption while meeting the delivery deadline is currently an essential direction for multi-objective optimization. Global optimization and local search techniques are used to optimize the maximum completion time and workload of machines in flexible job shop scheduling [9].

In recent studies, Abed-Alguni et al. proposed a cloud workflow scheduling method that combines particle swarm optimization (PSO) and idle slot-aware rules to improve resource utilization and save workflow execution costs under deadline constraints by making full use of idle time slots on virtual machines. Wang et al. [10] developed a multi-objective optimization algorithm (MDSS-MOGA-DE) based on a multi-region partitioning sampling strategy to consider flexible job-shop scheduling for preventive maintenance activities and transportation processes. The algorithm combines a genetic algorithm and a differential evolution algorithm to further enhance the algorithm's search capability.

Ma et al. [11] proposed a workflow scheduling algorithm based on the Infrastructure as a Service (IaaS) model, which takes into account task deadlines and execution costs, and aims to minimize the execution cost of tasks while satisfying workflow constraints. Duan et al. [12] proposed a heuristic multi-objective non-dominated genetic sorting algorithm (NSGA-II) based on real number coding and considering batch transfer rules for dynamic scheduling of flexible job shop under mechanical fault constraints, which can effectively reduce energy consumption and maximum completion time. Dai et al. [13] established a multi-objective optimization model aimed at minimizing energy consumption and maximizing completion time for flexible job-shop scheduling problems with transportation constraints. They then proposed an improved genetic algorithm to solve this problem, which provides a decision basis for energy-saving scheduling in flexible manufacturing systems. Zhou et al. [14] proposed a new algorithm, namely budget-deadline constrained workflow scheduling (BDCWS), that considers multiple quality of service (QoS) constraints, such as cost and time constraints, for workflow scheduling.

Ndamlabin Mboula et al. [15] proposed a new efficient workflow scheduling algorithm that optimizes the cost–time tradeoff by selecting the most suitable range of VM instance types for workflow execution. This avoids overpriced and underpriced options, which can lead to budget and deadline violations. The cost optimization of workflow applications described with time constraints is a challenging problem [16]. Petchrompo et al. [17] proposed a pruning method to simplify the Pareto front solution set, determine the most promising solutions, and reduce computational steps required for obtaining the Pareto optimal set in experiments. Arabnejad V et al. [18] introduced a heuristic scheduling algorithm, budget deadline aware scheduling (BDAS), which solves eScience workflow scheduling problems under budget and deadline constraints in an infrastructure as a service (IaaS) cloud. To address the difficult problem of complex linear production process optimization scheduling, Luo et al. [19,20] proposed a three-layer virtual workflow optimization algorithm (three-OVMG) that uses a segmentation strategy to calculate the optimal value of production quality within the segment and ultimately achieve the global optimal solution. Wu et al. [21] proposed a budget allocation mechanism based on the part of the critical path (PCP) to balance the critical path of the budget and solve the optimization problem of workflow execution time under the condition of a limited budget. The proposed mechanism is based on the sequential or parallel structure properties of the critical path [22]. Zhen et al. [23] proposed a virtual workflow modeling method for parallel manufacturing processes with multiple varieties of jobs, and based on this, introduced a multi-objective virtual workflow scheduling algorithm (MOVWSA) to address the optimization problem of nonlinear production processes with feedback.

As enterprises' production processes become more complex, collaborative, and refined, there is an increasing need for targeted algorithms to optimize these processes [24,25]. The current workflow modeling method can only handle a single process route and cannot achieve optimization for nonlinear multi-objective manufacturing. To address this, a phased virtual workflow modeling method based on pruning strategy has been proposed for multi-process and multi-service nonlinear manufacturing processes. This method establishes

virtual workflows by using task nodes and corresponding services and generates virtual node sets by using virtual task nodes with feedback processes. Virtual workflow diagrams are constructed based on the partial order relationship between nodes, resulting in a virtual workflow model. To facilitate optimal scheduling of actual production processes with multiple processes and services, a multi-objective optimization algorithm based on the dynamic virtual phased pruning strategy has been proposed using the improved virtual workflow model. This algorithm first identifies the indirect constraint relationship between processes through the pruning strategy based on the task scheduling sequence and startup time and eliminates the indirect path between nodes. Then, the directed acyclic graph (DAG) is simplified. Next, the hierarchical relationship of tasks is determined using the virtual hierarchical strategy, and the set of task nodes is divided into stages. The nodes are then processed hierarchically according to the sequential scheduling relationship of nodes between stages. Finally, the optimal service node set is calculated by backward iteration in stages, and the optimal solution set is obtained. The optimal task path is generated by forward scheduling.

By constructing virtual nodes, this method can solve the nonlinear process optimization problem, which combines workflow and virtual technology to achieve a multi-objective dynamic balance problem with nonlinear characteristics. This approach can help enterprises achieve an effective balance of production quality, time, and cost, change their production mode, and improve production efficiency and quality.

## 2. Problem Description

The enterprise's production equipment is interconnected via corresponding sequences, with each task represented as a node, resulting in a workflow that can be represented by a DAG. The optimal service is selected based on QoS criteria, and the optimal path under constraints is determined.

### Related Definitions

**Definition 1.** Task node set ( $P$ ). The set of all task nodes, defined as  $P = \{p_1, p_2 \dots p_m\}$  ( $m = 1, 2 \dots m$ ). For the task node  $p$ , it is called the precursor task  $pre(i)$  before, the successor task  $suc(i)$  after it, and the successor task set  $map = \{p_1, p_2 \dots p_n\}$ .

**Definition 2.** Service node set ( $S$ ).  $S$  is a set of services in a production process, defined as  $S = \{s_1, s_2 \dots s_n\}$  ( $n = 1, 2 \dots n$ ). Each service parameter is represented as  $s_{ij} = \{t_{ij}, q_{ij}, c_{ij}\}$ , where  $t_{ij}$  represents the production time parameter of task  $p_i$ ,  $q_{ij}$  represents the production quality after completing task node  $p_i$ , and  $c_{ij}$  represents the cost parameter of task  $p_i$ . The cumulative production quality of the current task node is represented as  $f_q(p_i, t_i)$ , and the cumulative production cost of the current node is represented as  $f_c(p_i, t_i)$ .

**Definition 3.** Restriction constraint  $R = (R_t, R_q, R_c)$ .  $R_t$  is the latest end time of the entire production process,  $R_q$  denotes the minimum production quality to be achieved, and  $R_c$  denotes the cost constraint of the entire production.

**Definition 4.** Task node degree of freedom ( $HSY(i)$ ).  $HSY(i)$  represents the time interval during which the task node  $i$  can be selected. It is defined as  $HSY(i) = [T_{ns}(i), T_{ne}(i)]$ , where  $T_{ns}(i)$  is the earliest possible start time of task node  $i$ , and  $T_{ne}(i)$  is the latest possible start time. The degree of freedom of the task node  $HSY(i)$  can be calculated using Formulas (1) and (2).  $T_{ns}(suc)$  represents the earliest start time of the immediate successor node of task node  $p_i$ , and  $T_{ne}(pre)$  represents the latest start time of the immediate predecessor node of task node  $p_i$ .

$$T_{ns}(i) = \begin{cases} 0 & (i = 1) \\ \max\{T_{ns}(i-1) + \min(t_{ij})\} & [T_{ns}(i-1) \in T_{ns}(pre), (1 \leq p \leq k)] \end{cases} \quad (1)$$

$$T_{ne}(i) = \begin{cases} V_t & (i = n) \\ \min\{T_{ne}(i-1) - \min\{t_{ij}\}\} & \\ [T_{ne}(i-1) \in T_{ne}(suc), (1 \leq m \leq l)] & \end{cases} \quad (2)$$

**Definition 5.** Task execution domain ( $HSY'(i)$ ). Suppose the set  $N' \in N$  ( $N = P \cup P'$ ), then the node interval represented by the minimum period  $Vt_{min}$  and maximum period  $Vt_{max}$  determined within a certain range in the set  $N'$  is taken as the candidate domain of the node set, expressed as  $HSY'(i) = [Vt_{min}, Vt_{max}]$ .  $Vt_{min}$  and  $Vt_{max}$  can be calculated by Equation (3).

$$\begin{cases} Vt_{min} = \max\{T_{ns}(p_j) - T_{ns}(p_i)\} \\ Vt_{max} = \min\{T_{ne}(p_j) - T_{ne}(p_i)\} \end{cases} \quad (3)$$

During the execution of the workflow model, it is assumed that a corresponding service set  $s_{aj} = \{t_{aj}, q_{aj}\}$  exists in task set  $p_k \in P$ . The selected service must satisfy certain constraints to ensure the algorithm can proceed and the production quality can be effectively improved within the given deadline. The constraints for the virtual workflow model are expressed in the following formula:

$$\begin{cases} Entire_t = \min \sum_{i \in v} \sum_{P_a \in P} G_{ia} \cdot t_{aj} \leq Vt \\ Entire_q = \max \prod_{P_a \in P} G_{ia} \cdot q_{aj} \\ G_{ia} \in \{0, 1\}, \forall i \in v, 0 \leq a \leq n \\ S.T \sum_{a=1}^m G_{ia} = 1, S_a = \{S_1, S_2, \dots, S_m\} \end{cases} \quad (4)$$

In Equation (4),  $G_{ia}$  is a Boolean function that ensures task  $P_i$  can only select one service from the service set  $S_a$ . When  $G_{ia} = 1$ , it means that service  $S_a$  in the corresponding service set  $S_a$  executed during task  $P_i$  is selected. When  $G_{ia} = 0$ , it means that this service is not selected.  $Entire_t$  indicates the total completion time of the task, which should be less than the time constraint  $Vt$ .  $Entire_t$  also represents the total quality parameter of the workflow. Service set  $s_{aj} = \{t_{aj}, q_{aj}, c_{aj}\}$  exists in service set  $P_i \in P$ .

**Definition 6.** Virtual node ( $s'$ ). The set of nodes consisting of several nodes, which are virtual from node  $s_i$  to node  $s_j$ , is denoted as  $s'[i, j]$ .  $S'$  denotes the set of virtual nodes consisting of several virtual nodes  $s'$ .  $p'_i = p[i, j]$  denotes a virtual node after recombination from adjacent task nodes  $p_i$  and  $p_j$ , and the virtual node set  $P'$  denotes the set consisting of all virtual migrated nodes  $p'_i$ .

**Definition 7.** Virtual workflow ( $XN(M, S, P', T', E, In, Out)$ ).  $XN$  is the virtual workflow name;  $M$  is the initial workflow before the virtual node of the manufacturing process;  $S$  is the quality check site set in the virtual reconstruction process according to the actual production needs, which can be expressed as  $S = (S_1, S_2, \dots, S_n)$ , where  $n$  is the code of the quality check site;  $E$  denotes the partial order relationship between tasks and task  $p_i$  can be executed only when all its parent tasks are completed;  $P'$  is the abstract set of several nodes for virtual reconfiguration into virtual task nodes, denoted as  $P' = (p'_1, p'_2, \dots, p'_i, \dots, p'_n)$ ;  $In$  and  $Out$  are the sets of the number of in and out degrees of each node in the workflow  $XN$ , respectively. Productive virtual node workflow model  $XNG = (S, XN, XNT)$  is an optimal scheduling model consisting of a collection of service nodes, a virtual workflow  $XN$ , and a virtual workflow graph  $XNT (XN, E)$ .

### 3. Workflow Multi-Objective Optimization Algorithm with a Virtual Phased Pruning Strategy

#### 3.1. Pruning Strategy

In a DAG, each node represents a task, and related nodes have a scheduling sequence. The task that must be executed before a particular task is called its  $pre(i)$ , while the task that must be executed after it is called its  $suc(i)$ . The execution of a subsequent task can only begin after the completion of the previous task. For example, consider a task set  $P = \{p_1,$

$p_2, p_3\}$ , where the following constraints exist: Task  $p_3$  must be executed after task  $p_2$  is completed, and task  $p_2$  must be executed after task  $p_1$  is completed. The constraints on task  $p_3$  and task  $p_1$  are indirect constraints, and the path connecting them is known as an indirect path, denoted as  $\langle p_3, p_1 \rangle$ .

**Definition 8.** Indirect path  $\langle p_i, p_j \rangle$ . In a DAG, an indirect path refers to a directed path between a node and its successor node that is not directly connected to the set of nodes. The constraint relationship between the preceding and following tasks that constitute an indirect path is called an indirect constraint relationship.

Basic Idea of the Pruning Strategy

The pruning strategy aims to reduce the complexity of the directed acyclic graph by eliminating indirect paths between nodes. Initially, each task node in the graph corresponds to several leading nodes. By examining the leading task set  $map = \{p_1, p_2 \dots p_n\}$ , the  $Out_{p_i}$  value of each node in the initial order is determined. If the  $Out_{p_i}$  value is less than or equal to 1, it indicates that the current node has a unique leading node and there is no indirect path, so it is not further processed and is output to the ordered set  $S_{out}$ . However, if the  $Out_{p_i}$  value is greater than 1, it means that there are multiple leading nodes for the current task, and the leading set of each task in the leading node set is traversed to identify identical tasks. When multiple tasks are found to be identical, the indirect path between the current task and the identical task is recorded as  $\langle p_i, p_m \rangle$ , and the indirect path is deleted. The current task  $p_i$  is then added to the ordered set, and the process is repeated until there are no more indirect paths for the task node.

The diagram in Figure 1 represents the production process of a manufacturing plant, with each node representing a specific production process. The indirect constraint relationships between the nodes are derived from the pruning strategy analysis. The red dashed lines in Figure 1 show the indirect paths between nodes with indirect constraint relationships, namely the paths between nodes  $p_1$  and  $p_3$ , between nodes  $p_4$  and  $p_6$ , and between nodes  $p_6$  and  $p_9$ . After applying the pruning strategy, the pruned DAG diagram in Figure 2 is obtained. The related algorithmic process is shown in Algorithm 1.

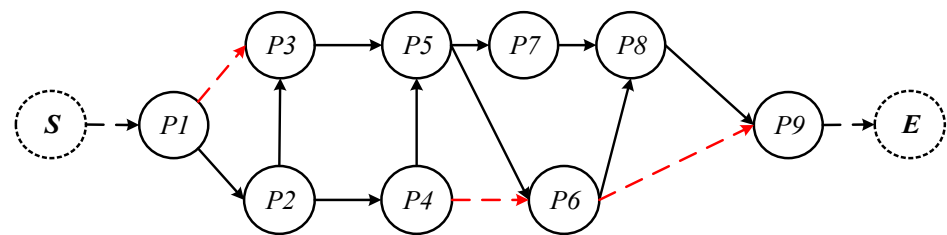


Figure 1. Initial production process DAG diagram.

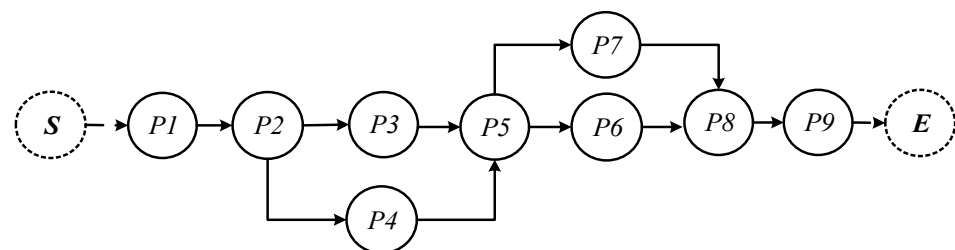


Figure 2. DAG diagram of the production process after pruning.

**Algorithm 1:** Pruning strategy.

**INPUT:** The task set  $P$ , the predecessor task set  $Map$  and  $Out$  set obtained by traversal, and the initial directed acyclic graph;

**OUTPUT:** The pruned directed acyclic graph, the initial workflow graph.

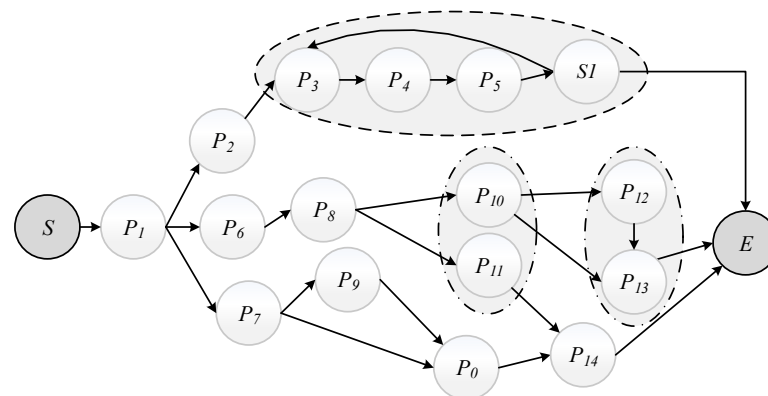
```

1: While ( $i = 1, i \leq n, i++$ ) do
2:   If ( $Out_{p_i} = 1$  or  $Out_{p_i} = 0$ ) Then
3:     Output to the ordered set  $S_{out}$ ;
4:   End
5:   Else ( $Out_{p_i} > 1$ ) Then
6:     foreach the  $map$  set  $map_x \dots map_y$  of all tasks in
7:        $map_i = \{p_x \dots p_y\}$ , generate a new set  $map_i' = \{map_x \dots map_y\}$ ;
8:       If  $map_i$  exists with the same task node  $p_m$  as  $map_i'$  Then
9:         the existence of indirect paths between  $p_i$  and  $p_m$  is noted
10:        as  $\langle p_i, p_m \rangle$ ;
11:        Delete  $\langle p_i, p_m \rangle$  from DAG;
12:      Else Output to the ordered set  $S_{out}$ 
13:    End
14:  End
15: End

```

## 3.2. Virtual Layering Strategy

**Definition 9.** Wrong Route (N-route). In the virtual node domain XNT construction, node  $p_i$  is reconstructed with other neighboring nodes  $p_j$  to form a virtual node, but there are multiple out-degrees of node  $p_i$ , i.e., the out-degree of node  $p_i$  is not unique, and the rest of the out-degrees can be reconstructed with the remaining node  $p_n$ , so the path composed of nodes  $p_i$  and  $p_n$  is called Wrong Route, denoted by N-route, where  $p_i, p_j$ , and  $p_n$  are not the same node. As shown in Figure 3,  $p_{[10-11]}$  and  $p_{[12-13]}$  are wrong route.



**Figure 3.** Identification process of dissimilar paths.

**Definition 10.** The optimal path ( $R_m$ ). It consists of virtual nodes and related service parameters, and is defined as  $R_m(P', N)$ , where  $P'$  is the set of virtual task nodes that compose the optimal path.  $N$  is the cumulative sum of various parameters of virtual task nodes in the virtualization process of  $R_m$ .

The virtual strategy is a pruning-based approach that abstracts certain tasks in a DAG as virtual nodes to address optimization problems in nonlinear production processes. By virtualizing, this technique combines workflow and virtualization methods to tackle multi-

objective dynamic equilibrium problems that exhibit nonlinear characteristics. The related algorithmic process is shown in Algorithm 2.

---

**Algorithm 2:** Virtual layering strategy.

---

**INPUT:** Workflow diagram after pruning,  $P, S', P', P_m', In, Out$ .

**OUTPUT:** Virtual workflow XN, Virtual workflow diagram XNT.

1. Initialize  $P', P_m'$  to be empty.
  2. Construct the original workflow model  $M$  by combining the pruned workflow graph, task node set  $P$  and virtual node set  $P'$  with the partial order relationship;
  3. Calculate the degree of freedom for each task node and store it in  $HSY(i)$ ;
  4. **Foreach** the  $M$  in order from the starting node  $S$  to find tasks with degree of
  5.     access  $> 1$  and add them to a list;
  6.     **While** ( $List$  is not null and there is a task in  $List$  with  $Out! = 1$ )
  7.         Fetch the first task  $p_i$  in the queue with  $Out > 1$ ;
  8.         Find the node  $p_j$  in the  $List$  with the smallest distance  $In$  from  $p_i$ ;
  9.         Find all service nodes  $s_i$  between tasks;
  10.         Virtualize task  $p_i$  and  $p_j$  into task  $p_i' = p[i, j]$ ;
  11.         Save it temporarily in the virtual node set  $P'$ ;
  12.         Save the service selection in the virtual migration node set  $PM'$ ;
  13.     **End**
  14. **End**
  15. **If** there is a  $N$ -route in the virtual node set  $P'$ , delete the virtual task nodes  $p_i'$  and restore them to their original positions, otherwise, update the virtual node.
  16.     **If** the set  $P'$  is not empty, the loop is judged again, otherwise mark and output;
  17.     **Foreach** the set of virtual task nodes  $P'$  and all virtual nodes  $p_i' = p[i, j]$  in the
  18.         model  $XN$ . Record the processing domain  $HSY'(i)$  of task nodes into model
  19.          $M$ , construct a virtual workflow graph  $XNT$  with all paths from virtual nodes
  20.          $p_i$  to  $p_j$  ( $i \neq j$ ) as directed edges.
  21.     **End**
  22. **Return** the virtual workflow graph  $XNT$ ;
- 

### 3.3. Algorithm Description

The multi-objective workflow optimization algorithm with a virtual hierarchical pruning strategy uses a pruning strategy to handle with nonlinear production processes. It applies a dynamic pruning strategy to eliminate the indirect constraint relationship between tasks and divides task nodes into stages using a virtual hierarchical strategy. It generates virtual nodes and workflow graphs using virtualization technology, calculates the set of stage optimal service nodes by inverse reversion, and seeks the global optimal solution by algorithmic integration.

Assuming an initial workflow model with  $n$  tasks, the production quality of node  $p_i$  at time  $t_i$  is denoted as  $f_q(p_i, t_i)$ , and the cumulative cost of completing task  $p_i$  is denoted as  $f_c(p_i, t_i)$ . The quality and cost parameters for selecting a service at node  $p_i$  are represented by  $q_{ij}$  and  $c_{ij}$ , respectively. The execution time candidate domain of task  $p_i$  is given by  $HXY(p_i)$ , subject to constraints on cumulative time and cost. The cumulative production parameters of task node  $p_i$  ( $1 \leq i \leq n$ ) are calculated using Equation (5).

$$\begin{cases} f_q(p_i, t_i) = \max\{q_{ij}\}, i = n, 0 < j \leq n \\ t \in HXY(p_i), t_i + t_{i-1} \leq R_t \\ f_c(p_i, t_i) = \max\{c_{ij}\}, i = n, 0 < j \leq n \\ t \in HXY(p_i), c_i + c_{i-1} \leq R_c \end{cases} \quad (5)$$

Equation (6) is obtained by reverse deriving from Equation (5). In this equation,  $f'_q(p_i, t_i)$

and  $f'_c(p_i, c_i)$  represent the maximum achievable quality and minimum cumulative cost within the given time constraint  $HXY(p_i)$ .

$$\begin{cases} f'_q(p_{i-1}, t_{i-1}) = \max \left\{ \begin{array}{l} f_q(p_i, t_i + t_{i-1}) * q_{i-1j} \\ t \in HXY(p_i) q_i * q_{i-1} > R_q \end{array} \right\} \\ f'_c(p_{i-1}, t_{i-1}) = \min \left\{ \begin{array}{l} f_c(p_i, t_i + t_{i-1}) + c_{i-1j} \\ t \in HXY(p_i) c_i + c_{i-1} \leq R_c \end{array} \right\} \end{cases} \quad (6)$$

### 3.4. Algorithm Steps

The algorithm is analyzed by reverse iteration to output the optimal scheduling path of the workflow model. The scheduling algorithm process for virtual layer pruning strategy can be found in Algorithm 3.

---

#### Algorithm 3: DVSP

---

**INPUT:** Virtual hierarchical pruned workflow graph with restriction constraints

$R = (R_t, R_q, R_c)$  and virtual workflow graph  $XNT$ .

**OUTPUT:** Set of production parameters, optimal path  $R_m$ .

1. **For** (int  $I = P.length$ ;  $i \leq 0$ ;  $i--$ ) **do**
  2.     **For**(int  $j = i-1$ ;  $j--$ )**Scan**( $XNT$ ) **do**
  3.         Using Equations (1) and (2) to derive  $f_q(p_0, t_0)$  and  $f_q(p_i, c_i)$  for task node  $P_0$  within its nodal degrees of freedom  $HSY(p_0)$ ;
  4.     **End**
  5. **For** (int  $i = P.length$ ;  $i \geq 1$ ;  $i--$ ) **do**
  6.     **If** ( $p_i$  in  $N-route$ ) **then**
  7.         Iteratively calculate and label the  $f_q(p_i, t_i)$  and  $f_q(p_i, c_i)$  of the node  $p_i$  within the task execution domain  $HSY'(i)$  using inverse reduction;
  8.     **Else**
  9.         **If** the node is not in the wrong path, calculate  $f_q(p_i, t_i)$  and  $f_q(p_i, c_i)$  for node  $p_i$  in its task execution domain  $HSY'(i)$  by inverse reductive iteration.
  10.     **End**
  11. **End**
  12. The set of task nodes  $P$  that satisfy the constraints  $R = (R_t, R_q, R_c)$  is found by forward scheduling;
  13. **Return** Optimization path  $R_m$  of the production process;
- 

## 4. Experimental Validation of DVSP

The giant motor of an intelligent equipment manufacturer is a combination of three semi-finished products, and each semi-finished product contains several production steps and inspection points, and there are several service sets for each step. The initial production process is shown in Figure 4, and there is a constraint relationship between task nodes in the actual production process. The workflow after pruning hierarchical processing is shown in Figure 5.

Due to the limitation of space, the following processes are mainly analyzed and studied for the convenience of analysis, and nodes  $p_1$  to  $p_9$  are selected as the objects of study, and the set of service nodes is shown in Table 1.



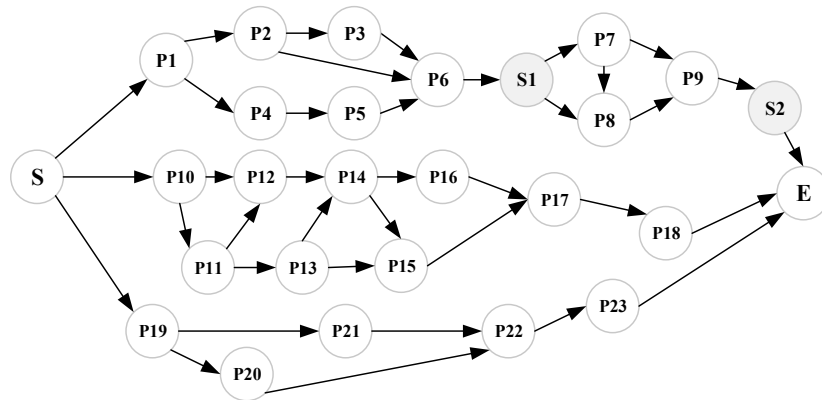


Figure 4. Relationship of the giant motor production process.

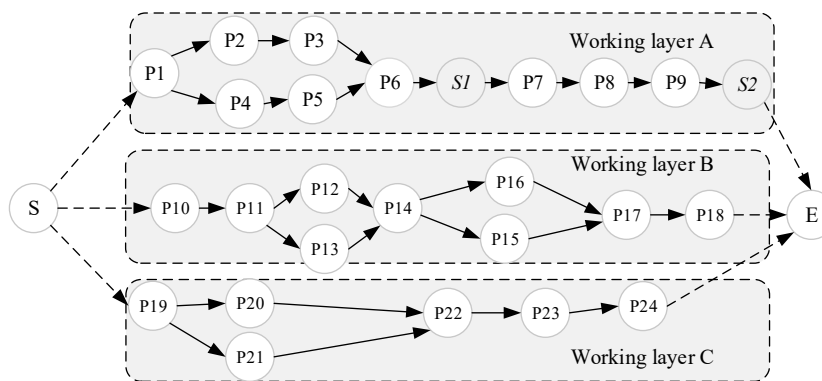


Figure 5. Relationship diagram after virtual phased pruning.

Table 1. Collection of production nodes and service time, quality, and cost parameters.

Task/ $p_i$	Type	Services/ $s_i$	Task/ $p_i$	Type	Services/ $s_i$
$p_1$	A	(2, 0.98, 0.3)	$p_{13}$	B	(3, 0.94, 0.3) (6, 0.97, 0.4)
$p_2$	A	(2, 0.94, 0.6) (4, 0.95, 1.2) (7, 0.96, 1.4)	$p_{14}$	B	(2, 0.93, 0.2) (5, 0.95, 0.5) (6, 0.96, 0.6)
$p_3$	A	(2, 0.95, 0.6) (5, 0.96, 1.5)	$p_{15}$	B	(2, 0.95, 0.3) (4, 0.96, 0.3)
$p_4$	A	(4, 0.95, 1.2)	$p_{16}$	B	(3, 0.93, 0.2)
$p_5$	A	(2, 0.93, 0.4) (5, 0.95, 1.5) (6, 0.96, 1.8)	$p_{17}$	B	(2, 0.94, 0.2) (4, 0.95, 0.3) (7, 0.96, 0.2)
$p_6$	A	(3, 0.95, 0.9)	$p_{18}$	B	(2, 0.95, 0.3) (5, 0.96, 0.4)
$p_7$	A	(2, 0.92, 0.6) (7, 0.95, 1.4)	$p_{19}$	C	(2, 0.92, 0.3) (7, 0.95, 0.7)
$p_8$	A	(3, 0.91, 0.9) (4, 0.93, 1.2) (6, 0.94, 1.8)	$p_{20}$	C	(2, 0.93, 0.2) (5, 0.95, 0.5) (6, 0.96, 0.6)
$p_9$	A	(2, 0.92, 0.4) (4, 0.95, 1.2)	$p_{21}$	C	(3, 0.95, 0.3)
$p_{10}$	B	(1, 0.91, 0.3) (3, 0.94, 0.5)	$p_{22}$	C	(1, 0.91, 0.1) (2, 0.94, 0.2)
$p_{11}$	B	(2, 0.98, 0.6) (3, 0.94, 0.8)	$p_{23}$	C	(2, 0.98, 0.2) (3, 0.94, 0.3)
$p_{12}$	B	(1, 0.91, 0.1) (2, 0.94, 0.2)	$p_{24}$	C	(2, 0.98, 0.1)

Workflow Scheduling Process

In order to verify the optimization performance of DVSP, several production processes are selected to analyze the performance of the algorithm. The following experimental environment is adopted: the programming language is Java, the operating system selected

for the server is Windows 10, the CPU selected for the PC is 3.8 GHz, and the memory size is 16 G.

In the initial stage of the algorithm, the task nodes in the production process are pruned to generate a pruned DAG. Then, the related service parameter sets of the task nodes are added to the workflow graph. Next, the virtual hierarchy strategy is used to virtualize the nodes, and a virtual workflow and virtual workflow model are created. The specific process is shown in Figure 6.

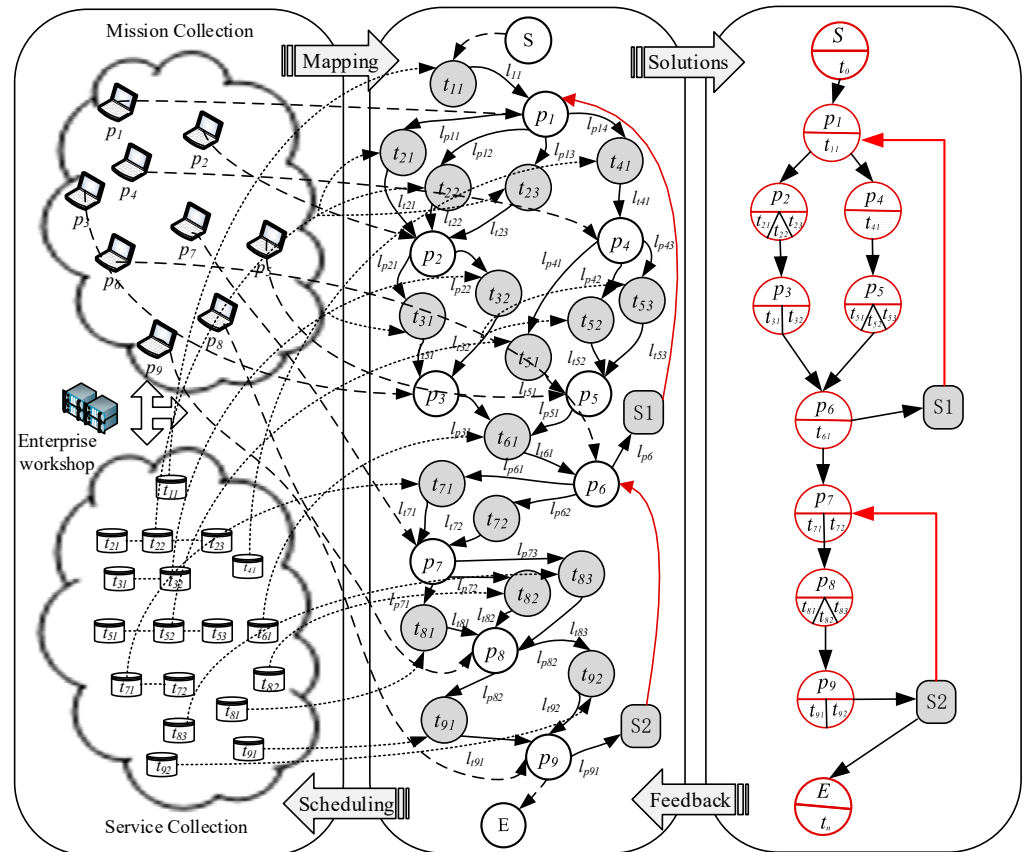


Figure 6. Virtual workflow model creation process.

The production process shown in Figure 2 is divided into two optimization intervals, interval A and B, based on the characteristics of the production process. The production time constraint for interval A is set as  $R_{t1} = 18$  days, while the time constraint for interval B is set as  $R_{t2} = 15$  days.

To ensure the overall production quality, the relevant production parameters are monitored in each interval through detection stations. Based on actual production data, the threshold value of monitoring station S1 is set to  $\beta_{q1} = 0.910$  and  $\beta_{c1} = 8.0$ , while the threshold value of monitoring station S2 is set to  $\beta_{q1} = 0.935$  and  $\beta_{c1} = 5.0$ . Thus, if the production parameters  $f_q(p_i, t_i)$  of the material flow produced by  $p_6$  and  $p_9$  do not meet the requirements at the detection station, they need to be sent back to the front production node for processing again. It has been statistically determined that reworking increases production time by 4 days and cost by 2.0. According to actual production regulations, the constraint vector  $R$  is:  $R_t = 33$  days,  $R_q = 0.920$ , and  $R_c = 13.0$  (Unit/ten thousand yuan).

Interval A has a time constraint of 18 days. After analyzing the interval nodes and determining that they meet the requirements of virtual reconstruction and there are no  $N$ -route, the execution domain of interval nodes is as follows:  $HSY(p_1) = [0,7]$ ,  $HSY(p_{[2-3]}) = [2,9]$ ,  $HSY(p_{[4-5]}) = [2,9]$ ,  $HSY(p_6) = [8,15]$ . The calculation process is shown in Table 2.

**Table 2.** DVSP algorithm interval A optimization steps.

Task/ $p_i$	Quality/ $f_q(pi,t)$ and Cost/ $f_c(pi,t)$
$P_6$	$f_q(p_6,15) = \max\{0.95\} = 0.95; f_c = 0.9;$ $f_q(p_6,14) = \max\{0.95\} = 0.95; f_c = 0.9;$ ... $f_q(p_6,8) = \max\{0.95\} = 0.95; f_c = 0.9;$
	$f_q(p_{[4-5]},9) = \max\{f_q(p_6,15)0.93 \times 0.95\} = 0.839; f_c = 2.5;$ $f_q(p_{[4-5]},8) = \max\{f_q(p_6,14)0.93 \times 0.95\} = 0.839; f_c = 2.5;$ ...
	$f_q(p_{[4-5]},4) = \max\{f_q(p_6,10)0.96 \times 0.95, 0.839, 0.857\} = 0.866; f_c = 3.9;$ $f_q(p_{[4-5]},3) = \max\{f_q(p_6,9)0.96 \times 0.95, 0.839, 0.857\} = 0.866; f_c = 3.9;$ $f_q(p_{[4-5]},2) = \max\{f_q(p_6,8)0.96 \times 0.95, 0.857\} = 0.866; f_c = 3.9;$
$P_{[2-3]}$	$f_q(p_{[2-3]},9) = \max\{f_q(p_6,15)0.95 \times 0.95\} = 0.857; f_c = 2.7;$ $f_q(p_{[2-3]},6) = \max\{f_q(p_6,12)0.95 \times 0.96\} = 0.866; f_c = 2.7;$ ...
	$f_q(p_{[2-3]},4) = \max\{f_q(p_6,10)0.95 \times 0.96, 0.85\} = 0.866; f_c = 2.9;$ $f_q(p_{[2-3]},3) = \max\{f_q(p_6,9)0.96 \times 0.97, 0.866, 0.857\} = 0.876; f_c = 3.8;$ $f_q(p_{[2-3]},2) = \max\{f_q(p_6,8)0.96 \times 0.97, 0.866\} = 0.876; f_c = 3.8;$
	$f_q(p_1,7) = \max\{f_q(p_{[2-5]},9) 0.98\} = 0.705; f_c = 5.5;$ $f_q(p_1,6) = \max\{f_q(p_{[2-5]},8) 0.98\} = 0.705; f_c = 5.5;$ $f_q(p_1,5) = \max\{f_q(p_{[2-5]},7) 0.98\} = 0.705; f_c = 5.5;$ $f_q(p_1,4) = \max\{f_q(p_{[2-5]},6) 0.98\} = 0.727; f_c = 6.8;$ ... $f_q(p_1,2) = \max\{f_q(p_{[2-5]},4) 0.98\} = 0.736; f_c = 7.1;$ $f_q(p_1,1) = \max\{f_q(p_{[2-5]},3) 0.98\} = 0.744; f_c = 8.0;$ $f_q(p_1,0) = \max\{f_q(p_{[2-5]},2) 0.98\} = 0.744; f_c = 8.0;$

According to the above process, the set of service time used by node  $p_1$  in interval A through reverse iterative calculation is  $\{15,16,15,15,14,11,11\}$ , and the set of service quality is  $\{0.744, 0.744, 0.736, 0.736, 0.727, 0.705, 0.705, 0.705\}$ , service cost collection is  $\{8.0, 8.0, 7.1, 7.1, 6.8, 5.5, 5.5, 5.5\}$ , in the range of A node after the  $p_6$  takes S1 testing station detection, the beta  $\beta_{q1} = 0.910$ , beta  $\beta_{c1} = 8.0$ . Some paths were eliminated due to exceeding the production time limit constraints, such as the path from  $f(p_1,0)$  to  $f(p_1,3)$ . After the rest of the path to the reprocessing:  $f_q(p_1,4) = 0.727 + 0.727 \times (1 - 0.727) = 0.926$ ,  $f_q(p_1,5) = f_q(p_1,7) = 0.705 + 0.705 \times (1 - 0.705) = 0.913$ . After comparing the second most significant factors, it can be concluded that the production time of the  $f(p_1,4)$  path is shorter and has the highest accumulated production quality. Hence, the current stage's Pareto optimal solution set is  $f_q(p_1,4) = 0.926$ ,  $f_t(p_1,4) = 18$  days,  $f_c(p_1,4) = 7.8$ . The optimal path through forward scheduling interval A is as follows:  $R_A = \{S, p_1(t_{11}), p_{[2-5]}(t_{23}, t_{32}, t_{41}, t_{52}), p_6(t_{61})\}$ .

Interval B is the subsequent workflow to interval A, and the deadline for interval B is 15. After analysis, it was determined that interval B satisfies the requirements for virtual reconstruction and there is no N-route. The execution domains for interval B's nodes are as follows:  $HSY(p_7) = [0,8]$ ,  $HSY(p_8) = [2,10]$ ,  $HSY(p_9) = [5,13]$ . The calculation process is illustrated in Table 3.

Interval B is a subsequent workflow that follows interval A. The production times of node  $p_7$  in interval B are  $\{13,12,13,12,10,10,9,8,7\}$ , and the corresponding production quality collection is  $\{0.778,0.761,0.761,0.761,0.753,0.753,0.737,0.730,0.713\}$ . The production cost for these nodes is  $\{3.9,3.6,3.9,3.6,3.0,3.0,2.7,2.4,2.1\}$ . When reaching node  $p_9$ , the production process needs to pass through S2 test station for quality inspection with parameters  $\beta_{q2} = 0.935$  and  $\beta_{c2} = 5.0$ . After inspection, all production paths require secondary processing, and the revised production times become  $\{16,17,16,17,14,14,13,12,11\}$ . Among them,  $f(p_7,4)$  to  $f(p_7,8)$  meet the requirements of production parameter monitoring stations, and their production quality collection is  $\{0.939,0.939,0.937,0.927,0.918\}$ . The production cost for these nodes is  $\{4.9,4.6,4.9,4.6,4.0,4.0,3.7,3.4,3.1\}$ . There is a group of the same quality in the service set, and the solution set with the highest cumulative production quality is preferentially

selected according to the Pareto optimal rule. Therefore, the Pareto optimal solution set at the current stage is  $f_q(p_7,4) = 0.939$ ,  $f_t(p_7,4) = 14$  days,  $f_c(p_7,4) = 5.0$ . The optimal path through forward analysis interval  $B$  is as follows:  $R_B = \{p_7(t_{71}), p_8(t_{82}), p_9(t_{92}), E\}$ .

**Table 3.** DVSP method interval B optimization steps.

Task/ $p_i$	Quality/ $f_q(p_i,t)$ & Cost/ $f_c(p_i,t)$
$P_9$	$f_q(p_9,13) = \max\{0.92\} = 0.92; f_c = 0.6;$ $f_q(p_9,12) = \max\{0.92\} = 0.92; f_c = 0.6;$ $f_q(p_9,11) = \max\{0.92,0.95\} = 0.95; f_c = 1.2;$
	...
	$f_q(p_9,5) = \max\{0.92,0.95\} = 0.95; f_c = 1.2;$
$P_8$	$f_q(p_8,10) = \max\{f_q(p_9,13)0.91\} = 0.837; f_c = 1.5;$ $f_q(p_8,7) = \max\{f_q(p_9,11)0.93\} = 0.884; f_c = 1.8;$ $f_q(p_8,6) = \max\{0.884,f_q(p_9,12)\} = 0.884; f_c = 2.1;$
	...
	$f_q(p_8,2) = \max\{f_q(p_9,8)0.94\} = 0.893; f_c = 4.0;$
$P_7$	$f_q(p_7,8) = \max\{f_q(p_8,10)0.92\} = 0.770; f_c = 2.1;$ $f_q(p_7,7) = \max\{f_q(p_8,9)0.92\} = 0.788; f_c = 2.4;$ $f_q(p_7,6) = \max\{f_q(p_8,8)0.92\} = 0.796; f_c = 2.7;$
	...
	$f_q(p_7,2) = \max\{f_q(p_8,4)0.92,f_q(p_8,9)0.95\} = 0.822; f_c = 3.2;$ $f_q(p_7,1) = \max\{f_q(p_8,3)0.92f_q(p_8,8)0.95\} = 0.822; f_c = 4.5;$ $f_q(p_7,0) = \max\{f_q(p_8,2)0.92f_q(p_8,7)0.95\} = 0.840; f_c = 5.4;$

According to the DVSP algorithm, intervals  $A$  and  $B$  are combined, and the global Pareto optimal production parameter solution set is obtained using the inter-segment accumulation method. Under the condition that the total time constraint is satisfied, the global Pareto optimal solution set is  $R_m = R_A + R_B = \{S, p_1(t_{11}), p_{[2-5]}(t_{23},t_{32},t_{41},t_{52}), p_6(t_{61}), p_7(t_{71}), p_8(t_{82}), p_9(t_{92}), E\}$ . Final cumulative production quality:  $f_q(S-E) = 0.869$ ; the cumulative time:  $f_t(S-E) = 18 + 14 = 32$  days; the cumulative cost of  $f_c(S-E) = 7.8 + 5.0 = 12.8$ .

### 5. Comparative Performance Analysis of Algorithms DVSP

#### 5.1. Algorithm Comparison

Figure 7 shows a comparison of the production quality achieved by the DVSP algorithm, the MCP [2], and the PCP-B2 [21] as the number of task nodes increases. The results demonstrate that the production quality achieved by the DVSP algorithm is consistently higher than that of the other two algorithms, while also achieving shorter production cycle times compared to the MCP and PCP-B2 algorithms. This indicates that the DVSP algorithm is more effective in optimizing production parameters and achieving a better balance between production quality and time in the experimental simulation environment.

To ensure a fair and relevant comparison experiment, all algorithms used the same set of task and service sets. After the equivalent parameter calculation for different algorithms, the optimization results of the MCP algorithm were obtained, as follows:  $f_q(p_1,3) = 0.925$ ,  $f_t(p_1,3) = 20$  days,  $f_c(p_1,3) = 8.1$  in interval  $A$ ; the optimal set in interval  $B$  is  $f_q(p_7,3) = 0.875$ ,  $f_t(p_7,3) = 13$  days,  $f_c(p_7,3) = 5.5$ . Therefore, the final production time, cost, and quality after the whole process optimization are  $f_q = 0.809$ ,  $f_t = 33$  days,  $f_c = 13.6$ . Similarly, the optimization results of the PCP-B2 algorithm are as follows: in the range of  $A$   $f_q(p_1,2) = 0.912$ ,  $f_t(p_1,2) = 17$  days,  $f_c(p_1,2) = 9.0$ , interval  $B$   $f_q(p_7,2) = 0.923$ ,  $f_t(p_7,2) = 16$  days,  $f_c(p_7,2) = 5.3$ , The final production time, cost, and quality after optimizing the whole process are  $f_q = 0.841$ ,  $f_t = 33$ , and  $f_c = 14.3$ , respectively. The same calculation was carried out for the improved NSGA-II [12] algorithm, and the relevant data pairs are shown in Table 4.

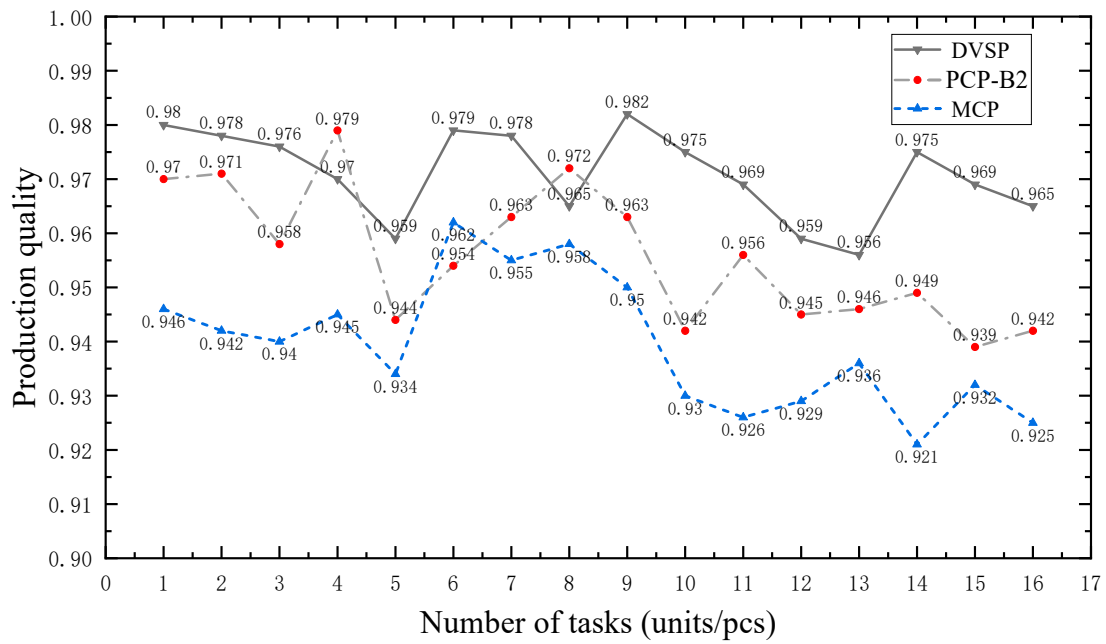


Figure 7. Algorithm effect comparison chart.

Table 4. Comparison of the results of different algorithms.

Algorithm	Interval A			Interval B			Total Interval		
	$f_q$	$f_t$	$f_c$	$f_q$	$f_t$	$f_c$	$f_q$	$f_t$	$f_c$
MCP	0.925	20	8.1	0.875	13	5.5	0.809	33	13.6
NSGA-II	0.882	17	8.2	0.910	15	4.7	0.802	32	12.9
PCP-B2	0.912	17	9.0	0.923	16	5.3	0.841	33	14.3
DVSP	0.926	18	7.8	0.939	14	5.0	0.869	32	12.8

The optimization of production quality and time can significantly affect the cost of industrial production due to the large number of products involved. In comparison, it is observed that the DVSP optimization algorithm proposed in this paper achieves a higher final optimized production quality while satisfying the constraints, compared to traditional single-objective algorithms. Additionally, the production cycle time is relatively shorter compared to the MCP and PCP-B2 algorithms. Thus, in the experimental simulation environment, the DVSP is more effective.

### 5.2. Analysis of Factors Affecting Algorithm Performance

After conducting research and analysis, it was determined that the primary factors affecting the algorithm include the number of distinct inspection stations  $S_m$ , the different times of constraint restrictions  $R_t$ , and the number of task nodes. Thus, this study focuses on these three key factors to evaluate the optimization performance of the algorithm in terms of production quality.

#### 5.2.1. Effect of Different Numbers of Testing Stations on the Accuracy of the Algorithm

To investigate how the algorithm’s performance is affected by different numbers of detection stations, various numbers were tested, and the results are presented in Figure 8. The figure shows the impact of different numbers of detection stations on production quality, and it is apparent that the production quality of PCP-B2 remains steady, but the algorithm’s production time exceeds the specified delivery date, making it impractical. In contrast, both MCP and the proposed DVSP algorithms exhibit similar results. When five detection stations are used, the production quality of DVSP is 0.961, while that of MCP is

0.929. Notably, the final results of the DVSP algorithm outperform those of the traditional critical path algorithm, and the overall production quality improves by 2.6%.

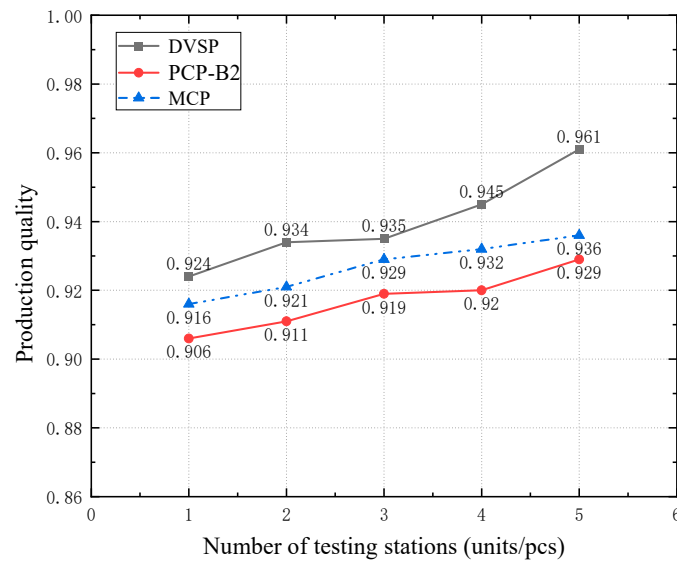


Figure 8. Effect of the number of detection stations on the accuracy of the algorithm.

### 5.2.2. Effect of Different Limiting Constraints on Time Conditions of the Algorithm

To investigate the impact of the finite-time  $R_t$  on the DVSP algorithm, we selected different finite times as variables and obtained the relevant  $R_t$  sets:  $\{(1 + 10\%), (1 + 15\%), (1 + 20\%), (1 + 25\%), (1 + 30\%)\}$ . Simultaneously, the study selected three different numbers of tasks as research variables. Figure 9 illustrates the effect of different confinement times on the accuracy of the DVSP algorithm. As depicted in the figure, the limited time significantly affects the execution performance of DVSP. The accuracy rate increases as the limit time increases, while still satisfying the production cost. The specific data changes are demonstrated in Figure 9.

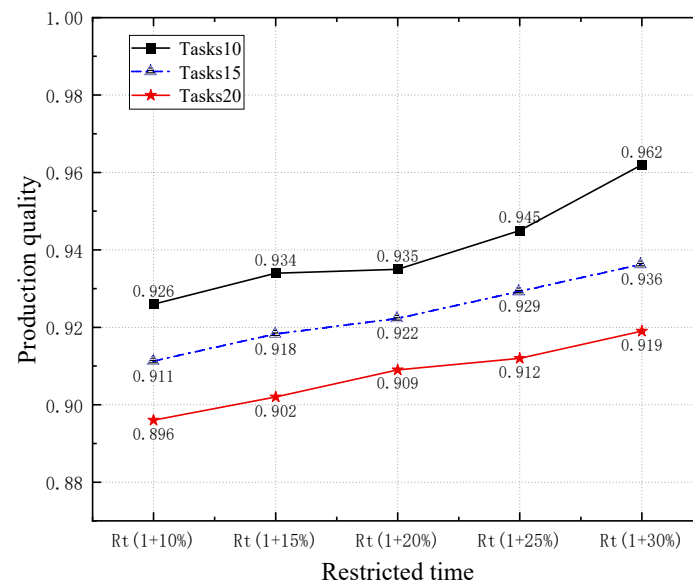


Figure 9. Effect of limiting time  $R_t$  on the accuracy of the example algorithm.

### 5.2.3. Effect of Different Numbers of Nodes P on the Performance of the Algorithm

Considering the impact of varying numbers of nodes on different algorithms, the experiment selected the number of nodes as a variable. Figure 10 illustrates how the accuracy of the algorithm is affected by different node numbers. As shown in the figure, the

DVSP, MCP, and PCP-B2 algorithms exhibit significant variations in execution performance based on the number of nodes. As the number of nodes increases, the accuracy gradually decreases while still satisfying the constraints. DVSP exhibits an overall higher optimization effect than the other two algorithms, with an average increase of 2.5% in total production quality. The trend of the influence is demonstrated in Figure 10.

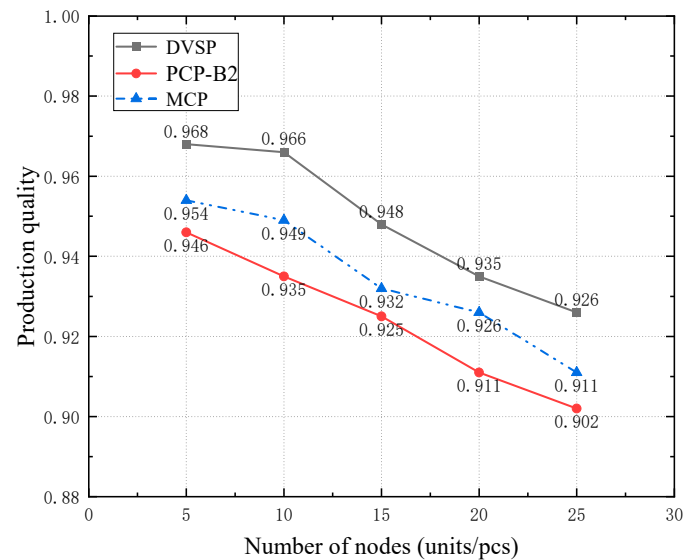


Figure 10. Effect of the number of production nodes on the accuracy rate.

## 6. Conclusions

This paper proposes a multi-objective workflow optimization algorithm based on a dynamic virtual stage pruning strategy to address the issue of mismatch between production time and quality in complex multi-stage nonlinear production processes. By analyzing the logical sequence relationship between tasks, a virtual workflow model is established and a pruning strategy is proposed to eliminate the indirect constraint relationship between tasks. The virtual hierarchical strategy is used to divide the task node set, and virtualization technology is utilized to generate virtual nodes and a virtual workflow graph. The Pareto optimal service set is calculated through backward iteration in stages, the optimal path is determined by forward scheduling, and the global optimal solution is obtained through algorithm integration. Compared to several algorithms, DVSP is found to effectively address the optimization problem of multi-stage nonlinear production processes, achieve effective balance between production quality, time, and cost, and improve enterprise production efficiency.

Currently, the DVSP algorithm has demonstrated significant optimization effects for small-scale production scheduling. The next step could involve focusing on intensive production and optimizing scheduling through multi-plant cooperation.

**Author Contributions:** Conceptualization, Z.L. and S.T.; methodology, Z.L.; software, S.T.; validation, Z.L., X.L. and S.T.; formal analysis, S.T.; investigation, X.L.; resources, H.X.; data curation, J.L.; writing—original draft preparation, S.T.; writing—review and editing, Z.L.; visualization, S.T.; supervision, Z.L.; project administration, J.L.; funding acquisition, Z.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Heilongjiang Provincial Natural Science Foundation of China (No. LH2021F030).

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** The data that support the findings of this study are available within the article.

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



## References

1. Han, Y.; Gong, D.; Jin, Y.; Pan, Q. Evolutionary multiobjective blocking lot-streaming flow shop scheduling with machine breakdowns. *IEEE Trans. Cybern.* **2017**, *49*, 184–197. [[CrossRef](#)] [[PubMed](#)]
2. Xie, Z.; Yang, J.; Zhou, Y.; Zhang, D.; Tan, G. A dynamic critical path multi-product manufacturing scheduling algorithm based on process sets. *J. Comput. Sci.* **2011**, *34*, 7–21.
3. Strnad, D.; Kohek, Š. Constrained Multi-Objective Optimization of Simulated Tree Pruning with Heterogeneous Criteria. *Appl. Sci.* **2021**, *11*, 10781. [[CrossRef](#)]
4. Ahmadi, E.; Zandieh, M.; Farrokh, M.; Emami, S.M. A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms. *Comput. Oper. Res.* **2016**, *73*, 56–66. [[CrossRef](#)]
5. Luo, Z.; Zhu, Z.; Xie, Z.; Sun, G. Research on multi-objective optimization algorithm for differential pollination workflow of flowers for cloud computing. *J. Electron.* **2021**, *49*, 470–476.
6. Han, S.; Zhu, K.; Zhou, M.C. A Novel Multiobjective Fireworks Algorithm and Its Applications to Imbalanced Distance Minimization Problems. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 1476–1489. [[CrossRef](#)]
7. Wang, Y.; Zuo, X. An effective cloud workflow scheduling approach combining PSO and idle time slot-aware rules. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 1079–1094. [[CrossRef](#)]
8. Abed-Elguni, B.H.; Alawad, N.A. Distributed Grey Wolf Optimizer for Scheduling of Workflow Applications in Cloud Environments. *Appl. Soft Comput.* **2021**, *102*, 107113. [[CrossRef](#)]
9. Prakash, V.; Bawa, S.; Garg, L. Multi-Dependency and Time Based Resource Scheduling Algorithm for Scientific Applications in Cloud Computing. *Electronics* **2021**, *10*, 1079–1094. [[CrossRef](#)]
10. Wang, H.; Sheng, B.; Lu, Q.; Yin, X.; Zhao, F.; Lu, X.; Luo, R.; Fu, G. A novel multi-objective optimization algorithm for the integrated scheduling of flexible job shops considering preventive maintenance activities and transportation processes. *Soft Comput.* **2021**, *25*, 2863–2889.
11. Ma, X.; Gao, H.; Xu, H.; Bian, M. An IoT-based task scheduling optimization scheme considering the deadline and cost-aware scientific workflow for cloud computing. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 249. [[CrossRef](#)]
12. Duan, J.; Wang, J. Energy-efficient scheduling for a flexible job shop with machine breakdowns considering machine idle time arrangement and machine speed level selection. *Comput. Ind. Eng.* **2021**, *161*, 107677. [[CrossRef](#)]
13. Dai, M.; Tang, D.; Giret, A.; Salido, M.A. Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints. *Robot. Comput.-Integr. Manuf.* **2019**, *59*, 143–157. [[CrossRef](#)]
14. Zhou, N.; Lin, W.; Feng, W.; Shi, F.; Pang, X. Budget-deadline constrained approach for scientific workflows scheduling in a cloud environment. *Clust. Comput.* **2020**, 1–15. [[CrossRef](#)]
15. Ndamlabin, M.b.o.u.l.a.; Kamla, V.; Tayou Djamegni, C. Cost-time trade-off efficient workflow scheduling in cloud. *Simul. Model. Pract. Theory* **2020**, *103*, 102107. [[CrossRef](#)]
16. Yuan, Y.; Li, X.; Wang, Q.; Zhu, X. Deadline division-based heuristic for cost optimization in workflow scheduling. *Inf. Sci.* **2009**, *179*, 2562–2575. [[CrossRef](#)]
17. Petchrompo, S.; Coit, D.W.; Brintrup, A.; Wannakrairot, A.; Parlikad, A.K. A review of Pareto pruning methods for multi-objective optimization. *Comput. Ind. Eng.* **2022**, *167*, 108022. [[CrossRef](#)]
18. Arabnejad, V.; Bubendorfer, K.; Ng, B. Budget and deadline aware e-science workflow scheduling in clouds. *IEEE Trans. Parallel Distrib. Syst.* **2018**, *30*, 29–44. [[CrossRef](#)]
19. Luo, Z.Y.; Wang, J.Y.; Xie, Z.Q. Research on nonlinear manufacturing process multi-objective optimization algorithm with three-layer virtual workflow model. *J. Autom.* **2022**, *48*, 896–908.
20. Liu, R.; Li, J.; Liu, J.; Jiao, L. A survey on dynamic multi-objective optimization. *Chin. J. Comput.* **2020**, *43*, 1246–1278.
21. Wu, F.; Wu, Q.; Tan, Y.; Li, R.; Wang, W. PCP-B2: Partial critical path budget balanced scheduling algorithms for scientific workflow applications. *Future Gener. Comput. Syst.* **2016**, *60*, 22–34. [[CrossRef](#)]
22. Xie, Z.; Wang, Q. Flexible Integrated Scheduling Algorithm Based on Reverse Order Layer Priority. *J. Electron. Inf. Technol.* **2022**, *44*, 1554–1562.
23. Quan, Z.; Wang, Y.; Ji, Z. Multi-objective optimization scheduling for manufacturing process based on virtual workflow models. *Appl. Soft Comput.* **2022**, *122*, 108786. [[CrossRef](#)]
24. Hosseinzadeh, M.; Ghafour, M.Y.; Hama, H.K.; Vo, B.; Khoshnevis, A. Multi-objective task and workflow scheduling approaches in cloud computing: A comprehensive review. *J. Grid Comput.* **2020**, *18*, 327–356. [[CrossRef](#)]
25. Zhang, L.; Zhou, L.; Salah, A. Efficient scientific workflow scheduling for deadline-constrained parallel tasks in cloud computing environments. *Inf. Sci.* **2020**, *531*, 31–46. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.