Advances in Production Engineering & Management
Volume 18 | Number 3 | September 2023 | pp 271–287
https://doi.org/10.14743/apem2023.3.472
ISSN 1854-6250
Journal home: apem-journal.org
Original scientific paper

An improved multi-objective Wild Horse optimization for the dual-resource-constrained flexible job shop scheduling problem: A comparative analysis with NSGA-II and a real case study

Peng, F.[a,b], **Zheng, L.**[a,*]

[a]Department of Industrial Engineering, Tsinghua University, P.R. China
[b]CRRC Academy Co. Ltd, Beijing, P.R. China

ABSTRACT

The equipment manufacturing industry needs skilled workers to operate a specific set of machines following process specifications. Optimizing machine and worker assignments to achieve maximum efficiency is a critical problem for workshop managers. This paper investigates a multi-objective dual-resource-constrained flexible job shop scheduling problem. An improved wild horse optimization (IWHO) algorithm is developed to simultaneously optimize three objectives: makespan, maximum machine workload, and total machine workload. To evaluate the quality of individuals in multi-objective optimization, the Pareto fast non-dominated sorting method is used, and the crowding distance is calculated. To update the algorithm's solution, the crossover and mutation operations are used. Further, a local neighborhood search strategy is employed to enhance searchability and avoid trapping into the local optima. The benchmark of the flexible job shop scheduling problem is extended to create test instances, and the performance of the suggested IWHO algorithm is evaluated compared with the NSGA-II. The computational results show that the IWHO algorithm provides a non-dominated efficient set within a reasonable running time. Furthermore, a buffers and chain coupler assembly process is designed to analyze the practical value of the IWHO algorithm. The proposed solutions can be used to generate daily schedules for managing machines, workers, and production cycles.

ARTICLE INFO

Keywords:
Dual resource constraints;
Flexible job shop scheduling;
Wild horse optimization;
Local search;
Multi-objective optimization;
NSGA-II;
Benchmark analysis

**Corresponding author:*
lzheng@mail.tsinghua.edu.cn
(Zheng, L.)

Article history:
Received 12 July 2023
Revised 25 October 2023
Accepted 29 October 2023

1. Introduction

Scheduling is crucial for increasing the efficiency of manufacturing resources in contemporary production systems. In the last few decades, numerous scheduling problems, including job shop scheduling problem (JSP) [1], flexible job shop scheduling problem (FJSP) [2-4], etc., have been exhaustively studied against the growing demand for flexible and customized manufacturing, typically characterized by short life cycles, small lot sizes, and changing product mixes [5]. In standard FJSP, only the machine flexibility is taken into account, which means that the operations of the jobs can be performed by more than one machine from the group of available machines [6]. However, physical manufacturing resources are not just machines and also cover human resources, materials, transferring tools, etc. [7]. Multiple resources are introduced into the production systems to make the scheduling scenario more realistic. Dual resources-constrained (DRC) systems, in which the capacity of both machines and human operators is limited, are studied in depth [8]. DRC systems are more complex than systems with a single resource, and the scheduling process must account for numerous additional technical challenges.

In this paper, we investigated a dual-resource constraint flexible job shop scheduling problem (DRCFJSP) motivated by the example of an engineering equipment manufacturer's assembled workshop. With the development of Industry 4.0 technologies, more and more factories have started to promote intelligent manufacturing [9]. In the actual workshop manufacturing system, constraints on production resource factors such as fixtures, transport equipment, and laborers exist in addition to equipment constraints. Due to the specific requirements of the production structure, the engineering equipment assembly workshop requires a substantial amount of labor. Robots are incapable of performing a thorough job of installing electric components into spatially complex curved surfaces. In addition, multiple workers are organized to execute the tasks sequentially. In such a case, the DRCFJSP must be considered when determining the operational worker and assistant machine orders.

In the production process with human-computer interaction, the generated schedule is challenging to execute if the worker factor is not considered or the worker efficiency is regarded as a fixed value [10]. The human factor largely determines the performance of the production system with manual operation, which makes the scheduling decision of the system more challenging due to the variability of personnel [11]. The processing performance of the production system is closely related to the operational efficiency of workers [12]. The production process in an assembled workshop environment is becoming increasingly reliant on the efficient management of skilled workers, and the unreasonable allocation of personnel will also result in rising wage costs. In addition, due to the recurring COVID-19 epidemic, the manufacturing industry has difficulty recruiting more skilled workers. Assigning a limited number of skilled workers to appropriate positions can increase production efficiency and ensure the plan's smooth progression.

The DRCFJSP has been extensively studied using various approaches such as mixed-integer programming, constrained programming, heuristics, and meta-heuristics. However, there is a need to devise practical alternate solutions for this problem. To address this issue, this paper proposes an enhanced version of the Wild Horse Optimization (WHO) algorithm tailored to solve the DRCFJSP. The primary contributions of this research are listed as follows.

- A multi-objective approach is proposed for the DRCFJSP that simultaneously minimizes the makespan, the maximum machine workload, and the total machine workload.
- The wild horse optimization algorithm is discretized using the crossover and mutation operators, and a local search strategy is added to avoid local optima.
- A real-world test instance is provided for assembling the buffers and chain couplers. And the performance of the designed algorithm is also analyzed by solving the practical instance.

2. Literature review

Our research aims to design an alternative method to solve the DRCFJSP. We discussed the closest two sides in the existing works: (i) DRCFJSP; and (ii) the involved wild horse optimization algorithm. For the flexible job shop scheduling problem, the readers can refer to the comprehensive surveys [2, 13].

It is essential to find a solution to the multi-objective flexible job shop scheduling problem (MOFJSP), considering various production resources for supporting the development of the manufacturing industry in intelligence, flexibility, and personalized customization. The MOFJSP problem is a complex NP-hard problem, which is widely used in various fields such as aviation equipment, the shipbuilding industry [14], agricultural machinery manufacturing [15], semiconductor manufacturing [16], etc. Among that, the DRCFJSP is studied widely from multiple perspectives. The DRCFJSP is introduced considering machine capacity and labor capacity and can be regarded as dual resource-constrained (DRC) systems that commonly exist in real-world situations [8]. In the DRCFJSP researches, the resources assignment problem and the operation sequence problem are studied concurrently. Nonetheless, the DRCFJSP is NP-hard in the strong sense since its simplified form, FJSP, is NP-hard.

In that case, various heuristics were designed to resolve the DRCFJSP. Lei and Guo [17] studied DRCFJSP to optimize the makespan based on variable neighborhood search. Then, they investigated a DRC job shop scheduling problem with interval processing time and heterogeneous resources and developed a dynamical neighborhood search algorithm [18]. Besides, simulated annealing (SA) and vibration damping optimization were designed to solve the DRCFJSP [19]. Moreover, some new population-based optimization algorithms were also presented to solve it, including genetic algorithm (GA) [20], fruit fly optimization algorithm [21], bat algorithm [22], etc.

Due to the increase in computer power, constraint programming techniques were developed to solve the DRCFJSP [23]. However, operations can not be processed if the workers are not available or lack the requisite skills. Wu *et al.* [24] considered the worker's learning ability in the DRCFJSP and suggested a genetic algorithm to obtain the optimal solutions. Besides, the loading and unloading time of the fixtures was introduced into the DRCFJSP, and a non-dominated sorting genetic algorithm II was proposed to minimize the makespan and the setup time [25]. Workers ' fatigue cannot be ignored when considering the DRCFJSP in the casting workshop. And Tan *et al.* discussed a fatigue-conscious DRCFJSP with the support of the NSGA-II, minimizing the maximum worker fatigue and makespan [26]. Due to the requirement for product customization and on-time delivery in make-to-order companies, the due date-related criteria (the mean tardiness) were introduced into the DRCFJSP, except for the makespan [27]. Recently, sequencing flexibility has been studied in DRCFJSP for minimizing the triple objectives, including makespan, maximal worker workload, and weighted tardiness [28]. Based on the above analysis, most of the current works focus on developing the heuristics in solving the DRCFJSP. Meanwhile, the multi-objective optimization approaches are insufficient, especially in triple objectives. Therefore, developing an alternative solution approach is necessary to solve the DRCFJSP.

The wild horse optimizer (WHO) is a new population-based optimization algorithm developed by Naruei and Keynia [29]. The WHO imitates the social lift behavior of wild horses, especially for the decency behavior of the horse. Compared to the existing algorithms, such as GA, particle swarm optimization, salp swarm algorithm, etc., the WHO performs better in solving the CEC2019 test functions. Since then, the WHO attracted the attention of many scholars. Li *et al.* [30] suggested four strategies to improve the optimization capability of the WHO and performed a demonstration of the improvement on CEC2017 and CEC2021. Vasanthkumar *et al.* [31] introduced deep learning to enhance the WHO for the estimation of the state of charge in hybrid electric vehicles. Ali *et al.* [32] used the WHO to optimize the distributed generation to increase the reliability, stability, and security of the electrical power systems. Alphonse *et al.* [33] adopted the WHO to allocate electric vehicle charging stations and photovoltaic energy resources in a smart grid simultaneously. Milovanović *et al.* [34] studied the multi-objective energy management problem using the WHO. Based on the applications of the WHO, it is rare to apply the WHO to scheduling problems. In this paper, we tried to design an improved WHO algorithm to fill the research gap in discretizing the WHO for solving DRCFJSP.

3. Problem Description

The DRCFJSP is described as follows: n jobs $J = \{1,2,\cdots,n\}$ needs to be processed on m machines $M = \{1,2,\cdots,m\}$ accompanied by w workers $W = \{1,2,\cdots,w\}$. Each job includes multiple operations, and the processing sequence of the operations must be followed. Each machine can only perform one operation at the same time. The operations can be processed on one of the available machines, and workers with the ability to operate the corresponding machines are assigned according to the process requirements. The worker number w should be less than or equal to the machine number m, and the processing time is determined by the operation efficiency of the selected workers and the capacity of the assigned machine. Each worker can operate more than one machine, and the machining operation efficiency of each worker is different.

To evaluate the effectiveness of the scheduling scheme in terms of production efficiency and machine utilization, we have selected three objectives for optimization, which include minimizing the makespan, maximal machine workload, and total machine workload. Among these goals, the makespan is crucial in determining production efficiency as it directly relates to the completion time of each job. The maximal machine workload identifies the bottleneck machine, and reducing it can help improve the workshop's production efficiency. Additionally, the total machine workload is closely related to machine idle losses and energy consumption. The scheduling process aims to provide an optimal resource allocation scheme and process sequence while adhering to process constraints and dual resource capacity constraints. Achieving the optimal schedule involves minimizing the aforementioned three objectives.

Table 1 Notation

J	set of jobs, $J = \{1,2,\cdots,n\}$
M	set of machines, $M = \{1,2,\cdots,m\}$
W	set of workers, $W = \{1,2,\cdots,w\}$
W_k	set of eligible workers that can operate machine k, $W_k \subseteq W$
J_i	set of operations of job i ($i \in J$), $J_i = \{1,2,\cdots,n_i\}$
O_{ij}	the j-th operation of job i ($i \in J, j \in J_i$)
C_i	the complete time of job i ($i \in J$)
s_{ij}	the starting time of the operation O_{ij}
c_{ij}	the completion time of the operation O_{ij}
p_{ijkr}	the processing time of operation O_{ij} is processed on machine k by worker r
Δ	a big number
x_{ijkr}	binary variable: 1, if the operation O_{ij} is processed on machine k by worker r; 0, otherwise
y_{ghijk}	binary variable: 1, if both operation O_{ij} and operation O_{gh} are processed on machine k, and O_{gh} is earlier than O_{ij}; 0, otherwise
z_{ghijr}	binary variable: 1, if both operation O_{ij} and operation O_{gh} are processed by worker r, and O_{gh} is earlier than O_{ij}; 0, otherwise
$T_{w_l}^S$	the idle start time for worker w_l
$T_{w_l}^E$	the idle end time for worker w_l

The notation is defined in Table 1. The mixed-integer programming model of the DRCFJSP with the triple objectives is given as follows.

$$\min f_1 = \max_{i\in N}\{C_i\} \tag{1}$$

$$\min f_2 = \max_{k\in M}\{\textstyle\sum_{r\in W_k}\sum_{i\in N}\sum_{j\in J_i} p_{ijkr}x_{ijkr}\} \tag{2}$$

$$\min f_3 = \textstyle\sum_{k\in M}\sum_{r\in W_k}\sum_{i\in N}\sum_{j\in J_i} p_{ijkr}x_{ijkr} \tag{3}$$

s.t.

$$\textstyle\sum_{k\in M}\sum_{r\in W_k} x_{ijkr} = 1 \ \ \forall j \in J_i, i \in N \tag{4}$$

$$c_{ij} \geq s_{ij} + \textstyle\sum_{k\in M}\sum_{r\in W_k}(p_{ijkr}x_{ijkr}) \ \forall j \in J_i, i \in N \tag{5}$$

$$c_{ij} \leq s_{i(j+1)} \ \forall j \in J_i \backslash \{n_i\}, i \in N \tag{6}$$

$$y_{ghijk} + y_{ijghk} \leq 1 \ \forall h \in J_g, j \in J_i, g, i \in N, k \in M \tag{7}$$

$$z_{ghijr} + z_{ijghr} \leq 1 \ \forall \ h \in J_g, j \in J_i, g, i \in N, r \in W \tag{8}$$

$$c_{gh} \leq s_{ij} + \triangle\,(3 - \textstyle\sum_{r\in W_k} x_{ghkr} - \sum_{r\in W_k} x_{ijkr} - y_{ghijk}) \forall h \in J_g, j \in J_i, k \in M, g, i \in N \tag{9}$$

$$c_{gh} \leq s_{ij} + \triangle\,(3 - \textstyle\sum_{k\in M} x_{ghkr} - \sum_{k\in M} x_{ijkr} - z_{ghijr}) \forall h \in J_g, j \in J_i, r \in W, g, i \in N \tag{10}$$

$$C_{max} \geq c_{ij} \ \forall j \in J_i, i \in N \tag{11}$$

$$x_{ijkr}, y_{ghijk}, z_{ghijk} \in \{0,1\} \forall h \in J_g, j \in J_i, r \in W_k, k \in M, g, i \in N \tag{12}$$

The objective functions consist of three parts. Eq. 1 minimizes the maximum makespan, Eq. 2 minimizes the maximum machine workload, and Eq. 3 minimizes the overall machine load. Constraint Eq. 4 ensures that each operation of a job can only be allocated to a single worker and performed simultaneously by a single piece of machine. The processing completion time of each operation is the sum of its earliest starting time and its processing time, as shown by Constraint Eq. 5. There is a processing sequence restriction between two adjacent job operations, according to constraint Eq. 6. The conflicting sequence of operations allocated to a machine is avoided by constraint Eq. 7. The conflicting sequence of operations allocated to a worker is avoided by constraint Eq. 8. Constraints Eqs. 9 to 10 establish the respective association between x, y, and z. Makespan is specified by constraint Eq. 11. The domains of three decision variables are set by constraint Eq. 12.

4. The improved wild horse optimization algorithm

Due to the intractable nature of the problem under study and the necessity of finding multiple Pareto optimal solutions, an improved multi-objective wild horse optimization algorithm is proposed. The wild horse optimizer (WHO) is a new meta-heuristic algorithm for solving continuous optimization problems [29]. This population-based optimization algorithm imitates the behavior of non-territorial wild horses to find the optimal solution in the solution space. Specifically, group behaviors, grazing, mating, dominance, and leadership are utilized to design search operators. As with other optimization algorithms, the WHO starts with an initial random population with N_{pop} members. Moreover, the G ($G = \lceil N_{pop} \times PS \rceil$, where PS is the percentage of stallions in the population) leaders (stallions) are selected based on their fitness values to determine the groups, and the remaining members ($N_{pop} - G$) are divided equally among these groups. The stallion can be regarded as the center of the grazing area, and the group members are searched around the center. When the position of the stallion is in a different direction, its members are attracted by using the position updating equation. Compared to other animals, the behavior of separating foals from the group and mating them prevents the father from mating with the daughter or siblings. In the WHO, the crossover operator is used to simulate the behavior of the departure and mating of horses. According to the horse mating behavior, the prematurity of the population can be prevented. The group leader must lead the group to a suitable area. The water hole can be taken as this suitable area. And the leaders must compete for this water hole so that the dominating group can use this water hole and other groups are not allowed to use the water hole. The WHO simulates this behavior to update the stallion's location and expedite the search process. In the initial iteration, the group's leaders are selected at random. In later iterations of the WHO, leaders are chosen based on their fitness. If one member of the group is superior to the group leaders, their positions will be switched. If the termination criteria are met, the algorithm is stopped, and the best solution is output. Otherwise, the algorithm is iterated.

The standard WHO algorithm was created to solve continuous optimization problems. Here, discrete optimization is required for the investigated DRCFJSP. The section introduces the specifics of the improved WHO algorithm. The enhanced components are designed based on the peculiarities of our problem.

4.1 Encoding and decoding scheme

In our algorithm, the encoding scheme adopts a three-vector string [21] to express the operation sequence (OS), machine assignment (MA), and worker assignment (WA) in a solution, as shown in Fig. 1. The first vector OS is the order of the jobs' operations. The number of occurrences for a specific job index corresponds to its operation number. The second vector MA and the third vector WA determine the corresponding suitable machine and worker for the specific operation of the job in OS. The lengths of the three vectors are $\sum_{i \in N} n_i$.

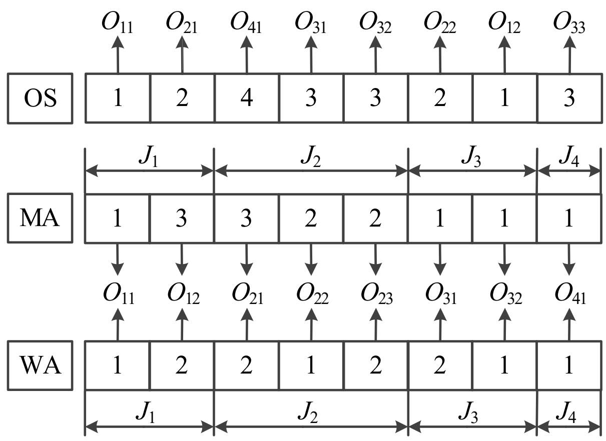


Fig. 1 Solution encoding scheme

Based on the above representation, the corresponding schedule can be obtained via the following five steps of the decoding procedure.

1) Take the operation O_{ij} from the OS one by one from left to right and find the corresponding machine k and worker r;
2) Based on the processing time p_{ijkr} of operation O_{ij} handling by machine k and worker r, the free interval $[T_k^S,T_k^E]$ of machine k and the free interval $[T_r^S,T_r^E]$ of worker r can be determined, in which T^S and T^E are the starting time and the ending time of the free interval, respectively.
3) Calculate the starting time T_{ij}^S of the operation O_{ij} via the Eq. 13;
$$T_{ij}^S = max(T_{i(j-1)}^E, T_k^S, T_r^S) \tag{13}$$
4) Evaluate whether the operation O_{ij} can be processed earlier. If $T_{ij}^S + P_{ijkr} \leq min(T_k^E, T_r^E)$, the operation O_{ij} can be inserted into the available free interval forward. If the operation O_{ij} can not be inserted into the existing free interval, it is appended to the processing list of the operations assigned to machine k. And the starting time of operation O_{ij} is the maximum between the ending time of the last operation assigned to machine k and the ending time of the last operation assigned to worker r;
5) Repeat the above steps 1-4 until all operations of the jobs have been processed.

4.2 Population initialization

To improve the WHO algorithm's population initialization, multiple groups of individuals should be initialized, in line with the algorithm's design philosophy. The exchange of information between these groups can expedite the search procedure, but the quality of initial solutions can also enhance the algorithm's ability to explore and exploit. To generate the OS, our algorithm selects a processing operation randomly from the list of candidate operations while maintaining the precedence relationship between the job operations. The MA and WA are then generated by assigning the most efficient machine and worker, respectively.

4.3 Pareto ranking and crowding distance

The fast-non-dominated sorting approach proposed in non-dominated sorting genetic algorithm II (NSGA-II) [35] is adopted to construct the set of Pareto optimal solutions. The sorting approach distinguishes all nondominated solutions based on the individuals' objective values and directs the population toward the set of Pareto optimal solutions. We consider two entities for each individual: 1) $n(i)$ represents the number of the solutions that dominate the individual i, and 2) $S(i)$, a set of individuals that the individual i dominates. The detailed computation process for the two entities can be found in reference [35]. Individuals with rank one belong to the non-dominated set which is also called the Pareto-optimal set. Once the non-dominated sorting is finished, a new population of size N_{pop} is formed with individuals of different non-dominated fronts.

After the Pareto ranking of the individuals, the priority relationship of the individuals located on the same front should be analyzed. In our improved WHO algorithm, individual sorting is used to guide the updating of the remaining individuals with the information of the best one. And the crowding distance is used to sort the individuals of the same front. The crowding distance of an individual from the nondominated solution set is the sum of the differences in the objective values of its two adjacent individuals. Here, we set the boundary value of the crowding distance to infinity for selecting other individuals at the next iteration. Before calculating the crowding distance, the individuals of various Pareto fronts are sorted based on their objective values. The following Eq. 14 is used to assign the crowding distance D_i to the individual i.

$$n_d = \sum_{h}^{H} \frac{f_h(i+1) - f_h(i-1)}{f_h^{\max} - f_h^{\min}} \tag{14}$$

In the above equation, H is the number of objective values at the same front, f_h^{max} is the maximum value of the objective function f_h, f_h^{min} is the minimum value of the objective function f_h, $f_h(i+1)$and $f_h(i-1)$ are the objective values of the two most adjacent individuals for i at the sorting sequence, respectively.

4.4 Position updating for group members

In the original WHO introduced in reference [29], the positions of group members are updated based on the grazing and mating behavior of horses. In our algorithm, the stallion of the group and the stallions of other groups guide the updating of the position of the individuals. Once the population has completed the Pareto ranking, and the calculation of crowding distance, the member with the greatest crowding distance is regarded as the group's stallion (the group leader). Eq. 15 motivates the group members to move and search around the leader of the same group or the leader of other groups.

$$X'_{i,j,(t+1)} = X_{i,j,t} \oplus X_{i,best,t} \oplus X_{k,best,t} \tag{15}$$

In the above equation, t is the iteration index, $X_{i,j,t}$ is the current position of the individual j of the group i at iteration t, $X_{i,best,t}$ is the position of the stallion of group i; $X_{k,best,t}$ is the position of the stallion of the group k ($k \neq i$). In particular, the group k is randomly selected from the remaining groups except for group i. Moreover, $\oplus$ is the crossover operator introduced later. Eq. 15 means the position of the group member should be updated by its group leader or another group leader. The grazing behavior and horse mating behavior can be embodied in Eq. 15. In the subsequent section, we introduce the crossover operator using Eq. 15.

Crossover operator

To improve the efficiency and effectiveness of the proposed algorithm, we implemented two crossover operators to obtain new group member positions. An improved precedence operation crossover (IPOX) is used to create the new sequence for the OS vector. The IPOX is not only able to retain excellent solutions for the subsequent iteration, but it can also effectively reduce the generation of solutions that are not feasible. The procedures of IPOX for OS are described in detail below. Each group member is paired with a random stallion from another group to form a crossover pair. PO1 and PO2 represent the parents. Let CO1 and CO2 represent their offspring. Some jobs are randomly selected to form a set S, where $|S| \leq n-1$. Then, all jobs are split into two sets S and $N-S$. The positions of the selected jobs S of parent PO1 are copied to CO1. The job of the set $N-S$ is taken from the PO2 in sequence and inserted into the remaining position of the CO1. The same procedure is executed for PO2 and CO2. An example of the IPOX operator is shown in Fig. 2.

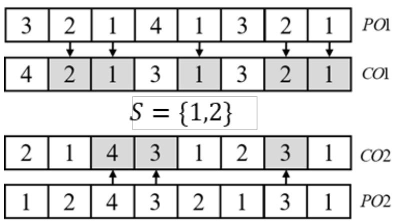


Fig. 2 IPOX for OS

Moreover, the multi-point crossover (MPX) is adopted to generate the new vectors for MA and WA, and the MPX randomly selects some points in the parent individuals for crossover. First, a random set R of 0-1 numbers is generated. In particular, $|R| = \sum_{i \in N} n_i$. The elements of R determine whether the crossover operator should be implemented. When the element is 1, the crossover should be implemented. Let PM1/PW1 and PM2/PW2 denote the parent individuals for MA and WA. Let CM1/CW1 and CM2/CW2 represent the offspring individuals for MA and WA. When the element of R is 1, the elements of two-parent individuals are exchanged to form the offspring individuals. Otherwise, the elements of two-parent individuals are copied into the offspring individuals. When the offspring individuals are infeasible, the parent individuals are kept. An example of an MPX operator is presented in Fig. 3.

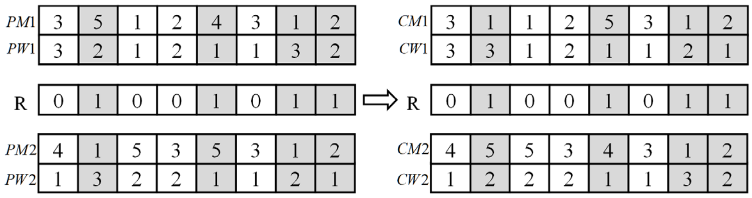


Fig. 3 MPX operator for MA and WA

Mutation operator

The mutation operator working on the elite individuals is turned to lasting diversity during the evolution process. This paper adopts two mutation operators for a three-level encoding scheme. The OS string uses the two-point reversing operator to generate the mutated individual. That is to say, two elements are selected randomly from the OS string, and all operations between the selected two elements are reversed. An example is shown in Fig. 4(a), where PO is the parent OS string, and CO is the chromosome OS string. The two-point random mutation operator is used for the MA and WA strings. In particular, one element is randomly selected from the MA string. And a different component is chosen from the set of available machines for the corresponding job operation. The above procedure is repeated again for another randomly chosen MA element. Moreover, the mutation operator of the WA is the same as the MA. Only the set of the available worker should be used. As shown in Fig. 4(b) and Fig. 4(c), the new MA string and the new WA string are generated based on the available candidate set.

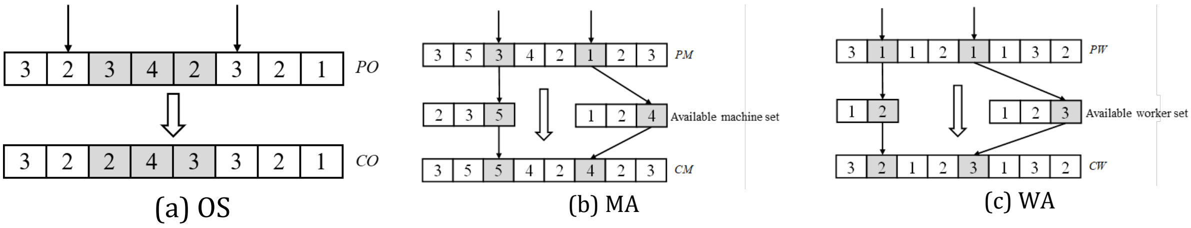


Fig. 4 Mutation operator

4.5 Local search operator

The primary purpose of the local search strategy is to prevent premature convergence and ensure the global search ability of the proposed algorithm. To achieve this, a local search method based on non-dominated sorting is designed. While the introduction of a local search strategy may increase the algorithm's run time, this paper balances run time and search effect by sorting the individuals of each non-dominated solution set by crowding distance, selecting the first 50 % of individuals to construct a search set, and performing a local search on this set. The neighborhood is perturbed by modifying the processes on the critical path.

The neighborhood structures are designed based on the following rules: 1) The two key operations at the end of the first key block on the critical path are exchanged to satisfy process requirements while keeping the machine and worker indexes of the operation unchanged; 2) If the key block is at the end of the critical path, the two key operations at the beginning of the block are exchanged according to the process precedence relationship, without modifying the machine and worker indexes; 3) For other key block operations, only the key operations adjacent to the block head and block tail are exchanged, and operations in the remaining key blocks are not perturbed. No perturbation is performed if there is only one critical operation in the critical block. 4) To avoid generating infeasible solutions, perturbation is not performed if the exchange of operations belongs to the same workpiece. The local search procedure for our multi-objective problem is outlined as follows:

Step1: Construct the neighborhood solution set $NO = \emptyset$ and the comparison solution set $CP = \emptyset$;

Step2: For each individual s_i of search solution set S, all neighborhood solutions are generated by the designed neighborhood structure;

Step3: Evaluate the dominant relationship between the individual s_i and its neighborhood solution s_i'. If $s_i \prec s_i'$, the neighborhood solution s_i' is abandoned; if s_i and s_i' are irrelated, the neighborhood solution s_i' is added into the set NO. If $s_i' \prec s_i$, the current solution s_i is replaced by s_i', the neighborhood solution s_i' is added into the set CP, the neighborhood solution set NO is set to empty, and the search of the individual s_i is stopped;

Step4: Once all neighborhood solutions of the individual s_i have been searching, if a non-dominated solution of s_i is found, s_i is put into the set NO. And the set NO and the set CP are merged. The set NO is set to empty, and the local search starts to implement the above search process for the next individual s_{i+1};

Step5: After searching all individuals of the set S, all individuals of the set CP are nondominated sorted. The non-dominated solutions with large crowding distances are selected to replace the individuals of the set S.

The local search strategy proposed in this paper is based on non-dominated sorting for individual selection, which can effectively select the best solution in the population so that the algorithm can be accelerated toward the convergence to the Pareto frontier.

4.6 Framework of the improved WHO algorithm

The procedure of the improved WHO algorithm solving the DRCFJSP is depicted as follows. Firstly, the algorithm parameters are determined and the instance parameters are input into the algorithm. A population with N_{pop} individuals is generated. The three-level encoding scheme and the greedy insert decoding scheme are used to obtain the schedule with triple objectives. The population based on the fast non-dominated sorting and crowding distance is evaluated. Then, the best and worst solutions are selected to guide the individual updating. A pair of solutions is randomly selected out of the population, and the crossover operation is executed. The crossover process is repeated until every individual is paired with the other population individual with the greatest crowding distance. The individuals from various groups are selected to execute the mutation operation. The above process is repeated until the number of individuals meets the predetermined size. After that, the local search is performed on the first 50 % of individuals based on the crowding distance of the nondominated individuals at the same level. The above process is repeated until the number of individuals meets the predetermined size. Subsequently, the elitist archive with the first Pareto front solutions of the new population is updated. Once the termination criteria are met, output the best individual.

5. Experimental results and analysis

The proposed IWHO algorithm is implemented in Python. The code is run on a PC with an Intel Core i5-6400 CPU (2.70GHz), 8GB RAM, and a 64-bit Windows 10 operating system. For evaluating the performance of the improved IWHO algorithm in solving the multi-objective DRCFJSP problem, the standard NSGA-II method is used as a benchmark here. The two algorithms are run ten times for each test instance to eliminate the effect of the random search.

5.1 Instance generation

Two groups of test instances were generated using the method proposed in reference [17] based on the BRdata and DPpaulli. The first group consists of ten instances with MK01-10[36] and the second group includes eight instances with DP01-08 [37]. Except for the processing information of the jobs, we added the data of machines and workers into the two groups of instances, as shown in Table 2. In particular, the processing time of each operation for all jobs is randomly picked from the interval $[p_{ijk}, p_{ijk} + \delta_{ij}]$, where p_{ijk} is the processing time of the job operation in the original benchmark instance, and δ_{ij} is a random number from the interval [2, 8].

Table 2 Machine-worker data for test instances

Instances	w	The set of eligible workers operating machine k
DMK1-2	4	$W_1 = \{1,3\}, W_2 = \{2,4\}, W_3 = \{1,4\}, W_4 = \{2,3,4\}, W_5 = \{1,2\}, W_6 = \{3\}$
DMK3-4; DP7-8	6	$W_1 = \{1,3\}, W_2 = \{2,4\}, W_3 = \{4\}, W_4 = \{2,3\}, W_5 = \{1,6\}, W_6 = \{3,4,5\}, W_7 = \{4,5\}, W_8 = \{5,6\}$
DMK5	3	$W_1 = \{1,3\}, W_2 = \{2,3\}, W_3 = \{1,3\}, W_4 = \{1,2\}$
DMK6; DMK10	8	$W_1 = \{1,8\}, W_2 = \{2,4\}, W_3 = \{3,8\}, W_4 = \{3,7\}, W_5 = \{6\}, W_6 = \{5\}, W_7 = \{2,5\}, W_8 = \{1,5,6\}, W_9 = \{4,7\}, W_{10} = \{1,6,8\}, W_{11} = \{2,3\}, W_{12} = \{4\}, W_{13} = \{4\}, W_{14} = \{7,8\}, W_{15} = \{5,7\}$
DMK7; DP1-6	4	$W_1 = \{1,4\}, W_2 = \{2,4\}, W_3 = \{1,3\}, W_4 = \{2,3\}, W_5 = \{1,4\}$
DMK8, DMK9	6	$W_1 = \{1,4\}, W_2 = \{2,6\}, W_3 = \{1,3\}, W_4 = \{2,3,6\}, W_5 = \{1,5\}, W_6 = \{5\}, W_7 = \{4,5\}, W_8 = \{3,6\}, W_9 = \{2,4\}, W_{10} = \{3,6\}$

5.2 Parameter setting

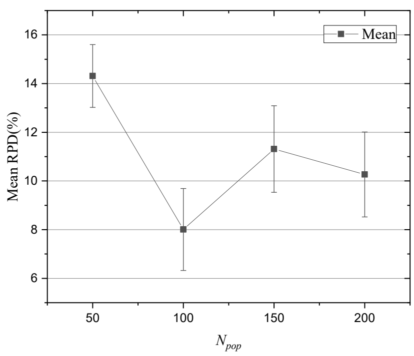
The IWHO algorithm does not need to tune the specific parameters involved in other population-based algorithms. Here we only considered two parameters, including population size N_{pop} and number of iterations $iter_{max}$. The candidate values of the two parameters are listed as follows. $N_{pop} \in \{50,100,150,200\}$, $iter_{max} \in \{50,100,200,500\}$.

Then, we had 16 combinations for the two parameters to tune. The IWHO solved the first four instances of Brdata with 16 combinations. The performance of each combination is measured by the relative percentage deviation (RPD) defined by $RPD(\%) = 100 \times (f_1(i) - f_1^*)/f_1^*$, where $f_1(i)$ is the makespan delivered by the IWHO with the ith parameter value and f_1^* is the best value obtained by the IWHO. The two-way variance analysis (ANOVA) test examined the computational results. The ANOVA results are shown in Table 3. The analysis indicates that parameters $iter_{max}$ and N_{pop} are significantly different. However, the interaction between the two parameters is not statistically significant due to its high P-value. That means it is unnecessary to consider the interaction in further analysis.

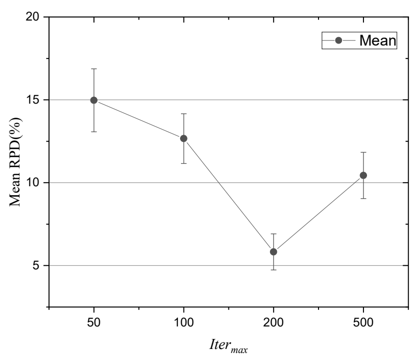
Furthermore, the two parameters are further analyzed by a multi-compared method. The means plot and 95 % confidence level Tukey's Honestly Significant Difference (HSD) intervals for these four candidate values of N_{pop} and $iter_{max}$ are described in Fig. 5(a) and Fig. 5(b), respectively. As can be seen from Fig. 5(a), Tukey's HSD interval with $N_{pop} = 100$ has a better mean RPD value. When considering the running time, we have chosen 100 as the best value of the parameter N_{pop}. The Tukey's HSD interval with $iter_{max}$=200 is significantly lower than that with the other three values. Therefore $iter_{max} = 200$ gives the best result for our algorithm. Therefore, we set the two parameters $N_{pop} = 100$ and $iter_{max} = 200$ in the following computations.

Table 3 ANOVA results for the experiment on tuning the parameters of IWHO

Source	DF	Sum of Squares	Mean Square	F Value	P Value
N_{pop}	3	329.008	109.669	3.042	0.038
$iter_{max}$	3	729.849	243.283	6.748	0.000
Interaction	9	104.434	11.603	0.321	0.963
Model	15	1163.293	77.552	2.151	0.023
Error	48	1730.395	36.049		
Corrected Total	63	2893.688			



(a) N_{pop}



(b) $iter_{max}$

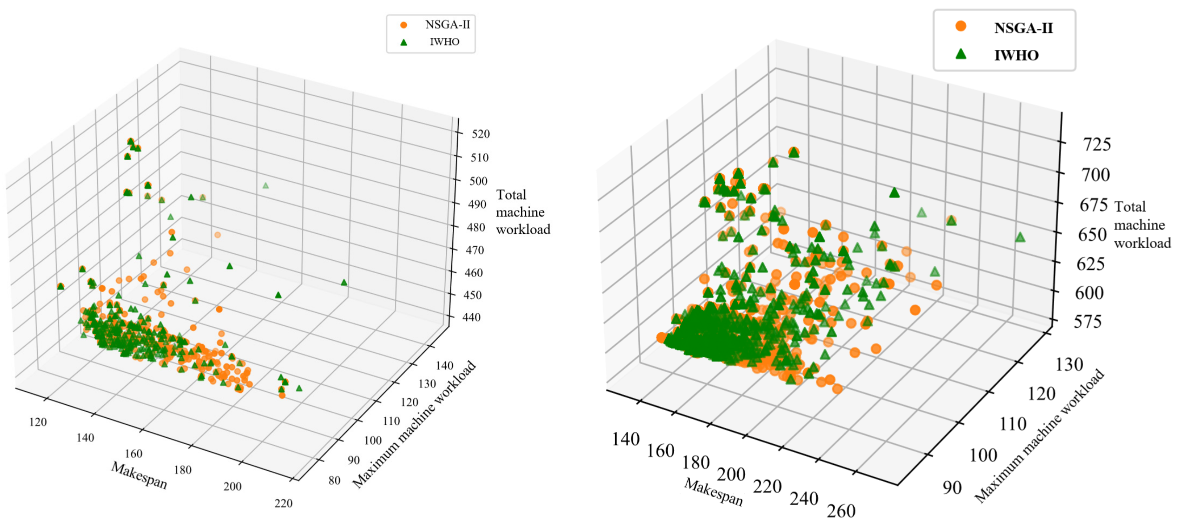
Fig. 5 Means plot and 99 % confidence level Tukey's HSD intervals for population size N_{pop} and $iter_{max}$.

5.3 Result analysis

To evaluate the effectiveness of the algorithm proposed in this paper, we compare the IWHO algorithm with the NSGA-II algorithm for solving the DRCFJSP. The computational results consist of three objectives: minimizing the maximum makespan, the maximal machine workload, and the total machine workload. Table 4 displays the computational results of the IWHO and NSGA-II algorithms for these three objectives, respectively. Moreover, the best objective values are highlighted. It can be observed that the solutions provided by our IWHO algorithm dominate those obtained by NSGA-II for most instances, indicating that the three objectives of the solutions given by IWHO are superior to those of NSGA-II. Upon analyzing the individual objectives, IWHO outperforms NSGA-II in terms of makespan. For total machine workload, 13 out of 18 instances achieved better solutions using IWHO, while for maximal machine workload, both algorithms performed comparably. The running time of the two algorithms is nearly identical. As a multi-objective problem, DRCFJSP requires balancing the three objectives. Our IWHO algorithm performs exceptionally well in solving the studied multi-objective scheduling problem.

Table 4 Computational Results for the IWHO algorithm

Instance				NSGA- II				IWHO			
	n	m	w	f_1	f_2	f_3	CPU(s)	f_1	f_2	f_3	CPU(s)
DMK01	10	6	4	76	52	240	1316	**71**	**52**	**224**	**1175**
DMK02	10	6	4	69	47	208	1586	**60**	**39**	**208**	**1211**
DMK03	15	8	6	310	220	1087	1964	**272**	**219**	**1043**	**1662**
DMK04	15	8	6	144	85	468	1531	**121**	87	**448**	**1228**
DMK05	15	4	3	370	210	814	1576	**332**	**210**	819	**1546**
DMK06	10	15	8	136	88	576	1672	**135**	97	**566**	1737
DMK07	20	5	4	304	185	792	1494	**268**	191	793	**1264**
DMK08	20	10	6	798	613	2833	1912	**744**	**604**	2882	**1903**
DMK09	20	10	6	661	348	2616	1878	**624**	**347**	2680	2045
DMK10	20	15	8	486	266	2258	2167	**462**	**261**	**2256**	**2066**
DP01	10	5	4	4667	2556	11368	1785	**3962**	2574	11418	**1644**
DP02	10	5	4	4586	2285	11361	1831	**3967**	2329	**11352**	**1774**
DP03	10	5	4	4157	2326	11311	1907	**4116**	**2307**	11360	**1836**
DP04	10	5	4	4693	2576	11354	1819	**3888**	2656	**11339**	1867
DP05	10	5	4	4140	2284	11251	1709	**3902**	2384	**11250**	1892
DP06	10	5	4	4418	2286	11132	1930	**3948**	**2269**	11214	**1920**
DP07	15	8	6	5247	2337	16863	2089	**4406**	2387	16925	2196
DP08	15	8	6	5209	2138	16696	2458	**4366**	2201	16748	**2006**



(a)DMK04 (a)DMK06

Fig. 6 Pareto front for DMK04 and DMK06 solutions delivered by NSGA-II and IWHO

To provide a visual representation of the performance of the two algorithms, a 3D scatter plot was used to plot the Pareto optimal fronts generated by the solutions for DMK04 and DMK06, as depicted in Fig. 6. The scatter plot clearly reveals the trade-offs between the three objectives being considered, indicating a trade-off between makespan and total machine workload, as well as between total machine workload and maximum machine workload. Additionally, there is a positive correlation between makespan and maximum machine workload. The IWHO algorithm Pareto fronts are distributed evenly and are positioned closer to the bottom of the graph, indicating its superior performance over NSGA-II.

Furthermore, the performance of the proposed IWHO algorithm is analyzed when changing the scale of the problem instances. The gap of the IWHO algorithm compared with the NSGA-II is calculated, such as gap $= \frac{f(\text{NSGA}-\text{II}) - f(\text{IWHO})}{f(\text{NSGA}-\text{II})} \times 100\%$. In particular, the gap value is larger, the proposed IWHO is better. When the gap value is greater than 0, the IWHO performs better than the NSGA-II. The size of the test instances is increased for the instances DMK01-DMK10. The gap curve of the proposed IWHO algorithm related to the NSGA-II is given in Fig. 7. It can be observed that the proposed IWHO algorithm tends to optimize the first objective (makespan) more, and there is no significant deterioration when the scale of the problem increases. That means the IWHO algorithm still performs better than NSGA-II, even increasing the problem complexity.

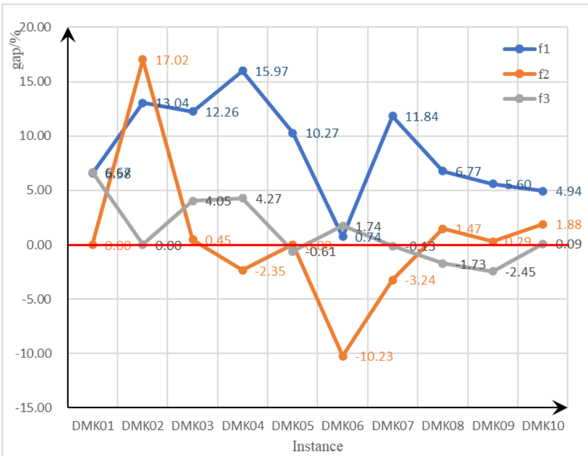


Fig. 7 The gap curve of the three objectives given by IWHO compared with NSGA-II

6. Real case study

To evaluate the practical applicability of the proposed algorithm, we conducted a case study at a railway rolling stock corporation located in Qingdao. The assembly process of the buffers and chain coupler for the high-speed rail was selected as the subject of our study, specifically focusing on the automatic buffers and chain coupler module. This module consists of several components, including the suspension system, buffer crushing system, connecting system, electrical control system, pneumatic control system, and connecting ring, as depicted in Fig. 8.

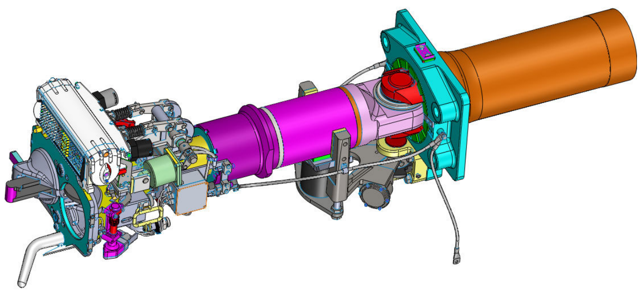


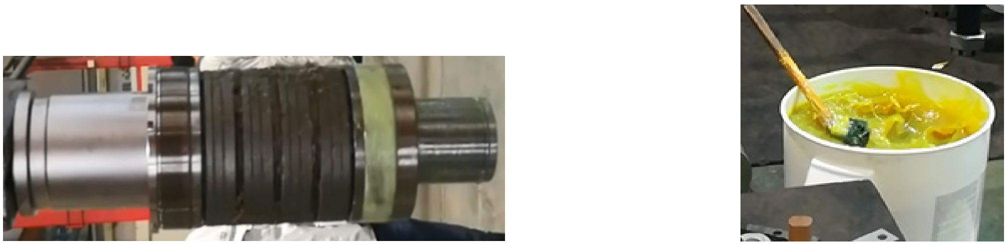
Fig. 8 Structure diagram of automatic buffers and chain coupler

The general assembly process of buffers and chain couplers is detailed as follows:

(1) To transfer the connecting system, a self-service crane hoists the subsequent connected system shown in Fig. 9 and places it on the spare parts table, where it awaits the following operation.

Fig. 9 Schematic diagram of the connecting system

(2) Lubrication of the shaft end of the buffer parts on Fig. 10(a) involves applying lubricating grease to the mating surface of the shaft end of the buffer parts using a brush, Fig. 10(b).



(a) Buffer parts (b) Grease

Fig. 10 Lubricating the shaft end of the buffer composition

(3) In preparation for composition, a press is used to drive two cylindrical pins into the pin holes of the lower connecting ring, and the guide plate is installed. The anti-loosening plate is tightened and torqued with hexagon head bolts to a torque of 70 Nm. Both ends of the anti-loosening plate are bent towards the hexagon head of the abutting bolt to prevent loosening. Once the lower connecting ring components are prepared, the inner ring surface is greased. The other elements of the connecting ring, including the upper half ring, connecting ring fastening bolts, nuts, anti-loosening plates, stop blocks, etc., are also prepared. Lubricating grease is applied to the inner surface of the upper half ring, and an appropriate amount of thread grease is applied to the threaded part of the bolt according to the working instructions. The whole process is shown in Fig. 11.

(a) Press (b) Lower connecting ring (c) Thread grease (d) Upper connecting ring

Fig. 11 Schematic of preparing the connection ring

(4) Installing the connection ring involves hoisting the connecting system to position the end of the rear crushing and centering buffer device and adjusting the position of the guide plate. Then, the connecting ring, stop block, bolt, anti-loosening plate, and nut are installed in sequence. The distance between the upper and lower rings is checked and adjusted to maintain the same spacing. Each nut is tightened three times in diagonal order, and torque is applied. Finally, the anti-loosening plate is blended until it is against the nut to prevent loosening.

(5) To install the cover plate, hexagon head bolts and anti-loosening plates are used to tighten and torque. An appropriate amount of thread grease is applied to the threaded parts of the bolts before fastening. After tightening the bolts, both ends of the anti-loosening plate are blended towards the side of the hexagonal head of the bolt to prevent loosening.
(6) The ground wire is installed, and the hexagon head bolts, anti-loose washers, and flat washers are connected and fastened.
(7) All tightened bolts with anti-loosening marks are scratched to signify their status.
(8) The grounding mark is pasted next to the terminal, and a layer of varnish is applied after pasting to protect it.
(9) Lubricating grease is applied to the circumference of the connecting bolts of the connecting ring and the mounting holes of the lower half ring.

The procedure outlined previously is identified by numbers (1) through (9). The assembly process for additional buffer and chain coupler types is comparable to the aforementioned process. However, the semi-permanent coupler that lacks a buffer does not necessitate the lubrication process for the shaft end (process 2). The workshop has 20 sets of equipment or consoles in nine groupings designated M1 through M20. The equipment accessible for each process varies, as indicated in Table 5.

Due to the variations in structure, weight, and process requirements of different coupler products, the time required for each process may vary. Additionally, the processing time for the same process step may vary due to differences in worker skills and proficiency. Table 6 presents the processing times of different processes for the automatic buffer and chain coupler used in high-speed rail for different workers. Please note that a blank value indicates that the worker assigned to that task is unable to perform the process.

The production data of the above automatic buffers and chain coupler was used in the DRCFJSP. Detailed test data can be obtained from the authors. The IWHO algorithm was used to solve the problem based on the best parameter combination determined by the ANOVA. The Gantt chart can be found in Fig. 12. Based on the comparison between the developed IWHO algorithm and the practical sequencing approach, and we found that the results delivered by our IWHO can achieve a 21.4 % improvement in the average three objective values.

Table 5 Available equipment for different assembly processes

Process	Available equipment
Process1	M1, M2
Process2	M3, M4, M5, M6
Process3	M7, M8, M9
Process4	M10, M11, M12
Process5	M10, M11, M12, M13, M14
Process6	M15, M16, M17, M18
Process7	M17, M18
Process8	M19, M20
Process9	M3, M4, M5, M6, M7

Table 6 Process timetable of the automatic buffers and chain coupler

Processes	Worker 1	Worker2	Worker3	Worker4	Worker5	Worker6	Worker7	Worker8
Process1	19		23					20
Process2	8	9	7	13	10	11	12	7
Process3	35	31			32	32		33
Process4			37	37	40	38	39	40
Process5					48	46	47	48
Process6		53					47	50
Process7	11	11					9	7
Process8			10	10	13	14		11
Process9	17	18	16	20	17	18	19	17

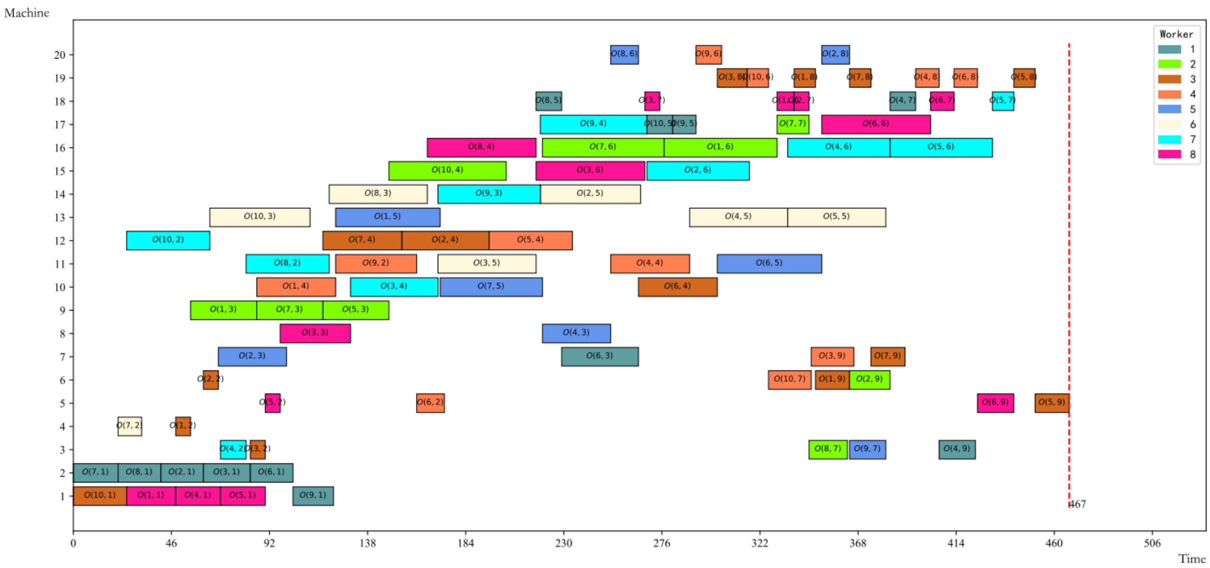


Fig. 12 Gantt chart for machines and workers in 10-job instance

7. Conclusions and future work

This paper proposes an IWHO algorithm designed to reduce makespan, maximum machine workload, and total machine workload for the DRCFJSP. The algorithm utilizes multi-group communication and three-level integer coding and implements a greedy insertion decoding technique to establish the initial schedule, taking into account the three optimization objectives. The algorithm also uses an elite preservation strategy to identify the best individuals in the population and incorporates the fast non-dominated sorting and crowding distance mechanism from the NSGA-II algorithm to sort the population and guide the WHO algorithm's members in updating their positions. To adhere to the characteristics of the WHO algorithm, the algorithm employs a crossover operator and a mutation operator to discretize the updating process. A local neighborhood search strategy centered on the critical path is introduced to prevent premature convergence and help avoid the local optimum. The computational test results demonstrate the effectiveness of the proposed algorithm for solving multi-objective DRSFJSP. Furthermore, a practical test scenario is designed to evaluate the performance of the suggested IWHO in assembling buffers and chain couplers. In the future, the study intends to explore dynamic problems in the workshop environment under resource constraints, such as rescheduling flexible job shops during emergencies such as order insertion, order cancellation, machine failure, or worker departure. Additionally, more effective improvement strategies, including simulated annealing, tabu search, etc., are encouraged to enhance the efficiency of the WHO algorithm.

References

[1] Çaliş, B. Bulkan, S. (2015). A research survey: Review of AI solution strategies of job shop scheduling problem, *Journal of Intelligent Manufacturing*, Vol. 26, No. 5, 961-973, doi: 10.1007/s10845-013-0837-8.

[2] Xie, J., Gao, L., Peng, K., Li, X., Li, H. (2019). Review on flexible job shop scheduling, *IET Collaborative Intelligent Manufacturing*, Vol. 1, No. 3, 67-77, doi: 10.1049/iet-cim.2018.0009.

[3] Amjad, M.K., Butt, S.I., Anjum, N., Chaudhry, I.A., Faping, Z., Khan, M. (2020). A layered genetic algorithm with iterative diversification for optimization of flexible job shop scheduling problems, *Advances in Production Engineering & Management*, Vol. 15, No. 4, 377-389, doi: 10.14743/apem2020.4.372.

[4] Lei, K., Guo, P., Zhao, W.C., Wang, Y., Qian, L., Meng, X., Tang, L. (2022). A multi-action deep reinforcement learning framework for flexible Job-shop scheduling problem, *Expert Systems with Applications*, Vol. 205, Article No. 117796, doi: 10.1016/j.eswa.2022.117796.

[5] Behnamian, J., Fatemi Ghomi, S.M.T. (2016). A survey of multi-factory scheduling, *Journal of Intelligent Manufacturing*, Vol. 27, No. 1, 231-249, doi: 10.1007/s10845-014-0890-y.

[6] Brucker, P., Schlie, R. (1990). Job-shop scheduling with multi-purpose machines, *Computing*, Vol. 45, 369-375, doi: 10.1007/BF02238804.

[7] Zhao, Z., Lin, P., Shen, L., Zhang, M., Huang, G.Q. (2020). IoT edge computing-enabled collaborative tracking system for manufacturing resources in industrial park, *Advanced Engineering Informatics*, Vol. 43, Article No. 101044, doi: 10.1016/j.aei.2020.101044.
[8] Xu, J., Xu, X., Xie, S.Q. (2011). Recent developments in Dual Resource Constrained (DRC) system research, *European Journal of Operational Research*, Vol. 215, No. 2, 309-318, doi: 10.1016/j.ejor.2011.03.004.
[9] Guo, D., Zhong, R.Y., Ling, S., Rong, Y., Huang, G.Q. (2020). A roadmap for Assembly 4.0: Self-configuration of fixed-position assembly islands under Graduation Intelligent Manufacturing System, *International Journal of Production Research*, Vol. 58, No. 15, 4631-4646, doi: 10.1080/00207543.2020.1762944.
[10] Xu, S., Hall, N.G. (2021). Fatigue, personnel scheduling and operations: Review and research opportunities, *European Journal of Operational Research*, Vol. 295, No. 3, 807-822, doi: 10.1016/j.ejor.2021.03.036.
[11] Grosse, E.H., Calzavara, M., Glock, C.H., Sgarbossa, F. (2017). Incorporating human factors into decision support models for production and logistics: Current state of research, *IFAC-PapersOnLine*, Vol. 50, No. 1, 6900-6905, doi: 10.1016/j.ifacol.2017.08.1214.
[12] Nourmohammadi, A., Fathi, M., Ng, A.H.C. (2022). Balancing and scheduling assembly lines with human-robot collaboration tasks, *Computers & Operations Research*, Vol. 140, Article No. 105674, doi: 10.1016/j.cor.2021.105674.
[13] Chaudhry, I.A., Khan, A.A. (2016). A research survey: Review of flexible job shop scheduling techniques, *International Transactions in Operational Research*, Vol. 23, No. 3, 551-591, doi: 10.1111/itor.12199.
[14] Li, H., Duan, J., Zhang, Q. (2021). Multi-objective integrated scheduling optimization of semi-combined marine crankshaft structure production workshop for green manufacturing, *Transactions of the Institute of Measurement and Control*, Vol. 43, No. 3, 579-596, doi: 10.1177/0142331220945917.
[15] Ren, W., Wen, J., Yan, Y., Hu, Y., Guan, Y., Li, J. (2021). Multi-objective optimisation for energy-aware flexible job-shop scheduling problem with assembly operations, *International Journal of Production Research*, Vol. 59, No. 23, 7216-7231, doi: 10.1080/00207543.2020.1836421.
[16] Ham, A. (2017). Flexible job shop scheduling problem for parallel batch processing machine with compatible job families, *Applied Mathematical Modelling*, Vol. 45, 551-562, doi: 10.1016/j.apm.2016.12.034.
[17] Lei, D., Guo, X. (2014). Variable neighbourhood search for dual-resource constrained flexible job shop scheduling, *International Journal of Production Research*, Vol. 52, No. 9, 2519-2529, doi: 10.1080/00207543.2013.849822.
[18] Lei, D., Guo, X. (2015). An effective neighborhood search for scheduling in dual-resource constrained interval job shop with environmental objective, *International Journal of Production Economics*, Vol. 159, 296-303, doi: 10.1016/j.ijpe.2014.07.026.
[19] Yazdani, M., Zandieh, M., Tavakkoli-Moghaddam, R., Jolai, F. (2015). Two meta-heuristic algorithms for the dual-resource constrained exible job-shop scheduling problem, *Scientia Iranica*, Vol. 22, No. 3, 1242-1257.
[20] Ahmadi, E., Zandieh, M., Farrokh, M., Emami, S.M. (2016). A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms, *Computers & Operations Research*, Vol. 73, 56-66, doi: 10.1016/j.cor.2016.03.009.
[21] Zheng, X.-L., Wang, L. (2016). A knowledge-guided fruit fly optimization algorithm for dual resource constrained flexible job-shop scheduling problem, *International Journal of Production Research*, Vol. 54, No. 18, 5554-5566, doi: 10.1080/00207543.2016.1170226.
[22] Xu, H., Bao, Z.R., Zhang, T. (2017). Solving dual flexible job-shop scheduling problem using a Bat Algorithm, *Advances in Production Engineering & Management*, Vol. 12, No. 1, 5-16, doi: 10.14743/apem2017.1.235.
[23] Ham, A. (2018). Scheduling of dual resource constrained lithography production: Using CP and MIP/CP, *IEEE Transactions on Semiconductor Manufacturing*, Vol. 31, No. 1, 52-61, doi: 10.1109/TSM.2017.2768899.
[24] Wu, R., Li, Y., Guo, S., Xu, W. (2018). Solving the dual-resource constrained flexible job shop scheduling problem with learning effect by a hybrid genetic algorithm, *Advances in Mechanical Engineering*, Vol. 10, No. 10, doi: 10.1177/1687814018804096.
[25] Wu, X., Peng, J., Xiao, X., Wu, S. (2021). An effective approach for the dual-resource flexible job shop scheduling problem considering loading and unloading, *Journal of Intelligent Manufacturing*, Vol. 32, No. 3, 707-728, doi: 10.1007/s10845-020-01697-5.
[26] Tan, W., Yuan, X., Wang, J., Zhang, X. (2021). A fatigue-conscious dual resource constrained flexible job shop scheduling problem by enhanced NSGA-II: An application from casting workshop, *Computers & Industrial Engineering*, Vol. 160, Article No. 107557, doi: 10.1016/j.cie.2021.107557.
[27] Andrade-Pineda, J.L., Canca, D., Gonzalez-R, P.L., Calle, M. (2020). Scheduling a dual-resource flexible job shop with makespan and due date-related criteria, *Annals of Operations Research*, Vol. 291, No. 1-2, 5-35, doi: 10.1007/s10479-019-03196-0.
[28] Vital-Soto, A., Baki, M.F., Azab, A. (2022). A multi-objective mathematical model and evolutionary algorithm for the dual-resource flexible job-shop scheduling problem with sequencing flexibility, *Flexible Services and Manufacturing Journal*, Vol. 35, 626-668, doi: 10.1007/s10696-022-09446-x.
[29] Naruei, I., Keynia, F. (2022). Wild horse optimizer: A new meta-heuristic algorithm for solving engineering optimization problems, *Engineering with Computers*, Vol. 38, Suppl. 4, 3025-3056, doi: 10.1007/s00366-021-01438-z.
[30] Li, Y., Yuan, Q., Han, M., Cui, R. (2022). Hybrid multi-strategy improved wild horse optimizer, *Advanced Intelligent Systems*, Vol. 4 , No. 10, Article No. 2200097, doi: 10.1002/aisy.202200097.
[31] Vasanthkumar, P., Revathi, A.R., Ramya Devi, G., Kavitha, R.J., Muniappan, A., Karthikeyan, C. (2022). Improved wild horse optimizer with deep learning enabled battery management system for internet of things based hybrid

electric vehicles, *Sustainable Energy Technologies and Assessments*, Vol. 52, Part C, Article No. 102281, doi: 10.1016/j.seta.2022.102281.

[32] Ali, M.H., Kamel, S., Hassan, M.H., Tostado-Véliz, M., Zawbaa, H.M. (2022). An improved wild horse optimization algorithm for reliability based optimal DG planning of radial distribution networks, *Energy Reports*, Vol. 8, 582-604, doi: 10.1016/j.egyr.2021.12.023.

[33] Alphonse, A.R.A., Raj, A.P.P.G., Arumugam, M. (2022). Simultaneously allocating electric vehicle charging stations (EVCS) and photovoltaic (PV) energy resources in smart grid considering uncertainties: A hybrid technique, *International Journal of Energy Research*, Vol. 46, No. 11, 14855-14876, doi: 10.1002/er.8187.

[34] Milovanović, M., Klimenta, D., Panić, M., Klimenta, J., Perović, B. (2022). An application of Wild Horse Optimizer to multi-objective energy management in a micro-grid, *Electrical Engineering*, Vol. 104, 4521-4541, doi: 10.1007/s00202-022-01636-y.

[35] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, 182-197, doi: 10.1109/4235.996017.

[36] Brandimarte, P. (1993). Routing and scheduling in a flexible job shop by tabu search, *Annals of Operations Research*, Vol. 41, No. 3, 157-183, doi: 10.1007/BF02023073.

[37] Dauzère-Pérès, S., Paulli, J. (1997). An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search, *Annals of Operations Research*, Vol. 70, 281-306, doi: 10.1023/A:1018930406487.