An effective multi-objective discrete grey wolf optimizer for a real-world scheduling problem in welding production

Chao Lu, Shengqiang Xiao, Xinyu Li, Liang Gao*

State Key Lab of Digital Manufacturing Equipment & Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, China

Abstract: This paper aims to provide a solution method for a real-world scheduling case from a welding process, which is one of the important processes in modern industry. The unique characteristic of the welding scheduling problem (WSP) is that multiple machines can process one operation at a time. Thus, WSP is a new scheduling problem. We first formulate a new multi-objective mixed integer programming model for this WSP based on a comprehensive investigation. This model involves some realistic constraints, controllable processing times (CPT), sequence dependent setup times (SDST) and job dependent transportation times (JDTT). Then we propose a multi-objective discrete grey wolf optimizer (MODGWO) considering not only production efficiency but also machine load on this real-world scheduling case. The solution is encoded as a two-part representation including a permutation vector and a machine assignment matrix. A reduction machine load strategy is used to adjust the number of machines aiming to minimize the machine load. To evaluate the effectiveness of the proposed MODGWO, we compare it with other well-known multi-objective evolutionary algorithms including NSGA-II and SPEA2 on a set of instances. Experimental results demonstrate that the proposed MODGWO is superior to the compared algorithms in terms of convergence, spread and coverage on most instances. Finally, MODGWO is successfully applied to this real-world WSP. This implies that the proposed model is feasible and the proposed algorithm can solve this real-world scheduling problem very well.

Keywords: Welding scheduling; Multi-objective evolutionary algorithm; Controllable processing times; Sequence dependent setup times; Transportation times; Grey wolf optimizer.

1. Introduction

Welding is one of the most crucial technologies of materials forming and processing in modern manufacturing sectors [1], which has been widely applied in a variety of industrial fields such as petroleum chemical industry, mechanical industry, space flight and aviation, and microelectronic engineering. Welding process accounts for a noticeable portion of the total manufacturing process [1]. In general, a reasonable

* corresponding author: gaoliang@mail.hust.edu.cn

welding schedule scenario can effectively improve production efficiency. Thus, welding scheduling problem (WSP) plays an important role in promoting an enterprise to be a manufacturing giant. However, studies on the WSP are relatively scarce in comparison with other real-world scheduling problems such as steelmaking scheduling issues [2-9]. The main reasons for few researches on the real-world WSP may be the following aspects.

- (1) The main difference between WSP and the traditional scheduling problem is the number of machines required to process the operation of one job. In most scheduling problems, one job is usually handled by one and only one machine at a time. However, in a realistic welding situation, it is common to observe that multiple welding machines can process a job at the same time. This characteristic should be considered when modelling and solving WSP. Thus, it increases the difficulty of modelling.
- (2) WSP can be regarded as a special permutation flow shop scheduling problem (PFSP). Thus, WSP is also NP-hard due to NP-hard nature of the PFSP [10]. Nevertheless, WSP is more sophisticated in the practical production since it involves some realistic constraints, sequence dependent setup times (SDST), job dependent transportation times (JDTT), and controllability of processing times at certain stages. However, most existing models are simplified versions ignoring some realistic assumptions, which cannot completely satisfy requirements of the real-world welding workshop.

The above reasons increase the complexity of this study. Clearly, there still exists a significant gap between theoretical research and practical application [8]. Therefore, WSP is an important research topic in terms of both theory and application.

This paper investigates a multi-objective WSP from a realistic welding workshop in China, which has guiding significance on the modern welding industry. In this realistic workshop, the welding process mainly includes five stages (see **Fig. 1**): (1) Splices of small pieces, in which many small pieces such as rib plate and cover plate are melted into several parts by the TIG welding machine, and then these parts are conveyed to the next stage. (2) Splices of large parts, in which several parts are melted into a complete part by the CO₂ shield semi auto welder. (3) Internal seam welding, in which continuously weld on the web, plate and so on. Turn over the part after the welding operation of one side of the web is completed, and perform the same operation on the other side of the web by the CO₂ shield semi auto welder. (4) Encapsulation, in which box beam is transferred down by the crane and fixed in a predetermined position. (5) Outside seam welding, in which the both sides of the box beam are welded by the submerged arc automatic welder. Furthermore, the real-world WSP is a multi-objective optimization problem (MOP) in nature. Purely single-objective scheduling problem cannot fully reflect the requirements of the real-world production system. Thus, this real-world WSP takes into account two main criteria, namely, makespan and machine

load or interference. One simple strategy for addressing multiple objectives is to combine them into a scalar function by giving fixed weights to each objective function, and then an optimization algorithm is adopted to solve this scalar objective problem. Nevertheless, in most MOPs, objectives have different scales and thus it is difficult to determine weight values [11]. Therefore, it is better to handle multiple objectives with knowledge about Pareto dominance. Pareto-based multi-objective evolutionary algorithm (MOEA) is suitable for solving multi-objective scheduling problems since it can yield the non-dominated solutions in a single run [12]. For example, Han et al. [13] presented an improved NSGA-II and applied it to tackle the lot-streaming flow shop scheduling problem with four criteria. Ciavotta et al. [11] developed a new MOEA and provided a comprehensive study on multi-objective permutation flow shop with sequence dependent setup times. Wang et al. [14] developed a multi-objective genetic algorithm based on the immune and entropy concept for flexible job shop scheduling problem. Amin-Tahmasbi and Tavakkoli-Moghaddam [15] proposed a multi-objective immune system to address a bi-objective flow shop scheduling problem with SDST. Zheng and Wang [16] developed a Pareto-based estimation of distribution algorithm to solve a production scheduling problem. Rifai et al. [17] proposed a novel multi-objective adaptive large neighborhood search algorithm to address a distributed permutation flow shop scheduling problem. A recent review on the multi-objective permutation flow shop scheduling problem is provided by Yenisey and Yagmahan [18]. Based on the merits of Pareto-based methods, they not only obtain non-dominated solutions in a single run but also reveal a tradeoff relation between several criteria. Therefore, we adopt a Pareto-based MOEA to address this considered problem.

Fig. 1. The process flow chart for a welding plant

Recently, many metaheuristics have been developed such as virus colony search [19], whale optimization algorithm [20], ant lion optimizer [21], grey wolf optimizer [22] and backtracking search algorithm [23]. Among them, grey wolf optimizer (GWO) [22] is a new swarm intelligence algorithm inspired from mimicking the leadership hierarchy and hunting mechanism of grey wolves. GWO is proven to be superior or competitive to other classical metaheuristics such as differential evolutionary, genetic algorithm and particle swarm optimization algorithm. GWO has been successfully applied to many engineering fields, such as parameter estimation in surface waves [24], optimization of controller's gains [25], the optimal power flow problem [26],

hyperspectral image classification [27] and designing photonic crystal waveguides [28]. Based on the effectiveness of GWO and the nature of the multi-objective (MOP), a new multi-objective discrete grey wolf optimizer (MODGWO) is proposed to solve this multi-objective WSP. The main reasons for applying MODGWO in WSP are that the problem under study is NP-hard and GWO has been demonstrated to be an effective approach for tackling the flow shop problem [29]. In addition, GWO is easy to implement due to its a few parameters and simple mechanism. These motivations drive us to design a new multi-objective algorithm based on GWO for this scheduling problem. To the best of the authors' knowledge, no paper has been reported that addresses a real-world WSP using MODGWO. Although a multi-objective GWO (MOGWO) has been proposed [30], MOGWO is designed to deal with continuous MOPs rather than combinatorial optimization MOPs. In this paper, we have accomplished at least three things in order to narrow the gap between theoretical research and practical application. First, a new multi-objective mathematical model for this WSP is formulated. Second, an efficient MODGWO is developed to obtain a trade-off curve between the makespan and the penalty of machine load (or machine interference). Third, the proposed MODGWO is used to solve a real-world welding scheduling case, which achieves good results.

The remainder of this paper is organized as follows. The mathematical model for this scheduling problem is formulated in Section 2. Section 3 describes a new multi-objective optimization approach for this WSP. Sections 4 and 5 present experimental results and a case study, and Section 6 provides the conclusions and future work.

2. Problem statement and mathematical modelling

In this section, we first give a problem statement and then present a detailed multi-objective mathematical model for the WSP which derives from a realistic welding workshop.

2.1 Problem Statement

The WSP under this study can be described as follows. A set of n jobs has to be processed through five stages in the same process. Each job has at most one operation at each stage. Preemption and interrupt are not allowed. Multiple welding machines can process the same job at a time, and each welding machine can handle at most one job at a time. Each operation has a normal processing time when only one machine is used to process the corresponding job. The normal operation processing times can be controlled (i.e., compressed) by allocating multiple welding machines at the stages of welding processes, which incurs additional machine load and machine interference. Clearly, this WSP is an extension of PFSP with controllable processing times (CPT), which is more complex than the traditional PFSP.

In addition, the main characteristics of the welding process can be summarized below.

- All the jobs go through the same stage or route in the welding workshop: splices

of small pieces, splices of large parts, internal seam welding, encapsulation and outside seam welding.

- One and only one machine can process one job at a time at encapsulation stage. The processing duration of each operation can be controlled (i.e., compressed) by allocating additional welding machines to the operation at the other stages.
- Setup times are taken into account in this welding workshop due to some preparation work such as cleaning and preheating, and the setup duration is associated with the similarities between the two adjacent jobs. Thus, setup times are sequence dependent.
- Release times of all the jobs at each stage are relatively short, and thus can be neglected in this welding shop.
- Transportation times between two consecutive stages are also considered. In this welding workshop, transportation times depend not only on the distance between consecutive stages (i.e., stations) but also the job to be transported. Therefore, transportation times are job dependent. In addition, there are enough transporters to carry the job from one stage to another. The loading and unloading times are included in the transportation times.
- When the setup times and transportation times are considered simultaneously, the overlapping of the setup times and transportation times is permitted. Thus, one operation cannot start until the setup times of the operation and transportation times of the preceding operations are completed (e.g., **Fig. 2**).

Fig. 2. Overlapping of setup and transportation times when (a) the start time is determined by setup times and (b) the start time is determined by transportation times [31]

2.2 Mathematical modelling for WSP

To formulate a multi-objective mathematical model for WSP, the notations and decision variables used throughout the paper are given below.

1) Notations:

j : job j , $j \in \mathcal{J}$.

n : the total number of jobs.

m : the total number of all the stages, that is, $m = 5$.

π : a feasible sequence.

$\pi(j)$: job in the j -th position in a sequence.

C_{max} : the makespan of the schedule, i.e., the completion time of the last job

transported to the warehouse.

$\mathcal{M}$: a set of machines.

$\mathcal{M}_i$: a set of available machines at stage i .

TM_i : the number of available machines to process each job at stage i .

O_{ij} : the operation of job j at the stage i .

N_{ij} : the number of the machines to process the operation O_{ij} .

p_{ij} : the normal processing time of the operation O_{ij} . That is, the processing time of the operation should satisfy condition $N_{ij} = 1$.

p_{ij}^a : the actual processing time of the operation O_{ij} , which is inversely proportional to the number of the corresponding machine, namely, $p_{ij}^a = \frac{p_{ij}}{N_{ij}}$.

S_{ij} : the start time of the operation O_{ij} .

sT_{i0j} : the setup time required to process job j first at the stage i .

$sT_{ijj'}$: the setup time between job j and job j' at the stage i .

$t_{jii'}$: the transportation time of job j from stage i to i' .

t_{j0i} : the transportation time of job j from warehouse to stage i .

t_{jiw} : the transportation time of job j from stage i to warehouse.

u_i : the coefficient of penalty for using machine at stage i .

L : a very large positive number.

2) Decision variables:

N_{ij} : the number of the machines to process the operation O_{ij} .

S_{ij} : the start time of the operation O_{ij} . The S_{ij} should meet the following conditions:

S_{ij}

$$= \begin{cases} \max(t_{j0i}, sT_{i0j}) & \text{if } O_{ij} \text{ is the first operation of job } j \text{ and the first operation at stage } i \text{ (i.e., } i=1\text{).} \\ \max(t_{j0i}, S_{ij'} + p_{ij'}^a + sT_{ij'j}) & \text{if } O_{ij} \text{ is the first operation of job } j \text{ and is started just after } O_{ij'} \text{ at stage } i \text{ (i.e., } i=1\text{).} \\ \max(S_{i'j} + p_{i'j}^a + t_{ji'i}, sT_{i0j}) & \text{if } O_{i'j} \text{ is completed at stage } i' \text{ and then at stage } i \text{ and } O_{ij} \text{ is the first operation on stage } i. \\ \max(S_{i'j} + p_{i'j}^a + t_{ji'i}, S_{ij'} + p_{ij'}^a + sT_{ij'j}) & \text{if } O_{i'j} \text{ is completed at stage } i' \text{ and then at stage } i \text{ and } O_{ij} \text{ is processed just after } O_{ij'}. \end{cases}$$

$$x_{i'ij} = \begin{cases} 1, & \text{if } O_{i'j} \text{ precedes } O_{ij} \\ 0, & \text{if } O_{ij} \text{ precedes } O_{i'j} \end{cases}$$

$$y_{ij'j} = \begin{cases} 1, & \text{if } O_{ij'} \text{ precedes } O_{ij} \\ 0, & \text{if } O_{ij} \text{ precedes } O_{ij'} \end{cases}$$

$$z_{ijk} = \begin{cases} 1, & \text{if machine } k \text{ processes job } j \text{ at stage } i \\ 0, & \text{otherwise} \end{cases}$$

With the above notations, the WSP can be defined as a mixed integer programming (MIP) model for minimizing the makespan and the total penalty value of machine load simultaneously, which is formulated as follows.

$$\begin{cases} \min f_1 = C_{max} \\ \min f_2 = \sum_{i=1}^m \sum_{j=1}^n u_i N_{ij} \end{cases} \quad (1)$$

Subject to

$$\sum_{\mathcal{M}_i} z_{ijk} = N_{ij} \geq 1 \quad i \in \{1,2,\dots,5\}; j \in \mathcal{J}; k \in \mathcal{M} \quad (2)$$

$$\sum_{\mathcal{M}_4} z_{4jk} = N_{4j} = 1 \quad j \in \mathcal{J}; k \in \mathcal{M} \quad (3)$$

$$N_{ij} \leq TM_i \quad i \in \{1,2,\dots,5\}; j \in \mathcal{J} \quad (4)$$

$$S_{ij} \geq t_{j0i} \quad i \in \{1,2,\dots,5\}; j \in \mathcal{J} \quad (5)$$

$$S_{ij} \geq sT_{i0j} \quad i \in \{1,2,\dots,5\}; j \in \mathcal{J} \quad (6)$$

$$L(1 - x_{i'ij}) + S_{ij} \geq S_{i'j} + \frac{p_{i'j}}{N_{i'j}} + t_{ji'i} \quad i, i' \in \{1,2,\dots,5\}; j \in \mathcal{J} \quad (7)$$

$$Lx_{i'ij} + S_{i'j} \geq S_{ij} + \frac{p_{ij}}{N_{ij}} + t_{jii'} \quad i, i' \in \{1,2,\dots,5\}; j \in \mathcal{J} \quad (8)$$

$$L(1 - y_{ij'j}) + S_{ij} \geq S_{ij'} + \frac{p_{ij'}}{N_{ij'}} + sT_{ij'j} \quad i \in \{1,2,\dots,5\}; j, j' \in \mathcal{J} \quad (9)$$

$$Ly_{ij'j} + S_{ij'} \geq S_{ij} + \frac{p_{ij}}{N_{ij}} + sT_{ijj'} \quad i \in \{1,2,\dots,5\}; j, j' \in \mathcal{J} \quad (10)$$

$$C_{max} \geq S_{ij} + \frac{p_{ij}}{N_{ij}} + t_{jiw} \quad i \in \{1,2,\dots,5\}; j \in \mathcal{J} \quad (11)$$

$$x_{i'ij} = \{0,1\}, y_{ij'j} = \{0,1\}, N_{ij} \in N^+, i \in \{1,2,\dots,5\}; j \in \mathcal{J} \quad (12)$$

Equations (1) define the objective to minimize both makespan and the total sum of penalty for machine load. Constraints (2) ensure that a job can be processed at least one machine at each stage, which is fully different from the traditional scheduling problem. Constraints (3) restrict that one and only one machine can process the jobs at encapsulation stage. Constraints (4) ensure that the number of available machines to process each operation is within its respective range. Constraints (5) guarantee that the job cannot be started before it is transported from the warehouse to the stage conducting its first operation. Constraints (6) impose that the first operation of the job at each stage cannot start until its setup time is completed. Constraints (7)-(8) ensure that each job cannot be started until its preceding operations are completed and the job is conveyed to stage i . Constraints (9)-(10) ensure that each job cannot be started before its preceding jobs at the same stage are completed and its setup time is finished. Constraints (11) specify that the makespan is equal to the time that the last job is completed and conveyed to the warehouse. Constraints (12) restrict that decision variables x and y are binary, and N_{ij} is a positive integer number. This model can be easily extended to other scheduling problems such as job shop and group job shop to some extent.

The two objectives in this problem are in conflict with each other because when a job is processed on multiple machines at one stage, its processing time decreases whilst the corresponding machine load or interference increases. In addition, due to the NP-

hard characteristic of the problem, large scale problem cannot be addressed within a reasonable computational time by the proposed MIP model and exact algorithms. Therefore, an efficient MODGWO should be developed to solve this type of large scale instances in this paper.

3. The proposed MODGWO for WSP

In this section, we first introduce an original GWO, and then present the proposed MODGWO for WSP in detail.

3.1 The original GWO

GWO is a new metaheuristic inspired by the grey wolf hunting for the prey in nature [22]. The main steps of GWO are provided in this subsection.

To establish a social hierarchy of wolves when developing GWO, all the grey wolves are classified into four kinds of the wolf according to fitness value. The best wolf (solution) in GWO is denoted as the alpha (α). Similarly, the second and third best wolves are called beta (β) and delta (δ) respectively. The rest of wolves are considered to be omega (ω). In GWO, the search process is primarily guided by α , β and δ . The ω wolves obey these three wolves.

Grey wolves tend to encircle the prey when the hunt. To model encircling behavior, the equation is defined as follows:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (13)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (14)$$

where t represents the current iteration, $\vec{X}_p$ denotes the position vector of the prey and $\vec{X}$ denotes the position vector of a grey wolf. The vector $\vec{A}$ and $\vec{C}$ are formulated as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (15)$$

$$\vec{C} = 2\vec{r}_2 \quad (16)$$

where $\vec{a}$ is linearly decreased from 2 to 0 during the search process, and it is used to emphasize exploration and exploitation respectively. $\vec{r}_1$ and $\vec{r}_2$ are random vectors in range [0,1]. The hunt is usually guided by the alpha. The beta and delta might also join in on the hunting. However, the position of prey (optimum) is unknown in an abstract search space. To simulate the hunting behavior of grey wolves, assume that the alpha,

beta, and delta have better knowledge about the potential location of the prey. Therefore, the first three best solutions found so far are saved and guide the other wolves toward the potential location of the prey. The equation of hunting is formulated as follows [22].

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \vec{D}_\gamma = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \quad (17)$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha, \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta, \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \quad (18)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (19)$$

To sum up, the GWO is beginning with a random population of grey wolves. The search process is mainly guided by the α , β and δ . They diverge from each other to search for prey when $|\vec{A}| > 1$ and converge to attack prey when $|\vec{A}| < 1$. Finally the optimal solution (i.e., the prey) is output if the stop criterion is satisfied.

3.2 The proposed MODGWO

As stated previously, we propose a MODGWO to tackle the above WSP. A basic flowchart of MODGWO is presented in **Fig. 3**. The following subsections give a detailed explanation about the improvement steps of MODGWO. These improvement procedures mainly include the following steps: encoding and decoding scheme, initialization, social hierarchy, search for the prey, and reduction machine load strategy.

Fig. 3. Outline of the proposed MODGWO

3.2.1 Encoding and decoding scheme

The encoding and decoding scheme is an important procedure before applying the proposed MODGWO to a particular problem. It constructs a bridge between solution space of the considered problem and the search space of the search process. Therefore, designing an appropriate encoding and decoding scheme is a very essential issue that affects the performance of the algorithm.

Unlike other scheduling problems, this welding scheduling problem has to deal with the processing sequence of jobs and the machine assignment of each operation simultaneously. Thus, we propose a new encoding scheme which contains two parts. The first part is the job permutation denoted by notation π . The second part is the machine assignment matrix denoted by symbol N . To illustrate this encoding scheme, we give an example with 4 jobs and 5 stages **as follows**.

$$\pi = [2,4,3,1]$$

$$N = \begin{bmatrix} 1,1,3,2 \\ 1,1,2,1 \\ 2,2,3,1 \\ 1,1,1,1 \\ 2,2,3,1 \end{bmatrix}$$

In this solution representation, the job sequence is [2-4-3-1]. The element N_{ij} in the machine assignment matrix denotes the number of the machine to process the operation of job j at the stage i . To further illustrate this solution representation, **Fig. 4** presents a Gantt chart of the above solution representation. For example, N_{13} represents that the number of machines that are assigned to job 3 at stage 1 (i.e., O_{13}) is 3. The O_{13} is the red box with two white bars that denotes the additional two machines to process the operation. The color box without white bars represents the operation with normal processing times (e.g., O_{12}), that is, the operation O_{12} is processed by only one machine. The makespan is 255.83 and machine load is 32 in this Gantt chart (we assume that the coefficient of penalty for all machine load is equal to 1).

Jobs	Normal processing times				
	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
1	10	15	30	20	25
2	30	10	20	25	15
3	35	10	25	5	30
4	10	30	20	10	15

Jobs	Transportation times					
	w to s1	s1 to s2	s2 to s3	s3 to s4	s4 to s5	s5 to w
1	6	10	15	20	10	14
2	5	10	20	15	15	14
3	4	15	10	25	10	14
4	3	10	10	15	5	14

Jobs	Setup times			
	1	2	3	4
1	-	10	20	15
2	12	-	10	20
3	5	9	-	16
4	10	5	9	-

Jobs	Setup times				
	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
1	10	5	14	13	6
2	4	8	7	10	18
3	8	10	5	12	7
4	9	18	7	16	12

Fig. 4. An example of solution representation

3.2.2 Initialization

Based on the above encoding scheme, each solution (wolf) is constructed by a permutation vector and a machine assignment matrix. To ensure the high quality and good diversity of solutions, one-third of the initial population corresponding to different machine assignment matrixes are created by NEH method [32]. The other initial individuals are randomly generated based on the proposed encoding scheme.

3.2.3 Social hierarchy

The core idea of the original GWO is that the search process is guided by the best three solutions (wolves). However, due to Pareto dominance nature of the MOP, the optimal result is usually not a single solution but a set of non-dominated solutions also called ‘trade-off solutions’ in MOP. That is, for a trade-off solution, the improvement of one objective can cause the deterioration of at least another objective. Thus, each solution is related to a rank equal to the non-dominated level. The population can be divided into several level ranks according to Pareto dominance relationship. The solutions at the first level rank (non-dominated solution or trade-off solutions) can be denoted as α solutions followed by the second and third level rank as β and δ if the population exists more than three ranks. In this research, the first best three solutions are obtained by the following assumptions.

1. If the current population is at the non-dominated level or first rank, α , β and δ are

- randomly selected from the population.
2. If the current population only has two ranks, α and β are randomly obtained from the first and second rank, respectively. δ is randomly selected from solutions at the second rank.
 3. If the current population has more than two ranks, α , β and δ are obtained from the first, second, and third rank, respectively.

3.2.4 Search for the prey

The search operator of GWO is originally developed for solving continuous optimization problems. Thus, it cannot be directly adopted to address combinatorial problems. To ensure the feasibility of the obtained solutions, the modified search operator is composed of two parts, which are formulated as follows.

For the first part, namely, job permutation:

$$\pi_i^{t+1} = \begin{cases} \text{shift}(\pi_i^t, A \cdot (C \cdot \pi_\alpha^t - \pi_i^t)) & \text{if } \text{rand} < \frac{1}{3} \\ \text{shift}(\pi_i^t, A \cdot (C \cdot \pi_\beta^t - \pi_i^t)) & \text{else if } \text{rand} < \frac{2}{3} \\ \text{shift}(\pi_i^t, A \cdot (C \cdot \pi_\delta^t - \pi_i^t)) & \text{otherwise} \end{cases}$$

where $\text{shift}(x, d)$ function means that the element x can be moved to the right or left with $|d|$ units, note that if the shift units are beyond the boundary range, the boundary element is connected to that in the opposite direction, rand is a uniform random value between $[0,1]$, A and C are the control factor. In this paper, A and C are set to 1.0. **Fig. 5** gives an example of the proposed search operator.

Fig. 5. An example of search for the prey

For the second part, namely, machine assignment matrix:

$$N_{i,j}^{t+1} = \begin{cases} N_{i,j}^t \in N(\alpha)_{i,j}^t \text{ or } N(\beta)_{i,j}^t \text{ or } N(\delta)_{i,j}^t & \text{if } \text{rand} \leq a \\ N_{i,j}^t & \text{otherwise} \end{cases}$$

where the new machine assignment element $N_{i,j}^{t+1}$ is generated from two rules. (a)

previous good solutions and (b) random selection. In the first rule, the trial element is selected from one of the first best machine assignment matrix elements with the probability of a ($a = 1 - t/T_{max}$), t and T_{max} are the current iteration number and the maximum iteration number, respectively. In random selection, $N_{i,j}^{t+1}$ can be any feasible value not limited to the first rule with the probability of $(1 - a)$.

3.2.5 Reduction machine load procedure

Local search approaches are usually embedded into the single and multi-objective metaheuristics for improving the quality of solutions [33]. However, local search in a MOP is different from that in a single-objective problem. As a matter of fact, the improvement in one objective may lead to the deterioration in another objective in the MOP [11]. In this paper, therefore, any local search strategies are not applied in the proposed algorithm. Instead, one improvement procedure is designed to further enhance the quality of solutions. This procedure does not need to change the job sequence, but can reduce the total penalty value of machine load to some extent. The following algorithm states the reduction of machine load procedure in **Fig. 6**.

Reduction of machine load procedure

Input: A given schedule.

Output: An improved schedule with less machine load.

Strategy 1 in the last stage.

For $i = 1, 2, \dots, n - 1$ **do**

While $N_{m,\pi(i)} > 1$ **do**

 Set $N_{m,\pi(i)} = N_{m,\pi(i)} - 1$, where $N_{m,\pi(i)}$ represents the number of machine to process the job in the i -th position at the last stage.

 Set $C_{m,\pi(i)} = S_{m,\pi(i)} + p_{m,\pi(i)}/N_{m,\pi(i)}$, where $C_{m,\pi(i)}$ represents the completion time of the job in the i -th position at the last stage, $p_{m,\pi(i)}/N_{m,\pi(i)}$ denotes the actual processing time of the job in the i -th position at the last stage. $S_{m,\pi(i)}$ is the start time of the job in the i -th position at the last stage.

 Set $setupC_{m,\pi(i+1)} = C_{m,\pi(i)} + setup_{m,\pi(i+1)}$, where $setup_{m,\pi(i+1)}$ represents the setup time between the $(i+1)$ -th job and the i -th job, $setupC_{m,\pi(i+1)}$ is the completion time of setup operation of the job in the $(i+1)$ -th position at the last stage.

 Set $wt_{m,\pi(i)} = S_{m,\pi(i+1)} - setupC_{m,\pi(i+1)}$, where $wt_{m,\pi(i)}$ represents the idle time between the i -th job and the $(i+1)$ -th job, $S_{m,\pi(i+1)}$ denotes the start time of the job in the $(i+1)$ -th position at the last stage, $setupC_{m,\pi(i+1)}$ is the completion time of setup operation of the job in the $(i+1)$ -th position at the last stage.

end while

if $wt_{m,\pi(i)} < 0$

 Set $N_{m,\pi(i)} = N_{m,\pi(i)} + 1$, and update $C_{m,\pi(i)}$, $setupC_{m,\pi(i+1)}$, and $wt_{m,\pi(i)}$.

end if

end for

Strategy 2 in the job in the last position.

For $k = 1, 2, \dots, m - 1$ **do**

While $N_{k,\pi(n)} > 1$ **do**
 Set $N_{k,\pi(n)} = N_{k,\pi(n)} - 1$.
 Set $C_{k,\pi(n)} = S_{k,\pi(n)} + p_{k,\pi(n)}/N_{k,\pi(n)}$.
 Set $TTC_{k,\pi(n)} = C_{k,\pi(n)} + TT_{k,\pi(n)}$, where $TT_{k,\pi(n)}$ represents the transportation time of the n -th job (the job in the last position) carried from stage k to $k+1$, $TTC_{k,\pi(n)}$ is the completion time of the n -th job carried from stage k to $k+1$.
 Set $wt_{k,\pi(n)} = S_{k+1,\pi(n)} - TTC_{k,\pi(n)}$, where $wt_{k,\pi(n)}$ represents the idle time of the n -th job between the stage k and $k+1$.
end while
if $wt_{k,\pi(n)} < 0$
 Set $N_{k,\pi(n)} = N_{k,\pi(n)} + 1$, and update $C_{k,\pi(n)}$, $TTC_{k,\pi(n)}$, and $wt_{k,\pi(n)}$.
end if
end for

Fig. 6. Reduction of machine load procedure

Fig. 7. Gantt charts for strategy 1

To explain this strategy 1 in the last stage, **Fig. 7** shows a procedure of strategy 1. As can be observed from **Fig. 7**, the schedule of the case (a) has a large enough idle time between the job 1 and job 2 at the last stage, while there is a smaller idle time in schedule of the case (b) by means of controlling processing times of job 1. That is, the processing time of the job 1 on the last stage can be expanded by decreasing the corresponding machine number, which leads to a reduction in machine load. On the other hand, the makespan maintains the same for different solutions, and thereby the old solution is dominated.

Fig. 8. Gantt charts for strategy 2

Fig. 8 illustrates the strategy 2. The schedule of the case (a) has a large enough idle time between two consecutive stages for the last job, while there is a smaller idle time in schedule of the case (b). The processing time of the last job on the former stage is expanded by decreasing machine number, which causes a reduction in machine load. On the other hand, the makespan maintains the same for two solutions, and thereby the old solution is dominated.

3.2.6 Replacement scheme

This phase of the MODGWO is different from that of the original GWO when solving MOPs, the newly generated solutions are evaluated in terms of both fitness values considered (i.e., makespan and total penalty of machine load). In particular, all the solutions in the population are sorted according to the non-dominated sorting technique which favors simultaneously the population diversity and the convergence toward Pareto optimal solutions. Thus, each solution is associated with a rank equal to its non-dominance level. Then, within each front a crowding distance strategy is used to define an ordering among individuals. The crowding distance is an indicator of the density of the solutions surrounding a current solution and can be obtained by calculating the sum of the distances to the closest individual along each objective. To achieve wide-spread Pareto fronts (solutions differently balancing makespan and total penalty of machine load), individuals with a large crowding distance are superior to ones with a smaller crowding distance for solutions that belong to the same non-dominated level. The new generated population Q_t is combined with existing parent population P_t to form $Q_t \cup P_t$ solution vectors. Then non-dominated sorting strategy and crowding distance technique are performed on the combined population $Q_t \cup P_t$. Finally, the best population with size N (N is population size) is selected from the combined population based on ranking and crowding distance for the next generation.

4. Numerical experiment

To evaluate the performance of the proposed MODGWO, we compare it with two

famous MOEAs: NSGA-II [34] and SPEA2 [35]. To make a fair comparison between these MOEAs. The same encoding scheme and reduction machine load strategy are used as MODGWO. Furthermore, in NSGA-II and SPEA2, the same operators are employed: binary tournament selection, partially mapping crossover (PMX) for the first part of solution representation and insert operation for the second part, swap operation for two parts in mutation. The details about these operators can be referred to [36-38]. All algorithms are coded in Java. The experimental tests are implemented on a computer with Intel Core i5, 2.39GHz, 4GB RAM, with Windows 8 operating system.

4.1 Performance metrics

Three performance metrics are used to test the behavior of the MOEAs and are given as follows.

(1) Spread (Δ) [34]. This metric is a diversity indicator that measures the extent of spread achieved in the front. The metric is defined as:

$$\Delta = \frac{\sum_{j=1}^m d_j^e + \sum_{i=1}^n |d_i - \bar{d}|}{\sum_{j=1}^m d_j^e + n \cdot \bar{d}} \quad (20)$$

where d_i denotes the Euclidean distance of each point in Pareto front (PF) to its closest point in the optimal Pareto front (PF*), $\bar{d}$ is the average value of all d_i , d_j^e is the Euclidean distance between the extreme solutions in the j -th objective and the boundary solutions of the obtained PF, n is the number of PF, and m is the number of objectives. If the spread value is zero, it means all the members of Pareto optimal front are evenly spaced. Lower values indicates better distribution and diversity.

(2) Generational Distance (GD) [34]. The GD metric indicates how far the obtained PF is from PF*. This metric is formulated as:

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \quad (21)$$

where n is the number of Pareto front points found so far and d_i is the Euclidean distance between the i -th member of PF obtained and the nearest member of PF*. A low GD value is desirable and denotes a good convergence performance. A normalization method is adopted in this metric.

(3) Inverse Generational Distance (IGD) [39]. This is a variant of the GD but represents a combined or comprehensive indicator. It measures the distances between each solution consisting of the optimal Pareto front and obtained front. IGD can be defined as follows:

$$IGD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \quad (22)$$

where n is the number of the optimal Pareto front and d_i is the Euclidean distance between each point of that front and the nearest member of the approximation. Fronts with a lower IGD value are desirable. This metric uses a normalization method.

Note that the true PF* of the considered problem may be unknown, thus, the non-dominated solutions found using different MOEAs for each instance in all the independent runs are regarded as PF* (reference points) on that instance.

4.2 Test instances

To assess the performance of different MOEAs, test data for different configurations with regard to problem scale should be provided. The instances we randomly generate are defined as shown in **Table 1**. Each instance can be labeled in the form of “ n_s ”. For example, ‘10_5’ represents that the problem is featured by 10 jobs and 5 stages.

Table 1. Data set distribution

input variables	distribution
number of jobs (n)	20, 40, 60 ,80, 100
number of stage (m)	2, 3, 4, 5
number of available machines at each stage ($\mathcal{M}$)	discrete uniform [1,4]
normal processing time (p)	discrete uniform [30,50]
setup time (sT)	discrete uniform [5,20]
transportation time (t)	discrete uniform [10,20]
coefficient of penalty for machine load (u)	continuous uniform [1,5]

4.3 Parameter settings

It is well known that that the parameter settings can affect the behavior of the algorithm. Thus, pilot experiments were conducted to assess different parameter configurations. Most parameters are constant in experiments. However, some parameters are sensitive to the problem scale. For example, a population with 100 size may be large for one 10-job instance, but may be small for one 100-job instance. Therefore, population size depends dynamically on the problem scale. To make a fair comparison, all MOEAs use the same maximum number of function evaluations (NFEs). More preciously, the parameters are summarized in **Table 2**. Each experiment was conducted 30 independent times on each test problem for each algorithm.

Table 2. The other parameter settings of MODGWO, NSGA-II, and SPEA2

MODGWO	NSGA-II	SPEA2
--------	---------	-------

Population size: 50 and NFEs: 50,000 (20 and 40 jobs) Population size: 100 and NFEs: 80,000 (60 and 80 jobs) Population size: 150 and NFEs: 100,000 (100 jobs)	Population size: 50 and NFEs: 50,000 (20 and 40 jobs) Population size: 100 and NFEs: 80,000 (60 and 80 jobs) Population size: 150 and NFEs: 100,000 (100 jobs)	Population size: 50 archive size: 50 and NFEs: 50,000 (20 and 40 jobs) Population size: 100 archive size: 100 and NFEs: 80,000 (60 and 80 jobs) Population size 150 archive size: 150 and NFEs: 100,000 (100 and 200 jobs)
Search for the prey (non- dominated solutions)	PMX and insert operation Crossover rate: 0.9	PMX and insert operation Crossover rate: 0.9
	swap operation mutation rate: 0.2	swap operation mutation rate: 0.2

4.4 Comparison MODGWO with other MOEAs

To evaluate the effectiveness of the proposed method, MODGWO is compared with other MOEAs including NSGA-II and SPEA2. **Table 3** shows the obtained statistical results, where mean represents the average value, std denotes the standard deviation. The bottom rows of this table list the ratio of dominant performance for the corresponding metric. For example, 3/21 indicates that the corresponding algorithm outperforms its rivals on 3 out of 21 problems for a given metric. The optimal results are highlighted with bold in the **Table 3**.

Table 3. Mean and standard deviation of all metrics obtained by NSGA-II, SPEA2, and MODGWO

Problem		NSGA-II Mean/std			SPEA2 Mean/std			MODGWO Mean/std		
		GD	Δ	IGD	GD	Δ	IGD	GD	Δ	IGD
20_2	mean	1.05e-02	1.28e+00	1.60e-02	9.21e-03	1.28e+00	1.58e-02	4.89e-03	5.33e-01	4.04e-03
	std	1.71e-03	5.10e-02	2.87e-03	1.83e-03	4.71e-02	2.68e-03	2.64e-03	1.12e-01	1.01e-03
20_3	mean	1.83e-01	1.04e+00	2.12e-01	1.83e-01	1.03e+00	2.09e-01	5.67e-02	1.39e+00	2.34e-02
	std	3.97e-02	4.62e-02	3.04e-02	4.10e-02	6.21e-02	3.61e-02	2.71e-02	8.28e-02	5.96e-03
20_4	mean	1.59e-02	1.17e+00	2.30e-02	1.43e-02	1.16e+00	2.45e-02	7.15e-03	5.93e-01	7.06e-03
	std	2.01e-03	4.74e-02	3.34e-03	2.12e-03	3.26e-02	3.50e-03	2.83e-03	8.02e-02	2.04e-03
20_5	mean	3.52e-02	1.15e+00	4.16e-02	3.28e-02	1.12e+00	4.26e-02	1.28e-02	1.03e+00	2.89e-02
	std	4.68e-03	7.94e-02	5.36e-03	2.91e-03	8.94e-02	7.04e-03	1.02e-02	8.44e-02	7.18e-03
40_2	mean	1.56e-02	1.13e+00	1.80e-02	1.45e-02	1.11e+00	1.83e-02	3.23e-03	7.05e-01	5.86e-03
	std	8.21e-04	2.39e-02	1.21e-03	1.32e-03	2.49e-02	1.64e-03	1.20e-03	5.64e-02	1.43e-03
40_3	mean	9.56e-01	1.31e+00	3.32e-01	8.66e-03	1.33e+00	1.64e-01	2.53e-03	1.33e+00	2.89e-02
	std	2.34e-03	4.89e-02	2.24e-01	1.06e-03	8.22e-02	1.34e-01	8.95e-04	6.03e-02	1.86e-03
40_4	mean	2.65e-01	1.34e+00	5.09e-01	1.79e-01	1.35e+00	4.50e-01	7.56e-02	1.23e+00	4.93e-01
	std	8.76e-02	4.59e-02	1.91e-02	1.14e-01	2.05e-01	9.35e-02	3.17e-02	7.29e-02	5.64e-03
40_5	mean	2.43e-01	1.22e+00	2.86e-01	1.91e-01	1.26e+00	2.17e-01	1.32e-01	9.14e-01	1.09e-01
	std	5.32e-02	4.82e-02	7.72e-02	7.03e-02	5.69e-02	6.84e-02	7.62e-02	1.23e-01	3.24e-02
60_2	mean	4.93e-01	1.44e+00	1.57e-01	4.13e-01	1.46e+00	6.95e-02	4.59e-02	1.46e+00	5.95e-02

	std	1.91e-01	3.84e-02	8.24e-02	2.70e-01	9.65e-02	2.00e-02	2.50e-02	6.72e-01	6.81e-04
60_3	mean	1.60e-01	1.00e+00	9.72e-02	1.76e-01	1.00e+00	2.00e-02	5.53e-03	1.14e+00	8.06e-03
	std	9.42e-02	0	1.54e-02	4.54e-02	0	1.32e-02	3.39e-03	1.42e-01	5.14e-03
60_4	mean	1.34e-01	1.11e+00	8.20e-02	2.04e-01	1.07e+00	8.21e-02	4.00e-02	1.36e+00	1.01e-02
	std	2.18e-02	7.12e-02	7.36e-03	1.23e-01	5.16e-02	5.20e-03	5.20e-02	6.46e-02	3.65e-03
60_5	mean	2.91e-01	1.04e+00	2.53e-01	4.29e-01	1.03e+00	2.62e-01	4.63e-02	1.36e+00	1.59e-02
	std	1.58e-02	2.52e-02	1.22e-02	2.81e-02	3.52e-02	9.77e-03	3.13e-02	4.51e-02	7.02e-03
80_2	mean	1.31e-02	1.12e+00	1.49e-02	1.74e-02	1.08e+00	1.48e-02	2.08e-03	8.48e-01	3.73e-03
	std	7.04e-04	2.16e-02	9.40e-04	1.07e-03	2.42e-02	1.01e-03	5.46e-04	6.12e-02	8.37e-04
80_3	mean	8.87e-01	1.42e+00	5.13e-01	9.31e-01	1.43e+00	1.74e-01	1.12e-01	1.50e+00	5.07e-02
	std	2.44e-01	2.71e-02	2.11e-01	1.42e+00	1.84e-01	1.30e-01	5.34e-02	3.71e-02	1.64e-02
80_4	mean	8.69e-01	1.40e+00	5.73e-01	7.91e-01	1.33e+00	2.46e-01	1.96e-01	1.44e+00	5.45e-02
	std	4.64e-01	5.03e-02	1.76e-01	1.54e+00	2.38e-01	1.24e-01	9.01e-02	9.31e-02	2.17e-02
80_5	mean	7.58e-02	1.30e+00	1.29e-01	1.10e-01	1.26e+00	1.40e-01	4.24e-02	1.11e+00	8.40e-02
	std	7.74e-03	5.70e-02	2.15e-02	1.42e-02	8.87e-02	2.87e-02	2.43e-02	1.73e-01	1.58e-02
100_2	mean	8.76e-02	1.00e+00	9.27e-01	8.62e-02	9.71e-01	7.56e-01	7.93e-02	1.00e+00	8.54e-01
	std	3.23e-01	6.63e-04	3.12e+00	1.02e-01	1.91e-01	1.12e+00	6.01e-01	1.54e-03	2.93e+00
100_3	mean	9.70e-02	1.32e+00	1.11e-01	1.24e-01	1.31e+00	9.13e-02	3.44e-02	1.04e+00	5.46e-02
	std	3.04e-02	8.06e-02	3.32e-02	4.31e-02	9.31e-02	2.62e-02	5.85e-02	1.57e-01	5.38e-03
100_4	mean	7.05e-02	1.33e+00	1.23e-01	8.45e-02	1.30e+00	1.00e-01	3.68e-02	1.07e+00	5.73e-02
	std	1.44e-02	7.52e-02	3.82e-02	3.02e-02	7.87e-02	2.32e-02	3.41e-02	1.74e-01	4.22e-03
100_5	mean	5.23e-01	1.26e+00	9.56e-01	6.82e-01	1.21e+00	9.56e-01	3.01e-01	9.81e-01	3.60e-01
	std	7.62e-02	5.30e-02	4.82e-01	1.36e-01	5.31e-02	2.22e-01	2.86e-01	1.83e-01	8.57e-02
Hit rate	mean	0/20	4/20	0/20	0/20	6/20	2/20	20/20	11/20	18/20

Table 3 presents the statistical results of GD, Spread, and IGD. From this table, we can observe that the proposed MODGWO outperforms its counterparts for most instances. Especially on the comprehensive metric IGD and convergence metric GD, the advantage of the MODGWO is overwhelming except for several problems. MODGWO is also better than NSGA-II and SPEA2 with regard to Spread metric. The good performance of the proposed MODGWO can be attributed to the social hierarchy scheme. The adoption of the social hierarchy scheme can improve the quality of population since the search process is guided by three good solutions. Meanwhile, to maintain diversity of population, three good solutions may be selected from different non-dominated levels, which can improve search diversity. Thus, MODGWO can effectively maintain a good balance between exploration and exploitation. By contrast, NSGA-II and SPEA2 do not contain this hierarchy scheme. Additionally, we can know from crossover operation that such defined crossover operator is possible to destroy a good operation sequence for NSGA-II and SPEA2.

Table 4. Wilcoxon signed rank test results based on the best metrics for each instance with 30 independent runs (a level of significant $\alpha=0.05$)

NO.	GD	Δ	IGD
-----	----	----------	-----

	MODGWO vs. NSGA-II			MODGWO vs. SPEA2			MODGWO vs. NSGA-II			MODGWO vs. SPEA2			MODGWO vs. NSGA-II			MODGWO vs. SPEA2		
	R ⁺	R ⁻	<i>p</i> -value/win	R ⁺	R ⁻	<i>p</i> -value/win	R ⁺	R ⁻	<i>p</i> -value/win	R ⁺	R ⁻	<i>p</i> -value/win	R ⁺	R ⁻	<i>p</i> -value/win	R ⁺	R ⁻	<i>p</i> -value/win
20_2	457	8	3.88e-06/+	438	27	2.37e-05/+	465	0	1.73e-06/+	465	0	1.73e-06/+	465	0	1.73e-06/+	465	0	1.73e-06/+
20_3	465	0	1.73e-06/+	464	1	1.92e-06/+	0	465	1.73e-06/-	0	465	1.73e-06/-	465	0	1.73e-06/+	465	0	1.73e-06/+
20_4	465	0	1.73e-06/+	462	3	2.35e-06/+	465	0	1.73e-06/+	465	0	1.73e-06/+	465	0	1.73e-06/+	465	0	1.73e-06/+
20_5	457	8	3.88e-06/+	457	8	3.88e-06/+	423	42	8.91e-06/+	393	72	9.63e-04/+	465	0	1.73e-06/+	457	8	3.88e-06/+
40_2	465	0	1.73e-06/+	465	0	1.73e-06/+	465	0	1.73e-06/+	465	0	1.73e-06/+	465	0	1.73e-06/+	457	8	3.88e-06/+
40_3	465	0	1.73e-06/+	446	19	1.13e-05/+	146	319	7.52e-03/-	257	208	6.14e-01/=	465	0	1.73e-06/+	465	0	1.73e-06/+
40_4	465	0	1.73e-06/+	403	62	4.53e-04/+	429	36	5.31e-05/+	418	47	1.36e-04/+	407	58	3.32e-04/+	134	331	4.28e-02/-
40_5	432	33	4.07e-05/+	370	95	4.68e-03/+	465	0	1.73e-06/+	465	0	1.73e-06/+	465	0	1.73e-06/+	465	0	1.73e-06/+
60_2	464	1	1.92e-06/+	460	5	2.88e-06/+	140	325	5.71e-02/=	200	265	5.03e-01/=	465	0	1.73e-06/+	396	69	7.71e-04/+
60_3	465	0	1.73e-06/+	465	0	1.73e-06/+	40	425	7.51e-05/-	40	425	7.51e-05/-	465	0	1.73e-06/+	465	0	1.73e-06/+
60_4	357	108	1.04e-02/+	450	15	7.69e-06/+	465	0	1.73e-06/+	465	0	1.73e-06/+	337	128	3.16e-02/+	300	165	1.65e-01/=
60_5	461	4	2.60e-06/+	379	86	2.58e-03/+	216	249	7.34e-01/=	407	58	3.32e-04/+	459	6	3.18e-06/+	448	17	9.32e-06/+
80_2	465	0	1.73e-06/+	465	0	1.73e-06/+	465	0	1.73e-06/+	465	0	1.73e-06/+	465	0	1.73e-06/+	465	0	1.73e-06/+
80_3	465	0	1.73e-06/+	461	4	2.60e-06/+	10	455	4.73e-06/-	76	389	1.28e-03/-	465	0	1.73e-06/+	437	28	2.60e-05/+
80_4	465	0	1.73e-06/+	452	13	6.34e-06/+	82	383	1.96e-03/-	44	421	1.06e-04/-	465	0	1.73e-06/+	464	1	1.92e-06/+
80_5	449	16	8.47e-06/+	465	0	1.73e-06/+	433	32	3.72e-05/+	400	65	5.71e-04/+	465	0	1.73e-06/+	465	0	1.73e-06/+
100_2	244	221	8.13e-01/=	418	47	1.36e-04/+	141	324	5.98e-02/=	89	376	3.16e-03/-	258	207	6.00e-01/=	87	378	2.77e-03/-
100_3	401	64	5.29e-04/+	448	17	9.32e-06/+	450	15	7.69e-06/+	456	9	4.29e-06/+	465	0	1.73e-06/+	462	3	2.35e-06/+
100_4	407	58	3.32e-04/+	431	34	4.45e-05/+	438	27	2.37e-05/+	429	36	5.31e-05/+	462	3	2.35e-06/+	463	2	2.13e-06/+
100_5	382	83	2.11e-03/+	443	22	1.49e-05/+	449	16	8.47e-06/+	438	27	2.37e-05/+	465	0	1.73e-06/+	465	0	1.73e-06/+
+/-/-	19/1/0			20/0/0			12/3/5			13/2/5			19/1/0			17/1/2		

Table 5. Wilcoxon signed rank test results based on mean results for all metrics on 20 instances (a level of significant $\alpha=0.05$)

Metrics	MODGWO VS. NSGA-II				MODGWO VS. SPEA2			
	R ⁺	R ⁻	<i>p</i> -value	win	R ⁺	R ⁻	<i>p</i> -value	win
GD	204	6	2.19e-04	+	202	8	2.93e-04	+
Δ	133	77	1.26e-01	=	129	81	1.71e-01	=
IGD	210	0	8.86e-05	+	189	21	1.70e-03	+

Due to the stochastic characteristic of all MOEAs, the statistical test should be conducted. A wilcoxon signed rank test [40] is used to detect the significant difference between the results obtained by different MOEAs. The confidence level for all tests is set to 95% (corresponding to a $\alpha=0.05$). The signs “+” or “-” indicate that the proposed MODGWO algorithm performs significantly better or worse than its compared one. The sign “=” denotes that there is no significant difference between MODGWO and its compared one. R⁺ is the sum of ranks for the problems in which the MODGWO outperformed the compared one, and R⁻ is the sum of ranks for the opposite. **Table 4**

reports the test results based on the best metrics for each instance with 30 independent runs. The p -values of the wilcoxon signed rank test are shown in **Table 4**. It can be observed from **Table 4** that MODGWO has clearly higher ‘+’ counts than other competitors. This implies that the MODGWO shows the statistically better performance than the two compared MOEAs. **Table 5** summarizes the test results based on mean values for all metrics on 20 instances. The p -values of the wilcoxon signed rank test are shown in **Table 5**. It means that MODGWO exhibits a significant improvement over NSGA-II and SPEA2 in terms of GD and IGD with $\alpha=0.05$.

To visualize the performance of these MOEAs, the obtained non-dominated solutions from the one run with the best IGD of each MOEA are selected to give graphical representation for the randomly selected small, medium and large scale instances in **Fig. 9**. These figures confirm some conclusions derived from the numerical analysis based on the performance criteria. As can be observed, the proposed MODGWO is capable of providing better solutions in comparison with NSGA-II and SPEA2 with regard to both quality and distribution.

Fig. 9. Non-dominated solutions with the best IGD run by different MOEAs

Based on the above results and presentations. The results in **Table 3** show the MODGWO is able to achieve the best results on most instances in terms of all metrics. The p -value in **Table 4** and **Table 5** denotes that the MODGWO outperforms significantly other MOEAs on most instances with regard to three metrics, especially for IGD and GD metrics. The MODGWO is a little worse than the other two MOEAs on “40-3”, “60-3”, “80-3” and “100-3” in terms of Spread metric. The reason for poor performance of MODGWO on these instances may be due to the hierarchy scheme. Although this hierarchy can improve the quality of solutions, it may contribute more to exploitation than exploration. Thus, the search agents take much time to search for the much more promising solution in neighborhood of the known good solution. However, it also lead to deterioration of exploration since this hierarchy also loses a larger chance to explore unknown areas of the search space. Consequently, it shows a poor performance on search diversity on these problems. Since the MODGWO exhibits significantly better performance compared with the other MOEAs on most instances, it seems that this behavior is not a major concern. It also reflects that hierarchy scheme has contributed most in searching for better results on this type of problem. It is concluded that MODGWO is much superior to NSGA-II and SPEA2 on the WSP.

The major reasons for the superiority performance of the MODGWO can be explained as follows. First, the adoption of the social hierarchy scheme can improve

the quality of population because the search process in the MODGWO is guided by three good solutions, and thus the MODGWO can quickly converge to the optimal (or near-optimal) solutions. This viewpoint has been supported by the above numerical experiment with regard to GD metric. Second, to maintain diversity of population, three good solutions may be selected from different non-dominated levels, which can improve search diversity. For Spread metric, MODGWO is also superior or competitive to the other MOEAs. Third, to further enhance comprehensive performance, the reduction machine load strategy is developed to improve solution quality. The effectiveness of this strategy has been verified in the following section. Therefore, the proposed MODGWO has a positive effect on solving WSP.

In the summary, the results in the above experiment verify that the MODGWO is much powerful than other well-known MOEAs on most instances. In addition, the statistical significant experiment between algorithms was conducted by the wilcoxon signed rank test. The superiority of the MODGWO is statistically significant on most instances. Therefore, it can be concluded that the MODGWO is very effective in solving this type of WSP.

4.5 Performance analysis of reduction machine load procedure

As stated in Section 3.2.5, the MODGWO has one main improvement strategy, namely, reduction machine load procedure. To evaluate the efficiency of the reduction machine load procedure, we test the MODGWO without reduction machine load procedure on a set of instances. In this experiment, MODGWO_{no} denotes that MODGWO has no reduction machine load procedure. MODGWO itself includes reduction machine load procedure. The other parameter settings of both MOEAs are the same for a fair comparison. **Table 6** reports the statistical metrics on two strategies over 30 independent runs.

Table 6. Mean and standard deviation value of all metrics between MODGWO_{no} and MODGWO

Problem	GD (Mean/std)		Spread (Mean/std)		IGD (Mean/std)	
	MODGWO _{no}	MODGWO	MODGWO _{no}	MODGWO	MODGWO _{no}	MODGWO
20_2	4.12e-03/1.4e-03	4.44e-03/1.4e-03	3.71e-01/6.3e-02	3.84e-01/6.0e-02	4.56e-03/1.2e-03	4.25e-03/9.9e-04
20_3	6.09e-02/3.9e-02	3.67e-02/1.2e-02	1.39e+00/8.6e-02	1.41e+00/8.2e-02	3.21e-02/1.1e-02	3.00e-02/1.0e-02
20_4	6.80e-03/2.8e-03	6.79e-03/2.3e-03	3.80e-01/7.2e-02	3.98e-01/5.9e-02	5.34e-03/1.5e-03	5.74e-03/1.3e-03
20_5	3.00e-02/1.2e-02	2.20e-02/1.0e-02	8.13e-01/1.3e-01	8.45e-01/1.4e-01	2.18e-02/5.5e-03	1.90e-02/6.4e-03
40_2	3.11e-03/9.7e-04	2.77e-03/9.6e-04	4.49e-01/5.7e-02	4.17e-01/3.4e-02	6.53e-03/2.1e-03	5.57e-03/2.0e-03
40_3	1.64e-02/7.0e-03	5.05e-02/4.8e-02	1.29e+00/7.4e-02	1.40e+00/1.0e-01	6.69e-02/1.2e-02	2.66e-02/9.0e-03
40_4	4.99e-02/5.3e-02	4.65e-02/6.9e-02	1.34e+00/6.9e-02	1.30e+00/6.7e-02	2.42e-02/7.9e-03	1.82e-02/7.4e-03
40_5	1.76e-02/1.4e-02	3.35e-02/2.7e-02	8.77e-01/1.2e-01	1.05e+00/1.3e-01	3.84e-02/7.3e-03	2.06e-02/6.9e-03
60_2	4.25e-02/2.5e-02	4.30e-02/2.4e-02	1.57e+00/8.9e-02	1.56e+00/7.5e-02	2.78e-02/1.1e-02	2.75e-02/7.1e-03
60_3	2.04e-02/2.3e-02	1.22e-02/1.3e-02	5.45e-01/1.1e-01	3.12e-01/2.3e-01	2.26e-03/3.1e-03	8.22e-04/3.5e-04
60_4	3.56e-02/4.2e-02	3.20e-02/3.9e-02	8.81e-01/1.4e-01	8.35e-01/1.4e-01	2.59e-02/8.3e-03	2.54e-02/6.9e-03
60_5	3.96e-02/3.5e-02	2.58e-02/2.5e-02	1.12e+00/1.8e-01	1.15e+00/9.1e-02	6.06e-02/1.5e-02	2.94e-02/8.8e-03

80_2	2.71e-03/5.3e-04	1.83e-03/6.2e-04	6.21e-01/4.6e-02	5.97e-01/6.7e-02	5.66e-03/1.5e-03	2.85e-03/6.9e-04
80_3	8.33e-02/3.6e-02	7.58e-02/3.4e-02	1.46e+00/9.8e-02	1.43e+00/4.9e-02	5.82e-02/2.1e-03	6.21e-02/1.9e-02
80_4	3.11e-02/1.3e-02	3.39e-02/1.9e-02	1.51e+00/5.4e-02	1.45e+00/9.9e-02	2.66e-02/1.0e-02	2.64e-02/9.3e-03
80_5	2.64e-01/1.8e-01	1.68e-01/1.3e-01	9.96e-01/2.0e-01	8.81e-01/2.3e-01	5.97e-02/1.8e-02	5.32e-02/1.9e-02
100_2	9.20e-02/3.0e-02	1.09e-01/5.6e-02	1.46e+00/4.4e-02	1.42e+00/1.2e-01	1.03e-01/2.2e-02	9.68e-02/3.1e-02
100_3	2.90e-01/4.0e-01	2.41e-02/1.3e-02	1.03e+00/2.2e-01	8.68e-01/7.4e-02	3.97e-02/8.9e-03	4.97e-02/9.4e-03
100_4	2.38e-01/1.7e-01	1.04e-01/1.1e-01	1.08e+00/2.3e-01	9.73e-01/1.5e-01	7.89e-02/2.5e-03	6.61e-02/2.2e-02
100_5	1.85e-01/1.6e-01	9.03e-02/1.1e-01	1.15e+00/1.8e-01	1.09e+00/1.3e-01	1.51e-01/5.0e-03	1.18e-01/2.9e-02
Hit rate	6/20	14/20	7/20	13/20	3/20	17/20

From **Table 6**, it can be observed that the MODGWO with reduction machine load procedure has a significantly better performance in comparison with the one without this procedure on most instances. This implies that the reduction machine load procedure can improve the behavior of the proposed MODGWO.

Table 7. Wilcoxon signed rank test results based on the best metrics for each instance with 30 independent runs (a level of significant $\alpha=0.05$)

NO.	GD			Δ			IGD		
	MODGWO vs. MODGWO _{no}			MODGWO vs. MODGWO _{no}			MODGWO vs. MODGWO _{no}		
	R ⁺	R ⁻	p-value/win	R ⁺	R ⁻	p-value/win	R ⁺	R ⁻	p-value/win
20_2	176	289	5.45e-01/=	190	275	3.82e-01/=	465	0	1.73e-06/+
20_3	367	98	5.67e-03/+	186	279	3.39e-01/=	282	183	3.09e-01/=
20_4	211	254	6.58e-01/=	126	339	2.85e-02/-	168	297	1.85e-01/=
20_5	347	118	1.85e-02/+	192	273	4.05e-01/=	305	160	1.36e-01/=
40_2	288	177	2.54e-01/=	371	94	4.38e-03/+	437	28	2.60e-05/+
40_3	31	434	3.41e-05/-	62	403	4.53e-04/-	465	0	1.73e-06/+
40_4	287	178	2.62e-01/=	155	310	1.11e-01/=	358	107	9.84e-03/+
40_5	57	408	3.07e-04/-	48	417	1.48e-04/-	461	4	1.73e-06/+
60_2	227	238	9.10e-01/=	215	250	7.19e-01/=	234	231	9.75e-01/=
60_3	357	108	1.04e-02/+	421	44	1.06e-04/+	465	0	1.73e-06/+
60_4	257	208	6.14e-01/=	292	173	2.21e-01/=	245	220	7.97e-01/=
60_5	289	176	2.45e-01/=	194	271	4.28e-01/=	461	4	2.60e-06/+
80_2	431	34	4.45e-05/+	347	118	1.85e-02/+	464	1	1.92e-06/+
80_3	269	196	4.53e-01/=	291	174	2.29e-01/=	219	246	7.81e-01/=
80_4	198	267	4.78e-01/=	266	199	4.91e-01/=	237	228	9.26e-01/=
80_5	325	140	5.71e-02/=	465	0	1.73e-06/+	287	178	2.62e-01/=
100_2	165	300	1.65e-01/=	350	115	1.57e-02/+	331	134	4.28e-02/+
100_3	389	76	1.29e-03/+	381	84	2.26e-03/+	68	397	7.16e-04/-
100_4	376	89	3.16e-03/+	357	108	1.04e-02/+	381	84	2.25e-03/+
100_5	328	137	4.95e-02/+	389	76	1.28e-03/+	459	6	3.18e-06/+
+/-/-	7/11/2			8/9/3			11/8/1		

Table 8. Wilcoxon signed rank test results based on mean results for all metrics on 20 instances (a level of significant $\alpha=0.05$)

Metrics	MODGWO VS. MODGWO _{no}			
	R ⁺	R ⁻	<i>p</i> -value	win
GD	154	56	6.74e-02	=
Δ	151	59	8.58e-02	=
IGD	180	30	5.10e-03	+

Table 7 reports the test results based on the best metrics for each instance with 30 independent runs. It can be observed from **Table 7** that MODGWO has obviously higher “+” counts than the other competitors. This implies that the MODGWO shows the statistically better performance than MODGWO_{no}. **Table 8** summarizes the test results based on mean values for all metrics on 20 instances. It is clearly shown in this table that MODGWO exhibits a significant improvement over MODGWO_{no} in terms of IGD with $\alpha=0.05$. In other words, the reduction machine load procedure is not sensitive to distribution or convergence performance of MODGWO for solving this type of scheduling problem. However, this procedure can assist to improve the overall performance of production systems, and consequently it contributes more to the comprehensive performance of MODGWO. The reason lies in the procedure of the problem property. The controllability of machine number may provide additional flexibility in searching for a promising solution which can improve the overall performance of the production system. In this case, a little improvement in machine load criterion will make it dominate many other solutions. Thus, MODGWO can benefit from the machine load strategy that aims at reducing machine load by adjusting machine number. By contrast, MODGWO_{no} doesn't include any heuristic rule developed to reduce machine load. Therefore, the reduction machine load procedure has a positive impact on MODGWO for solving the welding scheduling instances.

5. Case study

Consider the following WSP from a realistic welding workshop. This welding shop requires to process 10 types of jobs that should be passed through 5 stages in the same flow. The relative data including machine type, available machines at each stage, job number, normal processing times, coefficient of penalty for machine load at each stage, transportation and setup times is provided in **Table 9-14**.

Table 9

Available machines in the welding workshop

Machine type	Available machine number	Stage
TIG welding machine	2	Slicing of cover slab and gusset
CO ₂ shield semi auto welder	2	Web grouping
CO ₂ shield semi auto welder	2	Internal seam weld
Crane	1	Encapsulation

Table 10

The normal processing times of job on each stage

Job type	Span (m)	Job NO.	Processing time (min)				
			Slicing (Stage 1)	Web grouping (Stage 2)	Internal seam weld (Stage 3)	Encapsulation (Stage 4)	Fillet welding (Stage 5)
5tA5A6	22.5	1	20	25	30	20	24
5tA5A6	25.5	2	31	40	49	25	35
5tA5A6	28.5	3	35	40	45	36	39
10tA5A6	22.5	4	10	12	14	11	14
10tA5A6	25.5	5	29	35	41	27	33
10tA5A6	28.5	6	20	25	30	20	24
20t16tA5	22.5	7	31	40	49	25	35
20t16tA5	25.5	8	35	40	45	36	39
20t16tA5	28.5	9	10	12	14	11	14
20t16tA6	25.5	10	29	35	41	27	33

Table 11

The setup times of the first job to processed on each stage

Setup time(min)	Job1	Job2	Job3	Job4	Job5	Job6	Job7	Job8	Job9	Job10
Slicing	10	4	8	9	14	11	12	6	4	2
Web grouping	5	8	10	18	7	5	6	15	10	5
Internal seam weld	14	7	5	7	4	15	7	17	6	7
Assembling	13	10	12	16	9	3	10	2	14	8
Fillet welding	6	18	7	12	15	9	12	7	4	11

Table 12

The setup times between two consecutive jobs

Setup time(min)	Job 1	Job 2	Job 3	Job 4	Job 5	Job 6	Job 7	Job 8	Job 9	Job 10
Job 1	-	10	20	15	10	5	4	12	11	8
Job 2	12	-	10	20	6	9	10	18	2	13
Job 3	5	9	-	16	8	10	5	8	15	20
Job 4	10	5	9	-	14	17	6	10	12	15
Job 5	7	5	3	15	-	10	8	5	10	15
Job 6	15	10	8	5	8	-	8	12	10	12
Job 7	7	5	3	6	9	10	-	11	10	5
Job 8	11	15	10	6	8	5	12	-	12	7
Job 9	15	10	13	8	9	5	6	12	-	10
Job 10	12	8	10	6	9	5	10	15	12	-

Table 13

The transportation times of job between two consecutive stages or stage and warehouse

job	Transportation time(min)					
	Warehouse to stage 1	Stage 1 to Stage 2	Stage 2 to Stage 3	Stage 3 to Stage 4	Stage 4 to Stage 5	Stage 5 to warehouse
Job 1	4	10	15	20	10	14
Job 2	5	21	20	19	15	14
Job 3	4	15	10	25	16	14
Job 4	3	10	12	14	5	14
Job 5	6	9	5	12	17	14
Job 6	4	14	15	20	10	14
Job 7	5	21	15	24	15	14
Job 8	7	20	20	15	26	14
Job 9	10	5	8	24	10	14
Job 10	8	9	15	21	17	14

Table 14

The coefficient of penalty for machine load at each stage

Coefficient of penalty for machine load	stage 1	Stage 2	Stage 3	Stage 4	Stage 5
Unit machine	0.2	0.3	0.3	0.05	0.15

The Pareto front obtained by the proposed MODGWO and the original solution from the realist production are presented in **Fig. 10**. It is obviously observed that the proposed MODGWO is capable of providing high quality and good diversity solutions. It is also an interesting observation that the reduction of the makespan sacrifices its performance with regard to the total penalty value of machine load. For example, the minimum makespan (i.e., 413.5 min) can be identified at the point A where the total penalty value of machine load (i.e., 15.7) is maximum in **Fig. 10**. On the other hand, the minimum total penalty of machine load (i.e., 10) occurs at point B where the makespan (i.e., 597 min) is maximum. These observations further verify our views that the two objectives of the optimization problem are in conflict with each other. In addition, the point C (makespan is 564.5 min and machine load is 14.1) represents the original solution obtained by the First in and First out strategy (FIFO). We can find that not solutions by MODGWO dominate the original solution at point C. Also, the makespan at point A is decreased by 36.6% compared with the one at point C. The machine load at point B is reduced by 40% compared with the one at C. These points can be expressed as follows.

Solution at point A: $\pi = [4,6,10,8,3,5,9,2,7,1]$

$$N = \begin{bmatrix} 1,1,2,2,2,2,1,2,1,2 \\ 1,2,2,1,1,2,1,2,1,2 \\ 1,2,2,1,2,2,2,2,1,2 \\ 1,1,1,1,1,1,1,1,1,1 \\ 3,2,2,1,2,1,3,1,1,1 \end{bmatrix}$$

Solution at point B:

$$\pi = [4,6,1,5,10,8,3,7,9,2]$$

$$N = \begin{bmatrix} 1,1,1,1,1,1,1,1,1,1 \\ 1,1,1,1,1,1,1,1,1,1 \\ 1,1,1,1,1,1,1,1,1,1 \\ 1,1,1,1,1,1,1,1,1,1 \\ 1,1,1,1,1,1,1,1,1,1 \end{bmatrix}$$

Solution at point C:

$$\pi = [4,1,3,6,2,7,5,8,10,9]$$

$$N = \begin{bmatrix} 1,2,1,2,2,1,2,1,1,1 \\ 1,1,2,1,2,1,2,1,2,2 \\ 1,2,2,1,2,1,2,1,1,1 \\ 1,1,1,1,1,1,1,1,1,1 \\ 1,1,1,1,2,2,2,1,2,1 \end{bmatrix}$$

Fig. 10. Pareto front obtained by the proposed MODGWO

The Gantt charts of the solutions at point A and B are depicted in **Fig. 11** and **Fig. 12**, respectively. Note that the color box represents the actual processing time, where the number of the white bar in the color box denotes the number of the additional machines to process the operation. For example, the operation $O_{5,1}$ is assigned on three welding machines as this operation has additional two white bars in **Fig. 11**. Similarly, the operation $O_{4,1}$ is assigned on only one machine because it has no white bars in the Gantt chart of **Fig. 11**.

Fig. 11. The Gantt chart of the point A obtained by the proposed MODGWO

Fig. 12. The Gantt chart of the point B obtained by the proposed MODGWO

6. Conclusions and future work

In this paper, a real-world welding scheduling problem is studied from the theory and practical application perspectives. First, we proposed a bi-objective mathematical model which involves sequence dependent setup time, job dependent transportation times and controllable processing times. Then, we developed a new multi-objective grey wolf optimizer for solving this problem with the objective to minimize makespan

and the total penalty of machine load simultaneously. Furthermore, we compared the proposed MODGWO with well-known MOEAs: NSGA-II and SPEA2. The effectiveness of this MODGWO is supported by experiments on some instances in terms of convergence, spread and coverage metrics. In addition, the superiority of the MODGWO is significant by conducting the wilcoxon signed rank test. Finally, the MODGWO is successfully applied to a real-world case, and the result has a considerable instructive value in real-world production. In conclusion, the main contributions of this work are as follows:

- (1) A new scheduling model for welding process is formulated, which is easily extended to other types of scheduling problems.
- (2) A new multi-objective discrete grey optimizer is proposed to deal with this WSP.
- (3) A new encoding scheme is designed to accommodate the characteristic of the WSP.
- (4) A strategy that reduces machine load is proposed to improve the quality of solution.
- (5) This study can provide new enlightenment on modeling and addressing practical welding scheduling problems.

With respect to future work, an interesting issue worth studying is extending MODGWO to more complex scheduling problems such as a dynamic welding scheduling problem. Another research direction is to formulate a new mathematical model for welding scheduling problem from a new aspect of green manufacturing.

Acknowledgments

This research work was supported by the National Science Foundation of China (NSFC) under Grant nos. 51375004 and 51435009, National Key Technology Support Program under Grant no. 2015BAF01B04, Youth Science & Technology Chenguang Program of Wuhan under Grant no. 2015070404010187 and the Fundamental Research Funds for the Central Universities, HUST: Grant no. 2015TS061.

References

- [1] Li X, Wu C, Li W. Study on the Progress of Welding Science and Technology in China. *Chinese Journal of Mechanical Engineering*. 2012;48:19-31.
- [2] Jiang SL, Liu M, Hao JH, Qian WP. A bi-layer optimization approach for a hybrid flow shop scheduling problem involving controllable processing times in the steelmaking industry. *Computers & Industrial Engineering*. 2015;87:518-31.
- [3] Mao K, Pan Q-K, Chai T, Luh PB. An Effective Subgradient Method for Scheduling a Steelmaking-Continuous Casting Process. *IEEE Transactions on Automation Science and Engineering*. 2015;12:1140-52.
- [4] Hao J, Liu M, Jiang S, Wu C. A soft-decision based two-layered scheduling approach for uncertain steelmaking-continuous casting process. *Eur J Oper Res*. 2015;244:966-79.

- [5] Tang L, Wang G, Chen Z-L. Integrated Charge Batching and Casting Width Selection at Baosteel. *Operations Research*. 2014;62:772-87.
- [6] Mao K, Pan Q-k, Pang X, Chai T. A novel Lagrangian relaxation approach for a hybrid flowshop scheduling problem in the steelmaking-continuous casting process. *Eur J Oper Res*. 2014;236:51-60.
- [7] Li J-q, Pan Q-k, Mao K, Suganthan PN. Solving the steelmaking casting problem using an effective fruit fly optimisation algorithm. *Knowledge-Based Syst*. 2014;72:28-36.
- [8] Pan Q-K, Wang L, Mao K, Zhao J-H, Zhang M. An Effective Artificial Bee Colony Algorithm for a Real-World Hybrid Flowshop Problem in Steelmaking Process. *IEEE Transactions on Automation Science and Engineering*. 2013;10:307-22.
- [9] Tang L, Zhao Y, Liu J. An Improved Differential Evolution Algorithm for Practical Dynamic Scheduling in Steelmaking-Continuous Casting Production. *IEEE Trans Evol Comput*. 2014;18:209-25.
- [10] Garey MR, Johnson DS, Sethi R. The complexity of flowshop and jobshop scheduling. *Mathematics of operations research*. 1976;1:117-29.
- [11] Ciavotta M, Minella G, Ruiz R. Multi-objective sequence dependent setup times permutation flowshop: A new algorithm and a comprehensive study. *Eur J Oper Res*. 2013;227:301-13.
- [12] Durillo JJ, Nebro AJ. jMetal: A Java framework for multi-objective optimization. *Adv Eng Softw*. 2011;42:760-71.
- [13] Han YY, Gong DW, Sun XY, Pan QK. An improved NSGA-II algorithm for multi-objective lot-streaming flow shop scheduling problem. *Int J Prod Res*. 2014;52:2211-31.
- [14] Wang XJ, Gao L, Zhang CY, Shao XY. A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem. *Int J Adv Manuf Tech*. 2010;51:757-67.
- [15] Amin-Tahmasbi H, Tavakkoli-Moghaddam R. Solving a bi-objective flowshop scheduling problem by a Multi-objective Immune System and comparing with SPEA2+ and SPGA. *Adv Eng Softw*. 2011;42:772-9.
- [16] Zheng H-y, Wang L. Reduction of carbon emissions and project makespan by a Pareto-based estimation of distribution algorithm. *Int J Prod Econ*. 2015;164:421-32.
- [17] Rifai AP, Nguyen H-T, Dawal SZM. Multi-objective adaptive large neighborhood search for distributed reentrant permutation flow shop scheduling. *Appl Soft Comput*. 2016;40:42-57.
- [18] Yenisey MM, Yagmahan B. Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends. *Omega*. 2014;45:119-35.
- [19] Li MD, Zhao H, Weng XW, Han T. A novel nature-inspired algorithm for optimization: Virus colony search. *Adv Eng Softw*. 2016;92:65-88.
- [20] Mirjalili S, Lewis A. The Whale Optimization Algorithm. *Adv Eng Softw*. 2016;95:51-67.
- [21] Mirjalili S. The Ant Lion Optimizer. *Adv Eng Softw*. 2015;83:80-98.
- [22] Mirjalili S, Mirjalili SM, Lewis A. Grey Wolf Optimizer. *Adv Eng Softw*. 2014;69:46-61.
- [23] Civicioglu P. Backtracking Search Optimization Algorithm for numerical optimization problems. *Appl Math Comput*. 2013;219:8121-44.
- [24] Song XH, Tang L, Zhao ST, Zhang XQ, Li L, Huang JQ, et al. Grey Wolf Optimizer for parameter estimation in surface waves. *Soil Dynamics and Earthquake Engineering*. 2015;75:147-57.
- [25] Sharma Y, Saikia LC. Automatic generation control of a multi-area ST - Thermal power system using Grey Wolf Optimizer algorithm based classical controllers. *Int J Electr Power Energy Syst*. 2015;73:853-62.
- [26] El-Fergany AA, Hasanien HM. Single and Multi-objective Optimal Power Flow Using Grey Wolf Optimizer and Differential Evolution Algorithms. *Electric Power Components and Systems*.

2015;43:1548-59.

- [27] Medjahed SA, Saadi TA, Benyettou A, Ouali M. Gray Wolf Optimizer for hyperspectral band selection. *Appl Soft Comput*. 2016;40:178-86.
- [28] Mirjalili SM, Mirjalili SZ. Full Optimizer for Designing Photonic Crystal Waveguides: IMoMIR Framework. *IEEE Photonics Technology Letters*. 2015;27:1776-9.
- [29] Komaki G, Kayvanfar V. Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time. *J Comput Sci-Neth*. 2015;8:109-20.
- [30] Mirjalili S, Saremi S, Mirjalili SM, Coelho LdS. Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. *Expert Syst Appl*. 2016;47:106-19.
- [31] Ahmadizar F, Shahmaleki P. Group-shop scheduling with sequence-dependent set-up and transportation times. *Applied Mathematical Modelling*. 2014;38:5080-91.
- [32] Nawaz M, Ensore EE, Ham I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*. 1983;11:91-5.
- [33] Xie Z, Zhang C, Shao X, Lin W, Zhu H. An effective hybrid teaching–learning-based optimization algorithm for permutation flow shop scheduling problem. *Adv Eng Softw*. 2014;77:35-47.
- [34] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*. 2002;6:182-97.
- [35] Zitzler E, Laumanns M, Thiele L, Zitzler E, Zitzler E, Thiele L, et al. SPEA2: Improving the strength Pareto evolutionary algorithm. Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK); 2001.
- [36] Ting C-K, Su C-H, Lee C-N. Multi-parent extension of partially mapped crossover for combinatorial optimization problems. *Expert Syst Appl*. 2010;37:1879-86.
- [37] Han YY, Gong DW, Sun XY. A discrete artificial bee colony algorithm incorporating differential evolution for the flow-shop scheduling problem with blocking. *Eng Optimiz*. 2015;47:927-46.
- [38] Pan QK, Wang L, Gao L, Li WD. An effective hybrid discrete differential evolution algorithm for the flow shop scheduling with intermediate buffers. *Inform Sciences*. 2011;181:668-85.
- [39] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the Strength Pareto approach. *IEEE Trans Evol Comput*. 1999;3:257-71.
- [40] Derrac J, García S, Molina D, Herrera F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*. 2011;1:3-18.

List of Figures:

Fig. 1 The process flow chart for a welding plant

Fig. 2 Overlapping of setup and transportation times when (a) the start time is determined by setup

times and (b) the start time is determined by transportation times [16]

Fig. 3 Outline of the proposed MODGWO

Fig. 4 An example of solution representation

Fig. 5 An example of search for the prey

Fig. 6 Reduction of machine load procedure

Fig. 7 Gantt charts for strategy 1

Fig. 8 Gantt charts for strategy 2

Fig. 9 Non-dominated solutions with the best IGD run by different MOEAs

Fig. 10 Pareto front obtained by the proposed MODGWO

Fig. 11 The Gantt chart of the point A obtained by the proposed MODGWO

Fig. 12 The Gantt chart of the point B obtained by the proposed MODGWO