



Intelligent learning-based cooperative and competitive multi-objective optimization for energy-aware distributed heterogeneous welding shop scheduling

Fayong Zhang¹ · Caixian Li² · Rui Li³ · Wenyin Gong³

Received: 26 June 2023 / Accepted: 23 December 2023 / Published online: 10 February 2024
© The Author(s) 2024

Abstract

This research is focused on addressing the energy-aware distributed heterogeneous welding shop scheduling (EADHWS) problem. Our primary objectives are to minimize the maximum finish time and total energy consumption. To accomplish this, we introduce a learning-based cooperative and competitive multi-objective optimization method, which we refer to as LCCMO. We begin by presenting a multi-rule cooperative initialization approach to create a population that combines strong convergence and diversity. This diverse population forms the foundation for our optimization process. Next, we develop a multi-level cooperative global search strategy that explores effective genes within solutions from different angles and sub-problems. This approach enhances our search for optimal solutions. Moreover, we design a competition and cooperation strategy for different populations to expedite convergence. This strategy encourages the exchange of information and ideas among diverse populations, thereby accelerating our progress. We also introduce a multi-operator cooperative local search technique, which investigates elite solutions from various directions, leading to improved convergence and diversity. In addition, we integrate Q-learning into our competitive swarm optimizer to explore different regions of the objective space, enhancing the diversity of the elite archive. Q-learning guides the selection of operators within the small-size population, contributing to more efficient optimization. To evaluate the effectiveness of LCCMO, we conduct numerical experiments on 20 instances. The experimental results unequivocally demonstrate that LCCMO outperforms six state-of-the-art algorithms. This underscores the potential of our learning and knowledge-driven evolutionary framework in enhancing performance and autonomy when it comes to solving EADHWS.

Keywords Welding shop scheduling problem · Distributed heterogeneous factory · Competitive swarm optimizer · Reinforcement learning · Multi-objective optimization

Introduction

The welding process plays a crucial role in modern manufacturing, spanning various industries such as shipbuilding, aerospace, construction machinery, automotive, and others [1–3]. The efficiency of welding operations significantly impacts overall production timelines, with welded structures constituting over 50% of machinery parts [4, 5]. Notably, the welding of large-scale structural components is a critical aspect of manufacturing construction machinery. Enhancing the efficiency of welding shops has become a pressing concern for manufacturing enterprises. However, there is a scarcity of research addressing welding shop scheduling problems (WSP) from an engineering application perspective. Furthermore, as global trade continues to expand, distributed manufacturing has become the prevailing mode in

✉ Wenyin Gong
wygong@cug.edu.cn

Fayong Zhang
zhangfayong@cug.edu.cn

Caixian Li
183529812@qq.com

Rui Li
leerhpc@cug.edu.cn

¹ College of Geography and Information Engineering, China University of Geosciences, Wuhan 430074, China

² Assert Company, China University of Geosciences, Wuhan 430074, China

³ School of Computer Science, China University of Geosciences, Wuhan 430074, China

modern manufacturing [6–8]. Consequently, different factories exhibit diverse production environments, encompassing variations in machine numbers, types, workers' skills, and more. Hence, the study of distributed heterogeneous WSP (DHWSP) holds significance [9].

The WSP is an extension of the permutation flow shop scheduling problem [10], taking into account that a job can be processed by multiple welders in parallel [11]. The DHWSP further extends the WSP by considering multiple heterogeneous factories to handle a substantial volume of orders [12, 13]. Mainstream approaches for DHWSP encompass decomposition-based methods [9, 12] and Pareto domination-based methods [1, 13]. However, previous approaches have several notable disadvantages: (1) **Global Search:** In previous methods for solving energy-aware DHWSP, random elite selection is often employed for crossover during the global search phase. This approach can result in suboptimal solutions learning from other suboptimal solutions, leading to less-than-ideal outcomes. (2) **Local Search:** Previous methods, such as those mentioned in references [12, 13], apply local search operators either to the offspring or the entire population. This can cause the population to converge toward local optima, limiting the efficient utilization of computational resources. (3) **Solver Utilization:** Due to the complex nature of the search space in DHWSP, previous algorithms struggle to sufficiently explore the objective space using a single solver. Each solver, whether it's an evolutionary algorithm as in [12] or a reinforcement learning-based method as in [9], can only explore a portion of the space due to their inherent characteristics. As a result, cooperating with different solvers can combine their search capabilities to thoroughly explore various parts of the objective space, leading to more comprehensive and effective optimization.

The competitive swarm optimizer (CSO), as introduced by Zhang [14], is designed to accelerate the convergence of global search. Unlike traditional particle swarm optimization methods, CSO divides each generation's population into two distinct groups: winners and losers. In this process, only the winners undergo mutation to improve their own characteristics, while the losers gain valuable knowledge from the selected winners to generate new solutions. However, it is important to note that CSO was initially developed for continuous optimization problems and is not directly applicable to the context of the DHWSP. Nonetheless, the concept of competition-based cooperation among individual solutions, as employed in CSO, can be integrated into the DHWSP algorithm to overcome the limitations of previous global search methods. This adaptation may lead to improvements in solving the DHWSP by harnessing the power of competitive swarm optimization principles.

Therefore, this study proposes a learning-driven cooperative and competitive multi-objective optimizer (LCCMO) to address these challenges for DHWSP. The optimization

objectives include makespan and total energy computation (TEC). LCCMO makes the following contributions:

- (1) **Cooperative Optimization Framework:** We propose a learning-based cooperative and competitive multi-objective optimizer that encompasses cooperation at multiple levels, including the collaboration of strategies, populations, and solvers.
- (2) **Cooperation of Population:** In our approach, we partition the primary population into winner and loser subpopulations within the mating pool. We utilize a competitive and cooperative strategy to expedite the convergence of the global search. Additionally, we manage two distinct solver populations that retain elite solutions, thereby contributing to the overall performance improvement of the final non-dominated solution set.
- (3) **Cooperation of Solvers:** We leverage a cooperative approach that combines the strengths of both the reinforcement learning-based method and the evolution-based method. By doing so, we effectively explore distinct regions of the objective space, promoting improved convergence and diversity in our optimization process.
- (4) **Cooperation of Strategies:** We incorporate three heuristic rules to collaboratively produce a high-quality population. Additionally, we employ five knowledge-driven local search operators to enhance the convergence of our algorithm. Furthermore, we combine three crossover operators to create larger steps, facilitating a more rapid exploration of the objective space.

LCCMO is systematically compared with six state-of-the-art algorithms using 20 instances from [12, 13] to assess its effectiveness. The numerical experimental results unequivocally demonstrate the significant superiority of LCCMO.

The structure of this paper is organized as follows: Section “**Literature review**” provides a comprehensive literature review, emphasizing the research gap in the field. In Section “**Problem statement and modeling**”, we intricately construct the model for DHWSP. Section “**Our approach: LCCMO**” delves into the specific details of the proposed method, referred to as LCCMO. We present the experimental results in detail in Section “**Experimental results**”. Finally, the paper concludes in Section “**Conclusions**”, summarizing the findings and outlining future research directions.

Literature review

Welding shop scheduling

Lu [15] was the first to investigate WSP from a real-world factory perspective, considering controllable processing time, job-dependent transportation times, and sequence-dependent

setup times. Furthermore, Lu [16] delved into WSP with dynamic events in a real-world environment, factoring in issues such as jobs with poor quality, machine breakdown, and jobs with release time delay. Lu designed a gray wolf optimizer for solving WSP, aiming to simultaneously minimize machine load, makespan, and instability. Li [17] proposed a multi-objective artificial bee colony algorithm with knowledge-based local search operators for this problem. Rao [18] studied WSP with the goal of minimizing the machine interaction effect and the total tardiness simultaneously. Wang [12, 13] designed an effective whale swarm optimizer and a decomposition-based multi-objective evolutionary algorithm (MOEA/D) for DHWSP. Wang [19] constructed a welding shop inverse scheduling problem with dynamic events and developed an enhanced gray wolf optimizer for solving it. Wang [11] proposed MOEA/D with adaptive resource allocation for solving energy-efficient WSP. Wang [9] designed a cooperative memetic algorithm for energy-efficient DHWSP and achieved favorable results. Lu [1] introduced human-robot collaborative scheduling in WSP and designed a Pareto-based memetic algorithm for solving it.

Competitive swarm optimizer

$$\begin{cases} v_l(t+1) = r_0 v_l(t) + r_1 (x_w(t)' - x_l(t)'), \\ x_l(t+1) = x_l(t) + v_l(t+1), \\ x_w(t)' = x_w(t) + r_0 v_w(t), \\ x_l(t)' = x_l(t) + r_0 v_l(t), \end{cases} \quad (1)$$

The CSO was originally proposed for continuous optimization problems, with the primary learning schema being vector difference as depicted in Eq. (1). Leveraging its rapid-learning feature, CSO is frequently employed for solving multi-/many-objective optimization problems (MOPs) [20, 21]. Gu improved initialization and learning strategies of CSO which can get better balance convergence and diversity [22]. Huang introduced parameter adaptive CSO to enhance its intelligence [23]. Large-scale MOPs constitute the primary research domain of CSO [24], leading to the proposal of various improved versions of CSO. Mohapatra designed a tri-competitive schema-based CSO to enhance exploration [25–27], while Ge introduced inverse modeling to update winners and accelerate the convergence of CSO [28]. Liu devised three distinct competitive schemas to improve the diversity of CSO [29], and Qi incorporated a neighborhood search strategy to enhance CSO [30]. Huang divided CSO into three phases, yielding superior results compared to state-of-the-art approaches. Additionally, CSO finds applications in constrained MOPs [31], many-objective optimization

problems [32], feature selection [33], and wireless sensor networks [34].

Meta-heuristics with learning strategies

Learning-based meta-heuristics for addressing shop scheduling problems have garnered significant attention. These learning strategies encompass machine learning-based solution generation [35], machine learning-based operator selection [7, 36], reinforcement learning (RL)-based parameter selection [37], RL-based operator selection [38, 39], and deep RL-based operator selection [40, 41]. Recently, the integration of RL with meta-heuristics has been thoroughly explored. This collaboration is gaining traction because RL can also serve as a solver for optimization problems [42]. Therefore, this research focuses on the synergy between RL and meta-heuristic approaches, examining novel ways of cooperation between these two paradigms.

Problem statement and modeling

Problem statement

The DHWSP mainly solves following sub-problems: (i) Dispatching each job to every heterogeneous factory; (ii) Arranging the job sequence in all factories; and (iii) Determining the number of processing machines for each operation on each stage. A DHWSP instance has n jobs and n_f factories. Each job has to be processed in sequence on n_s stage in every factory. There are $m_{f,s}$ welders which can process an operation in parallel in each stage. Each job's original processing time is $p_{f,i,s}$ in stage s in factory f . Moreover, each job's real processing time $r_{f,i,s}$ is $p_{f,i,s}/\mu_{f,i,s}$, where $\mu_{f,i,s}$ is the welder number used in stage s . However, the welding energy consumption of operation $O_{i,s}$ does not increase in linearity. A nonlinear coefficient η is multiplied by the number of welders $\mu_{f,i,s}$ and the welding energy consumption of operation $O_{i,s}$ is $r_{f,i,s} \times \mu_{f,i,s} \times \eta$. The more $\mu_{f,i,s}$, the higher welding energy consumption. This modification is different from previous works [9, 12]. The primary goal of DHWSP is to solve these sub-problems in order to find a Pareto solutions set that minimizes both the makespan and TEC. Figure 1 illustrates a Gantt chart representing a solution for DHWSP. Importantly, each job can be processed by multiple welders simultaneously.

DHWSP operates under the following assumptions: (1) Each welder is assigned to process only one job at each stage. However, it's important to note that each job can be simultaneously processed by multiple welders. Additionally, each welder is linked to a single machine. (2) The processing times and associated energy consumption are deterministic and known in advance. (3) Each job cannot be split and pro-

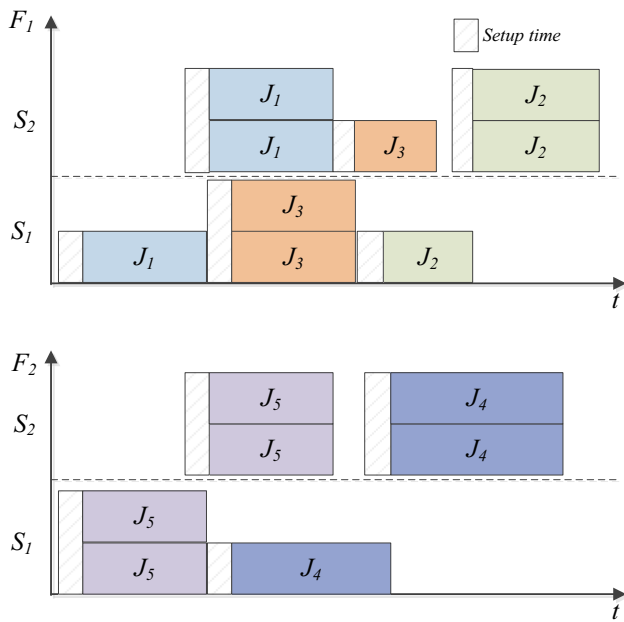


Fig. 1 The gantt chart of a DHWSP solution

cessed in two different factories. It must be entirely processed within a single factory. (4) The problem does not consider transportation times or dynamic events that might affect the scheduling.

Mathematical formulation

DHWSP’s notations are described as follows:

Indices:

- i : index of each job, $i \in \{1, \dots, n\}$;
- s : index of each stage, $k \in \{1, \dots, n_s\}$;
- f : index of each factory, $f \in \{1, \dots, n_f\}$;

Parameters:

- n : the number of jobs;
- n_f : the number of factories;
- n_s : the number of stage;
- $m_{f,s}$: maximum welders in each stage in f ;
- $p_{f,i,s}$: The original processing time for job i on stage s in factory f ;
- $s_{f,i,s}$: The setup time for job i on stage s in factory f ;
- W_B the basic power of whole factories;
- W_I machine idle power;
- W_O machine welding power;
- W_S machine setup power;
- K_w welding duty cycle;
- L : a integer large enough;

Decision variables:

- E_B the basic energy consumption;
- E_I machine idle energy consumption;
- E_O machine welding energy consumption;
- E_S machine setup energy consumption;
- $\mu_{f,i,s}$: The number of welders used by job i on stage s in factory f ;
- $C_{f,i,s}$: the completion time of job i on stage s in factory f ;
- C_{max} : the makespan of a schedule;
- TEC : the total energy consumption of a schedule;
- $X_{f,i,s}$: The value is equals to 1, if job i is processed on stage s in factory f ; otherwise it is set to 0;

In the context of DHWSP, the primary objectives to be considered are makespan (the time taken to complete all jobs) and total energy consumption (TEC). An interesting trade-off arises in this problem: (1) Makespan: When you have more welders, the processing time for each job (denoted as job i) tends to decrease, which in turn leads to a lower makespan. (2) Total Energy Consumption (TEC): However, a higher number of machines (welders) results in increased energy consumption, which grows linearly. In this case, the energy consumption is inversely related to the processing time because, with reduced processing time, machines are active for shorter durations. This relationship between TEC and makespan indicates a conflict between the two objectives. When one objective is optimized, it often leads to a compromise in the other. Formulations are designed to address this conflict and find the best compromise solution for DHWSP.

$$\min F_1 = C_{max} = \max \{C_{f,i,n_s}\}, \forall i, f. \tag{1}$$

$$\min F_2 = TEC = E_B + E_S + E_I + E_O, \tag{2}$$

$$E_B = W_B \times C_{max}. \tag{3}$$

$$E_S = W_S \times \sum_{f=1}^{n_f} \sum_{i=1}^n \sum_{s=1}^{n_s} s_{f,i,s}. \tag{4}$$

$$E_I = W_I \times \sum_{f=1}^{n_f} \sum_{s=1}^{n_s} \sum_{i=2}^n \left(C_{f,i,s} - C_{f,i-1,s} - s_{f,i,s} - \frac{p_{f,i,s}}{\mu_{f,i,s}} \right) \tag{5}$$

$$E_O = [W_I \cdot (1 - K_w) + W_O \cdot K_w] \times \sum_{f=1}^{n_f} \sum_{s=1}^{n_s} \sum_{i=1}^n p_{f,i,s} \cdot (1 + 0.5 \cdot \ln(\mu_{f,i,s})) \tag{6}$$

The TEC in DHWSP comprises four components: (1) E_B : This represents the basic energy consumption, which includes energy usage for lighting and air conditioning. (2)

E_S : It accounts for the energy consumption associated with machine setup before the welding process begins. (3) E_I : This component represents the energy consumption when machines are in an idle state, waiting for the previous job to be completed before starting the next one. (4) E_O : This part corresponds to the welding energy consumption. It's noteworthy that each welder has a cooling period after welding, lasting for $K_w\%$ of the time. Furthermore, there's a non-linear relationship between the number of welders and E_O , which is modeled as a logarithmic function. The mathematical formulation of DHWSP is introduced as follows, outlining the problem's variables, constraints, and objectives.

$$\begin{cases} \min F_1 = C_{\max} \\ \min F_2 = TEC \end{cases} \tag{8}$$

subject to:

$$\sum_{f=1}^{n_f} \sum_{i=1}^n \mathbf{X}_{f,i,s} = 1, \forall s \tag{9}$$

$$\sum_{f=1}^{n_f} \sum_{s=1}^{n_s} \mathbf{X}_{f,i,s} = 1, \forall i \tag{10}$$

$$1 \leq \sum_{i=1}^n \sum_{s=1}^{n_s} \mathbf{X}_{f,i,s} \cdot \mu_{f,i,s} \leq m_{f,s}, \forall f, s \tag{11}$$

$$C_{f,i,1} = \sum_{i=1}^n \mathbf{X}_{f,i,1} \cdot \left(\frac{p_{f,i,1}}{\mu_{f,i,1}} + s_{f,i,1} \right), \forall f, i \tag{12}$$

$$C_{f,i,s+1} \geq C_{f,i,s}, \forall f, i, s = \{1, \dots, n_s - 1\} \\ + \sum_{i=1}^n \mathbf{X}_{f,i,s} \cdot \frac{p_{f,i,s}}{\mu_{f,i,s}}, \forall f, i, s = \{1, \dots, n_s - 1\} \tag{13}$$

$$C_{f,i+1,s} \geq C_{f,i,s} + \sum_{i=1}^n \mathbf{X}_{f,i,s} \cdot \left(\frac{p_{f,i,s}}{\mu_{f,i,s}} + s_{f,i,s} \right), \\ f, s, i = \{1, \dots, n - 1\} \tag{14}$$

$$\mathbf{X}_{f,i,s} \in \{0, 1\}, \forall f, i, s \tag{15}$$

where Eq. (8) represents the objective function, including both the makespan and TEC. Equations (9) and (10) specify that each job cannot be dispatched to two factories simultaneously. Equation (11) imposes a constraint ensuring that the number of welders used for each job does not exceed the capacity limit. Equation (12) defines the finish time for the first job at each factory. Equation (13) enforces that each operation can only start once the preceding operation is completed. Equation (14) guarantees that each machine can process only one job at a time after the setup phase. Finally, Eq. (15) pertains to binary decision variables, which are part of the problem formulation.

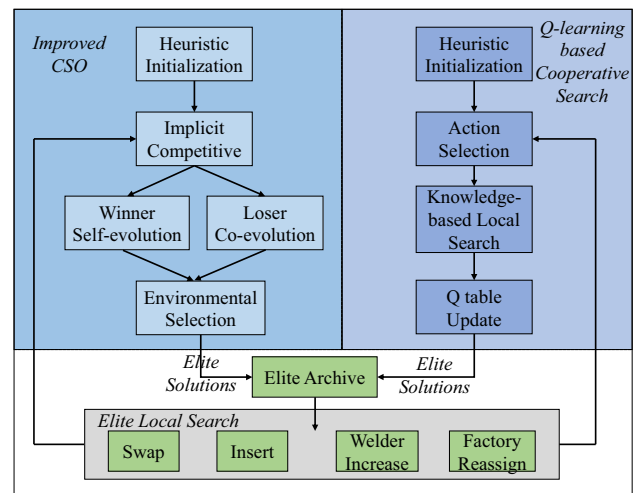


Fig. 2 LCCMO's Framework for solving DHWSP

Our approach: LCCMO

Framework of LCCMO

Figure 2 provides an overview of the LCCMO framework. The algorithm is structured around two populations, each serving a distinct role in the optimization process. The primary population, denoted as P_1 , utilizes a competitive and cooperative multi-objective optimizer to explore the decision space. The process begins with the initialization of P_1 using multiple heuristic rules. Subsequently, P_1 is split into two subpopulations: winners and losers, employing an implicit competitive strategy. The loser population learns from the winners, while the winners continue to evolve. Furthermore, the selection of offspring and parent for the next generation follows the NSGA-II strategy [43]. Additionally, an auxiliary population, labeled as P_2 , employs Q-learning-based multi-operator cooperative search to explore a limited range, complementing P_1 's broader but less focused search. Both P_1 and P_2 contribute their elite solutions to the archive Ω , where local search techniques are applied. Finally, the Pareto solutions are extracted from Ω as the final output.

Encoding and decoding

DHWSP represents a solution by two vectors and a welding scheduling matrix. Figure 3 shows DHWSP's encoding schema. JS is the job sequencing, FA is the factory assignment, and WN is the welding scheduling matrix, representing the number of welders utilized by each job in each stage.

Decoding schema: (1) Dispatching each job to the selected factory according to the FA . (2) Determining the job sequence in each factory according to the JS . (3) Calculating each job's real processing time in each stage according to the

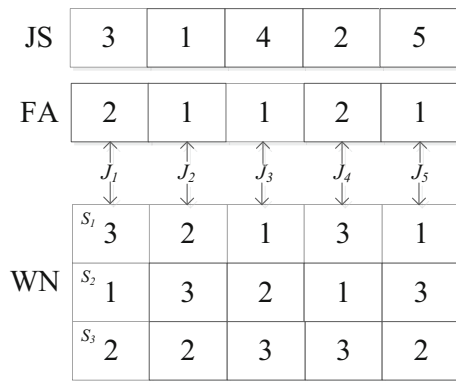


Fig. 3 An example for encoding schema for DHWSP

WN. (4) Calculating each job’s start and finish time under the constraints. (5) The idle time between two adjacent jobs can be obtained. (6) The makespan is the maximum completion time of all jobs in the final stage, and the TEC can be calculated according to Eq. 3.

Multi-rule cooperative initialization

High-quality initialization plays a crucial role in generating elite solutions with strong convergence, ultimately enhancing the search efficiency [37, 44, 45]. To achieve this, three heuristic rules have been designed to construct solutions with minimized makespan and TEC.

Rule1: Selecting more welders for each job can reduce processing time to decrease $Cmax$. Thus, initial JS and FA randomly, and each job in each stage selects the maximum number of welders.

Rule2: Each job chooses fewer welders can reduce welding energy consumption. Thus, randomly initial JS and FA, and every job in every stage selects the minimum number of welders.

Rule3: Balance each factory’s workload can reduce the makespan. The workload gap between all factories is smaller, the makespan is lower. Thus, initial JS and WN randomly. Calculate each job’s average processing time in each factory. Then, transferring the average time to selection probability P . The more average processing time, the lower P value. Finally, each job selects a factory with the minimum workload in the current scheduling. If several factories have the same workload, select the factory with P by roulette strategy.

The initial population is partitioned into four sub-populations, each consisting of $ps/4$ individuals. The first three subpopulations are initialized using the proposed heuristic rules to ensure high-quality solutions. In contrast, the fourth subpopulation is initialized randomly, introducing diversity into the population. In LCCMO, all populations are initialized using a cooperative initialization. It’s important to note that these

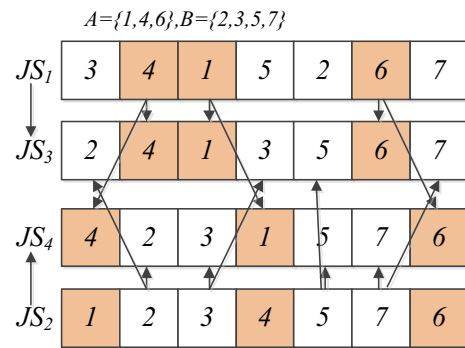


Fig. 4 Precedence operation crossover for JS

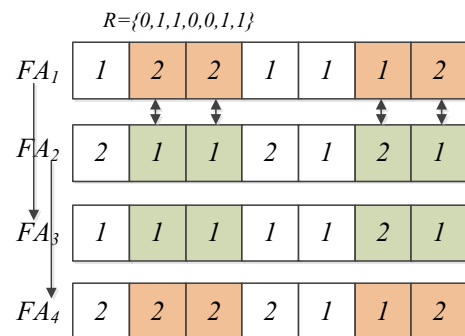


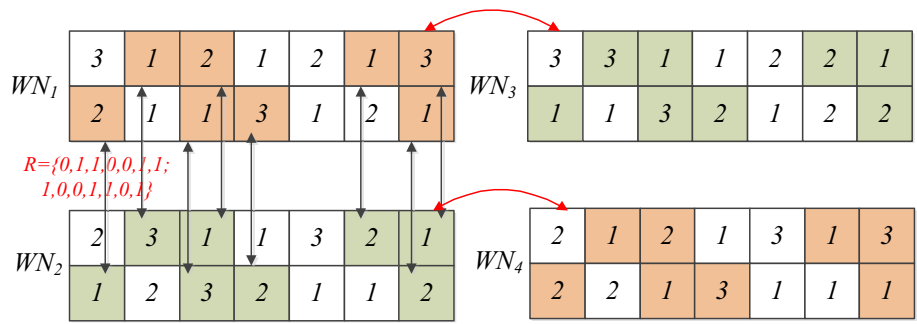
Fig. 5 Universal crossover for FA

two populations have different sizes, tailored to their respective roles in the algorithm.

Cooperative and competitive global search

In this section, we introduce a cooperative and competitive multi-objective optimizer, which is improved from CSO. This method is specifically tailored for addressing the DHWSP and involves three key enhancements over the original CSO [14]: (1) Improved Crossover Strategy: The original CSO, designed for continuous optimization problems, uses a vector difference crossover strategy that is not suitable for DHWSP. To adapt CSO for efficient global search in the context of DHWSP, two alternative crossover strategies are adopted: Precedence Operation Crossover (POX) [46] and Universal Crossover (UX) citeLR2022CAIE. (2) Implicit Many-to-Many Competition: In the original CSO, explicit one-to-one competition is employed, which can lead to suboptimal solutions learning from other suboptimal solutions. To overcome this limitation, an implicit many-to-many competition strategy is proposed. (3) Self-Evolution for Winners: In the original CSO, the winning solutions exclusively undergo mutation. In the improved CSO, a self-evolution strategy is introduced for the winner population, which accelerates the convergence of global search. The following

Fig. 6 Universal crossover for MN



sections provide detailed explanations of these improvements in the CSO, highlighting their adaptation to the DHWSP.

Algorithm 1 Implicit Competition.

```

1: Input: population  $\mathcal{P}$ , Fitness  $F$ 
2: Output: Winner  $\mathcal{W}$  and loser  $\mathcal{L}$  swarms
3: for  $i=1$  to  $ps-1$  do
4:   for  $j=i+1$  to  $ps$  do
5:     if  $\mathcal{P}_i < \mathcal{P}_j$  then
6:        $Dom(i, j) = 1$ .
7:     else if  $\mathcal{P}_i > \mathcal{P}_j$  then
8:        $Dom(j, i) = 1$ 
9:     end if
10:   end for
11: end for
12: for  $i=1$  to  $ps$  do
13:    $R_i = \sum_{j=2}^N Dom(i, j)$ .
14: end for
15: for  $i=1$  to  $ps$  do
16:   for  $j=1$  to  $ps$  do
17:     if  $i \neq j$  then
18:        $Dis(i, j) = ||F_i - F_j||$ .
19:     else
20:        $Dis(i, j) = \infty$ .
21:     end if
22:   end for
23: end for
24: for  $i=1$  to  $ps$  do
25:    $D_i = \frac{1}{\min(Dis(i, \cdot)) + 2}$ .
26:    $SF_i = R_i + D_i$ . //SPEA2 Fitness.
27: end for
28:  $I \leftarrow$  Sort  $SF$  by ascending.
29:  $\mathcal{W} \leftarrow \mathcal{P}(I(1 : ps/2), \cdot)$ .
30:  $\mathcal{L} \leftarrow \mathcal{P}(I(ps/2 + 1 : ps), \cdot)$ .

```

Implicit Competition: Algorithm 1 describes the detail of implicit competition. First, the number of being dominated R_i of each solution is calculated, regarding as a convergence metric. Second, each solution calculates the distance with the other solutions, and utilizes the closet distance’s reciprocal as diversity metric D_i . Finally, the comprehensive metric is calculated by adding R_i to D_i . The smaller SF_i , the better comprehensive performance. Sorting population with ascending SF . The first half is divided into the winner swarm, and the other half is the loser swarm.

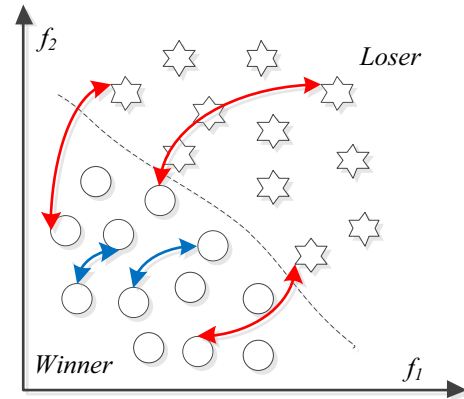


Fig. 7 Co-evolution and self-evolution

Co-evolution: In order to promote the loser to converge, the loser learns from the winner. For each loser, randomly selecting a winner, and adopting POX and UX to generate two offspring. The procedure of each vector’s crossover is shown in Figs. 4, 5, and 6. Moreover, before calculating the fitness of offspring, a repair operator is adopted for them to fix the infeasible solutions generated by crossover and mutation operators, because the number of welders in each stage in each factory is different. The out-of-range values in WN are fixed to the maximum number of welders. The co-evolution strategy is described in Fig. 7.

Self-evolution: As shown in Fig. 7, each winner randomly selects another winner and adopts crossover operators to generate two offspring. Moreover, each offspring employs three mutation strategies with probability P_m . (1) *JS* mutation: randomly choose two jobs and exchange their positions in *JS*. (2) *FA* mutation: randomly select a job and assign it to another factory. If infeasible solutions occur, generate a *FA* by random rule. (3) *WN* mutation: randomly select a job’s stage and change the number of utilized welders.

Environmental Selection: The improved CSO adopts NSGA-II’s environmental selection for population updating. First, the offspring is combined with the parent. Then, the fast non-dominated sorting is employed and the crowding distance strategy is adopted to keep diversity. Finally, the first ps_1

solutions are accepted and saved to next generation. In this work, $ps_1 = 100$.

Learning-driven cooperative local search

An agent-based population P_2 is employed to assist improved CSO to converge. P_2 focuses on searching elite solutions in a small range of objective space. P_2 sizes 20 and employed the Q-learning algorithm to select the optimal local search operator. Five knowledge-driven operators are designed to cooperatively search. The operators are introduced as follows:

\mathcal{N}_1 : Randomly choose two jobs and exchange their position to increase the diversity of population P_2 .

\mathcal{N}_2 : Find the critical path in critical factory, referring to [9]. Randomly select two critical jobs and exchange their positions to reduce makespan.

\mathcal{N}_3 : Choose two critical jobs randomly and insert the latter job into the front of the former job.

\mathcal{N}_4 : Randomly choose a critical job's stage and increase its number of used welders to reduce makespan. Because the more welders, the shorter the processing time, and the smaller makespan might be.

\mathcal{N}_5 : In order to reduce makespan, choose a critical job randomly and dispatch it to another factory.

Each solution employed a search operator to generate offspring. Moreover, Q-learning is adopted to improve the autonomy of cooperative search, and the design's details are as follows:

Q-table Design: The agent's states in the Q-learning framework are used to represent the success or failure of the current search operator. Consequently, each operator has two states, indicating whether it succeeded or failed to update the current solution. The agent itself has ten states, reflecting the different outcomes of the search operations. The available actions for the agent correspond to five local search operators. The structure of the Q-table, which guides the learning process, is presented in Fig. 8. For instance, when the agent is in its first state, denoted as S_1 , and selects action A_2 , if action A_2 fails to update the current solution, the agent transitions to the next state, S_4 . In this new state, the agent chooses the next action, and the process continues. This cycling of states within the agent's decision-making process ensures a continuous exploration of the solution space while adapting its actions based on past successes and failures.

Training Agent: The reward is 10 when the action successfully updates the current solution, but the reward is 0 when it fails. Then, the solution gives the reward feedback to the agent, and the Q-table is updated by following function:

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha [R_t + \gamma \max_{(S_{t+1}, A_t)} (Q(S_{t+1}, A_t) - Q(S_t, A_t))] \quad (16)$$

State	A ₁	A ₂	A ₃	A ₄	A ₅
S ₁ A ₁ S					
S ₂ A ₁ F					
S ₃ A ₂ S					
S ₄ A ₂ F					
S ₅ A ₃ S					
S ₆ A ₃ F					
S ₇ A ₄ S					
S ₈ A ₄ F					
S ₉ A ₅ S					
S ₁₀ A ₅ F					

Fig. 8 Detail of Q-table

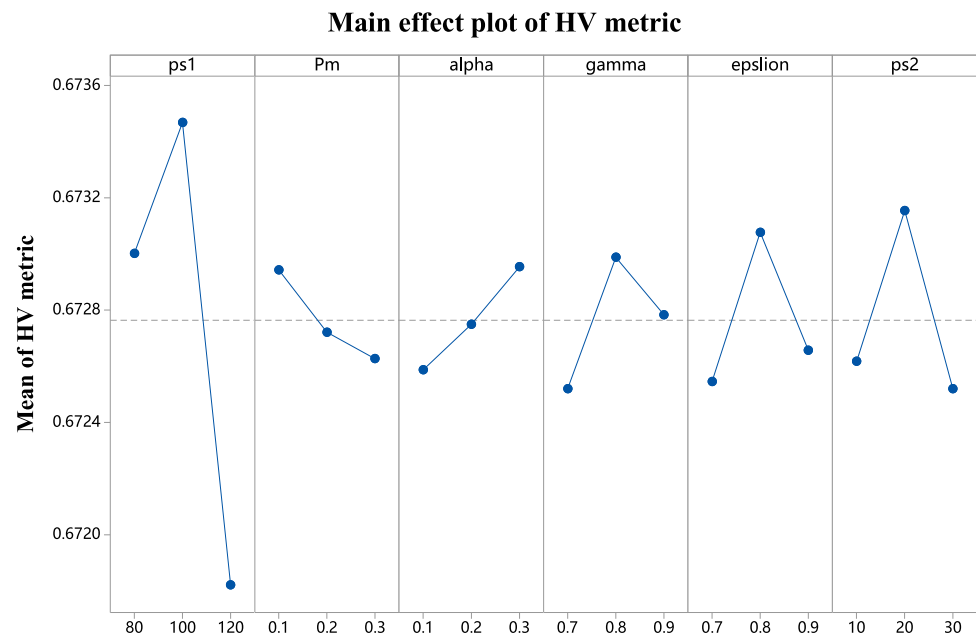
The environmental selection process in our approach aligns with NSGA-II [43]. The learning strategy comes into play when the subpopulation, with a size of 20, engages in cooperative search to explore the objective space. As depicted in Fig. 2, the learning mechanism is applied in the action selection phase, just before executing local search. Here is a breakdown of the steps involved in invoking the learning strategy: (1) Obtain the current state of the agent. (2) The agent selects the action with the highest q -value from the Q-table in the row corresponding to the current state. (3) The outcome of this action, whether it succeeds or fails, determines the agent's next state. (4) A reward is assigned based on the success or failure of the selected action. (5) The agent's Q-table is updated using the Q-learning equation. The additional computational cost incurred by the learning strategy primarily pertains to action selection and Q-table updates. Importantly, the computational complexity of this strategy is $O(N)$, which is notably smaller than the computational overhead of the evolutionary process.

Elite strategy

After the environmental selection, the two populations preserve their elite solutions to the elite archive Ω . In order to find more Pareto solutions, each elite solution randomly executes a local search operator \mathcal{N}_i , $i \in [1, 5]$. The elite archive deletes the repeat solutions and updates itself by saving the Pareto solutions. The final results of LCCMO are output by the elite archive.

Experimental results

Section “Our approach: LCCMO” has described the LCCMO in detail. Numerical experiments are employed to validate the effectiveness of LCCMO. Every algorithm is coded in MATLAB and tested on an CPU of Intel(R) Xeon(R) Gold 6246R with 3.4GHz and 384 G RAM.

Fig. 9 Main effects plot of HV metric

Instances and metrics

Wang [12] initially introduced the DHWSP and prepared a set of 20 instances for assessing algorithm performance. However, it appears that the benchmark dataset from [12, 13] is currently unavailable. Therefore, this study has created an additional set of 20 instances using the same methodology and parameters as described in [12, 13]. This approach ensures consistency and provides a reliable foundation for evaluating algorithms. The job number is selected from $n \in \{20, 40, 60, 80, 100\}$ and the number of factories ranges from $n_f \in \{2, 3\}$. The stage number is from $n_s \in \{2, 5\}$. The processing time $p_{f,i,s}$ ranges from [10, 50]. The setup time $s_{f,i,s}$ ranges from [1, 10]. The number of welders in each stage $\mu_{f,i,s}$ ranges from [2, 5]. The processing power is $W_O = 28$ kWh, the setup power is $W_S = 10$ kWh, the idle power is $W_I = 0.3$ kWh, the basic power is $W_B = 5$ kWh, and the welding duty cycle is $K_w = 0.8$.

Hypervolume (HV) [47], Generation distance (GD) [43], and Spread [48] are utilized to represent different algorithms' comprehensive performance, convergence, and diversity. The higher HV, the better comprehensive performance. The lower GD and Spread, the better convergence and diversity.

Experiment of parameters

LCCMO's parameter setting affects its performance for solving DHWSP. The LCCMO contains six parameters which are ICSO population size ps_1 , agent-based population size ps_2 , mutation rate P_m , learning rate α , discount factor γ , and greedy factor ϵ . A Taguchi approach of design-of-

experiment (DOE) [49] is adopted and an orthogonal array $L_{27}(3^6)$ is generated. The parameters' levels are indicated as follows: $ps_1 = \{100, 150, 200\}$; $P_m = \{0.1, 0.15, 0.2\}$; $\alpha = \{0.1, 0.2, 0.3\}$; $\gamma = \{0.7, 0.8, 0.9\}$; $\epsilon = \{0.7, 0.8, 0.9\}$; $ps_2 = \{10, 20, 30\}$. Every variant algorithm with different parameter settings runs 10 times independently. The same stop criterion is $\text{MaxNFEs} = 400 * n$. All metrics' average values for 10 runs are collected for Taguchi analysis. Figure 9 shows the HV metric's main parameters' effects plot. Based on the figure, the best configuration of parameter setting is that $ps_1 = 100$, $ps_2 = 20$, $\gamma = 0.8$, and $\epsilon = 0.8$. However, the P_m and α have no inflection point, thus their settings are discussed further. The control variable method is adopted. The first group is $P_m = \{0.1, 0.15, 0.2\}$, $\alpha = 0.2$, and other parameters are the best configuration in Fig. 9. The second group is $\alpha = \{0.1, 0.2, 0.3\}$, $P_m = 0.2$, and other parameters are the best setting. The results of HV metric among ten times run are shown in Fig. 10. In two groups, each parameter setting has no significant difference. Thus, the α and P_m are not sensitive and the α is set to 0.3 and P_m is 0.1.

Ablation experiment

Some variant algorithms are generated to validate the effectiveness of LCCMO's improvements. (1) CSO has no improvements utilized to compared with other variants; (2) CSO+I1: CSO with initial rule 1 and random rule; (3) CSO+I2 CSO with initial rule 2 and random rule; (4) CSO+I3 CSO with initial rule 3 and random rule; (5) CSO+HI is CSO with multi-rule cooperative initialization strategy; (6) CSO+L1: CSO with \mathcal{N}_1 ; (7) CSO+L2: CSO with \mathcal{N}_2 ; (8) CSO+L3: CSO with \mathcal{N}_3 ; (9) CSO+L4: CSO with \mathcal{N}_4 ;

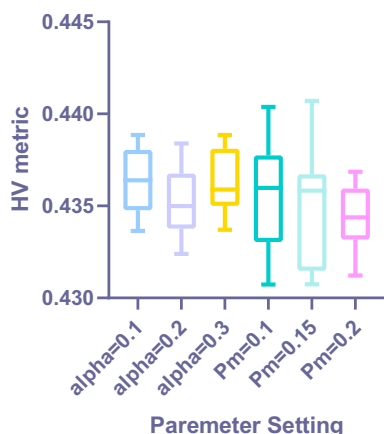


Fig. 10 Box plot for LCCMO with different settings of P_m and α

Table 1 Results of Friedman rank-and-sum test for all variant algorithms

MOEAs	HV		GD		Spread	
	Rank	<i>p</i> -value	Rank	<i>p</i> -value	Rank	<i>p</i> -value
CSO	12.95	0.0000	12.95	0.0000	10.9	0.0000
CSO+I1	6.25	0.0000	9.30	0.0105	1.60	
CSO+I2	4.20	0.0094	5.50	0.0000	2.55	0.4404
CSO+I3	9.70	0.0000	2.35		4.75	0.0105
CSO+HI	3.35	0.0563	11.50	0.0000	12.45	0.0000
CSO+L1	8.30	0.0000	4.30	0.1133	6.10	0.0002
CSO+L2	8.10	0.0000	5.05	0.0283	5.80	0.0006
CSO+L3	8.05	0.0000	4.70	0.0563	6.45	0.0000
CSO+L4	5.15	0.0008	4.20	0.1330	5.10	0.0044
CSO+L5	9.85	0.0000	5.45	0.0118	4.65	0.0132
CSO+LS	12.00	0.0000	10.70	0.0000	9.90	0.0000
ICSO	2.10	0.3717	9.00	0.0455	10.6	0.0000
LCCMO	1.00		6.00	0.0030	10.15	0.0000

Bold values of metrics' rank (such as HV, GD, Spread) indicating the algorithm with the best metric rank
 Italic indicates *p*-value smaller than 0.05, indicating the significance difference between current algorithm and the best algorithm

(10) CSO+L5: CSO with \mathcal{N}_5 ; (11) CSO+LS is CSO with multi-operator cooperative local search; (12) ICSO is CSO with initialization and local search strategies. (13) LCCMO is ICSO with an agent-based population. All algorithms independently run twenty times and the stop criterion is $\text{MaxNFES} = 400 * n \geq 2 * 10^4$. All algorithms are coded in MATLAB.

Tables S-I, S-II, S-III show all variant algorithmic statistical results of all metrics. Moreover, the **bold** values mean the best. The *p*-values indicate the significance between the best rank algorithm and the current variant. Based on observation, the LCCMO is better than all variant algorithms on HV metrics. However, LCCMO is worse than CSI+I3 and CSO+L4 for GD metric, and LCCMO is worse than CSI+I1

Table 2 Results of Friedman rank-and-sum test for all comparison algorithms

MOEAs	HV		GD		Spread	
	Rank	<i>p</i> -value	Rank	<i>p</i> -value	rank	<i>p</i> -value
MOEA/D	6.35	0.0000	6.65	0.0000	3.60	0.1243
NSGA-II	2.75	0.0104	2.65	0.0157	4.78	0.0011
SPEA2	2.30	0.0570	2.40	0.0404	5.08	0.0002
IMOEA/D	6.50	0.0455	6.00	0.0000	4.60	0.0027
MOWSA	3.95	0.0000	3.95	0.0000	3.10	0.4208
CMA	5.15	0.0000	5.35	0.0000	2.55	–
LCCMO	1.00	–	1.00	–	4.30	0.0104

for Spread metric. Furthermore, Table 1 lists the Friedman rank-and-sum test results. Some conclusions can be obtained as follows: (1) LCCMO has the best HV rank, indicating LCCMO has the best comprehensive performance. Moreover, *p*-values are smaller than 0.05 means that LCCMO is significantly better than other algorithms. (2) Comparing CSO, CSO+I1, CSO+I2, CSO+I3, and CSO+HI on all metrics indicates that proposed initial rules and their cooperation can increase the comprehensive performance. Compared to CSO+HI, CSO+I1, CSO+I2, and CSO+I3 obtain better GD and Spread because the signal rule makes the population uniformly covers a part of Pareto Front. (3) Comparing CSO, CSO+L1, CSO+L2, CSO+L3, CSO+L4, CSO+L5, CSO+LS, and ICSO demonstrate that the single local search operator can improve HV, GD, and Spread metric, and the random cooperation of each operator decreases the all metrics. However, combining multi-operator cooperative local search and multi-rule cooperative operators obtains the better HV metric. The key components are the cooperation of initialization rules and local search operators. (4) Comparing LCCMO and ICSO indicates the solvers cooperation can improve all metrics because different solvers can search different areas of objective space and the final non-dominated solution sets combine each solver's results to enhance convergence and diversity.

Comparison and discussions

To further evaluate the effectiveness of our approach, LCCMO is compared to the state-of-the-art algorithms including MOEA/D [50], NSGA-II [43], SPEA2 [51], IMOEA/D [12], MOWSA [13], and CMA [9]. The parameters are set with the best configuration in their references—crossover rate $P_c = 1.0$, mutation rate $P_m = 0.1$, and population size $ps = 100$ for all algorithms. The neighborhood updating range is $T = 10$ for MOEA/D, IMOEA/D, and CMA. The parameter setting of LCCMO is the same as Sect. 5.2. To conduct a fair comparison, all MOEAs run

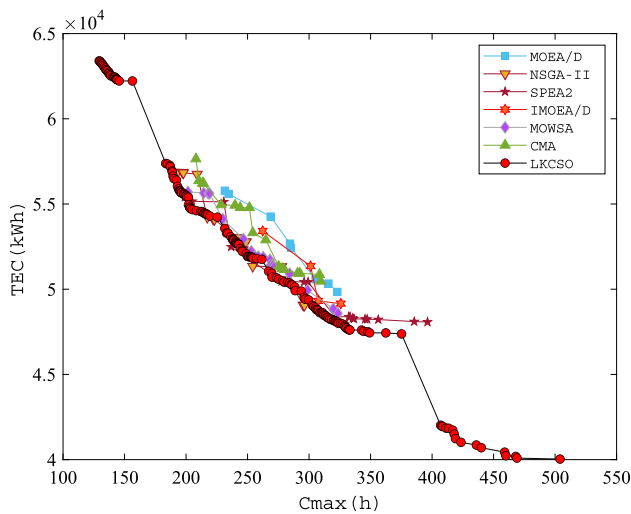


Fig. 11 Pareto Front comparison results on 20J2F2S

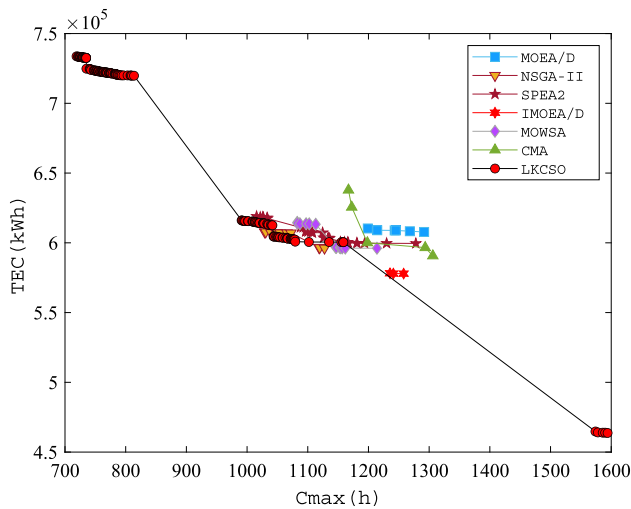


Fig. 12 Pareto Front comparison results on 100J3F5S

20 times and share the same stop criteria ($\text{MaxNFEs}=400 * n \geq 20,000$).

Tables S-IV, S-V, and S-VI record the statistical results of all comparison algorithms for three metrics. Symbols “- / = / +” means significantly inferior, equal, or superior to LCCMO. Meanwhile, the best value is marked in **bold**, and the last row of each table summarize the number of “- / = / +” symbols. As Tables S-IV and S-V show, LCCMO is significantly superior to compared algorithms for HV and GD metrics, indicating LCCMO has the best comprehensive performance and convergence for solving DHWSP. As for the Spread metric, Table S-VI indicates no significant difference between LCCMO and other algorithms, due to the complex and large search space of DHWSP. The local search step is too small for the WN matrix. Thus, the Pareto solutions are close to each others for all algorithms, resulting in the results of Spread. Table 2 indicates the results of the

Table 3 Comparison results by CPLEX solver and LCCMO

Instance	CPLEX		LCCMO	
	C_{\max}	TEC	C_{\max}	TEC
3J2F2S	12.25	1656.6	6	697.208
3J2F4S	25	2606.8	11.2	1635.176
4J2F2S	11.25	1704.5	5.2	815.656
4J2F4S	22	3878.6	12.57	1990.96
5J2F2S	18.25	2303.6	10.5	1203.824
5J2F4S	25	4626.5	14.7	2372.64
6J2F2S	21.5	2225.6	7.93	1170.32
6J2F4S	NA	5522.3	21.25	3252.136
7J2F2S	NA	2912.2	11.4	1501.376
7J2F4S	NA	6629.9	21.75	3791.168

Friedman rank-and-sum test for the comparison experiment. LCCMO ranks the best for HV and GD metrics, indicating LCCMO has the best comprehensive performance and convergence for solving DHWSP.

The effectiveness of LCCMO is contingent on its design. First, the proposed multi-rule cooperative initialization method yields a population with exceptional convergence and diversity by constructing solutions closely aligned with the lower bound. Second, the multi-operator cooperative local search enables solutions to explore different directions, thus preventing them from getting trapped in local optima. Furthermore, the collaboration of various solvers effectively amalgamates the search results from each solver, enhancing both the diversity and convergence of the final non-dominated solution set. In addition, Figs. 11 and 12 present a comparison of the Pareto front results for all algorithms, with the PF results selected based on the optimal HV value from twenty runs. As depicted in Figs. 11 and 12, LCCMO consistently yields Pareto solutions with superior convergence and comprehensive performance, underscoring its ability to effectively address DHWSP.

Model evaluation

The model is evaluated by CPLEX solver on ten generated small instances. The results are shown in Table 3. The CPLEX can solve instances 1–7 in 60s and the other cannot be solved because of the large amount of parameter. The proposed LCCMO obtains better results than mixed integer linear programming model because the cooperation of different solvers, multiple operators, and multiple rules. The CPLEX code can be downloaded from https://github.com/CUGLiRui/CPLEX_DHWSP.

Conclusions

This study introduced a competitive swarm optimizer driven by cooperative learning and knowledge for addressing energy-aware distributed heterogeneous welding shop scheduling problems. From the experimental results, several key conclusions have been drawn: (1) The implementation of multiple strategies and their cooperative interaction can significantly enhance the performance of algorithms. In the proposed algorithm, cooperative initialization, global search, and local search processes effectively contribute to population exploration. (2) Leveraging the cooperation of different solvers, including reinforcement learning and evolutionary algorithms, can effectively enhance the overall performance of the algorithms. (3) Building upon the principle of cooperation, the proposed algorithm outperforms the compared algorithms, showcasing its superior performance. In summary, this research demonstrates that a collaborative approach, involving multiple strategies and diverse solvers, can lead to more effective solutions for energy-aware distributed heterogeneous welding shop scheduling problems.

Some topics will be considered further. (i) Adopting LCCMO to other distributed heterogeneous shop scheduling problems; (ii) Considering dynamical events for DHWSP; and (iii) Studying end-to-end model for DHWSP.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s40747-023-01335-6>.

Acknowledgements This work was partly supported by the National Natural Science Foundation of China under Grant no. 62076225.

Data Availability The data and the source codes of LCCMO and Cplex that support the findings of this study are openly available at https://github.com/CUGLiRui/LCCMO_DHWSP.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Lu C, Gao R, Yin L, Zhang B (2024) Human-robot collaborative scheduling in energy-efficient welding shop. *IEEE Trans Ind Inform* 20(1):963–971
- Gao K, Huang Y, Sadollah A, Wang L (2020) A review of energy-efficient scheduling in intelligent production systems. *Complex Intell Syst* 6(2):237–249
- Zhao F, Xu Z, Wang L, Zhu N, Xu T, Jonrinaldi J (2023) A population-based iterated greedy algorithm for distributed assembly no-wait flow-shop scheduling problem. *IEEE Trans Ind Inf* 19(5):6692–6705
- Li X, Xiao S, Wang C, Yi J (2019) Mathematical modeling and a discrete artificial bee colony algorithm for the welding shop scheduling problem. *Memet Comput* 11(4):371–389
- Lu C, Gao L, Li X, Zheng J, Gong W (2018) A multi-objective approach to welding shop scheduling for makespan, noise pollution and energy consumption. *J Clean Prod* 196:773–787
- Li R, Gong W, Wang L, Lu C, Jiang S (2022) Two-stage knowledge-driven evolutionary algorithm for distributed green flexible job shop scheduling with type-2 fuzzy processing time. *Swarm Evol Comput* 74:101139
- Li R, Gong W, Wang L, Lu C, Zhuang X (2023) Surprisingly popular-based adaptive memetic algorithm for energy-efficient distributed flexible job shop scheduling. *IEEE Trans Cybern* 53(12):8013–8023
- Huang K, Li R, Gong W, Wang R, Wei H (2023) Brce: bi-roles co-evolution for energy-efficient distributed heterogeneous permutation flow shop scheduling with flexible machine speed. *Complex Intell Syst* 9:4805–4816
- Jj Wang, Wang L, Xiu X (2023) A cooperative memetic algorithm for energy-aware distributed welding shop scheduling problem. *Eng Appl Artif Intell* 120(105):877
- Li Y, Wang C, Gao L, Song Y, Li X (2021) An improved simulated annealing algorithm based on residual network for permutation flow shop scheduling. *Complex Intell Syst* 7(3):1173–1183
- Wang L, Jj Wang, Jiang E (2021) Decomposition based multiobjective evolutionary algorithm with adaptive resource allocation for energy-aware welding shop scheduling problem. *Comput Ind Eng* 162(107):778
- Wang G, Li X, Gao L, Li P (2021) Energy-efficient distributed heterogeneous welding flow shop scheduling problem using a modified moea/d. *Swarm Evol Comput* 62(100):858
- Wang G, Li X, Gao L, Li P (2022) An effective multi-objective whale swarm algorithm for energy-efficient scheduling of distributed welding flow shop. *Ann Oper Res* 310(1):223–255
- Zhang X, Zheng X, Cheng R, Qiu J, Jin Y (2018) A competitive mechanism based multi-objective particle swarm optimizer with fast convergence. *Inf Sci* 427:63–76
- Lu C, Xiao S, Li X, Gao L (2016) An effective multi-objective discrete grey wolf optimizer for a real-world scheduling problem in welding production. *Adv Eng Softw* 99:161–176
- Lu C, Gao L, Li X, Xiao S (2017) A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry. *Eng Appl Artif Intell* 57:61–79
- Li X, Lu C, Gao L, Xiao S, Wen L (2018) An effective multiobjective algorithm for energy-efficient scheduling in a real-life welding shop. *IEEE Trans Ind Inf* 14(12):5400–5409
- Rao Y, Meng R, Zha J, Xu X (2020) Bi-objective mathematical model and improved algorithm for optimisation of welding shop scheduling problem. *Int J Prod Res* 58(9):2767–2783
- Wang C, Zhao L, Li X, Li Y (2022) An improved grey wolf optimizer for welding shop inverse scheduling. *Comput Ind Eng* 163(107):809

20. Tian Y, Zheng X, Zhang X, Jin Y (2020) Efficient large-scale multiobjective optimization based on a competitive swarm optimizer. *IEEE Trans Cybern* 50(8):3696–3708
21. Wang X, Zhang B, Wang J, Zhang K, Jin Y (2022) A cluster-based competitive particle swarm optimizer with a sparse truncation operator for multi-objective optimization. *Swarm Evol Comput* 71(101):083
22. Gu Q, Liu Y, Chen L, Xiong N (2022) An improved competitive particle swarm optimization for many-objective optimization problems. *Expert Syst Appl* 189(116):118
23. Huang W, Zhang W (2022) Multi-objective optimization based on an adaptive competitive swarm optimizer. *Inf Sci* 583:266–287
24. Cheng R, Jin Y (2015) A competitive swarm optimizer for large scale optimization. *IEEE Trans Cybern* 45(2):191–204
25. Mohapatra P, Nath Das K, Roy S (2017) A modified competitive swarm optimizer for large scale optimization problems. *Appl Soft Comput* 59:340–362
26. Wang X, Zhang K, Wang J, Jin Y (2022) An enhanced competitive swarm optimizer with strongly convex sparse operator for large-scale multiobjective optimization. *IEEE Trans Evol Comput* 26(5):859–871
27. Huang C, Zhou X, Ran X, Liu Y, Deng W, Deng W (2023) Co-evolutionary competitive swarm optimizer with three-phase for large-scale complex optimization problem. *Inf Sci* 619:2–18
28. Ge Y, Chen D, Zou F, Fu M, Ge F (2022) Large-scale multiobjective optimization with adaptive competitive swarm optimizer and inverse modeling. *Inf Sci* 608:1441–1463
29. Liu S, Lin Q, Li Q, Tan KC (2022) A comprehensive competitive swarm optimizer for large-scale multiobjective optimization. *IEEE Trans Syst Man Cybern Syst* 52(9):5829–5842
30. Qi S, Zou J, Yang S, Jin Y, Zheng J, Yang X (2022) A self-exploratory competitive swarm optimization algorithm for large-scale multiobjective optimization. *Inf Sci* 609:1601–1620
31. Chen X, Tang G (2022) Solving static and dynamic multi-area economic dispatch problems using an improved competitive swarm optimization algorithm. *Energy* 238(122):035
32. He C, Li M, Zhang C, Chen H, Li X, Li J (2022) A competitive swarm optimizer with probabilistic criteria for many-objective optimization problems. *Complex Intell Syst* 8(6):4697–4725
33. Nguyen BH, Xue B, Zhang M (2022) A constrained competitive swarm optimiser with an svm-based surrogate model for feature selection. *IEEE Trans Evol Comput* 28(1):2–16
34. Musikawan P, Kongsorot Y, Muneesawang P, So-In C (2022) An enhanced obstacle-aware deployment scheme with an opposition-based competitive swarm optimizer for mobile wsns. *Expert Syst Appl* 189(116):035
35. Cao S, Li R, Gong W, Lu C (2023) Inverse model and adaptive neighborhood search based cooperative optimizer for energy-efficient distributed flexible job shop scheduling. *Swarm Evol Comput* 83(101):419
36. Zhao F, Zhang H, Wang L (2023) A pareto-based discrete jaya algorithm for multiobjective carbon-efficient distributed blocking flow shop scheduling problem. *IEEE Trans Ind Inf* 19(8):8588–8599
37. Li R, Gong W, Lu C, Wang L (2023) A learning-based memetic algorithm for energy-efficient flexible job-shop scheduling with type-2 fuzzy processing time. *IEEE Trans Evol Comput* 27(3):610–620
38. Zhao F, Di S, Wang L (2023) A hyperheuristic with q-learning for the multiobjective energy-efficient distributed blocking flow shop scheduling problem. *IEEE Trans Cybern* 53(5):3337–3350
39. Zhao F, Jiang T, Wang L (2023) A reinforcement learning driven cooperative meta-heuristic algorithm for energy-efficient distributed no-wait flow-shop scheduling with sequence-dependent setup time. *IEEE Trans Ind Inf* 19(7):8427–8440
40. Li R, Gong W, Wang L, Lu C, Dong C (2024) Co-evolution with deep reinforcement learning for energy-aware distributed heterogeneous flexible job shop scheduling. *IEEE Trans Syst Man Cybern Syst* 54(1):201–211
41. Li R, Gong W, Wang L, Lu C, Pan Z, Zhuang X (2023) Double dqn-based coevolution for green distributed heterogeneous hybrid flowshop scheduling with multiple priorities of jobs. *IEEE Trans Autom Sci Eng*. <https://doi.org/10.1109/TASE.2023.3327792>
42. Li P, Tang H, Hao J, Zheng Y, Fu X, Meng Z (2022) Erl-re2: Efficient evolutionary reinforcement learning with shared state representation and individual policy representation. [arXiv:2210.17375](https://arxiv.org/abs/2210.17375). <https://api.semanticscholar.org/CorpusID:253237390>. Accessed 30 Jun 2023
43. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans Evol Comput* 6(2):182–197
44. Li R, Gong W, Lu C (2022) A reinforcement learning based rmoea/d for bi-objective fuzzy flexible job shop scheduling. *Expert Syst Appl* 203(117):380
45. Li R, Gong W, Lu C (2022) Self-adaptive multi-objective evolutionary algorithm for flexible job shop scheduling with fuzzy processing time. *Comput Ind Eng* 168(108):099
46. Wang H, Li R, Gong W (2023) Minimizing tardiness and makespan for distributed heterogeneous unrelated parallel machine scheduling by knowledge and pareto-based memetic algorithm. *Egypt Inform J* 24(3):100383
47. While L, Hingston P, Barone L, Huband S (2006) A faster algorithm for calculating hypervolume. *IEEE Trans Evol Comput* 10(1):29–38
48. Wang YN, Wu LH, Yuan XF (2010) Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure. *Soft Comput* 14(3):193–209
49. Van Nostrand RC (2002) Design of experiments using the Taguchi approach: 16 steps to product and process improvement. *Technometrics* 44(3):289–289
50. Zhang Q, Hui L (2007) Moea/d: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(6):712–731
51. Zitzler E, Laumanns M, Thiele L (2002) Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. Technical Report Gloriastrasse

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.