

Article



Adaptive Population NSGA-III with Dual Control Strategy for Flexible Job Shop Scheduling Problem with the Consideration of Energy Consumption and Weight

Mingliang Wu^{1,2}, Dongsheng Yang^{1,2}, Bowen Zhou^{1,2}, Zhile Yang^{3,*}, Tianyi Liu^{1,2}, Ligang Li⁴, Zhongfeng Wang⁴ and Kunyuan Hu⁴

- ¹ Intelligent Electrical Science and Technology Research Institute, Northeastern University, Shenyang 110819, China; 1970754@stu.neu.edu.cn (M.W.); yangdongsheng@mail.neu.edu.cn (D.Y.); zhoubowen@ise.neu.edu.cn (B.Z.); 2070716@stu.neu.edu.cn (T.L.)
- ² Key Laboratory of Integrated Energy Optimization and Secure Operation of Liaoning Province, Northeastern University, Shenyang 110819, China
- ³ Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
- ⁴ Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110169, China; lilg@sia.cn (L.L.); wzf@sia.cn (Z.W.); hukunyuan@sia.cn (K.H.)
- Correspondence: zl.yang@siat.ac.cn

check for updates

Citation: Wu, M.; Yang, D.; Zhou, B.; Yang, Z.; Liu, T.; Li, L.; Wang, Z.; Hu, K. Adaptive Population NSGA-III with Dual Control Strategy for Flexible Job Shop Scheduling Problem with the Consideration of Energy Consumption and Weight. *Machines* **2021**, *9*, 344. https://doi.org/10.3390/machines 9120344

Academic Editor: Yuansong Qiao and Seamus Gordon

Received: 11 November 2021 Accepted: 1 December 2021 Published: 9 December 2021

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



Copyright: © 2021 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/). Abstract: The flexible job shop scheduling problem has always been the focus of research in the manufacturing field. However, most of the previous studies focused more on efficiency and ignored energy consumption. Energy, especially non-renewable energy, is an essential factor affecting the sustainable development of a country. To this end, this paper designs a flexible job shop scheduling problem model with energy consideration more in line with the production field. Except for the processing stage, the energy consumption of the transport, set up, unload, and idle stage are also included in our model. The weight property of jobs is also considered in our model. The heavier the job, the more energy it consumes during the transport, set up, and unload stage. Meanwhile, this paper invents an adaptive population non-dominated sorting genetic algorithm III (APNSGA-III) that combines the dual control strategy with the non-dominated sorting genetic algorithm III (NSGA-III) to solve our flexible job shop scheduling problem model. Four flexible job shop scheduling problem instances are formulated to examine the performance of our algorithm. The results achieved by the APNSGA-III method are compared with five classic multi-objective optimization algorithms. The results show that our proposed algorithm is efficient and powerful when dealing with the multi-objective flexible job shop scheduling problem model that includes energy consumption.

Keywords: flexible job shop scheduling problem; energy consumption; makespan; NSGA-III; dual control strategy

1. Introduction

Facing the global competition of the integration of the world economy, if the manufacturing industry wants to stand out from the cruel survival of the fittest, it must accelerate its response to external changes, improve product quality and performance, reduce various costs in the process, and provide customer-based personalized service on-demand [1,2]. At the same time, affected by the deterioration of the climate and the greenhouse effect, society and the country have also put forward higher and higher requirements for the green production of enterprises [3,4]. The manufacturing industry, which accounts for half of the total carbon emissions, has also attracted the attention of society and governments at all levels [5,6]. Therefore, ensuring a high-efficiency production rhythm in production and processing while seeking to improve energy efficiency and reduce total carbon emissions has become the direction of the study efforts of experts and scholars [7,8]. The manufacturing industry plays an indispensable role in the economic development and social construction of a country [9,10]. At the same time, the manufacturing industry is also an area where a country invests a tremendous amount of energy consumption [11,12]. According to the international energy outlook 2021 reference case, after a period of decline in coal consumption through 2030, consumption of all major fuels will grow from 2030 to 2050. Renewable energy consumption will more than double between 2020 and 2050, and renewable energy consumption will nearly equal liquid fuels consumption by 2050 [13]. In China, manufacturing companies consumed more than 50% of the national electrical energy and generated at least 26% of the total carbon dioxide emission. As the primary industry of the national economy, the manufacturing industry plays a crucial role in developing the national economy [14,15]. Still, it is also the main body of energy consumption and environmental pollution [16,17]. Promoting energy conservation and emission reduction in the manufacturing industry is of great significance to the global implementation of green manufacturing and actively responding to challenges such as global warming [18].

With the development of science and technology, the production scale and complexity of the manufacturing industry are getting higher and higher [19,20]. As a necessary means to solve the resource arrangement, flexible job shop scheduling problem (FJSP) plays an increasingly important position in the production and operation activities of the manufacturing industry [21,22]. FJSP refers to the process of making full use of the limited resources of the enterprise through reasonable allocation to meet or optimize one or more goals [23,24]. The FJSP is an essential part of the manufacturing process of the entire production job shop and controls the entire manufacturing system [25–27]. A reasonable scheduling plan can reduce the production cost of the enterprise, shorten the production cycle, improve the enterprise's competitiveness, and bring significant economic benefits to the enterprise [1,28]. With the advancement of green manufacturing, FJSP that considers energy saving and consumption reduction goals are essential for companies to achieve low-carbon development models [29,30].

Mouzon et al. devised a mathematical programming model for optimizing the scheduling problem of a single computerized numerical control (CNC) machine that considered the energy consumption and total completion time [31]. Afterward, Lei et al. deemed an energy-related objective should be considered thoroughly in FJSP with the advent of green manufacturing and proposed an FJSP with the consideration of energy consumption [32]. Additionally, to minimize makespan and total tardiness under the constraint that total energy consumption does not exceed a given threshold, a multi-objective FJSP with an energy consumption threshold is formulated by Wang et al. [29]. Jiang et al. established the mathematical model of the low-carbon FJSP with the objective of minimizing the sum of the energy consumption cost, and the earliness/tardiness cost. In another work, Meng et al. modeled the mixed integer linear programming (MILP) models for FJSP with the objective of minimizing total energy consumption [23].

However, research on reducing the energy consumption of manufacturing processes has concentrated chiefly on the machine, product, and production management levels [29]. There is much necessary energy consumption that has not been considered in previous studies [33]. Especially for small and medium-sized enterprises with small processing scales, improving processing efficiency is their first focus. With the continuous progress of the industrialization process, establishing an efficient scheduling model that contains more detailed energy consumption is an inevitable trend of development [2]. In this case, some minor energy consumption considerations become particularly important for saving costs and preventing environmental pollution [34]. For this reason, this paper considers various energy consumption components in the manufacturing industry that have not been considered in many previous studies but cannot be ignored, and designs the multiobjective FJSP considering the energy consumption of the transport, set up, processing, unload, and idle phase. Meanwhile, this paper also considers the weight property of the bars. The heavier the bar, the more energy will be consumed. A variety of multi-objective optimization algorithms have been proposed in many studies [35]. For multi-objective FJSP, Gao et al. combined the particle swarm algorithm and tabu search algorithm and proposed an effective hybrid algorithm to solve multi-objective FJSP with several conflicting and incommensurable objectives [36]. In another study, Gao et al. proposed a discrete harmony search algorithm for FJSP with minimization of multiple objective functions of the maximum of the completion time and the mean of earliness and tardiness [37]. Moreover, Li et al. proposed an effective hybrid tabu search algorithm (HTSA) to solve the FJSP with three minimization objectives of the maximum completion time (makespan), the total workload of machines, and the workload of the critical machine [38]. Lu et al. formulated the FJSP mathematical model with the objectives of minimizing both the makespan and the total additional resource consumption [39].

Most of the research has achieved excellent results for their issue. However, for FJSP as an NP-hard problem, the solution space will exponentially increase with jobs and operations [40–42]. Meanwhile, FJSP is a discrete problem; many algorithms that are good at dealing with continuous problems are not suitable for solving FJSP [43,44]. Many algorithms and improvement strategies are challenging to achieve a satisfactory effect for it [45,46]. The genetic algorithm (GA) is a very suitable tool for solving combinatorial optimization problems, such as FJSP. The non-dominated sorting genetic algorithm III (NSGA-III) algorithm is an extension of the GA in multi-objective optimization, which has been applied to various practical fields, so this paper adopts NSGA-III as a basic algorithm when solving FJSP. In order to further improve the solution ability of NSGA-III to prevent it from falling into the local optimum, and to enhance the diversity of the population, this paper proposes a dual control strategy (DCS) for multi-objective optimization capability of NSGA-III to avoid precocity and enhance the optimization capability of NSGA-III on solving combinatorial optimization problems such as FJSP. The highlights of this paper are as follows:

(1) The FJSP model considering energy consumption that covers the transport of AGV and set up, processing, unload, and idle of the machine is designed.

(2) The weight attribute of the bars is also taken into account. The heavier the bar, the more energy it consumes.

(3) The DCS for multi-objective optimization is formulated to enhance the solving ability of NSGA-III.

The rest of this work is arranged as follows. Firstly, the details of FJSP are elaborated in Section 2. Then, the procedure of the adaptive population non-dominated sorting genetic algorithm III (APNSGA-III) is described in Section 3. Followed by the extensive experiments that are conducted in Section 4. Finally, we summarize the paper in Section 5.

2. Problem Description

Classic FJSP can be described as: there are *n* jobs $J = (J_1, J_2, ..., J_n)$ that are processed on *m* machines $M = (M_1, M_2, ..., M_m)$ in a production job shop. Each job J_i consists of n_i operation, $O_{i,j}$ denotes the *jth* operation of the *ith* job. Each operation can be processed on any machine that belongs to machine set $M_{i,j}$. There are the following assumptions (constraints) about the FJSP [47].

(1) The completion time of the *jth* operation of the *ith* job is the sum of its start time and processing time.

$$C_{ijk} = S_{ijk} + T_{ijk} \tag{1}$$

where C_{ijk} , S_{ijk} , T_{ijk} implies the completion time, the starting time, and the processing time of the *jth* operation of the *ith* job on machine M_k , respectively.

(2) The start time of the *j*th operation of the *i*th job on the machine M_k is determined by the completion time of the previous operation q on the machine and the completion time of the previous operation of the job.

$$S_{ijk} = max\{S_{(i-1)qk} + T_{(i-1)qk}, C_{i(j-1)(k-1)}\}$$
(2)

(3) The downtime of machine M_k is the completion time of the last operation on the machine.

$$CT_k = C_{lk} \tag{3}$$

where *l* represents the last job on machine M_k , C_{lk} indicates the completion time of the last job on machine M_k .

(4) There are constraints on the machine required for the processing operation.

$$C_{gk} - C_{ik} + M(1 - x_{igk}) \ge T_{gk},$$

 $i, g = 1, 2, \dots, n; k = 1, 2, \dots, m$
(4)

where *n* and *m* denotes the number of jobs and machines, respectively. C_{gk} , T_{gk} indicates the completion time and processing time of job J_g on machine M_k , *M* means a sufficiently large positive number. x_{igk} represents the processing priority of job J_i and job J_g on machine M_k . If the job J_i is processed first, then x_{igk} is 1, otherwise it is zero.

(5) Operation O_{ij} and O_{gz} cannot be processed on any machine on machine set $M_{ij} \cap M_{gz}$ at the same time.

$$S_{ijk} \ge C_{gzk} - Z_{ijgzk} * M, \forall k \in M_{ij} \cap M_{gz}$$
(5)

$$S_{gzk} \ge C_{ijk} - (1 - Z_{ijgzk}) * M, \forall k \in M_{ij} \cap M_{gz}$$

$$\tag{6}$$

where Z_{ijgzk} represents that operation O_{ij} is processed before operation O_{gz} on machine M_k , 1 if yes, 0 otherwise. $M_{ij} \cap M_{gz}$ denotes the machine set that can process operation O_{ij} and operation O_{gz} at the same time.

(6) There is a priority relationship between the various operations of the same job.

$$S_{ijk} \ge C_{i(j-1)k}, k \in M_{ij},$$

 $\forall i = 1, 2, ..., n, \forall j = 2, ..., K_i$
(7)

where K_i represents the number of operation of job J_i .

(7) All jobs can be processed and all machines are optional when a task starts.

$$S_{ijk} \ge 0 \tag{8}$$

$$C_{ijk} \ge 0 \tag{9}$$

(8) Only one machine can be selected for each operation.

$$\sum_{k=1}^{m} R_{ijk} = 1$$
(10)

where R_{ijk} denotes whether the machine M_k processes operation O_{ij} , 1 means processed, 0 means unprocessed.

This paper considers two optimization objectives in the FJSP model that include the maximum completion time and the total energy consumption.

2.1. Maximum Completion Time (Makespan)

$$f_1 = \min(Makespan) = \min\max_{1 \le k \le m} \left\{ \max_{1 \le i \le n} CT_k \right\}$$
(11)

where *Makespan* represents the maximum completion time of all machines, it is the main indicator to measure the completion time of all jobs. CT_k is expressed as:

$$CT_k = \sum_{i=1}^{n} \sum_{j=1}^{K_i} (S_{ijk} R_{ijk} + R_{ijk} T_{ijk})$$
(12)

where R_{ijk} denotes whether the machine M_k processes operation O_{ij} , 1 means processed, 0 means unprocessed.

2.2. Total Energy Consumption (TEC)

The total energy consumption considered in this paper is mainly composed of the following five parts.

(a): The transporting energy consumption (*TTEC*)

TTEC is the energy consumed by the AGV transporting the job between machines. *TTEC* is related to the time AGV takes to transport the job and the weight of the job. It can be calculated by the following equation.

$$TTEC = \sum_{i=1}^{n} \sum_{j=1}^{K_i} T_{ij} W_i P$$
(13)

where T_{ij} implies the time spent in transporting the job J_i at the *jth* operation, W_i means the weight of the job J_i . *P* represents the energy consumed to transport a kilogram of objects per second.

(b): Total set up energy consumption (*TSEC*)

TSEC is the energy consumed by the machine to grab the job and set it on the processing table. Different operations of different jobs have different set up energy consumptions on different machines. Moreover, *TSEC* is positively related to the weight of the bar; the heavier the bar, the more energy it consumes. It can be calculated by the following equation.

$$TSEC = \sum_{i=1}^{n} \sum_{j=1}^{K_i} \sum_{k=1}^{m} S_{ijk} R_{ijk} W_i$$
(14)

where S_{ijk} represents the energy consumption per unit weight for machine M_k to set up O_{ij} , W_i denotes the weight of the job J_i .

(c): Total processing energy consumption (*TPEC*)

TPEC is the main energy consumption that is used to perform operations such as punching, cutting, laser printing, spray painting, etc. Different operations of different jobs have different processing energy consumptions on different machines. The calculation of *TPEC* is not given in Equation (15).

$$TPEC = \sum_{i=1}^{n} \sum_{j=1}^{K_i} \sum_{k=1}^{m} E_{ijk} R_{ijk}$$
(15)

where E_{ijk} means the energy consumed by machine M_k processing operation O_{ij} , R_{ijk} indicate whether the *jth* operation of the *ith* job is processed on machine M_k , 1 if yes, 0 otherwise.

(d): Total unloading energy consumption (*TUEC*)

Corresponding to *TSEC*, *TUEC* is the energy consumed by the unloading job by machine. Different operations of different jobs have different unload energy consumptions on different machines. Moreover, *TUEC* is positively related to the weight of the bar; the heavier the bar, the more energy it consumes. It can be calculated by the following equation.

$$TUEC = \sum_{i=1}^{n} \sum_{j=1}^{K_i} \sum_{k=1}^{m} U_{ijk} R_{ijk} W_i$$
(16)

where U_{ijk} represents the energy consumption per unit weight for machine M_k to unload processing operation O_{ij} , W_i denotes the weight of the job J_i .

(e): Total idling energy consumption (*TIEC*)

The machine in the job shop still consumes energy when it is not working, *TIEC* is the energy consumed when the machine is idle. Different machines consume different energy in standby state. It is given by following Equation (17).

$$TIEC = \sum_{i=1}^{m} N_i C_i \tag{17}$$

where N_i denotes the total idle time of machine M_i , C_i represents the energy consumption per minute when the machine is idle, *m* implies the number of machines. Therefore, the second optimization objective *TEC* can be formulated as follows:

$$f_{2} = min(TEC)$$

$$= min(TPEC + TTEC + TSEC + TUEC + TIEC)$$

$$= min(\sum_{i=1}^{n} \sum_{j=1}^{K_{i}} T_{ij}W_{i}P + \sum_{i=1}^{n} \sum_{j=1}^{K_{i}} \sum_{k=1}^{m} S_{ijk}R_{ijk}W_{i} + \sum_{i=1}^{n} \sum_{j=1}^{K_{i}} \sum_{k=1}^{m} E_{ijk}R_{ijk} + \sum_{i=1}^{n} \sum_{j=1}^{K_{i}} \sum_{k=1}^{m} U_{ijk}R_{ijk}W_{i} + \sum_{i=1}^{m} N_{i}C_{i})$$
(18)

Figure 1 shows the main structure of our FJSP model. Each job needs to go through I–IV statuses from raw material to finished product, namely, origin, set up, processing, and unload. Each machine has A–D functions, namely, set up, processing, unload, and idle. Automatic guided vehicle (AGV) has only one E function, which is to transport jobs. Status I to status II of the jobs need to be driven by AGV, and the energy consumed by this part will also be recorded in the total energy consumption, which is the transportation consumption. Status II–IV of one job needs to be implemented by one machine's A–C functions. Notably, the weight attribute of jobs is considered in the transport, set up, and unload stages. The heavier the bars, the more energy is consumed in these three stages. The processing stage of the job does not consider the weight attribute because in our model, we assume that the job is placed on the processing table at this stage, and there is no need to overcome gravity to do work. The machine is idle when it is not processing any jobs. Therefore, the total energy consumption of the workshop can be attributed to the sum of the energy consumption of the AGV transporting the job and the energy consumption of the four functions of the machine.



Figure 1. Main structure of our FJSP model.

To more vividly illustrate the details of energy consumption information mentioned before, we place the Gantt chart of the 3×3 scale FJSP into Figure 2. At the beginning of the task, the time for the job to reach the machine through transport will also be recorded in the scheduling. The job must go through the set up stage before being processed. After the operation of the job is processed, it must go through the unload stage. A machine can

only perform one of the set up, process, and unload stages on operations simultaneously. The transporting has nothing to do with any operation of the machine and it is related to AGV that is robot dedicated to transporting workpieces. As long as the job is finished performing the unload operation, it will be boxed and transported off the production line. Moreover, the job can be queued to be processed, as long as the unloading of the previous job on the machine is completed, and the setup stage can be carried out immediately.



Time

Figure 2. The Gantt chart of the instance mentioned before.

We can find that the idle time of the machine occupies a large proportion of the scheduling. It once again highlights the importance of investigating the energy consumption of the machine without a load.

An instance of the FJSP is given in Table 1. As it can be observed, there are three jobs to be processed on three machines. Job J_1 , J_2 , J_3 , all have three operations. The information of (set up time/ set up energy consumption), (processing time/process energy consumption), (unload time/unload energy consumption) are shown in Table 1. For instance, the available machines of the O_{11} are machine M_2 and machine M_3 . The processing time of O_{11} on machine M_2 and machine M_3 is 40 min and 46 min, respectively, while the set up energy consumption of O_{11} on machine M_2 and machine M_3 is 46 kWh and 65 kWh, respectively. The set up time of O_{11} on machine M_2 and machine M_3 is 9 min and 7 min, respectively, while the set up energy consumption of O_{11} on machine M_2 and machine M_3 is 2 min and 8 min, respectively, while the set up energy consumption of O_{11} on machine M_2 and machine M_3 is 2 min and 8 min, respectively, while the set up energy consumption of O_{11} on machine M_2 and machine M_3 is 11 kWh and 16 kWh, respectively. After completing the preceding operation of a job on a machine tool, the AGV will convey the job from the current machine to another machine. The transportation time between different machines is shown in Table 2.

Job	Operation	Machine	(Processing Time/Energy, Set Up Time/Energy, Unload Time/Energy)
J ₁	$O_{11} \\ O_{12} \\ O_{13}$	M_2, M_3 M_3, M_1 M_2, M_1	(40/46,9/21,2/11) (46/65,7/21,8/16) (53/46,7/9,7/6)(55/55,13/14,5/7) (55/64,8/18,6/4)(55/59,9/10,2/11)
J ₂	O ₂₁ O ₂₂ O ₂₃	M_1, M_2 M_2, M_3 M_2, M_3	(56/79,7/11,2/11) (53/77,10/48,8/7) (62/76,4/17,5/7)(54/77,4/26,5/16) (62/59,6/21,8/2)(52/59,12/17,7/10)
J ₃	$O_{31} \\ O_{32} \\ O_{33}$	$M_3, M_2 \ M_3, M_1 \ M_3, M_1$	(50/50,10/21,5/4) (32/52,7/21,7/4) (35/71,7/20,6/13)(41/83,8/23,9/15) (49/89,5/18,6/16)(63/76,12/16,9/14)

Table 1. The processing information of the instance mentioned before (unit: min/kWh).

Table 2. The transportation times of AGV between machines (unit: s).

Machine Number	Machine 1 (M_1)	Machine 2 (M_2)	Machine 3 (M ₃)
Machine 1 (M_1)	0	15	18
Machine 2 (M_2)	15	0	12
Machine 3 (M_3)	18	12	0

3. Improved NSGA-III

NSGA-III is the extension of the NSGA-II. NSGA-III and NSGA-II have a similar framework. The difference between the two is mainly the change in the selection mechanism. NSGA-II primarily relies on the crowding degree for sorting, and its role in high-dimensional target space is not prominent, while NSGA-III has drastically adapted the crowding degree sorting. Maintain the diversity of the population by introducing widely distributed reference points. The individual selection mechanism of NSGA-III is shown as follows.

3.1. Non-Dominated Sorting

Suppose the individual number of the current population P_t is N, that is $|P_t| = N$. Generate Q_t through genetic operator, that is $|Q_t| = N$. The parent and child populations are combined as $R_t = P_t \cup Q_t$. Sort R_t non-dominated and divide it into multiple non-dominated levels $(F_1, F_2, ...)$. Put the population members of the non-dominated level 1 to l into $|S_t|$ in turn. If $|S_t| = N$, $P_{t+1} = S_t$. If $|S_t| > N$, then part of the next generation is $P_{t+1} = \bigcup_{i=1}^{l-1} F_i$, the remaining individuals are selected in F_l . The detailed selection procedure will be expanded in the following subsection.

3.2. Determination of Reference Points on a Hyperplane

The reference points of NSGA-III can be generated by a structured manner or supplied preferentially by the user. There are a variety of popular reference point preset methods; we introduce two popular methods.

Das and Dennis's systematic approach: The reference point is on a (M-1)-dimensional hyperplane, and M is the dimension of the target space. If we divide each target into H parts, then the number of reference points is:

$$P = \binom{m+H-1}{H} \tag{19}$$

For a three-objective problem with H = 4, the reference points form a triangle and 15 reference points are generated according to formula (20).

The coordinates of each reference point are generated by the following rules:

(1): Let X be all the (M-1) combinations of $\{ \overset{0}{H}, \overset{0}{H}, \ldots, \overset{H+M-2}{H} \};$

(2): For each
$$x_{ij} \in X$$
, $x_{ij} = x_{ij} - \frac{j-1}{H}$;

(3): Let *S* be the reference point set, for each $s_{ij} \in S$ and $x_{ij} \in X$,

$$\begin{cases} s_{ij} = x_{ij} - 0, & j = 1\\ s_{ij} = x_{ij} - x_{i(j-1)}, & 1 < j < M\\ s_{ij} = 1 - x_{i(j-1)}, & j = M \end{cases}$$
(20)

Deb and Jain's method: It is suggested to use two layers of reference points, which are generated as follows [48]:

(1): Generate S_1 by Das and Dennis's method as the point set on the boundary layer; (2): Let S_2 be the point set on the inside layer, for each $S'_{ii} \in S_2$ and $s_{ij} \in S_1$,

$$S_{ii}' = \frac{1}{2}s_{ij} + \frac{1}{2M};$$
(21)

(3): The reference point set $S = S_1 \cup S_2$.

3.3. Adaptive Normalization of Population Individuals

First, the ideal point of the population S_t is determined by identifying the minimum value z_i^{min} for each objective function i = 1, 2, ..., M by constructing the ideal point $\bar{z} = (z_1^{min}, z_2^{min}, ..., z_M^{min})$. Compute each dimension of the ideal point by $z_j^{min} = min_{s \in S_t} f_j(s)$, and translate the objective by $f'_j(s) = f_j(s) - z_j^{min}$. Then, find the extra points corresponding to each coordinate axis using (22) and (23).

$$ASF = (x, w) = max_{i=1}^{M} f'_{i}(x) / w_{i}, x \in S_{t}$$
(22)

$$z^{i,max} = s : argmin_{s \in S_t} ASF(s, w^i), w^i = (\tau, \dots, \tau), \tau = 10^{-6}, w^i_j = 1$$
(23)

For the *ith* translate objective f'_i , generate an additional objective vector $z^{i,max}$. *M* additional objective vectors will form an *M*-dimensional linear hyperplane. Find the intercept $a_i, i = 1, 2, ..., M$. Then, the objective function can be normalized to:

$$f_i^n(x) = f_i'(x)/(a_i - z_i^{min}) = (f_i'(x) - z_i^{min})/(a_i - z_i^{min}), i = 1, 2, \dots, M$$
(24)

where the function value of the intersection of the normalized plane and the coordinate axis $f'_i = 1$, the points on this normalized hyperplane satisfy $\sum_{i=1}^{M} f_i^n = 1$.

3.4. Link the Individuals to the Reference Points

After normalizing each objective adaptively based on the extent of members of S_t in the objective space, it needs to associate each population member with a reference point. For this purpose, NSGA-III defines a reference line corresponding to each reference point on the hyperplane by joining the reference point with the origin. Then, calculate the perpendicular distance of each population member of S_t from each of the reference lines. The reference point whose reference line is closest to a population member in the normalized objective space is considered to be associated with the population member.

3.5. Select Individuals

It is worth noting that a reference point may have one or more population members associated with it or need not have any population member associated with it. NSGA-III counts the number of population members from $P_{t+1} = S_t/F_l$ that are associated with each reference point. Denote this niche count as ρ_j for the *jth* reference point. NSGA-III devises a new niche-preserving operation as follows. First, identify the reference point set $J_{min} = \{j : argmin_j\rho_j\}$ having minimum ρ_j . In the case of multiple such reference points, one $(\overline{j} \in J_{min})$ is chosen at random.

If $\rho_{\bar{j}} = 0$ (meaning that there is no associated P_{t+1} member to the reference point \bar{j}), there can be two scenarios with \bar{j} in set F_l . First, there exists one or more members

in front F_l that are associated with the reference point \overline{j} . In this case, the one having the shortest perpendicular distance from the reference line is added to P_{t+1} . The count $\rho_{\overline{j}}$ for reference point \overline{j} is then incremented by one. Second, the front F_l does not have any member associated with the reference point \overline{j} . In this case, the reference point is excluded from further consideration for the current generation.

In the event of $\rho_{\bar{j}} \ge 0$ (meaning that one member associated with the reference point already exists in S_t/F_l), a randomly² chosen member, if it exists, from front F_l that is associated with the reference point \bar{j} is added to P_{t+1} . The count $\rho_{\bar{j}}$ is then incremented by one. After niche counts are updated, the procedure is repeated for a total of *K* times to fill all of the vacant population slots of P_{t+1} .

3.6. The Framework of APNSGA-III

In the selection operator of ordinary GA, solutions with lower fitness will always be eliminated. Sometimes inferior individuals also have a probability of becoming a superior solution. Frequent elimination of them will not only waste resources but also reduce randomness and reduce global search capabilities. To this end, this paper designs a DCS for multi-objective optimization to enhance the performance of the NSGA-III.

The details of the APNSGA-III are shown in Algorithm 1 and Figure 3. First, *pop* that represents the entire population, *best* that is the best solution found so far, *Best_notEnhance* that records how many iterations *best* has not enhanced, and *NotEnhance*; that records how many iterations the solution x_i has not enhanced are initialized in lines 1 to 5. In every iteration, N needs to be saved in N_{tmp} before the main for loop because the N will change in the loop (pop_inc in line 16). The new solution m_i is generated and evaluated after the genetic operator, including the mutation operator and crossover operator (in line 9). If m_i is better than its predecessor vector x_i , m_i will replace the old x_i , and NotEnhance_i will be reset to zero in line 12; otherwise, *NotEnhance*, will increase by one because no better solution is found, and we still give a chance for the m_i entering the population. Therefore, if the best solution during the iteration has not been updated for a long time (*Best_notEnhance* \geq *T*), and the current population size N is smaller than N_{max} , then the pop_inc strategy will be used. After this, the best solution x_{best} in the current generation (in line 17), the worst solution in the current generation x_{worst} (in line 18), the best solution found so far best, and Best_notEnhance (in lines 19 to 23) are updated, respectively. After the generation of the whole population, the pop_dec strategy is used to delete some pessimistic individuals in line 24. Then, the iteration until the terminal condition is satisfied. The pop_inc and pop_dec strategies are introduced in the following sections.

3.6.1. The pop_inc Strategy

The population size of most meta-heuristic algorithms is fixed, but this will limit the diversity of the population to a certain extent and lead to a local optimum. When the algorithm has not improved the optimal solution for many consecutive generations or has little improvement, it is important to increase the population size and introduce new individuals to enhance diversity and escape from the optimal. Vectors m_i , generated by the genetic operation, will be helpful to increase the population diversity and may become promising solutions in the later evolution. Therefore, the pop_inc strategy is designed to give these worse vectors m_i the opportunity to be added into the population, shown as Algorithm 2. The pop_inc strategy will be triggered if the solution *best* has been not changed for *T* iterations and *N* is smaller than N_{max} in line 15 of Algorithm 1.



Figure 3. The flowchart of APNSGA-III.

If the *best* is not improved for a long time, the algorithm may stagnate and fall into a local optimum. Then, adding some worse vectors may help perturb the population and jump out of the local optimum, increasing the population's diversity.

Algorithm 1: APNSGA-III

1 II	nitialize <i>pop</i> and set other parameters;
2 S	elect the best individual as <i>best</i> ;
з В	$est_notEnhance \leftarrow 0;$
4 fe	or $i=1,i++,i\leq=N$ do
5	NotEnhance _i \leftarrow 0;
6 W	while Iter <max_iteration do<="" td=""></max_iteration>
7	$N_{tmp} \leftarrow N;$
8	for $i = 1, i + +, i <= N_{tmp}$ do
9	Generate m_i by the genetic operator;
10	if $f(m_i) < f(x_i)$ then
11	$x_i \leftarrow m_i;$
12	$NotEnhance_i \leftarrow 0;$
13	else
14	$NotEnhance_i := NotEnhance_i + 1;$
15	if $Best_notEnhance \ge T$ and $N < N_{max}$ then
16	Implement the pop_inc(<i>m</i> _i , <i>pop</i>);
17	Update <i>x</i> _{best} in the current iteration;
18	Update <i>x</i> _{worst} in the current iteration;
19	if $f(x_i) < f(x_{best})$ then
20	$best \leftarrow x_i, f(best) \leftarrow f(x_i);$
21	Best_notEnhance $\leftarrow 0$;
22	else
23	Best_notEnhance := Best_notEnhance + 1;
24	pop_aec(pop);
25	$_{1ter} = 1ter + 1;$

	• • •	•		•	/	,	`
	marithm		$n_{0}n_{1}$	100	m. ·	ทกท	۱.
A 13	201111111	<u> </u>	DUD		11111	vvv	
	0		г~г-		(г - г /	<i>'</i>

1	Add m_i into pop	$pop_{NP+1} \leftarrow m_i;$
2	N := N + 1;	
3	<i>NotEnhance</i> _N \leftarrow 0;	

3.6.2. The pop_dec Strategy

Although the pop_inc strategy will increase the diversity of the population to help the algorithm escape from the local optimum, if there is no population size, only increasing it will waste computer memory and delay the algorithm convergence speed. To this end, the pop_dec strategy is proposed to delete some inefficient individuals from the current population and ensure that *N* is smaller than N_{max} . As shown in line 24 of Algorithm 1, the pop_dec strategy procedure is executed after an iteration of the evolution. To decide which individuals are to be deleted, a factor named "exacerbation value" (the ex value) of solutions is designed as:

$$ex(x_i) = \frac{\sum_{i=1}^{n} (f(x_i) - f(best)) + 1.0}{\sum_{i=1}^{n} (f(x_{worst}) - f(best)) + 1.0}$$
(25)

where *n* denotes the number of optimization objectives. The addition of the constant "1.0" is to avoid the divisor of zero. If $ex(x_i)$ is large, it represents that $f(x_i)$ is worse and close to $f(x_{worst})$, or the solution x_i has not changed for a long time. Then, this solution x_i may be hopeless and will be deleted from the population.

The details of the pop_dec strategy are shown in Algorithm 3. If N is bigger than N_{min} , some inefficient solutions should be deleted from the population. For each solution

 x_i , the exacerbation value $ex(x_i)$ is calculated as (25). If $ex(x_i)$ is bigger than ex_{max} , and x_i ranks after 30% in population, this solution will be deleted in line 6 of Algorithm 3, and N will reduce by one. After deletion, if N reduces to N_{min} , the loop among lines 3 to 9 will end. Therefore, the population size N can be clamped in a certain range between N_{min} and N_{max} .

Algorithm 3: pop_dec(pop)

1 i	f $N > N_{min}$ then
2	$N_{tmp} \leftarrow N;$
3	for $i = 1, i + +, i <= N_{tmp}$ do
4	Calculate $ex(x_i)$;
5	if $ex(x_i) > ex_{max}$ and x_i ranked after 30% then
6	Delete x_i from pop;
7	N := N - 1;
8	if $N < N_{min}$ then
9	break;

The pop_dec strategy simultaneously considers fitness values and the update frequency of solutions, which greatly increases the fault tolerance rate that only considers fitness. Besides, there is no need to sort solutions by the fitness values in the pop_dec strategy, which helps save time.

Gao et al. designed an efficient chromosome representation to reduce decoding costs [49]. Due to its structure and coding rules, it does not require a repair mechanism and is very suitable for the integer optimization characteristics of NSGA-III. Therefore, this paper introduces this method to encode and decode our FJSP model, which helps the algorithm to solve the problem we put forward.

4. Experiment and Analysis

Since most of the previous studies take completion time as their main research goal, finding a benchmark containing time and energy consumption information is not easy. This paper learns from the Brandimarte rule and devises FJSP instances generation rules considering energy consumption and time, which can be described in Table 3.

Stage	aproc	mot
Process	40-60	[-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Set up	5-10	[-2, -1, 0, 1, 2, 3]
Unload	3–6	[-3, -2, -1, 0, 1, 2, 3]
Process	50-80	[-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Set up	10-20	[-6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6]
Unload	5-10	[-4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6]
	Stage Process Set up Unload Process Set up Unload	StageaprocProcess40-60Set up5-10Unload3-6Process50-80Set up10-20Unload5-10

Table 3. The FJSP instances generation rules.

aproc: time or energy consumption per operation; mot: offset time or energy consumption boundaries on per machine.

Without loss of generality, this paper randomly designs four FJSP instances according to the above rules to verify the effectiveness of the APNSGA-III. The general information of these four instances is shown in Table 4. The weight information of each job in each instance is placed in Table 5. The transport time between machines that are generated by arbitrary rules is placed in Table 6. In addition, the detailed information of above four instances is displayed in Tables 7–10. According to the suggestion of literature [50], the parameters *CR* (crossover rate) and *MR* (mutation rate) in APNSGA-III is set as 0.8 and 0.1, respectively. Referring to the recommendation of literature [51], *T* and ex_{max} in DCS is set as 15 and 50, respectively. As *p* in *TTEC* is just a positive proportionality coefficient and will not have

any effect on our experiments and models, we set it as 1. NSGA-II [52], NSGA-III [48], MOEA/D [53], MOWAS [54], and DEMO [55] are regarded as the comparison algorithm of APNSGA-III. The parameter settings of this algorithm refer to the suggested values of the original paper. The *N* (population size) and *Max_iteration* (maximum iteration) of all of the algorithms are set to 100 and 100. The N_{max} , N_{min} of APNSGA-III is set as 150 and 100, respectively. The algorithm automatically stops when iterated 100 times. Each algorithm is executed 10 counts independently on each instance.

Instance	njob	nmac	nop	meq
Instance 1	6	10	[1,2,3,4,5]	[1,2,3,4,5]
Instance 2	10	10	[2,3,4,5,6]	[1,2,3,4,5]
Instance 3	12	10	[2,3,4,5]	[1,2,3,4,5]
Instance 4	15	10	[2,3,4,5,6]	[1,2,3,4,5]

Table 4. The general information of four instances.

njob: number of jobs; nmac: number of machines; nop: minimum and maximum number of operation per job; meq: maximum number of available machine per operation.

In store on Nierros								Jo	bs						
Instance Name	J ₁	J ₂	J ₃	J_4	J_5	J ₆	J 7	J_8	J9	J ₁₀	J ₁₁	J ₁₂	J ₁₃	J ₁₄	J15
Instance 1	8	5	9	6	2	10	3	-	-	-	-	-	-	-	-
Instance 2	1	10	4	2	7	7	2	3	9	3	-	-	-	-	-
Instance 3	5	3	1	3	2	1	1	7	3	9	8	8	-	-	-
Instance 4	10	7	3	6	7	10	3	8	7	8	8	4	6	3	9

Table 5. The weight information of the job in each instance (unit: kg).

Machine	Origin	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	<i>M</i> ₁₀
Origing	0	144	134	185	165	194	172	116	146	149	150
M_1	144	0	152	170	193	100	112	173	165	142	133
M_2	134	152	0	131	138	152	169	120	170	162	143
M_3	185	170	131	0	160	151	140	132	171	122	140
M_4	165	193	138	160	0	140	140	165	170	140	198
M_5	194	100	152	151	140	0	103	102	170	180	192
M_6	172	112	169	140	140	103	0	142	140	148	150
M_7	116	173	120	132	165	102	142	0	150	162	160
M_8	146	165	170	171	170	170	140	150	0	141	120
M_9	149	142	162	122	140	180	148	162	141	0	153
M_{10}	150	133	143	140	198	192	150	160	120	153	0

Table 6. The transport time between machines (unit: s).

Job	Operation	Machine	Set Up Time (min)	Set Up Energy Consumption (kWh)	Processing Time (min)	Processing Energy Consumption (kWh)	Unload Time (min)	Unlaod Energy Consumption (kWh)
	<i>O</i> ₁₁	[6,1,7,8,4]	[10,4,4,5,10]	[8,13,15,5,6]	[43,48,47,36,53]	[62,70,71,58,71]	[5,3,7,7,3]	[14,11,13,7,5]
	O ₁₂	[4,7,9,1]	[11,8,8,11]	[19,10,18,17]	[69,55,58,51]	[64,58,54,64]	[7,3,5,4]	[5,9,5,12]
J_1	O ₁₃	[6,8,1]	[6,9,9]	[12,19,21]	[43,52,46]	[48,60,58]	[8,3,8]	[8,16,16]
	O_{14}	[3]	[7]	[12]	[54]	[74]	[6]	[9]
	O_{15}	[6,2,1,7]	[4,6,3,5]	[11,19,23,21]	[55,37,49,48]	[66,55,64,67]	[6,8,3,5]	[8,8,4,9]
	O ₂₁	[4]	[11]	[10]	[45]	[54]	[9]	[6]
J_2	O ₂₂	[4,1,3]	[7,12,12]	[23,15,24]	[56,44,56]	[]80,89,71]	[6,6,8]	[11,7,4]
	O ₂₃	[9,7,3]	[7,9,5]	[13,18,16]	[52,57,55]	[52,52,59]	[4,4,4]	[11,13,6]
J_3	O ₃₁	[7,4,10]	[17,15,17]	[15,18,17,12,16]	[42,55,40]	[77,79,70]	[2,2,5]	[9,12,10]
Ι.	O_{41}	[6,4]	[8,7]	[16,24]	[44,58]	[70,82]	[4,5]	[2,12]
J4	O ₄₂	[5,6,4,9,1]	[8,9,4,5,9]	[10,6,13,7,18]	[52,66,63,56,51]	[54,64,51,63,49]	[5,4,8,2,6]	[7,11,13,13,11]
J_5	O_{51}	[4,6,3,8]	[5,3,9,3]	[18,22,19,22]	[59,61,63,55]	[64,48,53,62]	[5,2,2,7]	[7,10,4,6]
	O ₆₁	[5]	[9]	[15]	[48]	[65]	[3]	[14]
J_6	O ₆₂	[3,6]	[8,10]	[10,19]	[40,42]	[65,56]	[7,7]	[14,10]
	O ₆₃	[1,10,7,3,8]	[5,10,11,8,6]	[17,16,7,10,7]	[43,45,52,42,46]	[72,70,80,77,71]	[8,7,8,4,5]	[14,9,12,10,13]
	O ₇₁	[9,5]	[8,10]	[17,15]	[34,33]	[53,49]	[5,5]	[12,11]
J_7	O ₇₂	[1,5]	[12,6]	[11,18]	[44,33]	[62,66]	[5,3]	[13,12]
	O ₇₃	[9,1]	[10,9]	[21,17]	[54,41]	[74,63]	[7,4]	[9,10]

Table 7. The job shop information of *Instance* 1.

Table 8. The job shop information of *Instance* 2.

Job	Operation	Machine	Set Up Time (min)	Set Up Energy Consumption (kWh)	Processing Time (min)	Processing Energy Consumption (kWh)	Unload Time (min)	Unlaod Energy Consumption (kWh)
	<i>O</i> ₁₁	[7,2,6]	[7,6,11]	[15,11,21]	[45,54,55]	[59,76,58]	[2,8,7]	[10,6,12]
	O ₁₂	[7,5]	[6,7]	[12,11]	[61,49]	[67,59]	[7,8]	[11,5]
L	O ₁₃	[1]	[7]	[8]	[44]	[51]	[5]	[12]
J1	O_{14}	[5,1,3,10]	[5,9,7,8]	[9,18,8,10]	[53,52,63,56]	[79,69,83,87]	[4,4,4,3]	[2,11,12,9]
	O ₁₅	[3,2,6]	[10,13,7]	[19,22,21]	[43,42,44]	[54,49,49]	[1,7,3]	[3,9,8]
	O_{16}	[1,7]	[9,11]	[15,11]	[40,48]	[47,48]	[6,9]	[5,2]
I.	O ₂₁	[2,3,9,10,7]	[8,11,7,10,8]	[19,21,22,23,17]	[53,43,43,54,47]	[59,52,62,68,63]	[2,1,4,2,1]	[4,10,2,3,11]
71	O ₂₂	[8,4]	[13,8]	[14,22]	[56,47]	[74,59]	[6,4]	[5,9]
	O ₃₁	[10,8]	[6,3]	[14,18]	[37,46]	[64,50]	[3,7]	[9,7]
Ŀ	O ₃₂	[9,6,8,10,2]	[7,9,9,9,12]	[12,14,10,22,13]	[52,45,56,48,51]	[68,63,52,51,56]	[8,8,8,6,6]	[14,2,14,13,7]
5	O ₃₃	[2]	[7]	[14]	[57]	[50]	[6]	[3]
	O ₃₄	[6]	[8]	[18]	[54]	[76]	[9]	[7]
I_4	O ₄₁	[2]	[6]	[14]	[45]	[88]	[14]	[11]
74	O ₄₂	[1,10,5,8,4]	[10,6,7,8,8]	[12,10,19,20,21]	[59,57,53,44,40]	[51,70,69,50,54]	[12,10,19,20,21]	[14,10,4,3,11]
	051	[5,9,1,8]	[9,7,6,10]	[15,9,10,6]	[47,49,49,52]	[84,84,70,77]	[3,2,3,3]	[16,9,7,14]
	052	[7,3,4,9,5]	[6,8,3,9,4]	[24,24,22,13,13]	[31,45,37,40,35]	[73,65,80,79,80]	[8,4,4,7,3]	[5,11,2,5,9]
J_5	053	[3,8]	[11,6]	[18,13]	[49,52]	[71,62]	[5,9]	[8,13]
	054	[10,1]	[8,9]	[9,15]	[44,32]	[83,69]	[1,5]	[6,12]
	055	[9,2]	[11,8]	[14,25]	[48,48]	[62,52]	[2,7]	[8,3]
	061	[2,9,8,3,7]	[6,6,11,6,9]	[12,18,11,18,7]	[52,39,45,49,34]		[5,8,3,4,5]	[7,10,8,8,10]
J_6	062	[2,1,7,3,10]	[11,13,11,7,10]	[19,16,16,12,18]	[50,53,53,54,59]	[/5,//,/5,64,/0]	[9,7,3,7,7]	[14,9,4,14,4]
	063	[1,7]	[/,/]	[14,10]	[40,39]	[37,30] [49.46.4E]	[1,2]	[5,10]
	064	[10,7,4]	[0,10,9]	[23,24,23]	[33,47,31]	[40,40,40]	[0,9,0]	[7,9,3]
	071	[7,10,3,6,4]	[12,0,7,10,11]	[3,7,13,13,6]	[43,33,32,36,47]		[9,0,3,0,0]	[4,4,7,1,7] [10 15 15 10 12]
I_	072	[2, 4, 0, 3, 0] [6 10 8 2 4]	[4,9,3,9,0]	[11,11,11,12,11]	[30,43,43,31,33]	[52,63,63,59,36]	[1, 4, 4, 5, 5] [3, 2, 2, 2, 5]	[8 2 9 5 5]
J7	073	[0,10,0,2,4]	[10,11,7,7,10]	[12,6,12]	[52 38 35]	[52,05,05,57,40]	[3,2,2,2,5]	[13 11 6]
	074	[7,4,0]	[3,2,0]	[12,0,12]	[32,30,35]	[57,04,37]	[3,3,5]	[7 15]
	0,5	[9 10]	[9 10]	[23 21]	[53,63]	[79 89]	[5 3]	[4 7]
	O_{81}	[14578]	[44697]	[17 16 9 7 14]	[54 70 62 52 52]	[53 54 57 56 49]	[43544]	[1469915]
I.	082	[64817]	[12.12.8.7.8]	[15.8.14.17.12]	[59,65,50,60,50]	[56,57,61,64,45]	[2,6,1,3,1]	[12.10.8.14.13]
18	0 ₈₄	[4,5,7,2,3]	[3.6.5.5.3]	[11.17.22.13.17]	[59,50,43,47,45]	[66,55,62,55,57]	[4.5.5.4.4]	[9.7.7.4.10]
	O ₈₅	[6]	[6]	[18]	[31]	[74]	[7]	[9]
	O_{91}	[6]	[6]	[13]	[49]	[77]	[4]	[12]
	O ₉₂	[1,8,3]	[6,8,10]	[13,21,15]	[53,54,51]	[61,64,51]	[7,7,4]	[13,2,7]
I9	O93	[6]	[6]	[16]	[60]	[63]	[6]	[4]
,,	O ₉₄	[1,4,9,7]	[8,6,8,7]	[8,14,12,9]	[46,48,61,49]	[68,69,58,71]	[8,3,6,2]	[15,6,5,14]
	O ₉₅	[7,8,1,10]	[9,7,9,10]	[7,17,16,11]	[66,53,58,64]	[65,73,72,61]	[6,4,6,3]	[5,10,12,10]
	O ₁₀₁	[9,10]	[12,8]	[9,16]	[48,41]	[67,62]	[6,4]	[14,8]
	O ₁₀₂	[7,1]	[6,11]	[26,20]	[34,52]	[55,61]	[3,7]	[9,2]
т	O ₁₀₃	[2,5,7,10,6]	[7,7,10,9,6]	[21,18,17,16,16]	[40,50,38,42,53]	[65,78,64,70,76]	[6,4,5,4,1]	[7,5,12,13,4]
J10	O ₁₀₄	[9,6,8]	[5,9,6]	[13,9,9]	[55,45,56]	[65,73,70]	[8,5,6]	[10,4,12]
	O ₁₀₅	[3,5]	[7,10]	[19,20]	[37,50]	[63,68]	[4,6]	[7,4]
	O ₁₀₆	[2,8,9,1,4]	[11,10,8,11,10]	[16,16,17,17,10]	[66,59,53,51,53]	[72,77,69,79,75]	[8,4,6,6,5]	[4,3,1,3,11]

Job	Operation	Machine	Set Up Time	Set Up Energy	Processing Time	Processing Energy	Unload Time	Unlaod Energy
			(1111)		(11111)		(1111)	
	<i>O</i> ₁₁	[10]	[9]	[9]	[52]	[69]	[7]	[15]
I1	<i>O</i> ₁₂	[8,3,4,9]	[9,6,4,8]	[12,15,13,13]	[46,48,46,44]	[71,68,78,69]	[3,6,3,4]	[4,9,12,9]
71	O_{13}	[6,1,5]	[10,12,12]	[17,11,8]	[42,56,38]	[58,52,57]	[3,6,7]	[11,11,11]
	O_{14}	[6]	[6]	[8]	[53]	[64]	[7]	[13]
	O ₂₁	[6,5]	[6,6]	[15,9]	[45,41]	[78,68]	[8,3]	[4,3]
In	O ₂₂	[1]	[4]	[18]	[46]	[50]	[9]	[10]
J2	O ₂₃	[8,4,9,6]	[7,8,8,9]	[15,16,19,18]	[56,42,51,52]	[52,68,59,56]	[3,5,7,8]	[10,7,6,10]
	O ₂₄	[3]	[9]	[12]	[46]	[71]	[7]	[3]
	O ₃₁	[7,10,8,9]	[10,9,8,10]	[15,12,20,21]	[51,56,58,44]	[68,56,50,61]	[4,7,3,6]	[8,13,3,13]
	O ₃₂	[2,1,7,9]	[6,8,5,9]	[11,11,21,16]	[60,68,49,58]	[68,79,75,85]	[8,5,7,7]	[9,11,12,8]
J_3	O ₃₃	[4]	[7]	[12]	[50]	[68]	[7]	[3]
	O ₃₄	[7]	[7]	[16]	[69]	[59]	[7]	[11]
	O ₃₅	[5,3,6,1,7]	[9,6,10,11,6]	[8,13,15,9,11]	[43,42,52,42,48]	[87,82,75,89,76]	[5,5,9,8,4]	[6,4,5,8,2]
т	O_{41}	[10,7,8,5,1]	[6,4,4,7,8]	[20,12,19,20,19]	[41,49,37,50,44]	[54,65,50,45,54]	[5,6,6,4,4]	[6,8,4,0,6]
J4	O ₄₂	[5,8,4,6]	[6,10,10,11]	[15,19,12,10]	[39,49,40,49]	[69,76,65,71]	[5,5,4,6]	[11,6,12,11]
	O ₅₁	[10,9,2,7]	[10,9,9,5]	[16,12,12,9]	[41,36,42,44]	[60,55,61,58]	[7,3,8,8]	[12,6,4,5]
J5	O ₅₂	[8]	[7]	[14]	[43]	[83]	[6]	[11]
	O ₅₃	[8,2]	[10,8]	[14,13]	[58,65]	[88,86]	[3,5]	[5,1]
	O ₆₁	[8,9,2]	[10,11,11]	[18,19,18]	[57,53,61]	[59,60,71]	[7,1,7]	[14,12,4]
т	O ₆₂	[3,5,10,1]	[7,9,7,6]	[8,13,15,16]	[42,51,57,41]	[50,60,64,48]	[2,4,2,2]	[8,1,8,7]
J6	O ₆₃	[5,1,6,9]	[13,12,9,9]	[15,23,13,18]	[36,47,37,48]	[52,52,65,72]	[5,6,9,5]	[6,6,14,11]
	O_{64}	[3,5,2,6]	[9,10,5,6]	[16,17,17,17]	[42,42,40,60]	[83,72,86,74]	[4,1,1,1]	[10,3,8,3]
	O ₇₁	[4,9,10,7,1]	[6,3,3,8,8]	[10,17,11,11,19]	[51,57,61,59,60]	[78,79,79,66,76]	[5,3,3,3,4]	[13,9,13,11,15]
J7	O ₇₂	[3]	[7]	[22]	[42]	[70]	[2]	[2]
	O_{81}	[5,2,6]	[7,5,5]	[14,24,22]	[46,45,38]	[80,76,65]	[3,4,4]	[8,9,9]
I_8	O ₈₂	[10,2,5,4,3]	[6,5,3,5,6]	[10,16,13,17,15]	[53,49,40,48,49]	[62,59,56,51,54]	[2,2,5,2,6]	[2,13,13,12,5]
	O ₈₃	[8,9,2]	[8,9,9]	[14,7,11]	[57,52,40]	[79,62,63]	[3,5,8]	[10,2,10]
	O ₉₁	[1,9,10,3,2]	[7,9,9,10,6]	[20,16,17,15,11]	[50,50,40,43,47]	[80,82,85,69,80]	[2,5,6,5,1]	[6,9,12,16,9]
J9	O ₉₂	[4,7]	[3,9]	[18,23]	[51,51]	[46,43]	[8,8]	[9,11]
	O ₁₀₁	[2]	[3]	[13]	[59]	[80]	[4]	[3]
T	O ₁₀₂	[6,4,2]	[10,10,13]	[19,21,17]	[50,55,64]	[62,60,55]	[5,3,6]	[10,13,3]
J ₁₀	O ₁₀₃	[2,5,4,1,10]	[11,13,7,8,13]	[13,7,12,16,11]	[49,62,49,48,55]	[67,78,80,69,71]	[6,3,3,6,8]	[4,2,11,9,12]
	O_{104}	[4,9,3,5]	[9,6,3,7]	[17,25,21,13]	[51,60,60,59]	[77,74,86,79]	[6,7,5,7]	[9,5,4,5]
	O ₁₁₁	[1,3]	[11,11]	[22,19]	[64,51]	[58,66]	[4,3]	[10,1]
	O ₁₁₂	[1,10,5,4,2]	[11,12,10,8,9]	[8,15,17,14,18]	[53,41,53,46,39]	[74,78,69,74,72]	[7,7,7,6,6]	[3,2,10,8,8]
J11	O ₁₁₃	[4]	[4]	[18]	[64]	[69]	[6]	[10]
	O ₁₁₄	[10,9,3,2]	[12,9,11,13]	[16,10,18,15]	[46,52,51,50]	[72,71,64,69]	[6,6,4,4]	[8,13,8,11]
	O ₁₂₁	[4,10,2,7,5]	[10,7,10,5,8]	[15,18,12,13,19]	[53,55,59,50,62]	[56,65,62,60,74]	[5,5,3,6,9]	[4,8,1,1,6]
J ₁₂	O ₁₂₂	[6,8,5,7]	[11,9,10,9]	[20,21,20,18]	[54,53,56,66]	[64,55,56,61]	[5,8,3,5]	[14,13,13,14]

Table 9. The job shop information of *Instance* 3.

Table 10. The job shop information of *Instance* 4.

Job	Operation	Machine	Set Up Time (min)	Set Up Energy Consumption (kWh)	Processing Time (min)	Processing Energy Consumption (kWh)	Unload Time (min)	Unlaod Energy Consumption (kWh)
	<i>O</i> ₁₁	[6]	[11]	[12]	[49]	[69]	[8]	[14]
	O ₁₂	[4,3]	[7,8]	[21,13]	[55,65]	[79,59]	[2,5]	[10,2]
J_1	O ₁₃	[3,8,1]	[13,11,8]	[22,22,25]	[46,37,50]	[63,61,59]	[8,5,6]	[10,11,8]
	O_{14}	[7,8,5,6]	[9,5,8,5]	[13,7,8,10]	[45,46,56,45]	[56,70,61,60]	[5,9,3,8]	[2,1,11,10]
	O_{15}	[2,5]	[7,7]	[12,18]	[35,46]	[66,79]	[5,5]	[1,9]
	O ₂₁	[8,5,3]	[10,8,9]	[23,21,17]	[50,57,56]	[65,60,58]	[8,2,7]	[7,10,9]
	O ₂₂	[2,9,4]	[7,10,9]	[16,16,14]	[49,59,53]	[73,66,82]	[3,6,1]	[0,6,9]
I.	O ₂₃	[1,3]	[7,10]	[23,12]	[65,61]	[81,67]	[6,2]	[9,13]
J2	O ₂₄	[10,2,3]	[12,9,13]	[8,13,9]	[38,44,38]	[60,58,45]	[8,2,7]	[0,5,8]
	O ₂₅	[7,6,8,5]	[5,8,7,7]	[21,18,22,22]	[44,46,46,59]	[65,53,51,56]	[2,4,2,3]	[4,13,5,11]
	O ₂₆	[10]	[9]	[10]	[49]	[70]	[3]	[15]
I.	O ₃₁	[4]	[6]	[17]	[59]	[66]	[8]	[3]
13	O ₃₂	[8]	[6]	[5]	[65]	[65]	[3]	[10]
L	O_{41}	[5,9,8]	[5,5,9]	[22,22,17]	[52,42,54]	[52,63,45]	[7,7,5]	[10,9,8]
J4	O ₄₂	[5,1,3]	[7,12,6]	[21,23,11]	[63,49,62]	[66,64,65]	[2,3,6]	[7,6,5]
	O_{51}	[5,4,8,6]	[9,7,6,5]	[14,8,7,13]	[58,54,51,42]	[64,73,61,75]	[5,7,2,5]	[12,4,1,10]
I-	O ₅₂	[2,4]	[5,11]	[22,17]	[43,54]	[63,71]	[7,4]	[6,6]
15	O_{53}	[10,2]	[8,13]	[19,11]	[41,58]	[54,52]	[1,6]	[9,12]
	O_{54}	[2,9]	[6,10]	[23,15]	[51,36]	[62,64]	[5,5]	[15,5]
L	O_{61}	[10,3]	[4,6]	[12,18]	[51,48]	[59,54]	[3,9]	[14,15]
J6	O ₆₂	[5]	[5]	[9]	[42]	[74]	[2]	[7]

Job	Operation	Machine	Set Up Time (min)	Set Up Energy Consumption (kWh)	Processing Time (min)	Processing Energy Consumption (kWh)	Unload Time (min)	Unlaod Energy Consumption (kWh)
	O ₇₁	[1]	[12]	[10]	[40]	[63]	[6]	[5]
Ŧ	O ₇₁	[1,8,10,4,7]	[11,8,9,11,11]	[14,19,22,12,19]	[53,68,50,56,65]	[70,71,64,61,71]	[4,4,4,2,6]	[5,14,10,14,3]
	O ₇₃	[10,9,7,4]	[5,6,11,6]	[13,22,13,11]	[39,35,32,48]	[57,62,67,59]	[3,9,5,3]	[5,3,3,6]
J7	O ₇₄	[9,1,10,8]	[7,3,7,8]	[11,12,9,9]	[43,45,50,54]	[72,61,63,58]	[6,8,5,6]	[2,12,4,6]
	O ₇₅	[5,1,10,4,3]	[10,6,8,6,12]	[17,11,20,14,9]	[56,48,51,59,60]	[61,66,58,46,60]	[8,4,7,9,6]	[11,10,11,5,10]
	O ₇₆	[3,5,1,8]	[10,6,8,4]	[10,15,8,16]	[63,54,49,53]	[72,79,79,65]	[6,3,1,5]	[9,6,6,7]
	O_{81}	[9,4,10]	[11,12,11]	[18,18,7]	[58,59,67]	[62,77,76]	[4,3,6]	[10,9,13]
J_8	O ₈₂	[1,8,2,5]	[8,4,8,9]	[12,15,14,14]	[36,37,53,53]	[60,42,45,51]	[6,2,5,3]	[4,2,4,13]
	O ₈₃	[4,9]	[4,7]	[9,11]	[50,50]	[62,69]	[5,3]	[11,15]
T	O_{91}	[9,2,1]	[11,6,6]	[19,19,21]	[51,43,55]	[65,59,57]	[4,2,2]	[7,15,10]
J9	O ₉₂	[8,7,9,10]	[5,8,9,4]	[7,12,7,17]	[63,49,55,61]	[68,75,71,67]	[6,8,5,3]	[7,7,7,15]
	O ₁₀₁	[9,3,5,6]	[10,5,10,5]	[7,13,15,16]	[41,46,47,45]	[72,78,68,76]	[3,4,3,7]	[5,11,14,16]
L.	O ₁₀₂	[1]	[12]	[16]	[51]	[76]	[5]	[9]
J10	O ₁₀₃	[3,9]	[6,7]	[11,19]	[45,44]	[67,59]	[4,4]	[9,12]
	O ₁₀₄	[5,1,7,6]	[6,11,7,6]	[18,15,16,23]	[50,50,61,51]	[63,49,51,48]	[3,6,6,6]	[10,3,2,9]
	O ₁₁₁	[7,2,1,6,5]	[12,12,9,11,12]	[22,14,15,22,10]	[48,52,56,44,53]	[64,54,55,67,47]	[6,2,7,8,2]	[13,5,8,10,3]
J ₁₁	O ₁₁₂	[4]	[8]	[16]	[51]	[70]	[3]	[5]
	O ₁₁₃	[1,6,5]	[5,3,6]	[17,23,15]	[52,50,55]	[73,76,86]	[2,4,3]	[13,13,11]
	O ₁₁₄	[4,1,3,2,10]	[8,4,4,7,8]	[21,14,18,22,21]	[47,62,53,54,54]	[52,50,53,45,57]	[9,8,7,3,8]	[3,4,3,4,8]
	O ₁₂₁	[9,10,6]	[7,6,9]	[19,12,16]	[48,59,53]	[77,75,69]	[4,2,2]	[11,14,3]
	O ₁₂₂	[4,9]	[13,10]	[24,14]	[40,46]	[70,64]	[2,2]	[3,9]
T	O ₁₂₃	[1,3,5]	[8,8,13]	[21,13,14]	[57,54,64]	[52,54,44]	[4,7,5]	[3,12,5]
J12	O ₁₂₄	[6,8]	[8,11]	[13,16]	[49,43]	[65,57]	[3,5]	[12,10]
	O ₁₂₅	[9,3,7,10,8]	[12,6,6,12,11]	[18,19,13,9,10]	[38,55,48,54,39]	[44,43,43,56,44]	[6,7,8,8,3]	[13,6,12,2,8]
	O ₁₂₆	[5,8]	[8,13]	[16,18]	[60,60]	[72,67]	[7,4]	[10,3]
	O ₁₃₁	[10,1,4,2,7]	[4,9,6,4,9]	[5,11,5,8,8]	[53,54,48,35,42]	[60,58,56,65,65]	[4,6,4,7,7]	[12,3,11,4,6]
	O ₁₃₂	[3,8,1,10,9]	[8,9,9,6,5]	[18,13,10,15,15]	[51,55,54,60,59]	[64,50,51,65,53]	[4,5,5,7,7]	[2,9,1,1,0]
T	O ₁₃₃	[10,5,9]	[8,4,5]	[22,16,23]	[58,57,61]	[67,79,78]	[3,8,4]	[5,6,7]
J13	O ₁₃₄	[7,1,2]	[8,10,10]	[20,12,13]	[38,47,39]	[77,77,78]	[7,5,6]	[8,10,8]
	O ₁₃₅	[9,6,3,2,10]	[9,11,7,10,10]	[13,18,12,11,10]	[35,47,41,37,52]	[64,45,58,63,60]	[4,1,4,7,6]	[3,13,12,12,10]
	O ₁₃₆	[5,7,3]	[6,7,11]	[17,19,9]	[46,42,52]	[63,64,65]	[8,2,6]	[14,6,9]
	O ₁₄₁	[1]	[9]	[15]	[34]	[81]	[3]	[15]
	O ₁₄₂	[10]	[9]	[13]	[54]	[65]	[7]	[6]
J_{14}	O ₁₄₃	[8,5,1,3,7]	[4,5,4,5,8]	[9,12,9,17,15]	[51,59,53,61,50]	[59,68,72,70,58]	[7,3,5,8,5]	[4,6,4,4,1]
	O ₁₄₄	[1,3,10]	[8,12,11]	[14,23,23]	[60,51,43]	[76,65,80]	[8,4,9]	[3,0,10]
	O ₁₄₅	[10,8,7]	[5,6,9]	[17,18,19]	[50,50,41]	[69,81,71]	[9,7,3]	[11,8,9]
	O ₁₅₁	[9,8,5,10,1]	[10,7,10,11,11]	[10,7,15,13,12]	[38,37,38,42,54]	[64,71,61,61,54]	[2,5,3,2,2]	[13,3,8,2,12]
	O ₁₅₂	[5,1,3]	[10,9,9]	[23,25,17]	[41,39,35]	[55,60,69]	[7,6,3]	[11,6,2]
L	O ₁₅₃	[7,9,5,6]	[9,9,5,7]	[8,10,14,15]	[62,56,55,48]	[86,87,80,86]	[5,3,9,8]	[14,4,12,13]
J15	O ₁₅₄	[1,5,8]	[8,8,9]	[18,13,21]	[53,48,46]	[75,56,59]	[2,4,1]	[13,4,6]
	O ₁₅₅	[9,7,10,8,4]	[8,8,9,8,9]	[15,18,9,8,15]	[47,42,41,48,48]	[55,63,59,63,59]	[8,4,4,6,7]	[1,10,0,10,5]
	O ₁₅₆	[4]	[7]	[9]	[44]	[69]	[4]	[11]

Table 10. Cont.

4.1. Result Show and Analysis

The final comparison results of *Instance* 1, 2 are placed in Table 11, while the final comparison results of *Instance* 3, 4 are placed in Table 12. BT and BE denotes the minimum completion time and minimum processing energy consumption, respectively. AT and AE implies the average completion time and average processing energy consumption, respectively. For *Instance* 1, APNSGA-III obtains the optima on BE and AE, while APNSGA-III can get the optima on BT, BE, and AE for *Instance* 3. Better, APNSGA-III can obtain the optima on BT, AT, BE, and AE for *Instance* 2 and *Instance* 4. The performance of APNSGA-III far exceeds the other five comparison algorithms. In order to show the comparison of the algorithms in each goal more clearly, we rank the data in Tables 11 and 12 on a level of 1–20 from big to small and visually show them in Figure 4. The area from $0 - \pi/2$, $\pi/2 - \pi$, $\pi - 3\pi/2$, $3\pi/2 - 2\pi$ represents the ranking situation in *BT*, *AT*, *BE*, and *AE*, respectively. The Arabic numerals 1–4 around the circle represent *Instance* 1–4. We can see that the area enclosed by the thick red line representing APNSGA-III is fit to solve the actual question we designed.



Figure 4. Ranking of six algorithm.

Table 11. The results of Instance 1 and Instance 2 (* represents the optimal result).

Algorithm		Inst	ance 1			Bost num			
Algorithm	BT	AT	BE	AE	BT	AT	BE	AE	Dest num
APNSGA-III	323.8	323.3	9255.0 *	9271.8 *	423.6 *	424.4 *	12,663.0 *	12,701.9 *	6 *
NSGA-III	323.2	324.0	9297.7	9272.4	438.5	439.9	12,894.6	12,921.6	0
NSGA-II	323.4	323.3	9285.2	9296.4	441.2	446.4	13,109.6	13,223.1	0
MOEA/D	322.7 *	322.9 *	9305.6	9299.2	429.2	432.7	12,886.7	12,909.5	2
MOWAS	323.7	324.1	9326.7	9352.0	464.6	464.7	13,516.3	13,558.8	0
DEMO	322.9	323.2	9340.2	9312.2	441.2	448.0	13,022.4	13,167.0	0

Table 12. The results of Instance 3 and Instance 4 (* represents the optimal result).

Algorithm		Ins	stance 3			Bostnum			
Algorithm	BT	AT	BE	AE	BT	AT	BE	AE	Dest num
APNSGA-III	361.2 *	368.1	10,275.5 *	10,300.2 *	536.9 *	537.7 *	20,202.1 *	20,236.3 *	7*
NSGA-III	368.1	361.2 *	10,552.3	10,552.3	552.9	573.4	20,717.6	21,080.8	1
NSGA-II	364.5	364.7	10,574.3	10,626.3	557.8	561.5	20,731.2	20,794.4	0
MOEA/D	361.9	363.1	10,536.1	10,520.3	550.7	559.6	20,624.7	20,772.8	0
MOWAS	379.4	381.4	11,038.8	11,084.0	582.9	583.2	21,534.8	21,584.9	0
DEMO	375.3	385.0	10,776.2	10,962.8	575.5	611.4	21,388.7	22,095.5	0

4.2. Two Independent Sample T-Tests

Due to the small sample size included in our experiment, we need to use two independent sample t-tests to test whether there is an accidental similarity between APNSGA-III and other comparison algorithms. We set 0.05 as the level of significance. The hypothesis is accepted when p > 0.05, thus the test results indicate no significant difference between the two compared algorithms. The hypothesis is rejected when p <= 0.05, thus the test

results indicate significant differences between the APNSGA-III and contrasted algorithms. We will use the results of ten rounds of experiments that have been used to generate the average results in Tables 11 and 12 as the test sample. The test results are shown in Tables 13 and 14. We denote the value of p more than 0.05 as "+" and denote the value of p less than 0.05 as "-". 'Same' denotes the total number of APNSGA-III has no significant difference with other algorithms, while 'Better' represents the total number of APNSGA-III that has significant differences with other algorithms.

From Tables 13 and 14, we find that APNSGA-III and other algorithms are significantly different in almost all instances. Among the compared algorithms, NSGA-III is the closest to APNSGA-III, but there are still many differences between both. In general, we can conclude that APNSGA-III is significantly more excellent than the other algorithms.

Algorithm		Insta	nce 1		Instance 2				Sama	Dottor
Aigontinii	BT	AT	BE	AE	BT	AT	BE	AE	Same	Detter
NSGA-III	+	_	_	_	_	+	_	_	2	6
NSGA-II	_	_	_	_	_	_	_	_	0	8
MOEA/D	+	_	_	_	_	_	_	_	1	7
MOWAS	_	_	_	_	_	+	_	_	1	7
DEMO	—	_	—	+	_	_	—	—	1	7

Table 13. The *p* values of APNSGA-III and other algorithms on *Instance* 1 and *Instance* 2.

Algorithm		Instance 3				Insta	nce 4	Samo	Dattar	
Algorithm	BT	AT	BE	AE	BT	AT	BE	AE	Jame	Detter
NSGA-III	+	_	_	_	_	+	_	_	2	6
NSGA-II	_	_	_	_	_	+	_	_	1	7
MOEA/D	_	+	_	_	_	_	_	_	1	7
MOWAS	_	_	_	_	_	_	_	_	0	8
DEMO	_	_	_	_	—	_	—	_	0	8

Table 14. The *p* values of APNSGA-III and other algorithms on *Instance* 3 and *Instance* 4.

4.3. Convergence Analysis

This subsection shows the convergence of BT, ET for *Instance* 1–4 in Figures 5 and 6, respectively, and discusses the algorithm's performance in the generation. The red line representing APNSGA-III always converges to the lowest position in the end except for part (a) of Figure 5. Moreover, the red line is almost below the other color lines in parts (b), (c), and (d) of Figures 5 and 6. It means the performance of each stage of the entire optimization process of APNSGA-III is better than other comparison algorithms. It once again confirms the improvement of DCS and the excellent applicability of APNSGA-III.

4.4. Gantt Chart Display and Analysis

The Gantt chart of *Instance* 1–4 with the smallest energy consumption is shown in Figure 7. For *Instance* 1, the *Makespan* and *TEC* is 324.45 min and 9230.75 kWh, respectively. For *Instance* 2, the *Makespan* and *TEC* is 423.75 min and 12,340.45 kWh, respectively. For *Instance* 3, the *Makespan* and *TEC* is 358.05 min and 10,069.35 kWh, respectively. For *Instance* 4, the *Makespan* and *TEC* is 504.85 min and 19,803.75 kWh, respectively. There are few gaps for *Instance* 2–4. It means that our method is effective, and the final scheduling plan has considerable advantages. In addition to the processing stage, the other three stages, set up, unload, and idle, also occupy a large proportion in the Gantt chart, especially in part (a); a large number of gaps means that the machine is at no load many times. This is also a large part of the energy consumption. If this part of energy consumption is not counted as the total energy consumption, it is impossible to make an accurate



energy consumption analysis based on the existing scheduling situation, and thus it is impossible to achieve effective energy conservation.

Figure 5. The convergence curve of BT and BE for *Instance* 1 and *Instance* 2. (a) BT of *Instance* 1; (b) BE of *Instance* 1; (c) BT of *Instance* 2; (d) BE of *Instance* 2.



Figure 6. Cont.



Figure 6. The convergence curve of BT and BE for *Instance* 3 and *Instance* 4. (**a**) BT of *Instance* 3; (**b**) BE of *Instance* 3; (**c**) BT of *Instance* 4; (**d**) BE of *Instance* 4.



Figure 7. The Gantt chart of *Instance* 1–4 with the smallest energy consumption. (**a**) The Gantt chart of *Instance* 1; (**b**) the Gantt chart of *Instance* 2; (**c**) the Gantt chart of *Instance* 3; (**d**) the Gantt chart of *Instance* 4.

5. Conclusions

Considering the previous research which mostly concentrates more on efficiency and ignores energy consumption, this paper designs the FJSP model with consideration of energy consumption. Meanwhile, the work includes non-negligible energy consumption, including the transport, set up, unload, and idle stage. The weight property of the job is also considered in our model. Besides, a dual control strategy for multi-objective optimization is designed to enhance the performance of NSGA-III when faced with multi-objective FJSP. Four FJSP instances are formulated to examine the metrics of our algorithm. The results obtained by APNSGA-III are compared with another five classic multi-objective optimization algorithms. All of the results reply that Diagonal Orthant Latent Dirichlet Allocation (DOLDA) owns a competitive superiority in solving the global optimization of these FJSP instances.

In conclusion, the proposed algorithm shows a performance improvement compared with other multi-objective optimization algorithms, but further studies on the model accuracy and computation time are required through investigating more energy-efficient scheduling problem. On the other hand, unexpected events, such as machine failure, rush orders, and job cancellation, which may occur in the practical applications of a manufacturing system, should be considered in our FJSP. Reducing the energy consumption in a dynamic scheduling problem should also be studied in the future.

Author Contributions: Conceptualization, Z.Y.; Data curation, M.W. and B.Z.; Funding acquisition, D.Y.; Investigation, M.W. and T.L.; Methodology, Z.Y.; Project administration, D.Y.; Resources, B.Z., L.L., Z.W. and K.H.; Software, M.W.; Writing—original draft, M.W.; Writing—review & editing, Z.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by [National Key R&D Program of China] grant number [2018YFB1700500] and [National Science Foundation of China under grants] grant number [52077213 and 62003332]. The APC was funded by [National Key R&D Program of China] grant number [2018YFB1700500] and [National Science Foundation of China under grants] grant number [52077213 and 62003332].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was supported by the National Key R&D Program of China (2018YFB1700500) and the National Science Foundation of China under grants (52077213 and 62003332).

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

References

- 1. 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]
- Caldeira, R.H.; Gnanavelbabu, A.; Vaidyanathan, T. An effective backtracking search algorithm for multi-objective flexible job shop scheduling considering new job arrivals and energy consumption. *Comput. Ind. Eng.* 2020, 149, 106863. [CrossRef]
- Gong, X.; De Pessemier, T.; Martens, L.; Joseph, W. Energy-and labor-aware flexible job shop scheduling under dynamic electricity pricing: A many-objective optimization investigation. J. Clean. Prod. 2019, 209, 1078–1094. [CrossRef]
- 4. Zhang, Z.; Wu, L.; Peng, T.; Jia, S. An improved scheduling approach for minimizing total energy consumption and makespan in a flexible job shop environment. *Sustainability* **2019**, *11*, 179. [CrossRef]
- 5. Salido, M.A.; Escamilla, J.; Giret, A.; Barber, F. A genetic algorithm for energy-efficiency in job-shop scheduling. *Int. J. Adv. Manuf. Technol.* **2016**, *85*, 1303–1314. [CrossRef]
- 6. Bányai, T. Optimization of Material Supply in Smart Manufacturing Environment: A Metaheuristic Approach for Matrix Production. *Machines* 2021, *9*, 220. [CrossRef]
- 7. Li, M.; Lei, D. An imperialist competitive algorithm with feedback for energy-efficient flexible job shop scheduling with transportation and sequence-dependent setup times. *Eng. Appl. Artif. Intell.* **2021**, 103, 104307. [CrossRef]
- Gordillo, A.; Giret, A. Performance Evaluation of Bidding-Based Multi-Agent Scheduling Algorithms for Manufacturing Systems. Machines 2014, 2, 233–254. [CrossRef]

- 9. Liu, Y.; Dong, H.; Lohse, N.; Petrovic, S.; Gindy, N. An investigation into minimising total energy consumption and total weighted tardiness in job shops. *J. Clean. Prod.* **2014**, *65*, 87–96. [CrossRef]
- 10. Feng, Y.; Liu, M.; Zhang, Y.; Wang, J. A Dynamic Opposite Learning Assisted Grasshopper Optimization Algorithm for the Flexible JobScheduling Problem. *Complexity* **2020**, *2020*, 8870783. [CrossRef]
- 11. Yin, L.; Li, X.; Gao, L.; Lu, C.; Zhang, Z. Energy-efficient job shop scheduling problem with variable spindle speed using a novel multi-objective algorithm. *Adv. Mech. Eng.* **2017**, *9*, 1687814017695959. [CrossRef]
- 12. Gabriel, F.; Baars, S.; Römer, M.; Dröder, K. Grasp Point Optimization and Leakage-Compliant Dimensioning of Energy-Efficient Vacuum-Based Gripping Systems. *Machines* 2021, *9*, 149. [CrossRef]
- 13. Zhu, Z.; Lu, L.; Zhang, W.; Liu, W. EIA's International Energy Outlook Analyzes Electricity Markets in India, Africa, and Asia. 2020. Available online: https://www.eia.gov/outlooks/ieo/ (accessed on 10 November 2021).
- 14. Pereira, A.C.; Romero, F. A review of the meanings and the implications of the Industry 4.0 concept. *Procedia Manuf.* 2017, 13, 1206–1214. [CrossRef]
- 15. Zhang, S.; Du, H.; Borucki, S.; Jin, S.; Hou, T.; Li, Z. Dual Resource Constrained Flexible Job Shop Scheduling Based on Improved Quantum Genetic Algorithm. *Machines* **2021**, *9*, 108. [CrossRef]
- 16. Gadaleta, M.; Pellicciari, M.; Berselli, G. Optimization of the energy consumption of industrial robots for automatic code generation. *Robot. Comput.-Integr. Manuf.* **2019**, *57*, 452–464. [CrossRef]
- 17. Del Pero, F.; Berzi, L.; Antonacci, A.; Delogu, M. Automotive Lightweight Design: Simulation Modeling of Mass-Related Consumption for Electric Vehicles. *Machines* 2020, *8*, 51. [CrossRef]
- Hong, E.; Yeneneh, A.M.; Sen, T.K.; Ang, H.M.; Kayaalp, A. A comprehensive review on rheological studies of sludge from various sections of municipal wastewater treatment plants for enhancement of process performance. *Adv. Colloid Interface Sci.* 2018, 257, 19–30. [CrossRef]
- 19. Wu, M.; Yang, D.; Yang, Z.; Guo, Y. Sparrow Search Algorithm for Solving Flexible Jobshop Scheduling Problem. In Proceedings of the International Conference on Swarm Intelligence, Qingdao, China, 17–21 July 2021; pp. 140–154.
- Gholami, O.; Sotskov, Y.N. A fast heuristic algorithm for solving parallel-machine job-shop scheduling problems. Int. J. Adv. Manuf. Technol. 2014, 70, 531–546. [CrossRef]
- 21. Chaudhry, I.A.; Khan, A.A. A research survey: Review of flexible job shop scheduling techniques. *Int. Trans. Oper. Res.* 2016, 23, 551–591. [CrossRef]
- 22. Gu, X.; Huang, M.; Liang, X. An improved genetic algorithm with adaptive variable neighborhood search for FJSP. *Algorithms* **2019**, *12*, 243. [CrossRef]
- 23. 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]
- 24. Zeng, Q.; Wang, M.; Shen, L.; Song, H. Sequential scheduling method for FJSP with multi-objective under mixed work calendars. *Processes* **2019**, *7*, 888. [CrossRef]
- 25. Manne, A.S. On the job-shop scheduling problem. Oper. Res. 1960, 8, 219–223. [CrossRef]
- Yang, D.; Wu, M.; Yang, Z.; Guo, Y.; Feng, W. Dragonfly algorithm for Solving Flexible Jobshop Scheduling Problem. In Proceedings of the 2020 35th Youth Academic Annual Conference of Chinese Association of Automation (YAC), Zhanjiang, China, 16–18 October 2020; pp. 867–872.
- 27. Zhang, H.; Xu, G.; Pan, R.; Ge, H. A novel heuristic method for the energy-efficient flexible job-shop scheduling problem with sequence-dependent set-up and transportation time. In *Engineering Optimization*; Taylor & Francis: Abingdon, UK, 2021; pp. 1–22.
- 28. Park, J.S.; Ng, H.Y.; Chua, T.J.; Ng, Y.T.; Kim, J.W. Unified genetic algorithm approach for solving flexible job-shop scheduling problem. *Appl. Sci.* **2021**, *11*, 6454. [CrossRef]
- 29. Lei, D.; Li, M.; Wang, L. A two-phase meta-heuristic for multiobjective flexible job shop scheduling problem with total energy consumption threshold. *IEEE Trans. Cybern.* **2018**, *49*, 1097–1109. [CrossRef]
- 30. Luan, F.; Cai, Z.; Wu, S.; Liu, S.Q.; He, Y. Optimizing the low-carbon flexible job shop scheduling problem with discrete whale optimization algorithm. *Mathematics* **2019**, *7*, 688. [CrossRef]
- 31. Mouzon, G.; Yildirim, M.B.; Twomey, J. Operational methods for minimization of energy consumption of manufacturing equipment. *Int. J. Prod. Res.* 2007, 45, 4247–4271. [CrossRef]
- 32. Lei, D.; Zheng, Y.; Guo, X. A shuffled frog-leaping algorithm for flexible job shop scheduling with the consideration of energy consumption. *Int. J. Prod. Res.* 2017, 55, 3126–3140. [CrossRef]
- 33. Wu, X.; Shen, X.; Li, C. The flexible job-shop scheduling problem considering deterioration effect and energy consumption simultaneously. *Comput. Ind. Eng.* **2019**, *135*, 1004–1024. [CrossRef]
- 34. Wu, X.; Sun, Y. A green scheduling algorithm for flexible job shop with energy-saving measures. J. Clean. Prod. 2018, 172, 3249–3264. [CrossRef]
- 35. Wang, F.; Liao, F.; Li, Y.; Wang, H. A new prediction strategy for dynamic multi-objective optimization using Gaussian Mixture Model. *Inf. Sci.* **2021**, *580*, 331–351. [CrossRef]
- 36. Zhang, G.; Shao, X.; Li, P.; Gao, L. An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem. *Comput. Ind. Eng.* **2009**, *56*, 1309–1318. [CrossRef]
- 37. Gao, K.Z.; Suganthan, P.N.; Pan, Q.K.; Chua, T.J.; Cai, T.X.; Chong, C.S. Discrete harmony search algorithm for flexible job shop scheduling problem with multiple objectives. *J. Intell. Manuf.* **2016**, *27*, 363–374. [CrossRef]

- Li, J.q.; Pan, Q.k.; Liang, Y.C. An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems. *Comput. Ind. Eng.* 2010, 59, 647–662. [CrossRef]
- 39. Lu, C.; Li, X.; Gao, L.; Liao, W.; Yi, J. An effective multi-objective discrete virus optimization algorithm for flexible job-shop scheduling problem with controllable processing times. *Comput. Ind. Eng.* **2017**, *104*, 156–174. [CrossRef]
- 40. Álvarez-Gil, N.; Rosillo, R.; de la Fuente, D.; Pino, R. A discrete firefly algorithm for solving the flexible job-shop scheduling problem in a make-to-order manufacturing system. *Cent. Eur. J. Oper. Res.* **2021**, *29*, 1353–1374. [CrossRef]
- 41. Chen, R.; Yang, B.; Li, S.; Wang, S. A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem. *Comput. Ind. Eng.* **2020**, *149*, 106778. [CrossRef]
- 42. Li, J.q.; Deng, J.w.; Li, C.y.; Han, Y.y.; Tian, J.; Zhang, B.; Wang, C.g. An improved Jaya algorithm for solving the flexible job shop scheduling problem with transportation and setup times. *Knowl.-Based Syst.* **2020**, 200, 106032. [CrossRef]
- 43. Caldeira, R.H.; Gnanavelbabu, A. A Pareto based discrete Jaya algorithm for multi-objective flexible job shop scheduling problem. *Expert Syst. Appl.* **2021**, 170, 114567. [CrossRef]
- 44. Wang, Z.Y.; Lu, C. An integrated job shop scheduling and assembly sequence planning approach for discrete manufacturing. *J. Manuf. Syst.* **2021**, *61*, 27–44. [CrossRef]
- 45. Wang, F.; Zhang, H.; Zhou, A. A particle swarm optimization algorithm for mixed-variable optimization problems. *Swarm Evol. Comput.* **2021**, *60*, 100808. [CrossRef]
- 46. Feng, W.A.; Yl, A.; Fl, A.; Hy, B. An ensemble learning based prediction strategy for dynamic multi-objective optimization-ScienceDirect. *Appl. Soft Comput.* 2020, *96*, 106592.
- 47. Gong, G.; Deng, Q.; Gong, X.; Huang, D. A non-dominated ensemble fitness ranking algorithm for multi-objective flexible job-shop scheduling problem considering worker flexibility and green factors. *Knowl.-Based Syst.* **2021**, 231, 107430. [CrossRef]
- 48. Deb, K.; Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evol. Comput.* **2013**, *18*, 577–601. [CrossRef]
- 49. Zhang, G.; Gao, L.; Shi, Y. An effective genetic algorithm for the flexible job-shop scheduling problem. *Expert Syst. Appl.* **2011**, *38*, 3563–3573. [CrossRef]
- 50. Ishikawa, S.; Kubota, R.; Horio, K. Effective hierarchical optimization by a hierarchical multi-space competitive genetic algorithm for the flexible job-shop scheduling problem. *Expert Syst. Appl.* **2015**, *42*, 9434–9440. [CrossRef]
- Zhang, X.; Zhan, Z.H.; Zhang, J. Adaptive Population Differential Evolution with Dual Control Strategy for Large-Scale Global optimization Problems. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; pp. 1–7.
- 52. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
- Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* 2007, 11, 712–731. [CrossRef]
- 54. Wang, G.; Gao, L.; Li, X.; Li, P.; Tasgetiren, M.F. Energy-efficient distributed permutation flow shop scheduling problem using a multi-objective whale swarm algorithm. *Swarm Evol. Comput.* **2020**, *57*, 100716. [CrossRef]
- 55. Han, Y.; Li, J.; Sang, H.; Liu, Y.; Pan, Q. Discrete evolutionary multi-objective optimization for energy-efficient blocking flow shop scheduling with setup time. *Appl. Soft Comput.* **2020**, *93*, 106343. [CrossRef]