A Decomposition-Based Coevolutionary Algorithm to Solve Distributed Heterogeneous Hybrid Flow Shop Scheduling Problem with Job Deadlines and Priorities

Hua Xu *, Lingxiang Huang *, Juntai Tao, Chenjie Zhang, Jianlu Zheng

Posted Date: 23 March 2026

doi: 10.20944/preprints202603.1709.v1

Keywords: distributed heterogeneous hybrid flow shop scheduling; job deadlines and priorities; distributionary environment; elite selection strategy; problem-based operator selection strategies

 doi:10.20944/preprints202603.1709.v1

Article

A Decomposition-Based Coevolutionary Algorithm to Solve Distributed Heterogeneous Hybrid Flow Shop Scheduling Problem with Job Deadlines and Priorities

Hua Xu *, Lingxiang Huang *, Juntai Tao, Chenjie Zhang and Jianlu Zheng

School of Artificial Intelligence and Computer Science, Jiangnan University, Li Hu Avenue, 214122 WuXi Jiangsu Province, China

* Correspondence: xuhua@jiangnan.edu.cn, 6233115011@stu.jiangnan.edu.cn

Abstract

Distributed heterogeneous hybrid flow shop scheduling with job deadlines and priorities (DHHFSP-JDP) is a combination of scheduling problem and distributionary environment. Addressing complex work sequences and energy consumption in distributed manufacturing with heterogeneous plants is a major challenge. It is necessary for optimizing total weighted delay (TWD) and total energy consumption (TEC) in distributed heterogeneous green hybrid flowshops. A model using mixed integer linear programming is applied to describe DHGHFSP-JDP and a decomposition-based coevolutionary algorithm (DBCEA) is considered to be the solution in this article. In this approach, (1) a decomposition-based heuristic initialization is proposed, in which an initialization strategy with a randomly sized population is adopted to establish effective initial schedules. (2) elite selection strategy based on the integration of an external archive and an elite archive. (3) four problem-based operator selection strategies embedded in a cooperative local search framework, and an Upper Confidence Bound (UCB) mechanism to design a strategy for selecting local search operators. In the end, The superiority of DBCEA is validated through comparative experiments against several advanced algorithms across 20 benchmarks, with results showing it often provides the best Pareto solution set.

Keywords: distributed heterogeneous hybrid flow shop scheduling; job deadlines and priorities; distributionary environment; elite selection strategy; problem-based operator selection strategies

1. Introduction

The swift evolution of automated and intelligent production technologies has significantly expanded the production capacity of current industrial systems. In this context, production scheduling has become a crucial decision-making problem in manufacturing and operations management [1]. Scheduling problems are prevalent in various real-life scenarios, including vehicle routing problem [2], automated material handling system [3], aircraft dispatch mission [4] and so on, highlighting their practical and theoretical importance.

Among various scheduling models, substantial studies have focused on traditional flow shop and job shop scheduling problems. Flow shop scheduling problem(FSP), such as the no-wait FSP [5], hybrid FSP [6,7], and distributed FSP [8], are widely used to describe production environments in which jobs follow the same processing route but may be handled by parallel or geographically distributed machines. Similarly, the job shop scheduling problem (JSP) is considered as a fundamental model for describing complex manufacturing systems in which jobs have different processing routes and machine requirements. However, the rising sophistication of modern industrial operations environments makes the classical JSP insufficient for representing real industrial systems, leading to the development of many variants such as dynamic job shop problem [9], fuzzy job shop problem [10]. In order to respond promptly to market requirements and minimized production durations, major industrial apparatus and customized products are typically produced across several geographically dispersed manufacturing

 doi:10.20944/preprints202603.1709.v1

sites. This distributed production mode make it possible for enterprises to better utilize resources, reduce production costs and risks, and improve overall system flexibility. Consequently, distributed scheduling problems, which integrate shop-floor scheduling with factory assignment decisions, have become increasingly important and more representative of real-world production environments.

In practical distributed manufacturing systems, production environments are usually more complicated than those described by classical flow shop or job shop models. Machines in different factories often have heterogeneous processing capabilities, multiple parallel machines exist at each production stage, and energy consumption and carbon emissions have become critical concerns due to increasingly strict environmental regulations. These characteristics give rise to the distributed heterogeneous hybrid flow shop scheduling problem (DHHFSP) [11] that incorporates factory assignment, hybrid flow shop scheduling, machine heterogeneity, and green production considerations into a unified framework. In fact, real-world production scheduling environments are even more complex. To more accurately reflect the time requirements and priority levels of job processing, the DHHFSP-JDP incorporates job due dates and production priorities.

In DHHFSP-JDP, jobs must first be allocated to geographically distributed factories. After that, jobs are processed executed across production stages, each of which may contain several heterogeneous parallel machines. Meanwhile, the scheduling process must simultaneously consider not only traditional production objectives such as makespan and workload balance, but also environmental indicators such as energy consumption and carbon emissions. In addition, each job should be scheduled according to its priority order, and every job should be arranged so as to be completed as much as possible before its due date. As an expansion of the HFSP, the DHHFSP-JDP is an NP-hard problem [12], and resolving this problem using straightforward analytical methods is challenging. Several methods have been suggested in the literature to handle these problems, such as: tabu search [13], NSGA-II [14], and so on. The present work investigates a decomposition-based coevolutionary algorithm aimed at solving DHHFSP-JDP, the following points highlight our principle contributions:

(1) Motivated by practical manufacturing scenarios, this study focuses on a complex production scheduling environment, where different plants exhibit distinct processing characteristics. In addition, time-constrained production requirements are considered by introducing job-specific due dates and priority levels. The problem has been estiablished in section 3.

(2) A hybrid approach combining decomposition-based algorithm with external archive and a problem-based local search. A multi-popularion initialization with hybrid initial algorithm.

(3) A external archive combined with elite archive to integrate high-quality solution sets. An effective neighborhood structure is designed for local search strategy based on problem characteristics.

The structure of the rest of this paper is outlined as follows: Section 2 introduces the related work of previous studies. In section 3, it describes the DHHFSP-JDP and constructs its mathematical representation. Section 4 describes the details of the framework and elements of the DBCEA. Section 5 presents detailed experimental results and data analysis findings. Section 6 concludes the key contributions of the paper and highlights avenues for future work.

2. State of the Art

Heuristic algorithms are predominantly used to address FSP [15]. There exist various scheduling problem ranging from simple to complex. As a fundamental scheduling model, there have been extensive expansions for FSP in various areas [16]. For example, in [5], to tackle the energy-efficient no-wait flow shop scheduling problem (EENWFSP), a two-phase cooperative evolutionary framework incorporating problem-oriented knowledge was developed. By thoroughly examining the structural characteristics of the problem, two tailored constructive heuristics were devised to produce high-quality initial solutions, thus improving the efficiency of the search process from the outset. Wang et al. [17], who considered the reentrant flexible flowshop scheduling problem (RFFSP), a reinforcement learning driven strategy selection mechanism was incorporated to dynamically determine appropriate crossover and mutation operators throughout the evolutionary process. This adaptive scheme enables

 doi:10.20944/preprints202603.1709.v1

the algorithm to adjust operator choices according to the search state, and improve exploration and exploitation balance and overall optimization performance.

Regarding the hybrid flow shop scheduling problem (HFSP), Janis et al. provided a comprehensive review of multi-objective HFSPs [7], systematically summarizing performance criteria for Pareto front evaluation. In [6], an energy-aware multi-objective optimization algorithm (EA-MOA) was proposed to address the HFSP with consideration of the setup energy consumptions, furthermore, eight distinct neighborhood structures were constructed to enhance local search capability, which has exhibited strong robustness and effectiveness. Wang et.al [18] considered a green hybrid flowshop rescheduling problem with consistent sublots (GHFRP_CS) and a constructive heuristic incorporating lot-splitting, sequencing, and local search rules have been designed to produce high-quality initial solutions, with further improvements realized via a multi-objective discrete artificial bee colony (MDABC) algorithm. In order to solve uncertainty in the energy-efficient HFSP, Wang et al. enhanced the Existing nondominated sorting genetic algorithm-II (ENSGA-II) to solve a fuzzy energy-aware HFSP that has adjustable machine processing speeds [19]. In [13], a hybrid evolutionary framework that employs dual solution encoding schemes was considered to enhance representational diversity. Additionally, a tabu search procedure grounded in a disjunctive graph model was embedded to intensify the search and explore a broader solution space. Through the integration of global evolutionary exploration and local refinement, the approach achieved a desirable balance between solution quality and computational cost.

The distributed nature of the system introduces substantial additional complexity to the scheduling problem. For distributed hybrid flow shop scheduling problem(DHFSP), Li et al. [20] examined distributed hybrid flow shop scheduling problem with batch-processing machines(DHFSP-BVS) and flexible sublot division, a novel multi-objective hybrid evolutionary algorithm integrating a dynamic weight adjustment mechanism (NHMOEA/D) was developed to concurrently minimize and optimize the trade-off makespan and energy consumption. Wang developed a bi-population cooperative memetic framework (BCMA), within which two problem-specific heuristic strategies were incorporated to strengthen search capability and alleviate premature convergence [21], the proposed framework consists of a cooperative initialization scheme, a bi-population co-evolution mechanism, and a local intensification strategy to enhance search efficiency and solution quality. Zheng et al. [22] considered a multi-objective fuzzy distributed hybrid flow shop problem (MFDHFSP) which has indeterminate operation times and delivery deadlines, and a co-evolutionary algorithm incorporating domain-specific strategies was developed by effectively integrating an estimation of distribution algorithm (EDA) with an iterative greedy search (IG). This hybridization enabled a balanced interplay between global and local search throughout the process. In [23], Zhang et al. investigated a distributed hybrid differentiated flow shop problem (DHDFSP) and proposed a distributed co-evolutionary memetic algorithm (DCMA) to efficiently address the scheduling complexities inherent in such systems. Wang et al. [24] examined a many-objective DHFSP and formulated its corresponding mathematical model. A multi-agent reinforcement learning enhanced algorithm (MRLEA) was then proposed, incorporating an evolution framework organized on a grid topology alongside a local search procedure guided by agent-based group decision frameworks.

Considering the heterogeneous factories in DHFSP, distributed heterogeneous hybrid flowshop scheduling problem (DHHFSP) becomes a more complex scenario. In [25], a distributed heterogeneous batching-integrated assembly hybrid flow shop scheduling (DHBIAHFS) problem in the pharmaceutical industry was investigated, and a multi-objective immune balancing algorithm (MOIBA) was proposed. The algorithm incorporates domain-adapted crossover and mutation mechanisms along with dynamically adaptive probabilities to enhance both solution diversity and convergence performance. In [11], a distributed heterogeneous hybrid flow shop scheduling problem with lot-streaming (DHHFSPLS) was considered and a knowledge-driven many-objective evolutionary algorithm (KD-MaOEA) was developed. The algorithm, characterized by strong exploitation capabilities, effectively optimizes each objective considering electricity tariffs that differ across usage periods. Shao et al. [26]

 doi:10.20944/preprints202603.1709.v1

proposed an ant colony optimization behavior driven multi-objective evolutionary algorithm based on decomposition (ACO_MOEA/D). The approach leverages problem-specific ant colony behaviors to generate descendant solutions, and integrates a strategy for minimizing energy consumption to enhance scheduling efficiency. Qin et al. [27] investigated a MILP model for the DHHFSP with blocking constraints and proposed a collaborative iterative greedy (CIG) algorithm. A a factory-specific local refinement strategy was incorporated to fine-tune the production sequence, enhancing solution quality.

Table 1 summarizes the methodologies employed in the reviewed studies including addressed problems, their model objectives, solution approaches.

Table 1. Methodologies employed of reviewed studies and solution approaches.

Reference	Addressed problem	Model objective(s)	Solution approach
zhao et al [5]	NWFSP	Makespan, TEC	TS-CEA
Wang et al [17]	RFFSP	Total tardiness	RLGA
Li et al [6]	HFSP	Makespan, TEC	EA-MOA
Wang et al [18]	GHFPR_CS	Makespan, TEC, System stability	MDABC
Wang et al [19]	EFHFSP	Makespan,TEC	ENSGA-II
Fan et al [13]	HFSP	Makespan	HEA
Li et al [20]	DHFSP-BVS	Makespan,TEC	NHMOEA/D
Wang et al [21]	DHFSP	Makespan	BCMA
Zheng et al [22]	MFDHFSP	Response value	CCA
Zhang et al [23]	DHDFSP	Makespan	DCMA
Wang et al [24]	DHFSP	Makespan, TEC, WC, WE, WT	MRLEA
Wang et al [25]	DHBIAHFSP	Makespan, tardy product count	MOIBA
Chen et al [11]	DHHFSPLS	Makespan, total earliness, total tardiness, TEC	KDMaOEA
Shao et al [26]	DHHFSP-NTOU	Makespan, TEC	ACO_MOEA/D
Qin et al [27]	DHHFSP with blocking constraints	Makespan	CIG

3. Problem Description and MILP Model for DHHFSP-JDP

3.1. Problem Description for DHHFSP-JDP

In DHHFSP-JDP, the composition varies from plant to plant. Consider a real-life production procedure in DHHFSP. A set of n jobs is required to be assigned to n_f factories. There are t processing stages, with m_t parallel machines available at stage t. Specifically, every job i is associated with a deadline D_i and a priority level R_i, and is expected to be completed no later than its specified due date. A smaller value of R_i indicates higher importance and urgency of the corresponding job.

Three key decision issues are involved in this model: (1) Assignment of jobs to specific factories. (2) Selection of parallel machines at each stage for every job. (3) Determination of the processing sequence throughout the factories. On top of that, the assumptions underlying the DHHFSP-JDP are summarized as follows:

(1) Every parallel mechine can process just a single job at a time.

(2) Every job is assigned exclusively to a single factory.

(3) Due to heterogeneity among factories, across all operational stages, job processing durations are distinct among the factories.

(4) All machines are assumed to be available at the first start time point.

(5) Disregarding transition times and stochastic scheduling events.

(6) Jobs are processed without preemption.

3.2. Mathematical Model for DBCEA

Some notations used in modeling DHHFSP-JDP of this work are as follows:

Indices:

i : Job index, $i \in \{1, 2, \ldots, N\}$

f : Factory index, $f \in \{1, 2, \ldots, F\}$

 doi:10.20944/preprints202603.1709.v1

s : Stage index, $s \in \{1, 2, \ldots, S\}$
m : Machine index, $m \in \{1, 2, \ldots, M\}$
q : Position index, $q \in \{1, 2, \ldots, Q\}$
Parameters:
N : Total job number
F : Total factory number
S : Total stage number
M : Total parallel machine number
Q : Total position number
L : A positive scalar of significant magnitude
EC_d : Idle-time per energy consumption
EC_p : Processing-time per energy consumption of
D_i : Job due date
R_i : Job priority, $R_i \in \{1, 2, 3\}$
$TC_{f,i,s}$: The time consumption associated with job i at stage s in factory f
$M_{f,s,m}$: Machine number of f factory, stage s, m machine
γ_{R_i} : Penalty parameter; for each R_i, $\gamma_{R_i} \in \{2.5, 1.5, 1\}$
Variable:
$T_{f,i,s}$: The processing start time of job i at stage s in factory f
$MT_{f,i,s}$: The termination time of job i during s at factory f
$PT_{f,s,m,q}$: The time at which machine $M_{f,s,m}$ begins processing at position q in stage s of factory f
$EC_{f,s,m,q}$: The energy consumed by machine $M_{f,s,m}$ during idle periods at position q in stage s of factory f
DC_i : The delay of job i beyond its due date, $DC_i = max(FT_{f,i,S} - D_i, 0)$
$A_{f,i}$: A decision variable indicating whether job i is assigned to factory f, (1) or not (0)
$B_{f,s,m,i,q}$: A decision variable indicating whether job i is assigned to position q on machine $M_{f,s,m}$ at stage s in factory f, (1) or not (0)

The DHHFSP-JDP is approached with the goal of diminishing two opposing optimization metrics, which are total energy consumption (TEC) and total weighted delay (TWD) [28], elaborated as follows:

(1) Energy consumption in the system is composed of contributions from both operational processing and non-operational standby periods.

$$TEC = \sum_{f=0}^{F} \sum_{i=0}^{N} \sum_{s=0}^{S} A_{f,j} * TC_{f,i,s} * EC_p + \sum_{f=0}^{F} \sum_{s=0}^{S} \sum_{m=0}^{M} \sum_{q=0}^{Q} EC_{f,s,m,q} \tag{1}$$

(2) Total weighted delay (TWD) reflects the operational revenue of the factories; a smaller TWD indicates better fulfillment of orders and a more appropriate balance between job priorities and due-date constraints.

$$TWD = \sum_{i=0}^{N-1} max(0, DC_i) * \gamma_{R_i} \tag{2}$$

The follows are the MILP model:

$$\begin{cases} min\ F_1 = TEC \\ min\ F_2 = TWD \end{cases} \tag{3}$$

 doi:10.20944/preprints202603.1709.v1

subject to:

$$\sum_{f=1}^{F} A_{f,i} = 1, \forall i \tag{4}$$

$$\sum_{f=1}^{F} \sum_{m=1}^{M} \sum_{q=1}^{Q} B_{f,s,m,i,q} \leqslant 1, \forall s, i \tag{5}$$

$$\sum_{m=1}^{M} \sum_{q=1}^{Q} B_{f,s,m,i,q} = A_{f,i}, \forall f, s, i \tag{6}$$

$$\sum_{i=1}^{N} B_{f,s,m,i,q} \geqslant \sum_{i=1}^{N} B_{f,s,m,i,q+1}, \forall f, s, m, q \in Q \tag{7}$$

$$\sum_{q=1}^{Q} \sum_{m=1}^{M} B_{f,s,m,i,q} * TC_{f,i,s} = MT_{f,i,s} - T_{f,i,s}, \forall f, i, s \tag{8}$$

$$\sum_{q=1}^{Q} \sum_{m=1}^{M} B_{f,s,m,i,q} * TC_{f,i,s} \leqslant T_{f,i,s+1} - T_{f,i,s}, \forall f, i, s \in S \tag{9}$$

$$\sum_{i=1}^{N} B_{f,s,m,i,t} * TC_{f,i,s} \leqslant PT_{f,s,m,q+1} - PT_{f,s,m,q}, \forall f, s, m, q \in Q \tag{10}$$

$$EC_{f,s,m,q} \geqslant (PT_{f,s,m,q+1} - PT_{f,s,m,q} - \sum_{i=1}^{n} B_{f,s,m,i,q} * TC_{f,i,s}) * EC_d, \forall f, s, m, q \in Q \tag{11}$$

$$DC_i \leqslant MT_{f,i,S} - D_i + L(1 - Y_{f,s,m,i,q}), DC_i \geqslant MT_{f,i,S} - D_i - L(1 - Y_{f,s,m,i,q}),$$
$$\forall f, m, i, q, s \in S \tag{12}$$

$$A_{f,i} \in \{0, 1\}, \forall f, i \tag{13}$$

$$B_{f,s,m,i,q} \in \{0, 1\}, \forall f, s, m, i, q \tag{14}$$

Formulas (1) and (2) define the two optimization objectives. Equation (3) minimizes TEC and TWD. Equation (4) stipulates each job is restricted to assignment at only one factory. Constraints (5) prevents a job from being executed on multiple machines concurrently. Constraint (6) guarantees every job passes through all processing stages within the designated factory. Constraint (7) determines arrangement of jobs along schedule of each machine. Constraint (8) specifies the operation time. Constraints (9) and (10) describe the precedence relationships between consecutive stages and adjacent positions, respectively. Constraint (11) calculates TEC. Constraint (12) enforces the due-date requirement for each job. Finally, Constraints (13) and (14) define the decision variables.

3.3. A Simple Illustrative Example of DHHFSP-JDP

An illustrative example is given below to demonstrate the DHHFSP-JDP. Table 2 is the data setting in this example. There are six jobs ($j_1, j_2, j_3, j_4, j_5, j_6$), two factories ($F_1, F_2$), three job priority levels ($R_i \in \{1, 2, 3\}$). Weights of $\gamma_{R_i} \in \{2.5, 1.5, 1\}$ are assigned to the respective priority levels. Each factory contains two processing stages, and two concurrent machines available at each stage. For every stage, the processing time of every job is provided for the various parallel machines within both factories. The job sequence of this example is $JSV = \{6, 3, 1, 5, 4, 2\}$, and the assignment of factory $FAV = \{1, 2, 1, 2, 1, 2\}$, implying that the job sets $\{j_6, j_1, j_4\}$ are allocated to factory F_1, the rest of jobs $\{j_3, j_5, j_2\}$ are assigned to factory F_2. The gantte chart of this scheme is shown at picture Figure 1. The idle and operating power consumption of each parallel machine are assumed to be 2 kW/h and 4 kW/h, respectively. Thus, in factory F_1, The energy consumption for processing workpieces j_4, j_1, and j_6 is $4 * 14 = 56$ kW, $4 * 11 = 44$ kW, and $4 * 14 = 56$ kW units, respectively. Only the $j_1(\gamma_{R_1} = 1.5)$ has exceeded the deadline, so the weighted delay is $(20 - 17) * 1.5 = 4.5$ h. For factory F_2, The energy

 doi:10.20944/preprints202603.1709.v1

consumption for processing workpieces j_5, j_2, and j_3 is $4 * 16 = 64$ kW, $4 * 13 = 52$ kW, and $4 * 13 = 52$ kW units, respectively. There is idle time from j_2 to j_3 at stage 2, so the idle energy consumption is $2 * 3 = 6$ kW. The $j_3(\gamma_{R_3}$=1) has exceeded the deadline, the weighted delay is $(20 - 18) * 1 = 2$ h. In summary, the total energy consumption $TEC = 56 + 44 + 56 + 64 + 52 + 52 + 6 = 330$ kW. the total weighted delay $TWD = 4.5 + 2 = 6.5$ h.

Table 2. A representative instance of the DHHFSP-JDP, comprising six jobs, two factories, and three distinct job priority levels.

	F_1		F_2			
Job	$S_{1,1}$	$S_{1,2}$	$S_{2,1}$	$S_{2,2}$	D_i	R_i
j_1	5	4	6	7	17	2
j_2	9	7	6	6	16	1
j_3	8	9	7	4	18	3
j_4	6	5	8	7	15	2
j_5	9	8	7	6	18	1
j_6	10	9	6	5	16	3

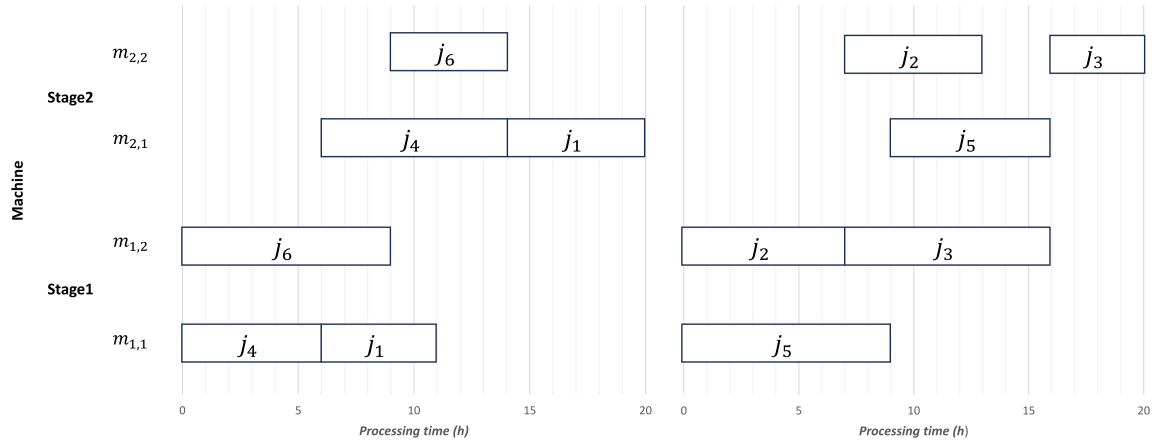

Figure 1. Gantte chart for Table 2.

4. Description of DBCEA

4.1. Framework of DBCEA

In this section, we provide a detailed execution framework for DBCEA. We present step-by-step descriptions of the encoding and decoding method, initialization strategy, energy-saving stategy, evolution operations, decomposition-based strategy and local search strategy. Algorithm 1 outlines the constituent elements in detail and procedures of the DBCEA.

 doi:10.20944/preprints202603.1709.v1

Algorithm 1 The framework of DBCEA

Require: Input Parameters:population size (ps), crossover rate (P_c), mutation rate (P_c).
Ensure: The Pareto solution set.

Let ps be the population size and initialize the population to generate PS. (c. f. subsection 4.3)
Assemble the elite archive according to the fitness of candidate solutions.
Establish the external archive(E_a).
for i=1 to $MaxIter$ **do**
    Calculate the fitness and generate neighborhood(T) based on the popsize. (c. f. subsection 4.4)
    Execute evolutionary operators on each neighborhood. (c. f. subsection 4.4)
    Update neighboring solutions, decide whether the existing solution should be replaced by the new one. (c. f. subsection 4.4)
    Update the external archive.
    Update the elite archive.
    Local search stategy on each individual. (c. f. subsection 4.5)
    Energy saving strategy. (c. f. subsection 4.6)
end for
Insert the elite solutions into the Pareto-optimal set.

4.2. Encoding and Decoding Method

(1) Encoding method

To represent solutions, two one-dimensional vectors are employed in the proposed approach. Factory allocation vector(FAV) contains the information about the factory assignment strategy for each job; Job sequence vector(JSV) gives the processing sequence information of different jobs.

For a clearer statement, Figure 2 shows a small example. There are six jobs and two heterogeneous factories in a scheduling plan, the set of the operation is $\{6, 3, 1, 5, 4, 2\}$ means the job sequence. FAV in Figure 2 indicates that values j_1, j_3, j_5 are allocated to the first factory, the j_2, j_4, j_6 are allocated to the second factory. The value of JSV in Figure 2 denotes job number, each number uniquely represents a job, and their sequence denotes the processing order.

(2) Decoding method

With regard to decoding method, it maps the encoding vectors JSV and FAV into valid scheduling schemes. Assignment of jobs to factories is guided by FAV, followed by determination of the job sequence within each factory using JSV. Finally, jobs are distributed across the parallel machines in compliance with all scheduling constraints.

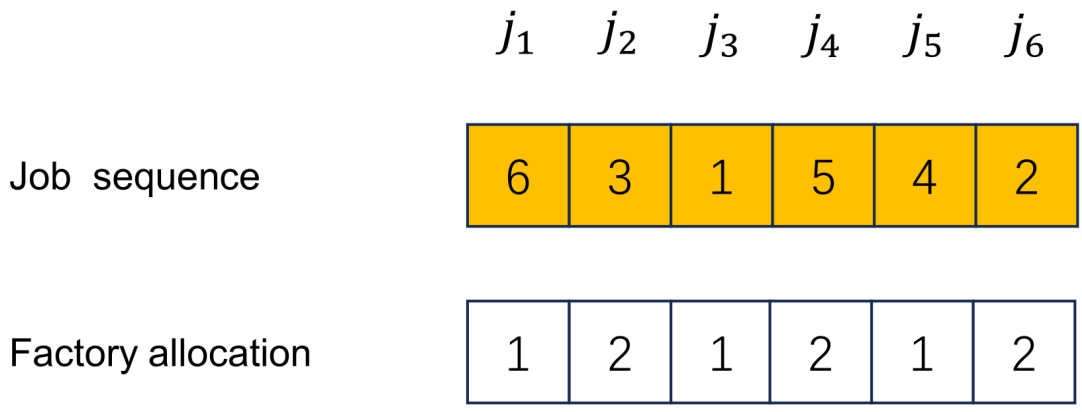

Figure 2. Solution representation.

4.3. Initialization Method

This section introduces the initialization of the DBCEA. When generating an initial population, there are currently many effective heuristic rules proposed by other studies, such as random rule [29], Nawaz-Enscore-Ham(NEH) algorithm [30], IG-based Heuristic initialization [31] and so on. To obtain a more diverse and effective initial solution set, several heuristic rules has been applied as follows:

 doi:10.20944/preprints202603.1709.v1

(1) Problem-based initialization, according to job due dates and priorities, shown in Algorithm 2. An index array is generated by ordering the jobs according to ascending priority, while a separate index array is constructed by arranging the jobs in ascending order of their due dates.

(2) Load balance initialization, every job is allocated to the factory exhibiting the lowest cumulative workload, ensuring a balanced distribution across all factories.

(3) Random rule, all initial individuals are randomly generated, and the population can be distributed as widely as possible in the solution space.

Besides, we have applied NEH algorithm, there are total four initial algorithm. These four rules are put into a pool, upon each initialisation, the original population is divided into four populations of random sizes, each matched one-to-one with the four methods in the initialized algorithm pool.

Algorithm 2 Rank based initialization

Require: Input: population size (ps), total job number (N), jobs due date (D_i), jobs priority (R_i).
Ensure: Output: job sequence vector JSV
1: RR: Ordered by increasing values of R_i
2: DD: Ordered by increasing values of of D_i
3: **for** Each $i \in [1, N]$ **do**
4: $\quad R_{RR_i} = (N - i)/N$
5: $\quad D_{DD_i} = (N - i)/N$
6: $\quad PRO = R_{RR_i} + D_{DD_i}$
7: **end for**
8: JSV: Ascending order of PRO

4.4. Decomposition-Based Strategy

After initialization, it comes decomposition-based stategy. The appropriate size(T) of neighborhood during the execution of the algorithm is ten persent of the original popsize. The initial domain matrix is randomly configured and it will be saved, every ten generations, the domain matrix undergoes an update. When employing new solutions to update the domain matrix, it is necessary to determine whether the new solutions are superior. Solution evaluation was performed through fast nondominated sorting in conjunction with a crowding distance metric to preserve diversity within the population. Algorithm 3 has shown the procedure of this strategy.

Algorithm 3 Decomposition-based strategy

Require: Input: population size (ps), total job number (N), neighborhood T, rate of crossover(P_c), rate of mutation (P_m).
Ensure: Output: External solution E_a
1: Generate External solution E_a
2: **for** i=1 to $MaxIter$ **do**
3: $\quad$ Generate neighbor matrix N_{ei} when $i \mod 10 = 0$
4: $\quad$ Traversing the pair problem, selection, crossover and mutation
5: $\quad$ Merge subproblems and Update the N_{ei}
6: $\quad$ Update the External solution E_a
7: **end for**
8: E_a

When dealing with paired problems, the evolution operator is important which resulting in a good offspring solution. It consists of two parts: crossover operator and mutation operator.

(1) Crossover operator

Since we use two-dimensional vectors to represent a solution, we adopt two crossover methods and Figures 3 and 4 provides an instance for understanding.

Universal crossover(UX) specifically designed for factory allocation in scheduling and process planning optimization. The operator introduces a randomized, template-guided gene exchange mechanism to strengthen the search ability of evolutionary algorithms within the solution space:

 doi:10.20944/preprints202603.1709.v1

•Step1 : A 0 or 1 vector, with its length corresponding to the overall count of operations, is generated at random..

•Step2 : Select two solution S_1 and S_2 Exchange their value , where it is 1 at the same position in 0 or 1 vector.

Precedence operation crossover(POX) is adopted to process job sequence vector.

• Step1 : The complete set of jobs is allocated into two subsets randomly(Set_1, Set_2).

• Step2 : Select two paternal individual ($P_{js}1$, $P_{js}2$), find the jobs in Set_1 from P_1 and copy them to $New_{js}1$ in the same location.

• Step3 : Identify the jobs in Set_2, and sequentially insert them into the unassigned positions of $New_{js}1$, preserving their original left-to-right order.

• Step4 : Follow the same procedure to generate $New_{js}2$.

(2) Mutation operator Job sequence mutation: Two positions in the operation sequence are drawn at random, after that, swap their values. Factory allocation mutation: Two positions in the factory assignment vector are randomly chosen, and a new factory is assigned to each from its respective candidate set.

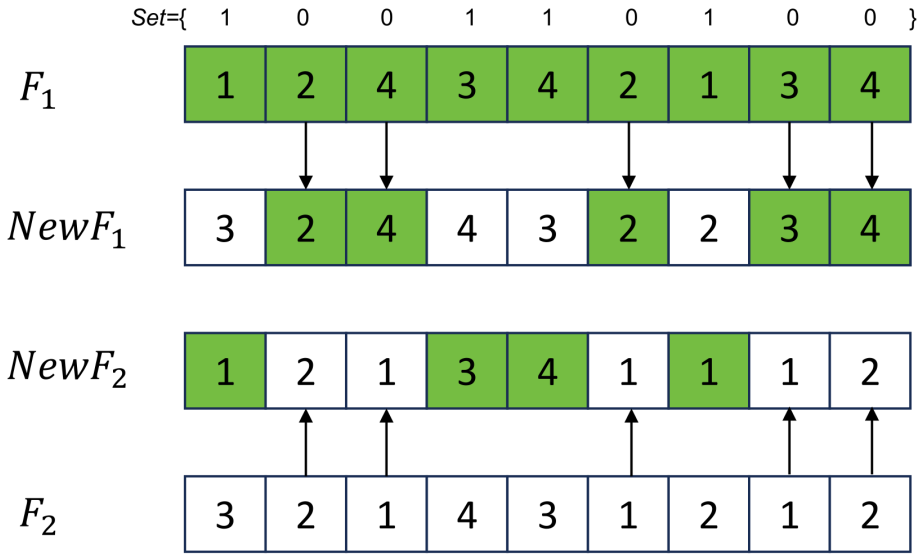

Figure 3. UX crossover.

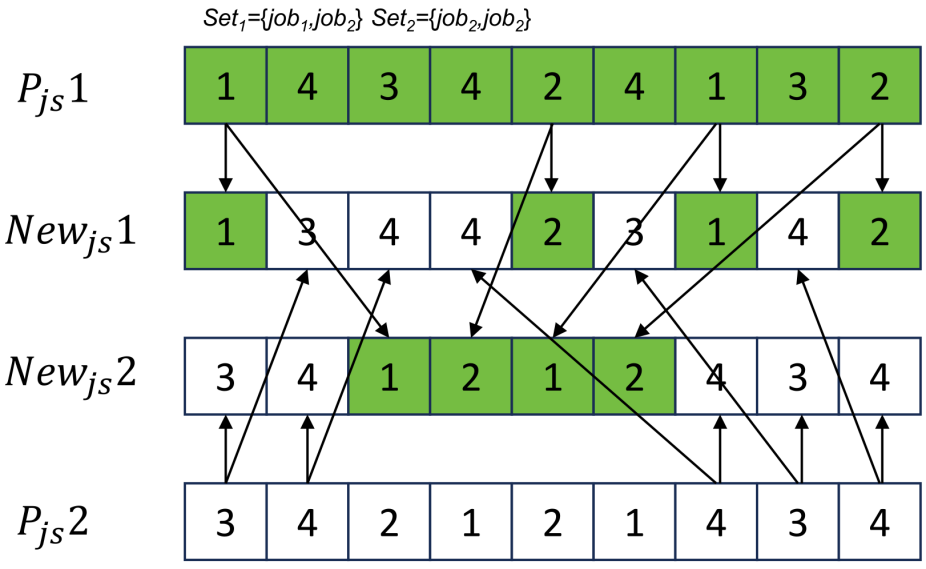

Figure 4. POX crossover.

 doi:10.20944/preprints202603.1709.v1

4.5. Local Search Strategy

Local search is an effective approach to enhancing solution quality in scheduling applications. Four neighborhood structures were designed in this study, all of which concentrate on the critical job, identified as the one exhibiting the largest total weighted delay (TWD). The local search heuristics is shown as follows and Figure 5 has shown the procedure:

(1) LS_1: Starting from the critical job and iterating backward through the sequence, a preceding job is examined at each step. If a job is found whose priority level is lower than that of the critical job, or whose due date is earlier when both jobs share the same priority, a swap operation is performed between the critical job and the identified job. This operator aims to improve the relative order of jobs by promoting higher-priority or more urgent jobs forward in the sequence.

(2) LS_2: Starting from the critical job and proceeds backward. When a job is designated a lower priority level, or having an earlier due date under equal priority, is identified, the critical job is removed from its current position and inserted immediately before the located job. Compared with the swap operator, this insertion-based strategy allows a more flexible positional adjustment while still emphasizing priority and urgency.

(3) LS_3: Starting from the critical job and iterating backward through the sequence, a swap is triggered if a preceding job has a earlier due date, or has a lower priority when both jobs share the same due date. this rule gives precedence to due dates over priority levels, thereby strengthening the algorithm's ability to reduce tardiness-related objectives.

(4) LS_4: Starting from the critical job and iterating backward through the sequence, once a job with an earlier due date, or with a lower priority under equal due dates, is found during the backward traversal, the critical job is extracted from its current position and inserted immediately before the identified job. This operator provides a finer-grained adjustment mechanism focused primarily on due date urgency.

In co-evolutionary algorithms, local search is typically used to refine and enhance current high-performing individuals in order to further improve the quality of the solution. However, the effectiveness of different local search operators often varies significantly across different search phases. For example, certain operators are better suited for broad-scale perturbations in the early stages, while others are more suitable for fine-tuning and optimization in the later stages. If local search operators are selected using fixed probabilities or a completely random approach, two problems may arise: first, highly efficient operators may not be fully utilized; second, operators that are potentially effective but have been tried fewer times may be overlooked for a long time. To address this, this paper introduces an Upper Confidence Bound (UCB) mechanism to design a local search operator selection strategy. This strategy treats each neighborhood operator as a "candidate arm" and, by comprehensively considering the operator's historical average return and usage frequency, adaptively determines which operator should be prioritized for the next call, thereby achieving a dynamic balance between exploitation and exploration. In this algorithm, the local search phase employs four distinct neighborhood structures, each of which perturbs and enhances the current solution from a different perspective, thereby exhibiting distinct search characteristics. By integrating these operators into a unified UCB selection framework, the algorithm can dynamically adjust the frequency of operator usage based on search feedback. Let the total number of local search operators be K . For the kth operator, let its cumulative reward be R_k, the number of times it has been selected be N_k, and the total number of selections for all operators be N. Then the upper confidence bound for the kth operator is defined as:

$$UCB_k = \frac{R_k}{n_k + \varepsilon} + c\sqrt{\frac{\ln(N+1)}{n_k + \varepsilon}} \tag{15}$$

ε is the smallest positive number that prevents the denominator from being zero, c is the exploration coefficient, used to adjust the balance between exploitation and exploration.

 doi:10.20944/preprints202603.1709.v1

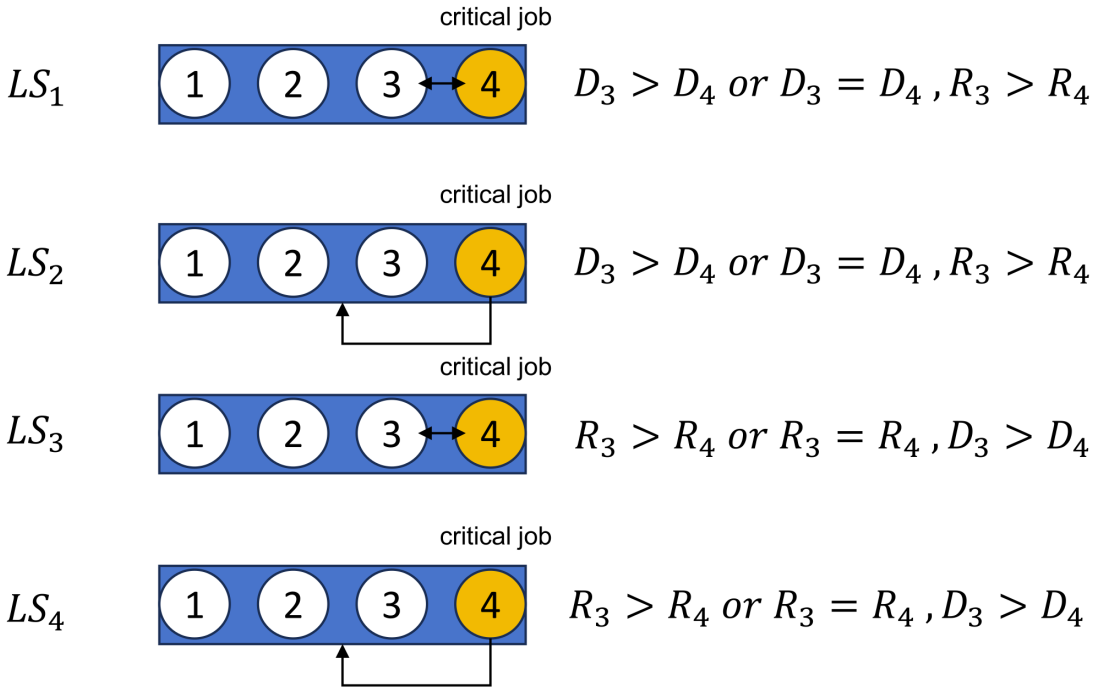

Figure 5. Local search strategy.

4.6. Energy-Saving Strategy

To achieve energy savings within the manufacturing process, a direct consideration of energy expenditure is of critical importance during machine idle time. In many scheduling solutions, machines may remain idle between consecutive operations, leading to unnecessary energy waste. In order to resolve this challenge, a right-shift strategy is adopted in this study established to reduce idle energy consumption without altering the feasibility or maximum completion time across all jobs.

Given a set of decoded scheduling solutions, a reverse decoding procedure is applied. Specifically, jobs are executed one by one in descending order, starting from the last scheduled job and moving toward the first. This backward adjustment enables the algorithm to exploit potential idle gaps that occur after operations while ensuring that all technological and temporal constraints remain satisfied.Before performing a right-shift operation on a job, several feasibility conditions must be verified: (1) Existence of idle time: There must be an idle interval immediately following the operation on the current machine, providing space for a rightward shift. (2) Due date criterion: The job's finishing time after rescheduling shall not surpass its predefined due date. (3) Machine sequence constraint: Within the same processing unit, the completion time of the shifted job must not go beyond the start time of the subsequent job, thereby preserving the processing order. (4) Stage precedence constraint: The time point when the job completes the present stage must not exceed the start time of its operation in the next stage, ensuring inter-stage precedence feasibility. Only when all the above conditions are satisfied can the job be shifted rightward. The right-shift operation is conducted within each stage independently, meaning that adjustments at one stage do not delay the finishing time of the job in its preceding stage. Therefore, the inter-stage processing sequence remains valid. By iteratively applying this right-shift mechanism, idle intervals on machines are effectively compressed or eliminated. As a consequence, the duration during which machines remain powered but unproductive is reduced, leading directly to lower idle energy consumption. Importantly, this strategy does not increase the overall makespan of the schedule; instead, it redistributes operations within existing slack time. Thus, the proposed method improves energy efficiency while maintaining the original production completion time and schedule feasibility. For example, in Figure 6, the gantte chart is the second factory in subsection 3.3 that right-shift strategy can be effectively utilized. The processing operation of j_2 in the second stage can be right-shifted, provided that all constraints are satisfied, so that its completion

 doi:10.20944/preprints202603.1709.v1

time becomes contiguous with the start time of j_3. In this way, machine idle time is eliminated and energy consumption is reduced. The amount of energy saved is calculated as: $2 * (16 - 13) = 6$ kW.

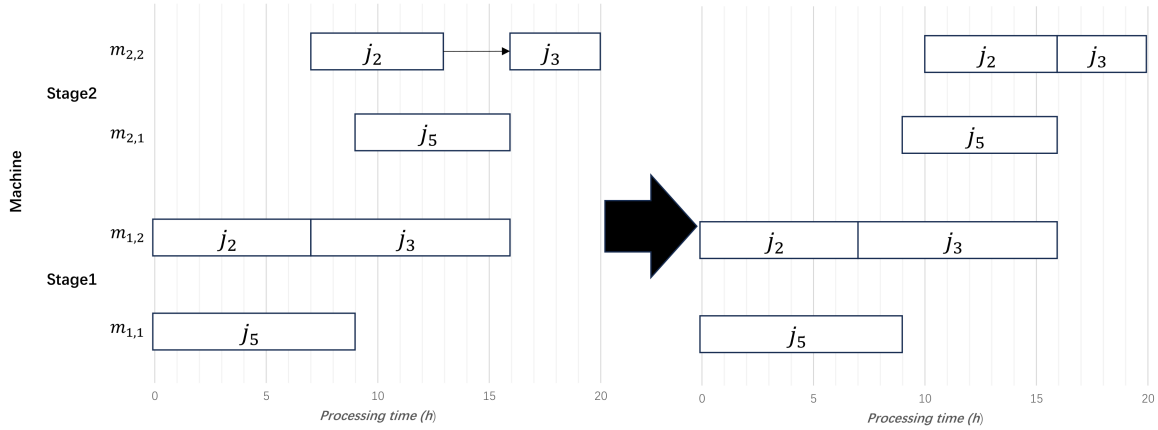

Figure 6. Energy-saving stategy.

5. Experimental Outcomes and Analysis

The DBCEA is implemented in PyCharm with Python 3.9 and CUDA 11.8, running on a Windows 11 environment. Computational experiments are performed on a workstation with an AMD Ryzen 7 6500H 3.2?GHz CPU, 16?GB RAM, and an NVIDIA GeForce RTX 3060 GPU. For consistency, on each benchmark instance,each algorithm is executed for 20 times and the mean values of the performance metrics are used as the reported results. The benchmark comes from Li [32]. In data benchmark, job numbers $n \in (20, 40, 60, 80, 100)$, $n_s \in (3, 5)$, and factory numbers $f \in (2, 3)$. The processing and idle energy consumption were fixed at $EC_p = 4.0$ kW/h and $EC_d = 1.0$ kW/h, respectively. Instances are labeled as aJbScF, where a, b and c indicate the sum of all jobs in one instance, the total stage numbers in each factory, and the total factory quantities.

5.1. Assessment Criteria

Assessment of the proposed DBCEA utilized the hypervolume (HV) and inverted generational distance (IGD) metrics. Prior to computing HV and IGD, the nondominated solutions were normalized according to the following procedure

$$x_i^{\text{norm}} = \frac{x_i - z_i^{\min}}{z_i^{\max} - z_i^{\min}}, i = 1, 2 \tag{16}$$

where $z_i^{\min}$ and $z_i^{\max}$ indicating the lower and upper bounds of the ith objective across the Pareto front.

(1) HV metric:

$$HV = V(\bigcup_{i=1}^{P} V_i) \tag{17}$$

The hypervolume (HV) measures spatial extent within the multi-objective space bounded by the nondominated solution set and a reference point. Let P denote the set of nondominated solutions, and V_i denotes the hypercube associated with each solution in P and the reference point (i.e., the maximum value). The solution set was normalized in this approach, and the reference point was set to (1,1). Higher hypervolume values denotes that the algorithm performs more effectively.

(2) IGD metric:

$$IGD(P, P^{'}) = \frac{1}{|P'|} \sum_{x \in P'} \min_{y \in P} d(x, y) \tag{18}$$

 doi:10.20944/preprints202603.1709.v1

The inverted generational distance (IGD) calcaulate mean Euclidean distance from the reference Pareto front to the obtained solutions. Let P be the acquired nondominated solution set, with $P^{'}$ indicating the reference Pareto front, and $d(x, y)$ the Euclidean distance between a reference point $x \in P$ and and a solution $y \in P$. A smaller IGD value indicates a closer approximation to the true Pareto front and, therefore, a higher-quality solution set.

5.2. Parameter Setting in DBCEA

In this section, we discuss the impact of different parameters combinations for algorithm performance. HV is adopted as performance metric. There are three parameters need to be determined:population size ps, crossover rate P_c, mutation rate P_m, The three parameters are determined using a design-of-experiment (DOE) approach [33], each parameter has three levels, with the parameter levels configured as shown below:

- $ps = 80, 100, 120$
- $P_c = 0.8, 0.9, 1.0$
- $P_m = 0.1, 0.2, 0.3$

A numerical orthogonal array $L_9(3^4)$ is executed for the experiment, Table 3 has shown the numberical result. To visually assess individual contribution of parameters individually affects the performance metrics, factor-level effects of the parameters are visualized in the plots, the result has shown in Figure 7. So the parameters selection can be determined as $ps = 100, P_c = 0.9, P_m = 0.1$.

Table 3. Levels of parameters.

Trial	Level			HV
	ps	P_c	P_m	
1	1	1	1	0.4932
2	1	2	2	0.3542
3	1	3	3	0.3931
4	2	1	2	0.4160
5	2	2	3	0.3854
6	2	3	1	0.6164
7	3	1	3	0.3138
8	3	2	1	0.3249
9	3	3	2	0.4283

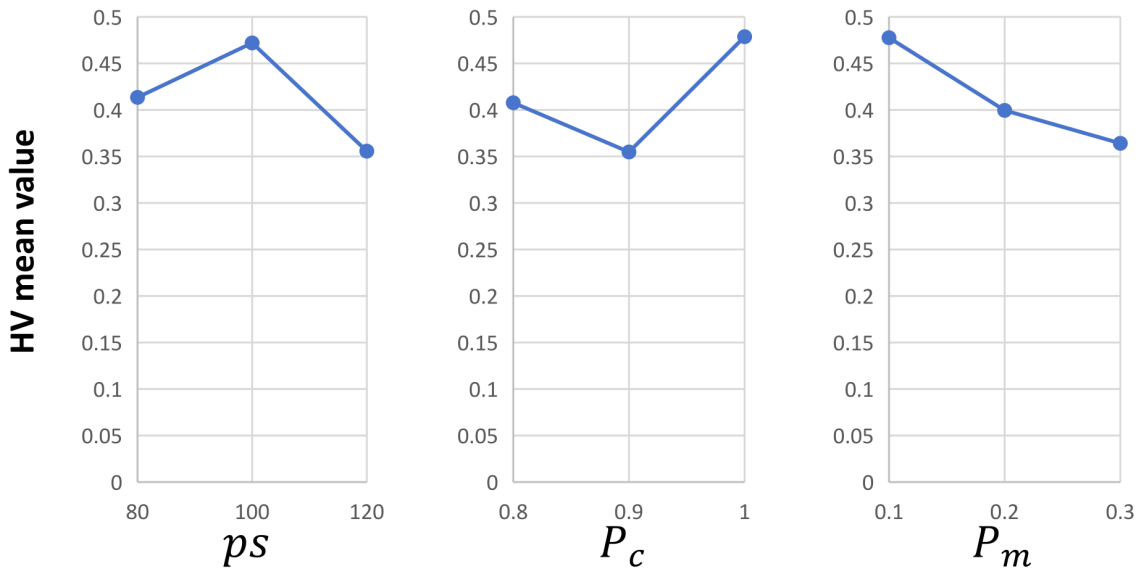

Figure 7. Parameter levels

 doi:10.20944/preprints202603.1709.v1

5.3. Assessment of the Individual Strategies Within DBCEA

This section presents a comparison of different experimental settings to assess the performance of each strategy. Four experiments were designed according to the corresponding improvements and are labeled as EX_i, $i \in \{1, 2, 3, 4\}$, The EX_1 denotes the DBCEA without initialization strategy, and all individuals are randomly generated. The EX_2 represents the DBCEA without external archive. The EX_3 means the DBCEA without the local search stategy and The EX_4 remove the energy-saving stategy.

Table 4 shows the numerical experimental results, and the best results are marked in bold. It can be clearly seen that DBCEA has achieved the best results in most benchmarks. Table 5 reports the outcomes of the Friedman rank-sum test conducted at a significance level of $\alpha = 0.05$. Based on these results, it will be inferred that a p-value less than 0.05 indicates that DBCEA significantly outperforms all tested variants, confirming the efficacy of the strategies employed within the algorithm.

Table 4. Mean HV and IGD values obtained for all variants across all benchmarks.

	HV				
Benchmarks	EX_1	EX_2	EX_3	EX_4	DBCEA
20J3S2F	0.7134	0.5840	0.7036	0.6627	**0.7940**
20J3S3F	0.6369	0.6593	0.6936	0.6080	**0.7681**
20J5S2F	0.5928	0.6828	0.5960	0.5634	**0.6831**
20J5S3F	0.7193	0.6276	0.6545	0.6326	**0.7389**
40J3S2F	0.6633	0.5596	0.5397	0.7225	**0.8009**
40J3S3F	0.6560	0.6207	0.6496	0.6221	**0.7395**
40J5S2F	0.5750	0.5871	0.5031	0.6217	**0.7475**
40J5S3F	0.6107	0.4148	0.5564	0.5951	**0.8497**
60J3S2F	0.5949	0.5547	0.5837	0.5539	**0.6118**
60J3S3F	0.6048	0.6157	0.6276	0.5913	**0.6768**
60J5S2F	0.6870	0.6599	0.5011	0.5545	**0.7157**
60J5S3F	0.6748	0.5634	0.6807	0.5310	0.7040
80J3S2F	0.6302	0.6219	0.6003	0.6785	**0.7116**
80J3S3F	**0.7589**	0.6117	0.6150	0.6632	0.7311
80J5S2F	0.6906	0.4768	0.6237	0.6953	**0.7502**
80J5S3F	0.5408	**0.6548**	0.6200	0.6011	0.6313
100J3S2F	0.5597	0.6058	0.5613	0.5520	**0.6131**
100J3S3F	0.6254	0.6237	0.5806	0.7019	**0.7578**
100J5S2F	0.6525	0.5892	0.5323	0.5618	**0.6978**
100J5S3F	0.6922	0.6495	0.6505	0.5883	**0.7381**
	HV				
Benchmarks	EX_1	EX_2	EX_3	EX_4	DBCEA
20J3S2F	0.1429	0.2034	0.1514	0.1475	**0.1092**
20J3S3F	0.1575	0.1634	0.1121	0.1410	**0.0511**
20J5S2F	0.2648	0.2436	0.2419	0.2382	**0.2105**
20J5S3F	0.1429	0.1807	0.1563	0.1302	**0.1147**
40J3S2F	0.1789	0.2025	0.1580	0.1479	**0.1083**
40J3S3F	0.1648	0.2141	0.2383	0.2077	**0.1054**
40J5S2F	0.1352	0.1648	0.2144	0.1230	**0.0654**
40J5S3F	**0.0850**	0.1644	0.1178	0.1713	0.1006
60J3S2F	0.1554	0.0846	0.1402	0.1141	**0.0818**
60J3S3F	0.2455	0.1940	0.1058	0.1524	**0.0841**
60J5S2F	0.2523	0.2349	0.2315	0.1990	**0.1473**
60J5S3F	0.1596	0.1436	**0.1272**	0.1924	0.1389
80J3S2F	0.1887	0.2919	0.1304	0.1642	**0.0976**
80J3S3F	0.1248	0.1480	0.1369	0.1907	**0.1075**
80J5S2F	0.2905	0.2189	0.2347	0.2307	**0.1709**
80J5S3F	0.1982	0.2352	0.1971	0.2828	**0.1384**
100J3S2F	0.2681	0.2066	0.1257	0.2183	**0.1117**
100J3S3F	0.2235	0.2123	0.1895	**0.1066**	0.1292
100J5S2F	0.0919	0.1392	0.1793	0.1512	**0.0416**
100J5S3F	0.1592	0.2846	0.1714	0.2128	**0.1060**

 doi:10.20944/preprints202603.1709.v1

Table 5. Friedman test and associated statistical values for DBCEA variants (α = 0.05).

	HV			IGD		
Algorithms	**mean value**	**rank**	**p-value**	**mean value**	**rank**	**p-value**
EX_1	0.6641	2.85	3.6464E-08	0.1815	3.55	1.4074E-07
EX_2	0.6002	3.70		0.1965	3.95	
EX_3	0.6036	3.60		0.1680	3.15	
EX_4	0.6150	3.75		0.1761	3.20	
DBCEA	**0.7231**	**1.10**		**0.1109**	**1.15**	

Table 6. Mean HV and IGD values obtained for all algorithm across all benchmarks.

	HV				
Benchmarks	NSGA-II	MOEA/D	ENSGA-II	D2QCE	DBCEA
20J3S2F	0.6374	0.5796	0.7633	0.7814	**0.8120**
20J3S3F	0.5611	0.6229	0.6603	0.7206	**0.7681**
20J5S2F	0.6700	0.5990	0.6517	0.7239	**0.7418**
20J5S3F	0.6881	0.6823	0.7575	0.7466	**0.8743**
40J3S2F	0.6342	0.6403	0.7899	0.7626	**0.8124**
40J3S3F	0.7130	0.6827	0.7119	0.6809	**0.8181**
40J5S2F	0.6492	0.5301	0.6439	0.6676	**0.8032**
40J5S3F	0.6991	0.5211	0.7178	0.7022	**0.7612**
60J3S2F	0.6586	0.6407	**0.7466**	0.6556	0.7311
60J3S3F	0.7071	0.5704	0.6852	0.6969	**0.7770**
60J5S2F	0.6924	0.6722	0.7152	0.6137	**0.8220**
60J5S3F	0.7037	0.6184	0.6848	0.7678	**0.7959**
80J3S2F	0.7504	0.7453	0.6663	0.5895	**0.7624**
80J3S3F	0.6345	0.6828	0.8017	**0.8718**	0.8107
80J5S2F	0.7295	0.5938	0.7608	0.7837	**0.8214**
80J5S3F	0.6621	0.5858	0.6568	0.7633	**7719**
100J3S2F	0.7863	0.5972	0.6638	0.5123	**0.8164**
100J3S3F	0.6987	0.7554	0.8160	0.7685	**0.8209**
100J5S2F	0.7538	0.5261	0.6868	0.4613	**0.8519**
100J5S3F	0.6947	0.6708	0.6339	0.6297	**0.7862**
	HV				
Benchmarks	NSGA-II	MOEA/D	ENSGA-II	D2QCE	DBCEA
20J3S2F	0.1004	0.2247	0.0536	0.0772	**0.0223**
20J3S3F	0.1095	0.1017	0.0804	0.0793	**0.0654**
20J5S2F	0.1772	0.2365	0.0665	**0.0453**	0.0489
20J5S3F	0.0905	0.1204	0.0693	0.0621	**0.0391**
40J3S2F	0.1443	0.1074	0.0624	0.0827	**0.0528**
40J3S3F	0.1263	0.1819	**0.0410**	0.0692	0.0580
40J5S2F	0.1180	0.1587	0.0726	0.0951	**0.0534**
40J5S3F	0.1191	0.2377	0.0632	0.0744	**0.0411**
60J3S2F	0.1519	0.1464	0.0794	0.0587	**0.0511**
60J3S3F	0.1358	0.1921	**0.0506**	0.0558	0.0731
60J5S2F	0.1425	0.1128	0.0677	0.1074	**0.0392**
60J5S3F	0.1439	0.2618	0.0664	0.0784	**0.0627**
80J3S2F	0.1822	0.2253	0.1187	0.0922	**0.0753**
80J3S3F	0.2291	0.2122	0.1082	0.1285	**0.0786**
80J5S2F	0.0861	0.1760	0.1478	0.0921	**0.0603**
80J5S3F	0.1875	0.1420	0.0975	0.1051	**0.0576**
100J3S2F	0.1073	0.2126	0.1197	0.1760	**0.1031**
100J3S3F	0.1621	0.2059	0.0847	0.0858	**0.0430**
100J5S2F	0.1316	0.1825	0.0892	0.0789	**0.0412**
100J5S3F	0.1452	0.1358	0.1142	0.0831	**0.0625**

 doi:10.20944/preprints202603.1709.v1

Table 7. Friedman test and associated statistical values for all algorithm ($\alpha = 0.05$)

	HV			IGD		
Algorithms	**mean value**	**rank**	**p-value**	**mean value**	**rank**	**p-value**
NSGA-II	0.6862	3.30	1.3782E-08	0.1395	4.10	7.2010E-12
MOEA/D	0.6258	4.40		0.1787	4.65	
ENSGA-II	0.7107	3.00		0.0827	2.40	
D2QCE	0.6950	3.10		0.0864	2.50	
DBCEA	**0.7993**	**1.10**		**0.0564**	**1.35**	

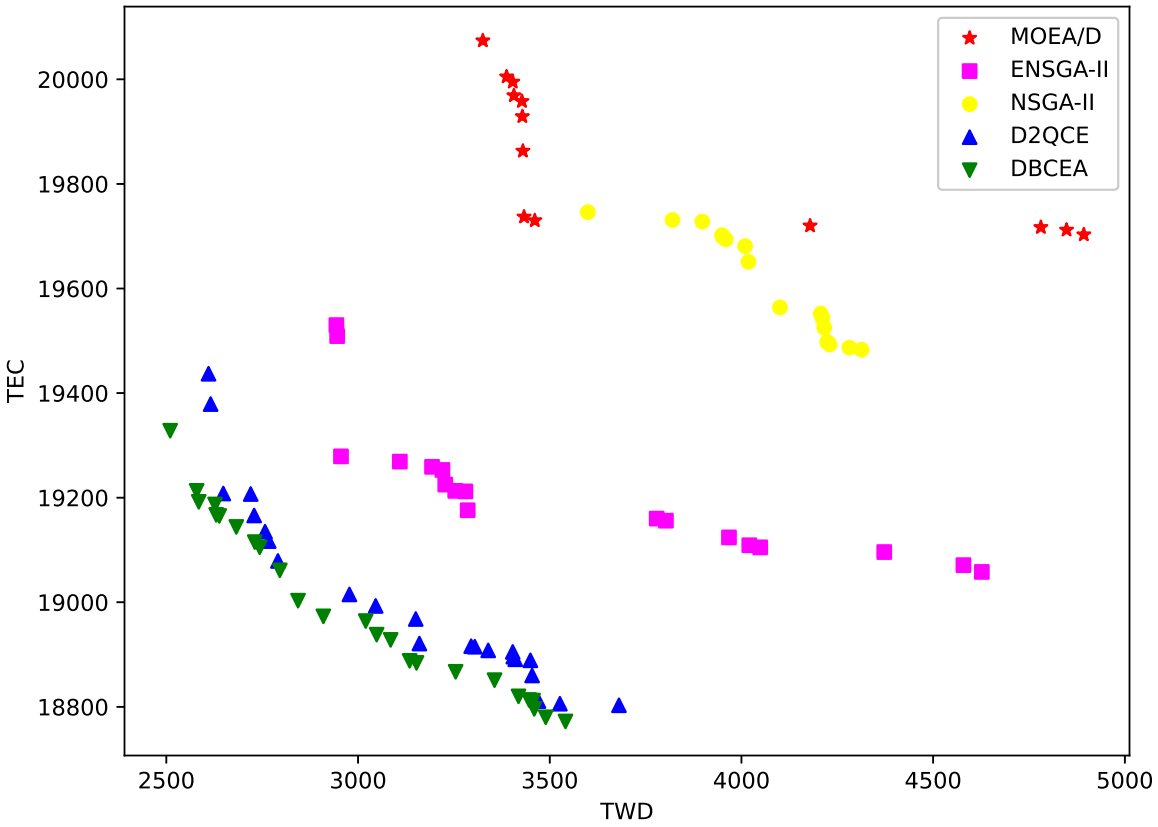

Figure 8. Pareto front result comparison on 40J5S2F

5.4. Comparative Analysis with Other Methods

In this section, to further verify the effectiveness of DBCEA, five state-of-art algorithms are used for comparison, which are divided to two parts. The first part contains the classical multiobjective meta-heuristics algorithms: NSGA-II [34] and MOEA/D [35], corresponding to multi-objective optimization algorithms utilizing pareto dominance and multi-objective decomposition strategies, respectively. The second part contains two recently proposed algorithms:ENSGA-II [19], D2QCE [32]. For performance evaluation, all algorithms were executed using an identical population size($ps = 100$), crossover rate($P_c = 1.0$), mutation rate($P_m = 0.1$), while all other parameters were retained as specified in the respective original publications.

For each compared algorithm, detailed comparisons were based on the average performance across 20 independent runs, Table 6 presents the results of the experiments. The best result is highlighted in bold. Table 7 summarizes the outcomes of the Friedman rank-sum test conducted among all algorithms, using a confidence level of $\alpha = 0.05$.

As observed in Table 6, DBCEA achieve the best performance for HV except two instances. In regard of IGD, DBCEA achieve the best performance for seventeen instances. This indicates that the DBCEA is capable of producing a superior solution set.

 doi:10.20944/preprints202603.1709.v1

Moreover, the Friedman rank-sum test was employed to assess superiority of the DBCEA. Since each algorithm was executed 20 times per benchmark instance, 20 Friedman tests were conducted. Table 7 presents the average statistical values, ranks, and p-value, within p-value < 0.05 indicating significant differences among the algorithms. The DBCEA consistently demonstrated superior performance in terms of HV and IGD, demonstrating superior performance across most benchmark instances.

Figure 8 illustrates the Pareto fronts generated by the different algorithms on the 40J5S2F. The green pentagram represents the solution set generated by DBCEA. This indicates that the DBCEA solutions are closest to the coordinate origin, suggesting that it yields a higher-quality solution set than other competing algorithms.

6. Conclusion

In this work, we introduced a decomposition-based coevolutionary algorithm to solve distributed heterogeneous hybrid flowshop scheduling with job deadlines and priorities. An initialization strategy with a randomly sized population is adopted to generate high-quality initial solutions. Subsequently, a dedicated energy-saving strategy was used to further reduce overall energy consumption. In the end, a selection strategy utilizing elite solutions based on the integration of an external archive and an elite archive and four problem-based operator selection strategies embedded in a cooperative local search framework to get solution set.

A set of experiments was performed to assess the effectiveness of the DBCEA. To assess the contribution of its enhanced strategies, the DBCEA was tested against multiple variants combining different improvements. Additionally, the algorithm was benchmarked against several well-established methods across 20 instances of varying scale, consistently demonstrating superior performance and producing a more favorable Pareto front.

Future research may address more intricate shop environments, incorporating factors such as dynamic events, machine speed variations, and heterogeneous parallel resources, which would further broaden the practical applications of this methodology in scheduling problems.

Author Contributions: Conceptualization, H.X., L.H., J.Z. and J.T.; methodology, H.X., L.H., and C.Z.; software, H.X., L.H. and C.Z.; validation, H.X., L.H. and J.Z.; formal analysis, L.H. and J.T.; writing—original draft preparation, H.X. and L.H.; writing—review and editing, J.T., J.Z. and C.Z.; visualization, H.X. and L.H.; supervision, J.T., J.Z. and C.Z.; project administration, H.X. All authors have read and agreed to the published version of the manuscript.

Funding: The authors report that this work was conducted without any financial support, grants, or external assistance.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Due to confidentiality restrictions, the data supporting the findings of this study are not publicly accessible, however it can be requested from the corresponding author subject to authorization.

Conflicts of Interest: The authors declare that there is no conflict of interest or competing interests.

References

1. Adel, A. Future of industry 5.0 in society: human-centric solutions, challenges and prospective research areas. *JOURNAL OF CLOUD COMPUTING-ADVANCES SYSTEMS AND APPLICATIONS* **2022**, *11*. https://doi.org/10.1186/s13677-022-00314-5.
2. Arda, Y.; Cattaruzza, D.; François, V.; Ogier, M. Home chemotherapy delivery: An integrated production scheduling and multi-trip vehicle routing problem. *European Journal of Operational Research* **2024**, *317*, 468–486. https://doi.org/https://doi.org/10.1016/j.ejor.2024.03.039.

 doi:10.20944/preprints202603.1709.v1

3. Lin, C.C.; Peng, Y.C.; Chang, Y.S.; Chang, C.H. Reentrant hybrid flow shop scheduling with stockers in automated material handling systems using deep reinforcement learning. *Computers & Industrial Engineering* **2024**, *189*, 109995. https://doi.org/https://doi.org/10.1016/j.cie.2024.109995.
4. Liu, Y.; Han, W.; Su, X.; Cui, R. Optimization of fixed aviation support resource station configuration for aircraft carrier based on aircraft dispatch mission scheduling. *CHINESE JOURNAL OF AERONAUTICS* **2023**, *36*, 127–138. https://doi.org/10.1016/j.cja.2022.06.023.
5. Zhao, F.; He, X.; Wang, L. A Two-Stage Cooperative Evolutionary Algorithm With Problem-Specific Knowledge for Energy-Efficient Scheduling of No-Wait Flow-Shop Problem. *IEEE TRANSACTIONS ON CYBERNETICS* **2021**, *51*, 5291–5303. https://doi.org/10.1109/TCYB.2020.3025662.
6. Li, J.q.; Sang, H.y.; Han, Y.y.; Wang, C.g.; Gao, K.z. Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions. *JOURNAL OF CLEANER PRODUCTION* **2018**, *181*, 584–598. https://doi.org/10.1016/j.jclepro.2018.02.004.
7. Neufeld, J.S.; Schulz, S.; Buscher, U. A systematic review of multi-objective hybrid flow shop scheduling. *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* **2023**, *309*, 1–23. https://doi.org/10.1016/j.ejor.2022.08.009.
8. Li, Y.Z.; Pan, Q.K.; Gao, K.Z.; Tasgetiren, M.F.; Zhang, B.; Li, J.Q. A green scheduling algorithm for the distributed flowshop problem. *APPLIED SOFT COMPUTING* **2021**, *109*. https://doi.org/10.1016/j.asoc.2021.107526.
9. Liu, R.; Piplani, R.; Toro, C. Deep reinforcement learning for dynamic scheduling of a flexible job shop. *Int. J. Prod. Res.* **2022**, *60*, 4049–4069. https://doi.org/10.1080/00207543.2022.2058432.
10. García-Gómez, P.; Rodríguez, I.G.; Vela, C.R. Enhanced memetic search for reducing energy consumption in fuzzy flexible job shops. *Integr. Comput. Aided Eng.* **2023**, *30*, 151–167. https://doi.org/10.3233/ICA-230699.
11. Chen, S.; Wang, X.; Wang, Y.; Gu, X. A knowledge-driven many-objective algorithm for energy-efficient distributed heterogeneous hybrid flowshop scheduling with lot-streaming. *SWARM AND EVOLUTIONARY COMPUTATION* **2024**, *91*. https://doi.org/10.1016/j.swevo.2024.101771.
12. He, L.; Sun, S.; Luo, R. A hybrid two-stage flowshop scheduling problem. *ASIA-PACIFIC JOURNAL OF OPERATIONAL RESEARCH* **2007**, *24*, 45–56. https://doi.org/10.1142/S0217595907001036.
13. Fan, J.; Li, Y.; Xie, J.; Zhang, C.; Shen, W.; Gao, L. A Hybrid Evolutionary Algorithm Using Two Solution Representations for Hybrid Flow-Shop Scheduling Problem. *IEEE TRANSACTIONS ON CYBERNETICS* **2023**, *53*, 1752–1764. https://doi.org/10.1109/TCYB.2021.3120875.
14. Wang, Y.J.; Wang, G.G.; Tian, F.M.; Gong, D.W.; Pedrycz, W. Solving energy-efficient fuzzy hybrid flow-shop scheduling problem at a variable machine speed using an extended NSGA-II. *ENGINEERING APPLICATIONS OF ARTIFICIAL INTELLIGENCE* **2023**, *121*. https://doi.org/10.1016/j.engappai.2023.105977.
15. Qiao, J.; He, M.; Sun, N.; Sun, P.; Fan, Y. Factors affecting the final solution of the bike-sharing rebalancing problem under heuristic algorithms. *COMPUTERS & OPERATIONS RESEARCH* **2023**, *159*. https://doi.org/10.1016/j.cor.2023.106368.
16. Nicosia, G.; Pacifici, A.; Pferschy, U.; Russo Russo, A.; Salvatore, C. Flow shop scheduling with inter-stage flexibility and blocking constraints. *Computers & Operations Research* **2025**, *184*, 107219. https://doi.org/https://doi.org/10.1016/j.cor.2025.107219.
17. Wang, X.; Liu, C.; Wang, R.; Yu, Z.; Yang, S. Improved Genetic Algorithm Using Reinforcement Learning to Solve the Re-entrant Flexible Flow Shop Scheduling Problem. In Proceedings of the 2025 IEEE Congress on Evolutionary Computation (CEC), 2025, pp. 1–8. https://doi.org/10.1109/CEC65147.2025.11043111.
18. Wang, W.; Zhang, B.; Jiang, X.; Jia, B.; Sang, H.; Meng, L. Decomposition-based multi-objective approach for a green hybrid flowshop rescheduling problem with consistent sublots. *International Journal of Production Research* **2024**, *62*, 7904–7932, [https://doi.org/10.1080/00207543.2024.2333943]. https://doi.org/10.1080/00207543.2024.2333943.
19. Wang, Y.J.; Wang, G.G.; Tian, F.M.; Gong, D.W.; Pedrycz, W. Solving energy-efficient fuzzy hybrid flow-shop scheduling problem at a variable machine speed using an extended NSGA-II. *ENGINEERING APPLICATIONS OF ARTIFICIAL INTELLIGENCE* **2023**, *121*. https://doi.org/10.1016/j.engappai.2023.105977.
20. Li, C.; Han, Y.; Zhang, B.; Wang, Y.; Li, J.; Gao, K. A novel multi-objective hybrid evolutionary algorithm based on variable weight strategy for distributed hybrid flowshop scheduling with batch processing machines and variable sublots. *Applied Soft Computing* **2025**, *170*, 112650. https://doi.org/https://doi.org/10.1016/j.asoc.2024.112650.

 doi:10.20944/preprints202603.1709.v1

21. Wang, J.J.; Wang, L. A Bi-Population Cooperative Memetic Algorithm for Distributed Hybrid Flow-Shop Scheduling. *IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE* **2021**, *5*, 947–961. https://doi.org/10.1109/TETCI.2020.3022372.
22. Zheng, J.; Wang, L.; Wang, J.j. A cooperative coevolution algorithm for multi-objective fuzzy distributed hybrid flow shop. *KNOWLEDGE-BASED SYSTEMS* **2020**, *194*. https://doi.org/10.1016/j.knosys.2020.105536.
23. Zhang, G.; Liu, B.; Wang, L.; Yu, D.; Xing, K. Distributed Co-Evolutionary Memetic Algorithm for Distributed Hybrid Differentiation Flowshop Scheduling Problem. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION* **2022**, *26*, 1043–1057. https://doi.org/10.1109/TEVC.2022.3150771.
24. Wang, B.; Wang, H.; Yan, Q.; Ma, E. Multi-agent reinforcement learning-aided evolutionary algorithm for a many-objective distributed hybrid flow shop scheduling problem. *Swarm and Evolutionary Computation* **2025**, *97*, 101991. https://doi.org/https://doi.org/10.1016/j.swevo.2025.101991.
25. Hao, H.; Zhu, H.; Luo, Y. A multi-objective Immune Balancing Algorithm for Distributed Heterogeneous Batching-integrated Assembly Hybrid Flowshop Scheduling. *Expert Systems with Applications* **2025**, *259*, 125288. https://doi.org/https://doi.org/10.1016/j.eswa.2024.125288.
26. Shao, W.; Shao, Z.; Pi, D. An Ant Colony Optimization Behavior-Based MOEA/D for Distributed Heterogeneous Hybrid Flow Shop Scheduling Problem Under Nonidentical Time-of-Use Electricity Tariffs. *IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING* **2022**, *19*, 3379–3394. https://doi.org/10.1109/TASE.2021.3119353.
27. Qin, H.X.; Han, Y.Y. A collaborative iterative greedy algorithm for the scheduling of distributed heterogeneous hybrid flow shop with blocking constraints. *EXPERT SYSTEMS WITH APPLICATIONS* **2022**, *201*. https://doi.org/10.1016/j.eswa.2022.117256.
28. Vasquez, O.C. On the complexity of the single machine scheduling problem minimizing total weighted delay penalty. *OPERATIONS RESEARCH LETTERS* **2014**, *42*, 343–347. https://doi.org/10.1016/j.orl.2014.05.009.
29. Gao, K.Z.; Suganthan, P.N.; Pan, Q.K.; Tasgetiren, M.F. An effective discrete harmony search algorithm for flexible job shop scheduling problem with fuzzy processing time. *INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH* **2015**, *53*, 5896–5911. https://doi.org/10.1080/00207543.2015.1020174.
30. Zuo, Y.; Zhao, F.; Zhang, J. A bi-population cooperative scatter search algorithm for distributed hybrid flow shop scheduling with machine breakdown. *Computers & Industrial Engineering* **2024**, *197*, 110624. https://doi.org/https://doi.org/10.1016/j.cie.2024.110624.
31. Liu, H.; Zhao, F.; Wang, L.; Xu, T.; Dong, C. Evolutionary Multitasking Memetic Algorithm for Distributed Hybrid Flow-Shop Scheduling Problem With Deterioration Effect. *IEEE Transactions on Automation Science and Engineering* **2025**, *22*, 1390–1404. https://doi.org/10.1109/TASE.2024.3365518.
32. Li, R.; Gong, W.; Wang, L.; Lu, C.; Pan, Z.; Zhuang, X. Double DQN-Based Coevolution for Green Distributed Heterogeneous Hybrid Flowshop Scheduling With Multiple Priorities of Jobs. *IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING* **2024**, *21*, 6550–6562. https://doi.org/10.1109/TASE.2023.3327792.
33. Reynolds, P.S. Between two stools: preclinical research, reproducibility, and statistical design of experiments. *BMC RESEARCH NOTES* **2022**, *15*. https://doi.org/10.1186/s13104-022-05965-w.
34. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION* **2002**, *6*, 182–197. https://doi.org/10.1109/4235.996017.
35. Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION* **2007**, *11*, 712–731. https://doi.org/10.1109/TEVC.2007.892759.