A Novel Multi-Objective Memetic Algorithm for Multi-Task Fuzzy Flexible Job Shop Scheduling

Shaojin Geng, Yi Mei, *Senior Member, IEEE*, Fangfang Zhang, *Member, IEEE*, Lei Wang, *Member, IEEE*, Qidi Wu, *Senior Member, IEEE*, and Mengjie Zhang, *Fellow, IEEE*

***Abstract*—Multi-task multi-objective fuzzy flexible job shop scheduling (FFJSS), different from traditional multi-objective FFJSS, offers the advantage of simultaneously addressing distinct multi-objective scheduling tasks. Evolutionary multi-task is a popular optimization paradigm conserves computational resources and enhances algorithm performance by sharing and leveraging knowledge between different tasks. However, we face the challenges of both multi-task and multi-objective optimization. To address these challenges, we propose a novel multi-objective memetic algorithm (MOMA). First, we design a unified representation with two vectors for both routing and sequencing decisions, which leads to a unified search space within the multifactorial framework. We then propose a dual-neighborhood multifactorial transfer strategy by adjusting the size of the neighborhoods adaptively, which can effectively balance the convergence and diversity of the population and identify transferrable solutions. Additionally, an improved neighborhood update strategy is proposed to improve the rate of positive transfer. To verify the effectiveness of the proposed algorithm, eight distinct multi-task FFJSS scenarios are constructed based on existing FFJSS datasets. The results show that MOMA consistently outperformed the state-of-the-art algorithms across a wide range of test instances. Further analyses of the transfer process have shown that complex tasks can benefit from simple tasks. .**

***Index Terms*—Multi-task multi-objective fuzzy flexible job shop scheduling, evolutionary multi-task optimization, multifactorial evolutionary algorithm, knowledge transfer.**

I. Introduction

THE multi-objective fuzzy flexible job shop scheduling (FFJSS) [1], [2], [3] is a complex combinatorial optimization problem. The goal is to allocate multiple jobs in a workshop environment with different machines, minimizing objectives such as fuzzy makespan and fuzzy energy consumption while satisfying constraints such as job priority, machine availability, and operation dependencies. Two main decisions must be addressed simultaneously: **routing** which involves allocating operations to machines, and **sequencing** which involves determining the order of operations on each machine. The inherent uncertainty in operation processing times within real-world production environments led to a surge of interest in flexible job shop scheduling problems that incorporate fuzzy processing times. Recently, the interval type-2 fuzzy logic system [4] has been used to better predict the uncertainty of timing constraints. Heuristic algorithms [5], [6], [7] have effectively solved multi-objective FFJSS problems.

In the real world, one may face a wide range of FFJSS problems, which share most characteristics but are different in objectives, constraints, problem size and variable distributions. However, existing algorithms for multi-objective FFJSS problems typically address each problem in isolation when optimizing new problems, but solving them in isolation could lead to resource wastage and inefficiencies. In large-scale manufacturing facilities, such as those producing smartphones and smartwatches, different product lines often require simultaneous scheduling. Although the operations and machine configurations may vary, they typically exhibit structural similarities in job routes and machine availability. These similarities allow knowledge transfer across scheduling tasks, making them well-suited to be modeled as multi-task multi-objective FFJSS problems. Solving such problems within a unified framework not only improves resource utilization but also enhances overall scheduling efficiency.

Evolutionary multi-task optimization [8], [9], [10] is an emerging framework that solves multiple similar problems simultaneously by improving algorithm performance through knowledge transfer, which has been successfully applied to different practical problem areas [11], [12], [13]. The Multifactorial Evolutionary Algorithm (MFEA), first proposed by Gupta et al. [14], is a widely adopted framework for evolutionary multi-task optimization. One of its key features is knowledge transfer-that is, the process of leveraging genetic information from one task to assist optimization in other tasks by exploiting inter-task complementarities within a unified search space—which enables effective information sharing among tasks through mechanisms such as assortative mating and vertical cultural transmission. To successfully extend the transfer mechanism to FFJSS problems, we need to focus on the following key factors.

Received 19 May 2025; revised 15 July 2025; accepted 13 August 2025. *(Corresponding author: Lei Wang.)*

Shaojin Geng is with the College of Electronic and Information Engineering, Tongji University, Shanghai 201804, China, and also with the Centre for Data Science and Artificial Intelligence & School of Engineering and Computer Science, Victoria University of Wellington, Wellington 6140, New Zealand (e-mail: shaojin_geng@163.com).

Yi Mei, Fangfang Zhang, and Mengjie Zhang are with the Centre for Data Science and Artificial Intelligence & School of Engineering and Computer Science, Victoria University of Wellington, Wellington 6140, New Zealand (e-mail: yi.mei@ecs.vuw.ac.nz; fangfang.zhang@ecs.vuw.ac.nz; mengjie.zhang@ecs.vuw.ac.nz).

Lei Wang and Qidi Wu are with the College of Electronic and Information Engineering, Tongji University, Shanghai 201804, China (e-mail: wanglei@tongji.edu.cn; qidi@tongji.edu.cn).

Recommended for acceptance by C. VONG.

Digital Object Identifier 10.1109/TETCI.2025.3603106

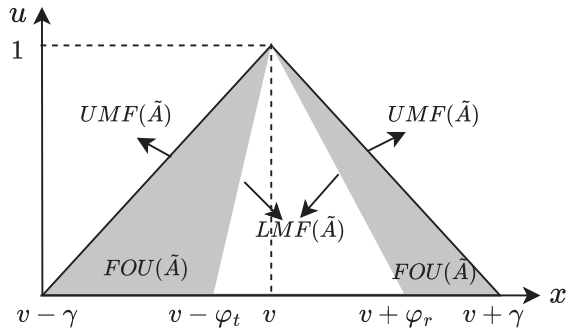


Fig. 1. Nonsymmetrical triangular interval type-2 fuzzy system.

The motivation of this work lies in improving the optimization performance for FFJSS problem through adaptive and efficient multi-task knowledge transfer. First, MFEA requires designing a unified representation for all problems to be solved, which enables individuals to be simultaneously applicable across the search spaces of multiple tasks. As we all know, different FFJSS problems often necessitate different genetic representations due to the different number of jobs and operations, coupled with diverse machine availability for each operation. Existing works on solving multi-task combinatorial optimization problems such as vehicle routing [15], [16], [17], dynamic flexible job shop scheduling [18], [19], [20], feature selection [21], [22], [23], each with its own customized unified representation. However, the characteristics of existing problems are different from FFJSS, making these encoding schemes inapplicable. Therefore, a new unified encoding scheme needs to be designed. Second, decomposition-based multi-objective algorithms [24] address problems by dividing them into single-objective subproblems, optimizing the problem through problem simplification and collaborative optimization of neighboring subproblems. While existing decomposition-based algorithms have been applied to multi-task area [25], [26], [27], they ignore the influence of neighborhood size and negative transfer individuals on the transfer process. Neighborhood size impacts the balance of convergence and diversity of the population [28], as well as the selection of transferred individuals. Additionally, the potential benefits of negative transfer individuals should be further investigated. This motivates our design of an effective knowledge transfer strategy. Third, it is unclear how the effectiveness of transfer is in different multi-task FFJSS scenarios.

This paper aims to design a multi-objective memetic algorithm for solving multiple FFJSS tasks simultaneously, improving algorithm efficiency and exploring the potential connection between different FFJSS tasks. The specific contributions of this paper are given as follows.

- A unified representation is designed for multi-task FFJSS scenarios, consisting of two vectors for routing and sequencing respectively, that serve as the foundation for knowledge transfer among different FFJSS tasks within a unified search space.
- A novel multi-objective memetic algorithm (MOMA) is proposed. Specifically, we introduce a dual-neighborhood-based transfer strategy that evolves the population using knowledge from both intra-task and inter-task neighborhoods. Then, an adaptive neighborhood size selection mechanism dynamically adjusts the transfer scope based on feedback from convergence and diversity. Furthermore, an improved neighborhood update strategy leverages transferred individuals from a global perspective to improve the rate of positive transfer.
- We design eight different multi-task FFJSS scenarios, varying in terms of job and machine numbers. These scenarios fall into three categories: no difference, small difference, and big difference, with the aim of studying the transfer effects in different scenarios.

The rest of this paper is organized as follows. Section II introduces the problem formulation, the multi-objective multifactorial evolutionary algorithm (MFEA) framework, and related work. Section III provides a detailed description of the proposed algorithm. In Section IV, the experimental design is outlined and the experimental results are discussed and analyzed. Further analysis is presented in Section V. Finally, the paper is summarized in Section VI.

II. Background and Related Work

A. Problem Formulation

Interval Type-2 Fuzzy Sets (IT2FS) extend traditional fuzzy sets by introducing interval-valued membership functions, allowing better modeling of uncertainty in complex environments. In this study, IT2FS are employed to represent the uncertain processing durations of operations, providing a more realistic characterization of uncertainty in manufacturing. As shown in Fig. 1, a typical nonsymmetrical triangular IT2FS is bounded by the Upper and Lower Membership Functions (UMF and LMF), which together define the Footprint of Uncertainty (FOU)—the shaded area representing the interval of possible membership grades for a given input x. This modeling approach captures both the variability and ambiguity of uncertain durations. To compare fuzzy processing times during scheduling, a hierarchical centroid-based method [29] is adopted.

Based on the fuzzy modeling framework, we define the multi-task multi-objective fuzzy flexible job shop scheduling (FFJSS) problem. Specifically, a single-task multi-objective FFJSS problem involves scheduling a set of jobs $J = \{J_1, J_2, J_3, \ldots, J_n\}$ on a set of machines $M = \{M_1, M_2, M_3, \ldots, M_m\}$ to minimize a set of objective functions. Each job J_i consists of a sequence of operations $O = \{O_{11}, O_{12}, O_{13}, \ldots, O_{nt_n}\}$, where O_{nt_n} represents there are t_n operations of job n since different jobs have different operations. Each operation must be processed on different available machines. To be more specific, two main decisions need to be addressed simultaneously: routing, which involves selecting a machine for each operation from its candidate machines; and sequencing, which involves determining the order of operations on each machine. The constraints are given as follows.

1) The processes of each operation must be executed in sequence.
2) Each operation can only be processed on one available machine at any time.

3) Each machine can only execute one operation at a time and cannot be interrupted.

In a multi-task multi-objective FFJSS problem, multiple tasks are considered simultaneously. Each task has its own set of jobs and machines.

- Tasks: $T = \{T_1, T_2, T_3, \ldots, T_q\}$
- Jobs for task T_i: $J_{T_i} = \{J_{T_i,1}, J_{T_i,2}, J_{T_i,3}, \ldots, J_{T_i,n_i}\}$
- Machines for task T_i: $M_{T_i} = \{M_{T_i,1}, M_{T_i,2}, M_{T_i,3}, \ldots, M_{T_i,m_i}\}$
- Operations for job $J_{T_i,j}$: $O_{T_i} = \{O_{T_i,j,1}, O_{T_i,j,2}, O_{T_i,j,3}, \ldots, O_{T_i,j,t_{T_i,j}}\}$

This paper investigates a scheduling problem with two objectives, Makespan and Energy Consumption as performance criteria [30].

1) *Makespan:* Makespan refers to the time required to complete all operators in the entire job shop process, that is, the total time elapsed from the beginning to the completion of all production tasks. The calculation of Makespan objective $Makespan$ is shown as follows:

$$Makespan = \max(C_1, C_2, \ldots, C_m), 1 \leq m \leq M \quad (1)$$

where C_m is the completion time of machine m.

2) *Energy Consumption (EC):* In manufacturing, EC is a key measure of resource efficiency. Lower EC indicates more efficient production, aiding in reducing carbon emissions at the manufacturing system level. The calculation is shown as follows:

$$EC = P_e * \sum_{i=1}^{J} T_i + P_w * \left(\sum_{m=1}^{M} (C_m - E_m) \right) + P_s \quad (2)$$

where P_e is the execution power, P_w is the waiting power, and P_s is the power of on/off machine. The total execution time is the sum of the execution times of all operations $\{T_i, 1 \leq i \leq J\}$ on their machines. Where E_m represents the execution time of all operations on machine m. Note that the idle and on/off switching states are mutually exclusive, and thus their energy P_w and P_s are not double-counted.

Due to the complexity of practical FFJSS problems, intuitively describing similarities between tasks can be challenging. This paper investigates eight distinct multi-task FFJSS scenarios, varying in job and machine numbers: same difference, small difference, and big difference.

B. Multi-Objective Multifactorial Evolutionary Algorithm

The multi-objective multifactorial evolutionary algorithm (MOMFEA) [31] is a commonly used multi-task optimization algorithm using non-dominated sorting and crowding distance in NSGA-II [32] to evaluate individuals. Fig. 2 shows the flowchart of the MOMFEA algorithm. The assortative mating and vertical cultural transmission operators are important parts of knowledge transfer. The skill factor is an attribute of each individual, which represents the task performed by the individual. We assume that there are two tasks T_1&T_2. X and Y are two variables that represent the skill factor of two individuals. RMP is

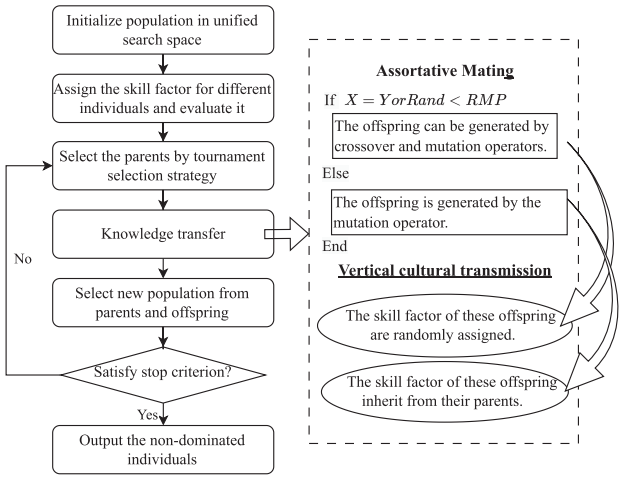


Fig. 2. The flowchart of MOMFEA [31].

the transfer probability. $Rand$ is a random number with the range [0,1]. The offspring is generated by assortative mating. When the skill factors of the parents are the same ($X = Y$), both crossover and mutation operators are used. When the skill factors of the parents are different ($X \neq Y$) and the transfer rate condition is met ($Rand < RMP$), individuals from different tasks perform both crossover and mutation operators, too. When the skill factors are different and the transfer rate condition is not met ($X \neq Y \& Rand \geq RMP$), only mutation is performed. The vertical cultural transmission operator is used to determine the offspring's skill factor. When offspring are generated through crossover and mutation operators, the skill factors of the offspring are randomly assigned. In other words, genetic information can be transferred to different tasks by randomly assigning skill factors. When offspring are generated only by mutation operators, the skill factors of the offspring are the same as those of the parents.

C. Related Work

Evolutionary multi-task optimization [33], [34], [35], [36] has been applied to various combinatorial optimization problems, mainly falling into two categories: one involves solving the original task by constructing auxiliary tasks, and the other focuses on solving different tasks simultaneously.

The primary goal of auxiliary tasks is to accelerate the resolution of the original task, representing a unidirectional process of knowledge transfer. In [37], Ning et al. optimize both the original FJSS task and an auxiliary task simultaneously. The auxiliary task is similar to the original one but differs in initialization and selection strategy. In [38], Li et al. construct an auxiliary task by modifying the set of customers in the original vehicle routing problem. In [39], a multi-task framework combining evolutionary algorithms and mathematical programming is applied to coal mining. The scale and weight vector of multiple auxiliary tasks are dynamically adjusted based on the current population's state. Each auxiliary task is a weighted single-objective problem with the same constraints as the original problem. In [40], task offloading problems in mobile-edge computing networks are

addressed using a multi-task framework. Two types of auxiliary tasks are constructed to independently optimize the overheads of edge and local processing parts, providing scheduling schemes for the original task. In [41], a sampling-based multi-task point cloud registration framework is proposed. Both the original and sampled tasks are derived from the same pair of point clouds, with the sampling registration task reducing the number of points to improve time efficiency, simplify the task, and provide effective knowledge for the original task. In [42], an auxiliary task is constructed with lower computational complexity but high similarity to the main task, enabling knowledge transfer within a multitasking optimization framework to improve the efficiency and performance of solving the target task. While these studies have shown the benefits of incorporating auxiliary tasks to improve performance, they have not tackled the challenge of handling two distinct tasks concurrently.

When solving different tasks, the difference in tasks presents a challenge to the transfer mechanism. In [43], the vehicle routing problem with heterogeneous capacity, time windows, and occasional drivers is studied using a multi-task optimization algorithm. The key difference in task characteristics lies in the customers' time windows. In [44], the number of customers is varied for the same task in the capacitated vehicle routing problem. In [45], the distributed hybrid flow shop scheduling problem is addressed by optimizing two distinct tasks, involving learning job-to-job mapping to associate different tasks and facilitate knowledge transfer. In [46], [47], [48], [49], [50], a dynamic flexible job shop scheduling problem is studied, optimizing two different tasks simultaneously. The multi-task scenario is constructed based on the simulation utilization level. Currently, research on multi-task FFJSS involving multiple different tasks is still in its early stages, indicating a promising area for future exploration and development.

Multi-task optimization algorithms aim to design an efficient transfer strategy, and numerous advanced multi-objective FFJSS algorithms [3], [30], [51] have been proposed. Therefore, this paper further validates the effectiveness of the designed transfer strategy based on an advanced multi-objective FFJSS algorithm. Recently, Li et al. [30] proposed a learning-based reference vector memetic algorithm (LRVMA). This algorithm achieves superior performance by initializing with a high-quality population, employing a set of local search strategies to improve the quality of the offspring, and dynamically adapting neighborhood size through Q-learning to maintain population diversity. These improvements contribute to the algorithm's effectiveness in addressing FFJSS problems.

In the process of designing multi-task FFJSS algorithms, there are two main challenges, which is why existing algorithms cannot be directly applied. The first challenge is the unified representation method. FFJSS requires simultaneous routing and sequencing [52], and the complexity increases when considering the availability of machines for operations and the inconsistency in the number of jobs and operations across different tasks. Yuan et al. [53] verified that permutation-based unified encoding is better than random-key representation for job shop scheduling problems. This approach focuses solely on the sequencing of operations, as the assignment of different operations is fixed. Ning et al. [37] only need to consider the encoding method of the original FJSS task as the unified representation because the auxiliary task is the same as the original task. However, they did not explore the effective knowledge between different tasks. Zhang et al. [54] addressed the multi-task dynamic flexible job shop problem using a genetic programming algorithm. The tree-based representation can effectively transfer knowledge between different tasks without considering the heterogeneity between different tasks. In summary, it is necessary to design a unified representation method suitable for FFJSS.

The second challenge is the issue of negative transfer, which is common in various transfer mechanisms [55], [56], [57]. Specifically, the neighborhood size significantly impacts both the balance between population convergence and diversity and the performance of the transfer within the LRVMA framework. In other words, the selection of parents directly affects the performance of the offspring. There are some studies on evolutionary multi-task optimization based on the MOEA/D framework [24]. In [25], Yao et al. proposed a multifactorial based on decomposition strategy, and the single-objective subproblems with a fast evolution rate can get more computation resources. However, different tasks transfer knowledge through random mating, ignoring the correlation between different tasks, resulting in negative knowledge transfer. In [26], Lin et al. consider the potential of each solution and propose the hybrid transfer strategy to promote knowledge transfer. However, the impact of neighborhood size is not addressed. Wang et al. [27] further explore the knowledge between different tasks by considering the internal and external neighborhood. However, there is still negative knowledge transfer in the process of information interaction between internal and external neighborhoods. At the same time, the size of the neighborhood also directly affects the amount of information interaction between different task sub-problems. Therefore, the selection of neighborhood size is crucial.

In summary, research on multi-task FFJSS is still in its infancy and still faces challenges in representation and negative transfer. To address these issues, this paper proposes a novel multi-objective memetic algorithm (MOMA), which aims to improve algorithm performance by constructing a unified representation and improving the transfer strategy.

III. Proposed Algorithm

A. Framework

The overall framework is extended from the existing LRVMA [30], and introduces several newly designed modules, including initialization, dual-neighborhood construction and update, knowledge transfer strategy, and Q-learning-based neighborhood size adjustment. As shown in Fig. 3, the algorithm begins with initializing multiple populations, each corresponding to a different task. For each individual, a dual-neighborhood is constructed: the internal neighborhood consists of individuals with the T closest weight vectors in the same task, based on Euclidean distance, while the external neighborhood includes individuals transferred from other tasks. The external neighborhood is dynamically updated based on the performance of

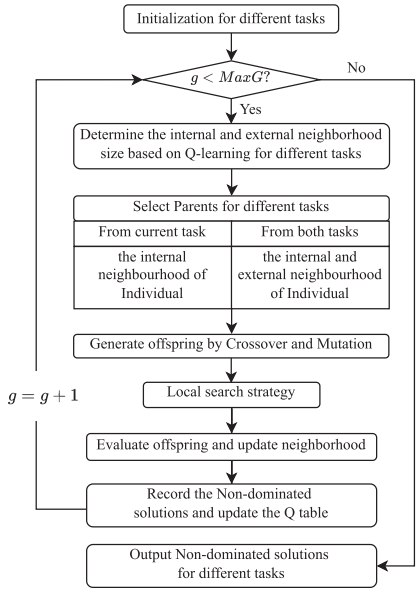


Fig. 3. The flowchart of the proposed MOMA. (1) Initialization via unified representation; (2) Q-learning module for adaptively determining neighborhood size; (3) Dual-neighborhood-based transfer strategy for parent selection and offspring generation; (4) Improved neighborhood update strategy for population evolution.

transferred individuals, as described in Section III-F. To enable effective inter-task knowledge transfer, all individuals share a unified representation across tasks. At each generation, parents are selected from the dual-neighborhood to perform crossover and mutation operations. Offspring quality is further enhanced through a local search procedure [6]. The optimal neighborhood size is adaptively adjusted using Q-learning, and the Q-table is updated accordingly. The populations evolve iteratively until the termination condition is met, and the non-dominated solutions for each task are then output.

B. Unified Representation for Multi-Task FFJSS

For the multi-task FFJSS problem, routing and sequencing are two crucial decisions. To address the structural heterogeneity of multi-task FFJSS, we propose a unified representation that encodes routing and sequencing decisions in a shared format across all tasks. This design resolves representation incompatibilities arising from differences in jobs, operations, and machines, enabling all tasks to be expressed within a common search space and supporting effective cross-task knowledge transfer. Therefore, the unified representation can be composed of two parts:

- *A real-valued vector for routing:* The dimension is calculated as: $DR_{\max} = \max_{m\in(1,K)}(\sum_{n=1}^{j_m} I_{mn})$, where I_{mn} denotes the number of operations in the nth job of the mth task, and j_m is the number of jobs in task m. Each dimension takes a value in the interval [0,1]. During decoding, the routing chromosome of task m is extracted by selecting the first $|O_m|$ dimensions, where $|O_m|$ is the total number of operations in task m. Each real-valued gene $x \in [0, 1]$ is converted into a discrete machine index using $i = \lfloor x \cdot s \rfloor + 1$, where s denotes the number of candidate machines for the corresponding operation, and $\lfloor \cdot \rfloor$ is the floor function.
- *A permutation-based vector for sequencing:* The dimension is calculated as: $DS_{\max} = \sum_{n=1}^{\max(j_m)} \max_{m\in(1,K)}(I_{mn})$. This constructs a unified search space covering the maximum number of operations for each job across all tasks. Each dimension represents a single operation, encoded by the index of the corresponding job. When decoding, the kth occurrence of a job index from left to right is interpreted as the kth operation of that job. For example, if job J_3 appears for the second time at a certain position, that position will correspond to the second operation of J_3.

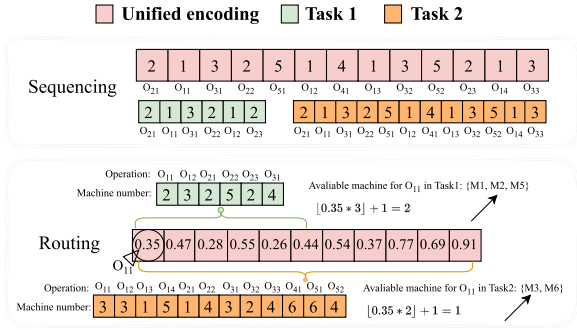


Fig. 4. An example of unified encoding and decoding. It shows how the unified sequencing and routing vectors are mapped to task-specific operation sequences and machine assignments for different tasks.

TABLE I
AN EXAMPLE OF MULTI-TASK FFJSS INSTANCE

Job	Task	Operation	M1	M2	M3	M4	M5	M6
Job1	Task1	O_{11}	✓	✓			✓	
		O_{12}		✓	✓	✓		✓
	Task2	O_{11}			✓			✓
		O_{12}	✓		✓	✓		
		O_{13}	✓			✓		
		O_{14}		✓	✓		✓	✓
Job2	Task1	O_{21}		✓			✓	
		O_{22}			✓		✓	✓
		O_{23}	✓	✓			✓	✓
	Task2	O_{21}	✓		✓			✓
		O_{22}			✓	✓	✓	
Job3	Task1	O_{31}		✓	✓	✓	✓	✓
	Task2	O_{31}		✓	✓	✓		
		O_{32}	✓	✓			✓	✓
		O_{33}	✓		✓	✓		
Job4	Task1							
	Task2	O_{41}		✓	✓			✓
Job5	Task1							
	Task2	O_{51}				✓	✓	✓
		O_{52}	✓	✓		✓		✓

Fig. 4 presents a unified encoding and decoding example based on Table I. The pink blocks denote the unified chromosome, while green and brown blocks represent Task1 and Task2, respectively. The sequencing vector has 13 positions, determined by the maximum number of operations per job across all tasks. During decoding, job IDs are sequentially scanned and assigned to each task based on its job-operation structure. The routing vector has 12 values, equal to the total operations in Task2—the largest among all tasks. The first 6 values are used for Task1 and all 12 for Task2. As shown, the value 0.35 maps to the second machine (among three) for a Task1 operation, and the first machine (among two) for Task 2, depending on the number of candidate

Algorithm 1: Dual-Neighborhood-Based Multifactorial Transfer Strategy.

Input: The population Pop, rate of knowledge transfer δ, current task i, current individual j, dual-neighborhood size T, the internal neighborhood Nei_{In}, the external neighborhood Nei_{Ext}.
Output: The offspring O_1, O_2

```
if Rand() < δ then
    I1 ← Nei_In(i, j)[1 : T] with skill factor τ1
    I2 ← Nei_Ext(i, j)[1 : T] with skill factor τ2
    I ← I1 ∪ I2
    [P1, P2] ← Select two parents randomly from I
    [O1, O2] ← Crossover and Mutation in unified search space (P1, P2).
    Assign skill factors to [O1, O2] randomly.
else
    I ← Nei_In(i, j)[1 : T] with skill factor τ1.
    [P1, P2] ← Select two parents randomly from I
    [O1, O2] ← Crossover and mutation (P1, P2).
    The skill factor of [O1, O2] is τ1.
```

machines. This unified representation resolves incompatibilities arising from structural differences among tasks. By mapping all tasks into a unified search space, it enables consistent genetic operations and effective cross-task knowledge transfer, thereby addressing the challenge of heterogeneous task structures in multi-task optimization.

C. Initialization

To construct high-quality and diverse populations for each task, we employ five different initialization strategies, each generating approximately $N/5$ individuals. Each individual consists of an operation sequence (P_{chrom}) and a machine assignment (M_{chrom}), which are initialized based on heuristics such as minimum processing time, machine workload, completion time, machine availability, and randomness [30]. Detailed implementations are provided in the Section S-I of supplementary material.

All individuals are represented in two forms: a task-specific representation and a unified representation. The unified representation is used solely during knowledge transfer. Additionally, each individual maintains two neighborhoods: an internal neighborhood (fixed) and an external neighborhood (adaptively updated via an improved neighborhood update strategy).

D. Q-Learning for Selecting Dual-Neighborhood Size

In multi-task optimization, neighborhood size influences both population convergence and the scope of inter-task knowledge transfer. To address this, we adopt Q-learning—a lightweight, model-free reinforcement learning method capable of learning optimal adjustment strategies through reward feedback. Its simplicity and low parameter dependency make it well-suited for dynamically controlling internal and external neighborhood sizes. To formalize this process, we define the Q-learning mechanism [58] as follows:

At each generation g, the agent first observes the current state S_g, reflecting the convergence and diversity status of the population. It then selects an action A_g from a predefined set of neighborhood sizes. After one generation of evolution, the agent receives a scalar reward r_{g+1}, and the population transitions to a new state S_{g+1}. The Q-table is subsequently updated based on this feedback using the following equation:

$$Q(s_g, a_g) = Q(s_g, a_g) + \alpha \Big[r_{g+1} + \gamma \max_{a'} Q(s_{g+1}, a') - Q(s_g, a_g) \Big] \tag{3}$$

where $Q(s_g, a_g)$ is the current value of taking action a_g in state s_g, r_{g+1} is the immediate reward, and $\gamma \in [0, 1]$ is the discount factor that balances short-term and long-term rewards. The learning rate $\alpha \in (0, 1]$ determines the update step size. Following the general Q-learning framework, we define the components of our model as follows:

- *State S_g:* Represents the change of the Pareto front from generation $g-1$ to g, quantified by convergence improvement ΔC (the average Euclidean distance between current non-dominated solutions and the reference set), and diversity improvement ΔD (variance of adjacent solution distances, where lower variance indicates better distribution uniformity). According to the signs of ΔC and ΔD, four discrete states are defined:(1) $\Delta C \le 0$, $\Delta D \le 0$;(2) $\Delta C \le 0$, $\Delta D > 0$;(3) $\Delta C > 0$, $\Delta D \le 0$;(4) $\Delta C > 0$, $\Delta D > 0$.
  These states reflect different convergence-diversity patterns of the population.
- *Action A_g:* Refers to the neighborhood size selected at generation g, drawn from the action space $A = \{5, 10, 15, 20\}$.
- *Reward r_{g+1}:* A scalar reward designed to encourage diversity improvement, following the parameter setting in [30], the reward can be defined as:

$$r_{g+1} = \begin{cases} 10, & \text{if } \Delta D > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

- *Q-table:* The Q-table is a 4×4 matrix, where each row corresponds to a discrete state and each column to a possible action. Each entry stores the expected value of taking an action in a given state and is iteratively updated via Equation (3) based on the observed reward and estimated future return.

E. Dual-Neighborhood-Based Multifactorial Transfer Strategy

Algorithm 1 outlines the proposed dual-neighborhood-based multifactorial transfer strategy. It takes as input: the population *Pop* with multiple tasks, the knowledge transfer rate δ, the current task index i, the current individual j, and the total number of tasks TN. For each individual, the dual-neighborhood of size T is composed of: (1) the internal neighborhood Nei_{in}, comprising individuals from the same task; (2) the external neighborhood Nei_{ext}, dynamically updated based on the performance of transferred individuals from other tasks. The offspring $[O_1, O_2]$ are generated using the individuals in the selected neighborhoods. The probability δ controls whether to include inter-task individuals for crossover. When $Rand() < \delta$, parents are selected from both neighborhoods to increase diversity. Otherwise, the crossover and mutation are performed within the internal neighborhood.

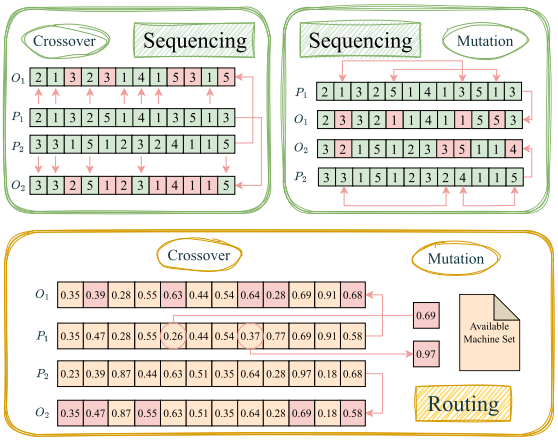


Fig. 5. An example of crossover and mutation operation for sequencing and routing process.

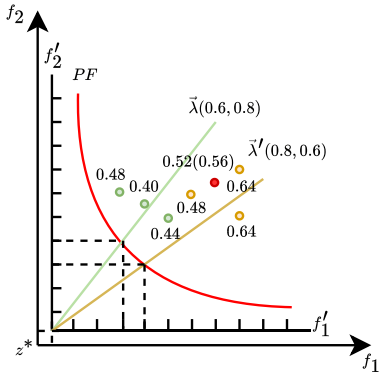


Fig. 6. An example of a transferred individual performing better on other subproblems rather than current subproblem.

We extend the crossover and mutation operators for routing and sequencing, employed in [30], to the multi-task domain based on the proposed unified representation. For sequencing, a precedence operation crossover (POX) operator is used. There is a set of jobs J, which includes all the jobs from multiple tasks. The $J1$ and $J2$ are two randomly generated subsets with $J1 \cap J2 = \emptyset$. Select jobs from sets $J1$ and $J2$ in parents $P1$ and $P2$, respectively, and copy them to offspring $O1$ and $O2$. Fill the remaining positions in each offspring with operations from the opposite parent not already present, preserving the original order. Mutation involves swapping two randomly selected positions within a parent's sequence. For routing, a uniform crossover (UX) is applied to exchange genes at corresponding positions between individuals. Mutation is performed by randomly selecting two positions and substituting the existing machines with new ones chosen from the corresponding candidate machine sets.

Fig. 5 shows the crossover and mutation operator details, where the green box is for sequencing, and the yellow box is for routing. In the left green box, job sets $J1$ and $J2$ are defined as $1, 2, 4$ and $3, 5$, respectively. In offspring O_1, green squares represent jobs inherited from parent P_1, while red squares indicate jobs inherited from parent P_2. Conversely, in offspring O_2, green squares originate from P_2, and red squares from P_1. In the right green box, two positions are randomly selected within both $P1$ and $P2$. The resulting exchanged sequences are displayed in the red squares of $O1$ and $O2$. For the yellow box, the red square in O_1 and O_2 is from the corresponding position in $P2$ and $P1$. For mutation, the machines at positions 0.26 and 0.37 are randomly chosen for replacement. New machines, 0.69 and 0.97, are selected from their corresponding candidate machine sets and assigned to these positions.

F. Improved Neighborhood Update Strategy

The motivation of this strategy is the potential value of the transferred individuals in the entire problem domain of the task, rather than in a single subproblem. The unified search space enables the generated offspring to inevitably contain information from other tasks, and combined with local search, the quality of the offspring is further improved. Therefore, the evaluation of the offspring should be from a global perspective.

In the proposed algorithm, the optimization involves two objectives: makespan and energy consumption. To handle them simultaneously, we adopt a decomposition-based strategy. Specifically, the multi-objective problem is decomposed into a set of scalar subproblems, each associated with a reference weight vector. During the search process, each solution is evaluated using the Tchebycheff aggregation function [24]:

$$g^{tche} = \max_{1 \le i \le m} \left\{ \lambda_i^j | f_i(x) - Z_i^* | \right\} \tag{5}$$

where g^{tche} is the scalar fitness of solution x, m is the number of objectives, λ^j is the weight vector of subproblem j, and Z^* is the ideal point in the objective space. This formulation converts the multi-objective vector into a scalar value, enabling direct fitness comparison among individuals within the neighborhood of each subproblem. Whether the Nei_{In} and Nei_{Ext} of the current individual will be updated depending on the skill factor of the offspring. If the skill factor of the offspring is the same as the current task, its internal neighborhood is updated. Otherwise, the parent's external neighborhood will be updated. The Tchebutcheff aggregation function compares the offspring's performance to the parent's neighborhood. This comparison guides the update of corresponding individuals in the population.

Fig. 6 illustrates a scenario where a new individual (the red node with the value 0.52(0.56) is generated near subproblem $\vec{\lambda}$ through knowledge transfer. However, it does not exhibit superior performance within its current neighborhood but demonstrates improved performance on another subproblem (the yellow vector $\vec{\lambda}'$). This phenomenon arises from the conflicting nature of multiple objectives, making it challenging to achieve optimal performance in a fixed evolutionary direction. To address the above problem, the individuals generated by transfer need to consider the potential value for other subproblems as they are generated in a larger search space with a diverse performance. Finally, if the external neighborhood remains unchanged, new neighbors are generated by selecting the individuals from the target task randomly. Otherwise, the updated neighborhood is preserved.

TABLE II
MULTI-TASK FFJSS SCENARIOS

Jobs \ Machines	No Difference	Small Difference	Big Difference
No Difference	\	$< J20M7, J20M8 >$ $< J30M8, J30M9 >$	$< J30M7, J30M10 >$
Small Difference	$< J40M8, J50M8 >$ $< J80M7, J100M7 >$	$< J80M6, J100M8 >$ $< J80M8, J50M7 >$	$< J40M6, J50M10 >$ $< J40M10, J50M6 >$
Big Difference	$< J20M9, J50M9 >$ $< J40M9, J100M9 >$	$< J20M10, J80M9 >$ $< J100M6, J40M7 >$	$< J20M6, J100M10 >$ $< J30M6, J80M10 >$

TABLE III
MEAN (STANDARD DEVIATION) OF **IGD RESULTS** OF THE 30 RUNS AMONG COMPARED ALGORITHMS

Instances	MOMA	NSGA-II [32]	MOEA/D [24]	LRVMA [30]	MOEA-DCH [51]	MOMFEA [31]	MTEA/D-DN [27]
J20M7	0.10(0.04)	0.16(0.05)(-)	0.22(0.07)(-)	0.13(0.05)(-)	0.12(0.05)(-)	**0.09(0.02)**(≈)	0.12(0.03)(-)
J20M8	0.24(0.10)	0.41(0.15)(-)	0.55(0.12)(-)	0.31(0.14)(≈)	0.30(0.13)(≈)	**0.21(0.08)**(≈)	0.27(0.09)(≈)
J30M7	**0.10(0.04)**	0.22(0.07)(-)	0.27(0.07)(-)	0.15(0.05)(-)	0.13(0.04)(-)	0.11(0.04)(≈)	0.15(0.05)(-)
J30M10	**0.13(0.07)**	0.42(0.10)(-)	0.58(0.12)(-)	0.26(0.09)(-)	0.23(0.12)(-)	0.24(0.06)(-)	0.27(0.11)(-)
J40M8	**0.13(0.06)**	0.46(0.07)(-)	0.47(0.11)(-)	0.24(0.09)(-)	0.19(0.07)(-)	0.22(0.07)(-)	0.24(0.11)(-)
J50M8	**0.13(0.06)**	0.44(0.09)(-)	0.49(0.11)(-)	0.25(0.10)(-)	0.18(0.06)(-)	0.20(0.07)(-)	0.25(0.10)(-)
J80M6	0.20(0.05)	0.24(0.09)(≈)	0.36(0.10)(-)	0.28(0.06)(-)	0.25(0.06)(-)	**0.18(0.04)**(≈)	0.28(0.07)(-)
J100M8	**0.19(0.08)**	0.42(0.11)(-)	0.44(0.11)(-)	0.30(0.10)(-)	0.25(0.07)(-)	0.29(0.06)(-)	0.30(0.07)(-)
J40M10	**0.16(0.10)**	0.62(0.11)(-)	0.71(0.12)(-)	0.30(0.11)(-)	0.26(0.09)(-)	0.41(0.08)(-)	0.31(0.10)(-)
J50M6	0.15(0.05)	**0.15(0.04)**(≈)	0.29(0.07)(-)	0.22(0.05)(-)	0.22(0.06)(-)	0.17(0.05)(≈)	0.22(0.06)(-)
J20M9	0.15(0.07)	0.37(0.08)(-)	0.46(0.12)(-)	0.20(0.07)(-)	0.19(0.06)(-)	**0.15(0.04)**(≈)	0.20(0.09)(-)
J50M9	**0.12(0.06)**	0.47(0.09)(-)	0.44(0.09)(-)	0.17(0.08)(-)	0.16(0.07)(-)	0.25(0.06)(-)	0.22(0.09)(-)
J20M10	**0.26(0.15)**	0.55(0.10)(-)	0.73(0.14)(-)	0.34(0.10)(-)	0.36(0.11)(-)	0.29(0.09)(≈)	0.31(0.12)(≈)
J80M9	**0.14(0.09)**	0.51(0.06)(-)	0.53(0.09)(-)	0.31(0.12)(-)	0.24(0.10)(-)	0.35(0.09)(-)	0.31(0.09)(-)
J30M6	0.18(0.05)	0.16(0.04)(≈)	0.28(0.05)(-)	0.21(0.05)(-)	0.20(0.05)(≈)	**0.18(0.04)**(≈)	0.20(0.05)(≈)
J80M10	**0.15(0.07)**	0.57(0.08)(-)	0.59(0.10)(-)	0.34(0.11)(-)	0.32(0.09)(-)	0.42(0.06)(-)	0.38(0.12)(-)
+/-/=		25/1/4	30/0/0	29/0/1	25/0/5	14/3/13	27/0/3
Average Ranking	**1.4**	5.2333	6.9333	4.3333	3.1	2.5667	4.4333

16 out of 30 test instances are presented. The full results can be found in the supplementary material.

G. Computational Complexity Analysis

Given K tasks, population size N per task, M objectives, and N_L local search strategies, the proposed algorithm comprises several key components with the following time complexities. For a single task, Q-learning-based neighborhood size selection is performed once based on non-dominated solutions, resulting in $\mathcal{O}(MN^2)$. In the evolutionary phase, each individual undergoes crossover and mutation with T internal and external neighbors, contributing $\mathcal{O}(N)$, followed by evaluation using the Tchebycheff scalarization function, which adds $\mathcal{O}(MNT)$. A local search is then applied to all individuals, with a cost of $\mathcal{O}(MNN_L{}^2)$. Summing all components yields a per-generation complexity of $\mathcal{O}(MN^2 + N + MNT + MNN_L{}^2)$, which simplifies to $\mathcal{O}(MN^2)$ under the assumptions that T is constant and $N_L \ll N$. For K tasks, the total cost becomes $\mathcal{O}(K \cdot MN^2)$.

IV. EXPERIMENTAL STUDIES

A. Multi-Task FFJSS Datasets

In this paper, each instance is named by the number of jobs J and the number of machines M. For example, J20M6 means that there are 20 jobs and 6 machines in this instance. The number of operations assigned to each job is uniformly distributed between $[M/2, M]$. We designed three different multi-task scenarios with the same, small differences, and big differences types based on the number of jobs and machines. The details are shown in Table II. For example, $J20M7$ and $J20M8$ have the same job number and small differences of machine number. $J30M6$ and $J80M10$ have big differences in job numbers and different machine numbers. The datasets can be found in [3].

B. Compared Algorithms

This paper aims to validate the efficacy of a proposed transfer strategy for multi-task multi-objective FFJSS problems. Given the scarcity of research on multi-task optimization algorithms for this domain, we investigate advanced transfer strategies within the established LRVMA framework, a single-task multi-objective optimization algorithm based on decomposition [30].

Initially, we integrate the multifactorial transfer strategy of [31] into LRVMA, resulting in the Multi-Objective Multifactorial Evolutionary Algorithm (MOMFEA). The transfer process occurs within the neighborhood of the same subproblem of different tasks. Subsequently, we explore the multi-task evolutionary algorithm based on decomposition with dual neighborhood (MTEA/D-DN) [27]. The transferred solutions are generated in different sub-problems of different tasks. This paper proposes the MOMA, which not only designs the adaptive dual-neighborhood size method to balance the convergence and diversity of the population and explore more promising target task individuals simultaneously but also proposes an improved neighborhood update strategy to improve the the rate of positive transfer. Therefore, two variants are used to verify the effectiveness of the two strategies. (1) Variant1: excluding the improved neighborhood update strategy. (2) Variant2: excluding the adaptive external neighborhood size method. For comparative analysis, the multi-objective algorithms, NSGA-II [32], MOEA/D [24] (The

TABLE IV
MEAN (STANDARD DEVIATION) OF **HV RESULTS** OF THE 30 RUNS AMONG COMPARED ALGORITHMS

Instances	MOMA	NSGA-II [32]	MOEA/D [24]	LRVMA [30]	MOEA-DCH [51]	MOMFEA [31]	MTEA/D-DN [27]
J20M7	**0.82(0.05)**	0.69(0.05)(-)	0.64(0.09)(-)	0.77(0.06)(-)	0.81(0.04)(≈)	0.77(0.08)(-)	0.79(0.05)(-)
J20M8	0.68(0.11)	0.47(0.12)(-)	0.34(0.09)(-)	0.60(0.14)(≈)	**0.70(0.10)**(≈)	0.59(0.14)(-)	0.64(0.09)(≈)
J30M7	**0.85(0.06)**	0.66(0.08)(-)	0.60(0.08)(-)	0.79(0.06)(-)	0.83(0.07)(≈)	0.80(0.07)(-)	0.78(0.07)(-)
J30M10	**0.78(0.10)**	0.46(0.09)(-)	0.31(0.09)(-)	0.62(0.10)(-)	0.64(0.08)(-)	0.65(0.12)(-)	0.62(0.11)(-)
J40M8	**0.74(0.08)**	0.36(0.06)(-)	0.35(0.09)(-)	0.59(0.10)(-)	0.60(0.08)(-)	0.61(0.08)(-)	0.59(0.11)(-)
J50M8	**0.79(0.08)**	0.43(0.09)(-)	0.37(0.10)(-)	0.63(0.09)(-)	0.68(0.07)(-)	0.68(0.08)(-)	0.64(0.10)(-)
J80M6	**0.82(0.08)**	0.62(0.11)(-)	0.51(0.11)(-)	0.69(0.08)(-)	0.76(0.06)(-)	0.72(0.07)(-)	0.70(0.11)(-)
J100M8	**0.74(0.10)**	0.49(0.09)(-)	0.44(0.10)(-)	0.60(0.11)(-)	0.60(0.06)(-)	0.63(0.08)(-)	0.59(0.07)(-)
J40M10	**0.76(0.11)**	0.29(0.09)(-)	0.21(0.07)(-)	0.58(0.11)(-)	0.47(0.08)(-)	0.61(0.10)(-)	0.58(0.11)(-)
J50M6	**0.83(0.06)**	0.71(0.06)(-)	0.59(0.08)(-)	0.73(0.07)(-)	0.77(0.06)(-)	0.73(0.07)(-)	0.74(0.08)(-)
J20M9	**0.71(0.10)**	0.42(0.07)(-)	0.35(0.09)(-)	0.66(0.10)(≈)	0.69(0.06)(≈)	0.63(0.07)(-)	0.67(0.10)(≈)
J50M9	**0.81(0.09)**	0.48(0.07)(-)	0.46(0.08)(-)	0.73(0.08)(-)	0.65(0.07)(-)	0.75(0.09)(-)	0.69(0.11)(-)
J20M10	**0.67(0.16)**	0.37(0.09)(-)	0.23(0.10)(-)	0.59(0.10)(-)	0.63(0.10)(≈)	0.55(0.11)(-)	0.61(0.13)(≈)
J80M9	**0.76(0.10)**	0.35(0.07)(-)	0.33(0.08)(-)	0.57(0.13)(-)	0.49(0.09)(-)	0.61(0.10)(-)	0.55(0.09)(-)
J30M6	**0.84(0.05)**	0.74(0.06)(-)	0.64(0.07)(-)	0.77(0.06)(-)	0.83(0.05)(≈)	0.79(0.07)(-)	0.78(0.07)(-)
J80M10	**0.78(0.09)**	0.36(0.07)(-)	0.33(0.08)(-)	0.55(0.11)(-)	0.48(0.06)(-)	0.56(0.09)(-)	0.52(0.11)(-)
+/-/=		30/0/0	30/0/0	28/0/2	18/0/12	28/0/2	27/0/3
Average Ranking	**1.0667**	5.9667	6.9667	3.9667	3.1	2.9	4.0333

16 out of 30 test instances are presented. The full results can be found in the supplementary material.

initialization strategy, crossover strategy and mutation strategy are similar with LRVMA), and MOEA-DCH [51], are included.

C. Parameter Settings and Performance Metrics

Building upon the methodology outlined in [30], this work employs the same parameter configuration except the knowledge transfer rate. The algorithm utilizes a population size of 100 individuals per task, totaling 200 for the two tasks examined. Experiments were conducted across 30 independent runs, with each run limited to a maximum of 200 generations. A mutation probability of 0.8 was maintained throughout. Q-learning parameters were set with a discount factor of 0.8, a learning rate of 0.1, and a greedy factor of 0.95. Meanwhile, the knowledge transfer rate is 0.1. For MOEA/D, the neighborhood size is 10 [27]. For dual-neighborhoods, the maximum neighborhood size is 20.

In this paper, the Inverted Generational Distance (IGD) [59] and Hypervolume (HV) [60] metrics are used to measure the algorithm's performance. IGD evaluates the algorithm by calculating the average distance between the obtained Pareto front and the true Pareto front. The experimental results of different algorithms running 30 times on the same dataset are pooled and subjected to non-dominated sorting to obtain the approximated Pareto front. The smaller the value of IGD, the better the performance of the algorithm. HV measures the volume of the objective space dominated by the obtained Pareto front, bounded by a reference point [1,1]. A larger HV value signifies a Pareto front that is closer to the approximated Pareto front and has a good distribution of solutions.

D. Results and Discussions

To ensure comprehensive evaluation, we select one representative instance pair from each of the eight multi-task FFJSS scenarios for experimental analysis. The full results can be found in Section S-2 of the supplementary material. We also performed statistical significance analyses, including the Wilcoxon rank-sum test and the Friedman test. The Wilcoxon rank-sum test (at 5% significance level) was used to assess pairwise performance differences between MOMA and other algorithms on individual instances, with results presented in Tables III – VI. The '+/-/=' symbols indicate the number of instances where other algorithms are significantly superior, inferior, or similar to MOMA. The Friedman test was employed to compare and rank all algorithms across multiple instances.

1) Mean and Standard Deviation Analysis With Other Algorithms: To evaluate the performance of the proposed MOMA, six different algorithms are compared. Tables III and IV present the mean (standard deviation) of IGD and HV metrics, respectively. Based on the results, the algorithms can be ranked as follows: MOMA > MOMFEA > MOEA-DCH > LRVMA > MTEA/D-DN > NSGA-II > MOEA/D. The proposed algorithm is significantly superior to all other algorithms.

2) Comparison of Approximated Pareto Fronts With Other Algorithms: Fig. 7 shows the approximated Pareto front obtained by different algorithms on partial instances. The abscissa represents Makespan, while the ordinate represents Energy Consumption (EC). To ensure a fair comparison, the approximated Pareto fronts displayed are from the median run (based on HV value) out of 30 runs for each algorithm. We selected eight groups of multi-task instances from different scenarios. Analyzing the convergence and diversity of the approximate Pareto front, our proposed algorithm MOMA exhibits superior performance in most instances, with exceptions in instances J20M7, J20M8, J30M6, and J30M7. From the perspective of multi-task, instances with more machines but an equal number of jobs generally demonstrated better performance, as exemplified by J20M8 and J30M10. Instances with more jobs and the same number of machines show even greater performance improvements, such as J50M8 and J50M9. When the number of machines and jobs varies, more complex instances, particularly those involving a combination of simple and complex tasks (e.g., J80M9, J80M10), benefit significantly from our algorithm.

3) Mean and Standard Deviation Analysis With Variants: To further verify the effectiveness of the proposed strategies, we compared the proposed algorithm with its variants Variant1, Variant2, and MTEA/D-DN. Variant 1 and Variant 2 are improved based on the MTEA/D-DN algorithm. Variant 1

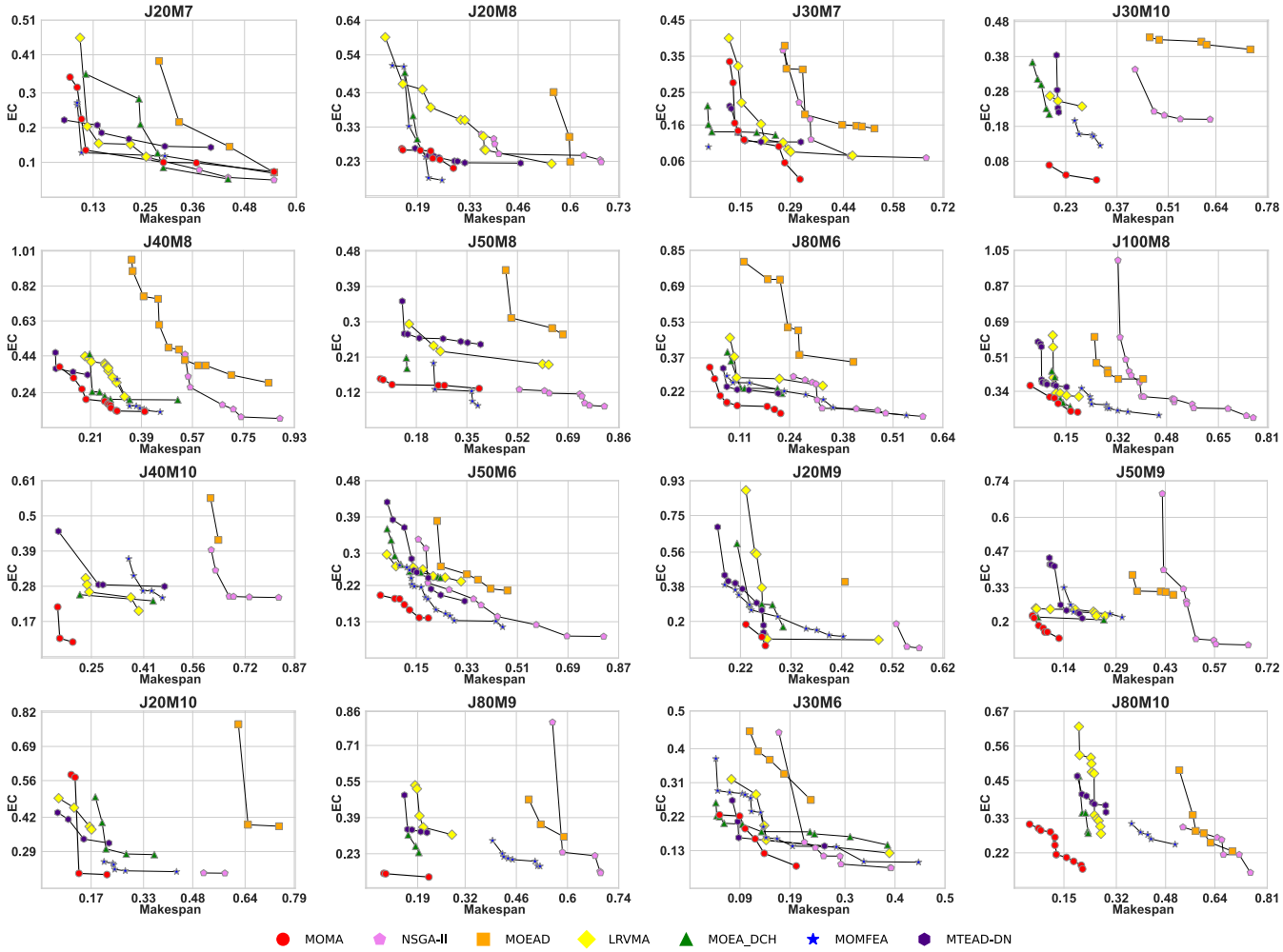


Fig. 7. Approximated Pareto front obtained by different algorithms on partial instances.

TABLE V
MEAN (STANDARD DEVIATION)OF **IGD RESULTS** OF THE 30 RUNS AMONG ALL MOMA VARIANTS

Instances	MOMA	MTEA/D-DN [27]	Variant1	Variant2
J20M7	**0.13(0.05)**	0.16(0.04)(–)	0.15(0.06)(≈)	0.14(0.05)(≈)
J20M8	0.33(0.13)	0.37(0.12)(≈)	**0.32(0.13)**(≈)	0.34(0.13)(≈)
J30M7	**0.15(0.06)**	0.23(0.07)(–)	0.18(0.06)(≈)	0.17(0.07)(≈)
J30M10	0.20(0.08)	0.32(0.12)(–)	0.25(0.09)(–)	**0.20(0.07)**(≈)
J40M8	**0.18(0.07)**	0.32(0.12)(–)	0.27(0.09)(–)	0.22(0.10)(≈)
J50M8	**0.15(0.06)**	0.30(0.12)(–)	0.19(0.08)(–)	0.17(0.10)(≈)
J80M6	**0.13(0.09)**	0.24(0.12)(–)	0.20(0.11)(–)	0.19(0.08)(–)
J100M8	**0.29(0.12)**	0.47(0.10)(–)	0.43(0.14)(–)	0.36(0.11)(–)
J40M10	**0.20(0.10)**	0.39(0.13)(–)	0.33(0.13)(–)	0.29(0.11)(–)
J50M6	**0.12(0.06)**	0.22(0.09)(–)	0.17(0.05)(–)	0.19(0.07)(–)
J20M9	0.20(0.08)	0.25(0.09)(–)	0.24(0.09)(≈)	**0.19(0.07)**(≈)
J50M9	**0.25(0.12)**	0.42(0.17)(–)	0.36(0.13)(–)	0.26(0.11)(≈)
J20M10	0.33(0.14)	0.37(0.13)(≈)	0.31(0.12)(≈)	**0.30(0.12)**(≈)
J80M9	**0.19(0.10)**	0.38(0.10)(–)	0.33(0.15)(–)	0.24(0.09)(≈)
J30M6	**0.14(0.06)**	0.20(0.08)(–)	0.17(0.06)(–)	0.15(0.05)(≈)
J80M10	**0.21(0.09)**	0.49(0.14)(–)	0.38(0.12)(–)	0.30(0.08)(–)
+/-/=		28/0/2	20/0/10	13/0/17
Average ranking	**1.2**	4	2.8	2

16 out of 30 test instances are presented. The full results can be found in the supplementary material.

TABLE VI
MEAN (STANDARD DEVIATION) OF **HV RESULTS** OF THE 30 RUNS AMONG ALL MOMA VARIANTS

Instances	MOMA	MTEA/D-DN [27]	Variant1	Variant2
J20M7	**0.76(0.07)**	0.72(0.06)(–)	0.74(0.08)(≈)	0.75(0.06)(≈)
J20M8	0.56(0.15)	0.51(0.12)(≈)	**0.56(0.14)**(≈)	0.55(0.12)(≈)
J30M7	**0.75(0.09)**	0.64(0.10)(–)	0.72(0.09)(≈)	0.74(0.10)(≈)
J30M10	0.69(0.12)	0.53(0.12)(–)	0.61(0.13)(–)	**0.69(0.11)**(≈)
J40M8	**0.67(0.10)**	0.50(0.12)(–)	0.57(0.12)(–)	0.62(0.12)(≈)
J50M8	**0.75(0.10)**	0.57(0.12)(–)	0.69(0.09)(–)	0.73(0.12)(≈)
J80M6	**0.84(0.10)**	0.69(0.13)(–)	0.75(0.13)(–)	0.76(0.09)(–)
J100M8	**0.59(0.14)**	0.37(0.09)(–)	0.43(0.14)(–)	0.50(0.11)(–)
J40M10	**0.70(0.13)**	0.47(0.13)(–)	0.53(0.13)(–)	0.58(0.13)(–)
J50M6	**0.80(0.08)**	0.68(0.11)(–)	0.74(0.07)(–)	0.72(0.09)(–)
J20M9	0.65(0.11)	0.61(0.11)(≈)	0.62(0.11)(≈)	**0.69(0.10)**(≈)
J50M9	**0.66(0.14)**	0.46(0.16)(–)	0.52(0.13)(–)	0.63(0.13)(≈)
J20M10	**0.57(0.18)**	0.49(0.14)(≈)	0.54(0.14)(≈)	0.56(0.14)(≈)
J80M9	**0.67(0.13)**	0.43(0.09)(–)	0.52(0.15)(–)	0.62(0.11)(≈)
J30M6	**0.79(0.07)**	0.71(0.10)(–)	0.75(0.07)(–)	0.78(0.07)(≈)
J80M10	**0.70(0.12)**	0.39(0.12)(–)	0.51(0.12)(–)	0.59(0.09)(–)
+/-/=		27/0/3	20/0/10	13/0/17
Average ranking	**1.1**	4	2.8667	2.0333

16 out of 30 test instances are presented. The full results can be found in the supplementary material.

further adaptively learns the size of the dual-neighborhood, and Variant 2 improves the neighborhood update strategy. The comparison results are presented in Table V and Table VI. According to the average ranking from the Friedman test, the ranking results are as follows: MOMA > Variant2 > Variant1 > MTEA/D-DN. The full ablation results can be found in Table S3 and Table S4 of the supplementary material.

In summary, MOMA effectively explores knowledge transfer among different tasks through the proposed knowledge transfer strategy, thereby achieving effective knowledge transfer.

TABLE VII
DETAILS OF TRANSFER INDIVIDUALS FOR DIFFERENT MULTI-TASK OPTIMIZATION ALGORITHMS

Scenarios	MOMA				MOMFEA [31]				MTEA/D-DN [27]				Variant1				Variant2			
	N_T	N_W	N_P	R	N_T	N_W	N_P	R	N_T	N_W	N_P	R	N_T	N_W	N_P	R	N_T	N_W	N_P	R
$J20M7\&J20M8$	8028	1798	15995	199.24%	8148	23	299	3.67%	7824	129	457	5.84%	8124	321	4533	55.80%	8092	1045	8009	98.97%
$J30M7\&J30M10$	7696	1629	15616	202.91%	7990	42	636	7.96%	8168	103	420	5.14%	7986	278	3527	44.16%	7850	1249	9974	127.06%
$J40M8\&J50M8$	7882	2431	19687	249.77%	8204	57	545	6.64%	7768	106	371	4.78%	8126	295	3547	43.65%	8006	1265	11768	146.99%
$J80M6\&J100M8$	7990	2780	16956	212.22%	8118	56	693	8.54%	7982	92	437	5.47%	8062	262	3205	39.75%	7812	1116	10924	139.84%
$J40M10\&J50M6$	8158	1728	17268	211.67%	8124	45	417	5.13%	7870	79	340	4.32%	7896	306	4005	50.72%	7950	1175	8438	106.14%
$J20M9\&J50M9$	8012	1585	14811	184.86%	8142	40	419	5.15%	7848	107	424	5.40%	8164	259	3105	38.03%	8246	1411	11477	139.18%
$J20M10\&J80M9$	8162	2045	17671	216.50%	7986	61	655	8.20%	8082	98	452	5.59%	8052	349	4286	52.23%	8054	1132	12684	157.49%
$J30M6\&J80M10$	7988	1787	12608	157.84%	8040	52	543	6.75%	8190	68	279	3.41%	7844	299	3604	45.95%	8140	1316	11730	144.10%

16 out of 30 test instances are presented. The full results can be found in the supplementary material.

Although implicit transfer cannot be directly reflected in the actual representation of shared knowledge, the generation of high-quality offspring within the unified search space further enhances the performance of MOMA.

V. FURTHER ANALYSIS

This section further analyzes the proposed algorithm in terms of transfer effectiveness and sensitivity to transfer probability. Full results are available in Section S-II of the supplementary material.

A. Transfer Effectiveness Analysis

This section delves into the details of positive knowledge transfer in various multi-task instances. We define the number of transferred individuals, N_T, as an offspring generated when the transfer probability is met. The number of positive transferred individuals, N_P, represents the number of neighboring individuals updated by an offspring. The ratio $R = N_P/N_T * 100\%$ indicates the average number of neighboring individuals updated by each transferred individual, with a higher R value signifying greater positive transfer efficiency. The number of sub-problems successfully updated by a transferred individual is denoted as N_W. Table VII presents detailed numerical results of our proposed algorithm MOMA compared with two multi-task optimization algorithms, MOMFEA, and MTEA/D-DN, and two variants algorithms on different multi-task FFJSS instances.

In terms of the number of transfer individuals generated, five algorithms exhibit similar N_T values across all scenarios. Based on the N_W values, the ranking of the number of sub-problems (N_W) updated by different algorithms is MOMFEA $<$ MTEA/D-DN $<$ Variant1 $<$ Variant2 $<$ MOMA. Meanwhile, to more fairly compare the positive transfer performance of different algorithms, the R value is used for comparison. Overall, the ranking is MOMFEA $<$ MTEA/D-DN $<$ Variant1 $<$ Variant2 $<$ MOMA. More specifically, although MOMFEA updates fewer weight vectors than MTEA/D-DN, it updates more individuals, leading to better performance of MOMFEA. In other words, by randomly selecting individuals from the target task to form the external neighborhood, diversity is effectively improved during the transfer process, but convergence is affected. Furthermore, compared to MTEA/D-DN, Variant1 updates more sub-problems by dynamically adjusting the size of the dual-neighborhood, and the positive transfer efficiency is significantly improved. This further confirms the two-fold impact of neighborhood size: the imbalance between population convergence and diversity, as well as the negative transfer problem. Compared with MTEA/D-DN, Variant2 further exploits the potential of transferred individuals, who can not update in current subproblem, to improve the rate of positive transfer, updates more sub-problems, and achieves a higher positive transfer rate. Intuitively, crossover and mutation between parents of different subproblems from different tasks facilitate more knowledge exchange. Finally, combining the advantages of Variant1 and Variant2, the proposed algorithm MOMA not only updates more sub-problems but also has a higher positive transfer rate, which is the direct reason for the optimal performance.

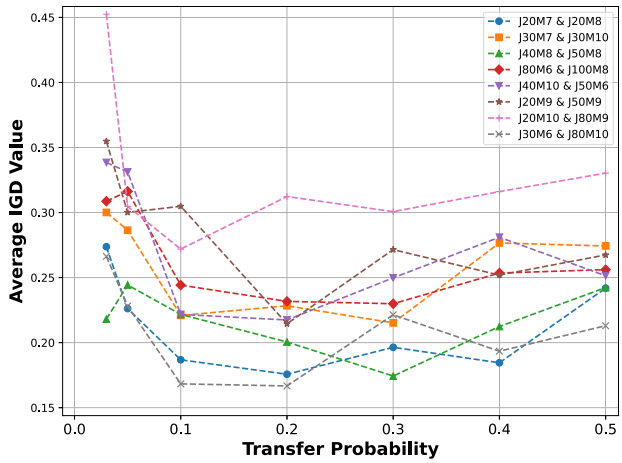


Fig. 8. Average IGD values of different multi-task FFJSS scenarios with different transfer probability obtained by MOMA. 16 out of 30 test instances are presented. The full results can be found in the supplementary material.

B. Transfer Probability Sensitivity Analysis

Transfer probability plays a critical role in the algorithm's performance. We investigate its sensitivity by setting the parameter to $\{0.03, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$ while maintaining the proposed algorithm MOMA's parameters consistent with the above settings. Fig. 8 illustrates the performance curves of the MOMA across various multi-task FFJSS scenarios. The x-axis

represents the transfer probability, while the y-axis indicates the average IGD values. Overall, the algorithm's performance degrades at higher transfer probabilities.

It can be seen from the figure that the optimal transfer probability is generally distributed between 0.1 and 0.3. Specifically, for Scenario 1 ($J20M7\&J20M8$), the optimal probability is 0.2; for Scenario 2 ($J30M7\&J30M10$), it is 0.3; for Scenario 3 ($J40M8\&J50M8$), it is also 0.3. The optimal probabilities for the two task pairs in Scenario 4 are 0.3 ($J80M6\&J100M8$). In Scenario 5, the optimal probabilities are 0.2 ($J40M10\&J50M6$). In Scenario 6, the optimal probabilities are 0.2 ($J20M9\&J50M9$). For Scenarios 7 and 8, the optimal probability is 0.1.

In summary, determining the optimal transfer probability is challenging due to the difficulty in quantifying task similarity across different scenarios. This implies that the optimal value can vary significantly depending on the specific dataset or problem type. A notable finding is that a transfer probability of the interval $[0.1, 0.3]$ is recommended for multi-task scenarios involving big differences between machines and jobs, as evident in different scenarios.

VI. Conclusions

The main goal of this article is to design an effective multi-task optimization algorithm for solving multi-task FFJSS problems through efficient knowledge transfer. This goal has been achieved by the proposed MOMA. Our contributions are threefold. First, we design a unified representation for FFJSS in a unified search space, which is important for knowledge transfer. Second, we improve upon multifactorial algorithms by designing an enhanced dual-neighborhood transfer strategy and an improved neighborhood update strategy to improve the rate of positive transfer. Finally, we verified the effectiveness of the proposed MOMA on eight diverse multi-task FFJSS scenarios.

Experimental results demonstrate that our algorithm exhibits strong performance across various scenarios. Further analysis confirms that knowledge transfer significantly benefits more complex tasks, especially when optimizing both simple and complex tasks simultaneously. Additionally, by tracking transferred individuals during the evolutionary process, we gain deeper insights into the mechanisms of effective transfer and the role of transfer probability in the multi-task FFJSS problem.

Considering the superior performance of the MOMA, it is worth exploring its application in other combinatorial optimization problems. Additionally, the proposed algorithm is limited by the use of a fixed transfer probability and the lack of interpretability in the transferred knowledge. In the future, we would like to address these limitations by developing adaptive transfer probability mechanisms and exploring the interpretability of the transferred knowledge.

References

[1] M. Xu, Y. Mei, F. Zhang, and M. Zhang, "Genetic programming and reinforcement learning on learning heuristics for dynamic scheduling: A preliminary comparison," *IEEE Comput. Intell. Mag.*, vol. 19, no. 2, pp. 18–33, May 2024.

[2] J. Luo et al., "A new multitask joint learning framework for expensive multi-objective optimization problems," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 8, no. 2, pp. 1894–1909, Apr. 2024.

[3] J. Li, Y. Han, K. Gao, X. Xiao, and P. Duan, "Bi-population balancing multi-objective algorithm for fuzzy flexible job shop with energy and transportation," *IEEE Trans. Automat. Sci. Eng.*, vol. 21, no. 3, pp. 4686–4702, Jul. 2024.

[4] P. Melin and O. Castillo, "A review on type-2 fuzzy logic applications in clustering, classification and pattern recognition," *Appl. soft Comput.*, vol. 21, pp. 568–577, Aug. 2014.

[5] J. Li, J. Li, L. Zhang, H. Sang, Y. Han, and Q. Chen, "Solving type-2 fuzzy distributed hybrid flowshop scheduling using an improved brain storm optimization algorithm," *Int. J. Fuzzy Syst.*, vol. 23, pp. 1194–1212, Mar. 2021.

[6] J.-q. Li, Z.-m. Liu, C. Li, and Z.-x. 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.

[7] Z. Pan, D. Lei, and L. Wang, "A bi-population evolutionary algorithm with feedback for energy-efficient fuzzy flexible job shop scheduling," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 8, pp. 5295–5307, Aug. 2022.

[8] T. Wei, S. Wang, J. Zhong, D. Liu, and J. Zhang, "A review on evolutionary multitask optimization: Trends and challenges," *IEEE Trans. Evol. Comput.*, vol. 26, no. 5, pp. 941–960, Oct. 2022.

[9] A. Gupta, L. Zhou, Y.-S. Ong, Z. Chen, and Y. Hou, "Half a dozen real-world applications of evolutionary multitasking, and more," *IEEE Comput. Intell. Mag.*, vol. 17, no. 2, pp. 49–66, May 2022.

[10] G. Chen, Y. Guo, M. Jiang, S. Yang, X. Zhao, and D. Gong, "A subspace-knowledge transfer based dynamic constrained multiobjective evolutionary algorithm," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 8, no. 2, pp. 1500–1512, Apr. 2024.

[11] T. Guo, Y. Mei, K. Tang, and W. Du, "Cooperative co-evolution for large-scale multiobjective air traffic flow management," *IEEE Trans. Evol. Comput.*, vol. 28, no. 6, pp. 1644–1658, Dec. 2024.

[12] C. Wang, H. Ma, G. Chen, and S. Hartmann, "Using an estimation of distribution algorithm to achieve multitasking semantic web service composition," *IEEE Trans. Evol. Comput.*, vol. 27, no. 3, pp. 490–504, Jun. 2023.

[13] Y. Mei, S. Nguyen, B. Xue, and M. Zhang, "An efficient feature selection algorithm for evolving job shop scheduling rules with genetic programming," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 1, no. 5, pp. 339–353, Oct. 2017.

[14] A. Gupta, Y. S. Ong, and L. Feng, "Multifactorial evolution: Toward evolutionary multitasking," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 343–357, Jun. 2016.

[15] Y. Huang, L. Feng, M. Li, Y. Wang, Z. Zhu, and K. C. Tan, "Fast vehicle routing via knowledge transfer in a reproducing kernel hilbert space," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 53, no. 9, pp. 5404–5416, Sep. 2023.

[16] L. Feng, A. Gupta, K. Tan, and Y. Ong, "Evolutionary multi-task optimization for generalized vehicle routing problem with occasional drivers," in *Evolutionary Multi-Task Optimization: Foundations and Methodologies*. Berlin, Germany: Springer, 2022, pp. 97–122.

[17] L. Zhou, L. Feng, J. Zhong, Y.-S. Ong, Z. Zhu, and E. Sha, "Evolutionary multitasking in combinatorial search spaces: A case study in capacitated vehicle routing problem," in *Proc. IEEE Symp. Ser. Comput. Intell.*, 2016, pp. 1–8.

[18] M. Xu, Y. Mei, F. Zhang, and M. Zhang, "Niching genetic programming to learn actions for deep reinforcement learning in dynamic flexible scheduling," *IEEE Trans. Evol. Comput.*, early access, May 1, 2024, doi: 10.1109/TEVC.2024.3395699.

[19] F. Zhang, Y. Mei, S. Nguyen, K. C. Tan, and M. Zhang, "Task relatedness-based multitask genetic programming for dynamic flexible job shop scheduling," *IEEE Trans. Evol. Comput.*, vol. 27, no. 6, pp. 1705–1719, Dec. 2023.

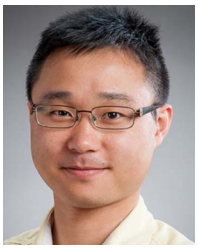
[20] F. Zhang, Y. Mei, S. Nguyen, and M. Zhang, "Multitask multiobjective genetic programming for automated scheduling heuristic learning in dynamic flexible job-shop scheduling," *IEEE Trans. Cybern.*, vol. 53, no. 7, pp. 4473–4486, Jul. 2023.

[21] J. Lin, Q. Chen, B. Xue, and M. Zhang, "Evolutionary multitasking for multiobjective feature selection in classification," *IEEE Trans. Evol. Comput.*, vol. 28, no. 6, pp. 1852–1866, Dec. 2024.

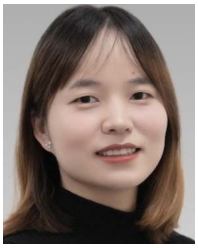
[22] Y. Feng, L. Feng, S. Liu, S. Kwong, and K. C. Tan, "Towards multi-objective high-dimensional feature selection via evolutionary multitasking," *Swarm Evol. Comput.*, vol. 89, 2024, Art. no. 101618.

[23] Y. Bi, B. Xue, and M. Zhang, "Multitask feature learning as multiobjective optimization: A new genetic programming approach to image classification," *IEEE Trans. Cybern.*, vol. 53, no. 5, pp. 3007–3020, May 2023.
[24] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
[25] S. Yao, Z. Dong, X. Wang, and L. Ren, "A multiobjective multifactorial optimization algorithm based on decomposition and dynamic resource allocation strategy," *Inf. Sci.*, vol. 511, pp. 18–35, Feb. 2020.
[26] Q. Lin, Z. Wu, L. Ma, M. Gong, J. Li, and C. A. C. Coello, "Multiobjective multitasking optimization with decomposition-based transfer selection," *IEEE Trans. Cybern.*, vol. 54, no. 5, pp. 3146–3159, May 2024.
[27] X. Wang, Z. Dong, L. Tang, and Q. Zhang, "Multiobjective multitask optimization-neighborhood as a bridge for knowledge transfer," *IEEE Trans. Evol. Comput.*, vol. 27, no. 1, pp. 155–169, Feb. 2023.
[28] K. Li, "A survey of multi-objective evolutionary algorithm based on decomposition: Past and future," *IEEE Trans. Evol. Comput.*, early access, Nov. 12, 2024, doi: 10.1109/TEVC.2024.3496507.
[29] J. M. Mendel and X. Liu, "Simplified interval type-2 fuzzy logic systems," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 6, pp. 1056–1069, Dec. 2013.
[30] R. Li, W. Gong, C. Lu, and L. Wang, "A learning-based memetic algorithm for energy-efficient flexible job-shop scheduling with type-2 fuzzy processing time," *IEEE Trans. Evol. Comput.*, vol. 27, no. 3, pp. 610–620, Jun. 2023.
[31] A. Gupta, Y. S. Ong, L. Feng, and K. C. Tan, "Multiobjective multifactorial optimization in evolutionary multitasking," *IEEE Trans. Cybern.*, vol. 47, no. 7, pp. 1652–1665, Jul. 2017.
[32] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[33] J. Liu, A. Gupta, C. Ooi, and Y.-S. Ong, "ExTrEMO: Transfer evolutionary multiobjective optimization with proof of faster convergence," *IEEE Trans. Evol. Comput.*, vol. 29, no. 1, pp. 102–116, Feb. 2025.
[34] X. Wang and Y. Jin, "Distilling ensemble surrogates for federated data-driven many-task optimization," *IEEE Trans. Evol. Comput.*, early access, Jul. 15, 2024, doi: 10.1109/TEVC.2024.3428701.
[35] J. Huang, C. Chen, C.-M. Vong, and Y.-M. Cheung, "Broad multitask learning system with group sparse regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 5, pp. 8265–8278, May 2025.
[36] H. Zhao, X.-H. Ning, J.-Y. Li, and J. Liu, "Evolutionary multi-task framework with bi-knowledge transfer for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, early access, Jan. 1, 2025, doi: 10.1109/TEVC.2025.3551728.
[37] X. Ning, H. Zhao, X. Liu, and J. Liu, "An evolutionary multi-task genetic algorithm with assisted-task for flexible job shop scheduling," in *Proc. CCF Conf. Comput. Supported Cooperative Work Social Comput.*, 2022, pp. 367–378.
[38] J. Li, J. Cai, T. Sun, Q. Zhu, and Q. Lin, "Multitask-based evolutionary optimization for vehicle routing problems in autonomous transportation," *IEEE Trans. Automat. Sci. Eng.*, vol. 21, no. 3, pp. 2400–2411, Jul. 2024.
[39] J. Ma, Y. Zhang, Y. Wang, D. Gong, X. Sun, and B. Zeng, "A multitask multiobjective operation optimization method for coal mine integrated energy system," *IEEE Trans. Ind. Informat.*, vol. 20, no. 9, pp. 11149–11160, Sep. 2024.
[40] C. Yang, Q. Chen, Z. Zhu, Z.-A. Huang, S. Lan, and L. Zhu, "Evolutionary multitasking for costly task offloading in mobile-edge computing networks," *IEEE Trans. Evol. Comput.*, vol. 28, no. 2, pp. 338–352, Apr. 2024.
[41] H. Ding et al., "Point cloud registration via sampling-based evolutionary multitasking," *Swarm Evol. Comput.*, vol. 89, 2024, Art. no. 101535.
[42] L. Li and Z. Chai, "Energy-saving distributed flexible job-shop scheduling with fuzzy processing time in IIoT: A novel evolutionary multitasking algorithm," *J. Ind. Inf. Integration*, vol. 45, 2025, Art. no. 100829.
[43] L. Feng et al., "Solving generalized vehicle routing problem with occasional drivers via evolutionary multitasking," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3171–3184, Jun. 2021.
[44] L. Feng et al., "Explicit evolutionary multitasking for combinatorial optimization: A case study on capacitated vehicle routing problem," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3143–3156, Jun. 2021.
[45] H. Liu, F. Zhao, L. Wang, T. Xu, and C. Dong, "Evolutionary multitasking memetic algorithm for distributed hybrid flow-shop scheduling problem with deterioration effect," *IEEE Trans. Automat. Sci. Eng.*, vol. 22, pp. 1390–1404, 2025.
[46] F. Zhang, Y. Mei, S. Nguyen, M. Zhang, and K. C. Tan, "Surrogate-assisted evolutionary multitask genetic programming for dynamic flexible job shop scheduling," *IEEE Trans. Evol. Comput.*, vol. 25, no. 4, pp. 651–665, Aug. 2021.
[47] Z. Huang, F. Zhang, Y. Mei, and M. Zhang, "An investigation of multitask linear genetic programming for dynamic job shop scheduling," in *Proc. Eur. Conf. Genet. Program. (Part EvoStar)*, 2022, pp. 162–178.
[48] Z. Huang, Y. Mei, F. Zhang, and M. Zhang, "Multitask linear genetic programming with shared individuals and its application to dynamic job shop scheduling," *IEEE Trans. Evol. Comput.*, vol. 28, no. 6, pp. 1546–1560, Dec. 2024.
[49] X. Chen, J. Li, Z. Wang, Q. Chen, K. Gao, and Q. Pan, "Optimizing dynamic flexible job shop scheduling using an evolutionary multi-task optimization framework and genetic programming," *IEEE Trans. Evol. Comput.*, early access, Feb. 20, 2025, doi: 10.1109/TEVC.2025.3543770.
[50] F. Zhang, G. Shi, Y. Mei, and M. Zhang, "Multiobjective dynamic flexible job shop scheduling with biased objectives via multitask genetic programming," *IEEE Trans. Artif. Intell.*, vol. 6, no. 1, pp. 169–183, Jan. 2025.
[51] X. Zhang, S. Liu, Z. Zhao, and S. Yang, "A decomposition-based evolutionary algorithm with clustering and hierarchical estimation for multiobjective fuzzy flexible jobshop scheduling," *IEEE Trans. Evol. Comput.*, early access, Jan. 26, 2024, doi: 10.1109/TEVC.2024.3359120.
[52] Y. Du, J. Li, X. Chen, P. Duan, and Q. Pan, "Knowledge-based reinforcement learning and estimation of distribution algorithm for flexible job shop scheduling problem," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 4, pp. 1036–1050, Aug. 2023.
[53] Y. Yuan, Y. S. Ong, A. Gupta, P. S. Tan, and H. Xu, "Evolutionary multitasking in permutation-based combinatorial optimization problems: Realization with TSP, QAP, LOP, and JSP," in *Proc. IEEE Region 10 Conf.*, 2016, pp. 3157–3164.
[54] F. Zhang, Y. Mei, S. Nguyen, K. C. Tan, and M. Zhang, "Multitask genetic programming-based generative hyperheuristics: A case study in dynamic scheduling," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10515–10528, Oct. 2022.
[55] Z. Tan, L. Luo, and J. Zhong, "Knowledge transfer in evolutionary multi-task optimization: A survey," *Appl. Soft Comput.*, vol. 138, 2023, Art. no. 110182.
[56] H. Li et al., "Sparse hyperspectral unmixing with preference-based evolutionary multiobjective multitasking optimization," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 8, no. 2, pp. 1922–1937, Apr. 2024.
[57] M. A. Ardeh, Y. Mei, M. Zhang, and X. Yao, "Knowledge transfer genetic programming with auxiliary population for solving uncertain capacitated arc routing problem," *IEEE Trans. Evol. Comput.*, vol. 27, no. 2, pp. 311–325, Apr. 2023.
[58] Y. Fu, Z. Zhang, M. Huang, X. Guo, and L. Qi, "Multi-objective integrated energy-efficient scheduling of distributed flexible job shop and vehicle routing by knowledge-and-learning-based hyper-heuristics," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 9, no. 3, pp. 2137–2150, Jun. 2025, doi: 10.1109/TETCI.2025.3540422.
[59] C. A. C. Coello and N. C. Cortés, "Solving multiobjective optimization problems using an artificial immune system," *Genet. Program. Evolvable Mach.*, vol. 6, pp. 163–190, 2005.
[60] E. Zitzler and L. Thiele, "An evolutionary algorithm for multiobjective optimization: The strength Pareto approach," Comput. Eng. Commun. Netw. Lab (TIK), Swiss Federal Inst Technol. (ETH), Zurich, Switzerland, Tech. Rep. 43, 1998.

Shaojin Geng received the B.E. degree in software engineering from the Taiyuan University of Technology, Taiyuan, China, in 2018, and the M.E. degree in computer technology from Taiyuan University of Science and Technology in 2021. He is currently working toward the Ph.D. degree in control science and engineering with the College of Electronic and Information Engineering, Tongji University, Shanghai, China. From 2023 to 2025, he was sponsored by China Scholarship Council to carry on his research with the School of Engineering and Computer Science, Victoria University of Wellington, Wellington, New Zealand. His research interests include computational intelligence and combinatorial optimization.



Yi Mei (Senior Member, IEEE) received the B.Sc. degree in mathematics and the Ph.D. degree in computer science from the University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively. He is currently an Associate Professor with the School of Engineering and Computer Science, Victoria University of Wellington, Wellington, New Zealand. His research interests include evolutionary computation and machine learning for combinatorial optimisation, genetic programming, hyper-heuristics, and explainable AI. He is an Associate Editor of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, and *Journal of Scheduling*. He is also the Chair of IEEE Taskforce on Evolutionary Scheduling and Combinatorial Optimisation. He is a Fellow of Engineering New Zealand.



Fangfang Zhang (Member, IEEE) received the B.Sc. and M.Sc. degrees from Shenzhen University, Shenzhen, China, and the Ph.D. degree in computer science from Victoria University of Wellington, New Zealand, in 2014, 2017, and 2021, respectively. She was the recipient of ACM SIGEVO Dissertation Award, Honorable Mention, and IEEE CIS Outstanding PhD Dissertation Award, for her Ph.D. thesis. She is currently a Lecturer with the Centre for Data Science and Artifical Intelligence & School of Engineering and Computer Science, Victoria University of Wellington, New Zealand. Her research interests include evolutionary computation, hyper-heuristic learning/optimisation, job shop scheduling, surrogate, and multitask learning. She is an Associate Editor of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, *Expert Systems With Applications*, and *Swarm and Evolutionary Computation*. She is a Vice-Chair of the Task Force on Evolutionary Scheduling and Combinatorial Optimisation. She is the Vice-Chair of IEEE New Zealand Central Section.

Lei Wang (Member, IEEE) received the B.S. degree in electrical technology from the Jiangsu Institute of Technology, Changzhou, China, in 1992, and the M.S. and Ph.D. degrees in automation from Tongji University, Shanghai, China, in 1995 and 1998, respectively. He is currently a Professor with the Department of Control Science and Engineering, Tongji University. His research interest covers intelligent automation, computational intelligence, heuristic algorithm, and system engineering.

Qidi Wu (Senior Member, IEEE) received the B.S. degree in radio technology and the M.S. degree in automatic control from Tsinghua University, Beijing, China, in 1970 and 1981, respectively, and the Ph.D. degree in automation from ETH Zürich, Zürich, Switzerland, in 1986. She is currently a Professor with the Department of Electronics and Information Engineering, Tongji University, Shanghai, China. Her research interest covers control theory and engineering, intelligent automation, heuristic algorithm, complex systems scheduling and optimization, system engineering and management engineering, and smart scheduling of home energy management system.

Mengjie Zhang (Fellow, IEEE) received the Ph.D. degree in computer science from RMIT University, Melbourne, VIC, Australia, in 2000. He is currently a Professor of computer science, the Director of Centre for Data Science and Artificial Intelligence, Victoria University of Wellington, New Zealand. He has authored or coauthored more than 800 research papers in refereed international journals and conferences. His current research interests include evolutionary machine learning, genetic programming, image analysis, feature selection and reduction, job shop scheduling, and evolutionary deep learning and transfer learning. He is a Fellow of the Royal Society of New Zealand, a Fellow of Engineering New Zealand and an IEEE Distinguished Lecturer.