Preference-based Multi-objective Genetic Programming for Energy-efficient Dynamic Flexible Job Shop Scheduling

Zhuoyin Qiao, Fangfang Zhang✉, Yi Mei and Mengjie Zhang
Centre for Data Science and Artificial Intelligence & School of Engineering and Computer Science
Victoria University of Wellington
PO BOX 600, Wellington 6140, New Zealand
{zhuoyin.qiao, fangfang.zhang, yi.mei, mengjie.zhang}@ecs.vuw.ac.nz

2025 IEEE Congress on Evolutionary Computation (CEC) | 979-8-3315-3431-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/CEC65147.2025.11043041

***Abstract*—Energy-efficient dynamic flexible job shop scheduling (E-DFJSS) is a valuable real-world combinational optimisation problem. As an important variant of DFJSS, E-DFJSS aims to optimise trade-off between production effectiveness and energy consumption. Genetic Programming (GP) has been successfully used to learn dispatching rules in E-DFJSS. Nevertheless, different users have different preferences on the trade-off between production effectiveness and energy consumption, the studies of preference-based multi-objective optimisation algorithms are limited. These existing related preference-based methods face the issue of premature convergence in the early stage when solving E-DFJSS, which results in insufficient diversity. To address these challenges, we develop a preference-based multi-objective GP approach for E-DFJSS. A fusion *r*-dominance and achievement scalarising function dominance criterion is embedded into the proposed algorithm to solve E-DFJSS. Experimental results on training and test on three scenarios with four preferences show that the proposed method could improve the effectiveness of learned scheduling heuristics by balancing diversity and convergence of evolutionary progress.**

***Index Terms*—Energy-efficient job shop scheduling, Genetic programming, Preference-based multi-objective optimisation**

I. Introduction

Dynamic flexible job shop scheduling has attracted significant interest due to its ability to reflect dynamic events occurring in actual production and the flexibility of machine resources [1]. With the intensification of climate change, energy-efficient production methods have gained widespread attention. As a result, energy-efficient dynamic flexible job shop scheduling (E-DFJSS) has become a key area of research in industrial production [2]. In E-DFJSS, a series of jobs consisting of several operations arrives in the job shop for processing. Each operation can be processed by different machines. The goal of this problem is to obtain effective production schedules and energy efficiency by optimising operation sequence (*sequencing*) and machine assignment (*routing*) [3].

Without loss of generality, the conflicting production schedules and energy consumption can be described as a multi-objective optimisation problem (MOP) [4]. In this problem, the decision makers (DMs) aim to find a set of Pareto-optimal candidate solutions, which means that each solution cannot be further improved in one objective without sacrificing another. Multi-objective Genetic Programming (MOGP), as a hyper-heuristic algorithm, has been successfully applied to solve the E-DFJSS [5]. Unlike searching for solutions directly in the solution space, MOGP generates solutions by searching in the heuristic space for heuristic methods [6]. Specifically, MOGP maintains a population consisting of several individuals, and each individual represents a priority function (called the dispatching rule) formed by the features of DFJSS [7]. The population learns iteratively by initialisation, evaluation, parent selection, and breeding, aiming to find a set of Pareto-optimal solutions.

However, in real world production scheduling, different DMs with the same optimisation objectives often have different preferences [8]. For example, when faced with production tasks that have a generous delivery deadline, the DM may prioritise minimizing energy consumption while completing the production task. In contrast, when faced with production tasks with imminent deadlines, the DM's focus shifts to completing the production schedule on time, with less concern for energy consumption. Incorporating the preference information of the DM into the search process is of practical significance, and computational resources can be concentrated on searching in directions that integrate the preferences of DMs [9]. Generally, the embedding of preference information can be classified into *priori*, *posteriori*, and *interactive*. This paper focuses on integrating preference information in the priori manner due to its advantages over the traditional posteriori and interactive approaches that have been discussed in [10].

Currently, research on preference information has become quite extensive. Based on the ways of expressing preference information, it can be classified into weights, reference points, reference vectors, outranking, and implicit ranking. Among these, using reference points as a way to express preference information is particularly popular [11], [12]. Compared with other preference modelling tools, reference points represent a DM's preference information relatively intuitively. However, reference-point-based multi-objective algorithms face two main challenges: the selection of reference points [12] and

979-8-3315-3431-8/25/$31.00 ©2025 IEEE

the dominance criterion based on user preferences [13]. With respect to the selection of reference points, the number and position of reference points affect the search direction of the algorithm. In addition, selecting reference points requires the decision maker (DM) to have a general understanding of the Pareto Front (PF) in the given scenario. In terms of the dominance criterion, it incorporates the user's requirements into the dominance decision-making phase of the multi-objective algorithm. Through this criterion, the algorithm searches for Pareto candidate solutions based on preferences of users. As in [12], using a single reference point as the first step, this paper focuses on the dominance criterion based on user preferences.

In the research on dominance criterion based on user preferences, the most classic and commonly used approaches are r-dominance [14], g-dominance [15], and angle-based dominance [16]. At present, researchers are more inclined to integrate multiple dominance criteria according to user preferences to form new dominance criteria [8]. However, existing methods [9] often focus overly on convergence speed in the early stages, neglecting the diversity of candidate solutions, which hinders the provision of diverse breeding samples in the later evolutionary stages. Therefore, this paper aims to propose a dominance criterion based on user preferences that integrates r-dominance and weighted achievement scalarising function (WASF). This criterion is expected to maintain a high diversity of the population in the early stages to ensure the high effectiveness of the final PF. Based on this, the main research objectives of this paper are as follows:

1) Develop an effective preference-based multi-objective GP algorithm for solving E-DFJSS.
2) Propose a novel dominance criterion that can maintain diversity in the early stage of the evolution process. The dominance criteria under WASF and r-dominance are integrated and embedded into the proposed multi-objective GP algorithm.
3) Verify the performance of the proposed algorithm by analysing the results in terms of training and test.

II. Background

A. Problem Definition

In E-DFJSS, a set of jobs $\mathfrak{J} = \{J_1, J_2, \ldots, Jn\}$ need to be processed by a set of machines $\mathfrak{M} = \{M_1, M_2, \ldots, M_m\}$. Each machine has its speed which is randomly sampled from a discrete and finite set $\mathfrak{V} = \{v_1, v_2, \ldots, v_s\}$. Each job J_i in $\mathfrak{J}$ has the following attributes: a release time r_i, a due date dd_i, a weight w_i and a sequence of operations $[O_{i,1}, O_{i,2}, \ldots, O_{i,o}]$, where o is the number of operations of job J_i. Each operation can be processed by a set of machines. If operation $O_{i,j}$ is processed by machine M_k with speed v_z, then the actual processing time of it equals $\frac{PT_{i,j,k}}{V_{k,z}}$, $PT_{i,j,k}$ is basic process time of operation $O_{i,j}$ which is processed by machine M_k. In this paper, job arrival event is considered and all information about the new jobs is unknown until it arrives.

In the E-DFJSS, production providers expect to find a balance between production effectiveness and energy consumption. Three classical objectives are considered as the objectives to measure the production effectiveness in this paper, which are shown as follows:

$$Fmean = \frac{1}{n}\sum_{i=1}^{n}(C_i - r_i),\ Tmean = \frac{1}{n}\sum_{i=1}^{n}(C_i - dd_i, 0), \quad (1)$$
$$WTmean = \frac{1}{n}\sum_{i=1}^{n} W_i * (C_i - dd_i, 0),$$

where C_i, r_i and dd_i are the completion time, release time and due date of the job J_i. $Fmean$, $Tmean$ and $WTmean$ are the mean of flow time, the mean of tardiness and the mean of the weighted tardiness [3]. Moreover, the objective of Total Energy Consumption (TEC) can be described as follows:

$$\begin{aligned} TEC &= E_1 + E_2 \\ &= \sum_{i=1}^{n} PE_i + \sum_{k=1}^{m} IE_k \\ &= \sum_{i=1}^{n}\sum_{j=1}^{o}\sum_{k=1}^{m}\sum_{z=1}^{s} x_{i,j,k} v_{i,j,k,z} \frac{PT_{i,j,k}}{V_{k,z}} PP_{k,z} \\ &+ \sum_{k=1}^{m} IP_k * IT_k, \end{aligned} \quad (2)$$

where PE_i and IE_k are processing energy consumption and idle energy consumption, $x_{i,j,k}$ and $v_{i,j,k,z}$ are boolean values. $x_{i,j,k}$ represents whether operation $O_{i,j}$ is processed by machine k, $v_{i,j,k,z}$ represents whether the speed of machine k processing $O_{i,j}$ is $V_{k,z}$. $PP_{k,z}$ and IP_k represent the processing power of machine k with speed v_z and idle power of machine k. IT_k is the idle time of the machine k.

This paper combines the three objectives with energy consumption respectively to form multi-objective problems. The model of the problem is shown as follows:

$$\begin{gathered} \min\ \{(OBJ, TEC)\}, \\ \text{where } OBJ \text{ can be } Fmean, Tmean, WTmean \end{gathered} \quad (3)$$

s.t.:

$$\begin{gathered} T_k = \{ST_{k,l}, FT_{k,l}\}, \\ \forall k = 1, \ldots, m,\ \forall l = 1, \ldots, c. \end{gathered} \quad (4)$$

$$\begin{gathered} FT_{k,l} - ST_{k,l} = \frac{PT_{i,j,k}}{V_{k,z}}, \\ \forall k = 1, \ldots, m,\ \forall l = 1, \ldots, c. \end{gathered} \quad (5)$$

$$\begin{gathered} IT_k = \sum_{l=2}^{c}(IT_{k,(1,2)} + IT_{k,(2,3)} + \ldots + IT_{k,(l-1,l)}), \\ \forall k = 1, \ldots, m. \end{gathered} \quad (6)$$

$$\begin{gathered} IT_{k,(l-1,l)} = ST_{k,l} - FT_{k,l-1}, \\ \forall k = 1, \ldots, m,\ \forall l = 1, \ldots, c. \end{gathered} \quad (7)$$

$$\begin{gathered} \sum_{z=1}^{s} v_{i,j,k,z} = 1, \\ \forall i = 1, \ldots, n,\ \forall j = 1, \ldots, o,\ \forall k = 1, \ldots, m. \end{gathered} \quad (8)$$

$$v_{i,j,k,z} = \begin{cases} 1, & \text{if } O_{i,j} \text{ is processed on } M_k \text{ by } V_{k,z}, \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

$$ST_{k,l} - FT_{k,l-1} \geq 0, \\ \forall k = 1, \ldots, m, \ \forall l = 1, \ldots, c. \tag{10}$$

$$ST_{i,j} + (1 - y_{i,j,a,b,k})LN \geq FT_{a,b}, \\ \forall i, a = 1, \ldots, n, \ \forall j = 1, \ldots, o, \\ \forall b = 1, \ldots, o_b, \ \forall k = 1, \ldots, m. \tag{11}$$

$$y_{i,j,a,b,k} = \begin{cases} 1, & \text{if } O_{i,j} \text{ is processed on } M_k \text{ before } O_{a,b}, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

$$\sum_{k=1}^{m} x_{i,j,k} = 1, \\ \forall i = 1, \ldots, n, \ \forall j = 1, \ldots, o. \tag{13}$$

$$x_{i,j,k} = \begin{cases} 1, & \text{if } O_{i,j} \text{ is processed on } M_k, \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

$$ST_{i,j+1} - FT_{i,j} \geq 0, \\ \forall i = 1, \ldots, n, \ \forall j = 1, \ldots, o. \tag{15}$$

where Eq. (4) - (14) mean the relation and constraints based on machine. Among them, Eq. (4) - (7) means that the time of each progress l on machine M_k. T_k represents the time of machine k, $ST_{k,l}$ and $FT_{k,l}$ mean that the start time and the finish time of the process l on the machine k. Eq. (8) and Eq. (9) mean the idle time of each machine M_k. The idle time of the machine k is accumulated by the $IT_{k,(l-1,l)}$. $IT_{k,(l-1,l)}$ is the idle time between process $l-1$ and process$l$. Eq. (10) and Eq. (11) represent that just one speed option can be selected when an operation is processed on the machine. Eq. (12) - (14) means just one progress can be processed on the machine at the same time. In other words, each machine cannot start a new progress until the previous progress is finished. Finally, Eq. (15) - (17) mean the relation and constraints based on job. Eq. (15) - (16) represents that each operation of each job just can be processed by one machine. Eq. (17) indicates that each operation cannot be processed until the previous operation is finished.

B. Preference-based Multi-objective Optimisation

As mentioned before, appropriate dominance criterion based on preference is an important way to enhance the effectiveness of performance. The earliest known method to model preferences and incorporate them into multi-objective optimisation is proposed by Fonseca et al [17], Deb et al. [4] demonstrated a method that combines preference strategies with NSGA-II to improve solutions that better align with the preferences of DMs. This approach incorporates preference information into the calculation of crowding distance:

$$dist(x_i, \mathrm{rp}) = \sqrt{\sum_{j=1}^{m} w_j \left(\frac{x_{i,j} - rp_j}{f_{max,j} - f_{min,j}}\right)^2}. \tag{16}$$

where $dist(x_i, \mathrm{rp})$ represents the distance between the feasible solution x_i and reference point rp, and w_j is the weight of objective j based on the preference of users. $f_{max,j}$ and $f_{min,j}$ are the maximum fitness and minimum of objectives j, respectively.

Inspired by this, Said et al. [14] further proposed a variant of the Pareto dominance relation, where the strict partial order is established by calculating the distance from the Pareto candidates to the reference point. Furthermore, WASF-GA [18] is proposed to approximate the region of interest near the reference point by WASF. The WASF is calculated as follows:

$$WASF(x_i, \mathrm{rp}) = [\max_{j=1,\ldots,m} \{w_j(f_j(x_i) - (\mathrm{rp}_j - \varepsilon))\} \\ + \rho \sum_{j=i}^{m} w_j f_j(x_i)], \tag{17}$$

where, x_i, rp and w_j are same as before. $f_j(x_i)$ is the fitness of x_i with the objective j. ε is set to $0.1 * \mathrm{rp}_j$, and $(\mathrm{rp}_j - \varepsilon)$ is the utopian point. The parameter ρ is set to 0.001. In addition, a domination criterion named reference-point-and-angle-based dominance criterion is proposed [19]. This method determines the preference range by calculating the reference direction and preference angle and constructs the domination relationship to achieve optimisation.

Recently, an increasing number of researchers have been constructing dominance criteria by integrating multiple ways. In order to find the preferred solution near the PF projected by the reference point, Thiele et al. [20] combined the ASF with the quality indicator. In [21], the knee point information in the non-dominated solution and the approximate value of the hyper-volume are combined to enhance the convergence performance in the multi-objective optimisation. Yi et al. [9] introduced a preference-based dominance relation by merging angle-based and r-dominance with angular relationships. This method balances both expertise and population diversity. Zhao et al. [8] proposed a method to linearly adjust the achievement scalarising function and the weight sum function during the evolutionary process. This method uses a simple adaptive penalty scheme to balance search capability and robustness, effectively alleviating the selection pressure on the PF.

To the best of our knowledge, related studies based on user preferences to solve E-DFJSS with MOGP have been successfully implemented, but current research mainly aims to improve convergence, neglecting the maintenance of diversity. To fill this gap, we propose a user preference-based MOGP framework to address E-DFJSS.

III. Proposed Preference-based MOGP for E-DFJSS

A. Overview of the Proposed Algorithm

The flowchart for the proposed preference-based MOGP for E-DFJSS is shown in Fig. 1. In the flowchart, each individual in the initialised population consists of two trees [22]. The individuals are then passed into the E-DFJSS simulation for fitness evaluation. In E-DFJSS, each individual is evaluated in terms of its two objective values. If the termination condition is met, the PF is output; otherwise, the individuals are sorted by preference-based non-dominated sorting. In this part, we model the preference presentation of the DMs first. In this paper, a combination of reference points and weights as the representation is selected. After that, the preferences of users are incorporated into the preference representation model.

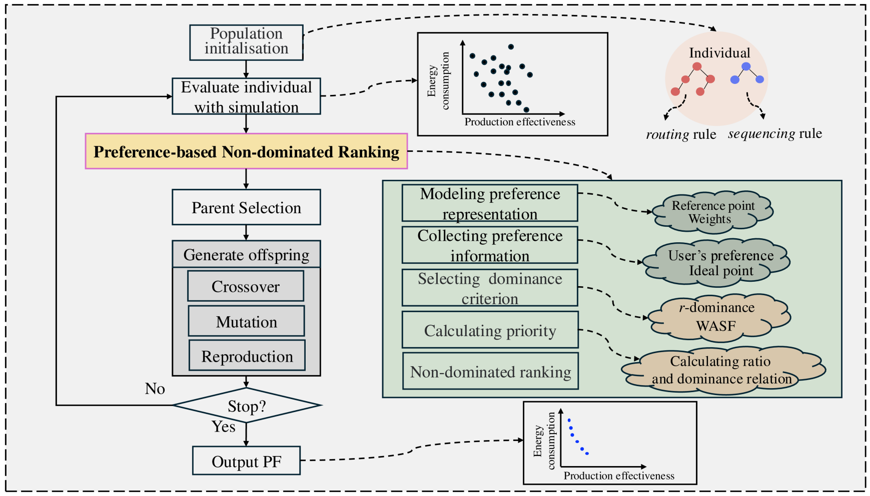

Fig. 1. Flowchart of proposed preference-based MOGP for E-DFJSS.

Subsequently, the preference dominance relation of each individual is determined based on the dominance criterion. The preference dominance relation is used for Non-dominated ranking. In the offspring generation phase, parent selection is first performed, followed by crossover, mutation, and reproduction operators applied according to probabilities. The next step proceeds to the simulation-based fitness evaluation and continues the loop.

B. Dominance Criterion based on WASF and r-dominance

As mentioned above, existing preference multi-objective algorithms based on dominance criterion are limited. When the existing dominance criteria are embedded in the proposed flowchart to solve the E-DFJSS problem, they struggle to balance the convergence and diversity of solutions. In other words, this limitation is usually manifested as existing methods converge too quickly in the early stages of evolution, leading to an inability to maintain good diversity [9]. Maintaining diversity is particularly important in multi-objective optimisation because it ensures the discovery of more diverse and potentially optimal solutions, rather than being confined to a narrow solution space. For this reason, we design a dominance criterion based on WASF and r-dominance (called WAr-dominance criterion). To fully use the properties of WASF and r-dominance, the combined form is as follows:

$$\Phi(x_i,\mathrm{rp}) = [\xi(t)dist(x_i,\mathrm{rp}) + (1-\xi(t))WASF(x_i,\mathrm{rp})] \quad (18)$$

where $\xi(t)$ adjusts the relative ratio of the WASF and r-dominace, which depends on the number of iterations. t denotes the current generation. Based on prior research [20], WASF has been shown to effectively maintain diversity in the current objective direction. Therefore, the ratio is designed as follows:

$$\xi(t) = \xi_{min} + (\xi_{max} - \xi_{min})e^{-(1-\frac{t}{T})} \quad (19)$$

where T is the maximum generation. Based on the research [9], sample ranges for combinations of more dominance criteria are considered. Therefore, ξ_{min} and ξ_{max} are set to 0.1 and 0.9, respectively.

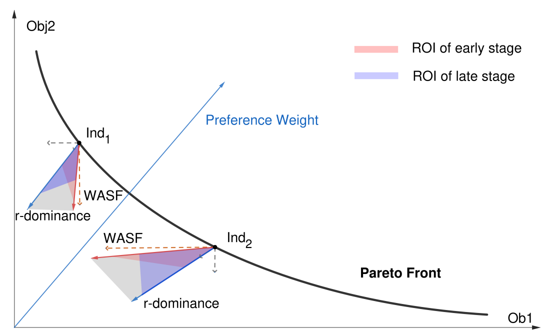

Fig. 2. Example of the comparison of solutions Ind_1 and Ind_2.

As shown in Fig. 2, an illustration of the preference priorities of two individuals located on different sides of the preference vector is provided. There are two individuals on the PF and one preference weight in the space. For the Ind_1, the contour line of WASF is computed (solid red line), the red dashed line represents the first part of Eq. (17), and the blue line is r-dominance contour line. Therefore, the red-shaded area represents the range of interest in the early stage of the evolution process. The range of interest in the later stage of the evolutionary process is represented by the blue-shaded area. The same holds for the Ind_2. In this way, the WAr-dominance criterion is expected to achieve a balance between diversity and convergence by maintaining good diversity during the early stages of evolution.

IV. Experiment Design

A. Simulation Model

In this paper, 5,000 jobs with processing tasks are handled by 10 machines. Each machine has a different processing

TABLE I
THE TERMINAL SET.

Notation	Description
MIT	Idle time of a machine
MIP	Idle power of a machine
MPP	Processing power of a machine
NMPP	Next processing power of a machine
WIQ	The workload in the queue of a machine
NIQ	The number of operations in the queue
PT	Processing time of an operation on a machine
NPT	Median processing time of the next operation
OWT	The waiting time of an operation
W	Weight of a job
NOR	The number of operations remaining for a job
TIS	Waiting time in the system for a job

TABLE II
TWELVE SCENARIOS.

Scenario	Objective 1	Objective 2	Preference
1	mean-flowtime	TEC	(0.9,0.1)
2	mean-flowtime	TEC	(0.8,0.2)
3	mean-flowtime	TEC	(0.2,0.8)
4	mean-flowtime	TEC	(0.1,0.9)
5	mean-tardiness	TEC	(0.9,0.1)
6	mean-tardiness	TEC	(0.8,0.2)
7	mean-tardiness	TEC	(0.2,0.8)
8	mean-tardiness	TEC	(0.1,0.9)
9	mean-weighted-tardiness	TEC	(0.9,0.1)
10	mean-weighted-tardiness	TEC	(0.8,0.2)
11	mean-weighted-tardiness	TEC	(0.2,0.8)
12	mean-weighted-tardiness	TEC	(0.1,0.9)

speed for different operations on different jobs, with processing speeds randomly selected from the speed pool {1, 1.3, 1.55, 1.75, 2.1} [23]. The base parameter ε of each machine follows a uniform distribution in the range [5, 10]. It depends on the properties of the machine. The processing power and idle power of the machines are set to $PP = \varepsilon \cdot V^2$ and $IP = \frac{\varepsilon}{4}$ [23]. The number of operations for each job follows a uniform distribution between 1 and 10, and the processing time for each operation is assigned from a uniform distribution between 10 and 99. Additionally, to simulate the varying importance of different jobs, weights of 1, 2, and 4 are assigned to each job, with the proportions of each job type being 20%, 60%, and 20%, respectively. The arrival frequency of jobs in the simulation reflects the busyness of the shop floor. We set the level to 0.95 to represent a busy environment. The due date factor for each job is set to 1.2. These settings are consistent with popular parameter settings [3].

B. Parameter Setting

In the experiments conducted in this paper, the terminal settings of MOGP are designed to reflect job shop-related, machine-related, and job-related features, which are shown in Table I. The parameters are set in the same way as the popular parameters [1], [3]. The function set consists of {+, -, *, /, max, min}, where "/" denotes protected division and returns 1 when the denominator is 0. The parameter settings for the main GP are as follows: the population initialization method employs the widely-used Ramped-half-and-half approach, and the population size is 1000. Each GP individual is represented by two subtrees with a maximum depth of 8 for each tree [22]. During the breeding phase, the probabilities for crossover, mutation, and reproduction are set to 0.80, 0.15, and 0.05, respectively. The selection probabilities for terminal and non-terminal nodes are 0.10 and 0.90, respectively. Parent selection is based on tournament selection with a tournament size of 7. The maximum number of generations is set to 51, and the parameter settings for the compared algorithms are configured according to the original paper.

C. Comparison Design

To verify the effectiveness of the proposed preference-based dominance criterion, this paper designs a total of 12 common scenarios with four types of preferences based on the three objectives mentioned earlier. Detailed information is provided in Table II. For comparison, we select five state-of-the-art preference-based dominance criteria (called *r*-dominance [14], R-dominance [4], WASF [20], ar-dominance [9], and LSF [8]) and integrate them into the proposed framework, resulting in five compared algorithms: *r*-NSGP, R-NSGP, WAF-NSGP, ar-NSGP, and LSF-NSGP. As mentioned in Section II, the former three are all classical preference-based single dominance criteria, while the later two belong to multiple dominance criteria. In addition, to verify whether the proposed criterion can maintain diversity during the early stages of the evolution process, we compare the proposed WAr-dominance criterion with two types of multiple dominance criterion(ar-dominance and LSF-dominance). The experiment is divided into two parts: training and test. In the training phase, each algorithm runs 30 times independently, resulting in 30 sets of PF. After training, the solutions on each PF are tested on 30 unseen instances.

V. Results and Analysis

To compare the effectiveness of the solutions obtained by the algorithm, the updated R-HV [24] and R-IGD [25] are used as quality indicators. Both metrics are derived from the R-metrics, which is designed for algorithm comparison based on user preferences according to multicriterion decision making. The updated R-HV is the latest version of the R-HV that has improved some limitations. In addition, we also analyse the diversity of the population during the evolution process to verify the effectiveness of the proposed algorithms. The diversity is measured using the Maximum Spread (MS) [13].

The performance of all algorithms based on 30 independent runs is analysed using the Friedman test and the Wilcoxon rank-sum test with Bonferroni correction, with a significance level of 0.05. "Average rank" represents the average rank of the algorithm across different scenarios. The symbols "↑", "↓", and "≈" indicate statistical significance, meaning that the result is significantly better, significantly worse, or approximately the same. For the updated R-HV and R-IGD metrics, a larger/smaller value of updated R-HV/R-IGD indicates better effectiveness.

TABLE III
THE MEAN (STANDARD DEVIATION) OF THE UPDATED R-HV AND R-IGD ON **TRAINING INSTANCES** OF WAR-NSGP WITH FIVE COMPARED ALGORITHMS BASED ON 30 INDEPENDENT RUNS IN 12 SCENARIOS.

Scenario	*r*-NSGP	R-NSGP	WASF-NSGP	ar-NSGP	LSF-NSGP	WAr-NSGP
			Updated R-HV of Training			
1	0.010(0.015)(≈)	0.001(0.003)(↑)	0.032(0.026)(≈)	0.010(0.017)(≈)	**0.036(0.023)**(↓)	0.020(0.021)
2	0.076(0.055)(≈)	0.020(0.020)(↑)	**0.102(0.049)**(≈)	0.060(0.052)(≈)	0.102(0.057)(≈)	0.085(0.060)
3	0.105(0.040)(≈)	0.014(0.044)(↑)	0.111(0.039)(≈)	0.096(0.047)(≈)	**0.117(0.042)**(≈)	0.108(0.047)
4	0.042(0.033)(≈)	0.004(0.021)(↑)	0.039(0.022)(≈)	0.023(0.032)(↑)	0.031(0.036)(↑)	**0.054(0.042)**
5	0.081(0.027)(↑)	0.023(0.019)(↑)	0.083(0.014)(↑)	0.071(0.019)(↑)	0.085(0.022)(≈)	**0.094(0.012)**
6	0.192(0.034)(≈)	0.058(0.040)(↑)	0.191(0.031)(≈)	0.179(0.051)(≈)	0.197(0.026)(≈)	**0.201(0.029)**
7	**0.136(0.054)**(≈)	0.026(0.052)(↑)	0.127(0.045)(≈)	0.095(0.058)(↑)	0.108(0.046)(≈)	0.122(0.046)
8	0.052(0.027)(≈)	0.018(0.042)(↑)	0.040(0.027)(↑)	0.031(0.022)(↑)	0.042(0.027)(↑)	**0.066(0.045)**
9	0.074(0.031)(≈)	0.020(0.019)(↑)	0.083(0.020)(≈)	**0.089(0.027)**(≈)	0.086(0.016)(≈)	0.075(0.029)
10	**0.192(0.022)**(≈)	0.057(0.042)(↑)	0.181(0.033)(≈)	0.179(0.043)(≈)	0.181(0.033)(≈)	0.189(0.026)
11	0.120(0.056)(≈)	0.013(0.033)(↑)	**0.128(0.058)**(≈)	0.106(0.061)(≈)	0.116(0.060)(≈)	0.119(0.049)
12	0.076(0.041)(≈)	0.013(0.040)(↑)	0.063(0.040)(≈)	0.030(0.033)(↑)	0.055(0.033)(↑)	**0.081(0.036)**
Win / Draw / Lose	1 / 11 /0	12 / 0 /0	2 / 10 / 0	5 / 7 / 0	3 / 8 / 1	N/A
Average Rank	3.00	5.60	2.87	3.75	2.99	**2.79**
			R-IGD of Training			
1	**0.042(0.020)**(≈)	0.534(0.164)(↑)	0.049(0.019)(≈)	0.089(0.035)(↑)	0.042(0.026)(≈)	0.042(0.018)
2	**0.030(0.021)**(≈)	0.554(0.165)(↑)	0.046(0.034)(≈)	0.093(0.042)(↑)	0.038(0.029)(≈)	0.032(0.018)
3	**0.023(0.013)**(≈)	0.577(0.222)(↑)	0.031(0.019)(≈)	0.044(0.022)(↑)	0.042(0.016)(↑)	0.033(0.022)
4	0.041(0.022)(≈)	0.633(0.190)(↑)	**0.038(0.020)**(≈)	0.042(0.026)(≈)	0.052(0.033)(≈)	0.050(0.027)
5	0.029(0.014)(≈)	0.463(0.197)(↑)	0.036(0.020)(↑)	0.077(0.047)(↑)	0.030(0.023)(≈)	**0.026(0.015)**
6	0.023(0.011)(≈)	0.465(0.203)(↑)	0.022(0.010)(≈)	0.088(0.046)(↑)	**0.021(0.013)**(≈)	0.022(0.011)
7	**0.028(0.014)**(≈)	0.533(0.236)(↑)	0.033(0.028)(≈)	0.051(0.031)(↑)	0.037(0.024)(≈)	0.030(0.023)
8	0.048(0.035)(≈)	0.466(0.230)(↑)	**0.034(0.021)**(≈)	0.051(0.026)(↑)	0.036(0.019)(≈)	0.041(0.031)
9	0.025(0.015)(≈)	0.467(0.200)(↑)	0.036(0.027)(↑)	0.082(0.044)(↑)	0.041(0.032)(↑)	**0.022(0.010)**
10	0.025(0.013)(≈)	0.515(0.195)(↑)	**0.023(0.017)**(≈)	0.072(0.044)(↑)	0.026(0.014)(≈)	0.027(0.013)
11	0.027(0.017)(≈)	0.548(0.224)(↑)	0.030(0.016)(≈)	0.052(0.026)(↑)	0.030(0.018)(≈)	**0.022(0.014)**
12	0.042(0.031)(≈)	0.527(0.246)(↑)	0.032(0.017)(≈)	0.047(0.016)(↑)	**0.032(0.018)**(≈)	0.036(0.023)
Win / Draw / Lose	0 / 12 /0	12 / 0 /0	2 / 10 / 0	11 / 1 / 0	2 / 10 / 0	N/A
Average Rank	2.65	5.99	2.71	4.21	2.80	**2.61**

A. Overall Performance of Training and Test

Table III shows the mean and standard deviation of updated R-HV and R-IGD of the proposed method and four compared methods in the training process. For updated R-HV, according to the average Friedman rank, the proposed multiple domination criteria based on *r*-dominance and WASF achieves the best performance. Specifically, the proposed method performs significantly better than R-NSGP on all scenarios. In the scenario 1 to scenario 4, WAr-NSGP is significantly better than two compound preference-based dominance criterion methods (called ar-NSGP and LSF-NSGP). For the scenario 5 to scenario 8, the proposed method gets the best results with three preferences. Particularly, WAr-NSGP is significantly better than the four compared methods with two preferences. As the comparison in scenario 9 to scenario 12, the overall performance of the proposed method is similar to the comparison methods, only significantly better than the two compared methods. For the R-IGD metric, according to the average rank, the proposed method gets the best performance by a small margin. From the table, the proposed is significantly better than ar-NSGP on 11 scenarios. At the same time, WAr-NSGP is better than WASF-NSGP and LSG-NSGP on 2 scenarios. In addition, the proposed achieved similar performance to *r*-NSGP on all scenarios. To summary, the above results verify the effectiveness of dispatching rules learned by the proposed algorithm.

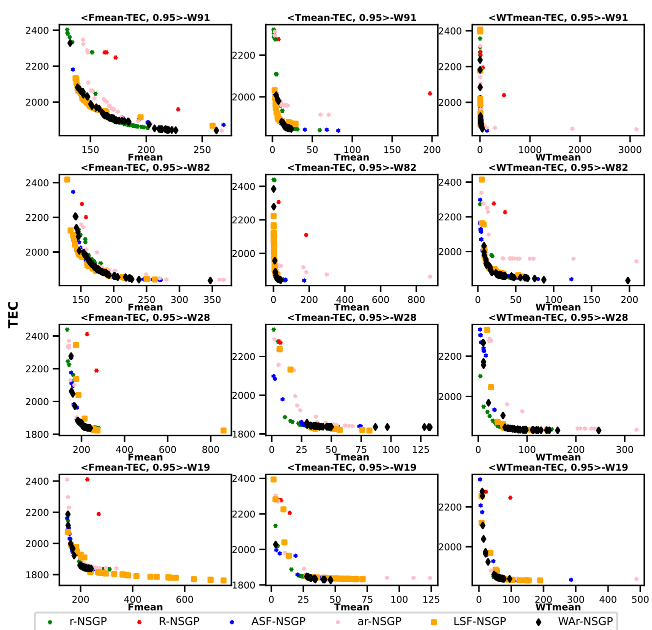

Fig. 3. Learned PF of the run with medium updated R-HV of six algorithms on 12 training scenarios.

Table IV shows the mean and standard deviation of updated R-HV and R-IGD of the proposed method and four compared methods in the test process. On the whole, the experimental results of WAr-NSGP show better effectiveness than the comparison algorithms. For updated R-HV, according to the average Friedman rank, the proposed method goes for the best performance. Specifically, the proposed method gets the best performance among the compared methods on 7 scenarios. Consistent with the training results, the training results of WAr-NSGP are significantly better than those of R-NSGP. The proposed method is significantly better than the four compared methods on the 2 scenarios. In addition to

TABLE IV
THE MEAN (STANDARD DEVIATION) OF THE UPDATED R-HV AND R-IGD ON 30 **TEST INSTANCES** OF WAR-NSGP WITH FIVE COMPARED ALGORITHMS BASED ON 30 INDEPENDENT RUNS IN 12 SCENARIOS.

Scenario	*r*-NSGP	R-NSGP	WASF-NSGP	ar-NSGP	LSF-NSGP	WAr-NSGP
	Updated R-HV of Test					
1	0.018(0.015)(↑)	0.002(0.003)(↑)	0.032(0.018)(≈)	0.020(0.014)(↑)	**0.040(0.016)**(≈)	0.037(0.021)
2	0.083(0.039)(≈)	0.013(0.019)(↑)	0.095(0.031)(≈)	0.074(0.033)(↑)	0.090(0.034)(≈)	**0.093(0.036)**
3	0.077(0.029)(≈)	0.007(0.015)(↑)	0.081(0.033)(≈)	0.085(0.034)(≈)	0.088(0.027)(≈)	**0.090(0.028)**
4	0.033(0.014)(≈)	0.003(0.009)(↑)	**0.038(0.015)**(≈)	0.037(0.021)(≈)	0.033(0.015)(≈)	0.036(0.014)
5	0.060(0.018)(↑)	0.012(0.011)(↑)	0.060(0.015)(↑)	0.059(0.020)(↑)	0.063(0.019)(≈)	**0.069(0.016)**
6	0.135(0.033)(≈)	0.031(0.032)(↑)	0.131(0.038)(≈)	0.125(0.039)(≈)	0.137(0.038)(≈)	**0.142(0.035)**
7	**0.093(0.043)**(≈)	0.017(0.033)(↑)	0.088(0.035)(≈)	0.085(0.031)(≈)	0.087(0.037)(≈)	0.085(0.042)
8	0.041(0.016)(↑)	0.006(0.012)(↑)	0.041(0.021)(↑)	0.047(0.020)(≈)	0.036(0.015)(↑)	**0.067(0.039)**
9	0.063(0.017)(≈)	0.009(0.011)(↑)	0.063(0.013)(≈)	**0.068(0.021)**(≈)	0.064(0.014)(≈)	0.067(0.023)
10	0.136(0.034)(≈)	0.024(0.022)(↑)	0.132(0.034)(≈)	0.131(0.034)(↑)	0.133(0.032)(≈)	**0.149(0.032)**
11	0.095(0.028)(≈)	0.011(0.031)(↑)	0.096(0.033)(≈)	0.089(0.029)(≈)	**0.096(0.030)**(≈)	0.091(0.035)
12	0.050(0.024)(≈)	0.008(0.017)(↑)	0.051(0.020)(≈)	0.050(0.019)(≈)	0.048(0.019)(≈)	**0.052(0.020)**
Win / Draw / Lose	3 / 9 /0	12 / 0 /0	2 / 10 / 0	4 / 8 / 0	1 / 11 / 0	N/A
Average Rank	3.14	5.85	3.08	3.27	3.01	**2.65**
	R-IGD of Test					
1	0.067(0.059)(≈)	0.310(0.166)(↑)	0.055(0.042)(≈)	0.078(0.107)(≈)	0.056(0.041)(≈)	**0.043(0.045)**
2	0.075(0.070)(≈)	0.323(0.155)(↑)	0.069(0.054)(≈)	0.052(0.047)(≈)	0.072(0.050)(≈)	**0.049(0.040)**
3	0.090(0.096)(↑)	0.347(0.139)(↑)	0.116(0.110)(↑)	**0.029(0.027)**(≈)	0.117(0.109)(↑)	0.044(0.046)
4	0.097(0.073)(↑)	0.364(0.137)(↑)	0.114(0.117)(↑)	0.048(0.051)(≈)	0.113(0.101)(↑)	**0.042(0.029)**
5	0.058(0.062)(↑)	0.177(0.115)(↑)	0.063(0.070)(↑)	0.043(0.062)(≈)	0.060(0.072)(≈)	**0.034(0.045)**
6	0.042(0.043)(≈)	0.211(0.133)(↑)	0.062(0.073)(≈)	0.047(0.064)(≈)	0.054(0.058)(↑)	**0.029(0.025)**
7	0.086(0.082)(↑)	0.269(0.152)(↑)	0.100(0.094)(↑)	**0.035(0.060)**(≈)	0.095(0.097)(↑)	0.052(0.072)
8	0.110(0.089)(↑)	0.247(0.120)(↑)	0.084(0.083)(↑)	**0.028(0.040)**(≈)	0.105(0.078)(↑)	0.049(0.059)
9	0.030(0.030)(≈)	0.226(0.161)(↑)	0.028(0.019)(≈)	0.034(0.036)(↑)	0.047(0.044)(↑)	**0.025(0.024)**
10	0.048(0.048)(≈)	0.268(0.160)(↑)	0.043(0.049)(≈)	**0.040(0.055)**(≈)	0.046(0.052)(≈)	0.042(0.041)
11	0.056(0.051)(≈)	0.301(0.150)(↑)	0.076(0.075)(≈)	**0.034(0.049)**(≈)	0.059(0.064)(≈)	0.058(0.069)
12	0.108(0.085)(↑)	0.294(0.162)(↑)	0.075(0.077)(≈)	**0.021(0.017)**(≈)	0.082(0.074)(≈)	0.053(0.076)
Win / Draw / Lose	6 / 6 /0	12 / 0 /0	5 / 7 / 0	1 / 11 / 0	6 / 6 / 0	N/A
Average Rank	3.23	5.72	3.20	2.72	3.52	**2.60**

achieving approximate results with the compared algorithms, WAr-NSGP is superior to *r*-NSGP, WASF-NSGP and ar-NSGP on 3, 2 and 4 scenarios respectively. For the R-IGD, according to the average Friedman rank, the proposed method goes for the best performance. The proposed method is significantly better than R-NSGP. This is consistent with the performance shown by the training results. From the results, WAr-NSGP is better than all compared algorithms in half the scenarios. To be specific, the proposed method outperforms *r*-NSGP, WASF-NSGP, ar-NSGP and LSF-NSGP on 6, 5, 1 and 6 scenarios.

This paper saves the PF at 10-generation intervals throughout the evolution process. From Fig. 3, the proposed algorithm achieves candidate solutions closer to PF in scenario 1, scenario 3, scenario 4, scenario 8, scenario 9, scenario 10, scenario 11, and scenario 12. The performance similar to that of the comparison algorithm is also shown in other scenarios. In general, from the perspective of updated R-HV and R-IGD, WAr-NSGP achieved the best performance result overall. This verifies the effectiveness and robustness of the proposed method. Besides the above-mentioned, from the consistency analysis of training results and test results, the proposed algorithm shows better performance in the test phase than in the training phase. This means that WAr-NSGP is more promising than other compared algorithms.

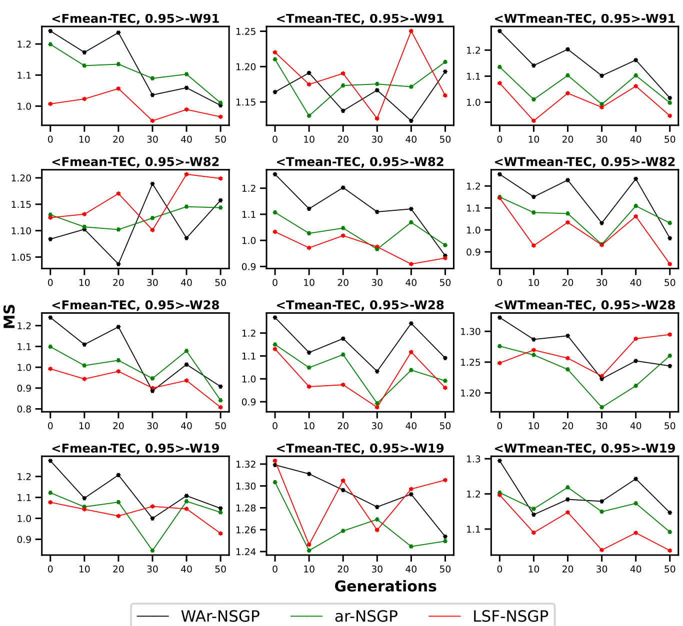

Fig. 4. The curves of average MS in the evolutionary process of proposed multiple dominance criterion with two compared multiple dominance criteria based on 30 independent runs in 12 scenarios.

B. Quality Evaluation of Candidate Solutions during Training

As mentioned before, we expect to design an algorithm that can maintain better diversity in the early stages of evolution. In this paper, diversity is measured by analyzing how populations cover PF. If the PF of the population can be distributed more widely, it indicates that the population has a good diversity. Therefore, the Maximum Spread metric is chosen as the quality indicator to measure diversity. It can be defined as follows:

$$MS(f(x_i)) = \sqrt{\sum_{j=1}^{m} \max\{f_j(x_i) - f_j(x_i')\}} \tag{20}$$

where m denotes the number of objectives. MS evaluates the range of the solution set by considering the maximum region for each objective. The higher values indicate a wider obtained region, meaning more diversification.

Fig. 4 shows the diversity analysis of the proposed criterion and the two multiple preference dominance criteria on the three scenarios with four user preferences. Since GP does not search the solution space directly but the heuristic space, instance rotation can significantly improve the generalisation ability of the algorithm. In addition, in 11 scenarios (except scenario 2), the proposed dominance criterion maintained better diversity early in their evolution (0-20 generations) relative to both composite dominance criteria. This verifies that the proposed dominance criterion can maintain diversity during the early stages of the evolution process. It is beneficial for the proposed algorithm to maintain diversity in the early stage of evolution, it cannot only find high-quality solutions in the later stage of training, but also the solutions found could obtain better performance in the test stage. In other words, in GP for E-DFJSS, maintaining good diversity in the early stage of evolution can promote the effectiveness of solutions.

VI. Conclusions and Future Work

In this paper, we proposed a preference-based multi-objective GP algorithm for solving E-DFJSS. Then, it addresses the limitation of diversity and convergence balance embedded in the framework by existing dominance criterion. A GP algorithm that sorts the individuals based on multiple dominance criteria is proposed. Experiment results verify that the proposed algorithm can solve the E-DFJSS problem by trade-off production effectiveness and energy consumption better than the state-of-the-art compared algorithms. Moreover, we verify that the proposed dominance criterion can maintain the diversity of the evolution process for effective exploitation. In addition, experimental results show that the proposed algorithm has promising generalisation ability.

In the future, the use of more or novel preference-based multiple dominance criteria will be considered. Moreover, preference studies based on multiple reference points will be considered and more objectives derived from scheduling will be investigated.

References

[1] F. Zhang, G. Shi, Y. Mei, and M. Zhang, "Multiobjective dynamic flexible job shop scheduling with biased objectives via multitask genetic programming," *IEEE Transactions on Artificial Intelligence*, vol. 6, no. 1, pp. 169–183, 2025.

[2] R. Zhang and R. Chiong, "Solving the energy-efficient job shop scheduling problem: a multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption," *Journal of Cleaner Production*, vol. 112, pp. 3361–3375, 2016.

[3] F. Zhang, Y. Mei, and M. Zhang, "Evolving dispatching rules for multi-objective dynamic flexible job shop scheduling via genetic programming hyper-heuristics," in *Proceedings of the Congress on Evolutionary Computation*, pp. 1366–1373, 2019.

[4] K. Deb and J. Sundar, "Reference point based multi-objective optimization using evolutionary algorithms," in *Proceedings of the Annual Conference on Genetic and Evolutionary Computation*, p. 635–642, 2006.

[5] F. Zhang, Y. Mei, S. Nguyen, and M. Zhang, "Multitask multiobjective genetic programming for automated scheduling heuristic learning in dynamic flexible job-shop scheduling," *IEEE Transactions on Cybernetics*, vol. 53, no. 7, pp. 4473–4486, 2023.

[6] F. Zhang, Y. Mei, S. Nguyen, and M. Zhang, "Survey on genetic programming and machine learning techniques for heuristic design in job shop scheduling," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 1, pp. 147–167, 2024.

[7] F. Zhang, S. Nguyen, Y. Mei, and M. Zhang, *Genetic Programming for Production Scheduling*. Springer, 2021.

[8] P. Zhao, L. Wang, Z. Fang, X. Pan, and Q. Qiu, "Preference-based multi-objective evolutionary algorithm with linear combination scalarizing function and reference point adjustment," *Applied Soft Computing*, vol. 153, p. 111296, 2024.

[9] J. Yi, J. Bai, H. He, J. Peng, and D. Tang, "ar-moea: A novel preference-based dominance relation for evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 788–802, 2019.

[10] S. Bechikh, M. Kessentini, L. B. Said, and K. Ghédira, "Chapter four - preference incorporation in evolutionary multiobjective optimization: A survey of the state-of-the-art," vol. 98, pp. 141–207, 2015.

[11] L. Xin, L. Wenjian, G. Naijie, and Z. Qingfu, "A novel dynamic reference point model for preference-based evolutionary multiobjective optimization," *Complex & Intelligent Systems*, vol. 9, pp. 1415–1437, 2023.

[12] K. Li, M. Liao, K. Deb, G. Min, and X. Yao, "Does preference always help? a holistic study on preference-based evolutionary multiobjective optimization using reference points," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 6, pp. 1078–1096, 2020.

[13] M. Li and X. Yao, "Quality evaluation of solution sets in multiobjective optimisation: A survey," *ACM Computing Surveys*, vol. 52, no. 2, 2019.

[14] L. Ben Said, S. Bechikh, and K. Ghedira, "The r-dominance: A new dominance relation for interactive evolutionary multicriteria decision making," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 801–818, 2010.

[15] J. Molina, L. V. Santana, A. G. Hernández-Díaz, C. A. Coello Coello, and R. Caballero, "g-dominance: Reference point based dominance for multiobjective metaheuristics," *European Journal of Operational Research*, vol. 197, no. 2, pp. 685–692, 2009.

[16] J. Zheng, N. Lai, and G. Guo, "ε-pareto dominance strategy based on angle preference in moea," *Pattern Recognition and Artificial Intelligence*, vol. 27, no. 6, pp. 569–576, 2014.

[17] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation discussion and generalization," in *Proceedings of the International Conference on Genetic Algorithms*, p. 416–423, 1993.

[18] R. S. Ana Belén Ruiz and M. Luque, "A preference-based evolutionary algorithm for multiobjective optimization: the weighting achievement scalarizing function genetic algorithm," *Journal of Global Optimization*, vol. 62, pp. 101–129, 2015.

[19] J. Zou, Q. Yang, S. Yang, and J. Zheng, "Ra-dominance: A new dominance relationship for preference-based evolutionary multiobjective optimization," *Applied Soft Computing*, vol. 90, p. 106192, 2020.

[20] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, "A preference-based evolutionary algorithm for multi-objective optimization," *Evolutionary Computation*, vol. 17, no. 3, pp. 411–436, 2009.

[21] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, 2015.

[22] F. Zhang, Y. Mei, and M. Zhang, "Genetic programming with multi-tree representation for dynamic flexible job shop scheduling," in *Proceedings of the Advances in Artificial Intelligence*, pp. 472–484, 2018.

[23] W. Li, L. He, and Y. Cao, "Many-objective evolutionary algorithm with reference point-based fuzzy correlation entropy for energy-efficient job shop scheduling with limited workers," *IEEE Transactions on Cybernetics*, vol. 52, no. 10, pp. 10721–10734, 2022.

[24] D. Yadav, P. Ramu, and K. Deb, "An updated performance metric for preference-based evolutionary multi-objective optimization algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference*, p. 612–620, 2024.

[25] K. Li, K. Deb, and X. Yao, "R-metric: Evaluating the performance of preference-based evolutionary multiobjective optimization using reference points," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 6, pp. 821–835, 2018.