# A Solution Space Transformation-Guided Co-Evolution for Energy-Saving Distributed Heterogeneous Flexible Job Shop Scheduling

**Tao Li**[1,2*], **Xingchen Li**[1], **Haoyue Ma**[1], **Zhi-Hui Zhan**[3*]
[1]The College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China
[2]The Engineering Lab of Intelligence Business and Internet of Things, Xinxiang 453007, China
[3]The College of Artificial Intelligence, Nankai University, Tianjin 300350, China
litao@htu.edu.cn, {2308283046, mahaoyue}@stu.htu.edu.cn, zhanapollo@163.com

## Abstract

Solving energy-saving distributed heterogeneous flexible job shop scheduling problem (ES-DHFJSP) aims to enhance industrial production efficiency while minimizing energy consumption. State-of-the-art co-evolutionary algorithms have emerged as effective approaches for addressing ES-DHFJSP. However, existing methodologies demonstrate compromised convergence rates and excessive computational overhead when confronted with vast search spaces. In this work, we propose a novel solution space transformation-guided co-evolution algorithm (SSTCE) to overcome this limitation. In SSTCE, we first establish an inter-job similarity metric and incorporate constrained hierarchical clustering with optimal leaf ordering (CHC-OLO) to generate clustered job sets, which are subsequently utilized for population initialization that achieves a favorable balance between convergence and diversity. To enhance search capability in expansive solution spaces, we devise a dynamic solution space transformation mechanism that effectively reduces inefficient searches within the algorithm. Furthermore, we develop tailored local search strategies leveraging domain-specific knowledge of DHFJSP properties. Extensive experimental evaluations across 20 benchmark instances demonstrate that SSTCE significantly outperforms existing evolutionary algorithms in solving ES-DHFJSP.

**Code** — https://github.com/real-laugh/SSTCE

## Introduction

The ES-DHFJSP represents one of the critical challenges in real-world distributed flexible manufacturing systems (Xuan, Li, and Li 2023). Implementing optimal scheduling schemes significantly enhances production efficiency (Smit et al. 2025). Furthermore, economic costs and energy consumption are both minimized by this approach. This problem encompasses three interdependent sub-problems: (1) ascertaining optimal job-to-factory deployment strategies; (2) Opting for suitable machines to each specific operation; and (3) formulating the operation processing sequence within each factory. This multidimensional decision-making process establishes ES-DHFJSP as an complex combinatorial optimization problem, classified as NP-hard problem (Luo et al. 2023).

With the advancements in metaheuristic algorithms including genetic algorithm (GA), particle swarm optimization (PSO) and the differential evolution (DE), the evolutionary algorithms have established themselves as the mainstream methodology for addressing complex DHFJSP with NP-hard characteristics. The fundamental framework of evolutionary algorithms comprises three core components: initialization, evolutionary process, and local search.

For ES-DHFJSP, the hybrid greedy-random initialization approach is the predominant methodology in current research. In machine selection, the greedy strategy is manifested through consistently assigning operations to either the shortest-processing-time machine or the machine with the smallest current workload (Zhao and Li 2024). Similarly, in factory allocation, the greedy strategy involves allocating jobs to either the factory with the shortest processing time or the factory exhibiting the smallest current workload (Cao and Yuan 2024). Furthermore, within the domain of local search, research efforts primarily focus on utilizing advanced techniques to select superior local search strategies, such as q-learning (Wu et al. 2023) and reinforcement learning (Yan, Wang, and Yang 2024).

**Motivations.** Although evolutionary algorithms (EAs) have attained notable success in optimizing ES-DHFJSP, existing EA-based approaches continue to exhibit limitations. Existing greedy strategies fail to account for their impact on other jobs, which limits their effectiveness in enhancing the quality of the initial population. Consequently, a clustering approach is introduced to comprehensively considers the overall allocation of all jobs, thereby achieving more stable and effective improvements. Regarding the local search component, there remains potential for further development of the ES-DHFJSP features. Thus, two properties are developed, while two corresponding local search strategies are designed to enhance the efficiency of local search. Furthermore, existing research rarely focuses on improving the evolutionary process itself. Evolutionary process constitutes the core of algorithm, improving this process represents a pivotal factor in boosting overall performance. Specifically for the ES-DHFJSP, an appropriate solution space partitioning method and a solution space transformation mechanism are

*Corresponding author.
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

developed. Upon thorough exploitation of a subspace, this transformation mechanism activates to mitigate redundant exploration.

The main contributions are summarized as follows:

1. A hierarchical clustering-based population initialization method is designed, which can generate a uniform and high-quality initial population, improving the convergence speed of the algorithm without losing diversity.
2. A dynamic solution space transformation mechanism is proposed to achieve exhaustive search in solution space and circumvent premature convergence in local optima through adaptive landscape modulation.
3. A prior knowledge-driven local search is developed. Two problem-specific characteristics are fully exploited to develop dedicated local search operators. Additionally, five supplementary local search operators are integrated, leveraging elite individuals as focal points and effectively exploiting potential optimal solutions through targeted exploitation.

## Related Work

**Distributed Flexible Job Shop Scheduling.** Distributed flexible job shop scheduling problems have remained a persistent research focus in industrial engineering. The primary challenge is the distributed production, which significantly increases the problem's complexity. Naderi and Azab (Naderi and Azab 2014) expanded the production environment from single-facility to distributed architectures, and created two different mixed integer linear programming models in the form of sequence and position based variables. Meng et al. (Meng et al. 2020) first introduced a constraint programming approach to develop the problem model. This model addressed the limitations of the mixed-integer linear model, which struggled with large-scale DFJSP instances. Subsequently, researchers are pioneering heterogeneous factory configurations and continuously improving the DHFJSP model, such as considerations of low-carbon objectives (Chen et al. 2024) and energy-aware (Wang, Han, and Wang 2024).

**Evolutionary Algorithm For DHFJSP.** With the advancements in evolutionary algorithms, the evolutionary algorithms have established themselves as the mainstream methodology for addressing DHFJSP. Regarding initial evolutionary population generation, existing researches not only utilized greedy machine selection and greedy factory assignment, but also considered of factory load balancing (Yang et al. 2025). This approach aims to ensure a relatively uniform distribution of jobs across factories in the initial population. Pan et al. (Pan et al. 2025) integrated evolutionary learning and statistical learning, coupling them with Pareto ranking and decomposition methods respectively, which significantly enhanced algorithmic efficiency for complex problems. Zhang et al. (Zhang et al. 2024) developed seven neighborhood structures based on critical path to enhance local search capability. Li et al. (Li et al. 2023b) utilized the surprisingly popular algorithm to assess operator effectiveness, subsequently selecting efficient operators. For the DHFJSP, researchers typically design local search operators focusing on critical factories, critical machines, and critical operations. However, existing operators exhibit noticeable randomness (Cao et al. 2023). This results in insufficient exploitation capability during the local search.

## Problem Description

ES-DHFJSP is characterized by heterogeneous factories, where each factory operates a distinct number of machines. Furthermore, individual operations may involve diverse sets of available machines. Normally, the ES-DHFJSP encompasses processing $N$ jobs across $F$ heterogeneous factories. $i$-th job consists of $n_i$ operations. $O_{i,j}$ represents the $j$-th operation of the $i$-th job. Each factory $f$ is equipped with $m_f$ non-identical machines. The problem is formulated under the following assumptions:

- Initially, all factories and machines are accessible, and each job can be processed in any factory.
- Factory assignment of each job cannot be modified midway.
- A machine cannot process two operations simultaneously and be interrupted during the machining process.
- An operation can only be processed on one machine simultaneously.
- For each job, there exists a precedence relationship among all consecutive processes.
- Machine immediately stop running when all operations on it are completed.

Consequently, ES-DHFJSP can be formulated as a multi-objective optimization problem that simultaneously minimizes two metrics: makespan and total energy consumption (TEC). This can be expressed as follows:

$$\min \mathbf{f(x)} = (f_1(x), f_2(x)) \tag{1}$$

$$f_1(x) = C_{max} = \max FT_{i,j,f,m} \tag{2}$$

$$\begin{aligned} f_2(x) = &\sum_{i=1}^{N}\sum_{j=1}^{n_i}\sum_{f=1}^{F}\sum_{m=1}^{m_f} T_{i,j,f,m} \times W_P \times x_{i,j,f,m} + \\ &\sum_{f=1}^{F}\sum_{m=1}^{m_f}\sum_{p=1}^{P_{f,m}} (SP_{f,m,p} - FP_{f,m,p-1}) * W_I \end{aligned} \tag{3}$$

where $C_{max}$ represents the makespan. $FT_{i,j,f,m}$ denotes the finish time of $O_{i,j}$ on machine $m$ in factory $f$. $T_{i,j,f,m}$ indicates the processing time of $O_{i,j}$ on the $m$-th machine in factory $f$. $x_{i,j,f,m}$ is a bool decision variable. If $O_{i,j}$ is processed on the $m$-th machine in factory $f$, the value is set to 1; otherwise, it is set to 0. $SP_{f,m,p}$ and $FP_{f,m,p}$ stand for the start and finish processing time of the $p$-th position on the $m$-th machine in factory $f$, respectively. $P_{f,m}$ is the number of positions on the $m$-th machine in factory $f$. $W_P$ and $W_I$ severally represent the process power and the idle power of all machines. The detailed notation descriptions and constraint conditions are presented in *Appendix A*.

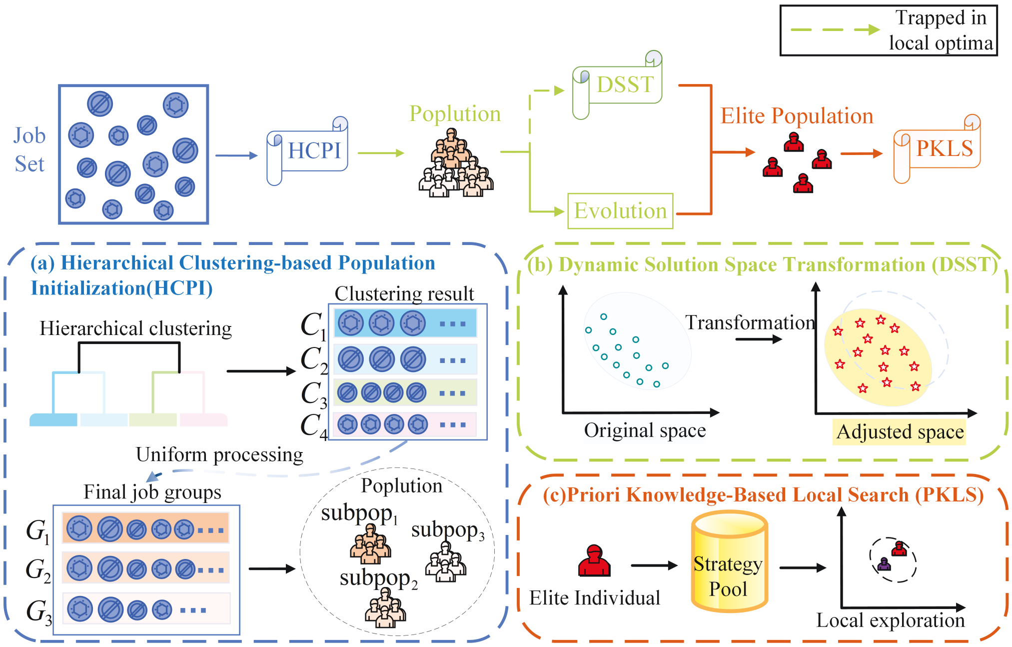


Figure 1: Framework of the proposed SSTCE.

## The Proposed SSTCE Algorithm

### Overall Framework

Figure 1 shows the framework of SSTCE. The SSTCE algorithm encompasses the following critical phases. Initially, a hierarchical clustering-based population initialization method is introduced to generate a uniform and high-quality initial population. Subsequently, the algorithm executes global search across the population. Meanwhile, the algorithm performs real-time monitoring, evaluating whether the solution space has been exploitation thoroughly. Upon detection of stagnation, a dynamic solution space transformation mechanism is automatically triggered to jump out of local optima. The elite individuals are transferred into the elite archive after concluding the global exploration phase. Consecutively, a prior knowledge-driven local search is employed to further exploit potential optimal solutions around the elite archive. Finally, the high-quality individuals generated through local search phase are fed back to the population.

### Encoding and Decoding Schema

This paper employs the three-vector encoding approach (Li et al. 2024b). Three distinct components are comprised: 1) operation sequence (OS) establishes the processing order of all operations, with its length being the total number of operations $SH = \sum_{i=1}^{N} n_i$; 2) machine selection (MS) dictates which machine processes each operation, having identical length $SH$; and 3) factory assignment (FA) determines which factory each job is processed in, with length defined as $N$.

### Hierarchical Clustering-based Population Initialization Method

A well-distributed and high-quality initial population can effectively mitigate unproductive exploration during the early period while accelerating algorithmic convergence. To address this requirement, we present a hierarchical clustering-based population initialization method.

Prior to the initialization phase, a preliminary grouping process is executed on the jobs. Initially, the average processing times per job across all factories are calculated. Specifically, the $AvgT_{i,f}$ denotes the average processing time of $i$-th job in factory $f$ and is calculated by Equation (4), where $M_{i,j,f}$ and $M_{i,j,f}^{n}$ represents the processing machine candidates of $O_{i,j}$ and the number of candidates, respectively.

$$AvgT_{i,f} = \sum_{j=1}^{n_i} \frac{\sum_{m \in M_{i,j,f}} T_{i,j,f,m}}{M_{i,j,f}^{n}} \tag{4}$$

Subsequently, factories are sorted in ascending order of $AvgT_{i,f}$, obtaining the factory ranking sequence denoted as $FR_i = \{f_1, f_2, ..., f_F\}$, where $f_1$ represents the factory index with the top-ranked.

Based the factory ranking sequence, factories are differentiated into three levels (high/medium/low) when the value of $F$ exceeds a certain threshold $\zeta$; otherwise, they are classified into two levels (high/low). Specifically, for the three-level classification, the top $\lfloor F/3 \rfloor$ ranked factories are designated as high-level, the bottom $\lfloor F/3 \rfloor$ as low-level, with the remaining factories categorized as medium-level. The two-level classification employs $\lfloor F/2 \rfloor$ as the division criterion. These level-based hierarchical attributes are encoded as feature vectors for clustering.

Due to the hierarchical relationship and the inherently ordinal nature of the feature vectors, the CHC-OLO (Bar-Joseph, Gifford, and Jaakkola 2001) is employed to cluster jobs. Jobs with similar factory rankings are clustered into the same cluster. Moreover, the CHC-OLO algorithm has an inherent advantage in sequential optimization of clustering results: it leverages its sequential allocation mechanism to directly assign samples to clusters while achieving balanced cluster sizes. Finally, $Num_g$ job groups are formed based on these optimized clusters, with jobs from each cluster being distributed evenly across all groups, where $Num_g$ is equal to the number of clusters.

Upon the completion of the grouping process, the initialization is implemented. First, the factory assignment initialization is as follows: For any group, the jobs within it are assigned to the factory with the minimum $AvgT_{i,f}$, while the remaining jobs are distributed stochastically. Each group generates $\lceil ps/Num_g \rceil$ individuals, and the last group accommodates the remaining individuals, where $ps$ represents the size of population.

Operation sequences for all individuals are generated randomly. Subsequently, with 80% probability, all operations are randomly assigned to any available machine. Meanwhile, with 20% probability, each operation is allocated to the machine with the shortest processing time.

## Dynamic Solution Space Transformation Mechanism

Traditional evolutionary algorithms suffer from substantial ineffective search and tend to get trapped in local optima. These limitations are derived from the attribute of distributed production (Xie et al. 2023). It motivates our proposed the dynamic solution space transformation mechanism.

In our algorithm, each factory assignment is defined as a solution space. After the initialization method, the initial solution space is established. Priority Operation Crossover (POX) and Universal Crossover (UX) are employed to perform crossover with a probability $P_c$ for the operation sequence and the machine selection, respectively. Furthermore, two distinct mutation strategies are implemented with a probability $P_m$ to enhance the diversity of the population: 1) implement positional interchange of two randomly selected operations within the scheduling sequence; 2) Redirect a stochastically chosen operation to a different available machine.

The elite archive from the previous iteration is denoted as $\mathcal{A}$, while $\mathcal{A}^{'}$ represents the elite archive upon completion of the evolutionary process and local search. Subsequently, the euclidean distance $D$ between the centroid of $\mathcal{A}$ and $\mathcal{A}^{'}$ is calculated. $D < \delta$ represents that the current iteration is ineffective, where $\delta$ denotes an error threshold. When $\theta$ consecutive iterations are identified as ineffective, the dynamic solution space transformation mechanism is activated.

The crossover operation UX is employed as the transformation strategy to update the factory assignment vectors of population. And a mutation strategy is executed, which modifies the production factory assignment for a randomly sampled job. This methodology orchestrates controlled solution space transformation through adaptive mechanism, enabling algorithm escape from local optima. Furthermore, the complexity of the search space is significantly reduced, which prevents the algorithm from performing excessive invalid searches and improves resource utilization.

## Prior Knowledge-driven Local Search

The collaborative integration of local search with global exploration demonstrates superior optimization capability, leading to its pervasive adoption in contemporary evolutionary algorithm architectures (Zhao et al. 2025a; Hou et al. 2025; Li et al. 2024a). In this work, we thoroughly deconstruct the characteristics of ES-DHFJSP, subsequently developing specialized local search operators tailored for each sub-problem domain. These strategies enable targeted exploitation of undiscovered optima surrounding the elite individuals within $\mathcal{A}$.

In the ES-DHFJSP, the critical factory is defined as the factory achieving the maximum makespan; the critical machine is the machine with the maximum completion time within its respective factory; and the critical operation represents the operation with its start time equal to the latest possible start time. These key components serve as pivotal factors influencing scheduling effectiveness. As these two factors collectively exert the most substantial influence on scheduling performance, we have developed specialized local search operators specifically tailored to critical factories, critical machines, and critical operations.

*Property 1:* The makespan can be reduced by reassigning the $i$-th job from the critical factory $F_c$ to another factory $f$, satisfying the following condition for all machines $m$:

$$MCT_{f,m} + \sum_{j=1}^{n_j} \max_{m \in m_f} T_{i,j,f,m} < FCT_{F_c} \tag{5}$$

where $MCT_{f,m}$ indicates the finish time of $m$ in factory $f$, $FCT_{F_c}$ represents the completion time of factory $F_c$.

*Remark:* The assignment of job $i$ from $F_c$ to factory $f$ will reduce $FCT_{F_c}$. Furthermore, for any machine $m$ in factory $f$, two critical values are defined as follows: 1) $MCT_{f,m}$ remains unchanged if no operations are assigned to it; 2) $\varpi = MCT_{f,m} + \sum_{j=1}^{n_j} \max_{m \in m_f} T_{i,j,f,m}$, when all assignable operations are allocated to it. This implies that the completion time of $f$ will lie within $[MCT_{f,m}, \varpi]$. Consequently, when the condition $\varpi < FCT_{F_c}$ is satisfied, allocating the job $i$ to factory $f$ guarantees a reduction in the makespan.

*Property 2:* For any factory, if there exists another machine $m$ such that $\max\{MCT_{f,m}, FT_{i,j-1,f,m}\} + T_{i,j,f,m} < MCT_{f,M_c}$ holds for the final operation $O_{i,j}$ on the critical machine $M_c$, then the completion time of $f$ can be reduced by reallocating $O_{i,j}$ to machine $m$.

*Remark:* It is established that $M_c$ exhibits the maximum completion time. The insertion of operation $O_{i,j}$ at the end of $m$ presents two distinct scenarios: 1) $MCT_{f,m} < FT_{i,j-1,f,m}$. $O_{i,j}$ must await the completion of $O_{i,j-1}$ before processing commences; 2) $MCT_{f,m} \geq FT_{i,j-1,f,m}$. $O_{i,j}$ can commence immediately after $MCT_{f,m}$. Therefore, the completion time of operation $O_{i,j}$ is constrained to either $FT_{i,j-1,f,m} + T_{i,j,f,m}$ or $MCT_{f,m} + T_{i,j,f,m}$, which is equal to the maximum completion time $maxT_m$ of machine $m$. Moreover, $MCT_{f,M_c}$ has decreased. Consequently, whenever $maxT_m < MCT_{f,M_c}$, $\max_{m \in m_f} MCT_{f,m} < MCT_{f,M_c}$ and the completion time of $f$ will decrease.

Based on the two aforementioned problem properties, we have developed two specialized local search operators:

- *LS1:* Select the job $i$ from Factory $F_c$ that meets two criteria: (a) It has operations processed on machine $M_c$; Among jobs satisfying (a), choose the one with the longest total processing time. If there exists a factory $f$ satisfying Property 1, job $i$ is assigned to $f$.
- *LS2:* Initially, stochastically choose a factory $f$. If there exists a machine $m$ that satisfies Property 2, the last operation of $M_c$ is reassigned as the terminal task on machine $m$.

Meanwhile, five specialized search operators are incorporated to enhance the exploration capability of the local search procedure.

- *LS3:* Under the $LS_1$ operator, job $i$ is directly assigned to the factory exhibiting the minimum completion time, bypassing the selection criteria outlined in Property 1.
- *LS4:* The random factory is designated as the new allocation target for job randomly sampled from critical factory.
- *LS5:* Two critical-path operations are selected in a randomized fashion for position interchange.
- *LS6:* Stochastically select an operation within the critical path and interchange its position with another randomly chosen operation.
- *LS7:* A critical operation is randomly chosen, and its machine selection is altered to the one with minimum processing time.

Furthermore, an adaptive mechanism is employed to regulate the allocation of computational resources. This mechanism restricts local search at the beginning. This focuses the algorithm on broader global search. As the search progresses, the number of local searches is incrementally increased until it reaches the maximum value $\varepsilon$.

## Experiments

To validate the advantages of the proposed SSTCE, we conduct extensive experiments structured into three phases. First, a sensitivity analysis is performed to examine the parameter configurations. Subsequently, ablation studies are executed to verify the effectiveness of the proposed components. Finally, comprehensive comparative evaluations against four state-of-the-art algorithms are conducted. All algorithms run 20 times independently, with identical termination criteria $MaxNFEs = 200 \times \sum_{i=1}^{N} n_i$. Furthermore, all algorithms run on an AMD Ryzen 9 7940HX with Radeon Graphics (2.40GHz) and 16GB of RAM. Prior to these analyses, we present detailed specifications of the benchmark instances and performance metrics.

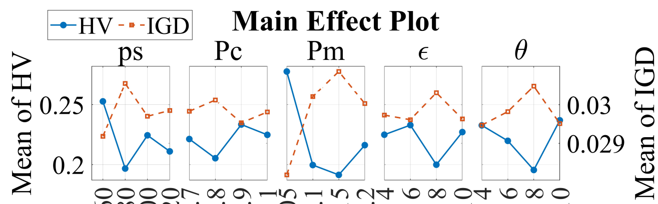


Figure 2: The parameter sensitivity analysis on two metrics.

### Experimental Setup

The 20 benchmark instances are utilized (Li et al. 2024b). Job counts span {10, 20, 30, 40, 50, 100, 150, 200} and factory numbers range from 2 to 7. Each factory is equipped with 5 machines. Detailed descriptions of the instances are provided in *Appendix B*. Two distinct energy consumption modes are adopted: processing energy consumption (4KWh) and idle state consumption (1KWh). Given the factory range, the $\zeta$ is initialized to 4. Moreover, $\delta = \sqrt{0.05^2 + 0.05^2}$. $ls < \varepsilon * \min\left(\frac{NFEs}{ps*20}, 1\right)$, where $ls$ represents the current number of local searches and $NFEs$ denotes the current number of function evaluations.

Two quantitative metrics are considered for performance evaluation: Inverted Generational Distance (IGD) and Hypervolume (HV). A lower IGD value indicates better overall algorithm performance, while a higher HV value signifies superior algorithm performance.

### Parameter Sensitivity Analysis

In this work, the Taguchi method (Nostrand 2002) is employed for parameter optimization, focusing on 5 critical parameters and assigning four distinct levels to each parameter: population size $ps = \{60, 80, 100, 120\}$, crossover probability $Pc = \{0.7, 0.8, 0.9, 1.0\}$, mutation probability $Pm = \{0.05, 0.1, 0.15, 0.2\}$, maximum local search times $\varepsilon = \{4, 6, 8, 10\}$, and failure termination threshold $\theta = \{4, 6, 8, 10\}$. The parameter configuration experiments utilized an $L_{16}(4^5)$ orthogonal table. The metrics of each parameter at all levels are shown in Figure 2. The result substantiates that the optimal parameter combination is $ps = 80$, $Pc = 0.8$, $Pm = 0.15$, $\varepsilon = 8$, $\theta = 8$.

### Ablation Experiment

To evaluate the effectiveness of individual model components, we perform systematic ablation analyses. We systematically construct three variant algorithms. In SSTCE1, the hierarchical clustering-based population initialization method is substituted with stochastic initialization. In SSTCE2, the dynamic solution space transformation mechanism is decoupled from the evolutionary process. Furthermore, in SSTCE3, the prior knowledge-driven local search is removed.

Table 1 presents the Friedman rank-and-sum test results across two evaluation metrics. Meanwhile, employing a statistical significance threshold of $\alpha = 0.05$, the experimental findings systematically demonstrate three principal observations. Firstly, the hierarchical clustering-based initialization

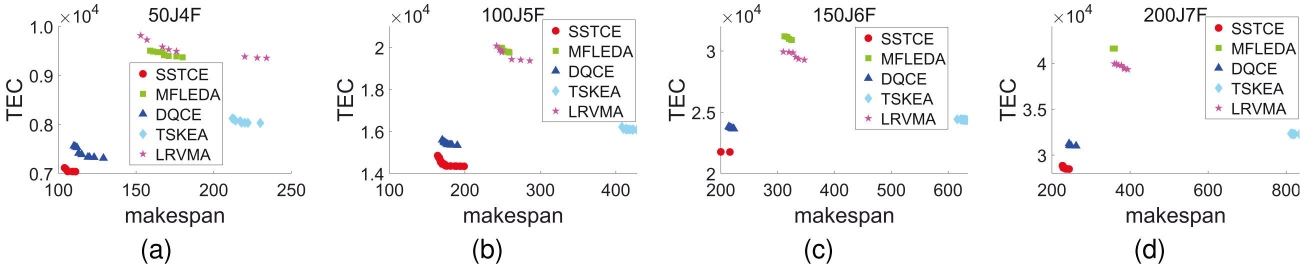


Figure 3: The final Pareto fronts obtained by five algorithms for four instances with varying scales.

| MOEAs | HV | | IGD | |
|---|---|---|---|---|
| | rank | $p$-value | rank | $p$-value |
| SSTCE | **1.25** | 7.607E-11 | **1.40** | 2.467E-10 |
| SSTCE1 | 3.9 | | 3.75 | |
| SSTCE2 | 1.8 | | 1.70 | |
| SSTCE3 | 3.05 | | 3.15 | |

Table 1: Friedman test rankings across all algorithms.

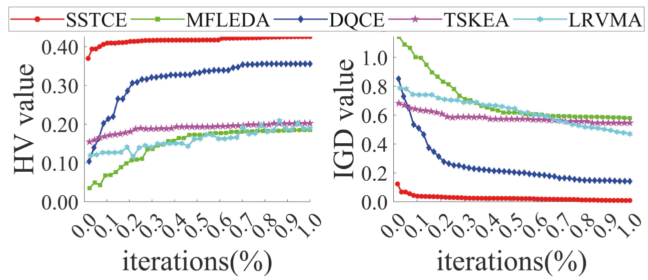


Figure 4: The convergence comparison in instance 100J4F across both metrics HV and IGD.

method demonstrates significant improvements in generating high-quality and well-distributed initial population, effectively enhancing both algorithmic convergence and diversity preservation. Secondly, the implemented dynamic solution space transformation mechanism successfully mitigates redundant search operations, enabling broader and more efficient exploration within constrained computational budgets. Finally, the prior knowledge-driven local search efficaciously exploits potential optima in the vicinity of elite solutions, substantially boosting the algorithm's exploitation capacity. Furthermore, ablation experiments on seven specialized local search operators are provided in *Appendix C*.

### Comparison and Analysis

The SSTCE undergoes rigorous capability profiling through comparisons with TSKEA (Li et al. 2022), LRVMA(Li et al. 2023a), DQCE(Li et al. 2024b) and MFLEDA(Zhao et al. 2025b), employing t-tests to assess statistically significant differences in performance. All critical parameters for the four comparative algorithms maintain strict adherence to their respective configurations as specified in source publications. The specific details are given in *Appendix D*.

Table 2 and Table 3 respectively present the comparative mean performance of five algorithms on two evaluation metrics, with accompanying the standard deviation. Furthermore, t-test is employed to validate statistical significance. The results are shown in the table's terminal row. ”-” denotes statistically significant underperformance of the comparative algorithm relative to the SSTCE, ”=” indicates statistical equivalence between methodologies, while ”+” signifies cases where the comparative algorithm demonstrates statistically significant superiority. To further demonstrate algorithmic comparisons, we select four instances with varying scales. Figure 3 presents the final Pareto fronts obtained by all algorithms for these instances. Meanwhile, the convergence of the five algorithms on instance 100J4F are illustrated in Figure 4, demonstrating their respective optimization trajectories and performance benchmarks. The comparison of the running times for all algorithms is provided in *Appendix E*. Experimental results demonstrate that SSTCE significantly outperforms the latest evolutionary algorithms in terms of two metrics when solving the ES-DHFJSP. Furthermore, it validates the effectiveness of the novel methods proposed herein this paper. These methods play crucial roles in enhancing the performance of the algorithm, enabling it to outperform existing algorithms.

## Conclusion and Future Work

In this paper, we propose a solution space transformation-guided co-evolution algorithm to address ES-DHFJSP. First, a hierarchical clustering-based population initialization method is proposed, which introduces the constrained hierarchical clustering with optimal leaf ordering to categorize jobs, ensuring uniform dispersion across clusters for final job groups formation. This population initialization method not only elevates initial population quality but also maintains population diversity. Second, we propose a dynamic solution space transformation mechanism that implements transformation when population meet predefined stagnation detection criteria. This mechanism effectively prunes the dimensionality of ES-DHFJSP and redundant search. Third, exploiting the distinctive problem characteristics of ES-DHFJSP, we devise seven specific local search operators that accelerate algorithmic convergence with efficacy. Finally, the comprehensive experimental results validate the superior efficacy of SSTCE in solving ES-DHFJSP.

| Ins | LRVMA | | TSKEA | | DQCE | | MFLEDA | | SSTCE | |
|---|---|---|---|---|---|---|---|---|---|---|
| | mean | std | mean | std | mean | std | mean | std | mean | std |
| 10J2F | 0.1797= | 0.016 | 0.1953= | 0.013 | **0.2106+** | 0.021 | 0.0633- | 0.016 | 0.1868 | 0.013 |
| 20J2F | 0.0970- | 0.019 | 0.1913- | 0.006 | 0.1900= | 0.013 | 0.0457- | 0.011 | **0.1969** | 0.010 |
| 20J3F | 0.0867- | 0.022 | 0.1205- | 0.006 | 0.1980- | 0.022 | 0.0513- | 0.015 | **0.2217** | 0.009 |
| 30J2F | 0.0517- | 0.012 | 0.1386- | 0.003 | 0.1638= | 0.009 | 0.0301- | 0.005 | **0.1691** | 0.007 |
| 30J3F | 0.0643- | 0.006 | 0.1166- | 0.004 | 0.1885- | 0.009 | 0.0466- | 0.011 | **0.2022** | 0.012 |
| 40J2F | 0.0526- | 0.017 | 0.1567- | 0.004 | 0.1746- | 0.013 | 0.0395- | 0.006 | **0.1894** | 0.004 |
| 40J3F | 0.0545- | 0.018 | 0.0924- | 0.004 | 0.2020- | 0.014 | 0.0590- | 0.006 | **0.2280** | 0.007 |
| 40J4F | 0.0616- | 0.025 | 0.0636- | 0.003 | 0.2302- | 0.017 | 0.0708- | 0.011 | **0.2425** | 0.012 |
| 50J3F | 0.0836- | 0.022 | 0.1649- | 0.003 | **0.2480=** | 0.011 | 0.0894- | 0.009 | 0.2476 | 0.007 |
| 50J4F | 0.0520- | 0.006 | 0.0522- | 0.004 | 0.2095- | 0.015 | 0.0510- | 0.010 | **0.2436** | 0.010 |
| 50J5F | 0.0941- | 0.044 | 0.0678- | 0.005 | 0.2695- | 0.013 | 0.1065- | 0.014 | **0.2798** | 0.010 |
| 100J4F | 0.0856- | 0.024 | 0.0802- | 0.004 | 0.2572- | 0.007 | 0.0904- | 0.008 | **0.3065** | 0.005 |
| 100J5F | 0.0988- | 0.046 | 0.0598- | 0.002 | 0.2984- | 0.011 | 0.1211- | 0.008 | **0.3336** | 0.006 |
| 100J6F | 0.1042- | 0.028 | 0.0531- | 0.003 | 0.2932- | 0.011 | 0.1058- | 0.011 | **0.3391** | 0.009 |
| 100J7F | 0.1092- | 0.052 | 0.0677- | 0.001 | 0.3458- | 0.009 | 0.1501- | 0.012 | **0.3982** | 0.008 |
| 150J5F | 0.0864- | 0.023 | 0.0513- | 0.002 | 0.2654- | 0.009 | 0.0802- | 0.005 | **0.3095** | 0.008 |
| 150J6F | 0.1103- | 0.047 | 0.0597- | 0.002 | 0.3094- | 0.005 | 0.1159- | 0.007 | **0.3646** | 0.007 |
| 150J7F | 0.1040- | 0.033 | 0.0495- | 0.001 | 0.2938- | 0.010 | 0.0920- | 0.007 | **0.3525** | 0.005 |
| 200J6F | 0.0782- | 0.012 | 0.0583- | 0.002 | 0.2748- | 0.008 | 0.0717- | 0.006 | **0.3400** | 0.007 |
| 200J7F | 0.1130- | 0.010 | 0.0464- | 0.001 | 0.2974- | 0.008 | 0.0809- | 0.110 | **0.3593** | 0.004 |
| -/=/+ | 19/1/0 | | 19/1/0 | | 16/3/1 | | 20/0/0 | | | |

Table 2: Results of HV metric for comparing five algorithms.

| Ins | LRVMA | | TSKEA | | DQCE | | MFLEDA | | SSTCE | |
|---|---|---|---|---|---|---|---|---|---|---|
| | mean | std | mean | std | mean | std | mean | std | mean | std |
| 10J2F | 0.1811= | 0.055 | 0.1599= | 0.028 | **0.1222+** | 0.040 | 0.8234- | 0.110 | 0.1667 | 0.034 |
| 20J2F | 0.4556- | 0.065 | 0.0831- | 0.020 | 0.0834- | 0.042 | 0.9274- | 0.093 | **0.0593** | 0.034 |
| 20J3F | 0.6834- | 0.114 | 0.4741- | 0.029 | 0.1727- | 0.035 | 0.9180- | 0.108 | **0.0793** | 0.029 |
| 30J2F | 0.8413- | 0.069 | 0.2304- | 0.010 | 0.1099- | 0.041 | 1.0880- | 0.047 | **0.0769** | 0.027 |
| 30J3F | 0.8341- | 0.047 | 0.4860- | 0.024 | 0.1305- | 0.036 | 0.9903- | 0.102 | **0.0761** | 0.047 |
| 40J2F | 0.8248- | 0.075 | 0.2036- | 0.011 | 0.1165- | 0.048 | 0.9851- | 0.050 | **0.0361** | 0.013 |
| 40J3F | 0.8270- | 0.070 | 0.6354- | 0.020 | 0.1253- | 0.048 | 0.8629- | 0.044 | **0.0356** | 0.017 |
| 40J4F | 0.7701- | 0.069 | 0.8309- | 0.016 | 0.1079- | 0.047 | 0.8494- | 0.071 | **0.0632** | 0.035 |
| 50J3F | 0.7278- | 0.045 | 0.3518- | 0.013 | 0.0613= | 0.033 | 0.7679- | 0.052 | **0.0445** | 0.018 |
| 50J4F | 0.9978- | 0.040 | 0.9090- | 0.019 | 0.2040- | 0.062 | 1.0047- | 0.074 | **0.0582** | 0.031 |
| 50J5F | 0.6456- | 0.051 | 0.8401- | 0.023 | 0.1016- | 0.030 | 0.7154- | 0.059 | **0.0560** | 0.023 |
| 100J4F | 0.7565- | 0.022 | 0.8120- | 0.016 | 0.1633- | 0.018 | 0.8021- | 0.044 | **0.0223** | 0.009 |
| 100J5F | 0.6703- | 0.035 | 0.9012- | 0.008 | 0.1326- | 0.028 | 0.6952- | 0.032 | **0.0370** | 0.009 |
| 100J6F | 0.7619- | 0.042 | 0.9460- | 0.012 | 0.1762- | 0.029 | 0.8036- | 0.046 | **0.0326** | 0.017 |
| 100J7F | 0.6290- | 0.046 | 0.9195- | 0.005 | 0.1594- | 0.026 | 0.7079- | 0.037 | **0.0241** | 0.014 |
| 150J5F | 0.7951- | 0.024 | 0.9532- | 0.009 | 0.1905- | 0.029 | 0.9071- | 0.022 | **0.0407** | 0.024 |
| 150J6F | 0.6868- | 0.030 | 0.9320- | 0.008 | 0.1660- | 0.014 | 0.7943- | 0.028 | **0.0219** | 0.016 |
| 150J7F | 0.7444- | 0.042 | 0.9636- | 0.005 | 0.1867- | 0.025 | 0.8916- | 0.028 | **0.0229** | 0.008 |
| 200J6F | 0.8255- | 0.015 | 0.9245- | 0.008 | 0.2263- | 0.026 | 0.9744- | 0.029 | **0.0193** | 0.015 |
| 200J7F | 0.7920- | 0.029 | 0.9791- | 0.004 | 0.2070- | 0.024 | 0.9513- | 0.040 | **0.0157** | 0.009 |
| -/=/+ | 19/1/0 | | 19/1/0 | | 18/2/0 | | 20/0/0 | | | |

Table 3: Results of IGD metric for comparing five algorithms.

In the future, we will extend solution space transformation mechanism, focusing on discriminating whether the adjusted space outperforms the original space. Meanwhile, we are committed to explore the characteristics of ES-DHFJSP to design more efficient search operators. Additionally, DHFJSP with varying numbers of machines remains a challenge that needs to be addressed.

## Acknowledgements

This work is partially supported by National Natural Science Foundation of China under Grant (No. 62406104, 62472147), the Training Program for Young Backbone

Teachers in Higher Education Institutions of Henan Province under Grant No. 2025GGJS033, and the Key Scientific Research Project of Henan Provincial Higher Education under Grant No. 25A520029. This work is also supported by the China Scholarship Council under Grant No. 202408410273.

## References

Bar-Joseph, Z.; Gifford, D. K.; and Jaakkola, T. S. 2001. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, 17(suppl_1): S22–S29.

Cao, S.; Li, R.; Gong, W.; and Lu, C. 2023. Inverse model and adaptive neighborhood search based cooperative optimizer for energy-efficient distributed flexible job shop scheduling. *Swarm and Evolutionary Computation*, 83: 101419.

Cao, S.; and Yuan, Y. 2024. A Novel Memetic Algorithm for Energy-efficient Distributed Heterogeneous Flexible Job Shop Scheduling: Case Studies in UAVs Delivery. *IEEE Internet of Things Journal*.

Chen, Y.; Liao, X.; Chen, G.; and Hou, Y. 2024. Dynamic Intelligent Scheduling in Low-Carbon Heterogeneous Distributed Flexible Job Shops with Job Insertions and Transfers. *Sensors*, 24(7).

Hou, Y.; Liao, X.; Chen, G.; and Chen, Y. 2025. Co-Evolutionary NSGA-III with deep reinforcement learning for multi-objective distributed flexible job shop scheduling. *Computers & Industrial Engineering*, 203: 110990.

Li, J.; Han, Y.; Gao, K.; Xiao, X.; and Duan, P. 2024a. Bi-Population Balancing Multi-Objective Algorithm for Fuzzy Flexible Job Shop With Energy and Transportation. *IEEE Transactions on Automation Science and Engineering*, 21(3): 4686–4702.

Li, R.; Gong, W.; Lu, C.; and Wang, L. 2023a. A Learning-Based Memetic Algorithm for Energy-Efficient Flexible Job-Shop Scheduling With Type-2 Fuzzy Processing Time. *IEEE Transactions on Evolutionary Computation*, 27(3): 610–620.

Li, R.; Gong, W.; Wang, L.; Lu, C.; and Dong, C. 2024b. Co-Evolution With Deep Reinforcement Learning for Energy-Aware Distributed Heterogeneous Flexible Job Shop Scheduling. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 54(1): 201–211.

Li, R.; Gong, W.; Wang, L.; Lu, C.; and Jiang, S. 2022. Two-stage knowledge-driven evolutionary algorithm for distributed green flexible job shop scheduling with type-2 fuzzy processing time. *Swarm and Evolutionary Computation*, 74: 101139.

Li, R.; Gong, W.; Wang, L.; Lu, C.; and Zhuang, X. 2023b. Surprisingly Popular-Based Adaptive Memetic Algorithm for Energy-Efficient Distributed Flexible Job Shop Scheduling. *IEEE Transactions on Cybernetics*, 53(12): 8013–8023.

Luo, C.; Gong, W.; Li, R.; and Lu, C. 2023. Problem-specific knowledge MOEA/D for energy-efficient scheduling of distributed permutation flow shop in heterogeneous factories. *Engineering Applications of Artificial Intelligence*, 123: 106454.

Meng, L.; Zhang, C.; Ren, Y.; Zhang, B.; and Lv, C. 2020. Mixed-integer linear programming and constraint programming formulations for solving distributed flexible job shop scheduling problem. *Computers & Industrial Engineering*, 142: 106347.

Naderi, B.; and Azab, A. 2014. Modeling and heuristics for scheduling of distributed job shops. *Expert Systems with Applications*, 41(17): 7754–7763.

Nostrand, R. C. V. 2002. Design of Experiments Using the Taguchi Approach: 16 Steps to Product and Process Improvement. *Technometrics*, 44(3): 289–289.

Pan, Z.; Wang, L.; Wang, J.; and Zhang, Q. 2025. A Bi-Learning Evolutionary Algorithm for Transportation-Constrained and Distributed Energy-Efficient Flexible Scheduling. *IEEE Transactions on Evolutionary Computation*, 29(1): 232–246.

Smit, I. G.; Wu, Y.; Troubil, P.; Zhang, Y.; and Nuijten, W. P. 2025. Neural Combinatorial Optimization for Stochastic Flexible Job Shop Scheduling Problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 26678–26687.

Wang, J.; Han, H.; and Wang, L. 2024. A Feedback Learning-Based Memetic Algorithm for Energy-Aware Distributed Flexible Job-Shop Scheduling With Transportation Constraints. *IEEE Transactions on Evolutionary Computation*.

Wu, F.-C.; Qian, B.; Hu, R.; Zhang, Z.-Q.; and Wang, B. 2023. A Q-Learning-Based Hyper-Heuristic Evolutionary Algorithm for the Distributed Flexible Job-Shop Scheduling Problem. In *Advanced Intelligent Computing Technology and Applications*, 251–261.

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

Xuan, H.; Li, W.; and Li, B. 2023. An artificial immune differential evolution algorithm for scheduling a distributed heterogeneous flexible flowshop. *Applied Soft Computing*, 145: 110563.

Yan, Q.; Wang, H.; and Yang, S. 2024. A Learning-Assisted Bi-Population Evolutionary Algorithm for Distributed Flexible Job-Shop Scheduling With Maintenance Decisions. *IEEE Transactions on Evolutionary Computation*.

Yang, Z.; Hu, X.; Li, Y.; Liang, M.; Wang, K.; Wang, L.; Tang, H.; and Guo, S. 2025. A Q-learning-based improved multi-objective genetic algorithm for solving distributed heterogeneous assembly flexible job shop scheduling problems with transfers. *Journal of Manufacturing Systems*, 79: 398–418.

Zhang, Q.; Shao, W.; Shao, Z.; Pi, D.; and Gao, J. 2024. Deep reinforcement learning driven trajectory-based metaheuristic for distributed heterogeneous flexible job shop scheduling problem. *Swarm and Evolutionary Computation*, 91: 101753.

Zhao, F.; Du, Y.; Zhuang, C.; Wang, L.; and Yu, Y. 2025a. An Iterative Greedy Algorithm for Solving a Multiobjective

Distributed Assembly Flexible Job Shop Scheduling Problem With Fuzzy Processing Time. *IEEE Transactions on Cybernetics*, 55(5): 2302–2315.

Zhao, F.; and Li, M. 2024. Reinforcement Learning-Based Estimation of Distribution Algorithm for Energy-Efficient Distributed Heterogeneous Flexible Job Shop Scheduling Problem. In *Advanced Intelligent Computing Technology and Applications*, 183–195.

Zhao, F.; Li, M.; Zhu, N.; Xu, T.; and Jonrinaldi. 2025b. Multi-objective fitness landscape-based estimation of distribution algorithm for distributed heterogeneous flexible job shop scheduling problem. *Applied Soft Computing*, 171: 112780.