

## Article

# Cost Optimal Production-Scheduling Model Based on VNS-NSGA-II Hybrid Algorithm—Study on Tissue Paper Mill

Huanhuan Zhang <sup>1</sup> , Jigeng Li <sup>1</sup>, Mengna Hong <sup>1,2</sup>, Yi Man <sup>1</sup>  and Zhenglei He <sup>1,\*</sup> <sup>1</sup> School of Light Industry and Engineering, South China University of Technology, Guangzhou 510640, China<sup>2</sup> China Singapore International Joint Research Institute, Guangzhou 510006, China

\* Correspondence: hezhenglei@scut.edu.cn

**Abstract:** With the development of the customization concept, small-batch and multi-variety production will become one of the major production modes, especially for fast-moving consumer goods. However, this production mode has two issues: high production cost and the long manufacturing period. To address these issues, this study proposes a multi-objective optimization model for the flexible flow-shop to optimize the production scheduling, which would maximize the production efficiency by minimizing the production cost and makespan. The model is designed based on hybrid algorithms, which combine a fast non-dominated genetic algorithm (NSGA-II) and a variable neighborhood search algorithm (VNS). In this model, NSGA-II is the major algorithm to calculate the optimal solutions. VNS is to improve the quality of the solution obtained by NSGA-II. The model is verified by an example of a real-world typical FFS, a tissue papermaking mill. The results show that the scheduling model can reduce production costs by 4.2% and makespan by 6.8% compared with manual scheduling. The hybrid VNS-NSGA-II model also shows better performance than NSGA-II, both in production cost and makespan. Hybrid algorithms are a good solution for multi-objective optimization issues in flexible flow-shop production scheduling.

**Keywords:** multi-objective optimization; scheduling; flexible flow-shop; agile manufacturing; NSGA-II



**Citation:** Zhang, H.; Li, J.; Hong, M.; Man, Y.; He, Z. Cost Optimal Production-Scheduling Model Based on VNS-NSGA-II Hybrid Algorithm—Study on Tissue Paper Mill. *Processes* **2022**, *10*, 2072. <https://doi.org/10.3390/pr10102072>

Academic Editors: Mohd Azlan Hussain and Jui-Yuan Lee

Received: 16 August 2022

Accepted: 4 October 2022

Published: 13 October 2022

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



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

## 1. Introduction

With the continuous growth of customer demand for product customization, efficient production scheduling plays a crucial role in the rapid response of manufacturing enterprises to the changing global market [1]. The complexity of the production process increases due to product customization and diversification. Since the production process requires the cooperation of multiple links, the improvement of complexity directly reduces the production efficiency, thus significantly increasing the production cost. How to shorten the production cycle, reduce inventory, and improve production efficiency through reasonable production scheduling under limited production capacity is the most effective way to reduce production costs [2–7].

A production plan is the decision-making process for the reasonable allocation of limited resources within a period of time to complete a series of work, and it is also the basis for organizing and guiding production activities in a planned way. The basic goal is to ensure the effective and efficient use of available resources [8]. Due to the complexity of the flexible flow-shop scheduling problem (FFSP), many achievements have been made in recent years [9]. The FFSP is a multi-stage production process, which is connected by two or more stages in series. There is at least one machine in each stage, and there are multiple machines in at least one stage. All jobs need to run through part or all of the production stages in a certain order before completion. Moreover, each stage needs to select a machine to handle the corresponding processing jobs. In the real production process, the flexible flow-shop scheduling mainly has the following problems. On the one hand, for a

considerable part of the production process, the reduction in its cost is not positively related to the optimization of material consumption, energy consumption or other production indicators (such as production scheduling). Sometimes it is even a negative correlation. How to properly balance the relationship between multiple optimization indicators requires a reasonable design in combination with the production process. On the other hand, in the inevitable product switching process, due to lack of scheduling optimization, a large amount of unnecessary waste is caused.

The production-scheduling problem of tissue paper mills (TPM) is a typical complex two-stage FFSP [10]. The production process of TPM can be simplified into two stages: one is the papermaking stage, and the other is the converting stage. The production process of the papermaking stage is continuous, which can be briefly described as pulp preparation and papermaking. The production process of the converting stage is also continuous, which includes the selection of winder, cutter, weighing, and packaging. In the production process of different products, there is a variety of equipment or production lines to choose in the papermaking stage and converting stage. As for TPM, their products have the characteristics of fast-moving consumer goods, and their product types and specifications are varied. Its product categories can reach dozens to hundreds. Due to the great influence of the market, the production plan changes frequently. Product switching not only increases the energy consumption and material consumption in the production process, but also brings great challenges to the papermaking stage that requires stable production. Thus, optimizing production scheduling is very important for TPM. At present, TPM mostly rely on the experience of production-scheduling staff to make a production-scheduling task list. As a result, production scheduling is inefficient, and products cannot even be delivered as planned. Although there are few types of production-scheduling software, they are often implemented based on the linear model or dispatch rules. These kinds of software cannot be optimized for multiple purposes at the same time. Moreover, for the more complicated production-scheduling problem, these kinds of software cannot guarantee the optimization of production-scheduling results.

To solve the problem of efficient production optimization, reduce production costs and improve production efficiency, this paper takes a TPM as the research object and establishes a FFSP model with transportation constraints, set-up constraints, and delivery time constraints. The proposed model could optimize production costs by arranging reasonable production scheduling while considering the optimization of makespan. Moreover, a VNS-NSGA-II algorithm is proposed to solve the proposed production-scheduling model. The application availability of the proposed model is verified by real-time data collected from an actual TPM.

## 2. Literature Review

FFSP is a general form of classical flow shop scheduling problem. It involves sorting flexibility and processing flexibility, which is more consistent with the actual production environment and more practical in practical applications). This paper studies the production process scheduling problem of a kind of flexible flow shop, and proposes a method to optimize the total cost and makespan. The total cost includes processing cost, transportation cost, set-up cost, inventory cost and tardiness cost. The proposed scheduling problem is about multi-objective optimization.

In recent years, many methods for solving multi-objective optimization problems (MOP) have been proposed, which can be divided into three categories. The first category is the non-inferior solution set method, which is suitable for small-scale problems. However, it is inefficient when solving large-scale problems [11]. The second category is the evaluation function method, which contains two ideas. The first idea is to reconstruct MOP into a function, namely the evaluation function. The evaluation function can transform the MOP into a single-objective optimization problem. The common evaluation functions include linear weighting method, division, and multiplication method. This approach has two obvious shortcomings: First, a small change of the weight coefficient could cause a

significant change in the target vector. Second, the significant changes in different weight parameters may lead to the similarity of solutions, which leads to the poor diversity of solutions. The second idea is to transform MOP into multiple single-objective optimization problems, and then optimize them one by one. The disadvantage is that the solution obtained is a finite solution and the balance between the solutions is poor [12]. The last category is the intelligent optimization algorithm, in which NSGA-II is one of the most famous multi-objective evolutionary algorithms (MOEA) [13].

Compared with the traditional optimization algorithm, the performance of the NSGA-II is better when solving the NP-Hard problem [10]. In particular, when the problem size is large, the difference effect will be more obvious. This paper has two objective functions, which belong to MOP. Compared with the single-objective optimization problem, MOP is more complicated, and the complexity is reflected in the relationship between the optimization objectives. In practice, there are often conflicts between multiple goals that need to be optimized; therefore, it is difficult to optimize all optimization goals at the same time. In other words, there is a certain solution that may be optimal for one goal, but for another goal that needs to be optimized, it may be sub-optimal solution or even worse. NSGA-II is suitable for dealing with optimization problems with multiple conflicting goals and large and complex search space. Thus, NSGA-II has become one of the most widely used MOEAs in recent years [13].

To the optimal FFSP, Tran et al. [14] studied the dual-objective (makespan and the total tardiness time of job) FFSP based on the limited intermediate buffer, and proposed a hybrid water-flow algorithm that combines local and global neighbourhood structures. Moslehi et al. [15] proposed a hybrid solution algorithm combining local search algorithm and particle swarm optimization (PSO) for the flexible job shop scheduling problem when considering the priority of the job. They also evaluated the efficiency of the proposed algorithm, and the evaluation results showed that the method they proposed runs faster and has better results. Luo et al. [14] studied FFSP considering production efficiency constraints, power cost constraints, and time-of-use power price constraints to minimize power costs. In order to better solve this problem, they proposed to integrate the list scheduling the algorithm, a right-shift procedure and the algorithm. Allahverdi et al. [16] studied the no-wait flow shop scheduling problem with a finite manufacturing period, and considered completion time and tardiness as two performance indicators. A hybrid algorithm combining the insertion algorithm and the simulated annealing algorithm is proposed to optimize the total delay. Zeng et al. [10] studied production-scheduling problems of tissue paper mills with minimal total energy consumption and material wastage. Aiming for the FFSP, Dai et al. [12] considered transportation constraints and established an optimization model for energy consumption and completion time. They solved the model through the proposed improved GA, revealing the relationship between transportation energy consumption and completion time. Lu et al. [17] discussed the optimization of energy consumption and makespan from the perspective of energy efficiency for the permutation flow shop. A mathematical model including processing time, idle time, and transmission time is established, which narrows the gap between theoretical research and practical application. On the one hand, the earlier the order is completed, the lower the probability of delay. On the other hand, the earlier the order is completed, the higher storage cost of the warehouse is. For enterprises with high storage costs, optimizing production scheduling for order completion time is important. Chandra et al. [18] solved the permutation flow shop scheduling problem with tardiness penalty, job deadline penalty and early penalty, which made the completion time of production jobs more consistent with the delivery time, thereby improving production efficiency. Schaller et al. [19] studied the scheduling problem with non-mandatory idle time in order to make the production jobs be processed as close as possible to the given delivery date. A total cost model is also established, and its cost includes the storage cost of the work in progress (WIP) and the penalty cost for late delivery of the product. The WIP inventory costs has been also considered in many recent works. For example, in order to obtain a

low-cost production planning and scheduling scheme, Mishra et al. [20] established a total cost model consisting of the WIP storage costs and late delivery penalty costs.

According to the above results, the research on production scheduling with energy consumption cost as the optimization goal is quite sufficient. The production cost minimization scheduling of flexible flow-shop is relatively less due to the more complicated cost issues. The cost of the production process is not only directly related to material consumption and energy consumption. It is also affected by other production factors, such as transportation costs, storage costs, and delay costs that cannot be delivered on time. There are complex coupling relationships between some costs. However, most studies often only consider some of these costs, but lack the comprehensive consideration of cost in the production process. Moreover, for the high-complexity two-stage FFSP, the cost-minimizing scheduling of other manufacturing systems is difficult to apply to this scenario.

### 3. Production-Scheduling Multi-Objective Optimization Model

#### 3.1. Cost Model

For different types of production-scheduling problems or the same type of scheduling problems, the cost components and proportions of different enterprises are not the same. Before establishing a cost-minimized production-scheduling model, the cost of the enterprise needs to be analyzed. Thus, it is necessary to analyze the cost of TPM.

The production cost of TPM mainly includes the following aspects: (1) Processing cost: it includes the energy consumption cost of steam and electricity for the papermaking process. (2) Transportation cost: in the actual production process, the distance between the paper workshop and the converting workshop is usually far. Thus, the raw paper produced in the papermaking process needs to be transported to the packaging workshop, thus generating transportation costs. (3) Set-up cost: when the current job and the next job are of different types, the machine needs to be adjusted to adapt to different products, thus causing the loss cost of power and materials. (4) Inventory cost: inventory cost refers to the cost of products occupying inventory due to the early completion of orders. (5) Tardiness cost: the default cost caused by delayed delivery of the order is called delay cost.

##### 3.1.1. Processing Cost

In the tissue paper enterprise, there are multiple production lines to choose in the first stage (papermaking stage) and the second stage (converting stage), and the production speeds of the production lines in different stages are not the same. Selecting different production lines for the same product needs different processing time, which results in different processing costs. Thus, how to choose a reasonable production path to reduce processing costs is a major optimization goal. The processing cost of the production process is as shown in Equation (1).

$$PC = \sum_{j=1}^m \sum_{i=1}^n O_{i,j} p_{i,j} \quad (1)$$

where  $PC$  is the total processing cost,  $i$  is the job number,  $j$  is the machine number,  $n$  is the number of jobs waiting for machine processing, and  $m$  is the number of machines available for processing jobs.  $O_{i,j}$  is a decision variable, and its value is either 0 or 1. If its value is 1, it means that the machine can handle the job; If it is 0, it means that the job is not allowed to be processed on the equipment.  $p_{i,j}$  is the processing cost of the  $i$ -th job in the  $j$ -th machine.

##### 3.1.2. Transportation Cost

The papermaking stage and the converting stage often contain many processing steps, resulting in a relatively long production line. Taking into account the capital and development, the factory's floor space is limited, not random expansion. In the actual factory layout, in order to make better use of the limited land area and improve space utilization, there is a certain distance between the papermaking stage and the converting stage. That is to say, after the completion of the papermaking stage, the converting stage cannot be

processed immediately. It takes a certain time to transport semi-finished products to the converting stage before processed, resulting in a transportation cost. The transportation cost is related to the distance between the two-stage workshops. If the distance between the two workshops is larger, the transportation cost will be higher. Thus, to the major optimization goal is to choose a reasonable production path to reduce transportation costs. The transportation cost of the production process is as shown in Equation (2).

$$TC = \sum_{i=1}^n f_t \left( \sum_{j=1}^{m_1} O_{i,j}, \sum_{j=1}^{m_2} jO_{i,j} \right) JS_i \quad (2)$$

where  $TC$  is the total transportation cost,  $JS_i$  is the size of the  $i$ -th job,

$m_1$  represents all processing machine available in the papermaking stage and  $m_2$  represents all processing machine available in the converting stage,  $f_t \left( \sum_{j=1}^{m_1} O_{i,j}, \sum_{j=1}^{m_2} jO_{i,j} \right)$  is the additional transportation cost required to switch production processes per unit product.

### 3.1.3. Set-Up Cost

When the type of the former processing job is inconsistent with that of the latter processing job, product switching is required. On the one hand, the equipment needs to adjust parameters to meet the production requirements. The adjustment time takes between ten minutes to several hours, which may lead to increasing completion time, then increasing the cost. On the other hand, when the product type is not consistent, the ratio of raw materials is not the same. Before processing, it is necessary to clean off the unused slurry which could lead to waste of raw materials and high costs. Thus, how to choose a reasonable production path to reduce the set-up cost is a major optimization goal. The set-up cost of the production process is as shown in Equation (3).

$$SC = \sum_{j=1}^m \sum_{i=2}^{n_j} sp_{i-1,i}^j \quad (3)$$

where  $SC$  is the total set-up cost,  $n_j$  is the number of processing jobs allocated to the  $j$ -th machine and  $sp_{i-1,i}^j$  represents the set-up cost when the  $(i-1)$ -th job changes to the  $i$ -th job in the  $j$ -th machine.

### 3.1.4. Inventory Cost

The delivery time of different jobs is not the same. If the job with a late delivery date is arranged too early, the product storage time will be too long, resulting in increasing the storage costs. In addition, it is possible to delay the delivery of some tasks with early delivery dates. Thus, how to choose a reasonable production plan to reduce inventory costs is a major optimization goal. The inventory cost of the production process is shown in Equation (4).

$$IC = \sum_{i=1}^n \alpha_i \max(0, d_i - e_i) JS_i \quad (4)$$

where  $IC$  is the total inventory cost in advance completion,  $\alpha_i$  represents the early penalty factor,  $d_i$  represents the delivery date of the  $i$ -th job, and  $e_i$  represents the completion time of the  $i$ -th job.

### 3.1.5. Tardiness Cost

If some jobs with early delivery date are arranged too late, the delivery may be delayed, and the image of the enterprise may be affected. It also has to pay a certain amount of liquidated damages. In addition, when there are many processing jobs, the delay is inevitable. Different production-scheduling schemes could delay different number of jobs. Thus, how to choose a reasonable production plan to reduce the cost of tardiness is a major optimization goal. The delayed cost of the production process is shown in Equation

(5). It should be noted that the delay penalty factor  $\beta$  should be set larger than the advance penalty factor  $\alpha$ .

$$DC = \sum_{i=1}^n \beta_i \max(0, d_i - e_i) JS_i \quad (5)$$

where  $DC$  is the total tardiness cost,  $\beta_i$  is the delay penalty factor.

### 3.2. Production-Scheduling Model

The FFSP can be described as follows: suppose that  $n$  jobs are processed in  $m$  stages, and at least two or more parallel machines are available in each processing stage. For the two-stage FFSP of TPM, the first stage includes  $m_1$  parallel production lines, and each job can select any production lines for processing. The second stage includes  $m_2$  parallel production lines and each job can select any production lines for processing. Each job must be processed in the first stage firstly before it can be processed in the second stage. In addition, the processing time of different production lines could be the same or different. This paper takes the makespan and production cost as two optimization goals. The specific production-scheduling model is as follows.

$$\left\{ \begin{array}{l} \min(\max(e_i)) \\ \min(\text{Cost}) \end{array} \right\} \quad (6)$$

$$\sum_{j=1}^{m_1} O_{i,j} = 1; i = 1, 2, \dots, n \quad (7)$$

$$\sum_{j=1}^{m_2} O_{i,j} = 1; i = 1, 2, \dots, n \quad (8)$$

$$S_{i,j} \geq E_{i-1,j}; i = 1, 2, \dots, n; j = 1, 2, \dots, m \quad (9)$$

$$m e_i = \sum_{j=1}^{m_2} \sum_{k=1}^{N_j} E_{k,j} f_a(i, k); i = 1, 2, \dots, n \quad (10)$$

$$f_a(i, k) = \left\{ \begin{array}{l} 0 \\ 1 \end{array} \right\}; i = 1, 2, \dots, n; k = 1, 2, \dots, n; i \neq k \quad (11)$$

$$\sum_{j=1}^{m_2} S_{i,j} O_{i,j} - \sum_{j=1}^{m_1} S_{i,j} O_{i,j} \geq f_b \left( JS_i, \sum_{j=1}^{m_1} j O_{i,j}, \sum_{j=1}^{m_2} j O_{i,j} \right); i = 1, 2, \dots, n \quad (12)$$

$$\text{Cost} = PC + TC + SC + IC + DC \quad (13)$$

where  $S_{i,j}$  is the completion time of the  $i$ -th job on the  $j$ -th machine,  $E_{i,j}$  is the completion time of the  $(i-1)$ -th job on the  $j$ -th machine. Equation (6) shows the two optimization objectives of this paper, namely, minimizing the makespan and the total cost. Constraint (7) guarantees that all processing jobs must go through the first stage of processing, and each job can only select one of all available equipment in the first stage. Constraint (8) represents that the  $i$ -th job can only be processed by any machines in the second stage and must be processed once. Constraint (9) stipulates that at a certain moment, a machine cannot handle two processing jobs simultaneously. Equation (10) is used to calculate the completion time of the job. The variable  $f_a(i, k)$  has two values in Constraint (11): a value of 1 indicates that the type of the  $i$ -th job is the same as the  $k$ -th job, and a value of 0 represents the  $i$ -th job is different from the type of the  $k$ -th job. Formula (12) indicates that there is a time interval between the first and second stages. This time interval is not only related to the size of the job, but also to the machining speed of the machine. The second stage needs to wait for sufficient semi-finished products to be produced by the first stage machine. If the time interval is too small, it could cause frequent downtime in the second phase. If the time interval is too large, it may cause excessive intermediate inventory and increase inventory costs. Equation (13) represents the total production cost, which consists of processing cost, transportation cost, set-up cost, inventory cost, and tardiness cost.

### 3.3. Algorithm

A production-scheduling model based on MOP problem is proposed. The optimization goal is to minimize production completion time and production costs. NSGA-II has a good

global search ability in solving MOP problem. However, in the process of evolution, the search ability for each solution neighborhood is relatively weak [7]. Therefore, this research will consider using a local optimization algorithm (variable neighborhood search algorithm, referred to as VNS) in combination with NSGA-II to improve NSGA-II's ability to find the best, so as to find a better production plan and improve production efficiency. The idea of combining VNS and NSGA-II is to first use the NSGA-II algorithm for global search. Then use the VNS algorithm to further search the optimal solution generated in the evolution process, and obtain the optimal solution as much as possible.

The scheduling problem of minimizing cost and makespan in this study belongs to the class of NP-Hard problems. NSGA-II algorithm has been proposed based on Deb's Nondominated Sorting Genetic Algorithm (NSGA) [21]. NSGA-II can be described as follows: First, each generation of individuals is classified by dominance; that is, the fast non-dominated sort (Pareto ordering) with low computational complexity is adopted to reduce the complexity of the non-inferior sorting algorithm. Second, when individuals belong to the same level, individuals in the Pareto solution can be uniformly distributed by evaluating an individual's crowded distance to ensure the diversity of the population. Third, the superior traits from the father are retained and directly enter the offspring, which avoids the loss of the obtained Pareto optimal solution, thus ensuring the convergence.

VNS is a local search algorithm. The key to the VNS algorithm is to use many different neighborhoods to search [22]. First, the search is in a smaller neighborhood. If no better solution is found in the smaller neighborhood, the search will continue in the larger neighborhood. If a solution superior to the current solution is found in a smaller neighborhood, the locally optimal solution is updated. Then focus on the updated local optimal solution, and search again in its nearby neighborhood to see if a better solution can be found. Repeat this process until the termination condition of the algorithm is reached. VNS has a simple principle, good optimization effect, and the algorithm structure has nothing to do with the problem. It is suitable for various optimization problems and can be embedded in other algorithms. Thus, it is widely used in practical engineering. The basic steps of VNS used in this article are detailed below.

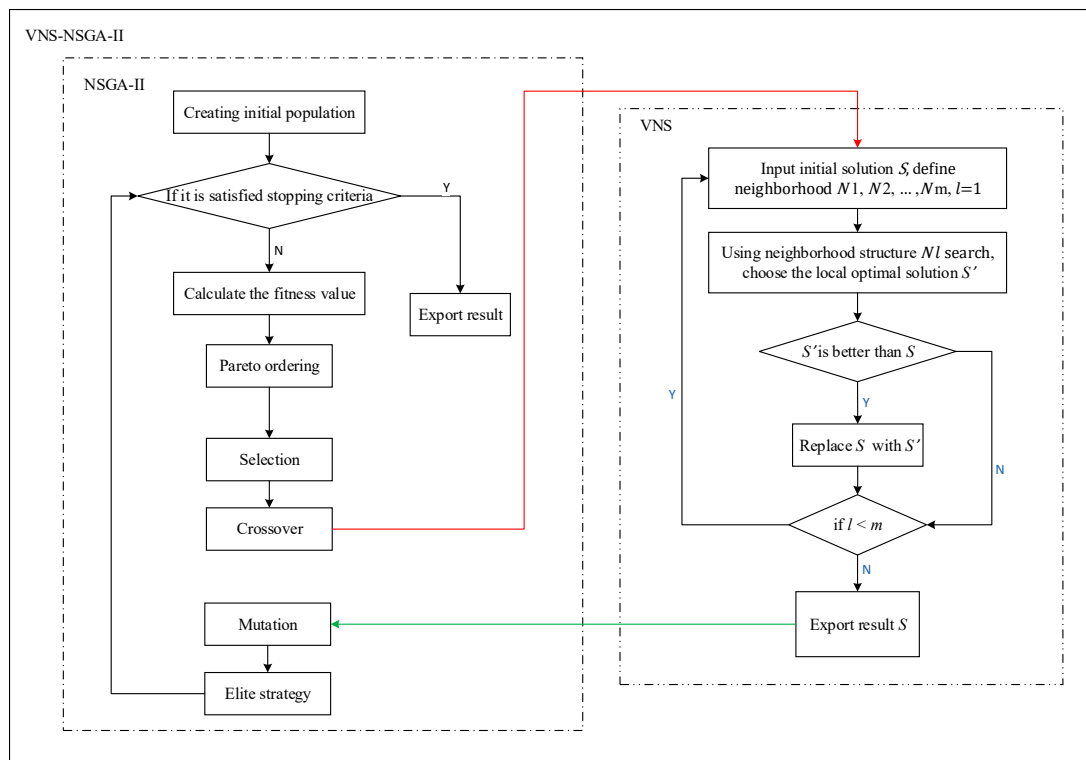
- (1) Initialization: The specific method is to select the neighborhood structure, denoted as  $S$ ,  $i = 1, 2, 3, \dots, K$ , set the maximum iteration number  $N$  of VNS and the maximum number of iterations  $M$  of each neighborhood search, and give an initial solution.
- (2) Set  $n = 1, k = 1, m = 1$ . According to the current solution, local search is conducted for the current neighborhood structure. If a better solution than the current solution can be found, the new solution found will replace the current solution; if no better solution is found, the current solution will not be updated. If  $m$  is less than or equal to  $M$ ,  $m = m + 1$ , repeat the process. When  $m$  is greater than  $M$ , if  $k < K$ , and  $k++$ , repeat this process.
- (3) If  $n < N$ , then  $n++$ , repeat step (2) until the end criterion  $n = N$  is satisfied.

The VNS algorithm has two core parts, and the construction of the neighborhood structure set is one of the core parts of the VNS algorithm. In general, the larger the neighborhood setting of the algorithm, the higher the probability that local search can obtain better solutions and the better the quality of the solutions. Thus, the more accurate the solution is. Thus, this paper designs three neighborhood structures, denoted as  $S_1$ ,  $S_2$ , and  $S_3$ , respectively. Local search is another core part of the VNS algorithm. When designing local search, this paper uses the way of exchanging genes to find new fields.  $S_1$  is produced by randomly selecting two genes on one parent chromosome. For example, the parent individual [1, 2], which has two permutations (1, 2) and (2, 1), can produce a new offspring individual.  $S_2$  is produced by randomly selecting three genes on one parent chromosome. For example, the parent individual [1, 2, 3] have a total of six sequence combinations, which are (1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 2, 1), (3, 1, 2), can produce five new offspring individuals.  $S_3$  is produced by randomly selecting four genes on one parent chromosome. For example, the parent individual [1, 2, 3, 4], which has twenty-four permutations and combinations, can produce twenty-three new offspring individuals.

In addition, to ensure the timeliness of VNS, this article sets the maximum number of iterations in the neighborhood structure and the maximum number of repetitions of the optimal solution as the end criterion.

### Hybrid VNS-NSGA-II Algorithm

Although NSGA-II algorithm is well suited to handle MOP problems with large and complex search spaces, it still has shortcomings. At each iteration, NSGA-II does not perform further searches in the neighborhood of the local optimal solution. In general, there is a greater likelihood of a better solution around the local optimal solution. Thus, this paper proposes a hybrid NSGA-II algorithm (VNS-NSGA-II) combining NSGA-II and VNS to improve optimization ability to find better solutions. Figure 1 is the VNS-NSGA-II algorithm framework. The detailed description of VNS-NSGA-II (including encoding and decoding, population Initialization, population classification, selection operator, crossover operator, mutation operator) can be described in the Supplementary Information.



**Figure 1.** The flowchart of VNS-NSGA-II.

As shown in Figure 1, the steps of VNS-NSGA-II algorithm in this paper are briefly described as follows: two offspring individuals will be generated after the completion of a crossover operation, denoted as Child 1 and Child 2. Taking these two offspring individuals as the parent, three kinds of local search and local optimal solution are firstly updated successively. Then, the domination level and crowding distance of local optimal solutions of Child 1 and Child 2 are compared and keep the optimal solution in the new population. This process is repeated until the number of new populations matches the number of old ones.

The specific process of VNS is as follows (taking Child 1 as an example).

- (1) Child 1 is used as the initial solution. A local search is performed within neighborhood  $S_1$ , and a new Child (3) is generated. The better individuals are retained by comparing the dominance levels and crowding distances of Child 1 and Child 3.

- (2) The optimal solution in the neighborhood  $S_1$  is used as the initial solution, and then the local search is performed in the neighborhood  $S_2$ . The optimal individual among the 6 individuals is retained, and the search continued in the next neighborhood.
- (3) The optimal solution in the neighborhood  $S_2$  is used as the initial solution, and then the local search is performed in the neighborhood  $S_3$ . Moreover, the best of the 24 individuals are preserved. Child 2 takes the same steps as Child 1. Repeat this process until the number of iterations of VNS is equal to the maximum number of iterations.

#### 3.4. Evaluation Method

VNS-NSGA-II based hybrid algorithm proposed in this study belongs to MOEA. At present, the quality index (QI) is mainly used to evaluate the three aspects of the performance of the MOEA solution, which are the distribution of the solution, the uniformity of the distribution, and the convergence. IGD and Hypervolume are two popular methods for comprehensive performance evaluation of the solution set, and can also evaluate the uniformity and extensiveness of the solution set. However, the IGD need to know the distribution of the Pareto optimal surface. In practice, the Pareto optimal surface is often unknown. Thus, this paper uses the Hypervolume method to evaluate the Pareto solution set. The Hypervolume method evaluates the quality of the solution set by computing the Hypervolume surrounded by the non-dominated set and the spatial reference point [23]. A larger Hypervolume value indicates better performance of diversity and convergence [24]. The detailed principles of Hypervolume calculation are shown in the Supplementary Information. Taking into account the stochastic nature of the intelligent optimization algorithm, this study used Wilcoxon's signed-rank test (WSRT) to measure the significance of difference between VNS-NSGA-II and NSGA-II. Among them, the significance level is set to 0.05, and the corresponding confidence level is set to 95%.

## 4. Results and Discussion

In order to more clearly analyze the advantages of VNS-NSGA-II, this paper compares the results of VNS-NSGA-II with the results of NSGA by using the same data. This paper designs seven groups of case studies. The first six case studies simulate production jobs by using the data generated by the actual production range. The seventh study case simulates production jobs by using real-time data collected from a real tissue paper mill. Each case study contains a minimum cost production-scheduling problem, which is solved by NSGA-II and VNS-NSGA-II, respectively. Each case study runs independently 10 times, and then analyzes and compares the results of the two algorithms.

#### 4.1. Experimental Data

The performance of VNS-NSGA-II in different scales of cost-based production-scheduling problems was tested through experiments. Each case study in this paper contains a production-scheduling problem based on cost minimization. The names, number of jobs, and product types of the seven cost-based production-scheduling case studies are shown in Table 1. In each cost-based production-scheduling case, the size of each job is randomly generated according to the actual job size of the tissue paper enterprise. The size range of jobs obeys a uniform distribution of [100,000, 500,000].

**Table 1.** Details of each production-scheduling case study.

Study Case	Numbers of Job	Numbers of Product Type
Job1	50	5
Job2	100	10
Job3	150	15
Job4	200	20
Job5	250	25
Job6	300	30
Job7	178	20

The same job may take different amounts of time to process on different machines, which may result in different processing costs. The job processing time,  $p_i$ , is calculated according to the size of the job and the speed of the machine. The speed range of each production line is shown in Tables 2 and 3, where Stage 1 represents the papermaking stage, which includes 6 production lines, donated as PL1–PL6. Stage 2 represents the converting stage, which includes 7 production lines, donated as BL1–BL7. In order to verify the impact of different setting times on the production plan, the setting time in this paper is randomly generated within a certain range. The set-up time in the papermaking stage is uniformly distributed according to [10, 60], and the set-up time of the converting stage obeys uniform distribution in the interval [10, 40]. The unit time set-up cost of each production line is shown in Tables 4 and 5. The transportation cost is closely related to the distance between the first stage and the second stage. The closer the layout between these two stages, the lower the transportation cost. Table 6 shows the cost of transporting a unit product from the first stage to the second stage. The rows represent the production lines available in the first stage, and the columns are the production lines available in the second stage. It should be noted that the electricity price used in this article is 0.7 CNY per kW.

**Table 2.** The highest speed of each production line in the papermaking stage.

Stage 1	Production Line Speed (m/min)
PL1	1000
PL2	960
PL3	1060
PL4	980
PL5	1100
PL6	1110

**Table 3.** The highest speed of each production line in the converting stage.

Stage 2	Rewinder Speed (m/min)	Small Packing Machines (bag/min)
BL1	200	215
BL2	320	264
BL3	310	254
BL4	320	160
BL5	330	274
BL6	335	280
BL7	340	285

**Table 4.** Set-up cost per unit time (h) of each production line in the papermaking stage.

Stage 1	Set-Up Costs (CNY/h)
PL1	1031
PL2	1132
PL3	1070
PL4	1232
PL5	1086
PL6	1200

**Table 5.** Set-up cost per unit time (h) of each production line in the converting stage.

Stage 2	Set-Up Costs (CNY/h)
BL1	66
BL2	77
BL3	81
BL4	57
BL5	72
BL6	74
BL7	75

**Table 6.** Transportation cost per unit of product (CNY/t).

Transportation Costs (CNY/t)	BL1	BL2	BL3	BL4	BL5	BL6	BL7
PL1	4	5	11	12	13	15	24
PL2	5	3	15	16	18	23	24
PL3	16	15	3	4	6	19	21
PL4	17	15	5	2	3	15	16
PL5	26	24	22	20	17	4	3
PL6	26	25	22	21	18	3	5

The earliness/tardiness cost is calculated by using the delay time and the earliness/tardiness penalty factor ( $\alpha_i/\beta_i$ ), which is related to job completion time and delivery time. The delivery time generation method is shown as follows [25]:  $p_i$  represents the processing time of job and  $d_i$  represents the delivery time. A number is randomly selected in the set  $T = \{0.2, 0.4, 0.6\}$ , denoted as  $t$ . A number is randomly selected in the set  $R = \{0.2, 0.95, 1.15\}$ , denoted as  $r$ . Then  $d_i$  takes a uniformly distributed integer in the interval  $[T(1 - t - 0.5r), T(1 - t + 0.5r)]$ . Because  $d_i$  needs to be bigger than or equal to  $p_i$ , ( $t, r$ ), and there could be seven possible combinations. That is, when  $t$  is 0.2 and 0.4, there are three kinds. When  $t = 0.6$ ,  $r$  can only be equal to 0.2. For the early penalty factor  $\alpha_i$  and the delayed penalty factor  $\beta_i$ ,  $\beta_i$  is randomly selected in the interval  $[0.002, 0.006]$ , and  $\alpha_i$  is randomly selected in the interval  $[0.0005, \beta_i]$ , which guarantees that the delay penalty is bigger than the early penalty.

#### 4.2. Experimental Parameter Settings

Important parameters of NSGA-II algorithm, such as population size, max iteration, crossover rate, and mutation rate, have important effects on the optimization performance of the algorithm. The important parameters of VNS algorithm include max iteration, the size of the neighborhood structure set, and the maximum number of iterations within the neighborhood structure. The smaller the number of feasible solutions in the neighborhood structure set is, and the faster the local search speed is. Maximum iteration for VNS and the number of iterations in the neighborhood structure have a direct and significant impact on the global convergence and timeliness of VNS. The important parameters of NSGA-II and VNS-NSGA-II in this experiment are shown in Table 7.

**Table 7.** Parameter for NSGA-II and VNS-NSGA-II.

NSGA-II	VNS-NSGA-II
Population size:100	Population size:100
Max iteration for NSGA-II:100	Max iteration for VNS-NSGA-II:100
Crossover rate:0.9	Crossover rate:0.9
Mutation rate:0.1	Mutation rate:0.1
	Max iteration for VNS:10
	VNS neighborhood structure set 1 sizes:2
	VNS neighborhood structure set 2 sizes:6
	VNS neighborhood structure set 3 sizes:24

#### 4.3. Analysis of Experimental Results

Considering the stochastic nature of the intelligent optimization algorithm, NSGA-II and VNS-NSGA-II run 10 times each when solving each production-scheduling study case. Then calculate the average value of the results of the 10 runs. Table 8 shows the hypervolumes of solutions and the  $p$ -value of WSRT for seven production-scheduling case studies, which were obtained by using NSGA-II and VNS-NSGA-II. As can be seen from Table 8, in each production-scheduling case study, the average hypervolume of the solution obtained using VNS-NSGA-II is bigger than that using NSGA-II. For the seven cases in Table 8, their  $p$ -values are all less than 0.005, which indicates that the results are significantly different. By comparing the size of the hypervolume value and  $p$ -value of the seven research cases, it can be found that the quality of the solution obtained by using VNS-NSGA-II is better. Therefore, VNS is feasible for improving the quality of NSGA-II solution set.

**Table 8.** The Hypervolume and  $p$ -value of Pareto solutions for 7 production-scheduling case studies obtained using two algorithms.

Study Case	NSGA-II	VNS-NSGA-II	$p$ -Value	Significant Difference
Job1	$5.14 \times 10^7$	$6.50 \times 10^7$	0.002	existence
Job2	$2.01 \times 10^8$	$2.53 \times 10^8$	0.002	existence
Job3	$2.36 \times 10^8$	$2.75 \times 10^8$	0.002	existence
Job4	$3.62 \times 10^8$	$4.40 \times 10^8$	0.002	existence
Job5	$3.70 \times 10^8$	$4.70 \times 10^8$	0.002	existence
Job6	$5.01 \times 10^8$	$6.63 \times 10^8$	0.002	existence
Job7	$2.31 \times 10^8$	$3.41 \times 10^8$	0.002	existence

This paper provides a set of bar graphs and two-dimensional scatter diagrams to show the solutions obtained by different algorithms and the Hypervolume of solutions. Figures 2 and 3 are bar graphs of the Hypervolume of the solution of NSGA-II and VNS-NSGA-II by running ten times. It can be seen from Figures 2 and 3 that on each production-scheduling problem, the Hypervolume value obtained using NSGA-II is less than VNS-NSGA-II. It shows that the solution obtained using NSGA-II is not as good as that obtained by VNS-NSGA-II. Figures 4 and 5 show the two-dimensional scatter diagrams of the solution obtained by running algorithm for ten times. In Figures 4 and 5, the closer the point position is to the origin, the higher the level of the solution. As shown in Figures 4 and 5, the orange point is closer to the origin of the coordinates than the blue point. This indicates that the Pareto rank obtained by using VNS-NSGA-II is higher. Moreover, this research combines the solution sets obtained by the two methods (VNS-NSGA-II and NSGA-II) into a new solution set. Then this article re-selects the Pareto optimal solution, and finally analyzes the proportion of the optimal solutions obtained by these two methods. Table 9 shows the percentage of the Pareto optimal solution obtained by each algorithm in the seven production-scheduling case studies as a percentage of this new Pareto optimal solution. Table 9 shows that only in Job3 and Job5, the proportion of Pareto optimal solution of NSGA-II is greater than 0, which is 6.25 and 7.29%, respectively. The Pareto optimal

solution based on VNS-NSGA-II accounts for more than 0 in all production-scheduling case studies, and the smallest proportion is 92.31%. Additionally, in Job1, Job2, Job4, Job6, and Job7, the Pareto optimal solution ratio of VNS-NSGA-II is 100%. That is to say, in this new Pareto optimal solution set, all the optimal solutions are from VNS-NSGA-II. This further illustrates that the Pareto rating of the solution obtained by VNS-NSGA-II is higher than that obtained by NSGA-II.

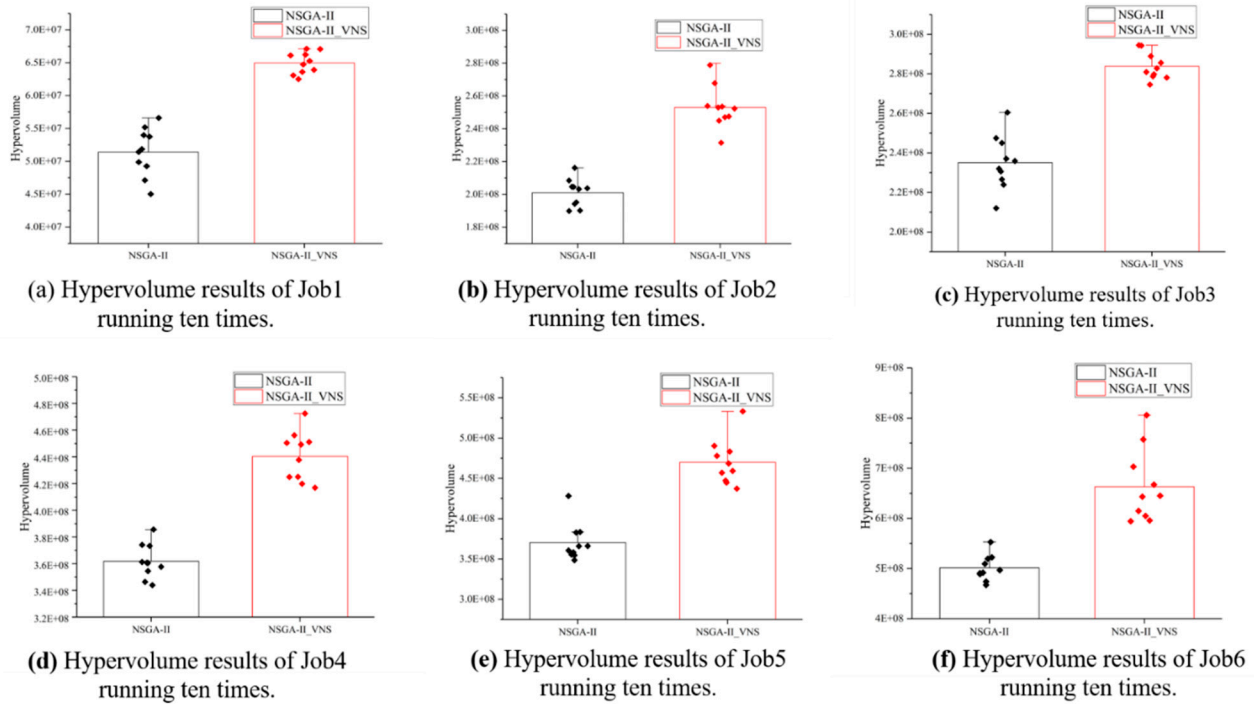
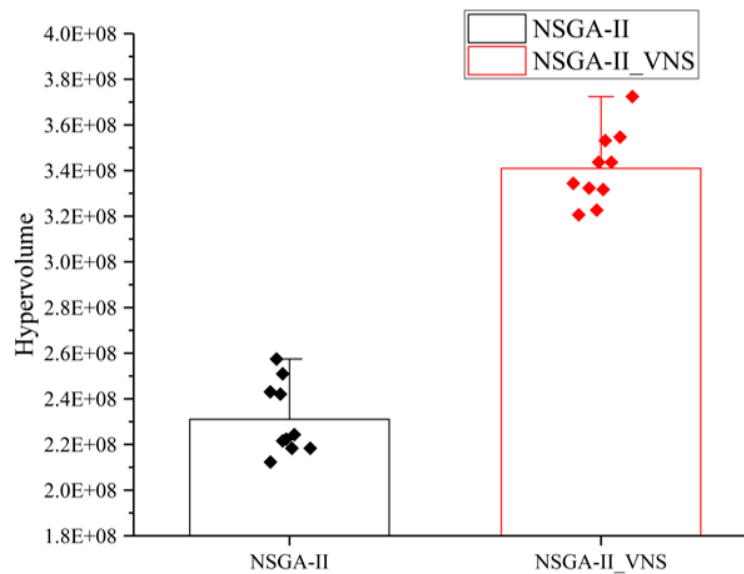
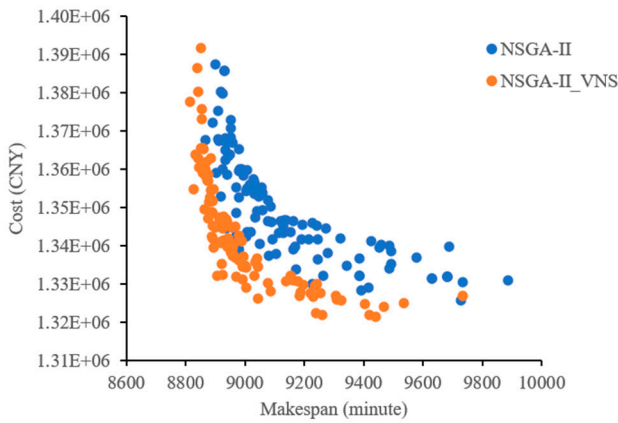


Figure 2. Bar graphs of Hypervolume for NSGA-II and VNS-NSGA-II running ten times. (a–f) correspond to the Job1–Job6 diagram.

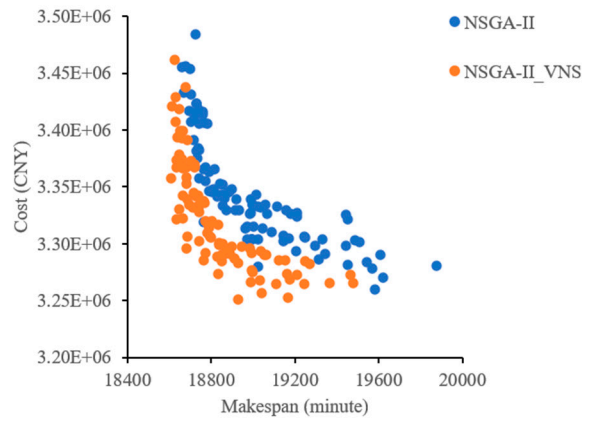


(g) Hypervolume results of Job7 running ten times.

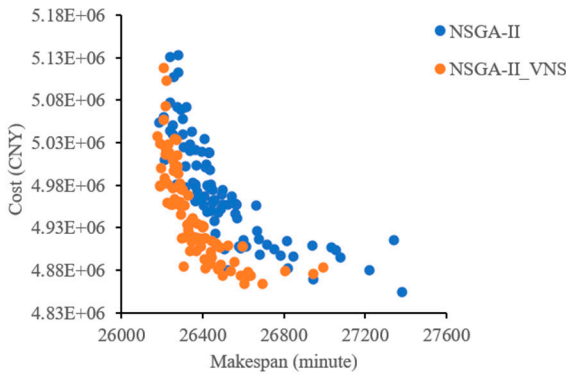
Figure 3. Bar graphs of Hypervolume for NSGA-II and VNS-NSGA-II running ten times. (g) Corresponds to the Job7 diagram. (g) Hypervolume results of Job7 running ten times.



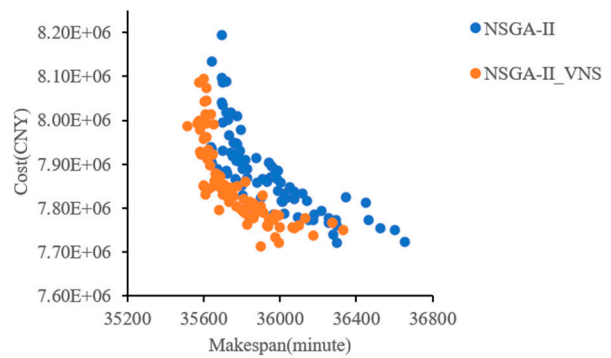
(a) Two-dimensional scatter diagram of Job1 among the ten runs.



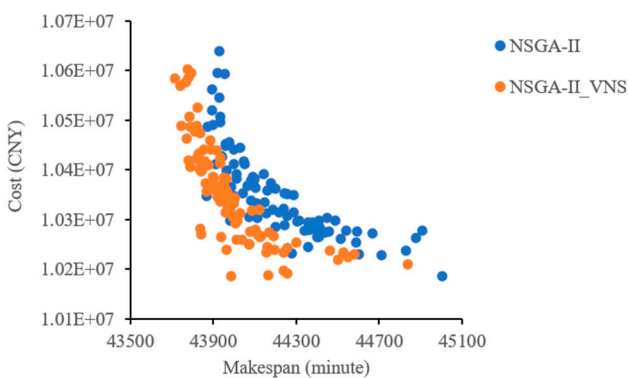
(b) Two-dimensional scatter diagram of Job2 among the ten runs.



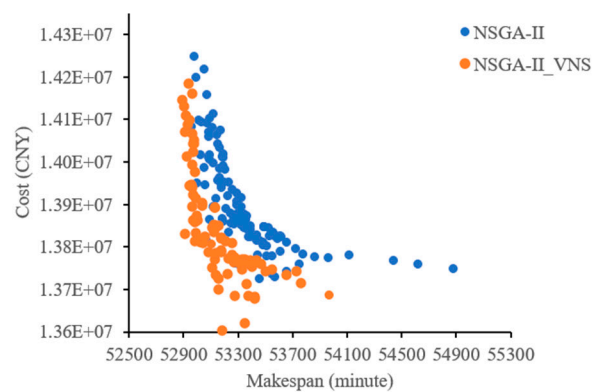
(c) Two-dimensional scatter diagram of Job3 among the ten runs.



(d) Two-dimensional scatter diagram of Job4 among the ten runs.

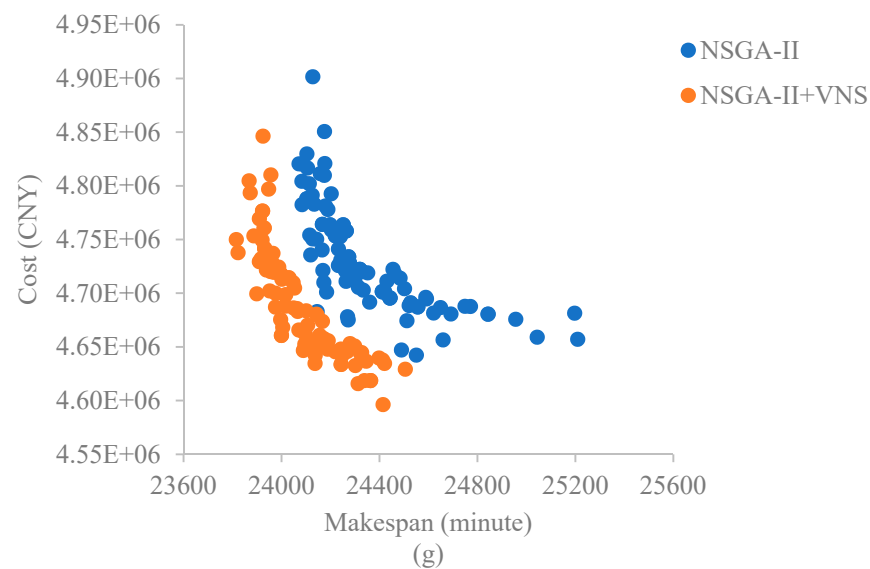


(e) Two-dimensional scatter diagram of Job5 among the ten runs.



(f) Two-dimensional scatter diagram of Job6 among the ten runs.

Figure 4. Two-dimensional scatter diagram of the solution of ten runs of NSGA-II and VNS-NSGA-II. (a–f) correspond to Job1–Job6.



**Figure 5.** Two-dimensional scatter diagram of the solution of ten runs of NSGA-II and VNS-NSGA-II. (g) corresponds to Job7. (g) Two-dimensional scatter diagram of Job7 among the ten runs.

**Table 9.** The percentage of the optimal Pareto solution set obtained using two algorithms.

Case Study	NSGA-II	VNS-NSGA-II
Job1	0	100
Job2	0	100
Job3	6.25	93.75
Job4	0	100
Job5	7.69	92.31
Job6	0	100
Job7	0	100

Table 10 shows the makespan and total cost of the solutions obtained by two algorithms. Table 10 shows that in the seven case studies, the average value of makespan obtained using VNS-NSGA-II is shorter and the average cost is lower. Thus, when solving the established FFSP using VNS-NSGA-II, it can not only reduce makespan but also reduce the total cost. In this study, the MOEA is used to solve the proposed cost-minimization production-scheduling problem. It is worth noting that what is obtained is a Pareto solution set. The solution set includes the solution with the lowest total cost and the solution with the shortest completion time. In practice, decision-makers can choose the best solution according to actual conditions. For example, when the production job is heavy, the solution with the shortest completion time could be selected to improve the utilization efficiency of the workshop and ensure that the job can be completed on schedule. Conversely, when production pressure is low, decision makers could choose the best solution with the lowest total cost and complete more processing jobs at the lowest cost.

**Table 10.** The average makespan and average total cost of the solution are obtained by NSGA-II and VNS-NSGA-II.

Study Case	Makespan (min)			Cost (CNY)		
	NSGA-II	VNS-NSGA-II	Difference Value	NSGA-II	VNS-NSGA-II	Difference Value
Job1	9148	9007	−141	1,349,667	1,343,285	−6382
Job2	18,997	18,841	−156	3,343,452	3,322,802	−20,650
Job3	26,498	26,369	−129	4,974,529	4,947,530	−26,999
Job4	35,940	35,775	−165	7,881,580	7,855,390	−26,190
Job5	44,214	43,992	−222	10,376,398	10,354,528	−21,870
Job6	53,355	53,164	−191	13,899,981	13,839,756	−60,225
Job7	24,350	24,091	−259	4,731,357	4,688,217	−43,140

Table 11 is the comparison result of case Job7 using algorithm and manual scheduling, respectively. The makespan of VNS-NSGA-II is reduced by 6.8%, and the production cost of VNS-NSGA-II is reduced by 4.2% compared with the manual scheduling.

**Table 11.** The solutions based on NSGA-II, VNS-NSGA-II, and manual scheduling.

Study Case	Makespan (min)			Cost (CNY)		
	NSGA-II	VNS-NSGA-II	Manual Scheduling	NSGA-II	VNS-NSGA-II	Manual Scheduling
Job7	24,350	24,091	25,849	4,731,357	4,688,217	4,895,954

## 5. Conclusions

This paper studies FFSP with machining constraints, transportation constraints, set-up constraints, early constraints and delay constraints. This paper proposes a MOP model for production scheduling. The proposed model is used to optimize production cost and makespan.

This paper combines NSGA-II and VNS to fine solutions of the proposed production-scheduling model, and seven groups of experiments are used to compare VNS-NSGA-II with NSGA-II. In the seven groups of experiments, the  $p$ -value is less than 0.005. Moreover, the hypervolume of the solutions using VNS-NSGA-II is larger. Thus, the performance of the proposed VNS-NSGA-II is better than NSGA-II. The results show that the production cost and the makespan of the solution obtained by the proposed VNS-NSGA-II are 0.5% less than those obtained by NSGA-II. Moreover, through a comparison of the production cases of a real-life paper company, the results show that the production cost and the makespan obtained by VNS-NSGA-II is 4.2 and 6.8% higher than that of manual scheduling.

**Supplementary Materials:** The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/pr10102072/s1>, Figure S1. Cross operation diagram; Figure S2. Mutation operation diagram.

**Author Contributions:** Conceptualization, H.Z.; Data curation, H.Z. and M.H.; Formal analysis, H.Z.; Funding acquisition, J.L. and Y.M.; Investigation, Z.H.; Methodology, H.Z.; Project administration, H.Z., J.L., M.H. and Z.H.; Resources, J.L., M.H., Y.M. and Z.H.; Supervision, J.L., M.H., Y.M. and Z.H.; Validation, H.Z.; Visualization, H.Z.; Writing—original draft, H.Z.; Writing—review and editing, J.L. and Y.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Science and Technology Program of Guangzhou, China (grant number No.202201010356), and State Key Laboratory of Pulp and Paper Engineering (grant number No.2022ZD02). and The APC was funded by Jigeng Li.

**Institutional Review Board Statement:** I promise that this study not involve human or animal research.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** Data sharing is not applicable to this article.

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

## References

1. Zhu, Z.; Zhou, X.; Shao, K. A novel approach based on Neo4j for multi-constrained flexible job shop scheduling problem. *Comput. Ind. Eng.* **2019**, *130*, 671–686. [[CrossRef](#)]
2. Geng, Z.; Yuan, J.; Yuan, J. Scheduling with or without precedence relations on a serial-batch machine to minimize makespan and maximum cost. *Appl. Math. Comput.* **2018**, *332*, 1–18. [[CrossRef](#)]
3. Georgios, G.P.; Elekidis, A.P.; Georgiadis, M.C. Optimization-Based Scheduling for the Process Industries: From Theory to Real-Life Industrial Applications. *Processes* **2019**, *7*, 438. [[CrossRef](#)]
4. Li, Z.; Yang, H.; Zhang, S.; Liu, G. Unrelated parallel machine scheduling problem with energy and tardiness cost. *Int. J. Adv. Manuf. Technol.* **2016**, *84*, 213–226. [[CrossRef](#)]
5. Luo, H.; Du, B.; Huang, G.Q.; Chen, H.; Li, X. Hybrid flow shop scheduling considering machine electricity consumption cost. *Int. J. Prod. Econ.* **2013**, *146*, 423–439. [[CrossRef](#)]
6. Vaccari, M.; di Capaci, R.B.; Brunazzi, E.; Tognotti, L.; Pierno, P.; Vagheggi, R.; Pannocchia, G. Implementation of an Industry 4.0 system to optimally manage chemical plant operation. *IFAC-PapersOnLine* **2020**, *53*, 11545–11550. [[CrossRef](#)]
7. Vaccari, M.; Bacci di Capaci, R.; Brunazzi, E.; Tognotti, L.; Pierno, P.; Vagheggi, R.; Pannocchia, G. Optimally managing chemical plant operations: An example oriented by Industry 4.0 paradigms. *Ind. Eng. Chem. Res.* **2021**, *60*, 7853–7867. [[CrossRef](#)]
8. Branke, J.; Nguyen, S.; Pickardt, C.W.; Zhang, M. Automated Design of Production Scheduling Heuristics: A Review. *IEEE Trans. Evol. Comput.* **2016**, *20*, 110–124. [[CrossRef](#)]
9. Luo, J.; Fujimura, S.; El Baz, D.; Plazolles, B. GPU based parallel genetic algorithm for solving an energy-efficient dynamic flexible flow shop scheduling problem. *J. Parallel Distrib. Comput.* **2019**, *133*, 244–257. [[CrossRef](#)]
10. Zeng, Z.; Hong, M.; Man, Y.; Li, J.; Zhang, Y.; Liu, H. Multi-object optimization of flexible flow shop scheduling with the batch process—Consideration total electricity consumption and material wastage. *J. Clean. Prod.* **2018**, *183*, 925–939. [[CrossRef](#)]
11. Meng, L.; Zhang, C.; Shao, X.; Ren, Y. MILP models for energy-aware flexible job shop scheduling problem. *J. Clean. Prod.* **2019**, *210*, 710–723. [[CrossRef](#)]
12. 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](#)]
13. Batista Abikarram, J.; McConky, K.; Proano, R. Energy cost minimization for unrelated parallel machine scheduling under real-time and demand charge pricing. *J. Clean. Prod.* **2019**, *208*, 232–242. [[CrossRef](#)]
14. Tran, T.H.; Ng, K.M. A water-flow algorithm for flexible flow shop scheduling with intermediate buffers. *J. Sched.* **2011**, *14*, 483–500. [[CrossRef](#)]
15. Moslehi, G.; Mahnam, M. A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. *Int. J. Prod. Econ.* **2011**, *129*, 14–22. [[CrossRef](#)]
16. Allahverdi, A.; Aydilek, H.; Aydilek, A. No-wait flow shop scheduling problem with two criteria; total tardiness and makespan. *Eur. J. Oper. Res.* **2018**, *269*, 590–601. [[CrossRef](#)]
17. Lu, C.; Gao, L.; Li, X.; Pan, Q.; Wang, Q. Energy-efficient permutation flow shop scheduling problem using a hybrid multi-objective backtracking search algorithm. *J. Clean. Prod.* **2017**, *144*, 228–238. [[CrossRef](#)]
18. Chandra, P.; Mehta, P.; Tirupati, D. Permutation flow shop scheduling with earliness and tardiness penalties. *Int. J. Prod. Res.* **2009**, *47*, 5591–5610. [[CrossRef](#)]
19. Schaller, J.; Valente, J.M.S. Heuristics for scheduling jobs in a permutation flow shop to minimize total earliness and tardiness with unforced idle time allowed. *Expert. Syst. Appl.* **2019**, *119*, 376–386. [[CrossRef](#)]
20. Mishra, A.; Shrivastava, D. A TLBO and a Jaya heuristics for permutation flow shop scheduling to minimize the sum of inventory holding and batch delay costs. *Comput. Ind. Eng.* **2018**, *124*, 509–522. [[CrossRef](#)]
21. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T.A.M.T. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2001**, *6*, 182–197. [[CrossRef](#)]
22. Hansen, P.M.; Ladenović, N.; Pérez, J. Neighborhood search: Methods and applications. *Ann. Oper. Res.* **2008**, *3*, 593–595. [[CrossRef](#)]
23. Zitzler, E.; Thiele, L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.* **1999**, *3*, 257–271. [[CrossRef](#)]
24. Fan, Z.; Fang, Y.; Li, W.; Cai, X.; Wei, C.; Goodman, E. MOEA/D with angle-based constrained dominance principle for constrained multi-objective optimization problems. *Appl. Soft. Comput. J.* **2019**, *74*, 621–633. [[CrossRef](#)]
25. Rosa, B.F.; Souza, M.J.F.; de Souza, S.R.; Filho, M.F.D.F.; Ales, Z.; Michelon, P.Y.P. Algorithms for job scheduling problems with distinct time windows and general earliness/tardiness penalties. *Comput. Oper. Res.* **2017**, *81*, 203–215. [[CrossRef](#)]