Computer Fraud and Security
ISSN (online): 1873-7056

Improved Artificial Hummingbird Algorithm for Solving Energy-Efficient Scheduling Problem of Distributed Hybrid Flow Shop

Ziqang Qiu[1], Hanning Chen[2,3,*], Keyi Liu[4], Yelin Xia[2], Zhixue Wang[5], Haifeng Xue[6,7]

[1]*School of Artificial Intelligence, Tiangong University, Tianjin 300387, China*

[2]*College of Artificial Intelligence, Tianjin University of Science and Technology, Tianjin 300457, China*

[3]*School of Computer Science and Technology, Tiangong University, Tianjin 300387, China*

[4]*School of mechanical and electrical engineering, Xinjiang Institute of Engineering, Wulumuqi 830023, Xinjiang, China*

[5]*School of Control Science and Engineering, Tiangong University, Tianjin 300387, China*

[6]*Engineering Research Center of Integration and Application of Digital Learning Technology, Ministry of Education. Beijing 100039, China*

[7]*Engineering Research Center of Integration and Application of Digital Learning Technology, Ministry of Education.2. No.2 Weigongcun Road, Haidian District, Beijing 100039, China*

**Corresponding Author.*

Abstract:

To tackle the challenges of time efficiency and energy consumption in the context of production diversification and green manufacturing, GDMOAHA was developed to minimize the makespan and total energy consumption. First, a problem-specific heuristic, NEH-SE, was designed to improve the initial population's diversity and quality. In addition, the golden sine strategy has been implemented to achieve a good balance between exploration and exploitation. Third, the inclusion of the Cauchy variation enhances the algorithm's ability to find optimal values. Compared with other algorithms, GDMOAHA outperforms other algorithms in terms of convergence and diversity. Therefore, it can solve the problem well.

Keywords: distributed flow shop, energy efficient scheduling, improved multi-objective artificial hummingbird algorithm, multi-objective optimization problem

INTRODUCTION

Over the past few years, the idea of sustainable manufacturing has attracted considerable focus, the integration of production scheduling with green manufacturing has emerged as a hot topic. Manufacturing is an important sector in terms of energy consumption, accounting for a significant portion of the total global energy use each year [1]. Effectively reducing energy consumption and improving energy utilization has gradually become the goal for an increasing number of countries and enterprises. The hybrid flow-shop scheduling problem (HFSP) is a classical problem model in the manufacturing industry. Zhang B et al. broaden the scope of energy-efficient HFSP by examining machines that have varying energy consumption rates, sequence-dependent setups, and inter-machine transport operations within the HFSP, and suggests a three-stage multi-objective approach based on decomposition (TMOA/D) to solve this problem [2]. Qin et al. investigated the Blocking Hybrid Flow Shop Group Scheduling Problem (BHFGSP) and introduced an iterative greedy algorithm designed to optimize the makespan. [3]. Wang Z et al. investigated the HFSP rescheduling problem under machine failure and introduced an improved multi-objective firefly algorithm to minimize productivity, power efficiency and manufacturing reliability [4]. Teng Y et al. proposed an improved co-evolutionary memetic algorithm (ICMA) based on the unconditional feasibility of HFSP (UFH) and all-active schedule (AAS) to solve HFSP [5]. The HFSP has been well-studied, and many results have been obtained.

Today, when facing uncertain market conditions and fluctuating customer demands, a single-factory production model is hardly sufficient to meet production needs. A large number of companies have shifted their manufacturing model from a single plant to having a geographically dispersed multi-plant supply chain. Distributed production planning and scheduling has been widely adopted such as semiconductor manufacturing, automotive industry, pharmaceutical industry, concrete industry, textile industry [6]. The Distributed Hybrid Flow-Shop Scheduling Problem (DHFSP) adheres to the distributed production paradigm and serves as an extended problem of the HFSP. DHFSP includes the task of allocating jobs to suitable factories. Multiple factories work together to produce a set of jobs. The HFSP model is used on the shop floor of each factory. When a job is designated to a particular factory, it implies that all processing stages of the job must be executed exclusively within the factory. Consequently, the DHFSP encompasses a broader set of constraints and presents a more intricate set of research challenges than the traditional HFSP. The majority of scholars currently utilize intelligent optimization algorithms to address the problem of DHFSP. Li Y et al. enhanced the Discrete Artificial Bee Colony (DABC) algorithm to minimize the makespan DHFSP. [7]. Cai J et al. introduced an innovative version of the shuffled frog-leaping algorithm, enhanced with memeplex quality (MQSFLA), aimed at reducing both total tardiness and makespan in DHFSP, considering the impact of sequence-dependent setup times [8]. Wang J and Wang L proposed a cooperative modular algorithm (CMA) based on reinforcement learning (RL) to solve energy-aware DHFSP [9]. Later, Wang J and Wang L further proposed a Collaborative Modelling Algorithm (CMA) with a local reinforcement strategy incorporating cooperative search and Q-learning to optimize the Energy-Aware Distributed Welding Shop Scheduling problem (EADFFASP) [10]. Cui H et al. proposed an Improved Multi-Population Genetic Algorithm (IMPGA) to optimize makespan for the Distributed Heterogeneous HFSP [11]. R. Li et al. tackled the DHHFSP with multiple work priorities by developing a dual deep Q-network based co-evolutionary (D2QCE) approach, aiming to minimize both the total weighted delay and total energy consumption [12]. Lu C et al. formulated a mixed integer linear programming (MILP) model for DHFSP. Subsequently, an Improved Iterative Greedy (IIG) algorithm was devised to address this DHFSP [13]. Later Lu C et al. made the first attempt to study the energy-aware DHFSP. A hybrid multi-objective iterative greedy (HMOIG) approach was proposed to minimize the manufacturing span and total energy consumption [14]. In summary, numerous studies have been conducted on the DHFSP in the past, yet research on energy-efficient hybrid flow shops with different numbers of machines across different plants remains sparse. The development of efficient algorithms for optimizing makespan and total energy consumption (TEC) in this context presents a challenging and significant task.

MOAHA identifies suitable solutions by modeling hummingbirds' flight and foraging strategies. MOAHA has demonstrated effectiveness in solving optimization problems [15-17]. The 'no free lunch' theorem posits that optimization algorithms have different application scenarios [18]. It is important to choose the right optimization algorithm for different optimization problems. Compared with other algorithms, MOAHA requires less tuning of parameters and demonstrates strong adaptability across different scenarios. MAOHA improves its ability to explore and exploit by having three unique flight skills and employing three distinct foraging strategies. Furthermore, it incorporates access tables into the iteration process, continuously updating the information regarding hummingbird visits to food sources. It assists hummingbirds in locating suitable food sources [15]. Therefore, considering these factors, MOAHA differs significantly from existing algorithms. However, MOAHA has not yet been applied to scheduling problems. In this paper, the Golden Sine Multi-Objective Artificial Hummingbird Algorithm (GDMOAHA) is proposed based on MOAHA and successfully applied to solve the scheduling problem.

The main contributions of this paper are summarized below:

(1) To solve the multi-objective DHFSP model that describes energy consumption in practical manufacturing, a GDMOAHA with hybrid initialization and golden sine is designed specifically.

(2) Unlike random initialization adopted by most multi-objective algorithms, a hybrid initialization method with NEH-SE and random is designed to produce high-quality populations.

(3) To overcome the problem of distance from food sources at the initial stage of MOAHA, the golden sine that can reduce search space is introduced in GDMOAHA.

In Section 2, the mathematical model is thoroughly outlined. In Section 3, the improved part is introduced. In Section 4, the experimental setup is explained. In Section 5, the effectiveness of the improvement and the performance of solving DHFSP is tested.

PROBLEM DESCRIPTION AND FORMULATION

Problem Description

Table 1. Example of the DHFSP

Job	Process time of each stage ($p_{j,k}$)						No. of machines	Run Power ($pp_{f,k,i}$)
	J1	J2	J3	J4	J5	J6		
Stage1	10	7	5	4	5	9	2/3	8,7/9,5,6
Stage2	6	5	4	5	6	3	1/2	4/6,8

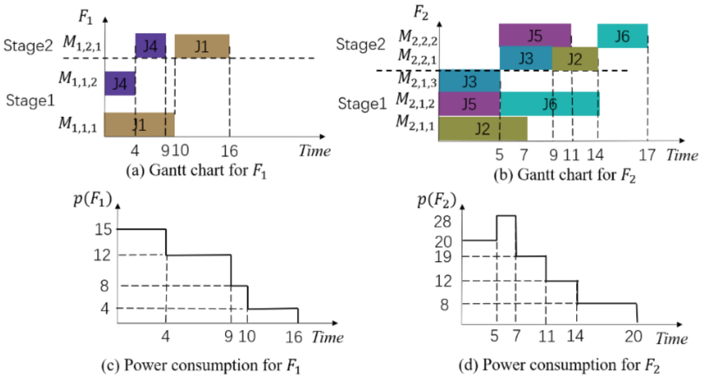


Figure 1. A possible scheduling Gantt chart and power consumption.

The solution to the DHFSP is achieved by addressing three primary subproblems: identifying a factory for the job, determining the machines for the job at each stage, and establishing a suitable sequence of jobs on the machines. Table 1 provides an example of the processing time required for job processing and the energy consumed per unit of time for machine processing. The two factories have 2 and 3 equivalent parallel machines in the first stage, respectively. In the second stage, there are 1 and 2 equivalent parallel machines respectively. The idle energy consumption of all machines is set to 1. Figure 1. shows a feasible scheduling Gantt chart and a trend plot of energy consumption over time with current scheduling. As shown in Figure 1. (a), J_1 and J_4 are processed in F_1. Specifically, J_1 is assigned to $M_{1,1,1}$ and J_4 to $M_{1,1,2}$ in the first stage. J_1 and J_4 are assigned to $M_{1,2,1}$ in the second stage. In Figure 1. (b), the jobs J_2, J_3, J_5 and J_6 are processed in F_2. Specifically, J_2 is assigned to $M_{2,1,1}$, J_3 is assigned to $M_{2,1,3}$, J_5 and J_6 are assigned to $M_{2,1,2}$. In the second stage, J_2 and J_3 are assigned to $M_{2,2,1}$, J_5 and J_6 are assigned to $M_{2,2,2}$.

If $O_{j,k}$ is processed on machine $M_{f,k,i}$, the energy consumption of the machine for processing the job is work time multiplied by energy consumption. For example, the completion times for F_1 and F_2 are 16 and 17, respectively. So Cmax is 17. During the machining of job J_2, the processing time of J_2 is 7, and $M_{2,1,1}$ energy consumption during processing is 9. The energy consumption for machining is calculated as $7 \times 9 = 63$. The total energy consumption of the machines at present is equal to the sum of their energy consumption during processing and their energy consumption while idle. The energy consumption at time 8 in F1 is P(t=8) =pp1,1,1+ pp1,2,1=12. Therefore, the TEC according to Figure 1. is TEC= (15×4+5×12+8×1+6×4) +(20×5+2×28+4×18+3×12+6×8) =464.

Mathematical Formulation

The definition of DHFSP is as follows. It consists of F factories. Each factory has a different processing capacity. It is assumed that there are n jobs to be processed, with each job needing to be allocated to one of F factories and then processed on parallel machines. Each job requires $p_{j,k}$ units of processing time at each stage. Furthermore, a job can only advance to the next stage upon completion of the previous one, and all subsequent processing must be conducted within the factory. The TEC of a machine encompasses both the energy consumption during the machining process and the idle energy consumption that occurs when the machine completes the current task and awaits the arrival of the next job.

Parameters:

F: Number of factories.

n: Number of jobs.

m: Number of processes for the job.

f: Index of F factories, $f \in \{1,2, \ldots F\}$.

j: Index of the jobs, $j \in \{1,2, \ldots n\}$.

k: Index of process, $k \in \{1,2, \ldots m\}$.

$O_{j,k}$: Represents the kth process for the jth job.

$l_{f,k}$: Number of machines in the kth process in the fth factory.

$M_{f,ki}$: Represents the ith machine for the k process in the f factory.

$p_{j,k}$: Standard machining time of the ith job in the kth process.

$pp_{f,k,i}$: The energy consumption per unit for the processing on the ith machine in the kth process at the fth factory.

$sp_{f,k,i}$: Energy consumption per unit at no load for machine ith in process kth in the fth factory.

$E_{c_{f,k,i}}$: Energy consumption in the fth factory for the kth machine processing of the ith process.

$E_{u_{f,k,i}}$: Energy consumption of the kth machine of the ith process in the fth factory at idle.

$S_{j,k}$: Start time of the kth process for job jth.

$x_{f,j}$: When the value is 1, job jth is assigned to factory fth, otherwise it is equal to 0.

$y_{f,k,j,i}$: When the value is 1 it means that the kth process of the job jth is assigned to the ith machine of the factory fth, otherwise it is equal to 0.

$z_{f,k,j,j'}$: The value of 1 indicates that job jth is machined before job j'th in process kth within factory fth.

$C_{j,k}$: End time of the kth process of job jth.

TEC: Total energy consumption.

C_{max}: Maximum completion time for a schedule

U: a large positive number

The mixed integer linear programming model for energy-efficient DHFSP is defined in the following equation, with optimization objectives of C_{max} and TEC.

$$Minimize(C_{max}) \tag{1}$$

Subject to:

$$\sum_{f=1}^{F} x_{f,j} = 1, \forall j \tag{2}$$

$$\sum_{i=1}^{l_{f,k}} y_{f,k,j,i} = x_{f,j}, \forall f, j, k \tag{3}$$

$$S_{j,1} \geq 0, \forall j \tag{4}$$

$$C_{j,k} = S_{j,k} + \sum_{f=0}^{F} \sum_{i=0}^{l_{f,k}} y_{f,k,j,i} \times p_{j,k}, \forall j, k \tag{5}$$

$$z_{f,k,j,j'} + z_{f,k,j',j} \leq 1, \forall f, k, j, j' \tag{6}$$

$$S_{j,k+1} \geq C_{j,k}, \forall j, k \tag{7}$$

$$z_{f,k,j,j'} + z_{f,k,j',j} \geq y_{f,k,j,i} + y_{f,k,j',i} - 1, \forall f, k, j' > j \tag{8}$$

$$S_{j',k} - C_{j,k} + U \times (3\text{-}y_{f,k,j,i} - y_{f,k,j',i} - z_{f,k,j,j'}) \geq 0 \forall j \neq j', k, f, i \in \{1,2,\ldots,l_{f,k}\} \tag{9}$$

$$Cj, k_{max} \tag{10}$$

$$E_{c_{f,k,i}} = \sum_{j=1}^{n} y_{f,k,j,i} \times pp_{f,k,i} \times p_{j,k} k, f, i \in \{1,2,\ldots,l_{f,k}\} \tag{11}$$

$$E_{u_{f,k,i}} = sp_{f,k,i}\{max(C_{j,k} \times y_{f,k,j,i}) - min(S_{j,k} \times y_{f,k,j,i}) - \sum_{j=1}^{n} y_{f,k,j,i} \times p_{j,k}\}$$

$$\forall f, k, i \in \{1,2,\ldots,l_{f,k}\} \tag{12}$$

$$TEC = \sum_{f=1}^{F} \sum_{k=1}^{m} \sum_{i=1}^{l_{f,k}} (E_{c_{f,k,i}} + E_{u_{f,k,i}}) \tag{13}$$

$$x_{f,j} \in \{0,1\}, \forall f, j \tag{14}$$

$$y_{f,k,j,i} \in \{0,1\}, \forall f, k, j, i \in \{1,2,\ldots,l_{f,k}\} \tag{15}$$

$$z_{f,k,j,j'} \in \{0,1\}, \forall f, k, j, i \tag{16}$$

(1) The optimization objectives are Cmax and TEC. (2-3) Each operation of every job is assigned to a machine. (4) Each process of a job can only be assigned to 1 machine of the corresponding process. (5) The start processing time for each job can only be greater than zero. (6) The start time of the next process for the same task can only be after the completion time of the previous process. (7-9) Only one job is processed per machine. (10) Maximum completion time is defined. (11-12) The energy consumption during processing and the idling is calculated. (13) The TEC of the whole process is calculated. (14-16) Binary decision variables are represented.

GDMOAHA

In this section, an improved MOAHA (named GDMOAHA) is proposed. In GDMOAHA, Initialization is changed and the movement pattern is modified. One variation of the strategy is added. These modifications have enhanced the performance in solving problems.

Framework

The framework of GDMOAHA is shown in Algorithm 1. The main differences of GDMOAHA and MOAHA are as follows. The hybrid initialization is done with NEH-SE and random generation. The golden sine strategy is used in the early stage of GDMOAHA. The Cauchy variant is incorporated into the GDMOAHA to facilitate the escape from local optima.

***Algorithm* 1**. Pseudo-code of GDMOAHA

```
Input: Max_Iteration, mu, n, d, Low, Up
Output: Archive
Initialise population with hybrid initialisation
Initialise Visit table
Initialize the EA with non-dominated solutions.
While current iterations (t)<Max_ Iteration
    If t<1/3 Max_ Iteration
        Golden-Sine
    Else
        Hummingbird mobile with stage1 at Algorithm 1
    End
    Cauchy variation
    Hummingbird mobile with stage2 at Algorithm 1
    Sort all the solutions according to crowing distance
    DECD maintenance EA
End
```

Encoding and Decoding

A DHFSP is equivalent to several HFSPs, where each job should be assigned to the proper machine in the proper factory. In HFSPs, the arrangement of the first process can directly affect the subsequent stages. Encoding the first process can simplify the solution space [19]. The scheme of real number coding that is commonly used in HFSP is adopted in GDMOAHA.

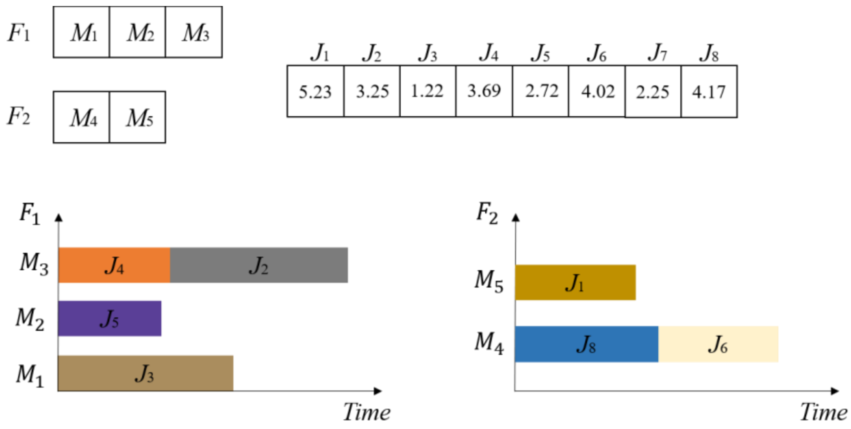


Figure 2. Example of encoding diagrams

An example of encoding is illustrated in Figure 2. The number of machines Mmax for the first process in each factory in the DHFSP is determined. The sequence of factory indexes F= $\{F_1, F_2...., F_n\}$. The sequence of machine index M = $\{M_1, M_2, ..., M_{max}\}$. The sequence of job index J = $\{J_1, J_2, ..., J_{max}\}$. For each job, Ji is assigned to random numbers in the range $[1, M_{max}]$.

An example of encoding diagrams is given in Figure 2. Each of the blocks denotes a job. F_1 and F_2 have $\{M_1, M_2, M_3\}$ and $\{M_4, M_5\}$ machines in the first process, respectively. J_3 is assigned to M_1. J_5 and J_7 is assigned to M_2. J_2 and J4 are assigned to M_3. J_6 and J_8 are assigned to M_4. J_1 is assigned to M_5. There are 4 jobs processed in F_1, and there are 4 jobs processed in F_2. Jobs are sequenced on M_3 with J_4 and J_2. Jobs are sequenced on M_4 with J_8, J_6.

Hybrid Initialization

Optimal solutions are generated more quickly when the initial population is of high quality. The Nawaz-Enscore-Ham (NEH) heuristic is one of the most efficient heuristics for HFSPs [20]. In GDMOAHA, an improved NEH-SE heuristic is proposed based on NEH. The framework of NEH-SE is as follows.

***Algorithm* 2.** NEH-SE

Output: A complete solution π'
1 π is generated by sorting each job in descending order according to its total completion time. $\pi = \{\pi_1, \pi_2, \ldots, \pi_n\}$
2 For *i*th from 1 to π_n
3 The probability of adding an equally spaced job to each job in π
4 End
5 The empty set π'
6 For *i*th from 1 to π_n
7 If *i*th probability> random
8 *i*th job is inserted into all possible locations in π'
9 Assigned to machine with ME
10 Select the sequence with the smallest time
11 Else
12 Assign the next job
13 End
14 End

1. The sequence π is generated in descending order based on each job's total completion time, the one with the longest remaining processing time is completed first.

2. To enhance population diversity, a probability distribution with equal spacing is assigned among jobs. The higher a job's position in the sequence π, the greater the likelihood assigned to it. Subsequently, the job must be compared to a random number within the range (0,1). If the probability of the job is high, the job is assigned. Otherwise, the job is exchanged with the next job position.

3. Jobs are sequentially extracted from π and embedded in all possible positions in π'. The sort that minimizes the makespan is chosen as the new sequence. The job is assigned to idle machines according to ME rules. This operation goes on until every job has been incorporated into the new sequence π'.

The NEH-SE heuristic and random population initialization were used cooperatively to generate half of the initial population. The non-dominated solutions of the iterative process are stored in a Pareto archive.

Golden Sine Strategy

The golden sine strategy is incorporated into the process of updating positions. It narrows the search space using the golden ratio. Since the actual DHFSP problem is discrete, the golden sine strategy can significantly enhance the search efficiency by narrowing down the search space. The golden-sine strategy is introduced as a new way of moving to improve search performance in the pre-iteration period. The golden sine in Eq. 24 is updated:

$$X_i^d(t+1) = X_i^d(t)|sin(r_1)| + r_2\,sin(r_1)\left|x_1 P^d(t) - x_2 X_i^d(t)\right| \quad (17)$$

$$x_1 = a\tau + b(1-\tau) \quad (18)$$

$$x_2 = b\tau + a(1-\tau) \quad (19)$$

Where r1 and r2 are random numbers, $r_1 \in [0,2\pi]$ determines the distance an individual moves in the next iteration. $r_2 \in [0,\pi]$ determines the update direction of the individual position for the next iteration. x1 and x2 are coefficients obtained by introducing the golden section number. x1 and x2 reduce the search space and lead individuals to gradually converge to the optimal value. x1 and x2 are obtained from Eqs. 25 & 26. The initial values of a and b are -π and π, respectively. $\tau$is the golden section$\frac{\sqrt{5}-1}{2}$.

Variation Strategy

After guided foraging and territorial foraging, some hummingbirds randomly search for superior food sources near their current food source. Therefore, the Cauchy Variation strategy is introduced in the updating formula. The Cauchy variation updates as follows [21].

$$X_{i,j}^{i+1} = X_{best}(t) + cauchy(0,1) \oplus X_{best}(t) \quad (20)$$

Where cauchy (0,1) is the standard normal distribution function.

Hummingbirds with the worst food sources will migrate to new food sources with the NEH-SE strategy. The migrated foraging updates are as follows.

$$\begin{cases} wor \in F_{end} \\ x_{wor}(t+1) = NEH - SE \end{cases} \quad (21)$$

NEH-SE represents the initialization method that was introduced in section 3.3.

EXPERIMENT SETUP

Experiment Environment

The experiments were tested with MATLAB on a computer with an Intel(R) Core (TM) i5-8300H CPU (2.30GHz) and 16GB of RAM.

Test Problem and Settings

To evaluate the performance, the EADHFSP test problem set established by Wang J is selected [9]. This problem set is widely used in HFSP and DHFSP [22, 23]. There are 450 specific problems in the problem set. 45 problems of different sizes are selected from the problem set for the experiment. The problem parameters are given as follows. The number of factories, F, is chosen from the set {2, 3, 4, 5, 6}, the number of jobs, n, is chosen from {20, 50, 100}, and the number of stages, m, is chosen from {2, 5, 8}. The standard processing time for each instance is sampled from the range [10, 50]. The number of uniform parallel machines in stage s of factory F is selected from {1, 2, 3, 4, 5}. The power consumption is uniformly distributed from a set {5, 6, 7, 8, 9, 10}. Additionally, each machine's electricity consumption in standby mode is designated as 1. All the machines are equivalent machines and the machines are running at speed 1. To verify the model's correctness, the CPLEX Studio IDE was utilized to construct the model and verify its correctness using small-scale examples. The output from CPLEX Studio IDE confirms the model's correctness.

Parameter Settings of Test MOEAs

GDMOAHA mainly contains key parameters: (1) population size (PS), (2) probability variation (PV), (3) migration coefficient (MC). The effect of parameters on the performance of GDMOAHA was investigated with the design-of-experiment (DOE) experimental approach [24]. For each parameter four different scales were designed i.e. PS∈{60,80,100,120}, PV∈{0,0.02,0.08,0.1}, MC∈{n,1.5n,2n,3n}, n is the current population size. A total of L16(43) matrices are generated. 16 orthogonal matrices are selected among all matrices. To reduce randomness and variation across scales, a small-scale instance (f=3, n=50, s=2) and a large-scale instance (f=5, n=100, s=8) were selected. Each combination was run independently

Table 2. Parameter index and HV value

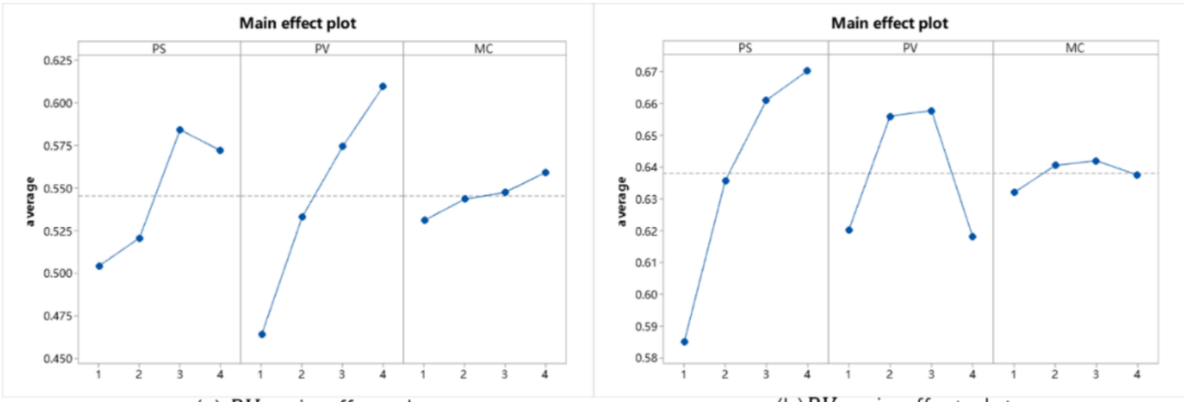
Experiment number	Parameter level			HV_S	HV_L
	PS	PV	MC		
1	1	1	1	0.399	0.539
2	1	2	2	0.483	0.624
3	1	3	3	0.538	0.621
4	1	4	4	0.597	0.556
5	2	1	2	0.457	0.616
6	2	2	1	0.491	0.656
7	2	3	4	0.550	0.656
8	2	4	3	0.585	0.615
9	3	1	3	0.506	0.659
10	3	2	4	0.597	0.671
11	3	3	1	0.605	0.673
12	3	4	2	0.629	0.641
13	4	1	4	0.493	0.667
14	4	2	3	0.561	0.673
15	4	3	2	0.605	0.681
16	4	4	1	0.629	0.660

Table 3. Average HV and the rank of each parameter

Experiment number	Small			Large		
	PS	PV	MC	PS	PV	MC
1	0.505	0.464	0.529	0.585	0.620	0.632
2	0.521	0.532	0.536	0.636	0.654	0.641
3	0.574	0.572	0.547	0.661	0.655	0.642
4	0.572	0.602	0.559	0.670	0.615	0.631
Delta	0.069	0.138	0.018	0.085	0.040	0.01
Rank	3	4	3	4	3	3

10 times and terminated at 150 iterations. The Pareto approximation set is recorded at the end of the GDMOAHA run. Table 2 lists the HV values for small scale instances (HVS) and large scale instances (HVL). The averages of the parameters are presented in Table 3, with Delta representing the maximum gap between different levels of the parameter. Figure 3. shows the trend of the factor levels for the three key parameters. The size of Delta indicates the impact of the current parameter on performance. Table 3 indicates that PM exerts the most significant

influence, with PS coming in second. Therefore, it is recommended to select the following parameters in all cases PS = 100, PV = 0.08, MC = 2n.



(a) RV_s main effect plot (b) RV_l main effect plot

Figure 3. Main effects plot for the GDMOAHA, average response values for (a) small-scale instances and (b) large-scale instances.

Table 4. Parameters setting of algorithms.

Algorithms	Parameters
MOPSO	$c_1 = 1$, $c_2 = 1$, w=0.4, pop = 100
NSWOA	pop = 100, b=1, $r \in [0,1]$
IMA	$a_1 = a_2 = a_3$=1.5, fl =1, g_{min}=0.4, g_{max}=1.3, *male mayfly*=50, *female mayfly*=50

In the following experiment, GDMOAHA had a population size of 100. All problems were run independently 20 times. Each run consists of 150 iterations. Table 4 gives the specific parameter settings of the other multi-objective algorithms.

Evaluation Indicator

Hyper volume (HV) is used as a measure that reflects both convergence and diversity of algorithms [25]. The coverage and distance metrics of the solution set are used to compare the algorithms. The point (1, 1) is used as a reference point. Large HV values represent good algorithm convergence and diversity. The calculation method of HV is shown as follows.

$$HV(f^{ref}, X) = \Lambda(\cup_{X_n \in X}[f_1(X_n), f_1^{ref}] \times \ldots \times [f_m(X_n), f_m^{ref}]) \quad (22)$$

Generational Distance (GD) is a widely used metric in multi-objective optimization. It assesses how close the solution set is to the true Pareto frontier by calculating the distance. A lower GD value indicates that the solution set is nearer to the actual Pareto frontier.

$$GD(S, P) = \frac{1}{|S|}\sum_{i=1}^{|P|} min_{s \in S}\|s - p_i\| \quad (23)$$

RESULTS AND ANALYSIS

This section presents experiments to evaluate the performance of GDMOAHA against MOPSO, IMA and NSWOA under the same conditions.

Comparison of Strategy Effectiveness

Table 5. Comparison on the average HV between GDMOAHA and MOAHA

F/n/s	GDMOAHA		MOAHA		p	
	HV	GD	HV	GD	p_{HV}	p_{GD}
2/20/2	0.6932	0.0054	0.6718	0.0182	0.01%	0.01%
2/20/5	0.5613	0.0075	0.5557	0.0066	0.01%	45.50%
2/20/8	0.4376	0.0124	0.4557	0.0248	0.01%	0.12%
2/50/2	0.6835	0.0151	0.4611	0.0218	0.01%	2.28%
2/50/5	0.6545	0.0223	0.4451	0.0392	50.20%	4.38%
2/50/8	0.5679	0.0220	0.5269	0.0374	0.02%	1.11%
2/100/2	0.6716	0.0126	0.4702	0.0128	0.01%	79.40%
2/100/5	0.5252	0.0420	0.4110	0.0379	0.01%	79.40%
2/100/8	0.3811	0.0217	0.1110	0.0387	0.01%	0.90%
3/20/2	0.7236	0.0118	0.7170	0.0174	0.01%	6.20%
3/20/5	0.6485	0.0046	0.6475	0.0132	0.01%	0.01%
3/20/8	0.5256	0.0095	0.5334	0.0098	0.01%	55.00%
3/50/2	0.4939	0.0151	0.3929	0.0269	21.80%	1.52%
3/50/5	0.7077	0.0055	0.6577	0.0139	85.20%	0.80%
3/50/8	0.5788	0.0052	0.5118	0.0131	1.11%	0.02%
3/100/2	0.5023	0.0158	0.4268	0.0339	0.01%	4.79%
3/100/5	0.5252	0.0154	0.3546	0.0131	0.01%	31.30%
3/100/8	0.5061	0.0110	0.2963	0.0401	0.01%	0.01%
4/20/2	0.7375	0.0156	0.7079	0.0190	0.01%	19.10%
4/20/5	0.6973	0.0119	0.6807	0.0097	0.01%	8.59%
4/20/8	0.5847	0.0091	0.5363	0.0130	0.01%	29.60%
4/50/2	0.7547	0.0089	0.7498	0.0479	0.01%	0.01%
4/50/5	0.5545	0.0074	0.4966	0.0094	0.01%	1.69%
4/50/8	0.6441	0.0092	0.4854	0.0231	0.01%	0.17%
4/100/2	0.7412	0.0113	0.6197	0.0165	6.74%	0.25%
4/100/5	0.6645	0.0110	0.5600	0.0202	0.01%	7.31%
4/100/8	0.5841	0.0142	0.3629	0.0176	0.01%	41.10%
5/20/2	0.7394	0.0056	0.6924	0.0179	0.01%	0.03%
5/20/5	0.5046	0.0148	0.4338	0.0323	0.01%	0.02%
5/20/8	0.6157	0.0105	0.5285	0.0175	0.01%	3.04%
5/50/2	0.6683	0.0120	0.4530	0.0189	0.01%	4.69%
5/50/5	0.7331	0.0051	0.6229	0.0278	50.20%	0.02%
5/50/8	0.5934	0.0088	0.5398	0.0128	0.02%	62.70%
5/100/2	0.6464	0.0181	0.5850	0.0243	0.01%	17.90%
5/100/5	0.5164	0.0110	0.4628	0.0232	0.01%	0.19%
5/100/8	0.5950	0.0048	0.4644	0.0093	0.01%	4.79%
6/20/2	0.7222	0.0079	0.7005	0.0203	0.02%	0.07%
6/20/5	0.6659	0.0174	0.4640	0.0093	0.01%	0.51%
6/20/8	0.5437	0.0152	0.3762	0.0194	0.01%	57.50%
6/50/2	0.6785	0.0146	0.6317	0.0466	0.25%	0.02%
6/50/5	0.7226	0.0098	0.5634	0.0173	0.01%	2.28%
6/50/8	0.5892	0.0071	0.5064	0.0206	0.01%	0.01%
6/100/2	0.6743	0.0132	0.6174	0.0542	0.07%	0.01%
6/100/5	0.7258	0.0115	0.3552	0.0117	0.01%	65.40%
6/100/8	0.7278	0.0041	0.7020	0.0188	0.01%	0.03%
	43/45	40/45			40/45	31/45

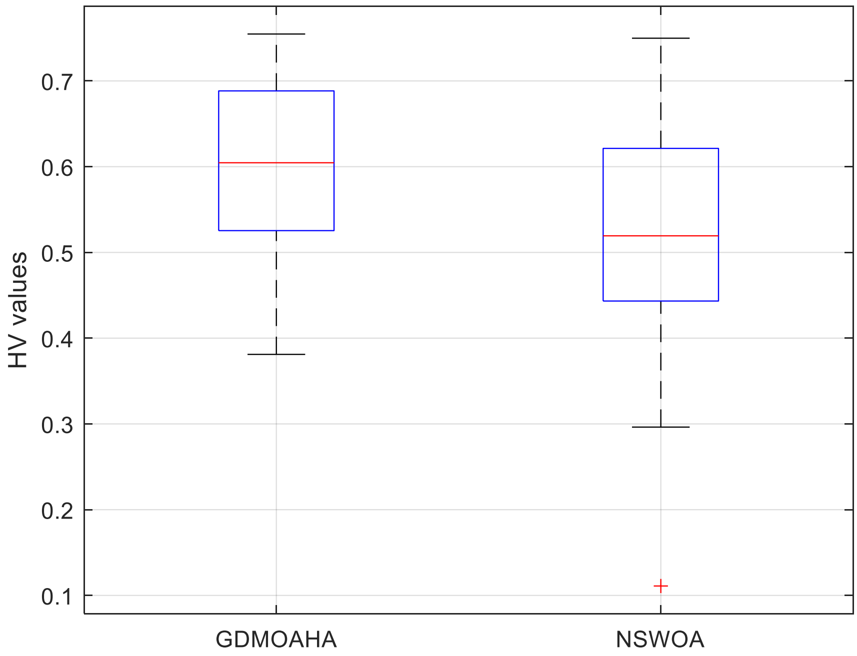


Figure 4. Boxplots of the average HV between GDMOAHA and MOAHA.

To illustrate the benefits of the optimization strategy, GDMOAHA is compared with the original MOAHA. The parameter settings are the same as those in Section 5.3. The best results are in bold. Table 5 gives a comparison of the average HV and GD obtained by GDMOAHA and MOAHA. Figure 4. gives boxplots of the average HV between GDMOAHA and MOAHA. In most cases, the average HV value for GDMOAHA is greater than MOAHA, and almost all of the GD values for GDMOAHA are less than MOAHA. The Wilcoxon signed rank test p satisfies 95% confidence level that the samples are significantly different. This implies that the effectiveness and convergence of GDMOAHA are better than those of MOAHA in the vast majority of cases.

Comparisons to Existing Algorithms and Analysis of the Experiment

The effectiveness of GDMOAHA is verified by a comparison with the NSWOA, MOPSO, and IMA. Table 6 and Table 7 present the all-instance average HV and average GD metrics for each scale (F, n, s), along with the p-values for all the data, as determined by the Wilcoxon signed rank test at a 95% confidence level. The p-values demonstrate the level of statistical difference among the non-dominated sequences of GDMOAHA, NSWOA, MOPSO, and IMA. The HV box diagrams for all instances are shown in Figure 5., Figure 6. compares the Pareto fronts of GDMOAHA, NSWOA, MOPSO, and IMA at different scales. From Table 6 and Figure 5., the HV value of GDMOAHA is much larger than that of NSWOA and MOPSO. And p is close to 0 in all instances. This indicates that GDMOAHA is much better than NSWOA and MOPSO in terms of convergence and diversity in all cases. In the vast majority of cases, the HV of GDMOAHA is greater than the HV of IMA. Only in rare cases (2/20/8, 4/50/5, 5/20/5, 6/20/2) is it less than or close to the HV value of GDMOAHA. It shows that the diversity and convergence of IMA is also excellent. However, in the vast majority of cases, GDMOAHA has better convergence and diversity than IMA. In Table 7, the four algorithms achieve the smallest GD values as follows: GDMOAHA (32/45), IMA (11/45), and MOPSO (2/45). The results show that in most instances, GDMOAHA has the smallest GD value and satisfies the 95% confidence level. This indicates that the GDMOAHA derived non-dominated solution set is more closely approximated to the true solution set. In summary, GDMOAHA can solve the problem better than MOPSO, NSWOA and IMA.

Table 6. Comparison on the average HV metric among GDMOAHA, NSWOA, MOPSO and IMA

F/n/s	HV						
	GDMOAHA	NSWOA	*p*	MOPSO	*p*	IMA	*p*
2/20/2	0.6932	0.4274	0.01%	0.5742	0.01%	0.6317	0.01%
2/20/5	0.5613	0.2441	0.01%	0.4135	0.01%	0.5013	0.01%
2/20/8	0.4376	0.2132	0.01%	0.3413	0.01%	0.4385	0.01%
2/50/2	0.6835	0.2772	0.01%	0.3949	0.01%	0.4763	0.01%
2/50/5	0.6545	0.1904	0.01%	0.2985	0.01%	0.4636	0.01%
2/50/8	0.5679	0.1987	0.01%	0.1615	0.01%	0.3840	0.01%
2/100/2	0.6716	0.0895	0.01%	0.4124	0.01%	0.4306	0.01%
2/100/5	0.5252	0.1515	0.01%	0.1007	0.01%	0.3284	0.01%
2/100/8	0.3811	0.1305	0.01%	0.1188	0.01%	0.1801	0.01%
3/20/2	0.7236	0.3267	0.01%	0.6294	0.01%	0.6507	0.01%
3/20/5	0.6485	0.2993	0.01%	0.4061	0.01%	0.5514	0.01%
3/20/8	0.5256	0.2727	0.01%	0.4303	0.01%	0.4893	0.01%
3/50/2	0.4939	0.1788	0.01%	0.3616	0.01%	0.3566	0.01%
3/50/5	0.7077	0.1966	0.01%	0.2595	0.01%	0.5482	0.01%
3/50/8	0.5788	0.1370	0.01%	0.3612	0.01%	0.4327	0.01%
3/100/2	0.5023	0.1555	0.01%	0.1256	0.01%	0.4612	0.01%
3/100/5	0.5252	0.1363	0.01%	0.1942	0.01%	0.5135	0.01%
3/100/8	0.5061	0.0842	0.01%	0.0918	0.01%	0.3240	0.01%
4/20/2	0.7375	0.4092	0.01%	0.4329	0.01%	0.7081	0.01%
4/20/5	0.6973	0.3892	0.01%	0.4218	0.01%	0.6746	0.01%
4/20/8	0.5847	0.3052	0.01%	0.4103	0.01%	0.5465	0.01%
4/50/2	0.7547	0.3145	0.01%	0.4479	0.01%	0.6275	0.01%
4/50/5	0.5545	0.3221	0.01%	0.2182	0.01%	0.6593	0.01%
4/50/8	0.6441	0.2410	0.01%	0.2630	0.01%	0.5252	0.01%
4/100/2	0.7412	0.2925	0.01%	0.4221	0.01%	0.5454	0.01%
4/100/5	0.6645	0.1917	0.01%	0.1896	0.01%	0.4157	0.01%
4/100/8	0.5841	0.1226	0.01%	0.1906	0.01%	0.3540	0.01%
5/20/2	0.7394	0.3900	0.01%	0.4305	0.01%	0.6622	0.01%
5/20/5	0.5046	0.3415	0.01%	0.3681	0.01%	0.5629	0.01%
5/20/8	0.6157	0.2725	0.01%	0.3093	0.01%	0.5511	0.01%
5/50/2	0.6683	0.3106	0.01%	0.3864	0.01%	0.6274	0.01%
5/50/5	0.7331	0.3181	0.01%	0.4706	0.01%	0.5203	0.01%
5/50/8	0.5934	0.1767	0.01%	0.2859	0.01%	0.5108	0.01%
5/100/2	0.6464	0.3350	0.01%	0.2931	0.01%	0.6400	0.01%
5/100/5	0.4897	0.1078	0.01%	0.0890	0.01%	0.4682	0.01%
5/100/8	0.5772	0.1575	0.01%	0.2997	0.01%	0.4219	0.01%
6/20/2	0.7222	0.5032	0.01%	0.4584	0.01%	0.7267	0.01%
6/20/5	0.6659	0.5230	0.01%	0.3402	0.01%	0.6463	0.01%
6/20/8	0.5437	0.2720	0.01%	0.3081	0.01%	0.4854	0.01%
6/50/2	0.6785	0.2886	0.01%	0.3559	0.01%	0.5361	0.01%
6/50/5	0.7226	0.2610	0.01%	0.3389	0.01%	0.6045	0.01%
6/50/8	0.5892	0.1804	0.01%	0.3075	0.01%	0.4993	0.01%
6/100/2	0.6743	0.1636	0.01%	0.3849	0.01%	0.4763	0.01%
6/100/5	0.7258	0.2465	0.01%	0.3647	0.01%	0.5214	0.01%
6/100/8	0.7278	0.2231	0.01%	0.4089	0.01%	0.5572	0.01%
	41/45		45/45		45/45		42/45

Table 7. Comparison on the average GD metric among GDMOAHA, NSWOA, MOPSO and IMA

F/n/s	GD						
	GDMOAHA	NSWOA	*p*	MOPSO	*p*	IMA	*p*
2/20/2	0.0126	0.0457	0.02%	0.0438	0.19%	0.0412	0.01%
2/20/5	0.0420	0.0407	60.10%	1.0000	0.14%	0.0134	5.69%
2/20/8	0.0217	1.0000	0.01%	0.0326	6.74%	0.0665	0.01%
2/50/2	0.0054	0.0228	0.01%	0.0174	0.08%	0.0113	0.03%
2/50/5	0.0076	0.0515	0.01%	0.0166	0.51%	0.0157	1.37%
2/50/8	0.0124	0.0333	0.05%	0.0513	0.01%	0.0110	39.10%
2/100/2	0.0151	1.0000	0.03%	0.0177	94.00%	0.0144	97.00%
2/100/5	0.0223	0.0444	0.46%	0.0361	0.28%	0.0266	73.70%
2/100/8	0.0220	0.0271	31.30%	0.0212	79.40%	0.0222	73.70%
3/20/2	0.0158	0.0548	0.03%	0.0583	0.01%	0.0164	68.10%
3/20/5	0.0154	0.0472	0.02%	0.0231	6.74%	0.0203	7.31%
3/20/8	0.0110	0.1600	0.01%	1.0000	6.25%	0.0258	0.09%
3/50/2	0.0118	0.0305	0.05%	0.0217	3.33%	0.0175	97.00%
3/50/5	0.0046	0.0260	0.01%	0.0209	0.01%	0.0091	0.46%
3/50/8	0.0095	0.0634	0.01%	0.0307	0.01%	0.0172	4.79%
3/100/2	0.0151	0.0533	0.01%	0.0230	15.60%	0.0190	11.70%
3/100/5	0.0055	0.0244	0.01%	0.0278	0.01%	0.0124	0.06%
3/100/8	0.0052	0.0485	0.01%	0.0331	0.01%	0.0063	20.40%
4/20/2	0.0113	0.0253	0.13%	0.0124	62.70%	0.0125	12.60%
4/20/5	0.0110	0.0349	0.01%	0.0235	0.10%	0.0102	50.20%
4/20/8	0.0142	0.0777	0.01%	0.0357	0.01%	0.0221	3.66%
4/50/2	0.0156	0.0180	70.90%	0.0229	33.20%	0.0090	1.87%
4/50/5	0.0119	0.0198	10.00%	0.0182	1.52%	0.0144	47.80%
4/50/8	0.0091	0.0249	0.32%	0.0324	0.06%	0.0134	15.60%
4/100/2	0.0089	0.0316	0.13%	0.0166	1.69%	0.0652	0.01%
4/100/5	0.0074	0.0296	0.01%	0.0212	0.04%	0.0326	0.01%
4/100/8	0.0092	1.0000	0.01%	0.0235	0.01%	0.0163	2.28%
5/20/2	0.0181	0.0663	0.02%	0.0181	70.90%	0.0163	33.20%
5/20/5	0.0110	1.0000	0.02%	1.0000	0.01%	0.0244	0.32%
5/20/8	0.0048	0.0599	0.01%	0.0167	0.02%	0.0077	0.13%
5/50/2	0.0056	0.0168	0.04%	0.0229	0.07%	0.0126	0.28%
5/50/5	0.0148	0.0729	0.01%	0.0307	0.10%	0.0129	65.40%
5/50/8	0.0105	0.0286	0.17%	0.0219	0.64%	0.0070	50.20%
5/100/2	0.0120	0.0536	0.01%	0.0240	0.06%	0.0076	3.04%
5/100/5	0.0051	0.0401	0.01%	0.0202	0.01%	0.0502	0.01%
5/100/8	0.0088	0.0534	0.01%	0.0280	0.01%	0.0066	4.38%
6/20/2	0.0132	0.0341	0.02%	0.0104	33.20%	0.0454	0.02%
6/20/5	0.0115	0.0173	7.31%	0.0221	4.00%	0.0257	6.74%
6/20/8	0.0041	0.0204	0.01%	0.0220	0.01%	0.0604	0.01%
6/50/2	0.0079	0.0211	0.01%	0.0203	0.01%	0.0145	0.72%
6/50/5	0.0174	0.0192	85.20%	0.0132	12.60%	0.0069	0.03%
6/50/8	0.0152	0.0434	0.15%	0.0218	7.31%	0.0159	62.70%
6/100/2	0.0146	1.0000	0.22%	0.0211	10.80%	0.0342	0.64%
6/100/5	0.0098	0.0435	0.01%	0.0371	0.04%	0.0117	21.80%
6/100/8	0.0071	0.0721	0.01%	0.0423	0.01%	0.0123	2.76%
	32/45		39/45		32/45		25/45

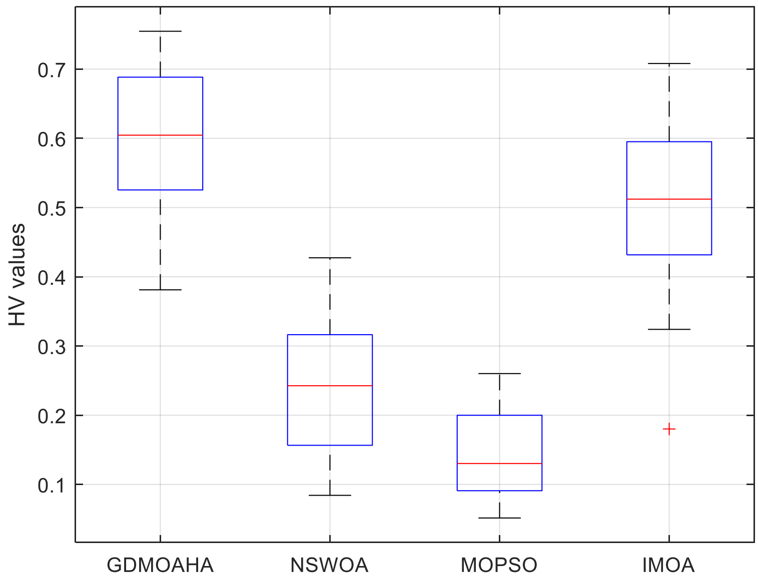
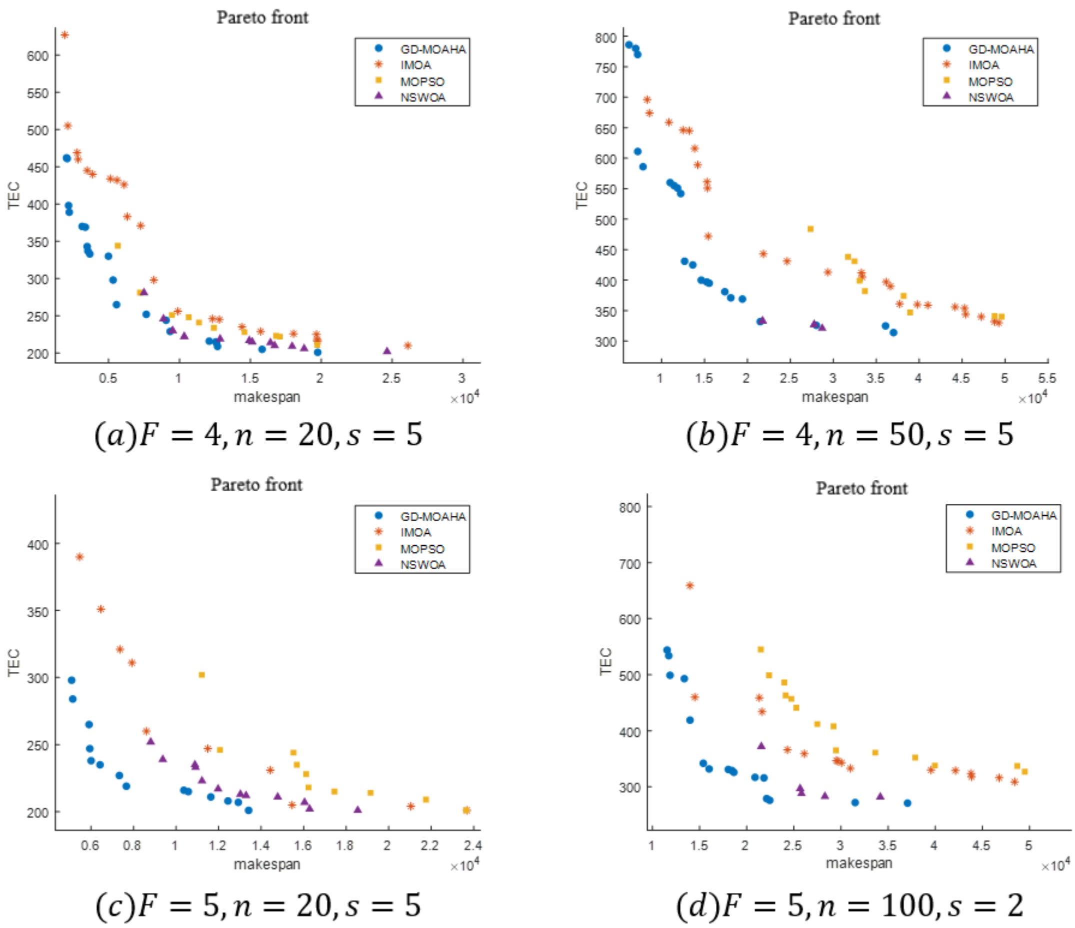


Figure 5. Boxplots of the average HV among GDMOAHA, NSWOA, MOPSO and IMA.



$(a) F = 4, n = 20, s = 5$

$(b) F = 4, n = 50, s = 5$

$(c) F = 5, n = 20, s = 5$

$(d) F = 5, n = 100, s = 2$

Figure 6. The Pareto fronts obtained by GDMOAHA, NSWOA, MOPSO and IMA for instances with different scales.

From the comparison of Pareto frontiers given in Figure 6., the GDMOAHA solutions are closer to the optimal solution than those of MOPSO and IMA. Although NSWOA and GDMOAHA are similar in terms of Pareto solutions quality, GDMOAHA obtains more non-dominated solutions in the Pareto frontier in all examples. The convergence and diversity of GDMOAHA are significantly better than those of NSWOA.

CONCLUSIONS AND FUTURE WORK

The purpose of this article is to solve the multi-objective optimization problem for DHFSP to optimize makespan and total energy consumption. A GDMOAHA is proposed for solving DHFSP. The comparison was conducted with MOPSO, NSWOA, and IMA on 45 instances. In most of these instances, the solution set generated by GDMOAHA is optimal in terms of HV and GD. This indicates that GDMOAHA has superior convergence and diversity, and the solutions it finds are closer to the true solution set.

In the future, the research will be further conducted. The actual production is affected by many factors. Therefore, more constraints can be added to the proposed energy efficient scheduling model to approach the actual production environment. Additionally, the algorithm has the potential to be expanded for addressing a variety of multi-objective scheduling challenges.

ACKNOWLEDGMENTS

This work was supported by the Innovation Project of Engineering Research Center of Integration and Application of Digital Learning Technology (No.1221003)

REFRENCES

[1] Utama D M, Primayesti M D, Umamy S Z, et al. A systematic literature review on energy-efficient hybrid flow shop scheduling. Cogent Engineering, 2023, 10(1): 2206074, doi: 10.1080/23311916.2023.2206074.

[2] Zhang B, Pan Q, Gao L, et al. A three-stage multiobjective approach based on decomposition for an energy-efficient hybrid flow shop scheduling problem. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019, 50(12): 4984-4999, doi: 10.1109/TSMC.2019.2916088.

[3] Qin H, Han Y, Wang Y, et al. Intelligent optimization under blocking constraints: A novel iterated greedy algorithm for the hybrid flow shop group scheduling problem. Knowledge-Based Systems, 2022, 258: 109962. https://doi.org/10.1016/j.knosys.2022.109962

[4] Wang Z, Shen L, Li X, et al. An improved multi-objective firefly algorithm for energy-efficient hybrid flowshop rescheduling problem. Journal of Cleaner Production, 2023, 385: 135738. https://doi.org/10.1016/j.jclepro.2022.135738

[5] Teng Y, Li X, Gao L, et al. An improved co-evolutionary memetic algorithm based on novel schedule type and unconditional feasibility for hybrid flow-shop scheduling problem. Computers & Industrial Engineering, 2024: 110324, doi: 10.1016/j.cie.2024.110324.

[6] Toptal A, Sabuncuoglu I. Distributed scheduling: a review of concepts and applications. International Journal of Production Research, 2010, 48(18): 5235-5262, doi: 10.1080/00207540903121065.

[7] Li Y, Li F, Pan Q K, et al. An artificial bee colony algorithm for the distributed hybrid flowshop scheduling problem. Procedia Manufacturing, 2019, 39: 1158-1166.https://doi.org/10.1016/j.cie.2020.106638

[8] Cai J, Lei D, Li M. A shuffled frog-leaping algorithm with memeplex quality for bi-objective distributed scheduling in hybrid flow shop. International Journal of Production Research, 2021, 59(18): 5404-5421, doi: 10.1080/00207543.2020.1780333.

[9] Wang J, Wang L. A cooperative memetic algorithm with learning-based agent for energy-aware distributed hybrid flow-shop scheduling. IEEE Transactions on Evolutionary Computation, 2021, 26(3): 461-475, doi: 10.1109/TEVC.2021.3106168.

[10] Wang J, Wang L. A cooperative memetic algorithm with feedback for the energy-aware distributed flow-shops with flexible assembly scheduling. Computers & Industrial Engineering, 2022, 168: 108126, doi: 10.1016/j.cie.2022.108126.

[11] Cui H, Li X, Gao L. An improved multi-population genetic algorithm with a greedy job insertion inter-factory neighborhood structure for distributed heterogeneous hybrid flow shop scheduling problem. Expert Systems with Applications, 2023, 222: 119805. doi: 10.1016/j.eswa.2023.119805

[12] R. Li, W. Gong, L. Wang, C. Lu, Z. Pan and X. Zhuang, "Double DQN-Based Coevolution for Green Distributed Heterogeneous Hybrid Flowshop Scheduling with Multiple Priorities of Jobs," in IEEE Transactions on Automation Science and Engineering, vol. 21, no. 4, pp. 6550-6562, Oct. 2024, doi: 10.1109/TASE.2023.3327792
[13] Lu C, Zheng J, Yin L, et al. An improved iterated greedy algorithm for the distributed hybrid flowshop scheduling problem. Engineering Optimization, 2024, 56(5): 792-810.doi: 10.1080/0305215X.2023.2198768
[14] Lu C, Zhou J, Gao L, et al. Modeling and multi-objective optimization for energy-aware scheduling of distributed hybrid flow-shop. Applied Soft Computing, 2024, 156: 111508. doi: 10.1016/j.asoc.2024.111508
[15] Zhao W, Zhang Z, Mirjalili S, et al. An effective multi-objective artificial hummingbird algorithm with dynamic elimination-based crowding distance for solving engineering design problems. Computer Methods in Applied Mechanics and Engineering, 2022, 398: 115223, doi: 10.1016/j.cma.2022.115223.
[16] Li G, Wei X, Yang H. A method for accurate prediction of photovoltaic power based on multi-objective optimization and data integration strategy. Applied Mathematical Modelling, 2024, 136: 115643. doi: 10.1016/j.apm.2024.115643
[17] Abid M S, Apon H J, Nafi I M, et al. Multi-objective architecture for strategic integration of distributed energy resources and battery storage system in microgrids. Journal of Energy Storage, 2023, 72: 108276. doi: 10.1016/j.est.2023.108276
[18] Wolpert D H, Macready W G. No free lunch theorems for optimization. IEEE transactions on evolutionary computation, 1997, 1(1): 67-82. doi: https://doi.org/10.1109/4235.585893
[19] Fernandez-Viagas V, Perez-Gonzalez P, Framinan J M. Efficiency of the solution representations for the hybrid flow shop scheduling problem with makespan objective. Computers & Operations Research, 2019, 109: 77-88, doi: 10.1016/j.cor.2019.05.002.
[20] Nawaz M, Enscore Jr E E, Ham I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega, 1983, 11(1): 91-95, doi: 10.1016/0305-0483(83)90088-9.
[21] Wang S, Cao L, Chen Y, et al. Gorilla optimization algorithm combining sine cosine and cauchy variations and its engineering applications. Scientific Reports, 2024, 14(1): 1-20, doi: 10.1038/s41598-024-58431-x.
[22] Q. K. Pan, L. Wang, J. Q. Li, and J. H. Duan, “A novel discrete artificial bee colony algorithm for the hybrid flowshop scheduling problem with makespan minimization,” Omega, vol. 45, pp. 42–56, Jun. 2014, doi: 10.1016/j.omega.2013.12.004.
[23] K. C. Ying and S. W. Lin, “Minimizing makespan for the distributed hybrid flowshop scheduling problem with multiprocessor tasks,” Expert Syst. Appl., vol. 92, pp. 132–141, Feb. 2018, doi: 10.1016/j.eswa.2017.09.032.
[24] Montgomery D C. Design and analysis of experiments. John wiley & sons, 2017, doi: 10.1198/tech.2006.s372.
[25] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm. TIK report, 2001, 103, doi: 10.3929/ethz-a-004284029.