IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 April 2023, accepted 13 May 2023, date of publication 16 May 2023, date of current version 23 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3276871

RESEARCH ARTICLE

Improved Multiverse Optimization Algorithm for Fuzzy Flexible Job-Shop Scheduling Problem

JIN-CHENG FANG[1], A-FENG ZENG[1,2], SHAO-FENG ZHENG[2], WEN-DI ZHAO[2], (Member, IEEE), AND XU HE[2]

[1]College of Management, Fujian University of Technology, Fuzhou, Fujian 350118, China
[2]Intelligent Transportation System Research Center, College of Transportation, Fujian University of Technology, Fuzhou 350118, China

Corresponding author: A-Feng Zeng (13067101183@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61976055, in part by the Special Fund of Education and Scientific Research of Fujian Provincial Department of Finance under Grant GY-Z21001, in part by the Leading Project of the Fujian Provincial Department of Science and Technology under Grant 2021H0025, and in part by the Major Project of Fujian Social Science Foundation under Grant FJ2021Z022.

ABSTRACT An improved multiverse optimization (IMVO) algorithm is proposed herein for the fuzzy flexible job-shop scheduling problem pertaining to the non-deterministic polynomial-time hard (NP-hard) problem. First, we designed a hybrid initialization method to improve the quality of the initial solution, and thereafter, introduced a self-crossing technique along with an insert-based heuristic algorithm to simulate the process of exchanging objects between black/white holes and wormholes, respectively. Second, a universe selection mechanism is proposed to reduce the possibility of the algorithm falling into a local optimum. Third, four kinds of neighborhood structures were designed to improve the local search ability of the algorithm. In the decoding operation, we adopted the shift-left strategy to completely utilize the idle time of the machine. Finally, numerous experiments were conducted on the three benchmark test sets of various types and sizes to investigate the performance of the proposed IMVO algorithm. The experimental results demonstrated the effectiveness of the algorithm, especially in large-scale instances, displaying a strong superiority with an average maximum enhancement efficiency of 50.44%.

INDEX TERMS Fuzzy flexible job-shop scheduling problem, multiverse optimization algorithm, variable neighborhood search, self-crossing technique, universe selection mechanism.

I. INTRODUCTION

Recent research on manufacturing systems is focused on the job-shop scheduling problem (JSP), as it is one of the most challenging problems to address in theoretical research. Excellent scheduling strategies are crucial for improving the optimality of production systems and their economic benefits [1], [2]. Among these, the flexible job-shop scheduling problem (FJSP) is a typical NP-hard problem that allows each operation to be processed on any machine. Although FJSP offers the flexibility of production processes compared to traditional JSP, the uncertainties related to the actual production process (*e.g.*, machine failure, workpiece transportation constraints, and personnel transfer) hinder the reliability of processing time. Thus, the development of a new algorithm that closely represents the actual production scenarios of enterprises bears strong practical significance for resolving the issues of uncertain production and processing time of products in FJSP.

In relevant research, the current mainstream method utilizes fuzzy theory to optimize the uncertainty of operation processing time, which is referred to as the fuzzy flexible job-shop scheduling problem (FFJSP). If the fuzzy processing time in this problem is used directly as an input quantity, it may negatively impact the production planning and scheduling, which ultimately affects the efficiency and quality of the entire production process. Therefore, adopting the triangular fuzzy method to convert the fuzzy target values into definite values can better reflect the actual situation, and thus, improve the efficiency and quality of production.

The associate editor coordinating the review of this manuscript and approving it for publication was Wanqing Zhao.

Jha et al. have successfully applied this method to practical production scheduling, demonstrating its effectiveness [3]. Recently, owing to the complexity of the FFJSP, scholars are primarily focusing on solving algorithms and performing multi-objective optimization [4]. Among them, Sakawa et al. adopted a triangular fuzzy number with a genetic algorithm (GA) to represent the uncertain operation processing time for solving the FFJSP [5]. Liu et al. adopted a series of transformation processes to simplify the dynamic FFJSP into a traditional static FFJSP and used an improved estimation distribution algorithm (EDA) to solve this problem [6]. Qin et al. proposed an improved iterated greedy algorithm by employing a heuristic algorithm to generate the initial solution and designing two classes of combined simulated annealing algorithms to ensure the performance of the algorithm. [7]. Gao et al. proposed a two-stage artificial bee colony algorithm to resolve the problems of fuzzy processing time and the constraints of inserting new jobs in FJSP [8]. Huang et al. utilized six heuristic rules to initialize the solution and proposed an improved discrete particle swarm optimization (IDPSO) to solve the FFJSP [9]. Liu et al. introduced the logistic chaotic mapping model and heuristic rules in a hybrid GA to avoid the algorithm from falling into the local optimal [10]. Zhong et al. improved the artificial crowd algorithm based on the local search operator and crossover operator of variable neighborhood search to obtain an adequately high search performance of the FFJSP [11]. Li et al. improved the artificial immune system algorithm to solve the FFJSP using asymmetric triangular interval values to characterize the processing time of each workpiece [12]. Li et al. propose a flexible job shop scheduling method that utilizes a self-adaptive multi-objective evolutionary algorithm with fuzzy processing time as a distinguishing feature. This method effectively addresses the flexible job shop scheduling problem while optimizing multiple objectives, thus enhancing scheduling flexibility and reliability [13]. J.C. et al. propose a solution method for FFJSP, which combines global neighborhood search with a hill-climbing algorithm and effectively mitigates the issue of local optima. The proposed method demonstrated promising performance in experiments [14].

In 2016, inspired by the physical multiverse theory, Mirjalili et al. proposed a new swarm intelligence optimization algorithm called multiverse optimization (MVO) [15]. As demonstrated, the MVO algorithm converges faster than conventional metaheuristic algorithms such as gray wolf optimizer, particle swarm optimization (PSO), GA, and gravitational search (GS) algorithm. This algorithm offers the advantages of a simple framework, few controlled parameters, self-organization, and self-adaptability [16]. Accordingly, reasonable results have been achieved in the optimal parameter configuration of proton exchange membrane fuel cells, optimization of support vector machine tuner parameters, network reconstruction, 3D flight path planning of unmanned aerial vehicles, and prediction of oil consumption [17], [18], [19], [20], [21], [22], [23]. Initially, Liu et al. applied the MVO algorithm to solve the FFJSP [24]. To solve the FFJSP, they simulated the object exchange process of black/white holes and wormholes using the path relink technology and plug-in heuristic algorithm, which yielded satisfactory results. As discussed, the MVO algorithm offers diverse applicability with superiority for solving various optimization problems including the FFJSP.

In the existing research, the algorithms solving FFJSP primarily synthesized the excellent characteristics of various algorithms and incorporated certain perturbation operators to enhance the exploration and developmental abilities of the algorithms. Although the population initialization fundamentally adopted random or chaotic initialization, the initial population did not exhibit a particularly high-quality. Moreover, the results indicated that the current algorithm easily falls into the local optimum as the problem size expands.

Thus, to alleviate the inadequate performance of the current algorithm and low initial population quality for solving large-scale FFJSP, this study proposed an improved multiverse optimization algorithm (IMVO) by minimizing the maximum fuzzy completion time. First, chaotic and greedy algorithms were used to initialize the population and ensure the diversity and quality of the population. Second, this research introduced a self-crossover technology and an insert-based heuristic algorithm to simulate the process of exchanging objects between the black/white holes and wormholes, respectively, for improving the global search capability of the algorithm. Thereafter, four kinds of neighborhood structures were used for variable neighborhood searches to enhance the local search ability. In addition, this research proposes a universe selection mechanism that facilitates rapid convergence of the algorithm and reduces its possibility of falling into the local optimal. Furthermore, this research employed the shift-left strategy in the decoding process to completely utilize the idle time on the machine and effectively reduce the maximum fuzzy completion period. Finally, this research simulated three distinct sizes and types of groups of fuzzy job-shop scheduling examples to verify the performance and superiority of the proposed algorithm in comparison with the existing advanced algorithms. The contributions of this research are stated as follows:

(a) The quality of the initial population was improved by combining a chaotic algorithm with a greedy algorithm to initialize the population.

(b) The process of exchanging objects between black/white holes and wormholes in the MVO algorithm was simulated by two heuristic algorithms.

(c) Three sets of distinguished scales and types of FFJSP examples were set up for the simulation experiments to verify the effectiveness of the IMVO algorithm.

II. FUZZY FLEXIBLE JOB-SHOP SCHEDULING PROBLEM

A. PROBLEM DESCRIPTION AND HYPOTHESIS

The FFJSP is described as follows:n workpieces to be processed on m machines; each workpiece includes one or more operations, and the sequence of operations is determined in advance; each operation can be processed on multiple machines with various processing times on each machine; simultaneously, the processing time of a given machine is uncertain, and this uncertainty is expressed using a fuzzy number. The major objective of the FFJSP is to select the most suitable machine for each operation and determine the most optimal processing routes along with the corresponding start times of all processes on each machine to minimize the maximum completion time of the entire system. Similar to the FJSP, the FFJSP is essentially composed of two subproblems, namely, the sorting problem of the process and the selection of the machine. The problem hypothesis of the FFJSP can be described as follows:

1) There are n workpieces, which are not connected. The i-th workpiece is represented asO_i.

2) Each workpiece includes $e_i$several intermediate operations, including a sequence constraint between the operations of the same workpiece, whereas no sequence constraint existed between the operations of multiple workpieces.

3) The$j$-th operation of the i-th workpiece is represented as O_{ij}.

4) There are m processing machines that can process the workpiece.

5) For each operation, a set of machines M_{ij} that can process the operations.

6) The processing time of each operation on machine k is ambiguous.

7) The workpiece must not be interrupted during processing on a machine.

8) Each processing machine can handle only one workpiece at a time.

9) The machine may be intermittently out of operation owing to periodic maintenance. The symbols and parameters used for model development are defined in Table 1.

B. ESTABLISHMENT OF THE FFJSP MATHEMATICAL MODEL

The FFJSP aims to minimize the maximum fuzzy completion time, and its objective function is expressed in Equation (1):

$$\begin{aligned} C_{max} &= \min\{\max \sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{e_i} (s_{ijk} + t_{ijk})\} \\ &= min\{\max \sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{e_i} (c_{ij})\}_{max} \end{aligned} \tag{1}$$

The constraints of an FFJSP are shown in Formulas (2–7):

$$t_{ij} \geq 0, s_{ij} \geq 0 \quad i = 1, 2, 3 \ldots n; j = 1, 2, 3 \ldots e_i \tag{2}$$

$$\begin{aligned} &s_{ij} + t_{ij} \leq s_{i(j+1)} \quad i = 1, 2, 3 \cdots n; j = 1, 2, 3 \cdots e_i \\ &\qquad s_{ij} + t_{ijk} \leq s_{lh} + \inf * \left(1 - y_{ijlhk}\right) \qquad (3) \\ &\quad i = 1, 2, 3 \cdots n;\ l = 1, 2, 3 \cdots n; \\ &\qquad j = 1, 2, 3 \cdots e_i;\ t = 1, 2, 3 \cdots e_l; \\ &\qquad k = 1, 2, 3 \cdots m; \end{aligned} \tag{4}$$

$$\sum\nolimits_{k=1}^{M_{ij}} x_{ijk} = 1 \quad i = 1, 2, 3 \cdots n;\ j = 1, 2, 3 \cdots e_i \tag{5}$$

$$x_{ijk} = \begin{cases} 1, & \textit{If operation } O_{ij} \textit{ selects } M_k \\ 0, & \textit{Otherwise} \end{cases} \tag{6}$$

$$y_{ijlhk} = \begin{cases} 1, & \textit{If} O_{ij} \textit{is processed before } O_{lh} \textit{at} M_k \\ 0, & \textit{Otherwise}, \end{cases} \tag{7}$$

where s_{ij} represents operation O_{ij} at the start time of processing, e_i denotes the total number of operations of the workpiece J_i, *inf* indicates an infinite positive number, and M_{ij} represents the set of optional processing machines for operation O_{ij}. Constraint relation (2) indicates that the processing time of the operation must be greater than 0; constraint relation (3) represents the sequence between the operations of each workpiece; constraint relation (4) indicates that a given machine can process only one workpiece at an instant; constraint relation (5) implies that a given workpiece can be processed only by one machine at an instant.

An exemplary instance of a three-workpiece, three-machine FFJSP is presented in Table 2, wherein the rows and columns in the table represent operations and machines, respectively. The entries in the table denote the fuzzy time for each operation to be processed on the corresponding machine. O_{11} corresponds to the first operation of the first workpiece.

C. OPERATION OF TRIANGULAR FUZZY NUMBER

As the processing time in the FFJSP problem is represented as a triangular fuzzy number (TFN), the method of evaluating this representation method differs from that of the general processing time. The operational rules of TFNs primarily include the addition operation, approximate max operation, and ranking operation. In principle, the addition operation is applied to evaluate the fuzzy makespan of operation, whereas the approximate max operation is utilized to determine the fuzzy beginning time of operation. The ranking operation is used to sort TFNs to obtain the maximum fuzzy makespan. For two arbitrary TFNs, *e.g.*, $X\ (x_1, x_2, x_3)$ and $Y\ (y_1, y_2, y_3)$, the aforementioned three operation rules can be expressed as follows:

Addition operation: $X + \mathrm{Y} = (x_1, x_2, x_3) + (y_1, y_2, y_3) = (x_1 + y_1, x_2 + y_2, x_3 + y_3)$;

Ranking operation:

1) Compare the values of $F_1(X) = \frac{x_1+2x_2+x_3}{4}$ and $F_1(Y) = \frac{y_1+2y_2+y_3}{4}$. If $F_1(X) > F_1(Y)$, then $X > Y$.

2) If $F_1(X) = F_1(Y)$, compare $F_2(X) = x_2$ with $F_2(Y) = y_2$. If $F_2(X) > F_2(Y)$, then $X > Y$.

3) IfF_1 andF_2 are equal, compare $F_3(X) = x_3 - x_1$ and$F_3(Y) = y_3 - y_1$. If $F_3(X) > F_3(Y)$, then $X > Y$.

Consider the approximate max operation: if$X > Y$, then$X \vee Y = X$; otherwise, $X \vee Y = Y$.

III. IMPROVED MVO ALGORITHM SOLUTION DESIGN

A. MULTIVERSE OPTIMIZATION ALGORITHM

In 2016, Mirjalili proposed the MVO algorithm [15] that stimulates the transfer of matter in the universe from white

TABLE 1. Meaning of sym.

Symbol	Meaning	Symbol	Meaning
J	Workpiece set	M	The machine set for processing a workpiece
m	Number of processing machines	n	Total number of workpieces
M_k	k-th machine	O_{lh}	h-th operation of l-th workpiece
O_{ij}	j-th operation of i-th workpiece	J_i	i-th workpiece
s_{ij}	Fuzzy start time of operation O_{ij}	t_{ijk}	Fuzzy processing time of operation O_{ij} on machine k
e_i	Number of operations for workpiece i	M_{ij}	Machinable machine set of j-th operation of i-th workpiece
C_{max}	Maximum fuzzy completion period	D	Total number of operations
c_{ij}	Completion time of operation O_{ij}	inf	Infinity
x_{ijk}	Boolean value (0,1)	y_{ijhk}	Boolean value (0,1)

TABLE 2. Instances of scale 3 × 3 completely FFJSP.

Operations	Machines		
	M_1	M_2	M_3
O_{11}	(4,10,12)	(7,10,13)	(6,11,14)
O_{12}	(5,8,10)	(6,7,9)	(7,9,11)
O_{13}	(6,9,10)	(3,7,8)	(2,3,4)
O_{21}	(5,7,10)	(9,13,16)	(2,5,6)
O_{22}	(5,7,11)	(8,13,17)	(9,12,19)
O_{31}	(4,7,10)	(3,6,8)	(2,7,10)
O_{32}	(6,7,8)	(4,6,9)	(3,4,5)

holes to black holes through wormholes. The MVO algorithm involves relatively few performance parameters, fundamentally including the wormhole existence probability and wormhole travel distance rate. In principle, the low-dimensional numerical experiments deliver relatively excellent performances. Similar to other swarm intelligence optimization algorithms, the optimization execution process of the MVO algorithm is segmented into two stages: exploration and development. White holes and black holes are used for exploration, whereas wormholes are used for development. The MVO algorithm obeys the following rules in the optimization process:

1) A higher inflation rate of the universe increases the likelihood of a greater number of white holes.

2) Conversely, if the inflation rate of a universe is relatively low, black holes are more likely to be formed.

3) The universe that created white holes will repel objects.

4) Conversely, the universe that created black holes will absorb objects.

5) Regardless of the inflation rate, other universes can transfer objects to the current optimal universe through wormholes.

The MVO algorithm creates the initial universe iterative loop based on the principle of black/white holes and wormholes. Specifically, the universe represents the feasible solution to a problem, the objects in the universe represent the components of the solution, and the inflation rate of the universe represents the fitness value of the solution. The mathematical model of the algorithm is stated as follows.

The object composition of the algorithm is stated as follows:

$$U = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^d \\ x_2^1 & x_2^2 & \cdots & x_2^d \\ \vdots & \vdots & \vdots & \vdots \\ x_n^1 & x_n^2 & \cdots & x_n^d \end{bmatrix}, \tag{8}$$

where d denotes the number of objects (variables) and n represents the number of universes (candidate solutions).

Since each universe manifests a unique inflation rate, the objects in the universe will migrate through the orbits of the white/black holes. This process follows a roulette wheel mechanism, as expressed in Equation (9).

$$x_i^j = \begin{cases} x_k^j & r_1 < NI\,(Ui) \\ x_i^j & r_1 \geq NI\,(Ui) \end{cases}, \tag{9}$$

where x_i^j denotes the position of the j-th object black hole in the i-th universe, U_i indicates the i-th universe, and $NI\,(Ui)$ represents the standard inflation rate of the i-th medium universe; x_k^jdenotes the position of the j-th object of the k-th universe generated by the roulette mechanism.

Regardless of the inflation rate, the local variations can be achieved and their inflation rates can be improved if the universe objects stimulate the internal objects to travel to the current optimal universe. The process of updating the object positions of wormholes in the universe follows Equation (10), as shown at the bottom of the next page, where the travel distance rate (TDR) is a dynamic parameter; H is the threshold, and the empirical value of H = 0.5 is taken here; $r_2, r_3, andr_4$ are random numbers in the interval [0,1]; lb and ub represent the lower and upper limits of the variables; $Bestx_i^j$ represents the location of the wormhole corresponding to the current optimal universe. After iteration, the optimal location is constantly updated. When x_{i+1}^j, a new optimal universe wormhole location is generated, and the universe inflation rate between x_{i+1}^j andx_i^j will be compared. If the fitness value of x_{i+1}^j is better than that of x_i^j, x_{i+1}^j will replace x_i^j; otherwise, x_i^j will be preserved for the next generation.

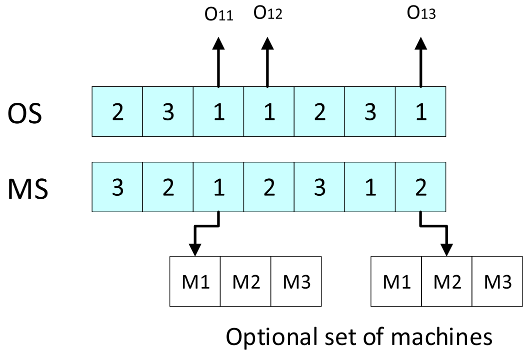

FIGURE 1. **Coding and decoding schematic figures.**

The expression of the wormhole existence rate (WEP) is as follows:

$$WEP = WEP\frac{l}{L}\left(WEPmin_{max}\right)_{min} \tag{11}$$

where WEP_{max} and WEP_{min} presume the empirical values $WEP_{max} = 1$ and $WEP_{min} = 0.2$, respectively, $l$denotes the current iteration number, and L represents the maximum iteration number. Moreover, the TDR can be expressed as follows:

$$TDR = 1 - \left(\frac{l}{L}\right)^{\frac{1}{p}} \tag{12}$$

where $p$indicates the precision of the mining capacity.

This study uses the improved MVO algorithm to resolve the FFJSP, where each universe corresponds to a solution and each object in the universe corresponds to an operation.

B. ENCODING AND DECODING

The FFJSPs consider operation sequencing and machine selection. Therefore, we adopted two coding methods, namely, operation sequence (OS) and machine selection (MS) coding. The OS coding is used to determine the order, where the encoding algorithm represents the workpiece. In contrast, the number of occurrences$j$ indicates the OS of the i-th workpiece and O_{ij} represents the corresponding code. In particular, the MS coding selects the processing machines in sequence according to the OS, and in this process, the MS code value indicates the corresponding serial number of the machine from the set of machinable machines, *i.e.*, M_{ijk}. The schematic processes of coding and decoding are displayed in Fig. 1.

An effective decoding scheme is essential for further improving the scheduling quality. The priority constraint relationship between the operations of the same workpiece can delay the completion of the prior operation during processing, which causes an idle period for the machine. To this end, Gao et al. proposed a shift-left strategy, which utilizes the idle time of the machine by shifting the operation toward the left. This scheme is highly effective in improving scheduling efficiency [25], and its principle is stated below.

Assuming an idle time $\left[t_k^S, t_k^E\right]$ for a machine, the operation O_{ij} performing the insertion operation in the idle time of the machine satisfy meet two conditions:

First, the operation O_{ij} is processed $\left[t_k^S, t_k^E\right]$ in the idle period of the machine M_k, and the completion time$newc_{ij}$ of the operation O_{ij} inserting machine M_k is less than the end time of the machine M_k during this period t_k^E. Based on the assumption of the FFJSP, the operation O_{ij} can be processed only after completing the previous operation $O_{i,j-1}$. If O_{ij} involves no preceding operation, then $C_{i,j-1} = 0$. Therefore, the start time of the operation O_{ij} in machineM_k processing $news_{ij}$ can be expressed as

$$news_{ij} = \max\left\{t_k^S, c_{i,j-1}\right\}. \tag{13}$$

The constraint on inserting the operation into the machine M_k processing can be expressed as follows:

$$t_k^E > newc_{ij} + t_{ijk}. \tag{14}$$

Second, to ensure that the idle time processing of the operation O_{ij}inserted in machine M_k can promote its completion time, the condition that the completion time of the operation O_{ij} after the insertion of $newc_{ij}$ should be less than the original completion time c_{ij}, *i.e.*, $newc_{ij} < c_{ij}$.

C. UNIVERSE INITIALIZATION

In solving multivariate problems with heuristic algorithms, the mass of the initial universe often poses considerable influence on the convergence rate of the universe as well as the final result. Thus, the generation of a high-quality initial universe is imminent. In this study, we adopted chaotic initialization for the OS [26]. Chaos is a form of motion in a nonlinear dynamic system, characterized by randomness, ergodicity, and sensitivity to initial values [27]. Using chaotic initialization, objects of the initial universe can be distributed discretely in the solution space, which effectively improves the diversity of the universe. Among several chaotic systems, logistic chaos mapping is the most typical chaotic system, explained by the following system equation:

$$X(k+1) = u \times X(k) \times [1 - X(k)]. \tag{15}$$

To increase the quality of the initialized universe and ensure the diversity of the universe, we selected a certain proportion R of the chaotic initialized universe. Moreover, greedy

$$x_{i+1}^j = \begin{cases} \begin{cases} \mathrm{Best}_i^j + TDR * \left(\left(ub_j - lb_j\right) * r_4 + lb_j\right) & r_3 < H \\ \mathrm{Best}_i^j - TDR * \left(\left(ub_j - lb_j\right) * r_4 + lb_j\right) & r_3 \geq H \end{cases} & r_2 < WEP \\ x_i^j & r_2 \geq WEP \end{cases} \tag{10}$$

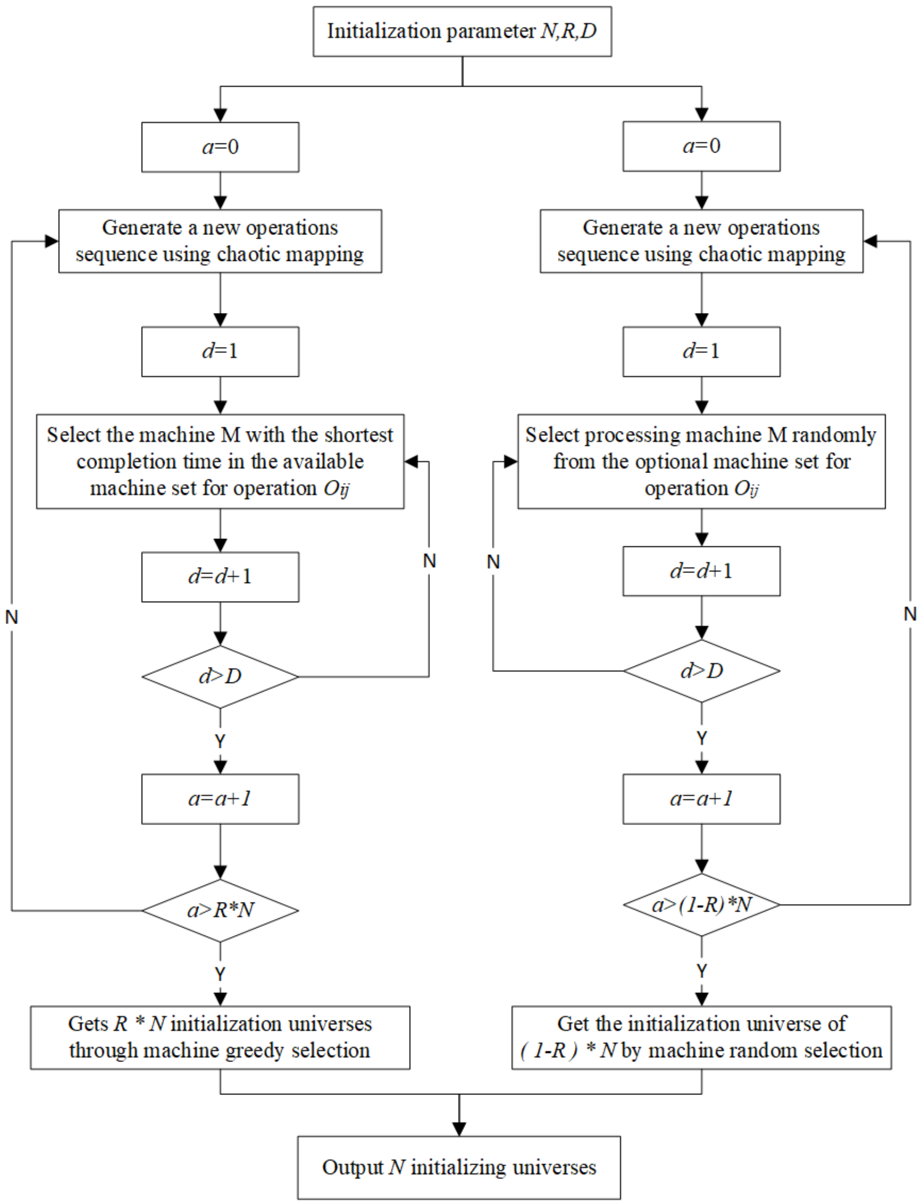

FIGURE 2. Hybrid initialization universe flow.

selection was adopted in the MS, and random selection was applied in the MS of the remaining chaotic initial universe. The principle of greedy selection is to select the machine with the shortest completion duration for each operation under the process sequence generated by chaotic initialization [28].

Considering the instance of FFJSP listed in Table 2 as an example. The concrete implementation steps of this hybrid proportion initialization method were combined with chaos mapping and greedy thought, stated as follows:

Step 1: Random generation of chaotic variables equal to the total working number, as listed in Table 3. In total, seven ordinal artifacts are presented in Table 3, and thus, seven chaotic variables were randomly generated. The sequence of the chaotic variables corresponds to that of the procedure code.

Step 2: The sequence of the chaotic variables was arranged in ascending order, and the operation code corresponding to the chaotic variables was moved accordingly. For instance, chaotic variable 0.78 corresponds to operation code 1 of position 2, but the variable 0.78 occupied the sixth position in ascending order; thus, process code 1 was rearranged to the sixth place.

This work selects $N \times R$ initial universe for MS by the greedy selection, calculates the completion time of the operation on all its optional machines, and selects the machine with the shortest processing period. The MS of the remaining

TABLE 3. The procedure of initial operation sequence.

Positions	1	2	3	4	5	6	7
Process sequence	O_{11}	O_{12}	O_{13}	O_{21}	O_{22}	O_{31}	O_{32}
Procedure code	1	1	1	2	2	3	3
The sequence of chaotic variables $X(n)$	0.67	0.80	0.11	0.37	0.51	0.20	0.88
$X(n)$ in ascending order	0.11	0.20	0.37	0.51	0.67	0.80	0.88
Procedure code after change	1	3	2	2	1	1	3
Modified process sequence	O_{11}	O_{31}	O_{21}	O_{22}	O_{12}	O_{13}	O_{32}

initial universe was conducted by randomly selecting one of the alternative machines corresponding to the operation.

The flow of the hybrid initialization methods is illustrated in Fig. 2.

D. SIMULATION OF BLACK/WHITE HOLES AND WORMHOLES FOR OBJECT EXCHANG

1) Self-crossing black/white hole object exchange technology.

Liu et al. reported that the traditional parental GA focused only on the effectiveness of the combination between various chromosomes and did not study the characteristics of the chromosomes themselves. Upon introducing the crossover operation of the parthenogenetic algorithm, the results revealed that the combination of the two could effectively improve the efficiency of the GA, known as self-cross GA [29]. Inspired by this aspect of self-crossing, we propose the black/white hole exchange technology herein. Assuming two universes, U_I and U_E, that generate black holes and white holes, respectively, the process of self-crossing of the black/white hole exchange technology between the two universes is described as follows.

First, the object position in U_I corresponding to the occurrence of the black hole is determined. Second, the position of the corresponding operation of the object in U_E, *i.e.*, the position of the white hole is determined through decoding and coding operations. Thereafter, in-universe U_I, the object at the black hole position and that at the corresponding position of the white hole in U_I were exchanged to generate a new transition universe, and we implemented the self-crossing technology proposed in this study. Furthermore, if the end condition is not satisfied, the subsequent location of the black hole in the U_I id determined, and the previous steps are repeated. Ultimately, the best intermediate universe is observed by comparing the expansion rate of the universe after all exchanges. In addition, we provide an example to illustrate the operation process of black/white hole object exchange based on the self-crossing technology depicted in Fig. 3.

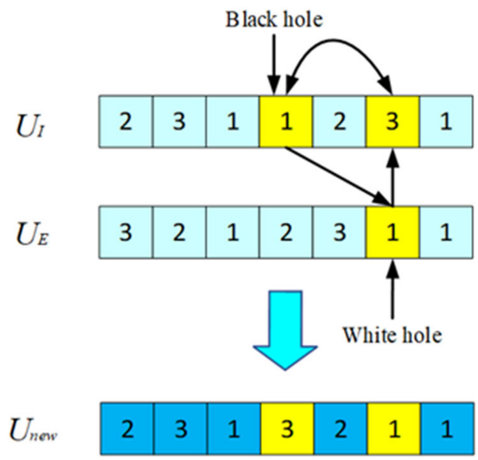

FIGURE 3. Black hole/white hole mass-exchange process based on self-crossing technique.

2) Object wormhole movement technology based on insertion.

Updating the object positions in the universe through wormholes is beneficial because the object in the universe stimulates the movement of the internal objects to the corresponding object positions in the current optimal universe to achieve local variations and improve the inflation rate. Based on insertion, Ruiz and Stutzle constructed an iterative greedy algorithm using a heuristic algorithm, which is an effective method [30], [31], [32], [33]. Inspired by this algorithm, the IMVO algorithm uses the local insertion search to simulate the movement of objects through the wormhole for generating a new solution.

In this study, the coded serial number a was introduced to locate the object position, and Equation (10) can be transformed to (16) *via* serial number transformation. Upon introducing the mobile step Δ (= $\mathrm{TDR}\times((\mathrm{D}-1)\times \mathrm{r}_4+1)$), we established the object wormhole in the optimal space location moving a step length. For instance, $\Delta = -1$ indicates that the object is displaced by one unit toward the left and inserted from the optimal universe object position. Conversely, $\Delta = 1$ indicates that it moves one unit toward the right and inserts that object position; $\Delta = 0$ implies that the wormhole object insertion position is the optimal universe object position. Based on the stated operations, the process of moving universe objects to the current optimal universe through wormholes is extensively simulated to identify a superior universe. To clearly explain this process, the simulation of wormhole object movement based on the insertion technique is depicted in Fig. 4, as in (16), shown at the bottom of the next page. where $Bestx_i^j$ represents the sequence number of the wormhole position corresponding to the current optimal universe.

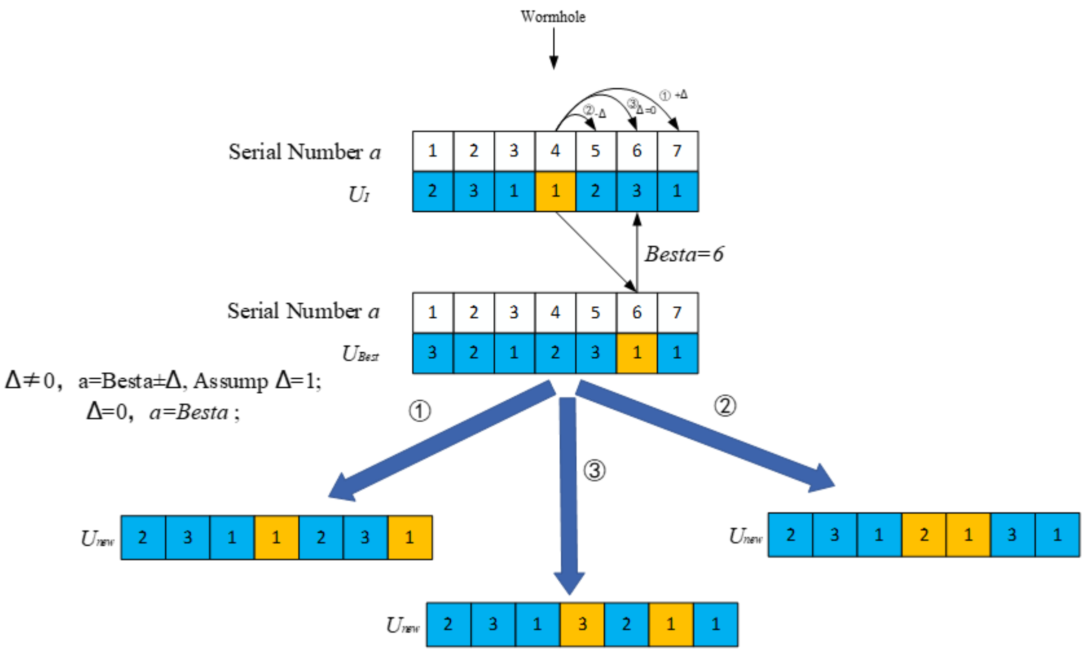

FIGURE 4. The simulation process is based on plug-in wormhole object movement technology.

E. UNIVERSE SELECTION MECHANISM

Wu et al. applied the ''elitist selection or elitism'' strategy to solve FJSP, which has been proven to be effective and feasible for algorithm optimization [34]. The concept of this strategy is to directly replicate the best objects of the population emerging from the evolutionary process in the subsequent generation. This approach preserves the best objects of the current population in the next generation, thereby limiting the algorithm to not converge to the global optimal solution.

Inspired by this, the present study improves this strategy and applies it in the universe update iteration. Through the mechanism of the black hole, white hole, and wormhole, a new universe $newU_i$ is obtained by exchanging the object of the universe U_i. If the inflation rate of the new universe $NI\ (newU_i)$ is greater than or equal to the inflation rate $NI\ (U_{best})$ of the existing optimal universe, the new universe is preserved. If the inflation rate of the new universe is less than the inflation rate of the existing optimal universe, the size of the random number and the relative inflation rate $RNI\ (U_i)$ are assessed. If the random number is greater than or equal to the relative expansion rate, the new universe will be reserved. Otherwise, the original universe will be retained.

The relative inflation rate is the ratio of the original universe inflation rate to the sum of the original and newer universe inflation rates. The mathematical discriminant of retaining the new universe in the subsequent iteration is expressed in Equations (17) and (18), as shown at the bottom of the next page, where r_5 represents a random number [0,1].

F. VARIABLE NEIGHBORHOOD SEARCH STRATEGY

In the case of using a swarm intelligence optimization algorithm to solve combinatorial optimization problems, the algorithm can easily fall into a local optimum. However, the variable neighborhood search (VNS) algorithm—a meta-heuristic algorithm proposed by Mladenovic in 1997—can improve this limitation. The basic concept of VNS is to expand the search scope by systematically altering the neighborhood structure set in the search process to continuously obtain the local optimal solution while increasing the neighborhood scope to prevent the search process from falling into the local optimal [35]. In principle, the VNS is a local search based on a neighborhood structure set instead of a single neighborhood, and therefore, it is more reasonable and effective. Owing to its positive local search ability, it has been successfully applied to solve JSPs. By moving, exchanging, inserting, and reversing an object, a better universe can be discovered near the current universe, and the local search ability of the algorithm can be improved. Adjusting the key operation

$$x_{i+1}^{j} = \begin{cases} \begin{cases} \mathrm{Best}_i^j + TDR * \left(\left(ub_j - lb_j \right) * r_4 + lb_j \right) & r_3 < H \\ \mathrm{Best}_i^j - TDR * \left(\left(ub_j - lb_j \right) * r_4 + lb_j \right) & r_3 \geq H \end{cases} & r_2 < WEP \\ x_i^j & r_2 \geq WEP \end{cases} \tag{16}$$

instead of any other operation in the NS can reduce the blindness of the search and remarkably improve the probability of yielding an improved solution. The longest path from the start point to the endpoint in the disjunctive graph corresponding to the scheduling scheme is called the pivotal path, and the operation constituting the pivotal path is the key operation O_h, which directly determines the maximum completion time of the scheduling scheme [36]. Accordingly, two neighborhood structures N1 and N2 along with two random neighborhood structures N3 and N4 were designed.

1) Neighborhood structure N1: Traverses every key operation in the universe and adjusts the key operation O_h to the idle time between other operations associated with the

same processing machine to maximize the compression of the maximum completion time.

2) Neighborhood structure N2: Traverses the idle time of each key operation in the pivotal path on other optional machines, excluding the current machine, and attempts to move O_h to another machine with idle time for processing to reduce the maximum completion time.

3) Neighborhood structure N3: Randomly selects an operation and exchanges this operation with the following operation. Reselects the operation if it is the terminal of the sequence.

4) Neighborhood structure N4: An operation is randomly selected as U_i and extracted to form U_i^*. Thereafter, the extracted operation is inserted into all positions at which U_i^* can be inserted to form multiple universes, from which the optimal one is selected.

The aforementioned operations are sequentially conducted. Upon adopting the new universe formed by the neighborhood structure N1, the selection is reserved according to the universe selection mechanism proposed herein, and the neighborhood structure N1 is used until the current universe cannot be adjusted and the search is relocated to another neighborhood.

G. SOLVING FFJSP BY IMPROVED MVO ALGORITHM

In this study, in the case of solving the FFJSP with the improved MVO method, the universe is initialized by hybrid initialization to improve the diversity and quality of the initial universe. Specifically, the coding of the universe is expressed in the traditional two-layer form. The shift-left strategy adopted in the decoding process can improve the decoding performance. To improve the local search capability of the algorithm, the VNS is introduced with four kinds of neighborhood operations to discover the optimal universe near the current universe. In the iterative process, the universe selection mechanism proposed herein was used to determine the next-generation multiverse, which ensured that the algorithm satisfied the global search ability and accelerated the convergence of the multiverse. The basic steps involved in solving the FFJSP by the improved MVO algorithm are stated as follows.

Step 1: Set parameters, including the number of universes, maximum number of iterations L, greedy initialization ratio R, the minimum and maximum probability of wormhole existence WEP_{min} and WEP_{max}, etc.

Step 2: Use the hybrid initialization method to generate the initial universe.

Step 3: Calculate the fitness value of each universe and the standardized universe inflation rate, and sort them to obtain the current optimal universe.

Step 4: Perform the object exchange between a black hole and a white hole; traverse all the objects in each universe; exchange the positions of the black hole and those of the objects in the universe selected by the roulette wheel to generate a new universe, before performing the universe selection mechanism.

Step 5: Perform the wormhole movement mechanism; traverse all objects in each universe; displace the position of the generated wormhole to that of the current optimal universe according to Equation 10; generate a new universe and assess its superiority in comparison to the current optimal universe, before implementing the universe selection mechanism.

Step 6: Perform the VNS and universe selection mechanism.

Step 7: If the maximum number of iterations is attained, the algorithm is terminated and the optimal universe is obtained as the output; otherwise, the process returns to step 3.

H. COMPUTATION COMPLEXITY ANALYSIS

The IMVO algorithm primarily contains two components in the iteration process. The first component is the global search stage which includes the black/white hole stage and the wormhole stage. The complexity of the black/white hole stage is represented as $O\left((D+m)\,n\right)$ and that of the wormhole stage is denoted as $O\left(Dn^2\right)$. The second component is the VNS stage, and its complexity is defined as $O\left(2\left(D+1\right)n\right)$.

Thus, the total computational complexity of each generation can be related as $O((D+m)n + Dn^2 + O(2(D+1)n)$, which can be simplified to $O\left(Dn^2\right)$. According to previous studies [24], [37], the HMVO algorithm and ABCNS

$$RNI\left(U_i\right) = \frac{NI\left(U_i\right)}{NI\left(U_i\right) + NI\left(\text{ new }_i\right)} \tag{17}$$

$$U_i = \begin{cases} \begin{cases} U_i & r_5 < RI\left(U_i\right) \\ newU_i & r_5 \geq RNI\left(U_i\right) \end{cases} & NI\left(\text{ new } U_i\right) < NI\left(U_{\text{best}}\right) \\ \text{new } U_i & NI\left(\text{ new } U_i\right) \geq NI\left(U_{\text{best}}\right) \end{cases} \tag{18}$$

TABLE 4. Basic information of three test instances.

Group	Instance	Is it completely flexible?	Number of jobs	Machine Number	Number of operations
A	1	No	3	3	15
	2	No	4	4	18
	3	No	5	4	20
B	1	Yes	10	10	40
	2	Yes	10	10	40
	3	Yes	10	10	50
	4	Yes	10	10	50
	5	Yes	15	10	80
C	Reman 1	No	5	4	23
	Reman 2	No	8	8	64
	Reman 3	No	10	6	81
	Reman 4	No	10	10	100
	Reman 5	No	15	8	171
	Reman 6	No	15	10	185
	Reman 7	No	20	10	308
	Reman 8	No	20	15	355

algorithm are similar advanced algorithms at present. Their algorithm complexity is denoted as $O\left((P+1/2)\,n^2\right)$ and $O\left((3k+m+10)\,n^2\right)$, respectively. As observed, the magnitude of the complexity of the IMVO algorithm is equivalent to that of the two advanced algorithms, which is acceptable for solving FFJSP.

IV. EXPERIMENTAL RESULTS

This study designed three groups of experiments to verify the effectiveness and superiority of the algorithm proposed for solving FFJSP. The experimental data of Group A is derived from prior research [24] that contains three instances in total. In particular, the instance with the smallest size contained three workpieces, three machines, and in total, 15 operations. The instance with the largest size considered five workpieces, four machines, and in total, 20 operations. The experimental data of Group B were sourced from existing literature as well [38], [39]. The experimental data of Group B included five instances of fully flexible manufacturing engineering, wherein fully flexible indicates that each operation can be processed on any machine. Their sizes range from 10 workpieces, 10 machines, and 40 operations to 15 workpieces, 10 machines, and 80 operations. Similarly, the experimental data of Group C were derived from the existing literature [40]. The experimental data of Group C contained eight instances of incomplete flexible remanufacturing engineering, wherein incomplete flexibility indicates that at least one operation can be processed by using only a portion of the machines. These data range in size from 5 workpieces, 4 machines, and 23 operations to 20 workpieces, 15 machines, and 355 operations. The basic information of the three groups of experimental data are presented in the table below.

This study uses MATLAB programming language and is executed on a computer configured with an AMD Ryzen 7

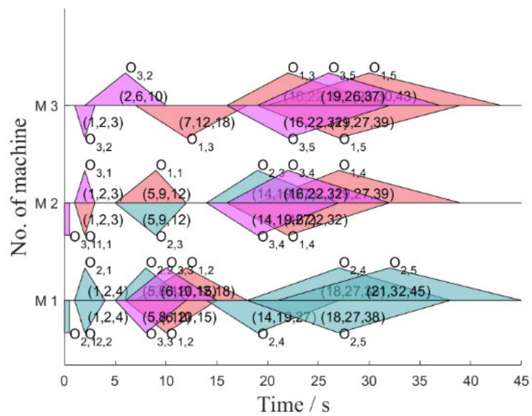

FIGURE 5. **Gantt chart of optimal scheduling scheme for instance 1 of group A.**

5800H with Radeon Graphics CPU@3.20 GHz and 24 GB RAM. Each instance is repeated 30 times, and each experimental result contains the optimal result, average result, worst result, and maximum lift rate sought by the algorithm,

where the maximum lift rate is the maximum ratio of the results obtained by the compared algorithms with respect to that evaluated by the IMVO algorithm in each experimental arithmetic case. To visually review the performance of the algorithm, the fundamental parameters of each group of experimental algorithms are listed in Table 5.

The results of Group A data simulation experiments were compared with the optimal solution obtained by BAB [41] and the results of the HMVO and ABCNS—the two most advanced algorithms at present. The comparison results are listed in Table 6, wherein the IMVO algorithm proposed herein is similar to the two advanced algorithms instanced

TABLE 5. Main parameters of the experimental comparison algorithm for each group.

Group number		Algorithm	Initial population number	Maximum iterations
		IMVO	100	100
A		ABCNS	200	100
		HMVO	200	100
		IMVO	200	100
		ABCNS	200	100
		HMVO	200	500
B		IABC	150	1000
		IDPSO	100	1000
		DHS	150	1000
		TLBO	200	500
		EDA	150	1000
		IMVO	200	100
		ABCNS	200	100
	Instance 1-4	HMVO	100	500
		GS	150	1000
		LS	150	1000
C		MReW	150	1000
		IMVO	200	500
		ABCNS	200	1000
	Instances 5-8	HMVO	100	1000
		GS	150	1000
		LS	150	1000
		MReW	150	1000

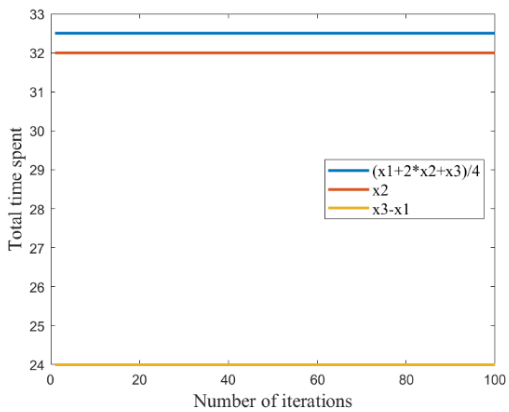

FIGURE 6. Convergence curve of instance 1 of Group A.

earlier, which can determine the best value. The Gantt chart of the optimal solution and its fuzzy processing time $t\,(x1, x2, x3)$ convergence curve for example 1 of Group A is depicted in Figs. 5 and 6, respectively. The convergence curve reflecting the optimal solution during the initialization highlights the effectiveness of the hybrid initialization method. This result completely verifies that the IMVO algorithm proposed herein can rapidly solve the FFJSP with high-quality results under small-scale instances.

To further verify the effectiveness and superiority of the IMVO algorithm for solving the FFJSP, two types of FFJSP were verified for fully flexible and incompletely flexible manufacturing based on relevant examples.

First, to verify the effectiveness and superiority of the IMVO algorithm for the fully flexible FFJSP, the simulations were performed on five fully flexible manufacturing examples with Group B experimental data, and the results of the IMVO algorithm were compared with those of seven advanced algorithms from the published literature, including the ABCNS [37], HMVO algorithm [24], IABC algorithm [10], IDPSO algorithm [41], DHS algorithm [42], TLBO algorithm [43], and EDA [44]. The experimental results of Group B are comparatively presented in Table 7.

In Table 7, the results presented in bold characters represent the best results obtained from the compared algorithms. According to these results, although the worst results of the second and fourth instances of the proposed algorithm were not better than all other algorithms, the optimal and average results of the present algorithm were better than those of other algorithms. This indicates the superior optimization ability of the present algorithm compared to the other comparison algorithms. For the remaining three instances, the proposed algorithm achieved relatively improved results. More importantly, the advantages of the IMVO algorithm become more prominent as the process scale increases and the lifting efficiency soars from 1.75% in Group B instance 1 to 10.53% in instance 5. The Gantt chart of the optimal scheduling result of Group B Example 5 is plotted in Fig. 7.

TABLE 6. Comparison of experimental results of small size instances data in group A.

Instance	Instance Algorithm	Average value	Best Value	Worst value
1	IMVO	(21,32,45)	(21,32,45)	(21,32,45)
	ABCNS	(21,32,45)	(21,32,45)	(21,32,45)
	HMVO	(21,32,45)	(21,32,45)	(21,32,45)
	BAB	—	(21,32,45)	—
2	IMVO	(18,26,35)	(18,26,35)	(18,26,35)
	ABCNS	(18,26,35)	(18,26,35)	(18,26,35)
	HMVO	(18,26,35)	(18,26,35)	(18,26,35)
	BAB	—	(18,26,35)	—
3	IMVO	(13,20,26)	(13,20,26)	(13,20,26)
	ABCNS	(13,20,26)	(13,20,26)	(13,20,26)
	HMVO	(13,20,26)	(13,20,26)	(13,20,26)
	BAB	—	(13,20,26)	—

TABLE 7. Comparison of experimental results in group B.

Instance	Instance Algorithm	Average value	Best Value	Worst value	Maximum lift rate
1	IMVO	**(21.0,28.0,37.0)**	**(21,28,37)**	**(21,28,37)**	1.75%
	ABCNS	(21.0,28.0,37.0)	(21,28,37)	(21,28,37)	
	HMVO	(19.9,28.0,38.1)	(21,28,37)	(19,28,39)	
	IABC	(20.1,29.4,40.3)	(19,28,39)	(22,30,42)	
	IDPSO	(20.60,29.87,40.17)	(19,28,40)	(21,31,41)	
	DHS	(19.9,29.6,40.6)	(17,29,39)	(19,31,44)	
	TLBO	(20.3,29.9,40.9)	(19,28,39)	(21,32,42)	
	EDA	(20.3,30.5,41.6)	(20,28,40)	(22,32,43)	
2	IMVO	**(30.8,44.7,57.4)**	**(32,44,57)**	**(30,45,58)**	2.26%
	ABCNS	(30.0,45.0,58.0)	(30,45,58)	(30,45,58)	
	HMVO	(30.0,45.0,58.0)	(30,45,58)	(30,45,58)	
	IABC	(32.3,46.2,57.3)	(33,45,58)	(35,46,57)	
	IDPSO	(32.47,46.17,57.53)	(30,46,58)	(35,47,57)	
	DHS	(32.1,46.2,57.3)	(30,46,58)	(35,46,57)	
	TLBO	(32.6,46.4,58.5)	(30,45,58)	(36,49,63)	
	EDA	(33.7,46.9,57.9)	(32,46,57)	(34,48,58)	
3	IMVO	**(30.6,42.8,58.0)**	**(31,42,58)**	**(30,43,58)**	5.46%
	ABCNS	(30.0,43.0,58.6)	(30,42,60)	(30,44,58)	
	HMVO	(31.0,43.8,58)	(29,44,53)	(32,44,59)	
	IABC	(31.8,45.8,59.6)	(31,45,57)	(33,47,63)	
	IDPSO	(32.47,46.20,60.90)	(33,45,60)	(35,49,64)	
	DHS	(31.6,45.9,59.9)	(31,45,58)	(33,48,62)	
	TLBO	(31.5,46.7,62.2)	(30,45,60)	(33,50,70)	
	EDA	(32.8,47.2,62.9)	(31,46,60)	(34,49,66)	
4	IMVO	**(22.2,33.0,47.2)**	**(20,33,46)**	**(24,33,47)**	7.69%
	ABCNS	(22.6,33.6,47.4)	(20,34,48)	(24,33,47)	
	HMVO	(22.5,34.0,48.0)	(25,33,47)	(25,34,48)	
	IABC	(24.1,36.1,50.9)	(25,34,49)	(24,38,55)	
	IDPSO	(24.90,36.67,51.00)	(23,35,49)	(27,38,51)	
	DHS	(24.1,36.1,50.9)	(24,35,48)	(26,37,53)	
	TLBO	(24.9, 36.5,50.8)	(21,36,50)	(26,40,57)	
	EDA	(24.8,37.2,51.9)	(21,36,50)	(24,39,57)	
5	IMVO	(34.0,51.7,71.1)	(34,50,70)	(35,53,72)	10.53%
	ABCNS	(35.8,53.8,73.6)	(36,53,71)	(36,52,75)	
	HMVO	(36.8,54.3,74.7)	(37,53,74)	(39,56,72)	
	IABC	**(37.8,55.8,77.7)**	**(36,54,74)**	**(42,59,84)**	
	IDPSO	(40.60,58.03,78.97)	(35,56,81)	(43,60,82)	
	DHS	(37.9,55.8,77.8)	(36,54,74)	(42,59,84)	
	TLBO	(36.1,57.5,78.2)	(36,55,72)	(37,61,82)	
	EDA	(38.6,56.9,78.3)	(36,55,73)	(40,60,81)	

In addition, the average execution time of the algorithm is presented in Fig. 8, which indicates that the average execution time of IMVO is minimal. Furthermore, the line graph of certain algorithms was magnified to highlight the

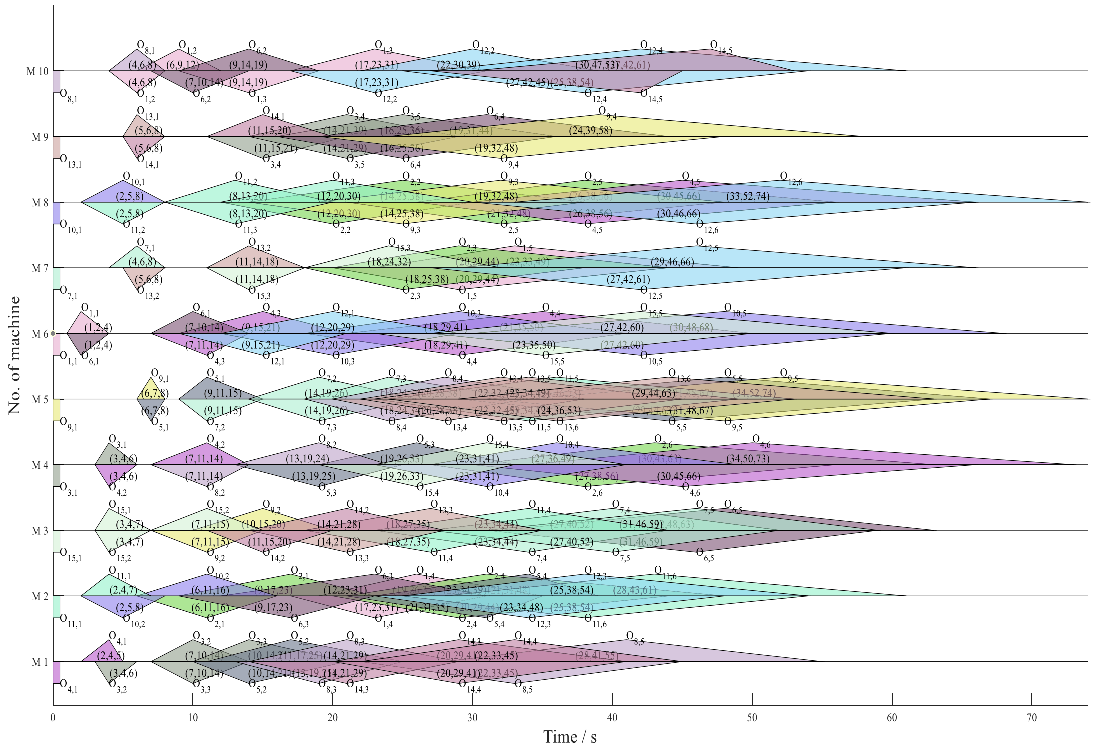

FIGURE 7. Gantt chart of optimal scheduling result of Group B Example 5.

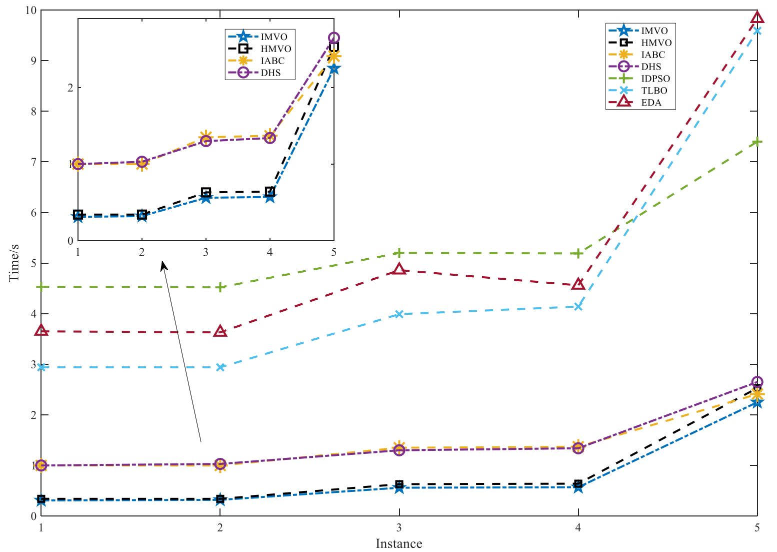

FIGURE 8. Line chart of the average execution time of each algorithm in Group B Example.

variations in the algorithm execution time. As observed, the average execution time of the IMVO algorithm was marginally less than that of the HMVO algorithm, especially in instances 1 to 4 of the Group B experiment. Nonetheless, the IMVO algorithm delivered the best solution performance among all the algorithms in terms of the solution results.

Second, to verify the performance of the IMVO algorithm for solving incomplete FFJSPs, eight incomplete flexible remanufacturing examples with Group C experimental data were simulated. The results obtained using the proposed algorithm were compared with those of the ABCNS algorithm [37], HMVO algorithm [24], and three heuristic methods [45] for a total of five current advanced algorithms.

TABLE 8. Comparison of experimental results in group C.

Instance	Instance Algorithm	Average value	Best Value	Worst value	Maximum lift rate
1	IMVO	**(19.0,26.0,33.0)**	**(19,26,33)**	**(19,26,33)**	28.2%
	ABCNS	(19.0,26.0,33.0)	(19,26,33)	(19,26,33)	
	HMVO	(17.2,26.0,34.8)	(19,26,33)	(17,26,35)	
	GS	(31.7,47.5,63.2)	(25,36,48)	(41,61,81)	
	LS	(29.0,46.6,61.2)	(17,34,45)	(40,61,79)	
	MReW	(29.6,47.3,64.7)	(20,33,45)	(58,80,107)	
2	IMVO	**(21.2,38.8,54.7)**	**(21,38,54)**	**(22,39,58)**	50.4%
	ABCNS	(22.7,39.5,55.8)	(21,40,53)	(24,39,57)	
	HMVO	(25.1,40.7,57.3)	(25,39,54)	(27,42,58)	
	GS	(53.6,83.8,116.3)	(44,66,91)	(73,116,160)	
	LS	(59.3,95.2,134.3)	(47,78,108)	(69,109,153)	
	MReW	(69.1,106.7,147.2)	(48,75,107)	(119,167,228)	
3	IMVO	**(38.6,55.8,75.7)**	**(38,54,75)**	**(42,55,75)**	45.3%
	ABCNS	(38.0,56.7,77.7)	(39,55,78)	(41,57,75)	
	HMVO	(37.9,59.9,80.8)	(41,58,77)	(40,61,81)	
	GS	(76:7,117.7,159.5)	(56,91,120)	(91,133,179)	
	LS	(82.9,128.7,173.9)	(67,101,135)	(108,160,210)	
	MReW	(94.9,144.6,194.0)	(61,102,138)	(142,198,262)	
4	IMVO	**(32.1,46.2,66.2)**	**(31,45,64)**	**(32,48,68)**	54.09%
	ABCNS	(33.3,48.7,68.0)	(34,48,66)	(32,50,71)	
	HMVO	(34.3,50.1,68.7)	(34,49,67)	(37,51,69)	
	GS	(67.2, 103.9, 141.7)	(58,91,126)	(94,131,176)	
	LS	(74.0,118.9,166.4)	(55,101,144)	(91,142,198)	
	MReW	(91.0,140.0,192.6)	(56,102,143)	(146,199,275)	
5	IMVO	**(51.6,82.4,115.8)**	**(51,80,116)**	**(51,83,116)**	54.96%
	ABCNS	(52.3,84.0,118.0)	(54,82,114)	(55,87,118)	
	HMVO	(53.1,85.8,118.8)	(46,85,120)	(50,88,122)	
	GS	(110.6,172.4,234.8)	(100,151,205)	(138,209,273)	
	LS	(143.5,217.5,291.4)	(120,182,242)	(166,248,334)	
	MReW	(147.6,234.8,317.4)	(111,173,230)	(237,324,418)	
6	IMVO	**(44.6,72.1,96.4)**	**(41,71,96)**	**(45,74,102)**	56.20%
	ABCNS	(45.2,73.3,98.7)	(46,72,95)	(46,75,103)	
	HMVO	(44.8,73.4,100.7)	(41,71,100)	(46,75,102)	
	GS	(96.0,156.1,213.9)	(84,133,173)	(108,178,246)	
	LS	(109.6,178.0,243.3)	(83,154,215)	(136,214,293)	
	MReW	(142.2,223.1,302.8)	(94,162,219)	(219,313,415)	
7	IMVO	**(62.4,109.1,158.2)**	**(63,107,157)**	**(58,110,162)**	57.28%
	ABCNS	(63.0,110.5,160.2)	(59,110,161)	(62,113,164)	
	HMVO	(63.5,110.6,157.6)	(66,107,155)	(58,113,164)	
	GS	(132.9,228.4,315.7)	(112,205,286)	(153,255,354)	
	LS	(149.8,265.4,368.5)	(136,236,330)	(169,297,414)	
	MReW	(189.8,322.5,446.9)	(134,258,366)	(329,471,631)	
8	IMVO	**(53.8,90.6,119.7)**	**(57,87,116)**	**(58,90,118)**	67.6%
	ABCNS	(55.7,91.0,126.3)	(52,90,127)	(59,91,126)	
	HMVO	(55.1,91.4,126.4)	(59,88,118)	(47,95,133)	
	GS	(120.2,202.2,278.0)	(102,178,251)	(139,231,317)	
	LS	(143.5,239.1,329.8)	(131,207,282)	(173,284,390)	
	MReW	(228.9,358.1,485.7)	(152,274,374)	(326,443,581)	

TABLE 9. Comparison of the execution time of group C instance algorithms.

Instance	1	2	3	4	5	6	7	8
IMVO	0.07	0.31	1.24	2.38	10.23	13.65	72.92	107.76
HMVO	0.04	0.31	1.37	2.54	11.21	15.98	81.49	120.37

These three methods included the local minimum processing time principle (LS), global minimum processing time principle (GS), and maximum remaining workpiece principle (MReW).

The average, optimal, and worst results of each algorithm tested on the experimental data of Group C are listed in Table 8. Based on these results, the proposed algorithm outperformed the other five algorithms for all instances of the incomplete FFJSP. Notably, the average maximum efficiency enhancement in all instances in Group C reached up to 50.44%. The results of the proposed algorithm from instance 2 to instance 5 were superior to those of the current optimal ABCNS algorithm in terms of the optimal, average, and worst performance. From instance 6 to instance 8, the results of the proposed algorithm were superior to those of the current best HMVO algorithm in terms of the average, optimal, and worst results. However, the current advanced ABCNS algorithm was not as good as the HMVO algorithm, implying that the ABCNS algorithm can easily fall into the local optima in large-scale instances, whereas the

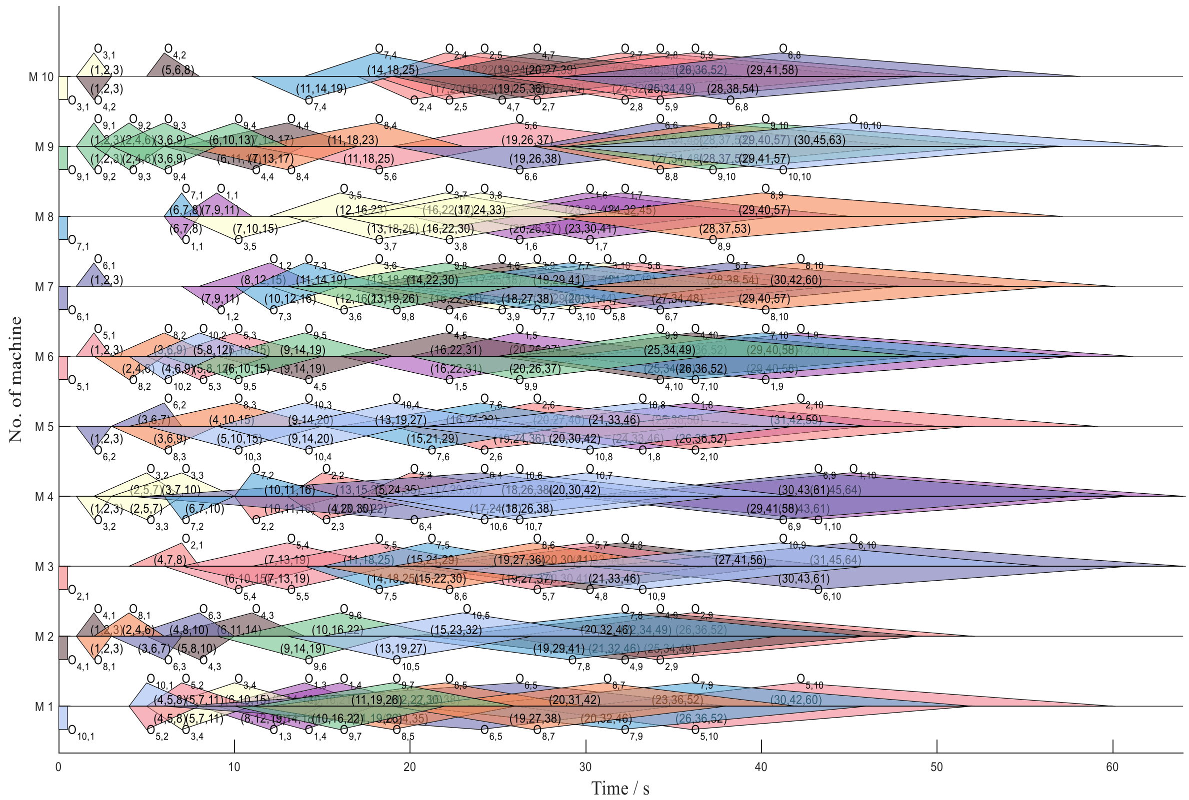

FIGURE 9. Gantt chart of optimal scheduling scheme for remanufacturing instance 4.

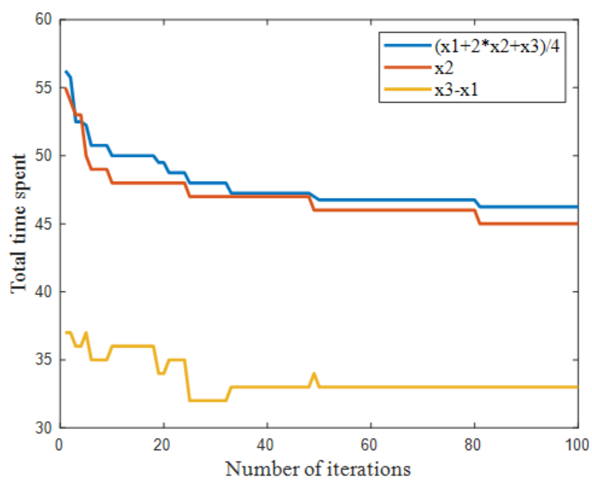

FIGURE 10. Convergence curve of remanufacturing instance 4.

proposed algorithm surpassed these limitations. In summary, the proposed algorithm is more advantageous for solving incomplete FFJSPs of various scales. The Gantt chart of the optimal scheduling result of remanufacturing instance 4 and its convergence curve is plotted in Figs. 9 and 10.

The average execution time of IMVO and HMVO algorithms for the eight instances in the Group C experiment is presented in Table 9, wherein the average execution time of the IMVO algorithm is superior to that of the HMVO algorithm, especially in large-scale instances 7 and 8, except for instance 1. In small-scale instance 1, the IMVO algorithm required a relatively higher average execution time, primarily because the initialization by the hybrid algorithm is time-consuming. However, hybrid initialization can significantly improve the population quality, which is advantageous for rapid convergence and ensures the solution quality. Specifically, this advantage is gradually reflected with the increase in the instance scale. As this advantage can provide an improved solution, a slightly longer execution period can be acceptable.

Based on the comprehensive evaluation of the experimental results, the IMVO algorithm achieved superior results and even optimal solutions for small-scale FFJSP solving at the initialization time. In most instances, the algorithm outperformed the comparison algorithm both in terms of solution results and solution time. These demonstrate the superiority and powerful performance of the IMVO algorithm for solving the FFJSP, which is primarily because of the high-quality initial solution provided by IMVO during initialization. Furthermore, the universe selection mechanism in the search process and the shift-left strategy in the decoding process ensured the rapid enhancement of the solution quality and reduced the possibility of the algorithm falling into a local optimum.

V. CONCLUSION

This study proposed an improved MVO algorithm for the FFJSP. The mathematical principles of the matter exchange through black/white holes and wormholes in the MVO were combined with two heuristic algorithms. In addition, VNS and universe retention strategies were incorporated to enhance its exploration and development capabilities. Simulation experiments were performed on three groups of FFJSPs with varying scales and types, and the results were compared with those obtained using algorithms reported in the relevant literature. The results revealed that the results of the IMVO algorithm were superior to those of the comparison algorithms, and the optimal results of certain instances are depicted in the form of Gantt charts, thereby verifying the superiority of the performance of the algorithm in this work. In the future, we intend to continue the current development based on the following aspects:

1) The proposed algorithm will be applied to solve the uncertain problems related to actual production, such as emergency job insertion and abrupt machine failure.

2) It can be employed to solve multi-objective scheduling problems such as minimum energy consumption, maximum machine utilization, and so on.

3) Further exploring the characteristics of the problem and proposing more effective algorithms.

REFERENCES

[1] B. Qian, Z.-C. Li, and R. Hu, "A copula-based hybrid estimation of distribution algorithm for *m*-machine reentrant permutation flow-shop scheduling problem," *Appl. Soft Comput.*, vol. 61, pp. 921–934, Dec. 2017.

[2] Z. C. Li, B. Qian, R. Hu, L. L. Chang, and J. B. Yang, "An elitist nondominated sorting hybrid algorithm for multi-objective flexible job-shop scheduling problem with sequence-dependent setups," *Knowl.-Based Syst.*, vol. 173, pp. 83–112, Jun. 2019.

[3] P. C. Jha and R. Shankar, "A study on application of fuzzy logic to determine production scheduling in a steel plant," *Int. J. Adv. Manuf. Technol.*, vol. 89, nos. 9–12, pp. 2529–2542, Dec. 2017.

[4] M. Gen, L. Lin, and H. Ohwada, "Advances in hybrid evolutionary algorithms for fuzzy flexible job-shop scheduling: State-of-the-art survey," in *Proc. 13th Int. Conf. Agents Artif. Intell.*, 2021, pp. 562–573.

[5] M. Sakawa and T. Mori, "An efficient genetic algorithm for job-shop scheduling problems with fuzzy processing time and fuzzy duedate," *Comput. Ind. Eng.*, vol. 36, no. 2, pp. 325–341, Apr. 1999.

[6] B. Liu, Y. Fan, and Y. Liu, "A fast estimation of distribution algorithm for dynamic fuzzy flexible job-shop scheduling problem," *Comput. Ind. Eng.*, vol. 87, pp. 193–201, Sep. 2015.

[7] H.-X. Qin, Y.-Y. Han, B. Zhang, L.-L. Meng, Y.-P. Liu, Q.-K. Pan, and D.-W. Gong, "An improved iterated greedy algorithm for the energy-efficient blocking hybrid flow shop scheduling problem," *Swarm Evol. Comput.*, vol. 69, Mar. 2022, Art. no. 100992.

[8] K. Z. Gao, P. N. Suganthan, Q. K. Pan, M. F. Tasgetiren, and A. Sadollah, "Artificial bee colony algorithm for scheduling and rescheduling fuzzy flexible job shop problem with new job insertion," *Knowl.-Based Syst.*, vol. 109, pp. 1–16, Oct. 2016.

[9] S. Huang, N. Tian, Y. Wang, and Z. Ji, "An improved version of discrete particle swarm optimization for flexible job shop scheduling problem with fuzzy processing time," *Math. Problems Eng.*, vol. 2016, pp. 1–13, Sep. 2016.

[10] L. Kuan, H. Cheng, and X. Y. Zhang, "A hybrid genetic algorithm to solve multi-objective fuzzy flexible job shop scheduling problem," *Frontiers Manuf. Eng.*, vol. 3, no. 1, pp. 5–10, 2015.

[11] Z. Yuguang, Y. Fan, and L. Feng, "Solving multi-objective fuzzy flexible job shop scheduling problem using MABC algorithm," *J. Intell. Fuzzy Syst.*, vol. 36, no. 2, pp. 1455–1473, Mar. 2019.

[12] J. Li, Z. Liu, C. Li, and Z. Zheng, "Improved artificial immune system algorithm for type-2 fuzzy flexible job shop scheduling problem," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 11, pp. 3234–3248, Nov. 2021.

[13] R. Li, W. Gong, and C. Lu, "Self-adaptive multi-objective evolutionary algorithm for flexible job shop scheduling with fuzzy processing time," *Comput. Ind. Eng.*, vol. 10, pp. 57968–57980, Jan. 2022.

[14] J. C. Seck-Tuoh-Mora, N. J. Escamilla-Serna, L. J. Montiel-Arrieta, I. Barragan-Vite, and J. Medina-Marin, "A global neighborhood with hill-climbing algorithm for fuzzy flexible job shop scheduling problem," *Mathematics*, vol. 10, pp. 52927–52938, Nov. 2022.

[15] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-Verse Optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, Feb. 2016.

[16] S. Acharya, S. Ganesan, D. V. Kumar, and S. Subramanian, "A multi-objective multi-verse optimization algorithm for dynamic load dispatch problems," *Knowl.-Based Syst.*, vol. 231, Nov. 2021, Art. no. 107411.

[17] A. Fathy and H. Rezk, "Multi-verse optimizer for identifying the optimal parameters of PEMFC model," *Energy*, vol. 143, pp. 634–644, Jan. 2018.

[18] H. Faris, M. A. Hassonah, A. M. Al-Zoubi, S. Mirjalili, and I. Aljarah, "A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture," *Neural Comput. Appl.*, vol. 30, no. 8, pp. 2355–2369, Oct. 2018.

[19] A. V. S. Reddy, M. D. Reddy, and Y. V. K. Reddy, "Feeder reconfiguration of distribution systems for loss reduction and emissions reduction using MVO algorithm," *Majlesi J. Electr. Eng.*, vol. 12, no. 2, pp. 1–8, Jan. 2018.

[20] G. Jain, G. Yadav, D. Prakash, A. Shukla, and R. Tiwari, "MVO-based path planning scheme with coordination of UAVs in 3-D environment," *J. Comput. Sci.*, vol. 37, Oct. 2019, Art. no. 101016.

[21] M. A. A. Al-Qaness, M. Abd Elaziz, A. A. Ewees, and X. Cui, "A modified adaptive neuro-fuzzy inference system using multi-verse optimizer algorithm for oil consumption forecasting," *Electronics*, vol. 8, no. 10, p. 1071, Sep. 2019.

[22] H. Jia, X. Peng, W. Song, C. Lang, Z. Xing, and K. Sun, "Hybrid multiverse optimization algorithm with gravitational search algorithm for multithreshold color image segmentation," *IEEE Access*, vol. 7, pp. 44903–44927, 2019.

[23] J. Liu, J. Wei, A. A. Heidari, F. Kuang, S. Zhang, W. Gui, H. Chen, and Z. Pan, "Chaotic simulated annealing multi-verse optimization enhanced kernel extreme learning machine for medical diagnosis," *Comput. Biol. Med.*, vol. 144, May 2022, Art. no. 105356.

[24] J. Lin, L. Zhu, and Z.-J. Wang, "A hybrid multi-verse optimization for the fuzzy flexible job-shop scheduling problem," *Comput. Ind. Eng.*, vol. 127, pp. 1089–1100, Jan. 2019.

[25] J. Gao, L. Sun, and M. Gen, "A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems," *Comput. Oper. Res.*, vol. 35, no. 9, pp. 2892–2907, Sep. 2008.

[26] P. Moradi, N. Imanian, N. N. Qader, and M. Jalili, "Improving exploration property of velocity-based artificial bee colony algorithm using chaotic systems," *Inf. Sci.*, vol. 465, pp. 130–143, Oct. 2018.

[27] C. Rim, S. Piao, G. Li, and U. Pak, "A niching chaos optimization algorithm for multimodal optimization," *Soft Comput.*, vol. 22, no. 2, pp. 621–633, Jan. 2018.

[28] K. Gao, F. Yang, M. Zhou, Q. Pan, and P. N. Suganthan, "Flexible job-shop rescheduling for new job insertion by using discrete Jaya algorithm," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1944–1955, May 2019.

[29] L. Jun and X. Renyue, "Self-crossing genetic algorithm," *Sci. Technol. Eng.*, vol. 9, no. 19, pp. 5840–5842&5846, Oct. 2009.

[30] R. Ruiz and T. Stutzle, "A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem," *Eur. J. Oper. Res.*, vol. 177, no. 3, pp. 2033–2049, Mar. 2007.

[31] G. Deng and X. Gu, "A hybrid discrete differential evolution algorithm for the no-idle permutation flow shop scheduling problem with makespan criterion," *Comput. Oper. Res.*, vol. 39, no. 9, pp. 2152–2160, Sep. 2012.

[32] M. F. Tasgetiren, Q.-K. Pan, P. N. Suganthan, and O. Buyukdagli, "A variable iterated greedy algorithm with differential evolution for the no-idle permutation flowshop scheduling problem," *Comput. Oper. Res.*, vol. 40, no. 7, pp. 1729–1743, Jul. 2013.

[33] J. Lin, "A hybrid biogeography-based optimization for the fuzzy flexible job-shop scheduling problem," *Knowl.-Based Syst.*, vol. 78, pp. 59–74, Apr. 2015.

[34] M. Wu, D. Yang, and T. Liu, "An improved particle swarm algorithm with the elite retain strategy for solving flexible jobshop scheduling problem," *J. Phys., Conf.*, vol. 2173, no. 1, Jan. 2022, Art. no. 012082.

[35] K. Fleszar and K. S. Hindi, "Solving the resource-constrained project scheduling problem by a variable neighbourhood search," *Eur. J. Oper. Res.*, vol. 155, no. 2, pp. 402–413, Jun. 2004.
[36] E. Balas, "Machine sequencing via disjunctive graphs: An implicit enumeration algorithm," *Oper. Res.*, vol. 17, no. 6, pp. 941–957, Dec. 1969.
[37] X. C. Zheng and W. Y. Gong, "An improved artificial bee colony algorithm for fuzzy flexible job-shop scheduling problem," *Control Theory Appl.*, vol. 37, no. 6, pp. 1284–1292, Jun. 2020.
[38] D. Lei, "A genetic algorithm for flexible job shop scheduling with fuzzy processing time," *Int. J. Prod. Res.*, vol. 48, no. 10, pp. 2995–3013, May 2010.
[39] D. Lei, "Co-evolutionary genetic algorithm for fuzzy flexible job shop scheduling," *Appl. Soft Comput.*, vol. 12, no. 8, pp. 2237–2245, Aug. 2012.
[40] K. Z. Gao, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "An effective discrete harmony search algorithm for flexible job shop scheduling problem with fuzzy processing time," *Int. J. Prod. Res.*, vol. 53, no. 19, pp. 5896–5911, Oct. 2015.
[41] P. Brucker, B. Jurisch, and B. Sievers, "A branch and bound algorithm for the job-shop scheduling problem," *Discrete Appl. Math.*, vol. 49, nos. 1–3, pp. 107–127, Mar. 1994.
[42] K. Z. Gao, P. N. Suganthan, T. J. Chua, C. S. Chong, T. X. Cai, and Q. K. Pan, "A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion," *Exp. Syst. Appl.*, vol. 42, no. 21, pp. 7652–7663, Nov. 2015.
[43] Y. Xu, L. Wang, S.-Y. Wang, and M. Liu, "An effective teaching-learning-based optimization algorithm for the flexible job-shop scheduling problem with fuzzy processing time," *Neurocomputing*, vol. 148, pp. 260–268, Jan. 2015.
[44] S. Wang, L. Wang, Y. Xu, and M. Liu, "An effective estimation of distribution algorithm for the flexible job-shop scheduling problem with fuzzy processing time," *Int. J. Prod. Res.*, vol. 51, no. 12, pp. 3778–3793, Jun. 2013.
[45] F. Pezzella, G. Morganti, and G. Ciaschetti, "A genetic algorithm for the flexible job-shop scheduling problem," *Comput. Oper. Res.*, vol. 35, no. 10, pp. 3202–3212, 2008.

JIN-CHENG FANG received the Ph.D. degree from Fuzhou University, in 2015. He is currently the Director of the Industrial Engineering Program and the Master's of Industrial Engineering Program, and a member of the MEM Training Steering Committee, Fujian University of Technology. His research interests include enterprise production management, intelligent manufacturing, technology innovation, and optimization management.

A-FENG ZENG received the bachelor's degree in transportation from the Fujian University of Technology, in 2020, where he is currently pursuing the master's degree in mechanical and industrial engineering. His main research interests include shop floor scheduling optimization, meta-heuristic algorithms, operations research, and optimization.

SHAO-FENG ZHENG received the master's degree from the Open University of Hong Kong, in 2006. He is currently a Senior Engineer and a Master's Tutor with the Fujian University of Technology. His main research interests include artificial intelligence, intelligent scheduling, and intelligent management.

WEN-DI ZHAO (Member, IEEE) is currently pursuing the master's degree with the School of Transportation, Fujian University of Technology. His research interests include fuzzy systems and intelligent transportation systems.

XU HE is currently pursuing the degree with the Department of Industrial Engineering, Fujian University of Technology. He has achieved many achievements. His research interests include supply chain management, lean production management, and multi-attribute decision analysis.