

Article

Effective Improved NSGA-II Algorithm for Multi-Objective Integrated Process Planning and Scheduling

Xiaoyu Wen , Qingbo Song, Yunjie Qian, Dongping Qiao, Haoqi Wang *, Yuyan Zhang and Hao Li

Henan Provincial Key Laboratory of Intelligent Manufacturing of Mechanical Equipment,
Zhengzhou University of Light Industry, Zhengzhou 450002, China

* Correspondence: haoqiwan0218@163.com

Abstract: Integrated process planning and scheduling (IPPS) is important for modern manufacturing companies to achieve manufacturing efficiency and improve resource utilization. Meanwhile, multiple objectives need to be considered in the realistic decision-making process for manufacturing systems. Based on the above realistic manufacturing system requirements, it becomes increasingly important to develop effective methods to deal with multi-objective IPPS problems. Therefore, an improved NSGA-II (INSGA-II) algorithm is proposed in this research, which uses the fast non-dominated ranking method for multiple optimization objectives as an assignment scheme for fitness. A multi-layer integrated coding method is adopted to address the characteristics of the integrated optimization model, which involves many optimization parameters and interactions. Elite and mutation strategies are employed during the evolutionary process to enhance population diversity and the quality of solutions. An external archive is also used to store and update the Pareto solution. The experimental results on the Kim test set demonstrate the effectiveness of the proposed INSGA-II algorithm.

Keywords: integrated process planning and scheduling; multi-objective optimization; mutation strategy; elite strategy

MSC: 90C27



Citation: Wen, X.; Song, Q.; Qian, Y.; Qiao, D.; Wang, H.; Zhang, Y.; Li, H. Effective Improved NSGA-II Algorithm for Multi-Objective Integrated Process Planning and Scheduling. *Mathematics* **2023**, *11*, 3523. <https://doi.org/10.3390/math11163523>

Academic Editor: Andrea Scozzari

Received: 13 July 2023

Revised: 12 August 2023

Accepted: 13 August 2023

Published: 15 August 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 the context of Industry 4.0, smart manufacturing is an effective way to realize sustainable production systems [1]. The application of new technologies, such as digital twins and networked manufacturing, promotes the progress of smart manufacturing and enhances data interconnection for process planning and scheduling. In order to improve the overall performance of manufacturing systems, it is important to study the integrated process planning and scheduling (IPPS) problem. Scheduling is the link between process planning and the implementation of production solutions [2,3]. Process planning is the link between part design and production, and scheduling is the link between process planning and the implementation of production solutions. Process planning establishes the processing route and resource allocation for tasks, as well as providing processing plans for scheduling. Scheduling assigns each processing task to the appropriate machine [3,4].

In the manufacturing process of most products, process planning and scheduling operate independently and are fragmented from each other, which may lead to conflicting optimization objectives and unbalanced system collaboration, and seriously affect the operational efficiency of the manufacturing process [5]. IPPS aims to bridge the gap between process planning and scheduling, unify the optimization objects, and realize system collaboration to make optimization more efficient, stable, and reasonable [6]. In a practical manufacturing system, there are many evaluation criteria, such as maximal

completion time of machines (makespan), maximum machine workload (MMW), total workload of machines (TWM), etc., and it is obvious that considering only a single objective cannot meet the needs of realistic production [7]. Therefore, the study of multi-objective IPPS problems has significant importance in both real-world production environments and in theory.

The IPPS problem is an extension of the flexible job shop scheduling (FJSP) problem, as both the job shop scheduling (JSP) problem and the FJSP problem are NP-hard problems [8]. The IPPS problem is more complex and difficult to solve because it is more flexible, with various process plans for each component [9], and because it considers multiple objectives for simultaneous optimization. Traditional exact methods cannot effectively solve large-scale scheduling problems in an acceptable period of time [10–12]. In order to better meet the needs of real-world problems, metaheuristic algorithms have become an important tool for solving such problems [13]. Given the multi-flexibility and complexity of IPPS problems, it is necessary to devise an efficient algorithm for solving multi-objective IPPS problems. An improved NSGA-II (INSGA-II) algorithm is proposed in this research. A multi-layer integrated coding approach is employed, which facilitates a comprehensive exploration of the solution space. This approach enables the simultaneous optimization of multiple parameters of the IPPS problem within a single iteration, enhancing the efficiency of the algorithm. The algorithm evolution process introduces an elite strategy and a variation strategy to improve the quality of population diversity and solution, and an external archive is used to save and update the Pareto solution. Finally, experimental results on the Kim test set validate the effectiveness of the NSGA-II algorithm.

The remaining sections are organized as follows: Section 2 examines the pertinent literature on IPPS problem-solving algorithms. Section 3 describes the IPPS problem. Section 4 presents the proposed INSGA-II algorithm and strategy. We experiment to determine the efficacy of the INSGA-II algorithm in Section 5 for solving the multi-objective IPPS problem. Section 6 concludes with a summary of the paper's findings and an outlook on future research directions.

2. Literature Review

In order to effectively address the game-theoretic model of the multi-objective IPPS problem, Li et al. [14] proposed a hybrid algorithm (hybrid algorithm of genetic algorithm (GA) and tabu search (TS)) based on minimizing the makespan, minimizing the MMW, and minimizing the TMW as optimization objectives for handling multiple objectives. A priority-based optimization approach was proposed by Ausaf et al. [15] to address the multi-objective IPPS issue. This algorithm makes effective optimization possible when many objectives are taken into account by allocating various objective functions varying degrees of priority. The algorithm also employs a variety of scheduling criteria to aid in selecting the optimal answer. Mahdi et al. [16] introduced optimization algorithms for IPPS and discussed how to increase productivity and decrease production costs by using IPPS. These algorithms included GAs, ant colony algorithms (ACO), and particle swarm optimization (PSO) algorithms, which can optimize production planning by maximizing resource utilization, reducing waiting times, using real-time monitoring and adjustment, and optimizing decision making to improve production. Right now, swarm intelligence optimization algorithms and evolutionary algorithms are the primary methodologies for solving multi-objective IPPS problems.

2.1. Swarm Intelligence Optimization Algorithm

Cao et al. [17] proposed an adaptive multi-strategy artificial bee colony (AMSABC) approach to address the issues with IPPS. In AMSABC, the employed bee and the observing bee are introduced to two search algorithms with distinct characteristics, referred to as "local search" and "global search", respectively. Zhao et al. [18] came up with a two-generation Pareto-based ACO algorithm to solve multi-objective JSPs with optional process paths and parallel machines that do not work together. In order to reduce the overall workload,

workload deviation, and total cost, the parent ACO resolves the subproblem of flexible process route selection and builds a “sub-tour” by choosing nodes from several alternative process plans and machines. The child ACO resolves the subproblem of process job sequencing. To find the Pareto solution set, Kwang et al. [19] presented a method combining the firefly algorithm and a GA. First, the approach makes use of the firefly algorithm’s population self-segmentation and multi-modal search capabilities to simultaneously search for many optimal sites while preserving the diversity of populations in the solution space. In particular, multi-objective optimization issues benefit from this characteristic. Second, by effectively constructing chromosomal encoding techniques, crossovers, and mutation procedures, GAs are completely exploited in addressing discrete combinatorial optimization problems. For challenges involving scheduling and multi-objective integrated process planning, Wang et al. [20] introduced a PSO-based approach. The technique presents a distinctive solution representation based on binary encoding and uses specific operations (e.g., mutation and crossover) to assure the validity and viability of the solution. It also uses exploratory search and quick convergence to efficiently explore enormous solution spaces.

2.2. Evolutionary Algorithms

Mohammadi et al. [21] developed a multi-objective problem-solving strategy based on two meta-heuristics: the simulated annealing (SA) and TS algorithms. To benefit from the exploration search and fast convergence, a multi-objective hybrid SA technique (MOHSA) was suggested. The NSGA-II algorithm was used to handle the computational complexity of the problem when Mohapatra et al. [7] considered IPPS as a multi-objective optimization problem in a reconfigurable manufacturing environment, taking into account manufacturing cycle time, processing cost, and machine utilization criteria. To solve the multi-objective IPPS problem, Jin et al. [22] suggested a multi-objective memetic algorithm (MOMA). The MOMA algorithm uses a problem-specific multi-objective local search approach that combines a variable neighborhood search (VNS) procedure with an effective goal-specific intensive search strategy to optimize all potential scheduling. In order to solve the multi-objective IPPS problem, Shokouhi et al. [23] proposed a GA with the objective functions of the makespan, minimization of the critical machine workload, and minimizing the TMW. In order to save energy and lessen emissions, Zhang et al. [24] suggested a hierarchical multi-strategy GA (NSHMSGGA) based on non-dominated ranking for the IPPS. A population degradation operator based on crowding distance ranking is added to the algorithm, which is built on the NSGA-II framework, to boost the diversity of repeated populations while also strengthening the genetic operator and coding approach. For the purpose of solving the multi-objective IPPS problem, Luo et al. [25] employed an effective multi-objective GA based on the immunity principle and external archives. The method uses an external archive to save non-dominated solutions for selection in following rounds and relies on the immunity principle to maintain population variety and prevent premature maturation. As an additional adaptation and assignment mechanism, the algorithm employs a quick, non-dominated sorting technique. Insights and research on using GAs to solve IPPS difficulties were offered by Chaudhry et al. [26]. To increase the effectiveness and precision of GA solutions, improved operator schemes and genetic representations were created. A control elite non-dominated sorting GA was presented by Mohapatra et al. [27], taking into account three separate goals: reducing production time, reducing processing costs, and reducing machine idle time. In order to generate a set of optimal solutions that are non-dominated under various goal functions, it blends evolutionary algorithms and non-dominated ranking techniques. To keep the present optimal solution and ensure the newly generated solution is not worse than the current optimal solution, the control elite strategy is utilized. The green multi-objective IPPS problem was addressed by Wen et al. [28] using the NSGA-II algorithm. The problem is broken down into two phases: the process planning phase, which is optimized using the fundamental NSGA-II algorithm to produce a close to optimal process plan, and

the scheduling phase, which employs an improved NSGA-II algorithm to discover a non-dominated scheduling plan. For the purpose of enhancing the multi-objective IPPS problem, Li et al. [29] suggested an effective GA. The excellent characteristics of the parent individuals can be better kept by introducing a new crossover operation approach, and a new local search method can be introduced to better explore the best solutions in the solution space.

The primary objectives of the solution are makespan, TWM and MMW, etc. It is clear that the usual solution approach for the multi-objective IPPS problem is an improved version of the algorithm. However, some issues still need to be addressed more effectively. One of the key factors is that these algorithms are either not created in accordance with the problem's features or are not sufficiently consistent with those qualities. In this study, a multi-objective IPPS model is made and an INSGA-II algorithm is proposed. The following three objectives were considered. The first objective is makespan. This objective represents the shortest time required for all tasks to be completed. Minimizing makespan can shorten production cycles and improve production efficiency. The second objective is TWM and represents the sum of the workloads of all machines. Minimizing TWM can reduce machine usage and lower production costs. The third objective is MMW and represents the largest workload among all machines. Minimizing MMW can make machine loads more balanced, avoiding overloading some machines and affecting production efficiency. For the properties of the integrated optimization model, where there are numerous optimization parameters and they interact with one another, a multi-layer integration coding method is utilized. In order to guarantee the quality of population diversity and Pareto solutions, elite and variation strategies are built during the algorithm's progress. The Pareto solutions are also stored and maintained in an external archive.

3. Problem Modeling

In order to better solve the multi-objective IPPS problem, it needs to be modeled. First, the problem is described in detail. Then, assumptions are made about the actual production and processing situation, and a mathematical model of the problem is developed based on the problem description and assumptions in order to solve it using an optimization algorithm.

3.1. Problem Description

IPPS is the organic integration and coordination of all production planning components for effective and adaptable manufacturing production processes. This is how it is explained: A machine set of k machines (M_1, M_2, \dots, M_k) is used to process a job set (J_1, J_2, \dots, J_i) having i jobs. Each job includes numerous processing features, each of which can be treated using a variety of processing techniques. If the limitations are met, the job can also have various processing orders. Different machines can execute each step of the job. Table 1 displays a case of an IPPS issue involving three jobs and five machines. Feature F_1 has two machining methods that correspond to processes O_1-O_2 and O_3-O_4 , and process O_4 can be machined by two machines. J_1 consists of three features to be machined, and feature F_2 must be processed before feature F_3 . Solving the production solution for this job set means determining the machining method for each job, the machining equipment, and the start and finish times of the process on the machine to ensure that the final scheduling solution meets multiple production requirements.

Table 1. The machining information of instance with three jobs and five machines.

Jobs	Features	Alternative Operations	Alternative Machines	Precedence Constraints	Processing Time
J ₁	F ₁	O ₁ -O ₂	M ₂ , M ₃ /M ₂ , M ₅	Before F ₃	4, 6/8, 10
		O ₃ -O ₄	M ₄ /M ₂ , M ₃		6/4, 6
	F ₂	O ₅	M ₃ , M ₄		6, 5
J ₂	F ₃	O ₆ -O ₇	M ₁ , M ₂ /M ₃	Before F ₃	8, 7/9
	F ₁	O ₁	M ₂ , M ₄		10, 9
	F ₂	O ₂	M ₂ , M ₃		8, 9
J ₃	F ₂	O ₃ -O ₄	M ₁ /M ₃ , M ₄	Before F ₂	7/6, 7
		F ₃	O ₅		M ₂ , M ₅
	F ₁	O ₁	M ₂ , M ₄		7, 9
	F ₂	O ₂	M ₃ , M ₅		6, 8
	F ₃	O ₃ -O ₄	M ₂ , M ₄ /M ₃		8, 9/3
	F ₄	O ₅	M ₂	7	

3.2. Mathematical Model

Based on the real manufacturing and processing environment, the following assumptions are established prior to developing the specific mathematical model:

- (1) Each machine is limited to processing one procedure at a time;
- (2) It is known how long each job will take to process;
- (3) Processes for the same job have sequential priorities, whereas they do not have sequential priorities for processes for distinct jobs;
- (4) A process cannot be stopped once it has begun;
- (5) Only one machine may be used to machine a given job at once;
- (6) There are no machine failures, and all machines are functional at time 0.

The definitions of several of the symbols used in this paper are provided in Abbreviations.

The mathematical model of multi-objective IPPS addressed in this paper is stated as follows:

$$\text{Min } C_{max} = \text{Max } C_i \forall i \in n \tag{1}$$

$$\text{Min } TWM = \left(\sum_{i \in n} \sum_{j \in h_i} \sum_{k \in m} t_{i,j,k} x_{i,j,k} \right) \tag{2}$$

$$\text{Min } MMW = \left(\text{Max} \left(\sum_{i \in n} \sum_{j \in h_i} t_{i,j,k} x_{i,j,k} \right) \right) \forall k \in m \tag{3}$$

$$S_{i,j} + t_{i,j,k} = f_{i,j} \forall i \in n, j \in h_i, k \in m \tag{4}$$

$$f_{i,j} \leq S_{i,j+1} \forall i \in n, j \in h_i \tag{5}$$

$$S_{i,j} + t_{i,j,k} \leq S_{h,p} + M \cdot (1 - y_{i,j,h,p}) \forall i, h \in n, j, p \in h_i, k \in m \tag{6}$$

$$\sum_{k \in m} x_{i,j,k} = 1 \forall i \in n, j \in h_i \tag{7}$$

$$S_{i,j} \geq 0 \forall i \in n, j \in h_i \tag{8}$$

$$f_{i,j} \geq 0 \forall i \in n, j \in h_i \tag{9}$$

Three objective functions are represented by Equations (1)–(3). Equations (4) and (5) place restrictions on the order in which processes must be carried out in accordance with the process route for each job. Equation (6) states that each machine can only process one

process at a time. Equation (7) consists of exclusivity constraints that ensure each process can only be carried out on one machine at a given time. Equations (8) and (9) state that the variables of each parameter must be positive.

4. Improved NSGA-II Algorithm for Solving Multi-Objective IPPS Method

As the multi-objective problem of IPPS requires simultaneous optimization of job machining methods, process routes, job sequences, and process machine assignments, it involves many optimization parameters. In addition, the optimization parameters interact with each other, e.g., when one of the machining methods is changed, the type of the candidate machine tool will also be changed, and the process machine assignments are changed as well. Therefore, a multilayer integrated coding method is adopted, based on which the NSGA-II algorithm is improved to realize the model solution. In the evolution, the elite strategy and the variation strategy are constructed. The quality of population diversity and solution is improved, and the localization of optimization results is improved.

4.1. INSGA-II for Scheduling

In this paper, a multilayer integrated coding method that simultaneously optimizes multiple parameters of the IPPS problem in a single iteration is used and an improved NSGA-II algorithm is designed to solve it. Figure 1 depicts the algorithm’s flowchart.

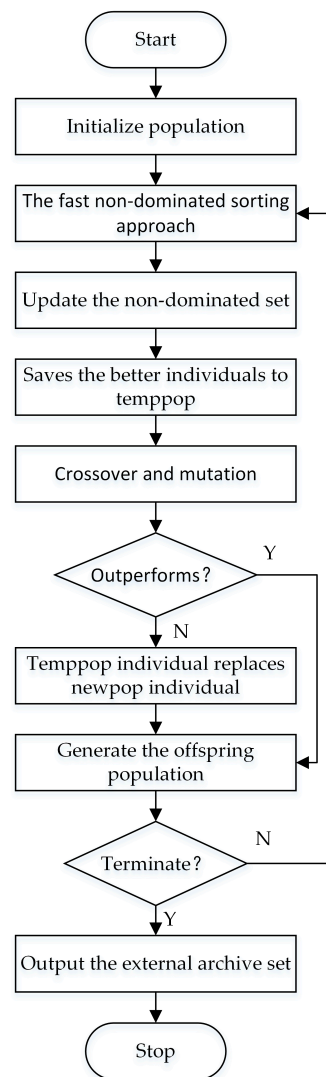


Figure 1. Workflow of the INSGA-II.

The main steps of INSGA-II for scheduling are described as follows:

- Step 1: Initialization parameters, such as the population size, the number of iterations allowed, the likelihood of variation and crossover, and the size of external archives;
- Step 2: Using the non-dominated sorting method [25] and the crowding calculation to identify individuals who are not dominated;
- Step 3: Modifying the external archive established under the external archiving plan;
- Step 4: Crossover and mutation. The elite strategy saves the better individuals to the elite library “temppop”, and, after performing crossover and mutation, if the children are better than the parents, it replaces it, otherwise, it remains unchanged and the remaining individuals are directly replaced with children;
- Step 5: Combining the children and parent populations, performing non-dominated sorting and calculating crowding, and selecting the optimal solution as the new population;
- Step 6: Check to see if the algorithm’s termination condition is met. Execute Step 7 if it is satisfied; otherwise, continue with Step 2;
- Step 7: Output the collection of Pareto-optimal solutions.

4.2. Encoding and Decoding Method

The integrated optimization model requires optimization of a large number of parameters, including the machining features of all jobs, the process route, the sequence of job machining, and the machine assignment of each process. The general coding method can only employ cyclic alternating optimization, first optimizing the machining method and process route, then the job machining sequence and the machine assignment of each process, and, finally, returning to optimizing the machining method and process route once more.

In this paper, a multi-layer integrated coding approach is used [30] and the first layer is the processing feature coding, which indicates the processing method corresponding to all the processing features of the job. The total length of the feature code is equal to the sum of all the features of the job, and the gene is a processing feature of a job. The first layer in Figure 2 consists of 10 gene bits to form the processing method code layer, which indicates the processing order of each of the three jobs; for example, the first three gene bits indicate that the processing order of the features of J_1 is $F_1-F_2-F_3$.

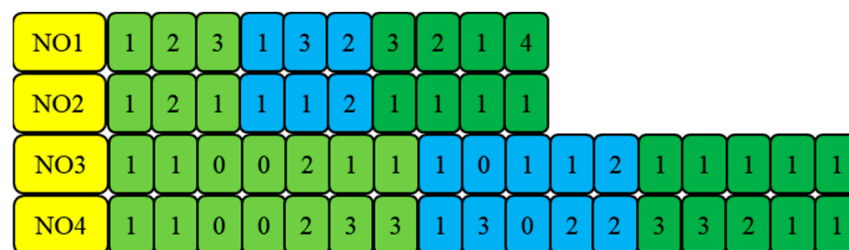


Figure 2. Multi-layer integrated encoding.

The second layer is the process selection coding, which represents the process selection scheme for all jobs. The total length of the process route coding is equal to the feature coding. The gene positions represent the processing sequence corresponding to each machining feature of each job in turn. Combined with the machining information in Table 1, for example, the digit 1 at the second gene position indicates that the machining feature F_2 of J_1 selects the first processing method, that is, operation O_5 .

The third layer is the machine selection coding, indicating that all jobs are machine selected according to the second layer of process selection after the length is equal to the sum of the total number of processes of all jobs, while each part is the total number of processes of its individual jobs. Combined with the processing information in Table 1, the fifth genetic digit 2 indicates that the O_5 processing machine is M_4 . The number 0 in the code indicates that the process was not selected when the job was processed.

The fourth layer is the processing sequence coding, which shows the order in which all job operations should be completed, and its total length is the same as the machine selection coding. The occurrence of a certain job number in the gene position indicates the processing sequence of that job, and the meaning of the number 0 in the coding is the same as that of the machine selection coding.

The process path and scheduling plan for all jobs are simultaneously optimized using this technology, which integrates four sets of parameters. By using this encoding technique, the search space can be shrunk, search effectiveness and optimization precision increased, and computational complexity decreased.

The machining information from Table 1 shows that there are three jobs and five machines. J_1 and J_2 have three machining features, while J_3 has 4 machining features. Since the coding schemes are related to one another, obtaining the machining information for the jobs requires decoding the coding. The following describes how to decode the encoding technique in Figure 2.

The gene bits are obtained from left to right in the order of processing characteristics for each task, starting with the first layer of decoding features:

$$\begin{aligned} J_1: & F_1-F_2-F_3; \\ J_2: & F_1-F_3-F_2; \\ J_3: & F_3-F_2-F_1-F_4; \end{aligned}$$

Following is the genetic position from left to right to determine the machining process chosen for the machining features of each job in the second layer of process selection decoding:

$$\begin{aligned} J_1: & O_1-O_2-O_5-O_6-O_7; \\ J_2: & O_1-O_5-O_3-O_4; \\ J_3: & O_3-O_4-O_2-O_1-O_5; \end{aligned}$$

To obtain the processing machine for each task in the sequence of processing processes, the third layer of machine selection decoding uses genetic position from left to right, as follows:

$$\begin{aligned} J_1: & O_1 (M_2)-O_2 (M_2)-O_5 (M_4)-O_6 (M_1)-O_7 (M_3); \\ J_2: & O_1 (M_2)-O_5 (M_5)-O_3 (M_3)-O_4 (M_4); \\ J_3: & O_3 (M_2)-O_4 (M_3)-O_2 (M_3)-O_1 (M_2)-O_5 (M_2); \end{aligned}$$

The fourth layer is in charge of sequential decoding. Using the decoding information from the first three layers, the active scheduling decoding method is used to create the final decoded Gantt chart shown in Figure 3.

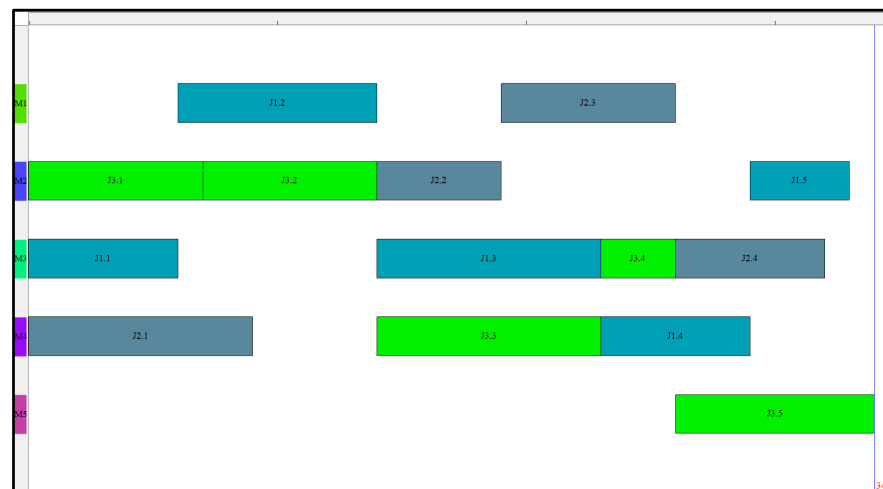


Figure 3. Example of INSGA-II decoding for IPPS.

4.3. Initial Population Acquisition and Crossover and Mutation Operations

According to the encoding approach, the initial population is made at random, and each layer of processing characteristics, process selection, machine selection, and processing order is made after the model constraints and encoding rules are met.

The precedence operation crossover (POX) crossover operator is made to adjust to the situation for the four-layer coding because crossover and variation are needed for each coding layer independently. This larger-scale genetic information exchange, as seen in Figure 4, is the first crossover operation with a strong global search capability. This can increase the population’s diversity and broaden the algorithm’s search scope. This is achieved by first randomly dividing the job numbers {1, 2, 3} into two non-empty job subsets {3} and {1, 2}. Then, the feature layer, process layer, and machine layer of the jobs belonging to the job set {3} in the parent generations P1 and P2 are copied to the corresponding positions of the children generations C1 and C2, respectively. Finally, the feature layer, process layer, and machine layer of the artifacts belonging to the job sets {1, 2} are copied to the corresponding positions of the children C2 and C1, respectively. The second crossover operation complements the first by performing a more thorough search of the solution space, and the two approaches work best together. When the population meets the requirements of the process selection coding layer with at least one artifact that shares the same coding information as its counterparts in the other population, this operation is initiated, and only two chromosomes, which correspond to the processing feature layer and the machine selection layer, need to be exchanged. Four variation operations are created to disturb each of the four layers in order to boost population variety and enhance the search dimension of the algorithm. Mutations and more detailed crossover manipulations are described in the literature [30]. In the algorithm iteration, these six evolutionary operations are carried out concurrently to search the solution space from various breadths and depths, enhancing the algorithm’s search capability.

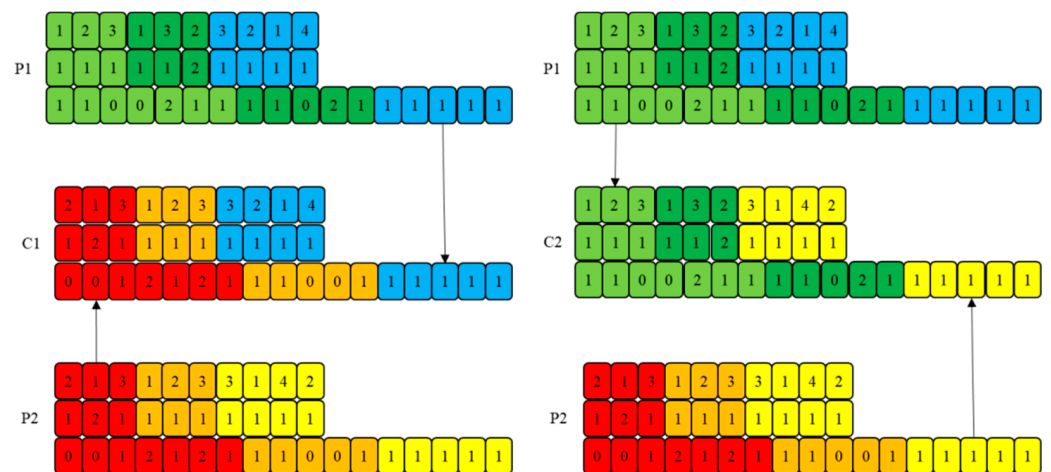


Figure 4. Crossover operation.

4.4. Mutation Strategy and Elite Strategy Solutions

Large mutation probabilities in evolution can introduce more diversity in the search space and increase the global exploration ability of the algorithm. When the algorithm is stuck in a local optimal solution, large mutation probabilities can quickly jump out of the local optimal solution and re-explore other possible solutions. However, large mutation probabilities may also lead to slower convergence of the algorithm because they increase the randomness of the algorithm. The role of the elite strategy is to retain the better individuals in the evolutionary process and avoid the loss of good information, thus speeding up the convergence of the algorithm. However, the implementation of the elite strategy may lead to a reduction in the diversity of the population, which affects the exploration ability of the

algorithm. To balance the relationship between elite individuals and diversity, it is often necessary to use it in combination with other optimization strategies.

The elite strategy and variation strategy are combined in this study using the following methodology: since the solutions consist of multiple targets, dominance rank is used as the fitness allocation scheme. Before each evolution, the population’s good solutions that satisfy the evolutionary conditions are stored separately in the elite pool. By determining the size of the dominance rank, it is decided which is better, the parent or the children; if the evolved children outperform their parent, the parent is removed, and the offspring is then stored in the new population. If not, the parent is retained in the new population and the offspring is discarded.

4.5. External Archive Maintenance Scheme

The result of a multi-objective optimization problem is the set of Pareto optimal solutions rather than a single solution. The solutions obtained throughout the optimization process, which are not dominated by one another, are stored and maintained in the external archive. The external archive is updated in accordance with the following steps whenever a new solution is discovered during an IPPS iteration: (1) if any solution in the external archive dominates the new solution, the new solution is discarded; (2) if any solution in the external archive dominates the new solution, it is removed from the archive and a new solution is added; (3) if all solutions in the external archive are non-dominated by the new solution and the archive is not full, the new solution is added to the archive; if the archive is full, the solution with the smallest crowding distance is removed [25].

4.6. Time Complexity of INSGA-II Algorithm

The time complexity of an algorithm is the number of basic operations required to complete the entire algorithm, given the overall scale, number of iterations, and problem size of the algorithm. In the algorithm, the number of times a statement is executed is called frequency, denoted as $T(n)$. $T(n)$ is a function of the overall scale, number of iterations, and problem size. For ease of calculation, we assume that the population size of the INSGA-II algorithm is N , the number of iterations of the population is G , and the variables F_n, P_n, O_n , and n_{ml} represent the number of features, process numbers, process numbers, and machine string lengths of the job; J_n represents the number of jobs and M_n represents the number of machines. The following is a frequency analysis of the main steps.

(1) Population initialization stage:

$$T_1 = N \times (J_n \times F_n^2 + J_n \times O_n + J_n \times F_n + J_n \times P_n + J_n \times O_n + J_n \times n_{ml}) = N \times J_n \times (F_n^2 + 2 \times O_n + F_n + n_{ml} + 16 + 10 \times M_n).$$

(2) Update external archive stage: $T_2 = 3 \times N$.

(3) Fast non-dominated sorting stage: $T_3 = 8 \times N + 8 \times N^2$.

(4) Genetic operation stage:

$$T_4 = 5 \times N^2 + 5 \times N \times J_n + N + 6 \times N \times J_n + J_n \times F_n + N \times J_n + N \times n_{ml} + N \times J_n^2 + N \times J_n + N \times P_n + N \times J_n \times F_n + N \times J_n \times O_n = 5 \times N^2 + N \times J_n^2 + J_n \times F_n + N \times (1 + 13 \times J_n + P_n + n_{ml} + J_n \times F_n + J_n \times O_n).$$

(5) Generate new population stage: $T_5 = 10 \times N$.

Therefore, the total frequency of INSGA-II algorithm is:

$$T_6 = T_1 + G \times (T_2 + T_3 + T_4 + T_5) = N \times J_n \times (F_n^2 + 2 \times O_n + F_n + n_{ml} + 16 + 10 \times M_n) + G \times (3 \times N + 8 \times N + 8 \times N^2 + 5 \times N^2 + N \times J_n^2 + J_n \times F_n + N \times (1 + 13 \times J_n + P_n + n_{ml} + J_n \times F_n + J_n \times O_n) + 10 \times N).$$

Constants in computation indicate the number of times an algorithm performs a certain operation. Their effect is usually not taken into account when calculating time complexity. Let the function $f = G \times N \times J_n^2$ be a function. When the problem size is infinitely large, T_6/f is a constant. Therefore, the time complexity of INSGA-II algorithm is $O(G \times N \times J_n^2)$.

5. Experimental Studies and Discussion

The present research tests the algorithm’s effectiveness using the IPPS problem test set recommended by Kim et al. Information on the specific cases is available in the literature [31]. For comparison, the problem was solved using many rounds of the NSGA-II and INSGA-II algorithms. The NSGA-II algorithm, in contrast to the INSGA-II algorithm, only uses multi-layer integrated coding methods. All of the algorithm programs in this paper were created in Visual C++ and ran on a machine with an Intel i7 12700F 2.10 GHz CPU and 16 GB of RAM. The configuration of parameters affects the performance of INSGA-II. The possible range of parameter values was determined based on literature and experience. Table 2 lists all possible parameter combinations. From the table, it can be seen that population sizes are 200, 400, 600, and 800, respectively, while crossover probability (P_C) and machine layer mutation probability (P_M) are 0.6, 0.8, and 0.9, respectively. Feature layer mutation probability (P_F), operation layer mutation probability (P_O), and scheduling layer mutation probability (P_S) are 0.1, 0.2, and 0.3, respectively. After verification, when the parameter combination is as shown in Table 3, the performance of INSGA-II is better.

Table 2. Parameters setting values.

No.	Parameters					
	Population Size	P_C	P_F	P_O	P_M	P_S
1	200	0.6	0.1	0.1	0.6	0.1
2	200	0.8	0.2	0.2	0.8	0.2
3	200	0.9	0.3	0.3	0.9	0.3
4	400	0.6	0.1	0.1	0.6	0.1
5	400	0.8	0.2	0.2	0.8	0.2
6	400	0.9	0.3	0.3	0.9	0.3
7	600	0.6	0.1	0.1	0.6	0.1
8	600	0.8	0.2	0.2	0.8	0.2
9	600	0.9	0.3	0.3	0.9	0.3
10	800	0.6	0.1	0.1	0.6	0.1
11	800	0.8	0.2	0.2	0.8	0.2
12	800	0.9	0.3	0.3	0.9	0.3

Table 3. The parameters in the proposed algorithm.

Parametric Description	Value
Maximum generations of iterations	400
Population size	400
Reproduction probability	0.2
P_C	0.8
P_F	0.2
P_O	0.2
P_M	0.8
P_S	0.2

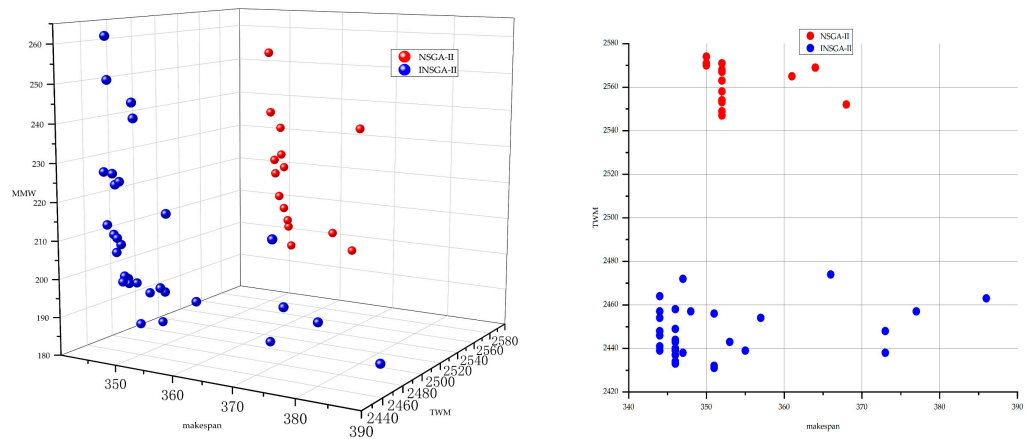
Table 4 shows the total number of non-dominated solutions each algorithm produced for each set of issues, and Table 5 compares the experimental outcomes of the INSGA-II algorithm to those of the other methods. Figure 5 compares the outcomes of the INSGA-II and NSGA-II algorithms. Figure 6 displays the Gantt chart for the instance 20.

Table 4. The number of Pareto solutions obtained by NSGA-II and INSGA-II.

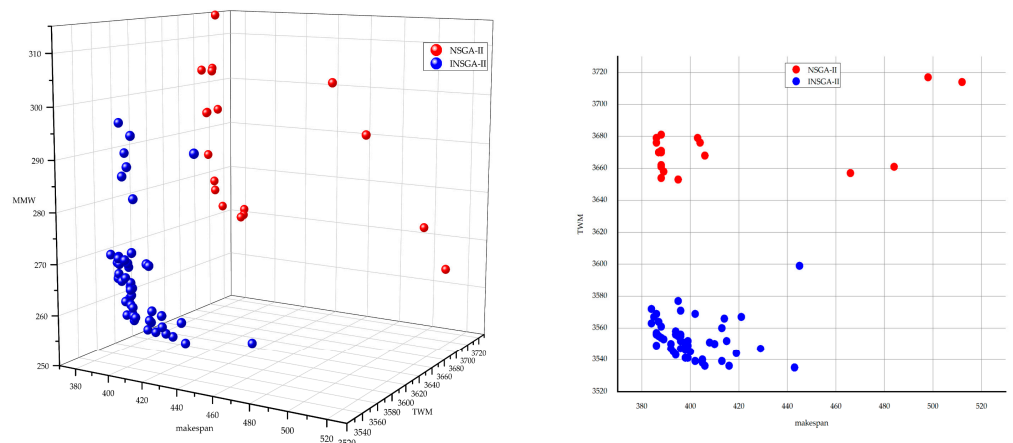
No.	Total Number of Pareto Solutions	NSGA-II	INSGA-II
1	13	0	13
2	24	0	24
3	22	0	22
4	12	0	12
5	30	0	30
6	10	0	10
7	26	0	26
8	20	0	20
9	15	0	15
10	18	0	18
11	30	0	30
12	30	0	30
13	17	0	17
14	21	0	21
15	15	0	15
16	14	0	14
17	50	0	50
18	45	0	45
19	33	0	33
20	50	0	50
21	24	0	24
22	35	0	35
23	50	0	50
24	50	0	50

Table 5. Objective values of resultant schedules.

No.	MOMA			INSGA-II		
	Makespan	TWM	MMW	Makespan	TWM	MMW
1	427	1822	150	427	1822	150
2	343	1623	174	343	1647	167
3	347	1713	166	344 *	1713 *	164 *
4	306	1433	148	306	1438	136
5	319	1588	159	318	1645	129
6	427	2134	175	427 *	2131 *	175 *
7	372	1826	189	372	1861	147
8	343	1686	148	343 *	1603*	141 *
9	427	1641	169	427	1668	153
10	428	2727	237	427	2764	226
11	348	2449	205	344 *	2448 *	205 *
12	320	2231	175	318	2275	175
13	427	2936	245	427	2955	228
14	375	2749	210	372 *	2744 *	209 *
15	427	2430	215	427	2456	196
16	427	3451	251	427	3502	248
17	359	3358	254	358	3408	250
18	329	3043	229	327	3095	227
19	440	3733	270	439	3802	268
20	400	3558	262	394 *	3558 *	259 *
21	427	3336	268	427	3414	267
22	448	4358	317	441	4410	309
23	418	4238	294	390	4278	317
24	482	5195	362	459	5237	368



(a) instance6



(b) instance20

Figure 5. The solutions obtained by NSGA-II and INSGA-II.

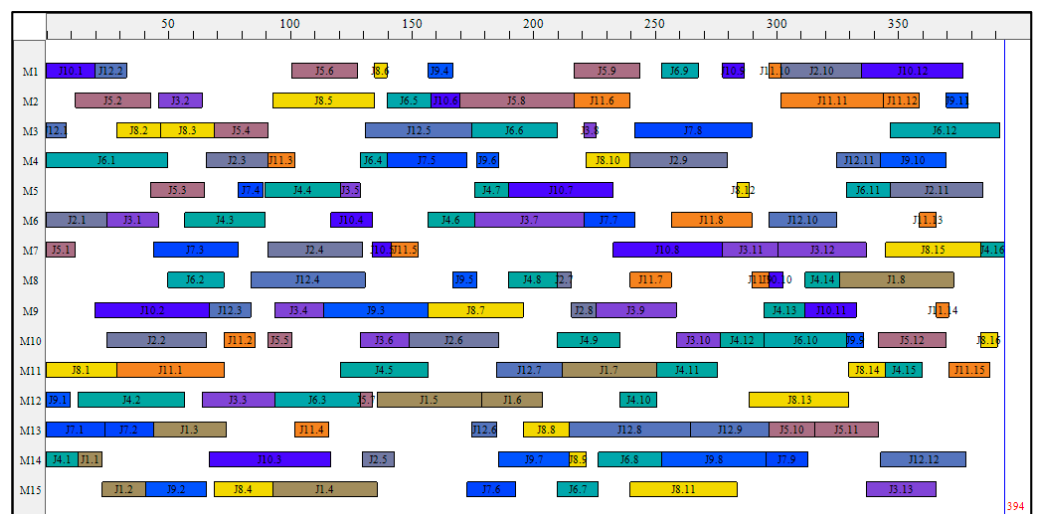


Figure 6. The Gantt chart of the instance 20.

In order to compare the optimization of the two algorithms in 24 instances, all the Pareto solution sets acquired by the two methods are combined, and Table 4 shows the number of non-dominated solutions in the combined solution set. To clearly show the value of the suggested method, the results of the two methods in instances 6 and 20 are displayed in Figure 5. It can be seen that the outcomes produced by INSGA-II are better than those produced by NSGA-II.

Table 5 lists the comparison of the results obtained by the INSGA-II algorithm and MOMA algorithm in literature [22]. The same method as in the literature was used to find the most satisfactory scheduling scheme from the Pareto front obtained by the algorithm for comparison. Where for the same instance if one is dominated by another, it is identified by an asterisk (*) on the non-dominated solution. It is not straightforward to discover that six solutions generated by the INSGA-II algorithm outperform solutions discovered in the literature in all metrics, and three of the examples (instances 3, 11, and 14) find the lower bound on the completion time. In five of the remaining 18 cases (instances 5, 7, 9, 13, and 15), the machine maximum workload metric is obviously superior in the non-dominated solution obtained by the INSGA-II method, and the instances 5, 10, and 12 reach the lower limit of the completion time. The suggested INSGA-II algorithm is superior in the remaining 18 instances, when two solutions are not dominated by one another.

Finally, an illustration is provided to show how it is critical to take several needs into account in a practical production setting. Figure 7 shows the Gantt chart for instance 5, which solely takes into account the makespan. The Gantt chart produced by the INSGA-II algorithm is preferable, even though the two schedules have the same total completion time (318), when the two metrics, maximum machine workload and total machine workload, are compared. Figure 8 displays the workloads on each machine in both Gantt charts. It is obvious that each machine has a more uniform workload in the scheduling scheme produced by the INSGA-II algorithm when several goals are taken into account. The maximum and minimum machine workloads for the scheduling scheme considering three targets are 129 and 61, respectively, in comparison to the scheduling scheme considering only one target. This shows that the latter workload is more evenly distributed across machines, which is significant for extending the machine’s life. The suggested technique also lowers the maximum machine workload by moving unnecessary operations to more practical computers. The scheduling schemes with single objective values of 1687 and multiple objective values of 1645 for the total machine workload index, respectively, show that the proposed algorithm also optimizes the total machine workload.

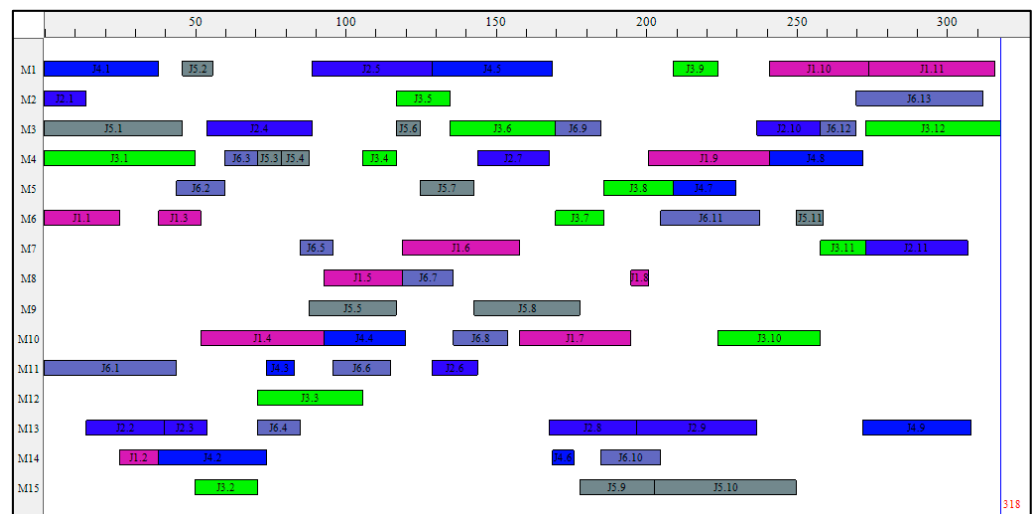


Figure 7. The Gantt chart of the instance 5 with makespan criterion only.

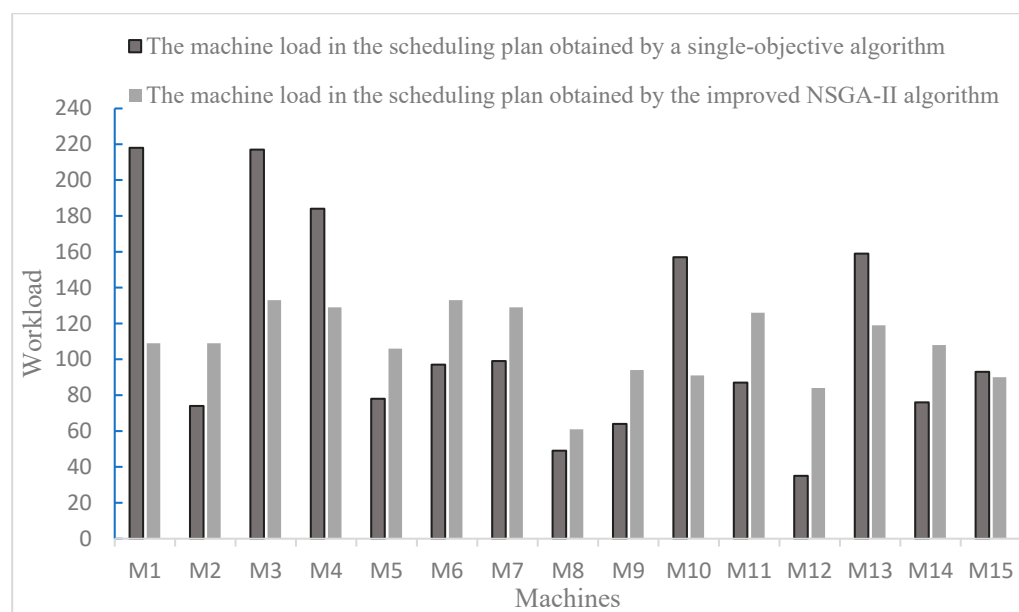


Figure 8. Workload of each machine.

6. Conclusions

The makespan, the MMW, and the TMW are all minimized in this paper's multi-objective IPPS problem model. To increase the depth of search for the features of the integrated optimization model, which has numerous optimization parameters and how they interact with one another, multi-layer integrated coding is utilized. An INSGA-II method using the elite strategy and variation strategy is proposed to overcome the issues of limited population diversity and low quality of the traditional NSGA-II solution set. The Pareto solutions are also maintained and stored in an external archive. Finally, the algorithm's performance was evaluated using the Kim test set, and the findings demonstrated the algorithm's efficiency in resolving problems relating to multi-objective IPPS.

In future research, considering the actual situation in production, several directions can be considered to further improve the proposed method. Firstly, uncertain processing times should be included, and a multi-objective algorithm for the IPPS problem with uncertain processing time can be considered. Secondly, more objectives, such as machining cost and energy consumption, can be taken into account. Thirdly, more effective diversity maintenance schemes can be explored. Finally, exploring more effective algorithms to solve multi-objective IPPS problem is also a promising direction.

Author Contributions: Conceptualization, X.W. and Q.S.; methodology, Y.Z. and D.Q.; investigation, H.L.; supervision, H.W.; software, Y.Q. and Q.S.; writing—original draft preparation, X.W. and Q.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research work is supported by the National Natural Science Foundation of China (grant nos. 51905494 and 52105536), Key Scientific and Technological Research Projects in Henan Province (grant no. 232102221009).

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the anonymous reviewers and the editor for their positive comments.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study, in the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

Explanation of symbols in mathematical model.

Symbol	Description
n	The set of jobs.
m	The set of machines.
k	The number of machines.
h_i	The set of operations in job i .
i, h	The number of jobs.
j, p	The number of operations.
$O_{i,j}$	The j -th operation of the i -th job.
$t_{i,j,k}$	The processing time of operation $O_{i,j}$ on machine k .
$S_{i,j}$	The starting time of $O_{i,j}$.
$f_{i,j}$	The ending time of $O_{i,j}$.
$y_{i,j,h,p}$	If process $O_{i,j}$ is processed before $O_{h,p}$ is 1, otherwise is 0, $\forall i, h \in n, j, p \in h_i$.
$x_{i,j,k}$	If $O_{i,j}$ processed on machine k is 1, otherwise is 0, $\forall i \in n, j \in h_i, k \in m$.
M	A very large positive integer.
C_i	The completion time of job i .
C_{max}	The maximal completion time of machines.
TWM	The total workload of machines.
MMW	The maximum machine workload.

References

1. Tripathi, V.; Chattopadhyaya, S.; Mukhopadhyay, A.K.; Sharma, S.; Li, C.; Di Bona, G. A Sustainable Methodology Using Lean and Smart Manufacturing for the Cleaner Production of Shop Floor Management in Industry 4.0. *Mathematics* **2022**, *10*, 347. [\[CrossRef\]](#)
2. Fan, J.; Shen, W.; Gao, L.; Zhang, C.; Zhang, Z. A Hybrid Jaya Algorithm for Solving Flexible Job Shop Scheduling Problem Considering Multiple Critical Paths. *J. Manuf. Syst.* **2021**, *60*, 298–311. [\[CrossRef\]](#)
3. Han, J.; Han, I.; Lee, E.; Yi, J. Manufacturing Feature Recognition toward Integration with Process Planning. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2001**, *31*, 373–380. [\[CrossRef\]](#)
4. Zhang, B.; Pan, Q.; Gao, L.; Li, X.; Meng, L.; Peng, K. A Multiobjective Evolutionary Algorithm Based on Decomposition for Hybrid Flowshop Green Scheduling Problem. *Comput. Ind. Eng.* **2019**, *136*, 325–344. [\[CrossRef\]](#)
5. Fang, Y.; Peng, C.; Lou, P.; Zhou, Z.; Hu, J.; Yan, J. Digital-Twin-Based Job Shop Scheduling Toward Smart Manufacturing. *IEEE Trans. Ind. Inform.* **2019**, *15*, 6425–6435. [\[CrossRef\]](#)
6. Laili, Y.; Lin, S.; Tang, D. Multi-Phase Integrated Scheduling of Hybrid Tasks in Cloud Manufacturing Environment. *Robot. Comput.-Integr. Manuf.* **2020**, *61*, 101850. [\[CrossRef\]](#)
7. Mohapatra, P.; Benyoucef, L.; Tiwari, M.K. Integration of Process Planning and Scheduling through Adaptive Setup Planning: A Multi-Objective Approach. *Int. J. Prod. Res.* **2013**, *51*, 7190–7208. [\[CrossRef\]](#)
8. Zhang, G.; Hu, Y.; Sun, J.; Zhang, W. An Improved Genetic Algorithm for the Flexible Job Shop Scheduling Problem with Multiple Time Constraints. *Swarm Evol. Comput.* **2020**, *54*, 100664. [\[CrossRef\]](#)
9. Wang, L.; Shen, W.; Hao, Q. An Overview of Distributed Process Planning and Its Integration with Scheduling. *Int. J. Comput. Appl. Technol.* **2006**, *26*, 3. [\[CrossRef\]](#)
10. Zhou, Y.-Z.; Yi, W.-C.; Gao, L.; Li, X.-Y. Adaptive Differential Evolution With Sorting Crossover Rate for Continuous Optimization Problems. *IEEE Trans. Cybern.* **2017**, *47*, 2742–2753. [\[CrossRef\]](#)
11. Lu, C.; Huang, Y.; Meng, L.; Gao, L.; Zhang, B.; Zhou, J. A Pareto-Based Collaborative Multi-Objective Optimization Algorithm for Energy-Efficient Scheduling of Distributed Permutation Flow-Shop with Limited Buffers. *Robot. Comput.-Integr. Manuf.* **2022**, *74*, 102277. [\[CrossRef\]](#)
12. Wang, L.-C.; Chen, C.-C.; Liu, J.-L.; Chu, P.-C. Framework and Deployment of a Cloud-Based Advanced Planning and Scheduling System. *Robot. Comput.-Integr. Manuf.* **2021**, *70*, 102088. [\[CrossRef\]](#)
13. Delgoshaei, A.; Ariffin, M.K.A.B.M.; Leman, Z.B. An Effective 4-Phased Framework for Scheduling Job-Shop Manufacturing Systems Using Weighted NSGA-II. *Mathematics* **2022**, *10*, 4607. [\[CrossRef\]](#)
14. Li, X.; Gao, L.; Li, W. Application of Game Theory Based Hybrid Algorithm for Multi-Objective Integrated Process Planning and Scheduling. *Expert Syst. Appl.* **2012**, *39*, 288–297. [\[CrossRef\]](#)
15. Ausaf, M.F.; Gao, L.; Li, X. Optimization of Multi-Objective Integrated Process Planning and Scheduling Problem Using a Priority Based Optimization Algorithm. *Front. Mech. Eng.* **2015**, *10*, 392–404. [\[CrossRef\]](#)
16. Mahdi, M.A.; Dawood, L.M. Optimization Algorithms for Integrated Process Planning and Scheduling: A Review. *Int. J. Mech. Eng.* **2022**, *7*, 1060–1066.
17. Cao, Y.; Shi, H. An Adaptive Multi-Strategy Artificial Bee Colony Algorithm for Integrated Process Planning and Scheduling. *IEEE Access* **2021**, *9*, 65622–65637. [\[CrossRef\]](#)

18. Zhao, B.; Gao, J.; Chen, K.; Guo, K. Two-Generation Pareto Ant Colony Algorithm for Multi-Objective Job Shop Scheduling Problem with Alternative Process Plans and Unrelated Parallel Machines. *J. Intell. Manuf.* **2018**, *29*, 93–108. [[CrossRef](#)]
19. Ri, K.-W.; Mun, K.-H. Firefly Algorithm Hybridized with Genetic Algorithm for Multi-Objective Integrated Process Planning and Scheduling. *Res. Sq.* **2023**. [[CrossRef](#)]
20. Wang, Y.F.; Zhang, Y.F.; Fuh, J.Y.H. A PSO-Based Multi-Objective Optimization Approach to the Integration of Process Planning and Scheduling. In Proceedings of the IEEE ICCA 2010, Xiamen, China, 9–11 June 2010; pp. 614–619.
21. Mohammadi, G.; Karampourhaghghi, A.; Samaei, F. A Multi-Objective Optimisation Model to Integrating Flexible Process Planning and Scheduling Based on Hybrid Multi-Objective Simulated Annealing. *Int. J. Prod. Res.* **2012**, *50*, 5063–5076. [[CrossRef](#)]
22. Jin, L.; Zhang, C.; Shao, X.; Yang, X.; Tian, G. A Multi-Objective Memetic Algorithm for Integrated Process Planning and Scheduling. *Int. J. Adv. Manuf. Technol.* **2016**, *85*, 1513–1528. [[CrossRef](#)]
23. Shokouhi, E. Integrated Multi-Objective Process Planning and Flexible Job Shop Scheduling Considering Precedence Constraints. *Prod. Manuf. Res.* **2018**, *6*, 61–89. [[CrossRef](#)]
24. Zhang, X.; Zhang, H.; Yao, J. Multi-Objective Optimization of Integrated Process Planning and Scheduling Considering Energy Savings. *Energies* **2020**, *13*, 6181. [[CrossRef](#)]
25. Luo, G.; Wen, X.; Li, H.; Ming, W.; Xie, G. An Effective Multi-Objective Genetic Algorithm Based on Immune Principle and External Archive for Multi-Objective Integrated Process Planning and Scheduling. *Int. J. Adv. Manuf. Technol.* **2017**, *91*, 3145–3158. [[CrossRef](#)]
26. Chaudhry, I.A.; Usman, M. Integrated Process Planning and Scheduling Using Genetic Algorithms. *Teh. Vjesn.-Tech. Gaz.* **2017**, *24*, 1401–1409. [[CrossRef](#)]
27. Mohapatra, P.; Nayak, A.; Kumar, S.K.; Tiwari, M.K. Multi-Objective Process Planning and Scheduling Using Controlled Elitist Non-Dominated Sorting Genetic Algorithm. *Int. J. Prod. Res.* **2015**, *53*, 1712–1735. [[CrossRef](#)]
28. Wen, X.; Wang, K.; Li, H.; Sun, H.; Wang, H.; Jin, L. A Two-Stage Solution Method Based on NSGA-II for Green Multi-Objective Integrated Process Planning and Scheduling in a Battery Packaging Machinery Workshop. *Swarm Evol. Comput.* **2021**, *61*, 100820. [[CrossRef](#)]
29. Li, X.; Wen, X.; Gao, L. An Effective Genetic Algorithm for Multi-Objective Integrated Process Planning and Scheduling with Various Flexibilities in Process Planning. *J. Univers. Comput. Sci.* **2014**, *20*, 1926–1950.
30. Wen, X.; Qian, Y.; Lian, X.; Zhang, Y.; Wang, H.; Li, H. Improved Genetic Algorithm Based on Multi-Layer Encoding Approach for Integrated Process Planning and Scheduling Problem. *Robot. Comput.-Integr. Manuf.* **2023**, *84*, 102593. [[CrossRef](#)]
31. Kim, Y.K.; Park, K.; Ko, J. A Symbiotic Evolutionary Algorithm for the Integration of Process Planning and Job Shop Scheduling. *Comput. Oper. Res.* **2003**, *30*, 1151–1171. [[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.