Affinity Propagation Hierarchical Memetic Algorithm for Multimodal Multi-objective Flexible Job Shop Scheduling with Variable Speed

Cong Luo, Xinyu Li, *Member, IEEE*, Wenyin Gong, *Member, IEEE*,
and Liang Gao, *Senior Member, IEEE*

***Abstract*—The flexible job shop scheduling, as the most typical production mode in industrial manufacturing, aims to improve production efficiency. However, the proposal of energy-saving and emission-reduction policy implies that it is impossible to increase the processing speed to improve productivity, and energy consumption is also becoming another important optimization objective. For the multi-objective flexible job shop scheduling problem, the optimization process tends to converge faster in some regions. This is because different scheduling sequences obtain the same objective values, *i.e.,* there is a multimodal characteristic, which is still hardly investigated. Therefore, optimizing the decision space and the objective space simultaneously has become an urgent challenge that needs to be solved. To overcome the above challenges, we model the multimodal multi-objective flexible job shop scheduling problem with variable speed (MMFJSP-S) and propose an affinity propagation hierarchical memetic algorithm (APHMA) to minimize makespan and total energy consumption. Firstly, four problem-specific neighborhood structures are employed to enhance the convergence; Then, an affinity propagation clustering combined with the random forests strategy is proposed to classify the global and local Pareto sets; Finally, a hierarchical environmental selection strategy is designed to ensure the convergence and diversity in the decision and objective spaces. Evaluations against seven advanced algorithms on MK and DP benchmarks demonstrate the competitive performance of APHMA in solving MMFJSP-S.**

***Index Terms*—Flexible job shop scheduling, multimodal multi-objective optimization, energy-efficient, memetic algorithm (MA), affinity propagation clustering.**

I. Introduction

The popularity of the intelligent manufacturing concept is reshaping the traditional manufacturing industry to be more efficient, flexible, and intelligent, increasingly highlighting the importance of scheduling problems in the overall industrial manufacturing environment [1], [2]. Meanwhile, in most real-world production environments, products and processing equipment are often flexibly related. This scenario defined the flexible job shop scheduling problem (FJSP), recognized as NP-hard [3]. Therefore, the research of FJSP has become a non-negligible part of industrial manufacturing.

However, with the greenhouse effect and extreme weather, energy-efficient scheduling is quickly becoming a hot topic [4], [5]. Generally speaking, production efficiency is often closely linked to machine speed, faster processing speed will shorten the production cycle, but at the same time, it will also put the machine in a long time in the high load state, which will lead to an increase in energy consumption [6]. Therefore, it is of great interest to study the speed allocation in the production process. Extended based on FJSP, flexible job shop scheduling problem with variable speed (FJSP-S) is more complex because it adds a new constraint: speed selection, and performs multi-objective optimization.

Previous research on FJSP-S has primarily focused on optimizing the conflict objectives, and its main focus is to investigate methods for deriving the Pareto set (PS) from its corresponding Pareto front (PF) [7]. However, FJSP-S also has a multimodal property, *i.e.,* meaning there can be multiple PSs that map to the same point within the PF, which belongs to a typical multimodal multi-objective problem (MMOP). MMOPs are commonly highlighted in many practical applications, *e.g.,* engine design [8], mission optimization [9], and brain imaging optimization [10]. These examples illustrate instances where multiple PSs share identical objective values [11]. However, unlike these problems mentioned, the multimodal multi-objective FJSP-S (MMFJSP-S) is a discrete optimization problem whose decision space correlations are often difficult to capture. Moreover, since MMFJSP-S involves the coupling of several subproblems, its discrete property makes it very easy to fall into local optimality, which presents great obstacles to solving the problem.

Fig. 1 provide an example of an MMOP, demonstrating that solutions s_1, s_2, and s_3 in PS_1 are equivalent to the solutions s_1', s_2', and s_3' in PS_2, since they correspond to the same F_1, F_2, and F_3 in PF. However, for MMFJSP-S, multiple different scheduling sequences will obtain the same objective values, which will greatly increase the complexity of solving this problem. Fig. 2 shows an example of MMFJSP-S where two different scheduling sequences surprisingly obtain the same makespan value and total energy consumption value (the total energy objective calculations and the encoding and decoding scheme are explained in Sections III and IV-C, respectively). Few studies have mentioned the concept of multimodal in FJSP, but in the actual process of problem solving it can

This work was supported in part by the National Natural Science Foundation of China under Grant 52188102 and Grant U21B2029, and in part by the Fundamental Research Funds for the Central Universities under Grant 2024BRA004. *(Corresponding author: Xinyu Li.)*

C. Luo, X. Li and L. Gao are with State Key Lab of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China. (Email: congluo@hust.edu.cn; lixinyu@hust.edu.cn; gaoliang@hust.edu.cn)

W. Gong is with School of Computer Science, China University of Geosciences, Wuhan 430074, China. (Email: wygong@cug.edu.cn)

be found that the algorithms are prone to converge to some regions prematurely, which will greatly reduce the efficiency of the problem solving [12]. Therefore, the study of the multimodal FJSP will have the following significance. Firstly, investigating the connections between solutions in the decision space can help to jump out the local optimums and thus improve the search performance of the algorithms. Secondly, obtaining multiple solution sets can reveal the underlying characteristics of the problem to design effective search operators. Finally, providing solution sets of equal quality according to different decision-makers' preferences and increasing the likelihood of finding robust solutions [13].

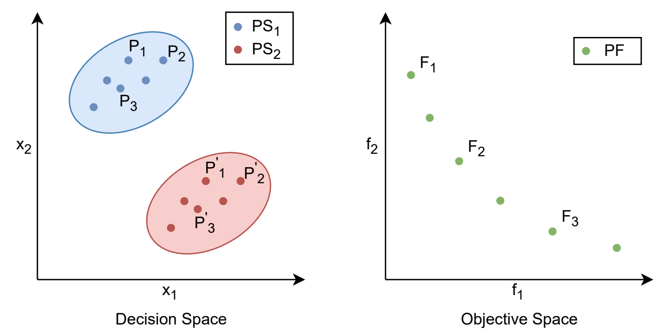


Fig. 1. Illustration of the MMOP.

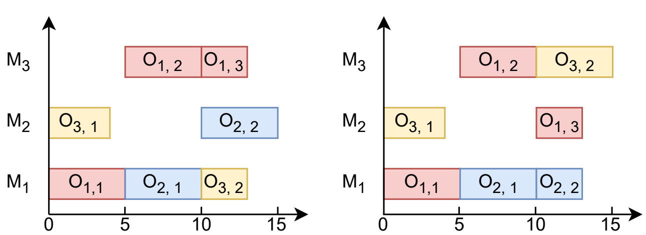


Fig. 2. Example of the MMFJSP-S (both scheduling sequences have the same setup time, processing time and idle time, and the machine speed remains the same, thus the total energy consumption value is also the same).

Although MMOPs can be treated as multi-objective optimization problems (MOPs). However, in general, solving MMOPs using multi-objective evolutionary algorithms (MOEAs) presents unique challenges. Most of them typically prioritize achieving convergence and diversity in the objective space while overlooking the decision space, which results in some PSs for the same PF being missed in the evolution process [14]. Efficiently designing multimodal MOEAs (MMOEAs) involves addressing three primary challenges: 1) maintaining convergence and diversity in the objective space for multi-objective optimization problems; 2) achieving convergence and diversity in the decision space for multimodal optimization problems; and 3) balancing convergence and diversity across both the objective and decision spaces for MMOPs.

Building on the aforementioned points, this study investigates the MMFJSP-S and proposes an affinity propagation hierarchical memetic algorithm (APHMA) to optimize both makespan and total energy consumption. The principal contributions are summarized as follows:

1) ***Problem Modelling:*** This work identifies the multimodal characteristics of FSJP-S and develops a corresponding mixed-integer linear programming (MILP) model. Moreover, a new perspective on FJSP-S is provided by the study of the MMFJSP-S.
2) ***Efficient Neighborhood Structures:*** Four problem-specific neighborhood structures are introduced to enhance convergence in the objective space, which focus on the crucial operation adjustment, machine assignment, and speed selection according to problem characteristics of MMFJSP-S.
3) ***Affinity Propagation Clustering Application:*** The affinity propagation clustering [15] is applied in the decision space to derive global PSs and local PSs, which in turn enhances the convergence in the decision space. Furthermore, a random forest approach is employed to extract key features, thus reducing the impact of dimensionality catastrophe.
4) ***Hierarchical Environmental Selection Strategy:*** A niche technique is formulated to eliminate solutions that exhibit similarity in the decision space, thereby promoting diversity in both the decision space and the objective space. Additionally, a hierarchical selection strategy is devised to preserve elite solutions, balancing convergence and diversity in both spaces.

This paper is structured as follows. Section II reviews related literature. Next, Section III defines MMFJSP-S and formulates a MILP model. Then, Section IV details the APHMA. Afterward, Section V presents and analyzes the experimental results. Lastly, Section VI summarizes findings and outlines future directions.

II. Related Works

A. Literature Review of FJSP

FJSP, as the most basic and mainstream production model, has been extended into much more complex research, *e.g.,* distributed FJSP [16], fuzzy FJSP [17], dynamic FJSP [18], assembly FJSP [19], stochastic FJSP [20], *et al.* The current mainstream approach to solving FJSP is to employ evolutionary algorithms combined with search operators, such as artificial bee colony algorithm embedded with local search operators [21], genetic algorithm with problem-specific variable neighborhood search [22], and MOEAs incorporate with neighborhood search structures [23], [24].

For the research of FJSP, Li and Gao [25] presented a hybrid algorithm based on the genetic algorithm and tabu search to refresh the benchmarks of FJSP, which is a very pioneering work. Li [26] *et al.* introduced a new co-evolutionary framework for distributed FJSP and applied deep Q-networks to accumulate the advantages of search operators. Yao [27] *et al.* developed a knowledge-based MOEA for solving FJSP, leveraging domain expertise to balance exploration and exploitation. Kasapidis [28] *et al.* established a constraint programming model to guide population search and effectively address resource allocation in FJSP. Zhang [29] *et al.* utilized evolutionary multitasking techniques to design genetic programming for solving FJSP with knowledge sharing to ensure the quality of elite solutions.

Recent studies have predominantly concentrated on optimizing within the objective space while neglecting the multimodal characteristics. This will greatly increase the possibility of falling into local optimums, which greatly wastes computational resources. Therefore, the research of multimodal FJSP will endeavor to overcome the above challenges and obtain better solutions with fewer computational resources. Our work provides a new research perspective and obtains more optimal solutions for decision-makers to choose from.

B. Multimodal Multi-objective Optimization

Recently, MMOEAs are categorized into three main types: 1) Pareto preferenced, 2) decomposition preferenced, and 3) indicator preferenced.

1) Pareto Preferenced: This MMOEAs uses Pareto dominance to solve MMOPs. The Omni-optimizer [30] and DN-NSGA-II [31] were pioneering designs for MMOPs. Liu [32] *et al.* introduced TriMOEA-TA&R, employing a dual archive strategy to preserve preferred individuals in different spaces. Wang [33] *et al.* proposed MO_Ring_CSO_SCD using a ring topology and a special congestion distance. Lin *et al.* [34] presented MMOEA/DC, integrating dual clustering in different spaces to balance global and local PSs. Zhang [35] *et al.* developed DN-MMOES, employing staged niche strategies and a density adaptive approach to enhance decision space density balance. In HREA [36], a hierarchy ranking strategy is designed to be used for population updating and to obtain local PSs by controlling a predefined parameter. More recently, Wei [14] *et al.* transformed the optimization objectives into convergence and diversity metrics to simultaneously optimize the objective and decision spaces.

2) Decomposition Preferenced: This MMOEAs decomposes the complex problem into sub-problems to be solved. Sun [37] *et al.* adopted a decomposition-based approach to maintain objective space diversity and introduced a dynamic subdivision distance to preserve decision space diversity. Tanabe and Ishibuchi [38] developed a decomposition-based framework that ensures diversity by comparing subproblems with their neighbors. Hu [39] *et al.* applied clustering techniques to determine the number and placement of Pareto sets and further explored the solution space based on these clusters.

3) Indicator Preferenced: This MMOEAs performs evolutionary optimization based on indicators. Liu [40] *et al.* introduced CPDEA, which employs a convergence-penalized density indicator to assess solution distances by converting them into density values. The MMEA-WI [41] designed by Li *et al.* developed a weighted metric for evaluating the potential convergence ability of individuals as a criterion for selecting parents. Xie [42] *et al.* developed a fuzzy preference indicator to provide the search directions for populations.

While numerous MMOEAs have been developed, most are tailored for continuous optimization problems and do not adequately address discrete multimodal multi-objective scheduling issues.

C. Memetic Algorithm Applications

The memetic algorithm improves the search performance of evolutionary algorithms by embedding local search strategies, which has been applied to many problems, *e.g.,* scheduling optimization [43], [44], many-objective optimization [45], feature selection optimization [46], and multi-task optimization [47]. Previous research shows that memetic algorithms are effective for complex optimization problems.

III. Problem Description

The MMFJSP-S involves n jobs and m machines, where each job comprises n_i operations assigned to specific machines, and each machine has a selectable speed level. The key to solving MMFJSP-S is to address three subproblems: 1) operation sequencing, 2) machine assignment, and 3) speed selection. The processing speed can be selected from low, medium-low, medium, medium-high, and high levels, and the faster the processing speed, the higher the production efficiency. At the same time, it will also increase energy consumption. Compared to the FJSP, the MMFJSP-S needs to consider a more coupled subproblem of speed selection, which undoubtedly enlarges the dimensions of the decision space, exacerbates the possibility of falling into local optimums, and greatly increases the computational resource requirements. MMFJSP-S aims to optimize both the makespan objective and the total energy consumption objective while maximizing the number of optimal solutions obtained. The calculation formulas for the two optimization objectives are as follows.

Part of the notations:

1) Indices:

i: index of jobs, $i = 1, 2, ..., n$;

j: index of operations, $j = 1, 2, ..., n_i$;

k: index of processing machines, $k = 1, 2, ..., m$;

l: index of positions on the machine, $l = 1, 2, ..., h_k$;

q: index of machine speed levels, $q = 1, 2, ..., s$;

2) Parameters:

J_i: the ith job;

$O_{i,j}$: the jth operation for J_i;

M_k: the kth processing machine;

$D_{k,l}$: the lth position on M_k;

V_q: the qth speed level;

n: number of jobs;

n_i: number of operations for J_i;

m: number of processing machines;

h_k: number of positions for M_k;

SP_k: actual setup power of M_k;

$PP_{k,q}$: actual processing power of M_k at V_q;

$IP_{k,q}$: actual idle power of M_k at V_q;

ST: setup time of the machines;

$T_{i,j,k,q}$: actual processing time of $O_{i,j}$ on M_k at V_q;

3) Ordinary variables:

C_{max}: makespan value;

TEC: total energy consumption value;

SE: setup energy consumption value;

PE: processing energy consumption value;

IE: idle energy consumption value;

$C_{i,j}$: completion time of $O_{i,j}$;

$B_{k,l,q}$: start time of $D_{k,l}$ at V_q;

$F_{k,l,q}$: completion time of $D_{k,l}$ at V_q;

4) Decision variables:

$\mathbf{X}_{i,j,k,l,q}$: equal to 1 if $O_{i,j}$ is processed on $D_{k,l}$ at V_q, otherwise equal to 0;

$$\min\ F_1 = C_{max} = \max\ C_{i,j}, \forall i = 1, ..., n; j = 1, ..., n_i \quad (1)$$

$$\min\ F_2 = TEC = SE + PE + IE \quad (2)$$

$$SE = \sum_{k=1}^{m} SP_k \cdot ST \quad (3)$$

$$PE = \sum_{i=1}^{n} \sum_{j=1}^{n_i} \sum_{k=1}^{m} \sum_{l=1}^{m_j} \sum_{q=1}^{s} PP_{k,q} \cdot T_{i,j,k,q} \cdot \mathbf{X}_{i,j,k,l,q} \quad (4)$$

$$IE = \sum_{k=1}^{m} \sum_{l=1}^{h_k-1} \sum_{q=1}^{s} IP_{k,q} \cdot (B_{k,l+1,q} - F_{k,l,q}) \quad (5)$$

Makespan measures the longest time required to complete all operations, whereas total energy consumption includes setup, processing, and idle energy. The relevant notations have been placed in the supplementary file.

MMFJSP-S has the following assumptions:

- All operations and machines are initialized at time zero.
- The processing time and the power are determined.
- Each machine can handle only one operation at a time.
- Machines cannot switch operations until completing the current one.
- The machine cannot change speed levels until the current operation has been completed.
- The machines generate a fixed setup time only on startup and shutdown.

Based on the MMFJSP-S, a MILP model was developed, which is included in the supplementary file due to space constraints.

IV. Proposed Approach: APHMA

A. Motivation

To design efficient MMOEAs to solve MMFJSP-S, the following four problems need to be solved urgently.

1) *Issue 1:* Ensuring convergence in the objective space.
2) *Issue 2:* Maintaining convergence in the decision space.
3) *Issue 3:* Preserving diversity in both the objective and decision spaces.
4) *Issue 4:* Balancing optimization across the objective and decision spaces.

Regarding *Issue 1*, we design four problem-specific neighborhood structures to enhance convergence in the objective space. The operation sequencing, machine allocation, and speed selection are adjusted based on the characteristics of MMFJSP-S to optimize both objectives simultaneously.

For *Issue 2*, affinity propagation clustering is employed in the decision space to derive global PSs and local PSs. Unlike traditional methods, affinity propagation clustering does not need to provide the number of clusters, which greatly frees it from the influence of parameters on the clustering results [15].

As for *Issue 3*, solutions that are proximal in the decision space typically exhibit similarity in the objective space as well [34]. To avoid falling into local optimums, a niche technique is devised to remove neighboring solutions, thus preserving diversity across both decision and objective spaces.

In *Issue 4*, the optimization process is to find the optimal values in the objective space by exploring the decision space, thus PSs and PFs are one-to-one mappings, and retaining the PFs in the objective space also means keeping the PSs in the decision space at the same time. Therefore, we propose a hierarchical environmental selection strategy for population updating, which balances the optimization of the decision space and the objective space by maintaining global and local PFs in a hierarchical manner.

B. Algorithm Architecture

APHMA is a Pareto preferenced algorithm whose overall architecture is shown in Algorithm 1. First, the population performs a hybrid initialization strategy. Second, a mating pool is constructed using the tournament approach to perform genetic operations. Third, a hierarchical environment selection strategy is applied to select a subset of the population for variable neighborhood search, exploring the solution space. Afterward, the hierarchical environment selection strategy updates the population while preserving the archive set. Lines 2-11 loop iterates only until the stopping criterion is satisfied. Lastly, energy saving strategies are employed for further optimization. It is worth noting that the hierarchical environment selection strategy is utilized twice, but both times for different purposes. The first time, it is utilized to select a portion of the elite individuals to perform the variable neighborhood search strategy, thus fully exploring and exploiting the solution space. The second time, it is utilized to update the population and retain the elite solutions for the next iteration.

C. Encoding and Decoding

The encoding and decoding schemes of MMFJSP-S are introduced as follows.

Encoding scheme: The scheduling of MMFJSP-S is represented by a scheduling sequence comprising three components: operation sequence (OS), machine sequence (MS), and speed sequence (SS). This is a more general encoding method [48], where the number in OS indicates the job number and the number of times it appears indicates the operation number of this job. The number in MS indicates the processing machine number, which is constructed based on the order of the jobs, and the number in SS indicates the speed level of the processing machine, which is a one-to-one mapping with MS. An example of coding is shown in Fig. 3.

Decoding scheme: As shown in Fig. 3, the numbers within the operation sequence denote the number of operations for each job. The machine sequence assigns machines in accordance with the job's operation order, while the speed

Algorithm 1 Architecture of APHMA

Input: population size (N), maximum number of function evaluations ($MaxFEs$), crossover rate (P_c), mutation rate (P_m), clustering threshold (β), niche threshold (σ)

Output: Pareto solutions (PS), Pareto front (PF)

$[P,\ \Omega]$ = Hybrid_Initialization (N); (Section IV-D)
while $FEs \leqslant MaxFEs$ **do**
  M = Tournament_Selection (P);
  Q = Genetic_Operations (M, P_c, P_m); (Section IV-E)
  $P = P \cup Q,\ FEs = FEs + N$;
  S = Hierarchical_Environmental_Selection ($P, \beta, \sigma, PS/4$); (Section IV-H)
  Q' = Variable_Neighborhood_Search (S); (Section IV-F)
  $P = P \cup Q',\ FEs = FEs + N$;
  P = Hierarchical_Environmental_Selection (P, β, σ, N); (Section IV-H)
  Ω = Update_Archive (P, Ω);
end while
$[PS,\ PF]$ = Energy-Saving_Strategy (Ω); (Section IV-I)

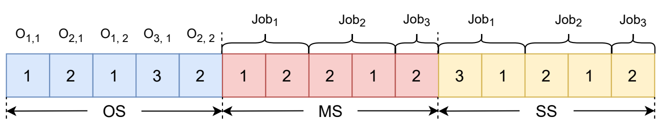


Fig. 3. Example of encoding scheme (3 jobs with 2 operations at most, 2 machines, and 3 speed levels).

sequence correlates each machine with a specific processing speed. Through this decoding scheme, each operation of the jobs is assigned to the machine and set to the corresponding processing speed.

D. Initialization

To achieve a high-quality initial population, a hybrid initialization strategy was devised.

1) Minimum workload rule $\mathcal{R}_1$: Assign the operations to the machine with the smallest workload, or randomly select it if there is more than one machine.

2) Earliest available rule $\mathcal{R}_2$: Allocate the operations to the earliest available machine, or randomly select it if there is more than one machine.

3) Shortest processing time rule $\mathcal{R}_3$: Assign the operations to the machine with the shortest processing time, or randomly select it if there is more than one machine.

4) Decelerate rule $\mathcal{R}_4$: Randomly select a machine and reduce its speed by one level, keeping it unchanged if the speed is at the lowest level.

The $\mathcal{R}_1 - \mathcal{R}_3$ are proposed for machine assignment and the $\mathcal{R}_4$ is developed for speed selection. To preserve population diversity, $\mathcal{R}_1 - \mathcal{R}_4$ are each executed with a probability of 0.15, and the other probabilities are randomly initialized.

E. Genetic Operators

Crossover and mutation are essential genetic operations for thoroughly exploring the solution space. In APHMA, the improved precedence operation crossover(IPOX) and multi-point crossover (MPX) operators are employed for crossover operation, and the inversion mutation (IM) and multi-point mutation (MPM) operations are utilized for mutation operation [48]. The procedural steps are detailed as follows: 1) A random number p was generated. If $p < P_c$ then two individuals are selected from the mating pool, then IPOX was performed for OS and MPX for MS and SS. 2) Similarly a random number p is generated. If $p < P_m$ then an individual is selected after crossover, then IM is performed for OS and MPM for MS and SS. The IPOX, MPX, IM, and MPM are detailed in the supplementary file.

F. Problem-specific Neighborhood Structures

For MMFJSP-S, the makespan is determined by its critical path, and optimizing critical operations along this path effectively reduces production cycle time [49]. However, if multiple critical paths exist, mere adjustments to critical operations may not necessarily shorten them. In this case, it is necessary to adjust the overlapping operations of the critical path, named crucial operations, to optimize the critical path [50]. Therefore, four problem-specific neighborhood structures are developed to solve the MMFJSP-S efficiently.

1) Neighborhood Structure ($\mathcal{NS}_1$): Xie [51] *et al.* demonstrated that relocating operations within the critical path block effectively reduces makespan and proposed the N8 neighborhood structure. Therefore, N8 extends N7 [49] can be applied to solve the MMFJSP-S which takes more into account moving inner operations out of the critical path block, then moving the first operation on the critical path block after the latter operation of the last operation, and finally moving the last operation on the critical path block before the previous operation of the first operation. The critical path length can be optimized by moving critical operations. Fig. 4 shows this process.

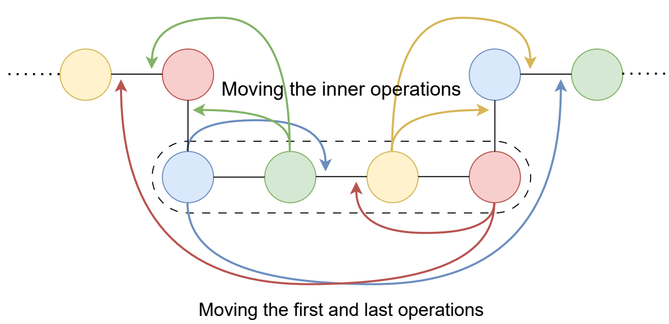


Fig. 4. Example of N8 neighborhood structure.

2) Neighborhood Structure ($\mathcal{NS}_2$): Machine reassignment neighborhood structure (MRNS) randomly selects a crucial operation and inserts it into a random candidate machine; if no such crucial operation exists, randomly select a critical operation. The makespan can be reduced by destroying critical path blocks.

3) Neighborhood Structure ($\mathcal{NS}_3$): Minimum time neighborhood structure (MTNS) randomly selects one of the last completed crucial operations and assigns it to the candidate machine with the shortest processing time; if no such crucial operation exist, randomly select a critical operation. The makespan and total energy consumption are optimized simultaneously by reducing the processing time of the overall operations.

4) Neighborhood Structure ($\mathcal{NS}_4$): Deceleration neighborhood structure (DNS) randomly selects an operation from the non-critical path and decreases the processing speed level of its assigned machine. The total energy consumption can be further reduced by decelerating the speed level of the processing machine for the non-critical operation.

The variable neighborhood search (VNS) is shown in Algorithm 2, where N8, MRNS, MTNS, and DNS are based on the characteristics of MMFJSP-S.

Algorithm 2 Variable_Neighborhood_Search

Input: elite population (E)
Output: population obtained by performing VNS (P)

```
P = ∅
for i = 1 to length(E) do
    for j = 1 to 4 do
        Π = NS_j (E(i, :));
        P = P ∪ Π;
    end for
end for
```

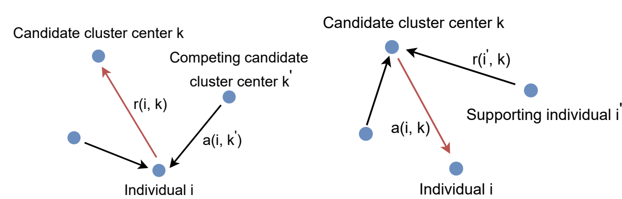
G. Affinity Propagation Clustering with Feature Extraction

To enhance convergence within the decision space, the idea of clustering can be applied to preserve both global and local PSs. Unlike other k-center clustering methods, the affinity propagation algorithm is adopted to cluster in the decision space, operating without requiring a predetermined number of clusters.

The affinity propagation clustering was published in *Science* 2007 [15], which achieves efficient and accurate clustering by iteratively transmitting responsibility information and availability information between clustered individuals, and finding the true clustering centers instead of generating new clustering centers. In the affinity propagation clustering, the $s(i, j)$ in the similarity matrix S records the similarity between individuals, the $r(i, k)$ in the responsibility matrix R indicates how well individual k serves as a cluster center for individual i, and the $a(i, k)$ in the availability matrix A denotes the suitability of individual i choosing individual k as its clustering center.

However, the high complexity of MMFJSP-S results in high-dimensional decision variables that bring many computational and optimization challenges. Therefore, to cope with the curse of dimensionality and save computational resources, the random forests (RF) strategy [52] is utilized to extract representative features. The specific procedure of affinity propagation clustering with feature extraction is as follows.

1) Normalize all individuals and extract representative decision variables utilizing the RF strategy. For computational efficiency, employ 100 decision trees and select a number of decision variables equal to the square root of the total operations count.
2) Initialize the matrices R and A to 0 and record the negative value of Euclidean distance between individuals to the matrix S. For the preference $s(i, i)$ of i as the center of clustering, it is generally replaced by the median in S. The maximum iterations $MaxIter$ is set to 500 to ensure successful clustering.
3) Update the responsibility matrix R, which accumulates evidence that individual k serves as a clustering center for individual i. Fig. 5(a) illustrates this process.
$$r_{t+1}(i,k) = s(i,k) - \max_{k' s.t. k' \neq k} \{s(i,k') + a_t(i,k')\} \quad (6)$$
where, $a_{i,k'}$ denotes the attribution value of individuals other than k to individual i.
4) Update the availability matrix A, which presents the cumulative proof that individual i chooses individual k as its clustering center. Fig. 5(b) illustrates this process.
$$a_{t+1}(i,k) = \min\{0, r_t(k,k) + \sum_{i' s.t. i' \notin \{i,k\}} \max\{0, r_t(i',k)\}\}, i \neq k \quad (7)$$
$$a_{t+1}(k,k) = \sum_{i' s.t. i' \notin \{i,k\}} \max\{0, r_t(i',k)\}, i = k \quad (8)$$
where, $r_{i',k}$ denotes the similarity value that represents individual k as the center of clustering of individuals other than i.
5) To prevent oscillations, introduce a damping factor λ, commonly set to 0.5.
$$r_{t+1}(i,k) = r_t(i,k) * \lambda + r_{t+1}(i,k) * (1 - \lambda) \quad (9)$$
$$a_{t+1}(i,k) = a_t(i,k) * \lambda + a_{t+1}(i,k) * (1 - \lambda) \quad (10)$$
6) Iterate 1) - 5) continuously until the iteration criterion is satisfied.



(a) Update the responsibility matrix R. (b) Update the responsibility matrix A.

Fig. 5. Update the responsibility matrix R and the availability matrix A.

After the global PSs are selected in the hierarchical environmental selection strategy (described in the next subsection), affinity propagation clustering with feature extraction is executed for the remaining populations. Firstly, feature extraction is performed using RF and the extracted features are employed for the initialization of R, A, and S. Afterward, R and A are continuously updated within $MaxIter$ iterations, which ultimately ensures the convergence of the clustering. Furthermore, to ensure the quality of global PSs and local PSs, clusters are preserved if their size exceeds the parameter β. Eventually, local PSs can be extracted from the clusters to ensure the diversity of the decision space. The additional

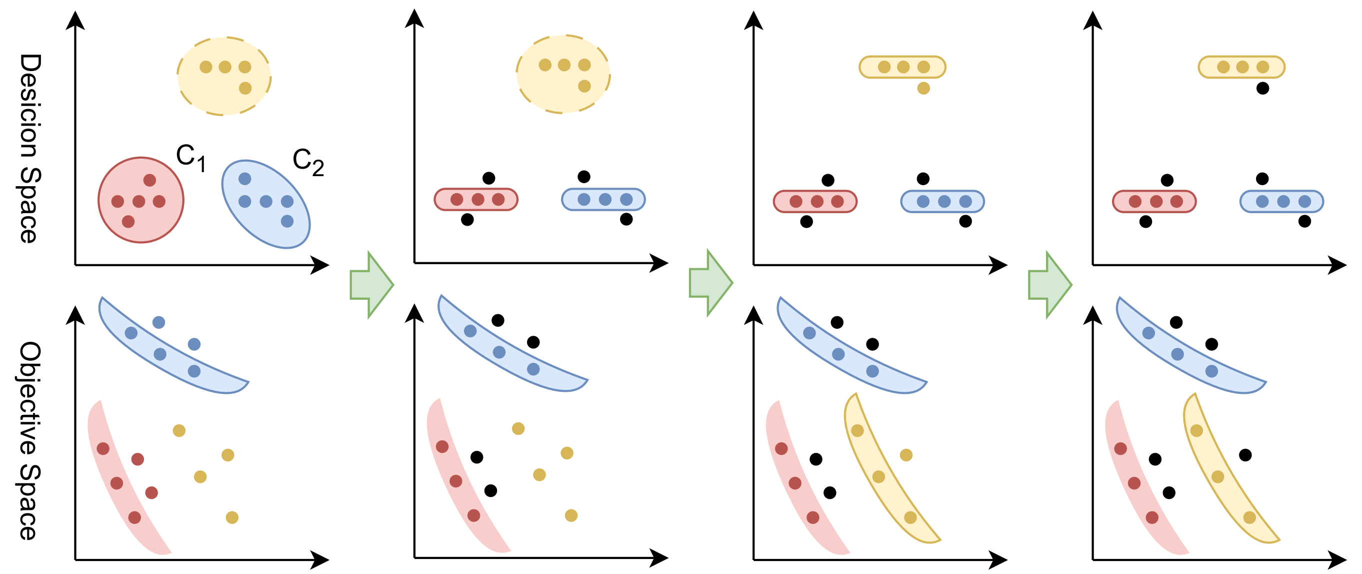


Fig. 6. Example of hierarchical environment selection (The clustering threshold β is 5).

Algorithm 3 AP_Clustering

Input: population (P), clustering threshold (β)
Output: clusters ($C_1, ..., C_k$)

$F =$ Feature_Extraction (P);
$[R, A, S] =$ Initialization (F);
for $i = 1$ to $MaxIter$ **do**
  Update the responsibility matrix R by Eq. (6);
  Update the availability matrix A by Eq. (7) - (8);
  Avoiding oscillations Eq. (9) - (10);
  $[C_1, ..., C_t]$ = Obtain_Clusters (R, A);
end for
$[C_1, ..., C_k]$ = Cluster_Filtering ($[C_1, ..., C_t], \beta$);

computational consumption consists of feature extraction and clustering convergence. More details are described in Algorithm 3.

H. Hierarchical Environmental Selection

A hierarchical environment selection strategy is proposed for balancing the optimization between the objective space and the decision space. Since there is a mapping relationship between PSs and PFs, local PFs can be preserved by retaining local PSs, even if their objective values are inferior to global PFs. Furthermore, since the solutions adjacent in the decision space will be close in the objective space [34], a niche technique is developed to remove the neighboring solutions near the PSs, thus ensuring the diversity in both the decision and objective spaces.

The process of the hierarchical environment selection strategy is illustrated in Algorithm 4. First, nondominated solutions are identified to form the global PFs (Lines 3-4). Then, affinity propagation clustering is adopted to perform clustering and find the nondominated solutions in each cluster to compose the local PFs (Lines 5-7). Next, the individuals whose distances from the global PSs and the local PSs are less than σ are deleted to ensure diversity (Lines 8-9). Finally, an elite retention strategy is employed to determine whether the number of elite populations exceeds N, and if it does, the local PSs of the generation are randomly deleted (Lines 10). If the elite populations are less than N after the iteration cycle, the population is replenished with the hybrid initialization strategy (Lines 12-15).

Algorithm 4 Hierarchical_Environmental_Selection

Input: population (P), population size (N), clustering threshold (β), niche threshold (σ)
Output: elite population (E)

1: $E = \emptyset$;
2: **while** $size(E, 1) < N$ and $\sim isempty(P)$ **do**
3:   $globalPop =$ Find_Nondominated_Solutions (P);
4:   $P = P \setminus \{globalPop\}$;
5:   $[C_1, ..., C_k]$ = AP_Clustering (P, β); (Section IV-G)
6:   $localPop$ $=$ Find_Nondominated_Solutions ($[C_1, ..., C_k]$);
7:   $P = P \setminus \{localPop\}$;
8:   P = Niche ($P, globalPop, localPop, \sigma$);
9:   $E = E \cup globalPop \cup localPop$;
10:   $E =$ Elite_Reservation (E);
11: **end while**
12: **if** $size(E, 1) < N$ **then**
13:   $P =$ Hybrid_Initialization ($N - size(E, 1)$); (Section IV-D)
14:   $E = E \cup P$;
15: **end if**

The selection process is presented in Fig. 6: 1) Employ affinity propagation clustering to obtain global PSs and local PSs (individuals enclosed by the yellow dotted line cannot be treated as a cluster because their number is less than β); 2) Select global PSs and local PFs based on nondominated relationships; 3) Delete individuals closed to global PSs and local PSs via the niche technique (black dots indicate deleted individuals); 4) Select the next level. Repeat 1) - 4) until conditions are met.

I. Energy-Saving Strategy

It is evident from Section III that the total energy consumption is composed of setup energy consumption, processing energy consumption, and idle energy consumption. Since the setup and processing times are fixed in the optimized scheduling sequence, further optimization can focus on reducing idle time [48]. A right-shift based energy-saving strategy is proposed for optimizing further total energy consumption without increasing the makespan. It traverses all the operations in the OS in reverse order, which ensures that the later processed operations can be handled earlier without considering the constraints of the successor operations, thus avoiding infeasible moves. After that, the operations can be shifted right if the constraints of processing are still satisfied after the right shift. The relevant pseudo-code can be found in the supplementary file.

An example of the energy-saving strategy is given in Fig 7. Traversing the operation sequence in reverse order, it can be noticed that $O_{2,1}$ can be shifted right while satisfying the constraints. The traversal continues after $O_{2,1}$ is shifted, and $O_{3,1}$ can be similarly right-shifted. Therefore, the idle time is reduced by the energy-saving strategy, thus further optimizing the total energy consumption.

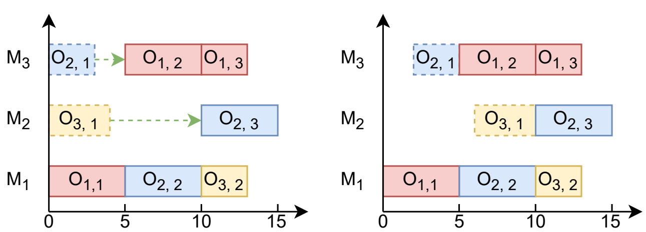


Fig. 7. Example of energy-saving strategy (the operation sequence is {1, 3, 2, 2, 1, 3, 2, 1} and the machine sequence is {1, 3, 3, 3, 1, 2, 2, 3}).

J. Time Complexity of APHMA

The time complexity of APHMA can be broken down into several components: 1) The genetic operations require a time complexity of $O(MaxFEs * 1/2 * N)$, where $MaxFEs$ is the maximum number of function evaluations and N is the population size; 2) The variable neighborhood search requires a time complexity of $O(MaxFEs * 1/4 * N)$; 3) the hierarchical environment selection requires a time complexity of $O(MaxFEs * MaxIter * N^2)$, where the $MaxIter$ is the maximum number of iterations for affinity propagation clustering; 4) the energy-saving strategy requires a time complexity of $O(|\Omega| * SH^2)$, where Ω is the archive and SH is the total number of operations. Based on this breakdown, the overall time complexity of APHMA is $O(MaxFEs * MaxIter * N^2)$.

V. Experimental Results and Analysis

A. Experimental Instances and Evaluation Indicators

To evaluate the performance of APHMA, all algorithms are implemented in MATLAB. Moreover, two FJSP benchmarks (MK [53] and DP [54]) are employed for this experiment. The basic setup power is uniformly distributed between [15, 20], the basic processing power is uniformly distributed between [5, 10], and the basic idle power is distributed between [1, 5], detailed in Table S-I in the supplementary file. Furthermore, the inverted generational distance in the decision space (IGDX) [55], inverted generational distance (IGD) [13], and hypervolume (HV) [56] are used as evaluation indicators, which are detailed in the supplementary file.

B. Model Validation

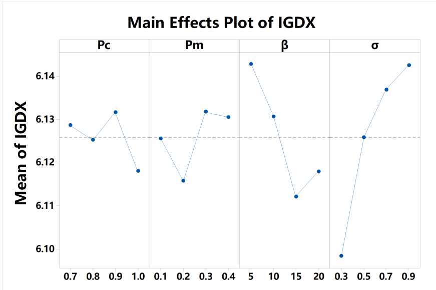
The ILOG CPLEX solver version 20.1 is applied to verify the correctness of the MILP model. However, it can only handle small-scale problems. While the smallest scale instance, *i.e.,* MK01, has reached $10 \times 6 \times 6 \times 5 = 1800$ dimensions of decision-making, the solution time of ILOG CPLEX solver will reach an unacceptable level as the dimensions increase. Therefore, we randomly generated five small-scale instances for validation based on MK01 with processing times randomly distributed in the interval [0, 100]. Since the ILOG CPLEX solver can only solve single-objective optimization problems, thus the two objectives can only be optimized separately. The results obtained by ILOG CPLEX solver are shown in Table I, where all instances can be solved, thus validating the correctness of the proposed MILP model.

TABLE I
Results Obtained by ILOG CPLEX Solver.

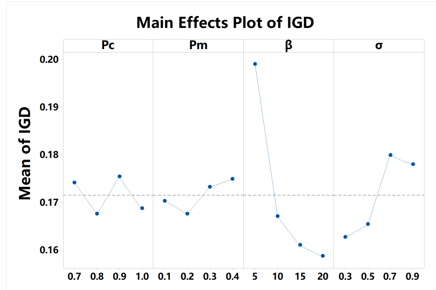
Instance	Jobs	Machines	Operations	Speeds	C_{max}	TEC
MK01	10	6	6	5	22.0	5.73E+03
Ins01	3	3	3	3	67.3	3.09E+03
Ins02	3	5	3	3	32.3	1.83E+03
Ins03	4	3	3	5	34.6	2.94E+03
Ins04	4	5	3	5	22.8	2.60E+03
Ins05	5	5	3	5	25.2	3.33E+03

C. Parameter Sensitivity Experiment

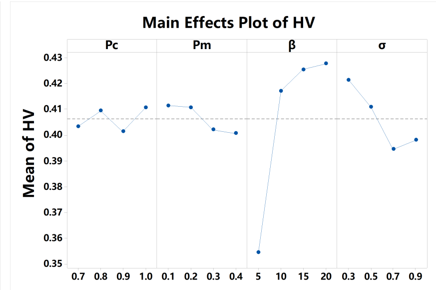
Parameter settings significantly influence algorithm performance, motivating the use of Taguchi design-of-experiments (DOE) [57] to identify the optimal parameter combination. The proposed APHMA involves four important parameters: 1) crossover rate P_c; 2) mutation rate P_m; 3) clustering threshold β; 4) niche threshold σ. Each parameter has four levels: $P_c = \{0.7, 0.8, 0.9, 1.0\}$, $P_m = \{0.1, 0.2, 0.3, 0.4\}$, $\beta = \{5, 10, 15, 20\}$, and $\sigma = \{0.3, 0.5, 0.7, 0.9\}$. Therefore, an orthogonal matrix $L_{16}(4^4)$ is constructed with 16 parameter combinations. To ensure robustness, each parameter combination is independently evaluated 20 times. For the remaining parameters, the population size N is set to 100, and the maximum number of function evaluations $MaxFEs$ is set to 50000. For the affinity propagation clustering with feature extraction, the number of decision trees n_{dt} is set to 100, and



(a) IGDX indicator



(b) IGD indicator



(c) HV indicator

Fig. 8. Main effects plots of three indicators.

the maximum number of iterations $MaxIter$ is set to 500 to save computational resources while ensuring convergence. As for the damping factor λ, it is usually set to 0.5 based on experience. Figure 8 illustrates the main effects plots of three indicators. Synthesizing the results of the above experiments observed, the optimal parameter configuration identified is $P_c = 1.0$, $P_m = 0.2$, $\beta = 20$, and $\sigma = 0.3$.

D. Ablation Experiments

To analyze the effectiveness of each proposed strategy and combinations of each improvement, seven variants of the algorithms are compared: 1) APHMA-BV: APHMA without problem-specific neighborhood structures, affinity propagation clustering with feature extraction, and hierarchical environment selection strategy; 2) APHMA-NS: APHMA with problem-specific neighborhood structures only; 3) APHMA-AP: APHMA with affinity propagation clustering with feature extraction only; 4) APHMA-ES: APHMA with hierarchical environment selection strategy only; 5) APHMA-NNS: APHMA without problem-specific neighborhood structures, *i.e.,* with the combination of affinity propagation clustering with feature extraction and hierarchical environment selection strategy; 6) APHMA-NAP: APHMA without affinity propagation clustering with feature extraction, *i.e.,* with the combination of problem-specific neighborhood structures and hierarchical environment selection strategy; 7) APHMA-NES: APHMA without hierarchical environment selection strategy, *i.e.,* with the combination of problem-specific neighborhood structures and affinity propagation clustering with feature extraction. Since the affinity propagation clustering with feature extraction and the hierarchical environment selection strategy are nested. When only the affinity propagation clustering with feature extraction is utilized, then non-dominated individuals are retained in the effective clusters. When only the hierarchical environment selection strategy is employed, then elite individuals are retained hierarchically based on the niche technique. To ensure fairness, each algorithm is run 20 times independently with a stopping criterion of $MaxFEs = 50000$.

The statistical results for the three indicators are presented in Table S-II, and the best results are shown in **bold**. The Wilcoxon rank-sum test is implemented in Table S-II, with symbols '+', '-', '=' indicating that the competitor's performance is significantly better than, worse than, or similar to APHMA. From Table S-II, it can be concluded that APHMA gets the best results and APHMA-BV gets the worse results, thus verifying the effectiveness of each proposed strategy. Moreover, to extract the key improvement component, the Friedman test experiment is set up and its results are recorded in Table II. From Table II, it can be analyzed that for the IDGX indicator, the hierarchical environmental selection strategy plays a dominant role, illustrating its superiority for optimizing the decision space. For the IGD and HV indicators, the problem-specific neighborhood structures strategy dominates, which further reflects the fact that it can fully explore and exploit the objective space. However, the affinity propagation clustering with feature extraction strategy did not work for the IGD and HV indicators, this is because it is originally designed for the decision space and a single affinity propagation clustering with feature extraction strategy makes it hard to obtain superior results.

TABLE II
FRIEDMANN TEST RESULTS FOR ALL VARIANT ALGORITHMS.

Algorithms	IGDX		IGD		HV	
	rank	p-value	rank	p-value	rank	p-value
APHMA-BV	7.00	**0.00E+00**	5.95	**0.00E+00**	6.38	**0.00E+00**
APHMA-NS	4.00	**9.82E-03**	2.45	4.39E-01	2.88	**2.19E-02**
APHMA-AP	4.25	**3.68E-03**	6.90	**0.00E+00**	6.93	**0.00E+00**
APHMA-ES	3.40	7.07E-01	3.40	**4.54E-02**	3.98	**2.06E-04**
APHMA-NNS	7.25	**0.00E+00**	6.20	**0.00E+00**	6.10	**0.00E+00**
APHMA-NAP	4.00	**9.82E-03**	2.90	1.75E-01	2.25	1.38E-01
APHMA-NES	4.10	**6.71E-03**	6.35	**0.00E+00**	6.40	**0.00E+00**
APHMA	**2.00**		**1.85**		**1.10**	

E. Comparison Experiments and Analysis

In this respect, the APHMA is compared with seven advanced algorithms to further verify its superiority. Three types of algorithms are listed as follows:

1) *MOEAs:* MOEA/D [58], NSGA-II [59], and MOEA/D-AAWN [60].
2) *MMOEAs:* HREA [36] and BOEA [14].
3) *Proposed for FJSP-S:* TIE [61] and ACML-BCEA [6].

All parameter settings are documented in Table S-III, and all comparison algorithms are run independently 20 times with the stopping criterion of $MaxFEs = 50000$.

The statistical results for all indicators are presented in Tables S-IV, where the best results are shown in **bold**. The symbols '+', '-', '=' denote whether a competitor algorithm significantly outperforms, underperforms, or performs similarly to APHMA. For the IGDX indicator, APHMA consistently outperforms most comparison algorithms in most instances and is only inferior to ACML-BCEA on the MK03 and MK08 instances because the improved angle-penalized distance can select PSs based on the shape of the PFs. For the IGD indicator, APHMA also shows good competition as well. APHMA is worse than HREA on 2 instances, BOEA on 5 instances, and ACML-BCEA on 4 instances. This is because of the hierarchy ranking method of HREA which can ameliorate the lack of convergence, and the objective transformation framework of BOEA which can balance the conflict between convergence and diversity. For the HV indicator, APHMA demonstrates its superiority except on the MK04 instance. Collectively, APHMA can demonstrate its superiority or be highly competitive.

TABLE III
FRIEDMANN TEST RESULTS FOR ALL COMPARISON ALGORITHMS.

Algorithms	IGDX		IGD		HV	
	rank	p-value	rank	p-value	rank	p-value
MOEA/D	7.40	**0.00E+00**	6.45	**0.00E+00**	6.55	**0.00E+00**
NSGA-II	3.30	**8.13E-03**	4.30	**1.25E-03**	5.40	**0.00E+00**
MOEA/D-AAWN	5.60	**0.00E+00**	7.90	**0.00E+00**	6.75	**0.00E+00**
HREA	5.60	**0.00E+00**	3.60	**2.01E-02**	3.65	**9.95E-04**
BOEA	2.00	3.33E-01	2.15	6.51E-01	2.00	2.45E-01
TIE	6.50	**0.00E+00**	6.35	**0.00E+00**	6.80	**0.00E+00**
ACML-BCEA	4.35	**6.30E-05**	3.45	**3.32E-02**	3.75	**6.24E-04**
APHMA	**1.25**		**1.80**		**1.00**	

Moreover, Table III shows the results of the Friedman test results for all comparison algorithms. APHMA achieved the best rankings except for BOEA, whose p-value is less than 0.05. This is because BOEA utilizes a hierarchical clustering strategy in the decision space to remove redundant solutions to enhance convergence, which results in the indicator values having few gaps compared to APHMA. However, the indicator values of BOEA in Tables S-IV are worse than those of APHMA, exhibiting the advantages of APHMA.

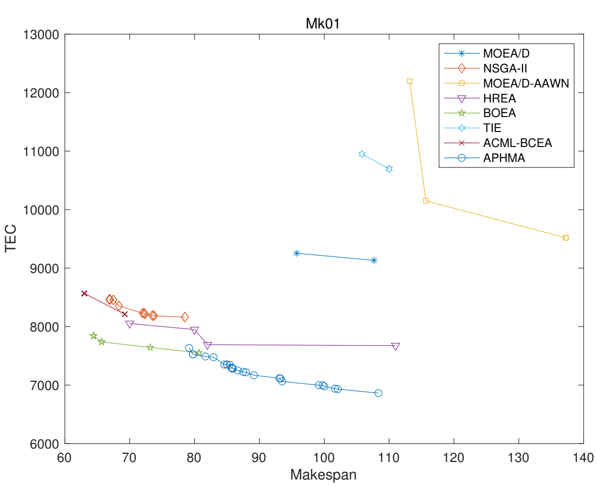
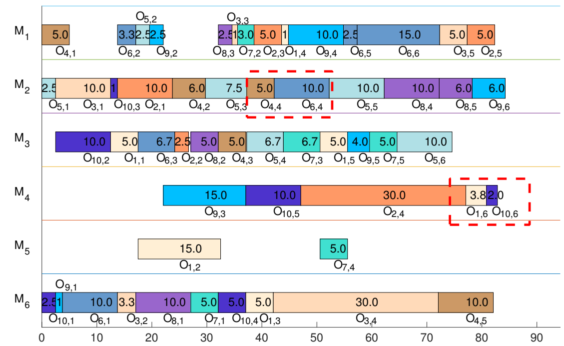
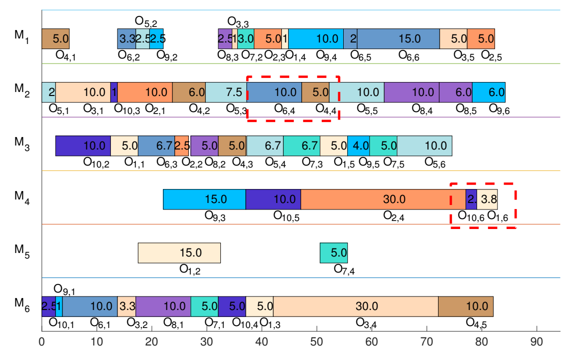


Fig. 9. Comparison of PF for all algorithms on MK01 instance.

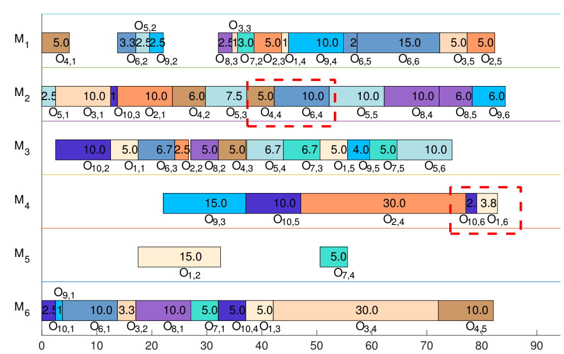
Fig. 9 shows the comparison of PF for all algorithms on MK01 instance, revealing that APHMA and BOEA achieve the most favorable outcomes. However, comparing APHMA and BOEA, the APHMA has better diversity performance, thus indicating the superiority or at least competitiveness of APHMA. Furthermore, Fig. 10 illustrates the Gantt charts of different optimal solutions on MK01 instance. It can be concluded that all three solutions obtain the same objective value ($C_{max} = 84.25; TEC = 7.59E + 03$), thus verifying the multimodal characteristics of the multi-objective FJSP-S. Based on the above analysis and discussion, APHMA can efficiently solve MMFJSP-S.



(a) Gantt chart of solution A



(b) Gantt chart of solution B



(c) Gantt chart of solution C

Fig. 10. Gantt charts of different optimal solutions on MK01 instance.

F. Discussion

The success of APHMA is due to the fact that the neighborhood structures are all designed based on problem characteristics that can guide the population in favorable directions for convergence. Moreover, the affinity propagation clustering with feature extraction extracts key decision variables for clustering, which ensures the diversity of the decision space while helping to find the elite solutions in each cluster. Finally, the hierarchical environmental selection strategy preserves global PFs and local PFs through a hierarchical approach, *i.e.,* it preserves global PSs and local PSs, balancing the optimization between the objective space and the decision space. Based on the above analysis and discussion, these components ultimately constitute the effectiveness of APHMA.

VI. Conclusions and Future Work

Multimodal multi-objective flexible job shop scheduling problem encompasses a vast search space, posing challenges in generating diverse PSs that also achieve favorable PFs. Therefore, this work proposes APHMA to solve this problem. To enhance the convergence in objective space, four problem-specific neighborhood structures are designed to guide population exploration toward promising directions. Furthermore, affinity propagation clustering is applied in the decision space, thus distinguishing between global and local PSs effectively. Finally, a hierarchical environmental selection strategy is devised to balance the optimization between the objective and decision spaces. Experimental results demonstrate the marked competitiveness of APHMA compared to other leading algorithms in addressing MMFJSP-S. To handle the multimodal characteristics of the scheduling problem, it is essential to consider the optimization of the decision space. Furthermore, it also becomes crucial to accommodate the selection in the decision space during the population update process.

However, there are still some limitations to our work. The first is that the multimodal properties of real engineering problems have not been explored, and the second is that the convergence in the objective space is slightly inadequate. Therefore, several future directions are under consideration: 1) Mining multimodal properties in real scheduling problems and combining problem characteristics to solve them; 2) Exploring the possibilities of some recent AI techniques for solving multimodal multi-objective scheduling problems; 3) Dealing with optimization in the decision spaces is also a more interesting direction.

References

[1] Q. Liu, C. Wang, X. Li, and L. Gao, "Mathematical modeling and a multiswarm collaborative optimization algorithm for fuzzy integrated process planning and scheduling problem," *Tsinghua Science and Technology*, vol. 29, no. 2, pp. 285–304, 2024.

[2] F. Zhao, C. Zhuang, L. Wang, and C. Dong, "An iterative greedy algorithm with q-learning mechanism for the multiobjective distributed no-idle permutation flowshop scheduling," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 5, pp. 3207–3219, 2024.

[3] J. Huang, L. Gao, and X. Li, "A hierarchical multi-action deep reinforcement learning method for dynamic distributed job-shop scheduling problem with job arrivals," *IEEE Transactions on Automation Science and Engineering*, pp. 1–13, 2024.

[4] C. Lu, J. Zhou, L. Gao, X. Li, and J. Wang, "Modeling and multi-objective optimization for energy-aware scheduling of distributed hybrid flow-shop," *Applied Soft Computing*, vol. 156, p. 111508, 2024.

[5] Z. Pan, D. Lei, and L. Wang, "A knowledge-based two-population optimization algorithm for distributed energy-efficient parallel machines scheduling," *IEEE Transactions on Cybernetics*, vol. 52, no. 6, pp. 5051–5063, 2022.

[6] Y. An, Z. Zhao, K. Gao, Y. Dong, X. Chen, and B. Zhou, "A self-adaptive co-evolutionary algorithm for multi-objective flexible job-shop rescheduling problem with multi-phase processing speed selection, condition-based preventive maintenance and dynamic repairman assignment," *Swarm and Evolutionary Computation*, vol. 89, p. 101643, 2024.

[7] Y. Li, Z. Yang, L. Wang, H. Tang, L. Sun, and S. Guo, "A hybrid imperialist competitive algorithm for energy-efficient flexible job shop scheduling problem with variable-size sublots," *Computers & Industrial Engineering*, vol. 172, p. 108641, 2022.

[8] F. Kudo, T. Yoshikawa, and T. Furuhashi, "A study on analysis of design variables in pareto solutions for conceptual design optimization problem of hybrid rocket engine," in *2011 IEEE Congress of Evolutionary Computation (CEC)*, 2011, pp. 2558–2562.

[9] O. Schütze, M. Vasile, and C. A. Coello Coello, "Computing the set of epsilon-efficient solutions in multiobjective space mission design," *Journal of Aerospace Computing Information & Communication*, vol. 8, no. 3, pp. 53–70, 2012.

[10] M. Sebag, N. Tarrisson, O. Teytaud, J. Lefevre, and S. Baillet, "A multi-objective multi-modal optimization approach for mining stable spatio-temporal patterns." in *IJCAI*, 2005, pp. 859–864.

[11] F. Ming, W. Gong, and Y. Jin, "Growing neural gas network-based surrogate-assisted pareto set learning for multimodal multi-objective optimization," *Swarm and Evolutionary Computation*, vol. 87, p. 101541, 2024.

[12] R. Li, W. Gong, L. Wang, C. Lu, and X. Zhuang, "Surprisingly popular-based adaptive memetic algorithm for energy-efficient distributed flexible job shop scheduling," *IEEE Transactions on Cybernetics*, vol. 53, no. 12, pp. 8013–8023, 2023.

[13] J. Liang, H. Lin, C. Yue, P. N. Suganthan, and Y. Wang, "Multiobjective differential evolution for higher-dimensional multimodal multiobjective optimization," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 6, pp. 1458–1475, 2024.

[14] Z. Wei, W. Gao, M. Gong, and G. G. Yen, "A bi-objective evolutionary algorithm for multimodal multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 1, pp. 168–177, 2024.

[15] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[16] J. Xie, X. Li, L. Gao, and L. Gui, "A hybrid genetic tabu search algorithm for distributed flexible job shop scheduling problems," *Journal of Manufacturing Systems*, vol. 71, pp. 82–94, 2023.

[17] M. Abdel-Basset, R. Mohamed, D. El-Shahat, and K. M. Sallam, "An efficient hybrid optimization method for fuzzy flexible job-shop scheduling problem: Steady-state performance and analysis," *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106249, 2023.

[18] Y. Li, W. Gu, M. Yuan, and Y. Tang, "Real-time data-driven dynamic scheduling for flexible job shop with insufficient transportation resources using hybrid deep q network," *Robotics and Computer-Integrated Manufacturing*, vol. 74, p. 102283, 2022.

[19] Q. Yan and H. Wang, "Double-layer q-learning-based joint decision-making of dual resource-constrained aircraft assembly scheduling and flexible preventive maintenance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 6, pp. 4938–4952, 2022.

[20] M. Flores-Gómez, V. Borodin, and S. Dauzère-Pérès, "Maximizing the service level on the makespan in the stochastic flexible job-shop scheduling problem," *Computers & Operations Research*, vol. 157, p. 106237, 2023.

[21] F. Yu, L. Yin, B. Zeng, C. Lu, and Z. Xiao, "A self-learning discrete artificial bee colony algorithm for energy-efficient distributed heterogeneous l-r fuzzy welding shop scheduling problem," *IEEE Transactions on Fuzzy Systems*, vol. 32, no. 6, pp. 3753–3764, 2024.

[22] J. Li, X. Li, L. Gao, and Q. Liu, "A flexible job shop scheduling problem considering on-site machining fixtures: A case study from customized manufacturing enterprise," *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2024.

[23] 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 Transactions on Evolutionary Computation*, vol. 27, no. 3, pp. 610–620, 2023.

[24] K. Tamssaouet and S. Dauzère-Pérès, "A general efficient neighborhood structure framework for the job-shop and flexible job-shop scheduling

problems," *European Journal of Operational Research*, vol. 311, no. 2, pp. 455–471, 2023.

[25] X. Li and L. Gao, "An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem," *International Journal of Production Economics*, vol. 174, pp. 93–110, 2016.

[26] R. Li, W. Gong, L. Wang, C. Lu, and C. Dong, "Co-evolution with deep reinforcement learning for energy-aware distributed heterogeneous flexible job shop scheduling," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 1, pp. 201–211, 2024.

[27] Y. Yao, Q. Wang, C. Wang, X. Li, L. Gao, and K. Xia, "Knowledge-based multi-objective evolutionary algorithm for energy-efficient flexible job shop scheduling with mobile robot transportation," *Advanced Engineering Informatics*, vol. 62, p. 102647, 2024.

[28] G. A. Kasapidis, D. C. Paraskevopoulos, I. Mourtos, and P. P. Repoussis, "A unified solution framework for flexible job shop scheduling problems with multiple resource constraints," *European Journal of Operational Research*, 2024.

[29] 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.

[30] K. Deb and S. Tiwari, "Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1062–1087, 2008.

[31] J. Liang, C. Yue, and B. Qu, "Multimodal multi-objective optimization: A preliminary study," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 2454–2461.

[32] Y. Liu, G. G. Yen, and D. Gong, "A multimodal multiobjective evolutionary algorithm using two-archive and recombination strategies," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 660–674, 2019.

[33] Y. Wang, Z. Yang, Y. Guo, J. Zhu, and X. Zhu, "A novel multi-objective competitive swarm optimization algorithm for multi-modal multi objective problems," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, 2019, pp. 271–278.

[34] Q. Lin, W. Lin, Z. Zhu, M. Gong, J. Li, and C. A. C. Coello, "Multimodal multiobjective evolutionary optimization with dual clustering in decision and objective spaces," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 130–144, 2021.

[35] K. Zhang, C. Shen, G. G. Yen, Z. Xu, and J. He, "Two-stage double niched evolution strategy for multimodal multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 4, pp. 754–768, 2021.

[36] W. Li, X. Yao, T. Zhang, R. Wang, and L. Wang, "Hierarchy ranking method for multimodal multiobjective optimization with local pareto fronts," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 1, pp. 98–110, 2023.

[37] Y. Sun and S. Zhang, "A decomposition and dynamic niching distance-based dual elite subpopulation evolutionary algorithm for multimodal multiobjective optimization," *Expert Systems with Applications*, vol. 231, p. 120738, 2023.

[38] R. Tanabe and H. Ishibuchi, "A framework to handle multimodal multiobjective optimization in decomposition-based evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 4, pp. 720–734, 2020.

[39] T. Hu, X. Wang, L. Tang, and Q. Zhang, "A clustering-assisted adaptive evolutionary algorithm based on decomposition for multimodal multiobjective optimization," *Swarm and Evolutionary Computation*, vol. 91, p. 101691, 2024.

[40] Y. Liu, H. Ishibuchi, G. G. Yen, Y. Nojima, and N. Masuyama, "Handling imbalance between convergence and diversity in the decision space in evolutionary multimodal multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 3, pp. 551–565, 2020.

[41] W. Li, T. Zhang, R. Wang, and H. Ishibuchi, "Weighted indicator-based evolutionary algorithm for multimodal multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 6, pp. 1064–1078, 2021.

[42] Y. Xie, J. Li, Y. Li, W. Zhu, and C. Dai, "Two-stage evolutionary algorithm with fuzzy preference indicator for multimodal multiobjective optimization," *Swarm and Evolutionary Computation*, vol. 85, p. 101480, 2024.

[43] F. Zhao, S. Di, and L. Wang, "A hyperheuristic with q-learning for the multiobjective energy-efficient distributed blocking flow shop scheduling problem," *IEEE Transactions on Cybernetics*, vol. 53, no. 5, pp. 3337–3350, 2023.

[44] H. Li, X. Li, and L. Gao, "An iterated greedy algorithm with acceleration of job allocation probability for distributed heterogeneous permutation flowshop scheduling problem," *Swarm and Evolutionary Computation*, vol. 88, p. 101580, 2024.

[45] H. Ge, N. Zhang, L. Sun, X. Wang, and Y. Hou, "A memetic evolution system with statistical variable classification for large-scale many-objective optimization," *Applied Soft Computing*, vol. 114, p. 108158, 2022.

[46] R. Devi Priya, R. Sivaraj, N. Anitha, and V. Devisurya, "Tri-staged feature selection in multi-class heterogeneous datasets using memetic algorithm and cuckoo search optimization," *Expert Systems with Applications*, vol. 209, p. 118286, 2022.

[47] Z. Xu, J. Xu, K. Zhang, X. Xu, J. He, and N. Wu, "Decision variable classification based multi-objective multifactorial memetic algorithm for multi-objective multi-task optimization problem," *Applied Soft Computing*, vol. 152, p. 111232, 2024.

[48] C. Luo, W. Gong, and C. Lu, "Knowledge-driven two-stage memetic algorithm for energy-efficient flexible job shop scheduling with machine breakdowns," *Expert Systems with Applications*, vol. 235, p. 121149, 2024.

[49] C. Zhang, P. Li, Z. Guan, and Y. Rao, "A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem," *Computers & Operations Research*, vol. 34, no. 11, pp. 3229–3242, 2007.

[50] J. Carruthers and A. Battersby, "Advances in critical path methods," *Journal of the Operational Research Society*, vol. 17, no. 4, pp. 359–380, 1966.

[51] J. Xie, X. Li, L. Gao, and L. Gui, "A hybrid algorithm with a new neighborhood structure for job shop scheduling problems," *Computers & Industrial Engineering*, vol. 169, p. 108205, 2022.

[52] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[53] P. Brandimarte, "Routing and scheduling in a flexible job shop by tabu search," *Annals of Operations research*, vol. 41, no. 3, pp. 157–183, 1993.

[54] S. Dauzère-Pérès and J. Paulli, "An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search," *Annals of Operations Research*, vol. 70, no. 0, pp. 281–306, 1997.

[55] A. Zhou, Q. Zhang, and Y. Jin, "Approximating the set of pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1167–1189, 2009.

[56] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 29–38, 2006.

[57] R. C. V. Nostrand, "Design of experiments using the taguchi approach: 16 steps to product and process improvement," *Technometrics*, vol. 44, no. 3, pp. 289–289, 2002.

[58] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

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

[60] Q. Zhao, Y. Guo, X. Yao, and D. Gong, "Decomposition-based multiobjective optimization algorithms with adaptively adjusting weight vectors and neighborhoods," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 5, pp. 1485–1497, 2023.

[61] J. Ding, S. Dauzère-Pérès, L. Shen, and Z. Lü, "A novel evolutionary algorithm for energy-efficient scheduling in flexible job shops," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 5, pp. 1470–1484, 2023.

Cong Luo received the M.S. degree in computer technology from China University of Geosciences, Wuhan, China, in 2024. He is currently pursuing the Ph.D. degree in mechanical engineering with Huazhong University of Science and Technology, Wuhan, China. His current research interests include evolutionary computation, machine learning, and their applications in production scheduling.

Xinyu Li (Member, IEEE) received the Ph.D. degree in industrial engineering from the Huazhong University of Science and Technology (HUST), China, in 2009.

He is currently a Professor with the Department of Industrial and Manufacturing Systems Engineering, State Key Laboratory of Intelligent Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, HUST. He has published more than 100 refereed articles. His research interests include intelligent algorithms, big data, and machine learning.

Prof. Li currently serves as an Associate Editor for IEEE Transactions on Automation Science and Engineering and Expert Systems with Applications.

Wenyin Gong (Member, IEEE) received the B.Eng., M.Eng., and Ph.D. degrees in computer science from China University of Geosciences, Wuhan, China, in 2004, 2007, and 2010, respectively.

He is currently a Professor with School of Computer Science, China University of Geosciences, Wuhan, China. He has published over 100 research papers in journals and international conferences. His research interests include evolutionary algorithms, evolutionary optimization, and their applications.

Prof. Gong currently serves as Associate Editor of Swarm and Evolutionary Computation, Expert Systems with Applications, Memetic Computing, etc.

Liang Gao (Senior Member, IEEE) received the B.Sc. degree in mechatronic engineering from Xidian University, Xi'an, China, in 1996, and the Ph.D. degree in mechatronic engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2002.

He is currently a Professor with the Department of Industrial and Manufacturing System Engineering, State Key Laboratory of Intelligent Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, HUST. He has published more than 400 refereed articles. His research interests include operations research and optimization, big data, and machine learning.

Prof. Gao currently serves as the Co-Editor-in-Chief for IET Collaborative Intelligent Manufacturing and an Associate Editor for Swarm and Evolutionary Computation and Journal of Industrial and Production Engineering.