Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 3974024, 22 pages
https://doi.org/10.1155/2020/3974024

Research Article

Research on Two-Stage Joint Optimization Problem of Green Manufacturing and Maintenance for Semiconductor Wafer

Jun Dong[1,2] **and Chunming Ye**[1]

[1]*Business School, University of Shanghai for Science and Technology, Shanghai 200093, China*
[2]*Henan Institute of Technology, Xinxiang, Henan 453000, China*

Correspondence should be addressed to Chunming Ye; yechm@usst.edu.cn

Received 25 October 2019; Revised 17 December 2019; Accepted 31 December 2019; Published 24 January 2020

Academic Editor: Francisco R. Villatoro

This paper proposes a two-stage joint optimization problem of green manufacturing and maintenance for semiconductor wafer (TSGMM-SW) considering manufacturing stage, inspection, and repair stage simultaneously, which is a typical NP-hard problem with practical research significance and value. Aiming at this problem, a green scheduling model with the objective of minimizing makespan, total carbon emissions, and total preventive maintenance (PM) costs is constructed, and an improved hybrid multiobjective multiverse optimization (IHMMVO) algorithm is proposed in this paper. The joint optimization of green manufacturing and maintenance is realized by designing synchronous scheduling and maintenance strategy for wafer manufacturing and equipment PM. The diversity of the population is expanded and the optimization performance of IHMMVO is improved by designing the initial population fusion strategy and subpopulation evolution strategy. In the experimental phase, we perform the simulation experiments of 900 test cases randomly generated from 90 parameter combinations. The IHMMVO algorithm is compared with other existing algorithms to verify the effectiveness and feasibility for TSGMM-SW.

1. Introduction

The concept of industry 4.0 refers to the fourth industrial revolution to realize intelligent manufacturing through the Internet of things, big data analysis, and other technical means, which is led by intelligent factories, intelligent production, and intelligent logistics. It is bound to have new higher requirements for the high-tech electronic information industry. As a pillar industry of the national economy, semiconductor industry has a very important strategic position. Wafer fabrication is one of the most complex processes in semiconductor manufacturing system. It needs to form the required circuit layer on wafer surface through various physical and chemical operations such as oxidation, deposition, injection, sputtering, photolithography, and cleaning. Its essence is reentry. Wafers may repeatedly visit some equipment at different processing stations. The manufacturing process can be viewed as the reentrant hybrid flow shop scheduling (RHFS) problem. The general job shop scheduling problem (JSP) is that there are multiple machines with different functions in the processing system. The jobs to be processed include multiple operations. Each operation is processed on one machine, and the processing routes of jobs are different from each other. There are sequence constraints between operations belonging to the same job. RHFS problem is that there are multiple processing stations in the processing system. All jobs have the same process constraints among each station and need to reentry the stations several times in the same order. At least one station has more than one machine. Jobs can be processed by any machine on the corresponding station. The RHFS problem is large in scale and high in complexity, and it belongs to the NP-hard problem [1]. Therefore, there is practical value and significance of designing a new intelligent optimization algorithm to solve it.

The development of global economy has also brought many environmental problems, among which greenhouse gas emissions, especially carbon dioxide emissions, have attracted the attention of scholars at home and abroad. Manufacturing industry is one of the main sources of carbon dioxide emissions. It is imperative to achieve energy saving and emission reduction in manufacturing industry and develop sustainable manufacturing. Once the semiconductor wafer manufacturing factories are established, they run 24 hours a day and 365 days a year. Machines will produce a degradation effect due to long-term continuous production, which will prolong processing time of jobs and increase machine energy consumption, resulting in a corresponding increase in carbon emissions. In addition, the equipment of semiconductor wafer manufacturing factories is generally expensive. If there is a breakdown for maintenance, it is difficult to find alternative processing equipment quickly, which will delay the production process of whole semiconductor production line and cause enormous economic losses for enterprises. Therefore, it is necessary to perform PM on the equipment of semiconductor wafer manufacturing factories. After PM, machines can restore its processing ability to a certain extent, reduce the frequency of random breakdown, shorten processing time, and reduce energy consumption and carbon emissions of machines so as to make some contribution to achieve green production.

Inspired by the idea of multiverse theory, Mirjalili et al. [2] designed a new heuristic algorithm, multiverse optimizer (MVO) algorithm. The algorithm involves fewer parameters, simpler structure, and stronger search ability. It has been applied in parameter optimization [3], image processing [4], and flexible job shop scheduling [5]. However, its application in RHFS problem is few. This paper carries out the research on TSGMM-SW, which only considers PM in the manufacturing stage. The contributions of this paper are as follows:

(1) The TSGMM-SW model is proposed for the first time, which is more in line with the actual situation of semiconductor wafer manufacturing line and can achieve the balance among makespan, total carbon emissions, and total PM costs.

(2) IHMMVO algorithm was designed for the first time to solve the problem, which is a combination of NSGA-II and MVO algorithm. The improved operations of the algorithm mainly include synchronous scheduling and maintenance strategy, initial population fusion strategy, subpopulation evolution strategy, etc.

(3) A large number of comparison experiments were performed on different scale test cases by IHMMVO and other existing algorithms, respectively. The effectiveness and feasibility of IHMMVO were verified.

The remainder of this paper is organized as follows. The related research literature is introduced in Section 2. The description and model of TSGMM-SW are introduced in Section 3. Section 4 states the IHMMVO algorithm in detail. Section 5 presents experimental results and analysis. Section 6 is a discussion to explain the main limitations and implications of the study. Section 7 provides the conclusions and future work directions.

2. Literature Review

In recent years, experts and scholars at home and abroad are also continually conducting related discussion and research in the field of semiconductor manufacturing. Kim and Lee [6] used the input, output, and production quantity as the scheduling coordination factors to update manufacturing lead time and available WIP (work in process) level, which realized the synchronization of semiconductor production plan and scheduling decision. Bitar et al. [7] started the research on unrelated parallel machine resource scheduling problem in semiconductor lithography workshop. Jamrus et al. [8] solved the flexible job shop scheduling problem in semiconductor manufacturing under uncertain processing time, with the objective of optimizing makespan. Jia et al. [9] addressed the problem on reentrant batch-processing machines with incompatible job families in the semiconductor wafer fabrication system. Hur et al. [10] studied the machine setting, batch allocation, and batch sequence problems in the semiconductor manufacturing system, which had certain practical significance. Foumani et al. [11–14] started the research on scheduling, production and inspection of robotic cells in the flow shop environment. It is well known that robotic cells are specifically designed for semiconductor wafer fabrication. Tirkel [15] proposed an approach for establishing a quantitative model to design the efficiency of in-line inspection in semiconductor wafer. Ulrich et al. [16] presented an automated and cycle time optimized path planning algorithm for robot-based inspection systems. To the best of our knowledge, most of the research on the semiconductor manufacturing is limited to the study of its manufacturing stage in the current literature, ignoring its inspection and repair stage, which has a certain degree of limitations. In the actual wafer processing process, wafers are manufactured layer by layer. The defects caused by a manufacturing layer will be covered by a new layer because they are not inspected and repaired in time [17]. As a result, the final wafer testing phase cannot track the error link, so the wafer cannot be repaired in an effective time, which increases the proportion of defective products. The specific approaches and optimization objectives of the above typical literature are shown in Table 1.

In recent years, heuristic algorithms have played a greater advantage in solving RHFS combinatorial optimization problems and have shown strong competitiveness. Mousavi et al. [18] studied a biobjective hybrid flow shop scheduling problem, in which they considered reentrant lines, setup times, and position-dependent learning effects. Shen et al. [19] and Ying et al. [20] solved the reentrant hybrid flow shop scheduling problem with the makespan and the total tardiness criteria. Zhang and Chen [21] addressed a production scheduling problem originating from a real rotor workshop. Given its specific characteristics, the problem was formulated as a reentrant hybrid flow shop scheduling problem with machine eligibility constraints. Sangsawang et al. [22] designed a two-stage RHFS problem with blocking constraints, aiming at minimizing makespan. Chamnanlor et al. [23] studied the RHFS problem with time window constraints, which often occurred in the

Table 1: Typical literature about semiconductor manufacturing research.

Literature	Approaches	Objectives
Kim and Lee [6]	Iterative algorithm	Realize the synchronization of production plan and scheduling decision
Bitar et al. [7]	Memetic algorithm	Minimize weighted flow time, maximize the number of processed products
Jamrus et al. [8]	Hybrid algorithm combining particle swarm optimization with Cauchy distribution and genetic algorithm	Minimize makespan
Jia et al. [9]	Improved combined scheduling algorithm	Minimize total weight tardiness
Hur et al. [10]	Greedy random adaptive search algorithm	Maximize weighted throughput during the 2–5 day planning period
Foumani et al. [11–14]	Theoretical analysis, theorem proving and heuristic method	Optimize the scheduling, production and inspection problems of robotic cells in flow shop environment
Tirkel [15]	Quantitative model and sensitivity analysis	Modelling the efficiency of in-line inspection
Ulrich et al. [16]	Probabilistic roadmap method applied on all measurement poses and A^* search algorithm	Finding the optimal path between measurement poses

manufacturing system producing the slider parts of hard disk drives. The above literature provides a good reference for the research of RHFS problem, but most of them involve only one stage of RHFS problem. The specific approaches and optimization objectives of the above typical literature are shown in Table 2.

There have been many studies on green manufacturing. Yin et al. [24] established a low-carbon mathematical scheduling model with machine speed as an independent decision variable and designed the corresponding encoding and decoding method and crossover mutation operator. The validity of the model and method for low-carbon job shop scheduling problem was illustrated by an engineering example. Liu et al. [25] proposed a method for calculating carbon footprint of products and a multiobjective model for minimizing production cycle and carbon footprint, which can effectively provide carbon labels for future life cycle products of manufacturing enterprises. Sihag and Sangwan [26] established a multiobjective mathematical model for optimizing carbon emissions and processing time in turneries, which could select the best cutting parameters for manufacturing enterprises, reduce carbon emissions, and improve processing efficiency. For single-machine scheduling and multivehicle routing problems, Wang et al. [27] designed a mathematical programming model with the objective of minimizing total carbon emissions. Experiments on an enterprise case and 20 simulation cases verified the effectiveness of the proposed algorithm in guiding green manufacturing. Foumani and Smith-Miles [28] put forward a mixed integer linear model aiming at optimizing production time and carbon emissions on the basis of considering three common carbon emission reduction policies of flow shop. He considered the impact of carbon emissions reduction policy on the economic competitiveness of manufacturing industry from the perspective of environmental policymakers. Wang and Feng [29] constructed the green degree evaluation system of equipment manufacturing industry and also proposed policy recommendations for the green degree of China's equipment manufacturing industry from the perspectives of improving relevant policies, technological upgrading and environmental protection input, applying legal policy systems, and increasing financial support. Zhang et al. [30] provided a general model and overall framework of green manufacturing for industrial enterprises. According to the needs and conditions, enterprises could construct and formulate a clear plan for the implementation of green manufacturing that conforms to their actual situation of enterprises. The specific approaches and optimization objectives of the above typical literature are shown in Table 3.

In a literature about PM research, Liu et al. [31] proposed a comprehensive decision-making model, which considered the coordination of preventive maintenance and single machine scheduling. The problem of maintenance shortage or overage caused by only considering PM was avoided. Huang et al. [32] designed cluster particle swarm optimization algorithm to shorten the manufacturing cycle and effectively reduce energy consumption aiming at the problem of cleaner production by adding deteriorating maintenance operations to flow shop with two-stage multiprocessors. Ladj et al. [33] addressed the PM problem of a single multifunctional machine based on health management. Xiao et al. [34] built a joint optimization model with the objective of minimizing the total costs, including production costs, PM costs, maintenance costs of unexpected breakdown, and delay costs. Optimal PM interval and allocation strategy for each job can be obtained by solving this problem. Golpîra and Tirkolaee [35] introduced an original concept entitled "stable maintenance tasks scheduling," which incorporated robust optimization concept into the maintenance and repair tasks scheduling problem. Goli et al. [36] addressed a novel robust flow shop scheduling problem with outsourcing option where jobs could be either scheduled for inside or outsourced to one of the available subcontractors. Capacity limitation for inside resource, just-in-time delivery policy, and uncertain processing time were the key assumptions of the proposed model. The specific approaches and optimization objectives of the above typical literature are shown in Table 4.

In summary, the current literature rarely uses MVO algorithm to solve the scheduling optimization integration problem, and it also rarely optimized makespan, total PM costs, and total carbon emissions simultaneously. In addition,

Table 2: Typical literature about RHFS research.

Literature	Approaches	Objectives
Mousavi et al. [18]	Improved genetic algorithm	Minimize makespan and tardiness
Shen et al. [19]	Improved teaching-learning-based algorithm	Minimize makespan and tardiness
Ying et al. [20]	Iterative greedy algorithm	Minimize makespan and tardiness
Zhang and Chen [21]	Improved discrete differential evolution algorithm	Minimize tardiness
Sangsawang et al. [22]	Combining adaptive genetic algorithm based on fuzzy logic controller with particle swarm optimization based on Cauchy distribution	Minimize makespan
Chamnanlor et al. [23]	Hybrid optimization algorithm based on genetic algorithm and ant colony algorithm	Minimize makespan

Table 3: Typical literature about green manufacturing research.

Literature	Approaches	Objectives
Yin et al. [24]	Multiobjective genetic algorithm based on simplex lattice	Optimize productivity, energy efficiency, and noise reduction
Liu et al. [25]	Improved fruit fly algorithm	Minimize makespan and energy consumption
Sihag and Sangwan [26]	Multiobjective genetic algorithm	Minimize makespan and carbon emissions
Wang et al. [27]	Hybrid tabu search algorithm	Minimize carbon emissions
Foumani and Smith-Miles [28]	CPLEX	Minimize makespan and carbon emissions
Wang and Feng [29]	Improved particle swarm algorithm	Measure the green degree
Zhang et al. [30]	System engineering method	Improve the green rate of manufacturing technology and manufacturing process and reduce the environmental impact rate of resources

Table 4: Typical literature about PM research.

Literature	Approaches	Objectives
Liu et al. [31]	Genetic algorithm	Minimize the production costs and tardiness costs
Huang and Yu [32]	Cluster particle swarm algorithm	Minimize makespan
Ladj et al. [33]	Improved genetic algorithm	Minimize the total interventions costs
Xiao et al. [34]	Stochastic key genetic algorithm	Minimize the total costs
Golpîra and Tirkolaee [35]	CPLEX	Minimize the total costs of the maintenance and maximize the stability/robustness of the schedule
Goli et al. [36]	CPLEX	Minimize the total-weighted time and the total costs of outsourcing

most of them study the RHFS problem about one stage. The research of this paper fills the gap. The TSGMM-SW model established in this paper studies the two-stage reentrant hybrid flow shop scheduling problem in wafer manufacturing process. That is, when wafers are processed one layer through all stations in manufacturing stage, they enter into the inspection and repair stage immediately. The wafers undergo inspection and repair in time to improve the qualification rate. In addition, the model can achieve the balance among makespan, total PM costs, and total carbon emissions.

3. Problem Description and Model Establishment

3.1. Description of TSGMM-SW. TSGMM-SW consists of two stages: the first stage is the manufacturing stage of semiconductor wafers and the second stage is the inspection and repair stage, in which each stage is an independent RHFS problem. The problem can be described as that n jobs need to be processed. job i Among them, needs to processed L_i layers (re-entry number). Besides, the re-entry number of each job can be different. There are s stations in the first stage, and station l has $m_l\,(m_l \geq 1)$ parallel machines that can be chosen. Besides, at least one station has more than one parallel machine. In the second stage, there are two stations, one is the inspection station and the other is the repair station, and each job needs to reenter L_i' times. All of the jobs pass through each station in the same sequence of processing constraints, and each job is processed $L_i\,(s + 2L_i')$ times in total [17]. It is assumed that there is infinite capacity of buffer between continuous processing stations. The processing process is uninterrupted. Each wafer can restart processing

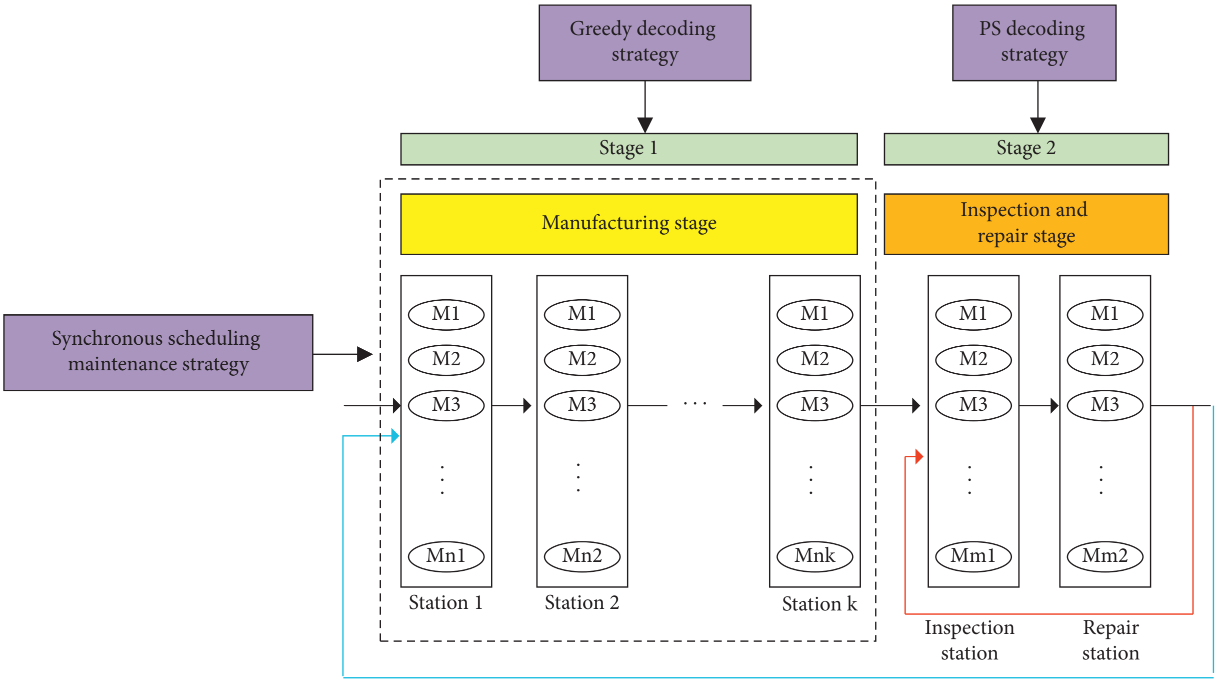


Figure 1: Schematic diagram of TSGMM-SW problem.

the next layer after the inspection and repair stage. The PM fixed costs are same for all machines in each station. Figure 1 shows the schematic diagram of TSGMM-SW problem.

Semiconductor wafer fabrication factories cannot keep machines in working state all the time; jobs to be processed have to wait for the process of the previous operation to be completed or the selected machine to be idle. The longer the waiting time is, the greater the change of the production environment will occur. The machine will produce degradation effect, and the actual processing time of jobs will gradually increase. The actual processing time of machines is calculated by the following equation:

$$T_{ijlk} = \overline{T}_{ijlk} \times t_{\omega}^{\tau}, \tag{1}$$

where T_{ijlk} is the actual processing time of o_{ij}, $\overline{T}_{ijlk}$ is the standard processing time of o_{ij}, t_{ω} is the waiting time before processing o_{ij}, and τ is degradation factor, assuming that the degradation factor of machines with the same serial number at each station is the same. In practical production applications, the two-parameter Weibull distribution tends to describe the failure rule of machines [37]. The failure rate function of machines is,

$$p(t) = \frac{m}{\beta}\left(\frac{t}{\beta}\right)^{m-1} \exp\left(-\left(\frac{t}{\beta}\right)^{m}\right), \tag{2}$$

where $m > 0$ and $\beta > 0$, m denotes shape parameters, and β denotes scale parameter.

The reliability R of machines at runtime t is shown in the following equation [38]:

$$R = \exp\left(-\left(\frac{z}{\beta}\right)\right)^{m}, \tag{3}$$

where Z denotes the service age.

According to equation (4), the service age when R reaches the threshold value is calculated.

$$z_0 = \beta\left(-InR_0\right)^{1/m}, \tag{4}$$

where R_0 is the minimum reliability threshold value of machines. During the processing, as long as service age of machines is not less than Z_0, the machine needs to perform PM at any time.

After PM, the machines cannot completely restore to their original state. The initial service age of machines after PM is shown in the following equation:

$$Z_k = Z'\left(1 - p_k\right), \tag{5}$$

where Z_k is the service age of machine k after PM, Z' is actual service age of machine k before PM, and pk is service age regression factor of machine k.

3.2. Model of TSGMM-SW

3.2.1. Model Description. TSGMM-SW studied in this paper aims at minimizing makespan ($C_{\max}$), total carbon emissions (TCT), and total PM costs (TCost). ($C_{\max}$) is the completion time of the last job in the inspection and repair stage. TCT is the sum of carbon emissions generated from the manufacturing stage and the inspection and repair stage. This paper only considers PM in manufacturing stage, so TCost is the sum of PM costs in manufacturing stage. Among them, TCT is the green indicator. The consumption of coolant, the loss of raw materials, and the carbon emissions from the power consumption of equipment switches are not related to workshop scheduling. So, this paper only considers the carbon emissions generated from power consumption during processing and idling of equipment. Also, the carbon emissions generated from lubricants used by equipment are considered too. The relevant parameters in TSGMM-SW model are shown in Table 5.

The objective functions and constraints are shown as below:

$$f1 = \min\left(C_{\max}\right), \tag{6}$$

$$f2 = \min\left(\mathrm{TCT}\right), \tag{7}$$

$$f3 = \min\left(\mathrm{TCost}\right), \tag{8}$$

where

$$C_{\max} \geq C_i, \quad i = 1, 2, \ldots, n, \tag{9}$$

$$\begin{aligned}\mathrm{TCT} = {} & \mathrm{TCPM} + \mathrm{TCPR} + \mathrm{TCIM} + \mathrm{TCIR} \\ & + \mathrm{TCRM} + \mathrm{TCRR},\end{aligned} \tag{10}$$

$$\mathrm{TCost} = \sum_{l=1}^{s}\sum_{k=1}^{m_l}\left(\mathrm{GCost}_{lk} + \mathrm{tp}_{lk} \times \mathrm{VCost}_{lk}\right), \tag{11}$$

$$\mathrm{TCPM} = \sum_{l=1}^{s}\sum_{k=1}^{m_l}\sum_{i=1}^{n}\sum_{j=1}^{n_i-2L_i'} r_{ijlk}F_e\mathrm{PW}_{lk}T_{ijlk}, \tag{12}$$

$$\mathrm{TCPR} = \sum_{l'=1}^{s'}\sum_{k'=1}^{m_i'}\sum_{i=1}^{n}\sum_{j=1}^{2L_i'} r_{ijl'k'}F_e\mathrm{PW}_{l'k'}T_{ijl'k'}, \tag{13}$$

$$\mathrm{TCIM} = \sum_{l=1}^{s}\sum_{k=1}^{m_l} F_e\mathrm{PI}_{lk}\mathrm{TI}_{lk}, \tag{14}$$

$$\mathrm{TCIR} = \sum_{l'=1}^{s'}\sum_{k'=1}^{m_l'} F_e\mathrm{PI}_{l'k'}\mathrm{TI}_{l'k'}, \tag{15}$$

$$\mathrm{TCRM} = \sum_{l=1}^{s}\sum_{k=1}^{m_l} \frac{\mathrm{tclose}_{l,k} - \mathrm{topen}_{l,k}}{\mathrm{TR}_{lk}}\mathrm{UR}_{lk}F_R, \tag{16}$$

s.t.

$$\mathrm{TCRR} = \sum_{l'=1}^{s'}\sum_{k'=1}^{m_l'} \frac{\mathrm{tclose}_{l',k'} - \mathrm{topen}_{l',k'}}{\mathrm{TR}_{l'k'}}\mathrm{UR}_{l'k'}F_R, \tag{17}$$

$$r_{ijlk}\left(S_{ij} + T_{ijlk}\right) \leq r_{i,j+1,l^*,k^*}S_{i,j+1}, \quad \forall i, j, l, l^* \text{ and } l \neq l^*, \tag{18}$$

$$\sum_{k=1}^{m_l} r_{ijlk} = 1, \quad \forall i, j, l, \tag{19}$$

$$\begin{aligned}& G \times \left(2 - r_{ijlk} - r_{ghlk}\right) + G \times \left(1 - Z_{ij,gh,lk}\right) + \left(S_{gh} - S_{ij}\right) \\ & \quad \geq T_{ijlk}, \quad \forall i, g, j, l, h, k, i \leq g,\ O_{ij} \in U_l, O_{gh} \in U_l,\end{aligned} \tag{20}$$

$$\begin{aligned}& G \times \left(2 - r_{ijlk} - r_{ij^*l^*k^*}\right) + G \times \left(1 - Z_{ij,ij^*,l^*k^*}\right) + \left(S_{ij^*} - S_{ij}\right) \\ & \quad \geq T_{ijlk}, \quad \forall i, j, j^*, l, l^*, k, k^*, j < j^*, O_{ij} \in U_l,\ O_{ij^*} \in U_l.\end{aligned} \tag{21}$$

Equations (6)–(8) are three objective functions. Equation (9) is the definition of makespan. Equation (10) is the definition of the total carbon emissions. Equation (11) is the definition of the total PM costs. Equations (12) and (13) indicate the total carbon emissions generated from machines processing in manufacturing stage and in inspection and repair stage, respectively. Similarly, equations (14) and (15) indicate the total carbon emissions generated from machines in idle state in manufacturing stage and in inspection and repair stage, respectively. Equations (16) and (17) indicate the total carbon emissions generated from the use of lubricants in manufacturing stage and in inspection and repair stage, respectively. Equation (18) ensures the completion time of O_{ij} must be earlier than the start time of O_{ij+1}. Equation (19) ensures any operation can be processed on only one machine in the corresponding station. Equations (20) and (21) ensure each machine in any station can only process at most one operation at the same time. Equations (18)–(21) are the constraints in the manufacturing stage, and they are similar to those in the inspection and repair stage; the value range of stations is [1, 2].

3.2.2. Model Complexity Analysis. Under the premise that the model structure is determined, the computational complexity is related to the number of decision variables and constraints. The TSGMM-SW model involves two stages, namely, the manufacturing stage and the inspection and repair stage. The number of its decision variables is $\sum_{l=1}^{s} m_l \left(\sum_{i=1}^{n} L_i + 1\right) + \sum_{l'=1}^{2} m_{l'} \left(\sum_{i=1}^{n} L_i' + 1\right)$. The number of constraints in the manufacturing stage is $ns^2 \left(\sum_{i=1}^{n} n_i + \sum_{i=1}^{n} (n_i/2)(n_i - 1)\left(\sum_{l=1}^{s} m_l\right)^2\right) + ns\sum_{i=1}^{n} n_i \left(1 + ((n-1)/2)\sum_{g>i}^{n} n_g\right)$. The number of constraints in the inspection and repair stage is $4n \left(\sum_{i=1}^{n} n_i + \sum_{i=1}^{n} (n_i/2)(n_i - 1)\left(\sum_{l=1}^{s} m_{l'}\right)^2\right) + 2n\sum_{i=1}^{n} n_i \left(1 + ((n-1)/2)\ \sum_{g>i}^{n} n_g\right) + n$. From the number of variables and constraints above, we can see that the TSGMM-SW model solves very complex intractable problem.

4. Proposed IHMMVO Algorithm

4.1. Basic MVO Algorithm [2]. MVO finds the solutions through the interaction between white holes, black holes, and wormholes. Each universe represents a viable solution to the problem. Each object in the universe represents the variables in each solution, and the expansion rate of universes represents the fitness value of the solution. Universes with higher expansion rates are considered to have white holes, and those with lower expansion rates are

Table 5: Mathematical notations and meanings.

Notations	Meanings
n	Total number of jobs
n_i	Total number of operations contained in job i
s	Number of stations in manufacturing stage
l	Serial number of stations in manufacturing stage, $l = 1, 2, \ldots, s$
m_l	Number of parallel machines at station l in manufacturing stage
k	Serial number of machines at station l, $k = 1, 2, \ldots, m_l$, $k = 1, 2, \ldots, m_l$
L_i	Reentry number of job i in manufacturing stage
s'	Number of stations in inspection and repair stage, $s' = 2$
l'	Serial number of stations in inspection and repair stage, $l' = 1, 2$
$m_{l'}$	Number of parallel machines at station l' in inspection and repair stage
k'	Serial number of machines at station l', $k' = 1, 2, \ldots, m_{l'}$
L_i	Reentry number of job i in inspection and repair stage
o_{ij}	The j-th operation of job i
T_{ijlk}	Processing time of o_{ij} on machine k at station l
$T_{ijl'k'}$	Inspection and repair time of o_{ij} on machine k' at station l'
G	A large positive number
PW_{lk}	Processing power of machine k at station l
PI_{lk}	Idle power of machine k at station l
$\mathrm{TI}_{\mathrm{lk}}$	Idle time of machine k at station k
$\mathrm{PW}_{l'k'}$	Processing power of machine k' at station l'
$\mathrm{PI}_{l'k'}$	Idle power of machine k' at station l'
$\mathrm{TI}_{l'k'}$	Idle time of machine k' at stage l'
TCPM	Total carbon emissions from machines processing in manufacturing stage
TCPR	Total carbon emissions from machines processing in inspection and repair stage
TCIM	Total carbon emissions from machines in idle state in manufacturing stage
TCIR	Total carbon emissions from machines in idle state in inspection and repair stage
TCRM	Total carbon emissions from lubricants used by machines in manufacturing stage
TCRR	Total carbon emissions from lubricants used by machines in inspection and repair stage
F_e	Carbon emission factor of electric energy
$\mathrm{tclose}_{l,k}$	The finishing time of machine k at station l (the finishing time of the last operation on machine k)
$\mathrm{topen}_{l,k}$	The start time of machine k at station l (the start time of the first operation on machine k)
$\mathrm{tclose}_{l',k'}$	The finishing time of machine k' at station l' (the finishing time of the last operation on machine k')
$\mathrm{topen}_{l',k'}$	The start time of machine k' at station l' (the start time of the first operation on machine k')
UR_{lk}	The usage amount of lubricants on machine k at station l
$\mathrm{UR}_{l'k'}$	The usage amount of lubricants on machine k' at station l'
TR_{lk}	Effective service time of lubricants on machine k at station l
$\mathrm{TR}_{l'k'}$	Effective service time of lubricants on machine k' at station l'
F_R	Carbon emissions factor of lubricants
C_i	The start time of $O_{i,j}$
U_l	The set of all operations processed at station l; due to the reentrant characteristics, the same job may appear multiple times in it
C_i	The completion time of job i
VCost_{lk}	PM unit activity costs of machine k at station l
GCost_{lk}	PM fixed costs of machine k at station l
tp_{lk}	PM time of machine k at station l
$\mathrm{Idletime}_{lk}$	The idle time of machine k at station l before the next operation starts processing
r_{ijlk}	1 indicates O_{ij} is processed on machine k at stage l; 0 indicates O_{ij} is not processed on machine k at stage l
$r_{ijl'k'}$	1 indicates O_{ij} is inspected or repaired on machine k' at stage l'; 0 indicates O_{ij} is not inspected and repaired on machine k' at stage l'
$Z_{ij,gh,lk}$	1 indicates O_{ij} is processed before O_{gh} on machine k at stage l; 0 indicates O_{ij} is not processed before O_{gh}

considered to have black holes. Particles in the universe move from white holes (sending) to black holes (accepting) through a medium called wormholes. Eventually, all matter in the universe reaches its optimum position by exploring the search space. In each iteration of the algorithm, the roulette method is used to select a white hole from universe individuals which are sorted by expansion rate. The population is described as

$$U = \begin{bmatrix} x_1^1 \, x_1^2 \ldots x_1^d \\ x_2^1 x_2^2 \ldots x_2^d \\ \ldots \\ x_n^1 x_n^2 \ldots x_n^d \end{bmatrix}, \tag{22}$$

where d is the number of variables, n is the number of universes (the number of candidate solutions), and x_i^j is the j-th dimension of the i-th universe, which is described as

$$x_i^j = \begin{cases} x_k^j, & r_1 < \mathrm{NI}(U_i), \\ x_i^j, & r_1 \geq \mathrm{NI}(U_i), \end{cases} \tag{23}$$

where U_i represents the i-th universe, $\mathrm{NI}(U_i)$ represents normalized expansion rate of the i-th universe, x_k^j represents the j-th dimension of the k-th universe selected by the roulette method, and r_1 is a random number between [0, 1].

In order to increase the expansion rate of objects through wormholes, it is assumed that wormhole tunnels are always built between each universe and the optimal universe. The location of black holes in the optimal universe is updated by the following equation:

$$x_i^j = \begin{cases} X_j + \mathrm{TDR} \times \left(\left(ub_j - lb_j\right) \times r_4 + lb_j\right), & r_3 < 0.5,\ r_2 < \mathrm{WEP}, \\ X_j - \mathrm{TDR} \times \left(\left(ub_j - lb_j\right) \times r_4 + lb_j\right), & r_3 \geq 0.5,\ r_2 < \mathrm{WEP}, \\ x_i^j, & r_2 \geq \mathrm{WEP}, \end{cases} \tag{24}$$

where X_j represents the j-th dimension of the current optimal universe, ub_j and lb_j are the upper and lower bounds of the j-th dimension respectively, r_2, r_3, and r_4 represent random numbers between [0, 1], respectively, WEP indicates the existence rate of wormholes, and it is calculated by equation (25), and TDR indicates the travel distance ratio of the object around the current optimal universe, and it is calculated by equation (26).

$$\mathrm{WEP} = \frac{\mathrm{WEP}_{\min} + l \times \left(\mathrm{WEP}_{\max} - \mathrm{WEP}_{\min}\right)}{L}, \tag{25}$$

$$\mathrm{TDR} = 1 - \frac{l^{1/p}}{L^{1/p}}, \tag{26}$$

where $\mathrm{WEP}_{\max}$ represents the maximum value of WEP, $\mathrm{WEP}_{\min}$ represents the minimum value of WEP, l represents the current number of iterations, L represents the maximum number of iterations, and p represents mining accuracy of the algorithm.

4.2. IHMMVO Algorithm for Solving TSGMM-SW

4.2.1. IHMMVO Algorithm Flow. The IHMMVO algorithm proposed in this paper adds fast nondominated sorting, crowding distance calculation, and elite strategy to solve the multiobjective optimization problem based on the single-objective MVO algorithm. In addition, the improvements mainly include the following aspects: synchronous scheduling and maintenance strategy, initial population fusion strategy, and subpopulation evolution strategy. The flowchart is shown in Figure 2.

4.2.2. Encoding and Decoding. The basic MVO algorithm is used to solve the single-objective optimization problem of continuous function. If the individual's position corresponds to the continuous vector value, it is impossible to update the processing sequence of jobs in the job shop scheduling problem. In this paper, the random key rule based on ascending order is used to map the position of individuals and the processing order of jobs. By changing the position information of individuals, the processing order of jobs is changed, so different scheduling schemes are constructed. For the manufacturing stage of TSGMM-SW, greedy strategy is adopted to decode, i.e., arrange the processing operation of each job in turn on the machine that can start processing it at the earliest time at each station. This decoding method can quickly calculate the service age of each machine and make it easier to judge whether the machine needs PM or not. For the inspection and repair stage of TSGMM-SW problem, PM operation is not considered, and the service age of machines is not required to be calculated. The PS decoding method [20] is used. That is, if the idle time is enough to process the operation, it will be inserted into the idle time gap for processing.

There are three scheduling strategies for joint optimization of production scheduling and maintenance, namely, scheduling before maintenance [39], maintenance before scheduling [40], and synchronous scheduling and maintenance [41]. The scheduling before maintenance strategy needs to arrange the scheduling plan of jobs first and then insert the PM into the idle gap of the machine. If the idle gap cannot meet the PM time, you need to dynamically adjust the start processing time of jobs. This strategy involves rescheduling, which is more complex and time consuming. Maintenance before scheduling strategy sets the PM period to be unavailable and schedules the jobs in the remaining time. This strategy ignores the actual processing situation of machines in the factory and excessively relies on the machine PM period determined by historical experience. The synchronous scheduling and maintenance strategy makes up for the shortcomings of the above two strategies. According to the machine reliability threshold, the service age threshold of machines is determined. Every time when a job is arranged, it is judged whether the service age of machines reaches the threshold value. If it reaches the threshold value, PM will be executed. Otherwise, the jobs will continue to be arranged. The actual processing situation of machines will be fully considered, and no additional dynamic adjustment of jobs is required. Therefore, the third synchronous scheduling and maintenance strategy is adopted in this paper. The specific process is as follows.

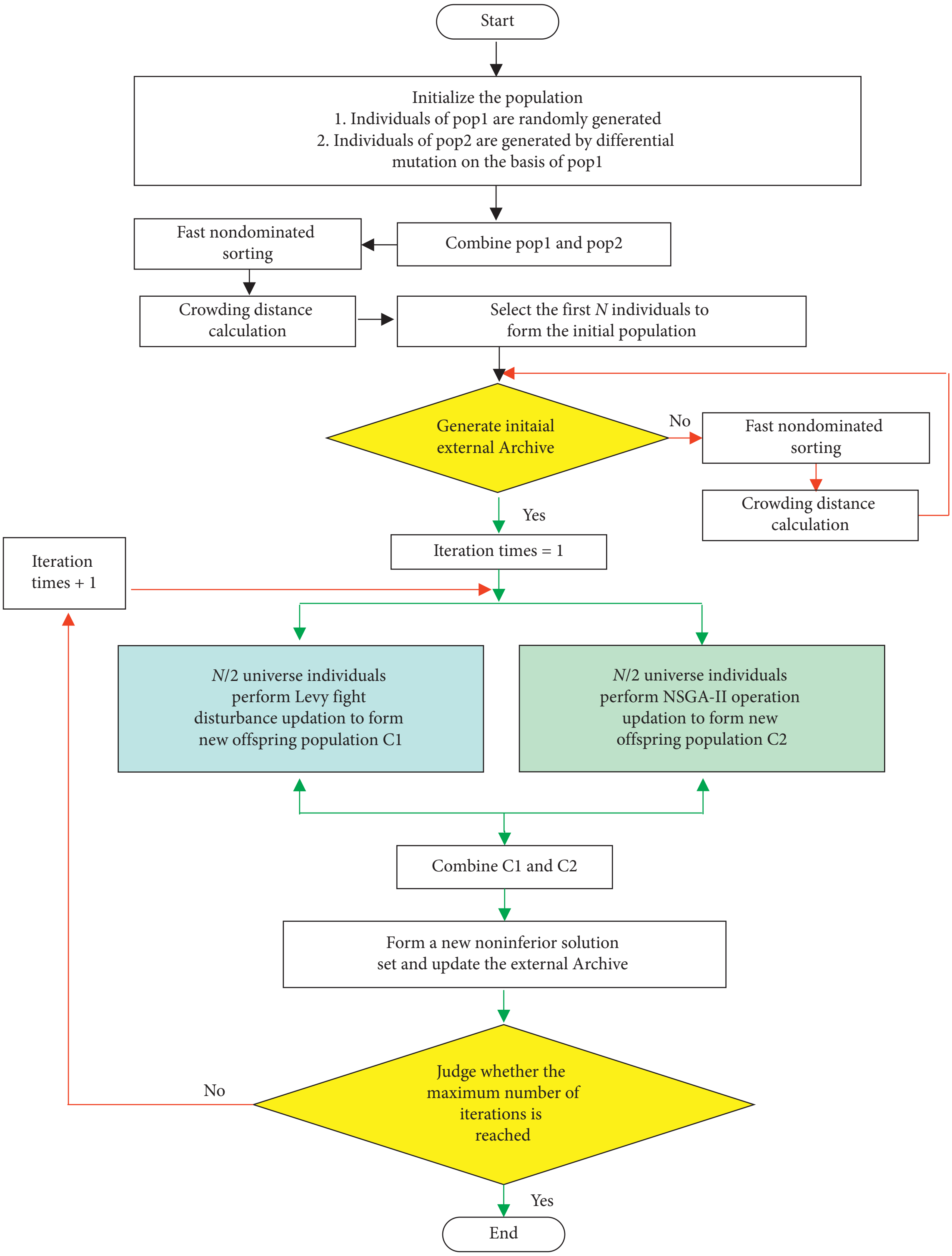

Start
Initialize the population
1. Individuals of pop1 are randomly generated
2. Individuals of pop2 are generated by differential mutation on the basis of pop1
Combine pop1 and pop2
Fast nondominated sorting
Crowding distance calculation
Select the first N individuals to form the initial population
Generate initaial external Archive
No
Fast nondominated sorting
Crowding distance calculation
Yes
Iteration times = 1
Iteration times + 1
N/2 universe individuals perform Levy fight disturbance updation to form new offspring population C1
N/2 universe individuals perform NSGA-II operation updation to form new offspring population C2
Combine C1 and C2
Form a new noninferior solution set and update the external Archive
Judge whether the maximum number of iterations is reached
No
Yes
End

Figure 2: The flowchart of IHMMVO.

Step 1. For the operation O_{ij+1} which waits for processing, the machine that can start to process it at the earliest time is selected at the corresponding station. Judge Z_t (service age of the selected machine k at the moment t); if $Z_t + T_{i,j+1,lk} < Z_0$, machine k does not need perform PM and O_{ij+1} is processed directly on the machine k. Otherwise go to Step 2.

Step 2. If $T_{i,j+1,lk} + Z_t > Z_0$ & $Z_t < Z_0$ & $\mathrm{Idletime}_{lk} < \mathrm{tp}_{lk}$, ts (the start time of PM) is the maximum of S_m (the start processing time of the selected machine k) and C_{ij} (the completion time of O_{ij}). The end time of PM is $\mathrm{te} = \mathrm{ts} + \mathrm{tp}_{lk}$. So, the start time of O_{ij+1} is $S_{ij+1} = \mathrm{te}$, and execution ends and goes to step 5. Otherwise go to Step 3.

Step 3. If $T_{i,j+1,lk} + Z_t > Z_0$ & $Z_t < Z_0$ & $\text{Idletime}_{lk} \geq \text{tp}_{lk}$, te is the maximum of S_m and C_{ij}. That is, PM is inserted into the idle time on machine k before starting O_{ij+1}, without affecting the start time S_{ij+1}. Then, ts = te − tp_{lk} and S_{ij+1} = te, and execution ends and goes to Step 5. Otherwise go to Step 4.

Step 4. If $Z_t = Z_0$, ts is the time when the machine can start processing O_{ij+1}, i.e., ts = S_m. Then, te = ts + tp_{lk}. S_{ij+1} is the maximum of te and C_{ij}, and execution ends and goes to Step 5.

Step 5. Record $S^*_{ij+1,k}$ (the earliest start time of O_{ij+1} on machine k) after considering PM on the selected machine k. Determine the earliest start time of O_{ij+1} on the remaining machines at the corresponding station in turn. First, judge whether the other machines need to perform PM if they process O_{ij+1}. If there is no need, then $S^*_{ij+1,k^*} = S_{ij+1}$. Otherwise, perform the calculation of S_{ij+1} according to Steps 1–4. Finally, the earliest start time of processing O_{ij+1} for all machines (considering PM) is obtained.

Step 6. According to the machine serial number order, select a machine that can start processing O_{ij+1} at the earliest time and assign it to process.

In order to describe synchronous scheduling and maintenance strategy more clearly, this paper gives a detailed description through a small-scale test example. The example includes four jobs; [1, 2, 4, 3] is the encoding mode, that is, decoding the job 1, 2, 4, and 3 in turn. In this example, all machines belonging to the same station in the two stages have the same processing power, idle power, effective use time, and usage of lubricants. For the manufacturing stage, there are 3 stations, and the number of parallel machines at the same speed is 3, 2, and 2, respectively. The PM fixed costs are the same for all machines belonging to the same station, but the PM unit activity costs are set differently. Reliability thresholds of the machines having the same serial number at each station are the same. There are two stations in the inspection and repair stage. One is the inspection station and the other is the repair station. There are two parallel machines at the same speed in each station. The relevant parameters and values for the two stages are shown in Tables 6 and 7. In addition, the carbon emission factor of electric energy is $1.8742 \times 10^{-7}\,\text{kgCO}_2/\text{J}$, and the carbon emission factor of lubricants is $469\,\text{kgCO}_2/\text{m}^3$ [42] ($1\,\text{L} = 0.001\,\text{m}^3$, $1\,\text{kw/h} = 3.6 \times 10^6\,\text{J}$). According to equation (4), the service age thresholds of machines for PM are [21, 28, 43] (unit: min).

Figure 3 shows the scheduling Gantt chart for this example. Take the first machine at station 1 as an example. Before processing the third layer of Job1, the first and second layers of it and the first layer of Job3 and the second layer of Job2 have already been arranged to process at station 1. The machine is idle before processing the second layer of Job1, and the idle time is 10 min, so it is necessary to consider the influence of machine's degradation effect. The machine's degradation factor is 0.1, and the actual processing time of Job1 on the first machine at station 1 is $T^2_{111} = \text{round}(2 \times 10^{0.1}) = 3$ min. Similarly, the machine is idle before processing the second layer of Job2, the idle time is 3 min, and $T^2_{211} = \text{round}(4 \times 3^{0.1}) = 4$ min. After the second layer of Job2 is processed, the machine' service age is the total processing time on the machine $Z' = 5 + 6 + 3 + 4 = 18$ min. If the third layer processing of Job1 is performed, the machine is idle before processing, and the idle time is 14 min; then, $T^3_{111} = \text{round}(4 \times 14^{0.1}) = 5$ min. The machine's service age threshold is 21 min and PM time is 4 min because $Z' + T^3_{111} = 18 + 5 = 23 > 21$ and the PM time is less than the idle time, so according to the scheduling and maintenance strategy, PM should be executed in the idle time of the machine before processing the third layer of Job1, without affecting the start time of it. The second machine and the third machine at the station 1 are also analyzed. If the third layer of Job 1 is processed on them, although PM is not required, the start times are same with the first machine. Then, when decoding, according to machine serial order, the final decision is to process the job on the first machine. The first machine of station 2 and station 3 is similar to the above example; due to limited space, there are no more details. On the second machine of station 3, the processing time of Job2 and Job3 is 13 min and 3 min, respectively, for the first layer. The machine is idle before processing the second layer of Job4, the idle time is 16 min, and the machine's degradation factor is 0.15. Then, the actual processing time is $T^2_{432} = \text{round}(5 \times 16^{0.15}) = 8$ min. Also, the machine is idle before processing the third layer of Job2, the idle time is 11 min, and the actual processing time is $T^3_{232} = \text{round}(3 \times 11^{0.15}) = 4$ min. At this moment, the machine's service age is $Z' = 13 + 3 + 8 + 4 = 28$ min. Because the second machine's service age is 28 min and $Z' = Z_0$, PM should be executed immediately after Job2 is processed. Finally, three objective values were obtained as follows: $C_{\max} = 77$ min, TCT = $0.3538\,\text{kgCO}_2$, and TCost = 143 RMB.

4.2.3. Initialization Population Fusion Strategy.

For the bionic intelligent optimization algorithm, the initial population has a very important impact on the search efficiency and the quality of the solutions. In order to expand the diversity of the population, a better initial population is screened by fusing the two populations. The two populations have the same size. The individual's position variables of population pop1 are generated randomly, and the individual's position variables of population pop2 are generated by differential evolution mutation operation based on the position information of pop1 individuals. Three individuals are randomly selected from pop1 individuals and mutated to form position variables of pop2 individuals. The mutation method is shown in the following equation:

$$\text{pop2}(i,:) = \text{pop1}(j,:) + \text{rand} \times (\text{pop1}(k,:) - \text{pop1}(p,:)), \tag{27}$$

where pop2$(i,:)$ is the position variable of the i-th individual in population pop2 and pop1$(j,:)$, pop1$(k,:)$, and pop1$(p,:)$ are the position variables of the j-th, k-th, and p-th individuals which are randomly selected from pop1. In addition, $i \neq j, k, p$. By enlarging the position variables of

Table 6: Related parameters and values.

Parameter	Value
Manufacturing stage	
Processing power of machines at each station	[6, 5, 4] (kw/h)
Effective use time of lubricants on machines at each station	[4.6, 4.3, 4.5] (h)
PM time of three machines at the first station	[4, 4, 3] (min)
PM time of two machines at the third station	[2, 5] (min)
PM unit activity costs of two machines at the second station	[3, 5] (RMB)
PM fixed costs of all machines at each station	[10, 12, 11] (RMB)
Degradation factor of the first machine at each station	0.1
Degradation factor of the third machine at each station	0.2
m in failure rate function	1.1
Idle power of machines at each station	[2, 1.5, 1] (kw/h)
The amount of lubricants used on machines at each station	[0.31, 0.32, 0.33] (L)
PM time of two machines at the second station	[2, 2] (min)
PM unit activity costs of three machines at the first station	[5, 3, 4] (RMB)
PM unit activity costs of two machines at the third station	[5, 5] (RMB)
Reliability thresholds of all machines at each station	[0.85, 0.8, 0.7]
Degradation factor of the second machine at each station	0.15
Regression factor of service age	0.9
β in failure rate function	110
Inspection and repair stage	
Processing power of machines at each station	[3, 3] (kw/h)
Effective use time of lubricants on machines at each station	[4.1, 4.2] (h)
Idle power of machines at each station	[2, 3] (kw/h)
The amount of lubricants used on machines at each station	[0.25, 0.24] (L)

Table 7: Processing time of jobs.

	Manufacturing stage									Inspection and repair stage			
Jobs	First processing (the first layer)			First reentry (the second layer)			Second reentry (the third layer)			First route (the first layer)		First reentry (the second layer)	
	Stations			Stations			Stations			Stations		Stations	
	1	2	3	1	2	3	1	2	3	1	2	1	2
Job1	5	6	4	2	6	8	4	3	6	2	1	2	1
Job2	5	4	13	4	6	7	5	2	3	2	0	1	2
Job3	6	4	3	2	5	7	6	2	0	1	0	2	1
Job4	9	5	8	6	2	5	6	3	4	1	2	2	1

individuals, more job scheduling schemes can be obtained by using random key coding rules. Combine the populations pop1 and pop2, calculate the crowding distance, and perform fast nondominated sorting [43]. Then, select the first N individuals to form the final initial population and get the initial external archive *Archive* which is composed of noninferior solution set.

4.2.4. Subpopulation Evolution Strategy. Divide the whole population into two subpopulations pop_x and pop_y, each with $N/2$ universe individuals. Implement different evolutionary strategies for the two subpopulations. Among them, Levy flight perturbation mutation operation is performed for universe individuals in pop_x, and NSGA-II operation is performed for universe individuals in pop_y. These operations can expand the search scope and the population's diversity, make the algorithm can jump out of local optimum in the process of evolution, and avoid premature convergence.

For subpopulation pop_x, WEP increases gradually during the iteration of the basic MVO algorithm. In the early iteration, it is less than the random number r_2 , and the j-th dimension variable of the i-th universe remains unchanged according to equation (24). This can easily cause the algorithm to fall into the local optimal value. Therefore, this paper introduces the Levy flight perturbation strategy to perturb the position variables of universe individuals in the early iteration and increase the diversity of the population. By introducing the adaptive weighted learning factor w, the offspring universe individuals can maintain the original position information and can also carry out Levy disturbance to a certain extent. The above operation increases the universe individuals' diversity and obtains a new offspring

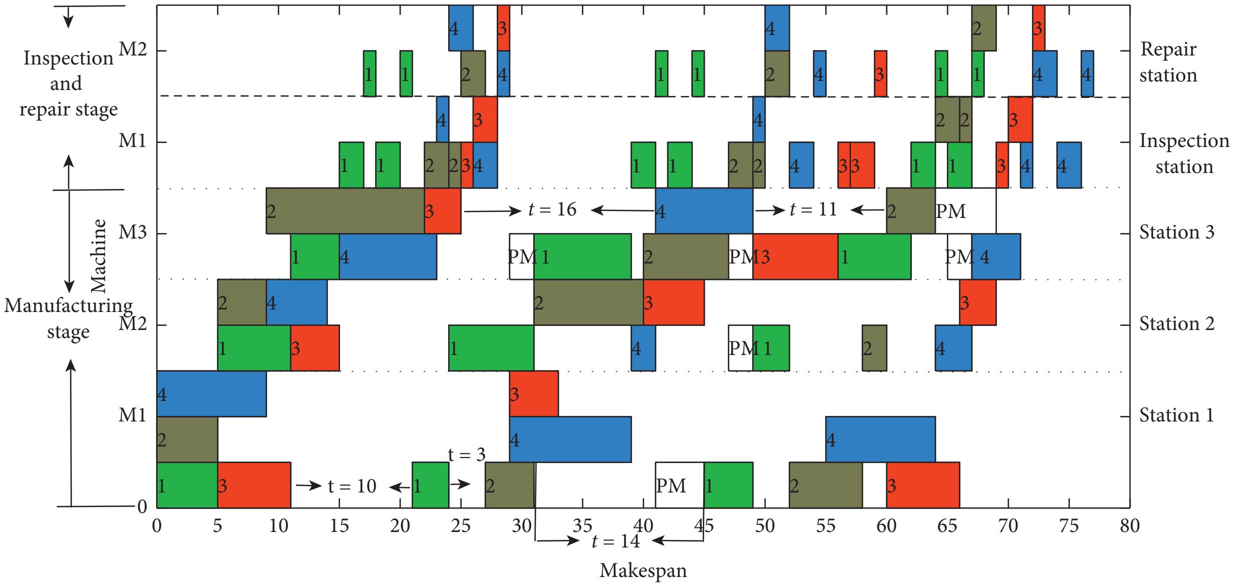


FIGURE 3: Scheduling Gantt chart of the synchronous scheduling and maintenance strategy.

population C1. Equations (28)–(31) are used for changing individuals' position variables.

$$x_i^j = \begin{cases} X_j + \mathrm{TDR} \times \left(\left(ub_j - lb_j\right) \times r_4 + lb_j\right), & r_3 < 0.5,\ r_2 < \mathrm{WEP}, \\ X_j - \mathrm{TDR} \times \left(\left(ub_j - lb_j\right) \times r_4 + lb_j\right), & r_3 \geq 0.5,\ r_2 < \mathrm{WEP}, \\ w x_i^j + (1 - w) \times \text{ Levy}, & r_2 \geq \mathrm{WEP}, \end{cases} \tag{28}$$

$$w = w_{\max} - \left(w_{\max} - w_{\min}\right) \times \left(\frac{l}{L}\right)^2 \tag{29}$$

$$\text{Levy} = \frac{u}{|v|^{1/\beta}}, \quad u \sim N\left(0, \sigma_u^2\right),\ v \sim N\left(0, \sigma_v^2\right), \tag{30}$$

$$\sigma_u = \left\{\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\beta\Gamma\left[(1+\beta)/2\right]^{(\beta-1)/2}}\right\}^{1/\beta}, \quad \sigma_v = 1, \tag{31}$$

where w is a weighted learning factor, Levy is Levy flight step, and β is a parameter between [1, 2], and we set it to 1.5. u and v obey normal distribution.

For subpopulation pop_y, NSGA-II algorithm is used to generate new subpopulations new1 and new2 by LOX crossover [44] and swap mutation. Combine them to calculate the crowding distance and perform fast nondominated ranking; the first $N/2$ individuals are selected to obtain a new offspring population C2. Finally, the subpopulations C1 and C2 are merged to form the final initial population C at the beginning of the next iteration. The noninferior solution set of the population C is obtained, and *Archive* is updated.

5. Simulation Experiments

In order to verify the effectiveness of the proposed IHMMVO algorithm for solving TSGMM-SW, the experimental phase is divided into two parts. The first part selects MMVO, MSSA, and IHMMVO algorithm for comparison. Among them, MMVO is the basic algorithm of IHMMVO, and MSSA has been proved to have high convergence and good divisibility in solving multiobjective optimization problems in literature [45]. Besides, both of them have simple structure, strong search ability, and fast convergence ability in solving complex problems, so we select them as comparative algorithms. In order to further verify the validity of IHMMVO algorithm, the second part selects two canonical multiobjective evolutionary algorithms (MOEA)—NSGA-II algorithm [43] and MOPSO algorithm [46] for comparative analysis, which are classical and effective algorithms for solving multiobjective problems. In the experimental stage, in order to make the three objective functions in the same order of magnitude, the min-max normalization was carried out. In order to ensure the fairness of the comparison, experimental simulation environment is set to Windows 10 operating system, with Intel (R) Core (TM) i7-4770 CPU @3.40 GHz and 8 GB memory, and programming software is MATLAB R2017a.

5.1. Test Cases and Parameter Setting. Since there is no dataset for solving TSGMM-SW in the published literature, the dataset setting method in [17] is referenced to generate test cases. The dataset includes 5 job number criteria, 2 reentry number criteria, 3 station number criteria, and 3 machine number criteria, a total of $5 * 2 * 3 * 3 = 90$ parameter combinations. For each parameter combination, 10 test cases are generated randomly, including $90 * 10 = 900$ test cases. The parameters and their values involved in the dataset are shown in Table 8.

The setting of algorithm parameters has an important influence on its performance. The IHMMVO algorithm designed in this paper involves four key parameters, namely, the maximum value of weighted learning factor $w_{\max}$, the minimum value of weighted learning factor $w_{\min}$, crossover probability $P_{\text{crossover}}$, and mutation probability P_{mutation}. The Taguchi method is used for experiments to study the solution results of various parameter combinations to the scheduling problem, and finally the optimal parameter combination is determined. The remaining parameters are selected based on experience. For the fairness of experiments, the same parameter involved in the rest of comparison algorithms is same with IHMMVO. The specific parameter settings are shown in Table 9.

5.2. Performance Test and Result Analysis. In this paper SP, GD, and IGD are selected as evaluation indicators [44]. For the investigated problem, the Pareto optimal solutions of all test algorithms are regarded as the final Pareto optimal solutions. For TSGMM-SW, 10 test cases are generated randomly by each parameter combination. Simulation experiments are carried out with all the comparison algorithms. The average value is taken as the final result of the parameter combination, and the optimal values of each evaluation indicator are shown in bold. Table 10 shows the experimental result comparison of IHMMVO, MMVO, and MSSA. It can be seen that for SP indicator, the dominant proportion of IHMMVO is 51%, the dominant proportion of MMVO is 43%, and the dominant proportion of MSSA is 6%. For GD indicator, the dominant proportion of IHMMVO is 91%, the dominant proportion of MMVO is 9%, and the dominant proportion of MSSA is 0%. For IGD indicator, the dominant proportion of IHMMVO is 96%, the dominant proportion of MMVO is 4%, the and dominant proportion of MSSA is 0%. All of the above indicates the diversity and convergence of Pareto frontier solutions obtained by IHMMVO algorithm are superior to MMVO and MSSA algorithms; for the distribution of Pareto frontier solutions, IHMMVO is superior to MSSA algorithm, just slightly better than MMVO.

Table 11 shows the experimental result comparison of IHMMVO, NSGA-II, and MOPSO. It can be seen that for GD and IGD indicators, IHMMVO algorithm is superior to NSGA-II and MOPSO algorithm and the dominant proportion is 100%, which indicates the diversity and convergence of Pareto frontier solutions obtained by IHMMVO algorithm are superior to NSGA-II and MOPSO algorithm. For SP indicator, the dominant proportion of IHMMVO is 69%, the dominant proportion of NSGA-II is 10%, and the dominant proportion of MOPSO is 21%. That is, for most of the test cases, IHMMVO is superior to other comparison algorithms in the distribution of Pareto frontier solutions.

Tables 10 and 11 show the advantages and disadvantages of several comparison algorithms from a macro perspective. Table 12 judges whether there is a significant difference between the evaluation indicator of each comparison algorithm by performing Wilcoxon signed rank test on all experimental results. The values outside the brackets are the experimental results between IHMMVO, MMVO, and MSSA, and the values inside the brackets are the experimental results between IHMMVO, NSGA-II, and MOPSO. It can be concluded that there are significant differences between IHMMVO algorithm and the other four comparison algorithms for GD and IGD indicators. For SP indicator, IHMMVO algorithm has significant differences with other comparison algorithms except MMVO. Through the analysis of the above experimental results, it can be inferred that IHMMVO algorithm proposed in this paper can effectively solve TSGMM-SW problem and has certain competitiveness and advantages compared with other comparative algorithms. The Pareto frontiers of several algorithms for $10 * 2 * 5 * 5$ case, $30 * 2 * 8 * 9$ case, $40 * 3 * 5 * 10$ case, and $50 * 2 * 6 * 11$ case are clearly illustrated in Figure 4. We can find IHMMVO is closer to the optimal frontier.

5.3. Case Analysis. In this paper, a $10 * 2 * 8 * 3$ test case is selected for concrete analysis. Assume that there are 8 stations in the production line, including oxidation, deposition, injection, metallization, lithography, etching, polishing, and cleaning. The number of parallel machines on each station is 3, and the number of jobs for processing is 10. The processing level is 2. The reliability thresholds of the three machines at each station are [0.5, 0.4, 0.55]. For other parameters, refer to Table 8. Compare the Pareto optimal solution set and job scheduling schemes obtained by considering PM and without considering PM. The results are shown in Table 13. Among them, the values outside brackets are the experimental results considering PM, while the values inside brackets are the experimental results without considering PM. It can be seen that makespan and total carbon emissions can be optimized to some extent by taking PM operation into account in semiconductor production

Table 8: Main parameters and values in the dataset.

Parameter	Value
Number of jobs	10, 20, 30, 40, 50
Number of stations in manufacturing stage	U [4, 6], U [5, 7], U [6, 8]
Processing time of jobs	U [2, 10]
Inspection time of jobs	U [1, 2]
Processing power of machines at each station in manufacturing stage	U [5, 8]
Reliability of machines at each station in manufacturing stage	U [0.4, 0.6]
m in failure rate function	1.1
Processing power of machines at each station in inspection and repair stage	U [3, 6]
The amount of lubricants used on machines at each station in manufacturing stage	U [0.2, 0.4]
The amount of lubricants used on machines at each station in inspection and repair stage	U [0.2, 0.4]
PM fixed costs of machines	U [10, 15]
PM time of machines	U [2, 5]
Carbon emissions factor of electric energy	1.8742×10^{-7}
Reentry number in manufacturing stage	U [1, 2], U [2, 3]
Number of machines at each station in manufacturing stage	[Number of jobs/5] +1 or +2 or +3
Stations in inspection and repair stage	2
Repair time of jobs	U [0, 2]
Idle power of machines at each station in manufacturing stage	U [1, 5]
Service age regression factor	0.9
β in failure rate function	110
Idle power of machines at each station in inspection and repair stage	U [1, 3]
Effective use time of lubricants on machines at each station in manufacturing stage	U [4, 6]
Effective use time of lubricants on machines at each station in inspection and repair stage	U [4, 6]
PM unit activity costs of machines	U [3, 5]
Machine degradation factor	U [0.1, 0.3]
Carbon emissions factor of lubricants	469

Note. Power unit: kw/h; effective service time unit of lubricants: h; usage unit of lubricants: L; processing time, inspection and repair time, and PM time unit: min; costs unit: RMB; carbon emission factor of electric energy unit: $kgCO_2/J$; carbon emission factor of lubricants: $kgCO_2/m^3$.

Table 9: Main parameters and values of each algorithm.

Parameter	Value
Iteration number	50
WEP_Max (IHMMVO, MMVO)	1
Wmax (IHMMVO)	0.95
$P_{\text{crossover}}$ (IHMMVO, NSGA-II)	0.8
Size of *Archive* (MOPSO)	50
Learning factor c1 (MOPSO)	1
Number of grids per dimension (MOPSO)	7
Popsize	50
WEP_Min (IHMMVO, MMVO)	0.2
Wmin (IHMMVO)	0.05
P_{mutation} (IHMMVO, NSGA-II)	0.3
Inertia weight w (MOPSO)	0.5
Learning factor c2 (MOPSO)	2
Mutation probability (MOPSO)	0.1

line. Besides, Figure 5 shows the scheduling Gantt chart of the third scheduling scheme 4-9-3-5-8-6-7-10-2-1 in Table 13. It can be found that most PM is executed in the idle period of machines, which has little impact on the start time of the next operation. But it can effectively alleviate the degradation effect of machines caused by long-term continuous production and reduce the makespan of jobs as a whole. To sum up in conclusion, PM is an important factor for semiconductor wafer fabrication enterprises to achieve cleaner production.

The sensitivity analysis method is used to analyze the importance degree of reliability threshold R for the three optimization objectives of this paper. The three objective function values are the averages of all the solutions in optimal Pareto solution set. The sensitivity coefficient is calculated by the following equation:

$$S_{fR} = \frac{\Delta f/f}{\Delta R/R}, \tag{32}$$

where S_{fR} is the sensitivity of objective function value to factor variable R, $\Delta f/f$ is the change rate of objective function value f, and $\Delta R/R$ is the change rate of factor variable R. Figure 6 shows the effect of $\Delta R/R$ on the optimization objectives. Figures 6(a)–6(c), respectively, represent the change trend of the three objectives values with the increase of $\Delta R/R$. It can be seen that $C_{\max}$ and TCT increase with the increase of $\Delta R/R$, while TCost decreases. Figure 6(d) shows the change trend of sensitivity coefficient of objective function values. The change curve of $C_{\max}$ and TCT basically coincides, and the latter is slightly larger than the former, indicating that the influence of $\Delta R/R$ on the two is almost the same. The change curve of TCost is obviously above the change curve of the former two, indicating that $\Delta R/R$ had greater influence on TCost than on $C_{\max}$ and TCT.

6. Discussion

From the experimental results in Figure 4, it can be seen that in the multiobjective optimization problem, the optimization of one indicator will inevitably cause the

Table 10: Experimental result comparison of IHMMVO, MMVO, and MSSA.

Cases	IHMMVO			MMVO			MSSA		
	SP	GD	IGD	SP	GD	IGD	SP	GD	IGD
10 ∗ [1, 2] ∗ [4, 6] ∗ 3	**0.2111**	**0.0351**	**0.1042**	0.2288	0.1092	0.2737	0.4912	0.1551	0.4199
10 ∗ [1, 2] ∗ [4, 6] ∗ 4	**0.1445**	**0.0147**	**0.0484**	0.1789	0.0898	0.2943	0.2559	0.0684	0.2558
10 ∗ [1, 2] ∗ [4, 6] ∗ 5	**0.1291**	**0.0261**	**0.1321**	0.1761	0.0536	0.2305	0.1842	0.0883	0.2977
10 ∗ [1, 2] ∗ [5, 7] ∗ 3	0.2960	**0.0211**	**0.0634**	**0.2084**	0.0848	0.2788	0.3067	0.0973	0.3282
10 ∗ [1, 2] ∗ [5, 7] ∗ 4	0.2129	**0.0118**	**0.0316**	**0.1105**	0.0493	0.2209	0.2158	0.1193	0.3420
10 ∗ [1, 2] ∗ [5, 7] ∗ 5	**0.1290**	**0.0332**	**0.0674**	0.2275	0.0972	0.3036	0.3259	0.1255	0.3944
10 ∗ [1, 2] ∗ [6, 8] ∗ 3	0.1984	**0.0492**	**0.1176**	**0.1950**	0.0806	0.2519	0.4591	0.1627	0.4179
10 ∗ [1, 2] ∗ [6, 8] ∗ 4	0.1591	**0.0318**	**0.1296**	0.1700	0.0556	0.1689	**0.1173**	0.1003	0.3046
10 ∗ [1, 2] ∗ [6, 8] ∗ 5	**0.1364**	**0.0208**	**0.0687**	0.1797	0.0413	0.1229	0.3332	0.1524	0.4083
10 ∗ [2, 3] ∗ [4, 6] ∗ 3	0.1487	**0.0204**	**0.0782**	0.1464	0.0373	0.1510	**0.1182**	0.0832	0.3044
10 ∗ [2, 3] ∗ [4, 6] ∗ 4	0.1134	**0.0114**	**0.0692**	**0.1088**	0.0226	0.1128	0.2637	0.1362	0.3881
10 ∗ [2, 3] ∗ [4, 6] ∗ 5	**0.1044**	**0.0103**	**0.0453**	0.1295	0.0186	0.1189	0.1599	0.0503	0.2027
10 ∗ [2, 3] ∗ [5, 7] ∗ 3	0.1767	**0.0283**	**0.1125**	**0.1589**	0.0423	0.1576	0.1936	0.1144	0.2736
10 ∗ [2, 3] ∗ [5, 7] ∗ 4	**0.1357**	**0.0120**	**0.0447**	0.1592	0.0441	0.2020	0.2112	0.0723	0.2319
10 ∗ [2, 3] ∗ [5, 7] ∗ 5	**0.1007**	**0.0154**	**0.0719**	0.1399	0.0397	0.1880	0.2064	0.0718	0.2581
10 ∗ [2, 3] ∗ [6, 8] ∗ 3	**0.1146**	**0.0078**	**0.0432**	0.1810	0.0428	0.1819	0.2413	0.0645	0.2459
10 ∗ [2, 3] ∗ [6, 8] ∗ 4	0.0946	**0.0056**	**0.0305**	**0.0832**	0.0276	0.1281	0.2211	0.0703	0.2024
10 ∗ [2, 3] ∗ [6, 8] ∗ 5	**0.0840**	**0.0093**	**0.0629**	0.1007	0.0179	0.1166	0.2063	0.0639	0.2406
20 ∗ [1, 2] ∗ [4, 6] ∗ 5	0.1833	**0.0194**	**0.0565**	**0.1716**	0.0645	0.2204	0.3157	0.1192	0.3174
20 ∗ [1, 2] ∗ [4, 6] ∗ 6	**0.1694**	**0.0337**	**0.1099**	0.1775	0.0499	0.1543	0.3961	0.1242	0.3253
20 ∗ [1, 2] ∗ [4, 6] ∗ 7	0.1814	**0.0210**	**0.0681**	0.1912	0.0588	0.2030	**0.1793**	0.1053	0.3042
20 ∗ [1, 2] ∗ [5, 7] ∗ 5	**0.1618**	**0.0140**	**0.0681**	0.2094	0.0458	0.1733	0.2015	0.0707	0.2335
20 ∗ [1, 2] ∗ [5, 7] ∗ 6	**0.1335**	**0.0254**	**0.0764**	0.1593	0.0529	0.2180	0.2362	0.0929	0.2714
20 ∗ [1, 2] ∗ [5, 7] ∗ 7	**0.1434**	**0.0193**	**0.0737**	0.1556	0.0633	0.2080	0.3405	0.1122	0.3353
20 ∗ [1, 2] ∗ [6, 8] ∗ 5	0.1837	**0.0170**	**0.0914**	**0.1662**	0.0554	0.1712	0.2963	0.0916	0.2387
20 ∗ [1, 2] ∗ [6, 8] ∗ 6	**0.1766**	**0.0225**	**0.1134**	0.1900	0.0381	0.1651	0.2844	0.1083	0.3180
20 ∗ [1, 2] ∗ [6, 8] ∗ 7	**0.1873**	**0.0272**	**0.0807**	0.2290	0.0515	0.2015	0.3201	0.1125	0.3471
20 ∗ [2, 3] ∗ [4, 6] ∗ 5	0.2254	**0.0434**	**0.1436**	**0.1568**	0.0536	0.1769	0.1780	0.0789	0.2320
20 ∗ [2, 3] ∗ [4, 6] ∗ 6	**0.1333**	**0.0258**	**0.0670**	0.2002	0.0455	0.1692	0.1852	0.0853	0.2867
20 ∗ [2, 3] ∗ [4, 6] ∗ 7	**0.1573**	**0.0404**	**0.1235**	0.1767	0.0901	0.2785	0.2100	0.1275	0.3291
20 ∗ [2, 3] ∗ [5, 7] ∗ 5	0.2091	**0.0117**	**0.0746**	**0.1981**	0.0590	0.2157	0.3242	0.1135	0.3237
20 ∗ [2, 3] ∗ [5, 7] ∗ 6	0.1411	**0.0284**	**0.0982**	0.1772	0.0398	0.1257	**0.1394**	0.0848	0.2450
20 ∗ [2, 3] ∗ [5, 7] ∗ 7	**0.1205**	**0.0248**	**0.1080**	0.1819	0.0547	0.2060	0.3256	0.0732	0.2785
20 ∗ [2, 3] ∗ [6, 8] ∗ 5	**0.1996**	0.0565	0.1985	0.2230	**0.0402**	**0.1414**	0.2734	0.1031	0.3560
20 ∗ [2, 3] ∗ [6, 8] ∗ 6	0.1981	**0.0234**	**0.0988**	**0.1317**	0.0428	0.1553	0.2952	0.1318	0.3438
20 ∗ [2, 3] ∗ [6, 8] ∗ 7	**0.0958**	**0.0055**	**0.0421**	0.1121	0.0259	0.1329	0.1731	0.0565	0.2503
30 ∗ [1, 2] ∗ [4, 6] ∗ 7	0.2310	**0.0521**	**0.1779**	**0.1998**	0.0640	0.2414	0.3125	0.1513	0.3938
30 ∗ [1, 2] ∗ [4, 6] ∗ 8	**0.1667**	**0.0034**	**0.0479**	0.2133	0.0608	0.2220	0.4183	0.1228	0.4149
30 ∗ [1, 2] ∗ [4, 6] ∗ 9	0.2160	**0.0197**	**0.0695**	**0.1413**	0.0711	0.2354	0.2631	0.1744	0.4338
30 ∗ [1, 2] ∗ [5, 7] ∗ 7	**0.2547**	**0.0493**	**0.1380**	0.2777	0.0610	0.1945	0.2641	0.0872	0.2833
30 ∗ [1, 2] ∗ [5, 7] ∗ 8	0.2812	0.0883	0.2083	**0.1767**	**0.0625**	**0.1643**	0.2840	0.1277	0.3452
30 ∗ [1, 2] ∗ [5, 7] ∗ 9	0.3230	**0.0382**	**0.1164**	**0.1678**	0.0721	0.2387	0.3144	0.1849	0.3605
30 ∗ [1, 2] ∗ [6, 8] ∗ 7	0.1680	**0.0062**	**0.0297**	**0.1288**	0.0646	0.2076	0.3130	0.1581	0.4306
30 ∗ [1, 2] ∗ [6, 8] ∗ 8	**0.1596**	**0.0289**	**0.1026**	0.1765	0.0576	0.1709	0.3042	0.1248	0.3437
30 ∗ [1, 2] ∗ [6, 8] ∗ 9	**0.1712**	**0.0733**	**0.2042**	0.2399	0.0805	0.2547	0.5323	0.1496	0.3852
30 ∗ [2, 3] ∗ [4, 6] ∗ 7	0.2274	**0.0680**	**0.1719**	**0.2096**	0.1028	0.2930	0.5621	0.1844	0.4635
30 ∗ [2, 3] ∗ [4, 6] ∗ 8	**0.1231**	**0.0354**	**0.1427**	0.1321	0.0404	0.1579	0.2152	0.1119	0.3282
30 ∗ [2, 3] ∗ [4, 6] ∗ 9	**0.1460**	**0.0273**	**0.1176**	0.2230	0.0598	0.2031	0.4261	0.1177	0.3573
30 ∗ [2, 3] ∗ [5, 7] ∗ 7	0.2043	**0.0447**	**0.1333**	**0.1698**	0.0642	0.2350	0.3514	0.1271	0.3313
30 ∗ [2, 3] ∗ [5, 7] ∗ 8	**0.1673**	**0.0565**	**0.1103**	0.2434	0.0777	0.2365	0.1714	0.1759	0.3362
30 ∗ [2, 3] ∗ [5, 7] ∗ 9	0.1879	0.0603	**0.1654**	**0.1465**	**0.0460**	0.1801	0.2571	0.1383	0.3494
30 ∗ [2, 3] ∗ [6, 8] ∗ 7	**0.1654**	**0.0284**	**0.0714**	0.3713	0.1120	0.3180	0.2981	0.0995	0.3402
30 ∗ [2, 3] ∗ [6, 8] ∗ 8	0.2027	**0.0312**	**0.0775**	**0.1340**	0.0509	0.2127	0.2732	0.1354	0.3612
30 ∗ [2, 3] ∗ [6, 8] ∗ 9	**0.1116**	**0.0169**	**0.0588**	0.1325	0.0346	0.1621	0.1676	0.0780	0.2187
40 ∗ [1, 2] ∗ [4, 6] ∗ 9	0.2580	**0.0000**	**0.0150**	**0.1020**	0.2355	0.2355	0.3105	0.1261	0.2428
40 ∗ [1, 2] ∗ [4, 6] ∗ 10	0.1923	**0.0216**	**0.0900**	**0.1749**	0.0646	0.1941	0.2886	0.0953	0.3135
40 ∗ [1, 2] ∗ [4, 6] ∗ 11	0.2267	**0.0504**	**0.1641**	**0.1435**	0.0525	0.1884	0.2550	0.1579	0.3280
40 ∗ [1, 2] ∗ [5, 7] ∗ 9	0.2595	**0.0352**	**0.1224**	**0.1887**	0.0564	0.2044	0.3557	0.1985	0.4714
40 ∗ [1, 2] ∗ [5, 7] ∗ 10	**0.1749**	**0.0228**	**0.0914**	0.1961	0.0532	0.2090	0.3160	0.1279	0.3693

Table 10: Continued.

Cases	IHMMVO			MMVO			MSSA		
	SP	GD	IGD	SP	GD	IGD	SP	GD	IGD
40 * [1, 2] * [5, 7] * 11	**0.1190**	**0.0543**	**0.1774**	0.1332	0.0728	0.2147	0.4126	0.1425	0.3818
40 * [1, 2] * [6, 8] * 9	0.3196	0.1227	**0.2472**	**0.2467**	**0.1090**	0.2966	0.3736	0.1264	0.3513
40 * [1, 2] * [6, 8] * 10	**0.2265**	**0.0422**	**0.1361**	0.2933	0.0652	0.2226	0.2682	0.1447	0.4057
40 * [1, 2] * [6, 8] * 11	**0.2202**	**0.0255**	**0.0765**	0.2498	0.0541	0.1677	0.3620	0.1869	0.3318
40 * [2, 3] * [4, 6] * 9	**0.2299**	**0.0637**	**0.1699**	0.2626	0.0724	0.2493	0.3363	0.1057	0.3606
40 * [2, 3] * [4, 6] * 10	**0.1372**	**0.0365**	**0.1183**	0.2655	0.0735	0.2437	0.3855	0.1329	0.3697
40 * [2, 3] * [4, 6] * 11	**0.2563**	**0.0198**	**0.0614**	0.2581	0.0607	0.1985	0.5137	0.1728	0.4426
40 * [2, 3] * [5, 7] * 9	**0.1658**	**0.0424**	**0.1410**	0.2285	0.0738	0.2398	0.2372	0.2051	0.4509
40 * [2, 3] * [5, 7] * 10	**0.1344**	0.0518	0.0981	0.3289	**0.0254**	**0.0757**	0.1929	0.1833	0.3366
40 * [2, 3] * [5, 7] * 11	0.1735	**0.0320**	**0.1288**	**0.1214**	0.0720	0.2185	0.2911	0.1492	0.3943
40 * [2, 3] * [6, 8] * 9	0.2417	**0.0236**	**0.0771**	**0.1682**	0.0695	0.2029	0.2877	0.2038	0.4049
40 * [2, 3] * [6, 8] * 10	0.3215	**0.0083**	**0.0234**	**0.1557**	0.0904	0.2742	0.2018	0.1802	0.3564
40 * [2, 3] * [6, 8] * 11	**0.1624**	**0.0531**	**0.1381**	0.2564	0.1118	0.3091	0.5177	0.1482	0.3666
50 * [1, 2] * [4, 6] * 11	0.2166	**0.0537**	**0.1364**	**0.1881**	0.1017	0.2887	0.2644	0.1725	0.3732
50 * [1, 2] * [4, 6] * 12	0.2169	**0.0150**	**0.0583**	**0.1846**	0.0542	0.2119	0.2241	0.1210	0.3495
50 * [1, 2] * [4, 6] * 13	**0.2045**	**0.0368**	**0.1039**	0.2645	0.0909	0.3295	0.2274	0.2018	0.4640
50 * [1, 2] * [5, 7] * 11	0.2773	**0.0752**	**0.1534**	0.3035	0.1322	0.3485	**0.2519**	0.1989	0.4616
50 * [1, 2] * [5, 7] * 12	0.3066	0.0925	**0.2462**	**0.1523**	**0.0909**	0.2790	0.3709	0.2361	0.4576
50 * [1, 2] * [5, 7] * 13	0.2241	**0.0160**	**0.0671**	**0.1557**	0.0631	0.2129	0.3143	0.1690	0.4525
50 * [1, 2] * [6, 8] * 11	0.1641	**0.0261**	**0.0741**	**0.1321**	0.0629	0.2408	0.3091	0.1342	0.3434
50 * [1, 2] * [6, 8] * 12	0.2204	0.0578	0.1773	**0.2047**	**0.0261**	**0.0965**	0.2712	0.1006	0.2909
50 * [1, 2] * [6, 8] * 13	0.2122	**0.0442**	**0.1612**	**0.2037**	0.0657	0.2182	0.2470	0.0938	0.3015
50 * [2, 3] * [4, 6] * 11	0.2491	**0.0727**	**0.2006**	**0.2207**	0.0770	0.2553	0.3589	0.2028	0.4872
50 * [2, 3] * [4, 6] * 12	0.2071	**0.0194**	**0.0735**	**0.1426**	0.0706	0.2373	0.1902	0.1747	0.4244
50 * [2, 3] * [4, 6] * 13	**0.2149**	**0.0451**	**0.1511**	0.2246	0.0630	0.1912	0.3652	0.1059	0.2977
50 * [2, 3] * [5, 7] * 11	**0.1617**	**0.0387**	**0.1454**	0.1688	0.0642	0.2320	0.3143	0.1248	0.3920
50 * [2, 3] * [5, 7] * 12	0.1642	**0.0430**	**0.1435**	**0.1551**	0.0870	0.2182	0.1815	0.1317	0.3426
50 * [2, 3] * [5, 7] * 13	**0.1894**	**0.0454**	**0.1295**	0.2769	0.1238	0.2686	0.5330	0.1729	0.4383
50 * [2, 3] * [6, 8] * 11	0.2460	**0.0287**	**0.1046**	**0.2150**	0.0639	0.2436	0.2273	0.1755	0.4135
50 * [2, 3] * [6, 8] * 12	**0.1911**	0.0873	**0.2021**	0.2173	**0.0841**	0.2384	0.3899	0.1646	0.4175
50 * [2, 3] * [6, 8] * 13	**0.1931**	**0.0779**	**0.1859**	0.2398	0.0798	0.2171	0.3755	0.2248	0.4421

Table 11: Experimental result comparison of IHMMVO, NSGA-II, and MOPSO.

Cases	IHMMVO			NSGA-II			MOPSO		
	SP	GD	IGD	SP	GD	IGD	SP	GD	IGD
10 * [1, 2] * [4, 6] * 3	**0.1496**	**0.0114**	**0.0214**	0.2228	0.1220	0.3443	0.2214	0.0874	0.2860
10 * [1, 2] * [4, 6] * 4	0.1913	**0.0222**	**0.0746**	0.2772	0.1132	0.3268	**0.1881**	0.0728	0.2473
10 * [1, 2] * [4, 6] * 5	0.2432	**0.0209**	**0.0544**	**0.2249**	0.1192	0.3119	0.2319	0.0924	0.2576
10 * [1, 2] * [5, 7] * 3	**0.1511**	**0.0179**	**0.0602**	0.2392	0.1075	0.2903	0.2383	0.0763	0.2610
10 * [1, 2] * [5, 7] * 4	**0.1569**	**0.0147**	**0.0423**	0.4381	0.1234	0.3865	0.1919	0.0651	0.2390
10 * [1, 2] * [5, 7] * 5	**0.1476**	**0.0004**	**0.0068**	0.3057	0.0800	0.2660	0.1952	0.0718	0.2466
10 * [1, 2] * [6, 8] * 3	**0.1465**	**0.0094**	**0.0248**	0.3124	0.1311	0.3478	0.2375	0.0933	0.2863
10 * [1, 2] * [6, 8] * 4	**0.1569**	**0.0142**	**0.0464**	0.2149	0.1743	0.3086	0.2935	0.0968	0.2570
10 * [1, 2] * [6, 8] * 5	**0.1875**	**0.0102**	**0.0264**	0.2880	0.1739	0.3256	0.2563	0.0844	0.2340
10 * [2, 3] * [4, 6] * 3	0.1861	**0.0094**	**0.0312**	0.2639	0.1598	0.3892	**0.1758**	0.0874	0.2632
10 * [2, 3] * [4, 6] * 4	**0.1355**	**0.0100**	**0.0294**	0.2797	0.1047	0.3226	0.1737	0.1024	0.2766
10 * [2, 3] * [4, 6] * 5	0.1945	**0.0142**	**0.0502**	0.2542	0.1202	0.3098	**0.1537**	0.0666	0.2144
10 * [2, 3] * [5, 7] * 3	0.2221	**0.0220**	**0.0755**	0.2965	0.1007	0.2773	**0.2068**	0.1219	0.2672
10 * [2, 3] * [5, 7] * 4	**0.1733**	**0.0048**	**0.0187**	0.2367	0.1251	0.3323	0.2024	0.0802	0.2624
10 * [2, 3] * [5, 7] * 5	**0.1800**	**0.0101**	**0.0336**	0.2202	0.1031	0.3052	0.2074	0.0606	0.2236
10 * [2, 3] * [6, 8] * 3	**0.1947**	**0.0088**	**0.0343**	0.2032	0.1341	0.3700	0.2596	0.1154	0.3584
10 * [2, 3] * [6, 8] * 4	**0.1851**	**0.0128**	**0.0546**	0.2653	0.1155	0.3100	0.2608	0.0697	0.2533
10 * [2, 3] * [6, 8] * 5	**0.1183**	**0.0129**	**0.0528**	0.2583	0.0774	0.2656	0.1846	0.0528	0.1955
20 * [1, 2] * [4, 6] * 5	**0.2346**	**0.0045**	**0.0049**	0.2913	0.1171	0.3346	0.2769	0.1098	0.2620
20 * [1, 2] * [4, 6] * 6	**0.1332**	**0.0137**	**0.0390**	0.4063	0.1255	0.3818	0.2866	0.0875	0.3024
20 * [1, 2] * [4, 6] * 7	**0.1617**	**0.0103**	**0.0358**	0.2988	0.1031	0.3000	0.2462	0.0797	0.2592
20 * [1, 2] * [5, 7] * 5	**0.1277**	**0.0066**	**0.0127**	0.3383	0.1149	0.3156	0.1578	0.0682	0.2148
20 * [1, 2] * [5, 7] * 6	**0.1635**	**0.0131**	**0.0531**	0.3107	0.1383	0.3316	0.2368	0.0979	0.2929

Table 11: Continued.

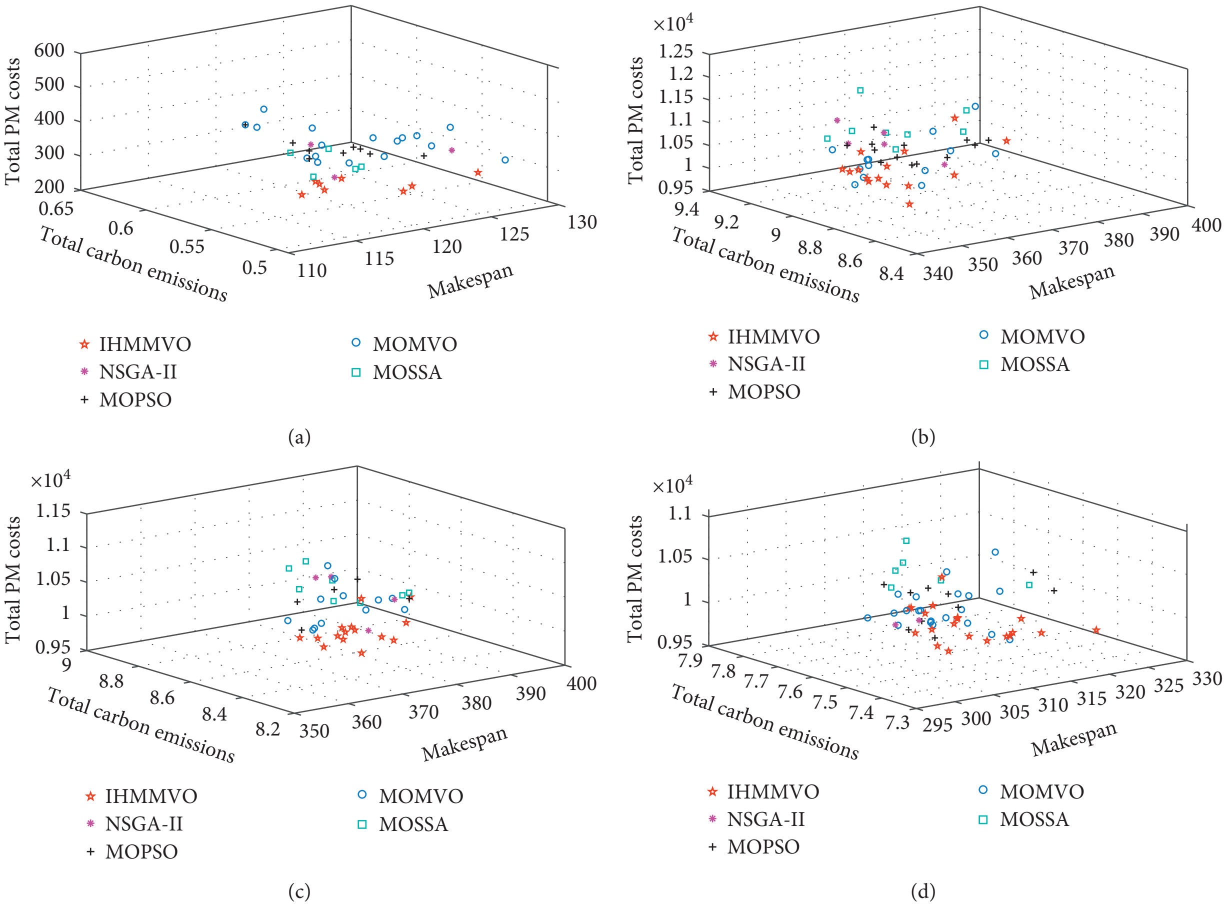
Cases	IHMMVO			NSGA-II			MOPSO		
	SP	GD	IGD	SP	GD	IGD	SP	GD	IGD
20 ∗ [1, 2] ∗ [5, 7] ∗ 7	**0.1171**	**0.0033**	**0.0094**	0.2986	0.0782	0.2499	0.1634	0.0603	0.2333
20 ∗ [1, 2] ∗ [6, 8] ∗ 5	**0.1461**	**0.0265**	**0.0741**	0.2586	0.1349	0.3585	0.3942	0.1171	0.3562
20 ∗ [1, 2] ∗ [6, 8] ∗ 6	0.1840	**0.0145**	**0.0504**	0.2142	0.1161	0.2747	**0.1662**	0.1235	0.2494
20 ∗ [1, 2] ∗ [6, 8] ∗ 7	**0.1736**	**0.0128**	**0.0425**	0.2102	0.0967	0.2755	0.2455	0.0846	0.2749
20 ∗ [2, 3] ∗ [4, 6] ∗ 5	0.2136	**0.0000**	**0.0360**	**0.1756**	0.1004	0.2870	0.3895	0.1135	0.3041
20 ∗ [2, 3] ∗ [4, 6] ∗ 6	0.1780	**0.0039**	**0.0150**	0.2095	0.1154	0.3015	**0.1335**	0.1038	0.2592
20 ∗ [2, 3] ∗ [4, 6] ∗ 7	**0.1351**	**0.0124**	**0.0502**	0.2775	0.1393	0.2922	0.3344	0.0943	0.3106
20 ∗ [2, 3] ∗ [5, 7] ∗ 5	**0.1645**	**0.0308**	**0.0552**	0.3234	0.1409	0.3149	0.2464	0.0877	0.2608
20 ∗ [2, 3] ∗ [5, 7] ∗ 6	**0.1980**	**0.0077**	**0.0271**	0.2731	0.1546	0.3670	0.3290	0.1072	0.3309
20 ∗ [2, 3] ∗ [5, 7] ∗ 7	**0.1645**	**0.0308**	**0.0552**	0.3234	0.1409	0.3149	0.2464	0.0877	0.2608
20 ∗ [2, 3] ∗ [6, 8] ∗ 5	**0.1154**	**0.0035**	**0.0129**	0.2387	0.1186	0.2863	0.2467	0.0719	0.1950
20 ∗ [2, 3] ∗ [6, 8] ∗ 6	0.1778	**0.0000**	**0.0000**	0.2698	0.0774	0.2489	**0.1188**	0.0923	0.2853
20 ∗ [2, 3] ∗ [6, 8] ∗ 7	**0.1745**	**0.0046**	**0.0260**	0.2857	0.1357	0.3467	0.2173	0.0758	0.2552
30 ∗ [1, 2] ∗ [4, 6] ∗ 7	**0.1331**	**0.0236**	**0.0734**	0.2678	0.0917	0.3017	0.1950	0.0897	0.2817
30 ∗ [1, 2] ∗ [4, 6] ∗ 8	**0.1489**	**0.0111**	**0.0386**	0.2955	0.0778	0.2497	0.2342	0.0751	0.2575
30 ∗ [1, 2] ∗ [4, 6] ∗ 9	**0.1391**	**0.0104**	**0.0432**	0.3944	0.0999	0.3627	0.2292	0.0655	0.2230
30 ∗ [1, 2] ∗ [5, 7] ∗ 7	0.1393	**0.0085**	**0.0448**	**0.1387**	0.1182	0.3251	0.1919	0.0672	0.2460
30 ∗ [1, 2] ∗ [5, 7] ∗ 8	**0.1258**	**0.0083**	**0.0348**	0.2823	0.1027	0.3061	0.1451	0.0754	0.2611
30 ∗ [1, 2] ∗ [5, 7] ∗ 9	**0.1248**	**0.0204**	**0.0640**	0.2511	0.1126	0.3278	0.3262	0.1087	0.3313
30 ∗ [1, 2] ∗ [6, 8] ∗ 7	**0.1568**	**0.0051**	**0.0289**	0.2510	0.1274	0.3376	0.2176	0.0458	0.1967
30 ∗ [1, 2] ∗ [6, 8] ∗ 8	**0.1214**	**0.0078**	**0.0279**	0.1920	0.0974	0.2897	0.2476	0.0713	0.2480
30 ∗ [1, 2] ∗ [6, 8] ∗ 9	**0.1229**	**0.0078**	**0.0267**	0.2318	0.1086	0.3159	0.1908	0.0629	0.2233
30 ∗ [2, 3] ∗ [4, 6] ∗ 7	0.1726	**0.0149**	**0.0365**	0.1738	0.1308	0.3174	**0.1446**	0.0780	0.2238
30 ∗ [2, 3] ∗ [4, 6] ∗ 8	**0.1302**	**0.0112**	**0.0309**	0.4057	0.0879	0.2913	0.1896	0.0683	0.2562
30 ∗ [2, 3] ∗ [4, 6] ∗ 9	**0.1468**	**0.0128**	**0.0555**	0.2925	0.1301	0.3654	0.2079	0.0591	0.2108
30 ∗ [2, 3] ∗ [5, 7] ∗ 7	0.1732	**0.0199**	**0.0536**	0.2506	0.0794	0.2627	**0.1502**	0.0708	0.2428
30 ∗ [2, 3] ∗ [5, 7] ∗ 8	**0.1800**	**0.0046**	**0.0328**	0.3314	0.1150	0.3387	0.2346	0.0805	0.2798
30 ∗ [2, 3] ∗ [5, 7] ∗ 9	**0.1184**	**0.0230**	**0.0835**	0.1863	0.1499	0.3355	0.2032	0.0792	0.2703
30 ∗ [2, 3] ∗ [6, 8] ∗ 7	**0.1348**	**0.0163**	**0.0507**	0.3173	0.1360	0.3672	0.2521	0.0989	0.2863
30 ∗ [2, 3] ∗ [6, 8] ∗ 8	**0.1603**	**0.0400**	**0.1018**	0.2183	0.1049	0.3053	0.2220	0.0900	0.2642
30 ∗ [2, 3] ∗ [6, 8] ∗ 9	**0.1481**	**0.0120**	**0.0321**	0.3486	0.1447	0.3748	0.1837	0.0581	0.2280
40 ∗ [1, 2] ∗ [4, 6] ∗ 9	**0.2309**	**0.0000**	**0.0000**	0.2676	0.1559	0.3808	0.2688	0.0908	0.2982
40 ∗ [1, 2] ∗ [4, 6] ∗ 10	**0.2662**	**0.0055**	**0.0281**	0.2913	0.1206	0.3186	0.3053	0.1052	0.3046
40 ∗ [1, 2] ∗ [4, 6] ∗ 11	**0.2553**	**0.0041**	**0.0104**	0.4165	0.2267	0.4868	0.3073	0.1817	0.4043
40 ∗ [1, 2] ∗ [5, 7] ∗ 9	**0.1763**	**0.0221**	**0.0550**	0.2919	0.1741	0.3998	0.3703	0.1186	0.3137
40 ∗ [1, 2] ∗ [5, 7] ∗ 10	**0.2126**	**0.0236**	**0.0949**	0.3213	0.1756	0.4166	0.2596	0.0997	0.3243
40 ∗ [1, 2] ∗ [5, 7] ∗ 11	0.2844	**0.0118**	**0.0395**	**0.2126**	0.1455	0.3484	0.3308	0.0809	0.2285
40 ∗ [1, 2] ∗ [6, 8] ∗ 9	0.2958	**0.0053**	**0.0235**	0.4481	0.1187	0.3412	**0.2480**	0.1116	0.3124
40 ∗ [1, 2] ∗ [6, 8] ∗ 10	0.2554	**0.0300**	**0.0703**	**0.1647**	0.1793	0.4216	0.3722	0.1930	0.4193
40 ∗ [1, 2] ∗ [6, 8] ∗ 11	0.3100	**0.0026**	**0.0143**	**0.2115**	0.1769	0.4070	0.3185	0.1315	0.3364
40 ∗ [2, 3] ∗ [4, 6] ∗ 9	**0.2689**	**0.0000**	**0.0068**	0.4875	0.2208	0.5339	0.4478	0.1239	0.3950
40 ∗ [2, 3] ∗ [4, 6] ∗ 10	**0.2122**	**0.0133**	**0.0269**	0.3338	0.1860	0.4170	0.3545	0.1514	0.3393
40 ∗ [2, 3] ∗ [4, 6] ∗ 11	0.1754	**0.0224**	**0.0473**	**0.1689**	0.1794	0.4592	0.2627	0.1073	0.2974
40 ∗ [2, 3] ∗ [5, 7] ∗ 9	**0.1080**	**0.0030**	**0.0086**	0.7716	0.0990	0.4251	0.1737	0.0902	0.3045
40 ∗ [2, 3] ∗ [5, 7] ∗ 10	0.2806	**0.0338**	**0.1003**	0.3181	0.1087	0.2586	**0.2172**	0.0945	0.2636
40 ∗ [2, 3] ∗ [5, 7] ∗ 11	0.1924	**0.0058**	**0.0297**	0.3633	0.1047	0.3497	**0.1763**	0.1082	0.3214
40 ∗ [2, 3] ∗ [6, 8] ∗ 9	**0.1887**	**0.0091**	**0.0112**	0.5673	0.2438	0.5562	0.2603	0.1405	0.3259
40 ∗ [2, 3] ∗ [6, 8] ∗ 10	**0.1484**	**0.0107**	**0.0277**	0.7492	0.1264	0.3498	0.2435	0.1019	0.2478
40 ∗ [2, 3] ∗ [6, 8] ∗ 11	**0.1735**	**0.0000**	**0.0103**	0.3329	0.1769	0.4305	0.1877	0.0983	0.2854
50 ∗ [1, 2] ∗ [4, 6] ∗ 11	**0.1878**	**0.0262**	**0.1002**	0.2299	0.1344	0.3134	0.3463	0.1234	0.3204
50 ∗ [1, 2] ∗ [4, 6] ∗ 12	**0.1564**	**0.0082**	**0.0161**	0.1860	0.1081	0.3421	0.2043	0.1413	0.3517
50 ∗ [1, 2] ∗ [4, 6] ∗ 13	**0.1903**	**0.0000**	**0.0000**	0.2726	0.1184	0.2874	0.2658	0.1225	0.2460
50 ∗ [1, 2] ∗ [5, 7] ∗ 11	**0.1558**	**0.0000**	**0.0034**	0.2823	0.1080	0.3100	0.1764	0.0940	0.2838
50 ∗ [1, 2] ∗ [5, 7] ∗ 12	**0.1787**	**0.0169**	**0.0575**	0.2181	0.1005	0.3092	0.3980	0.1292	0.3503
50 ∗ [1, 2] ∗ [5, 7] ∗ 13	0.2139	**0.0514**	**0.1253**	0.2611	0.1305	0.3363	**0.2134**	0.1159	0.2864
50 ∗ [1, 2] ∗ [6, 8] ∗ 11	0.2177	**0.0279**	**0.0661**	0.4901	0.1410	0.4360	**0.1843**	0.0890	0.2719
50 ∗ [1, 2] ∗ [6, 8] ∗ 12	0.2208	**0.0222**	**0.0608**	0.3102	0.1418	0.3927	**0.2175**	0.1094	0.3635
50 ∗ [1, 2] ∗ [6, 8] ∗ 13	**0.1434**	**0.0091**	**0.0181**	0.3467	0.1289	0.3968	0.3834	0.1187	0.3836
50 ∗ [2, 3] ∗ [4, 6] ∗ 11	0.2136	**0.0168**	**0.0939**	**0.1921**	0.1593	0.3851	0.3105	0.0848	0.2805

Table 11: Continued.

Cases	IHMMVO			NSGA-II			MOPSO		
	SP	GD	IGD	SP	GD	IGD	SP	GD	IGD
50 * [2, 3] * [4, 6] * 12	**0.1502**	**0.0226**	**0.0490**	0.2881	0.1591	0.3901	0.2282	0.1235	0.2532
50 * [2, 3] * [4, 6] * 13	0.2465	**0.0316**	**0.0804**	0.5516	0.2442	0.5482	**0.1684**	0.0846	0.2169
50 * [2, 3] * [5, 7] * 11	**0.2685**	**0.0105**	**0.0360**	0.3180	0.2367	0.4556	0.3790	0.0967	0.3229
50 * [2, 3] * [5, 7] * 12	0.2661	**0.0079**	**0.0204**	**0.1047**	0.1130	0.3439	0.3405	0.1174	0.3050
50 * [2, 3] * [5, 7] * 13	0.2424	**0.0132**	**0.0408**	0.3205	0.2062	0.4130	**0.1761**	0.0819	0.2534
50 * [2, 3] * [6, 8] * 11	**0.0925**	**0.0069**	**0.0040**	0.3357	0.1110	0.2952	0.2634	0.1017	0.2890
50 * [2, 3] * [6, 8] * 12	0.2268	**0.0038**	**0.0104**	0.3904	0.1292	0.3293	**0.2124**	0.1455	0.3110
50 * [2, 3] * [6, 8] * 13	0.2972	**0.0195**	**0.0557**	0.4237	0.1701	0.3785	**0.1748**	0.0980	0.2563

Table 12: The results of Wilcoxon signed rank test.

Indicator	MMVO		MSSA		NSGA-II		MOPSO	
	P	Sig ($P < 0.05$)	P	Sig ($P < 0.05$)	P	Sig ($P < 0.05$)	P	Sig ($P < 0.05$)
SP	0.624 (0.000)	N(Y)	0.000 (0.000)	Y(Y)	0.000 (0.000)	Y(Y)	0.000 (0.000)	Y(Y)
GD	0.000 (0.000)	Y(Y)	0.000 (0.000)	Y(Y)	0.000 (0.000)	Y(Y)	0.000 (0.000)	Y(Y)
IGD	0.000 (0.000)	Y(Y)	0.000 (0.000)	Y(Y)	0.000 (0.000)	Y(Y)	0.000 (0.000)	Y(Y)



(a)

(b)

(c)

(d)

Figure 4: Comparison diagrams of Pareto frontier. (a) 10 * 2 * 5 * 5. (b) 30 * 2 * 8 * 9. (c) 40 * 3 * 5 * 10. (d) 50 * 2 * 6 * 11.

deterioration of at least one other indicator. The obtained Pareto solution and the corresponding scheduling scheme can be an effective reference tool to help managers make intelligent decisions based on the actual situation of the enterprise and the goals they are focusing on when considering the production efficiency, green production,

Table 13: Pareto optimal solution set obtained by IHMMVO algorithm (part).

	TCT (kgCO$_2$)	TCost (RMB)	$C_{\max}$ (min)	Scheduling scheme
1	2.9 (3.7)	1022 (0)	366 (421)	1-3-8-9-10-6-5-4-2-7
2	2.9 (3.4)	985 (0)	349 (412)	4-3-6-10-5-8-7-1-2-9
3	3.1 (3.5)	847 (0)	355 (404)	**4-9-3-5-8-6-7-10-2-1**
4	3.2 (3.6)	813 (0)	361 (413)	8-4-3-1-9-7-6-10-5-2
5	3.0 (3.5)	936 (0)	359 (410)	9-3-4-10-2-8-7-1-5-6
6	3.1 (3.4)	1152 (0)	342 (379)	10-6-4-7-5-3-8-1-2-9
7	3.0 (3.2)	884 (0)	364 (389)	1-4-3-5-6-2-9-7-8-10
8	3.0 (3.5)	852 (0)	349 (411)	1-2-3-4-7-5-6-10-8-9
9	3.0 (3.5)	920 (0)	353 (430)	1-2-3-6-5-4-8-7-9-10
10	2.9 (3.3)	1025 (0)	348 (401)	1-3-7-4-5-6-2-8-10-9

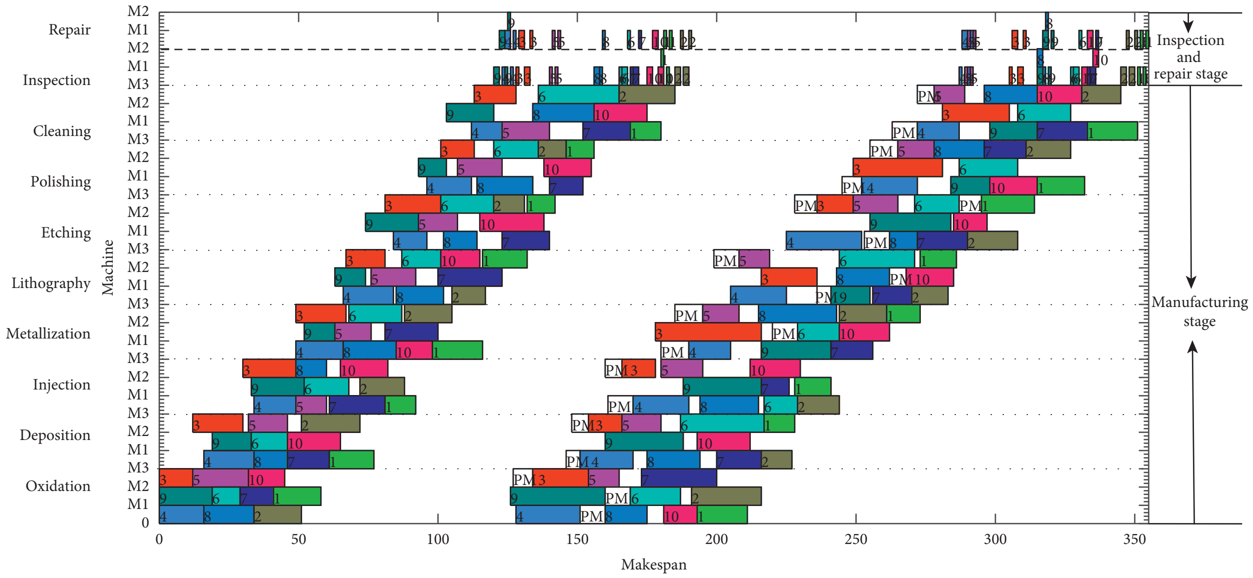
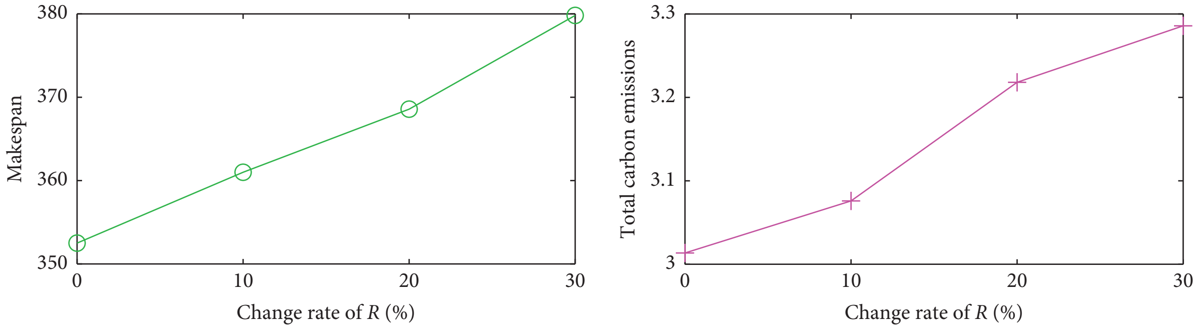


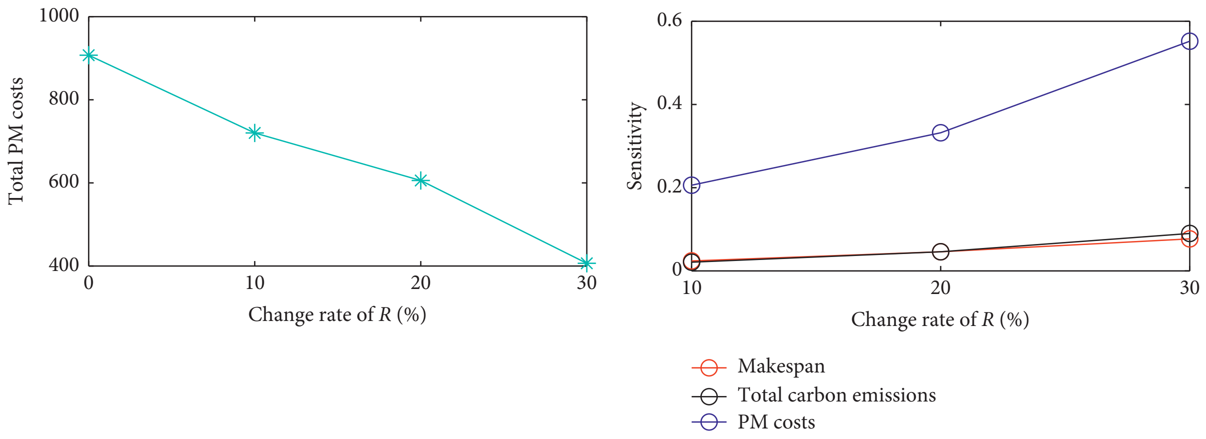
Figure 5: Scheduling Gantt chart of 4-9-3-5-8-6-7-10-2-1 scheme.



(a)

(b)

Figure 6: Continued.



(c)

(d)

Figure 6: The impact of $\Delta R/R$ on the optimization objectives.

and economic benefit balance in the semiconductor wafer manufacturing environment. In this paper, IHMMVO algorithm is used to perform a comparative experiment on a test case of $10 * 2 * 8 * 3$ scale, showing the optimization objective values of partial solutions obtained with and without considering PM operation. On the other hand, sensitivity analysis is performed to show how the change rate of R affects the three objective function values. It can be concluded that the change rate of R has little effect on $C_{\max}$ and TCT, which is basically the same, while the effect on TCost is greater than the former two. Based on this, enterprise decision makers can set appropriate machine reliability threshold to reduce PM costs and improve economic benefits on the basis of no significant impact on $C_{\max}$ and TCT indicators. In short, the research of this paper can provide a certain reference value for the future transition of semiconductor wafer manufacturing to a clean production mode while weighing production efficiency and economic benefits. However, this paper mainly focuses on the deterministic scheduling of semiconductor wafer manufacturing, without considering the uncertainty of the production process.

7. Conclusions and Future Work

In this paper, a two-stage joint optimization model of green manufacturing and maintenance for semiconductor wafers is constructed for the first time, and an IHMMVO algorithm is proposed to solve the problem. Initial population fusion strategy, subpopulation evolution strategy, and synchronous scheduling and maintenance strategy are designed in IHMMVO algorithm. The algorithm can jump out of local optimum and obtain better Pareto frontier solution set, which realizes the two-stage joint optimization of green manufacturing and PM. Enterprise decision makers can choose an optimal scheduling scheme according to the actual situation. The simulation results show that the proposed algorithm has certain advantages and competitiveness in solving TSGMM-SW. Finally, through the specific experimental results analysis of a simulation case in semiconductor wafer manufacturing stage, it can be seen that makespan and total carbon emission indicators can be optimized to a certain extent by considering PM operation on semiconductor production line under reasonable machine reliability threshold setting. It can make a contribution to green manufacturing of semiconductor wafer enterprises.

With the development of collaborative production mode, distributed manufacturing has become a trend. In the future, we will focus on the research of semiconductor wafer distributed green manufacturing scheduling problem considering equipment preventive maintenance. We will also pay more attention to the uncertain scheduling problem in semiconductor wafer production. In addition, with the pursuit of "things perception, things interconnection, and things intelligence" in manufacturing workshop, combining IoT, cloud computing, and big data technologies to build intelligent scheduling algorithms and scheduling rules libraries and combining deep learning and machine learning method to construct more efficient algorithms are also our future effort direction.

Data Availability

The data used to support the findings of this paper are from previously reported studies, which have been cited.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under grant no. 71840003 and Science and Technology Development Project of University of Shanghai for Science and Technology under grant no. 2018KJFZ043.

References

[1] H.-M. Cho, S.-J. Bae, J. Kim, and I.-J. Jeong, “Bi-objective scheduling for reentrant hybrid flow shop using Pareto genetic algorithm,” *Computers & Industrial Engineering*, vol. 61, no. 3, pp. 529–541, 2011.

[2] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, “Multi-verse optimizer: a nature-inspired algorithm for global optimization,” *Neural Computing and Applications*, vol. 27, no. 2, pp. 495–513, 2016.

[3] H. Faris, I. Aljarah, and S. Mirjalili, “Training feedforward neural networks using multi-verse optimizer for binary classification problems,” *Applied Intelligence*, vol. 45, no. 2, pp. 322–332, 2016.

[4] M. A. Elaziz, D. Oliva, A. A. Ewees, and S. Xiong, “Multi-level thresholding-based grey scale image segmentation using multi-objective multi-verse optimizer,” *Expert Systems with Applications*, vol. 125, pp. 112–129, 2019.

[5] J. Lin, L. Zhu, and Z.-J. Wang, “A hybrid multi-verse optimization for the fuzzy flexible job-shop scheduling problem,” *Computers & Industrial Engineering*, vol. 127, pp. 1089–1100, 2019.

[6] S. H. Kim and Y. H. Lee, “Synchronized production planning and scheduling in semiconductor fabrication,” *Computers & Industrial Engineering*, vol. 96, pp. 72–85, 2016.

[7] A. Bitar, S. Dauzère-Pérès, C. Yugma, and R. Roussel, “A memetic algorithm to solve an unrelated parallel machine scheduling problem with auxiliary resources in semiconductor manufacturing,” *Journal of Scheduling*, vol. 19, no. 4, pp. 367–376, 2016.

[8] T. Jamrus, C. F. Chien, M. Gen, and K. Sethanan, “Hybrid particle swarm optimization combined with genetic operators for flexible job-shop scheduling under uncertain processing time for semiconductor manufacturing,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, no. 1, pp. 32–41, 2018.

[9] W. Jia, Z. Jiang, and Y. Li, “Combined scheduling algorithm for re-entrant batch-processing machines in semiconductor wafer manufacturing,” *International Journal of Production Research*, vol. 53, no. 6, pp. 1866–1879, 2015.

[10] Y. Hur, J. F. Bard, and R. Chacon, “Hierarchy machine set-up for multi-pass lot scheduling at semiconductor assembly and test facilities,” *International Journal of Production Research*, vol. 57, no. 14, pp. 4351–4370, 2019.

[11] M. Foumani, A. Razeghi, and K. Smith-Miles, “Stochastic optimization of two-machine flow shop robotic cells with controllable inspection times: from theory toward practice,” *Robotics and Computer-Integrated Manufacturing*, vol. 61, Article ID 101822, 2020.

[12] M. Foumani, I. Gunawan, K. Smith-Miles et al., “Increasing throughput for a class of two-machine robotic cells served by a multifunction robot,” *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 1150–1159, 2017.

[13] M. Foumani, K. Smith-Miles, and I. Gunawan, “Scheduling of two-machine robotic rework cells: in-process, post-process and in-line inspection scenarios,” *Robotics and Autonomous Systems*, vol. 91, pp. 210–225, 2017.

[14] M. Foumani, K. Smith-Miles, I. Gunawan, and A. Moeini, “A framework for stochastic scheduling of two-machine robotic rework cells with in-process inspection system,” *Computers & Industrial Engineering*, vol. 112, pp. 492–502, 2017.

[15] I. Tirkel, “The efficiency of inspection based on out of control detection in wafer fabrication,” *Computers & Industrial Engineering*, vol. 99, pp. 458–464, 2016.

[16] M. Ulrich, G. Lux, L. Jürgensen, and G. Reinhart, “Automated and cycle time optimized path planning for robot-based inspection systems,” *Procedia CIRP*, vol. 44, pp. 377–382, 2016.

[17] B.-H. Zhou, L.-M. Hu, and Z.-Y. Zhong, “A hybrid differential evolution algorithm with estimation of distribution algorithm for reentrant hybrid flow shop scheduling problem,” *Neural Computing and Applications*, vol. 30, no. 1, pp. 193–209, 2018.

[18] S. M. Mousavi, I. Mahdavi, J. Rezaeian, and M. Zandieh, “An efficient bi-objective algorithm to solve re-entrant hybrid flow shop scheduling with learning effect and setup times,” *Operational Research*, vol. 18, no. 1, pp. 123–158, 2018.

[19] J.-N. Shen, L. Wang, and H.-Y. Zheng, “A modified teaching-learning-based optimisation algorithm for bi-objective re-entrant hybrid flowshop scheduling,” *International Journal of Production Research*, vol. 54, no. 12, pp. 3622–3639, 2016.

[20] K.-C. Ying, S.-W. Lin, and S.-Y. Wan, “Bi-objective reentrant hybrid flowshop scheduling: an iterated Pareto greedy algorithm,” *International Journal of Production Research*, vol. 52, no. 19, pp. 5735–5747, 2014.

[21] X. Y. Zhang and L. Chen, “A re-entrant hybrid flow shop scheduling problem with machine eligibility constraints,” *International Journal of Production Research*, vol. 56, no. 16, pp. 5293–5305, 2018.

[22] C. Sangsawang, K. Sethanan, T. Fujimoto, and M. Gen, “Metaheuristics optimization approaches for two-stage re-entrant flexible flow shop with blocking constraint,” *Expert Systems with Applications*, vol. 42, no. 5, pp. 2395–2410, 2015.

[23] C. Chamnanlor, K. Sethanan, M. Gen, and C.-F. Chien, “Embedding ant system in genetic algorithm for re-entrant hybrid flow shop scheduling problems with time window constraints,” *Journal of Intelligent Manufacturing*, vol. 28, no. 8, pp. 1915–1931, 2017.

[24] L. Yin, X. Li, L. Gao, C. Lu, and Z. Zhang, “A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem,” *Sustainable Computing: Informatics and Systems*, vol. 13, pp. 15–30, 2017.

[25] Q. Liu, M. Zhan, F. O. Chekem, X. Shao, B. Ying, and J. W. Sutherland, “A hybrid fruit fly algorithm for solving flexible job-shop scheduling to reduce manufacturing carbon footprint,” *Journal of Cleaner Production*, vol. 168, pp. 668–678, 2017.

[26] N. Sihag and K. S. Sangwan, “Development of a multi-criteria optimization model for minimizing carbon emissions and processing time during machining,” *Procedia Cirp*, vol. 69, pp. 300–305, 2018.

[27] J. Wang, S. Yao, J. Sheng, and H. Yang, “Minimizing total carbon emissions in an integrated machine scheduling and vehicle routing problem,” *Journal of Cleaner Production*, vol. 229, pp. 1004–1017, 2019.

[28] M. Foumani and K. Smith-Miles, “The impact of various carbon reduction policies on green flowshop scheduling,” *Applied Energy*, vol. 249, pp. 300–315, 2019.

[29] R. Wang and Y. Feng, “Evaluation research on green degree of equipment manufacturing industry based on improved particle swarm optimization algorithm,” *Chaos, Solitons & Fractals*, 2019.

[30] X. Y. Zhang, X. G. Ming, Z. W. Liu et al., “General reference model and overall frameworks for green manufacturing,” *Journal of Cleaner Production*, vol. 237, no. 10, Article ID 117757, 2019.

[31] Q. Liu, M. Dong, F. F. Chen, W. Lv, and C. Ye, “Single-machine-based joint optimization of predictive maintenance planning and production scheduling,” *Robotics and Computer-Integrated Manufacturing*, vol. 55, pp. 173–182, 2019.

[32] R.-H. Huang and S.-C. Yu, "Two-stage multiprocessor flow shop scheduling with deteriorating maintenance in cleaner production," *Journal of Cleaner Production*, vol. 135, pp. 276–283, 2016.

[33] A. Ladj, C. Varnier, and F. B.-S. Tayeb, "IPro-GA: an integrated prognostic based GA for scheduling jobs and predictive maintenance in a single multifunctional machine," *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 1821–1826, 2016.

[34] L. Xiao, S. Song, X. Chen, and D. W. Coit, "Joint optimization of production scheduling and machine group preventive maintenance," *Reliability Engineering & System Safety*, vol. 146, pp. 68–78, 2016.

[35] H. Golpîra and E. B. Tirkolaee, "Stable maintenance tasks scheduling: a bi-objective robust optimization model," *Computers & Industrial Engineering*, vol. 137, Article ID 106007, 2019.

[36] A. Goli, E. B. Tirkolaee, and M. Soltani, "A robust just-in-time flow shop scheduling problem with outsourcing option on subcontractors," *Production & Manufacturing Research*, vol. 7, no. 7, pp. 294–315, 2019.

[37] C. R. Cassady and E. Kutanoglu, "Integrating preventive maintenance planning and production scheduling for a single machine," *IEEE Transactions on Reliability*, vol. 54, no. 2, pp. 304–309, 2005.

[38] Z. Q. Lu, Y. J. Zhang, and X. L. Han, "Integrating run-based preventive maintenance into the capacitated lot sizing problem with reliability constraint," *International Journal of Production Research*, vol. 51, no. 5, pp. 1379–1391, 2015.

[39] S. Batun and M. Azizoğlu, "Single machine scheduling with preventive maintenances," *International Journal of Production Research*, vol. 47, no. 7, pp. 1753–1771, 2009.

[40] N. Zribi, A. El Kamel, and P. Borne, "Minimizing the makespan for the MPM job-shop with availability constraints," *International Journal of Production Economics*, vol. 112, no. 1, pp. 151–160, 2008.

[41] S. Gurel and M. Selim Akturk, "Scheduling preventive maintenance on a single CNC machine," *International Journal of Production Research*, vol. 46, no. 24, pp. 6797–6821, 2008.

[42] M. Leung, Y. Leung, and A. Chan, *Carbon Audit Toolkit for Small & Medium Enterprises in Hong Kong*, The University of Hong Kong, Hong Kong, China, 2010.

[43] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[44] J. Dong and C. Ye, "Research on collaborative optimization of green manufacturing in semiconductor wafer distributed heterogeneous factory," *Applied Sciences*, vol. 9, no. 14, p. 2879, 2019.

[45] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: a bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.

[46] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.