Engineering Applications of Artificial Intelligence 57 (2017) 61–79

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry

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

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

ARTICLE INFO

Keywords:
Welding scheduling
Grey wolf optimizer
Dynamic scheduling
Multi-objective optimization
Controllable processing times
Transportation times

ABSTRACT

Welding is one of the most important technologies in manufacturing industries due to its extensive applications. Welding scheduling can affect the efficiency of the welding process greatly. Thus, welding scheduling problem is important in welding production. This paper studies a challenging problem of dynamic scheduling in a real-world welding industry. To satisfy needs of dynamic production, three types of dynamic events, namely, machine breakdown, job with poor quality and job release delay, are considered. Furthermore, controllable processing times (CPT), sequence-dependent setup times (SDST) and job-dependent transportation times (JDTT) are also considered. Firstly, we formulate a model for the multi-objective dynamic welding scheduling problem (MODWSP). The objectives are to minimize the makespan, machine load and instability simultaneously. Secondly, we develop a hybrid multi-objective grey wolf optimizer (HMOGWO) to solve this MODWSP. In the HMOGWO, a modified social hierarchy is designed to improve its exploitation and exploration abilities. To further enhance the exploration, genetic operator is embedded into the HMOGWO. Since one characteristic of this problem is that multiple machines can handle one operation at a time, the solution is encoded as a two-part representation including a permutation vector and a machine assignment matrix. To evaluate the effectiveness of the proposed HMOGWO, we compare it with other well-known multi-objective metaheuristics including NSGA-II, SPEA2, and multi-objective grey wolf optimizer. Experimental studies demonstrate that the proposed HMOGWO outperforms other algorithms in terms of convergence, spread and coverage. In addition, the case study shows that this method can solve the real-world welding scheduling problem well.

1. Introduction

Welding is one of the most crucial technologies of materials forming and processing in modern manufacturing enterprises, which has been extensively applied in various industries such as automobile manufacturing, petroleum chemical industry and shipbuilding (Lu et al., 2016). Welding process has occupied a significant proportion of manufacturing process (Mendes et al., 2016). Its efficiency impacts the whole production efficiency greatly. Therefore, a reasonable welding schedule scenario can help to improve production efficiency. Consequently, welding scheduling problem plays a major role in promoting an enterprise to be a manufacturing giant. However, there is little research on welding scheduling problem. Unlike general scheduling problems, the special characteristic of the welding scheduling problem lies in the number of machines to process each job. In traditional scheduling problems, each job is usually processed on at most one machine at a time. However, in a realistic welding situation, it is common to observe that multiple welding machines can process one job at the same time. Thus, the machine assignment on each operation should be considered in the welding scheduling problem, which significantly increases the difficulty of problem solving. In addition, welding scheduling problem is also NP-hard since the classical scheduling problem such as flowshop is already NP-hard (Garey et al., 1976). Most research on the scheduling field is focused on static scheduling problems in which all the information is assumed to be unchanged during the period (Lu et al., 2016). Nevertheless, the practical welding process is usually affected by unexpected real-time events that can lead to deterioration of the original schedule. Thus, study on dynamic scheduling problem can better reflect the requirements of the actual production environment. Real-time dynamic events in the scheduling field may be categorized into two groups (Cowling and Johansson, 2002; Stoop and Wiers, 1996; Suresh and Chaudhuri, 1993).

(a) Resource-related: machine breakdown, tool unavailability, delay in the arrival or shortage of materials, etc.
(b) Job-related: rush job, job cancellation, changes in job processing

* Corresponding author.
E-mail addresses: luchao2006@163.com (C. Lu), gaoliang@mail.hust.edu.cn (L. Gao), lixinyu@mail.hust.edu.cn (X. Li), xiaoshq@hust.edu.cn (S. Xiao).

http://dx.doi.org/10.1016/j.engappai.2016.10.013
Received 10 March 2016; Received in revised form 22 October 2016; Accepted 24 October 2016
Available online 31 October 2016
0952-1976/

time, poor quality of job, etc.

Although preventive maintenance can reduce the disruption probability, it cannot eradicate uncertainties from the system (Katragjini et al., 2013). The current technologies for addressing the dynamic scheduling problem include completely reactive, pro-active and predictive-reactive scheduling (Ouelhadj and Petrovic, 2009). Among them, the predictive-reactive scheduling is the most commonly used rescheduling method in which an original schedule is adjusted in response to the dynamic environment. However, it may generate a totally different schedule from the previous one, which causes instability of the production system (Rangsaritratsamee et al., 2004). Consequently, dynamic scheduling problem should consider not only production efficiency (e.g., makespan) but also instability. Therefore, this issue is a multi-objective dynamic welding scheduling problem (MODWSP) and is very great of significance in real-world production systems.

There is no research addressing a realistic MODWSP in comparison with the majority of studies on static scheduling problems in other manufacturing processes. However, dynamic scheduling problems have attracted increasing attention due to its strong industrial background (Nie et al., 2013). Recent reviews of dynamic scheduling are provided by Ouelhadj and Petrovic (2009) and Priore et al. (2014). To solve these dynamic scheduling problems, some dynamic scheduling methods are developed. Yaochu and Branke (2005) provided a survey of evolutionary algorithms (EAs) in dynamic environments. Recently, Katragjini et al. (2013) considered a dynamic flowshop scheduling problem with the objective to minimize makespan and instability by using a weighted sum method. Li et al. (2015) proposed a single-objective teaching-learning-based metaheuristic to solve the flowshop under five uncertainties simultaneously. Tang et al. (2014) presented an improved differential evolutionary to solve the real-world dynamic scheduling in steelmaking-continuous casting production. However, the above methods for dealing with multi-objective scheduling problems are to combine them into a single-objective function by using a weighted sum approach. In most real-world scheduling problems, nevertheless, objectives are evaluated in different scales and thus it is difficult to determine weight values (Ciavotta et al., 2013). Therefore, it is better to handle multiple objectives with knowledge about Pareto dominance. Pareto-based multi-objective evolutionary algorithm (MOEA) is very suitable for solving multi-objective optimization problem since it can yield the non-dominated solutions in a single run (Gen and Lin, 2014; Shen and Yao, 2015). As mentioned above, the studied problem is a new scheduling issue, since it breaks a traditional concept that each job can be processed on at most one machine at a time in the conventional scheduling problem, whereas each job can be handled by multiple machines at a time in the MODWSP. The actual processing time of each operation is associated with number of the machine to process the corresponding operation. This characteristic makes the MODWSP unique comparing to other multi-objective dynamic scheduling problems. A new encoding scheme and search operators have to be designed to fit the characteristic of this new scheduling problem. Furthermore, most of original metaheuristics are developed to solve continuous problems and they cannot be directly applied to this scheduling problem. Therefore, it is imperative to study the welding scheduling problem in terms of both theory and application.

So far, various EAs have been adopted to solve a variety of optimization problems (El Sehiemy et al., 2013; Gao et al., 2015; Garg et al., 2014; Kazakov and Lempert, 2015; Lin et al., 2015; Loubière et al., 2016; Manjarres et al., 2013; Niu et al., 2013; Panda and Abraham, 2015; Precup et al., 2013; Zeng and Dong, 2016). Grey wolf optimizer (GWO) (Mirjalili et al., 2014) is a new swarm intelligence algorithm inspired from mimicking the leadership hierarchy and hunting mechanism of grey wolves. GWO is proven to be competitive or superior to other classical metaheuristics such as differential evolutionary and particle swarm optimization algorithm (Mirjalili et al., 2014). Furthermore, GWO is a very efficient metaheuristic algorithm due to its high convergence speed and simple mathematical model. The reason for the high convergence of GWO lies in the search mechanism. More precisely, the wolf population can be categorized into four groups (i.e., alpha, beta, delta and omega) based on their fitness. The search process in GWO is guided by the best three wolves at each generation. This search mechanism promotes the exploitation because all candidate wolves (solutions) are attracted toward the best three wolves, thereby converging faster to these good wolves. For this reason, the convergence performance of GWO can be improved. Additionally, GWO has been successfully applied to some application fields. For example, Saremi et al. (2015) proposed the use of evolutionary population dynamics in GWO. Noshadi et al. (2015) adopted GWO to solve PID-type fuzzy logic controller for a multi-input multi-output active magnetic bearing system. Precup et al. (2016a) applied GWO to the tuning of PI-Fuzzy controllers with a reduced process parametric sensitivity. Jayakumar et al. (2016) used the GWO to solve combined beat and power economic dispatch. Precup et al. (2016b) applied GWO to fuzzy control systems with reduced parametric sensitivity. Emary et al. (2016) developed a novel binary GWO to select the optimal feature subset for classification purposes. Based on the effectiveness of the GWO and nature of the multi-objective optimization problem (MOP), a new hybrid multi-objective grey wolf optimizer (HMOGWO) with genetic algorithm (GA) is proposed to solve this MODWSP. Although multi-objective grey wolf optimizer has been proposed (Mirjalili et al., 2016), it is used to solve continuous MOPs. The main reasons for developing HMOGWO for MODWSP are that the problem under this study is NP-hard, and GA has been demonstrated to be an effective approach for tackling the combinatorial optimization problem (Jalilvand-Nejad and Fattahi, 2015; Lin et al., 2013; Mou et al., 2014; Ruiz et al., 2006; Zhang et al., 2011). Furthermore, many MOEAs have been proposed to deal with practical engineering optimization problems. Deb et al. (2002) presented the classical NSGA-II, which adopts the fast non-dominated sorting and crowding distance strategies. Zitzler et al. (2001a) proposed the SPEA, which uses an external archive to manage non-dominated solutions and adopts a clustering scheme to maintain the size of the archive. However, no free lunch principle implies that one algorithm cannot obtain the better result than all the others on all the problems. In general, an appropriate combination of different search schemes may improve the performance of the algorithm, and even conquer the limitation of the single metaheuristic (Li et al., 2010; Ma et al., 2014; Tang and Wang, 2013). As mentioned earlier, the candidate wolves are attracted toward the potential optimal solutions by the best three solutions according to the fitness. But as a result of such an exploitative effect, the population of GWO is prone to stagnation in a local optimum. However, GA has a good exploration ability, and hybridizing of GWO and GA can balance exploration and exploitation of the proposed algorithm in this paper. The above reasons motivate us to design a hybrid multi-objective GWO with GA for this real-world dynamic scheduling problem.

To the best of authors' knowledge, a real-world MODWSP has been not yet reported in the previous research. The purpose of this paper is to develop a high-performance multi-objective predictive-reactive scheduling method for this MODWSP in order to narrow the gap between theoretical research and applicable practice. In this paper, we have accomplished the following five aspects: (1) a mathematical model for MODWSP is constructed. This model considers three objectives consisting of makespan, machine load and instability simultaneously. (2) A hybrid multi-objective optimization algorithm based on GWO and GA is developed to address this MODWSP. (3) An effective encoding scheme is presented according to characteristic of the considered problem. (4) The effectiveness of each improvement strategy of HMOGWO is validated by a considerable of experiments, and (5) the HMOGWO has been successfully applied to a real-world dynamic

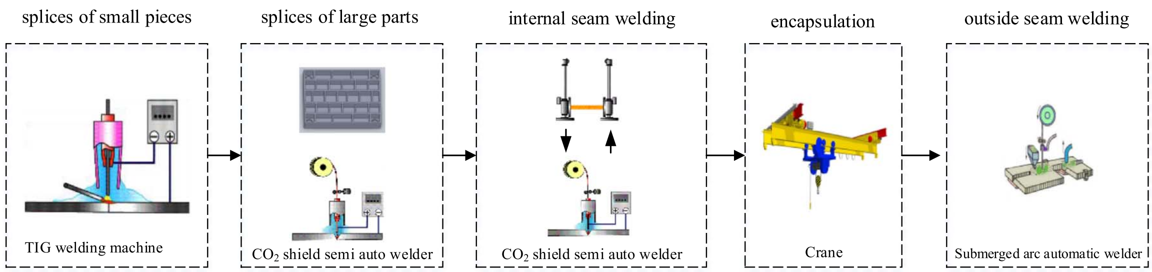

Fig. 1. The process flow chart for a welding plant.

welding scheduling problem from a company in China.

The remainder of this paper is organized as follows. The problem description and mathematical model for the MODWSP are given in Section 2. Section 3 describes a new multi-objective optimization approach. Section 4 presents experimental results and a case study, and Section 5 provides the conclusions and future work.

2. Problem description and formulation

In this section, the definition of a welding scheduling problem is given at first, and then the problem modelling is established.

2.1. Problem description

This paper investigates a MODWSP from a real-world welding workshop in China. The welding scheduling process, as presented in Fig. 1, includes five stages: (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 carried to the next stage. (2) Splices of large parts, in which several parts are melted into a complete part by the CO_2 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 using the CO_2 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.

In brief, the MODWSP can be stated as follows. A set of n jobs should be processed through five stages in the same process. Each job has at most one operation at each stage, and all the jobs are processed at each stage in the same order. Preemption is not allowed. As previously mentioned, the unique characteristic of MODWSP lies in the number of machines to process the operation. That is, multiple machines can process the operation of one job at a time. Each operation has a normal processing time when the one and only one machine is required to process the operation. 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. Moreover, to adapt to dynamic environments, job with release times delay, machine breakdown and job with poor quality are considered simultaneously in this work. Thus, this MODWSP is an extension of the multi-objective dynamic permutation flow shop with controllable processing times, which is more difficult than traditional dynamic permutation flow shop scheduling problem.

The characteristics of this real-world welding process can be summarized below.

- All the jobs go through the five stages in the welding shop floor: splices of small pieces, splices of large parts, internal seam welding, encapsulation and outside seam welding.
- One and only one machine can process the operation at a time at encapsulation stage. One or multiple machines can process one operation at a time at the other welding stages. That is, the processing duration of each operation can be controlled (i.e., compressed) by assigning additional welding machines at these stages. However, using multiple machines would bring machine load.
- Setup times are taken into account in this welding workshop due to 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 are considered, and these release times may be randomly changed according to the real-world environment.
- The poor quality of welding job occasionally occurs in welding process. The job with poor quality should be repaired when this uncertain event takes place. Machine breakdown also happens by chance during manufacturing process. Because the welding stages have alternative machines when machine breakdown, machine breakdown is only occurred at encapsulation stage in this study.
- 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).

2.2. Mathematical modelling of MODWSP

To formulate a mathematical model for MODWSP, the notations and decision variables are given below.

1) **Notations**

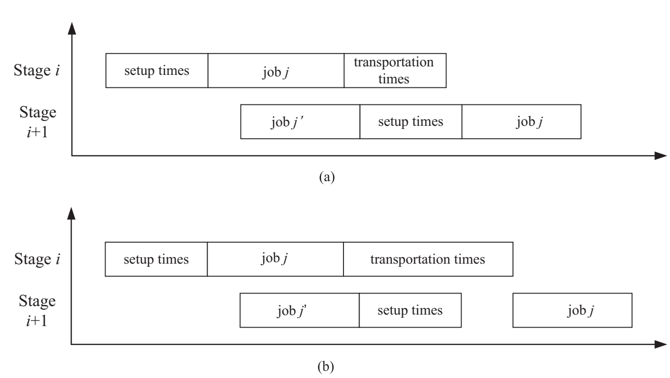

Fig. 2. Overlapping of setup and transportation times when (a) the start time is determined by setup times and (b) the start time is determined by transportation times (Ahmadizar and Shahmaleki, 2014).

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.
r_j: the release time for job j.
O_{ij}: the operation of job j at the stage i.
N_{ij}: the number of the machine 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 $N_{ij}=1$ when dynamic events do not happen.
rt_{ij}: the repaired time of the operation O_{ij} when the O_{ij} exists quality problem.
p^a_{ij}: the actual processing time of the operation O_{ij}. It is inversely proportional to the number of the corresponding machine when dynamic events do not happen, namely, $p^a_{ij}=\frac{p_{ij}}{N_{ij}}(N_{ij}>0)$. $p^a_{ij}=\frac{p_{ij}}{N_{ij}}+rt_{ij}$ when the O_{ij} is required to repair due to the quality loss of the 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.
B^s_i: the start time point of machine breakdown at stage i.
B^e_i: the end time point of machine breakdown at stage i.
u_i: the coefficient of penalty for using machine at stage i.
L: a very large positive number.
C_{ij}: the completion time of the operation O_{ij} in an original schedule.
S_{ij}: the start time of the operation O_{ij} in an original schedule. The S_{ij} should meet the following conditions.

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

2) **Decision variables:**

p^a_{ij}: the actual processing time of the operation O_{ij}.
$S'_{i,j}$: the start time of the operation O_{ij} in a new schedule.
$C'_{i,j}$: the completion time of the operation O_{ij} in a new schedule.

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

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

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

$$h_{ij1}=\begin{cases}1, & \text{if } C'_{i,j}\leq B^s_i\\ 0, & \text{otherwise}\end{cases}$$

$$h_{ij2}=\begin{cases}1, & \text{if } S'_{i,j}\leq B^s_i < C'_{i,j}\\ 0, & \text{otherwise}\end{cases}$$

$$h_{ij3}=\begin{cases}1, & \text{if } B^s_i < S'_{i,j}\\ 0, & \text{otherwise}\end{cases}$$

With the above notations, the MODWSP can be defined as a multi-objective mathematical model for minimizing the makespan, the total penalty of machine load, and instability simultaneously. This model is formulated as follows.

$$f_1=\min C_{max} \tag{1}$$

$$f_2=\min \sum_{i=1}^{m}\sum_{j=1}^{n} u_i N_{ij} \tag{2}$$

$$f_3=\min \sum_{i=1}^{m}\sum_{j=1}^{n} |S_{i,j}-S'_{i,j}| \tag{3}$$

where f_1 denotes the makespan, f_2 represents the total sum of penalty for machine load, f_3 is the instability which evaluates the deviation between the new and previous schedule. It can be calculated as the sum of operations whose starting times are anticipated or delayed in a new schedule. The constraints in this problem are given below.

$$\sum_{\mathcal{M}_i} z_{ijk}=N_{ij}\geq 1\quad i\in\{1,2,\ldots,5\};j\in\mathcal{J}\ \ k\in\mathcal{M} \tag{4}$$

$$\sum_{\mathcal{M}_4} z_{4jk}=N_{4j}=1\quad j\in\mathcal{J} k\in\mathcal{M} \tag{5}$$

$$N_{ij}\leq TM_i\quad i\in\{1,2,\ldots,5\};j\in\mathcal{J} \tag{6}$$

$$S'_{ij}\geq r_j+t_{j0i}\quad i\in\{1,2,\ldots,5\};j\in\mathcal{J} \tag{7}$$

$$S'_{ij}\geq sT_{i0j}\quad i\in\{1,2,\ldots,5\};j\in\mathcal{J} \tag{8}$$

$$L(1-x_{i'ij})+S'_{ij}\geq S'_{i'j}+p^a_{i'j}+t_{ji'i}\, i,\ i'\in\{1,2,\ldots,5\};j\in\mathcal{J} \tag{9}$$

$$Lx_{i'ij}+S'_{i'j}\geq S'_{ij}+p^a_{ij}+t_{jii'}\, i,\ i'\in\{1,2,\ldots,5\};j\in\mathcal{J} \tag{10}$$

$$L(1-y_{ij'j})+S'_{ij}\geq S'_{ij'}+p^a_{ij'}+sT_{ij'j}\, i\in\{1,2,\ldots,5\};j,j'\in\mathcal{J} \tag{11}$$

$$Ly_{ij'j}+S'_{ij'}\geq S'_{ij}+p^a_{ij}+sT_{ijj'}\, i\in\{1,2,\ldots,5\};j,j'\in\mathcal{J} \tag{12}$$

$$C'_{ij}-S'_{ij}-p^a_{ij}+L(1-h_{ij1})\geq 0\, i=5;j\in\mathcal{J} \tag{13}$$

$$C'_{ij}-S'_{ij}-p^a_{ij}-L(1-h_{ij1})\leq 0\, i=5;j\in\mathcal{J} \tag{14}$$

$$C'_{ij}-S'_{ij}-p^a_{ij}-B^e_i+B^s_i+L(1-h_{ij2})\geq 0\, i=5;j\in\mathcal{J} \tag{15}$$

$$C'_{ij}-S'_{ij}-p^a_{ij}-B^e_i+B^s_i-L(1-h_{ij2})\leq 0\, i=5;j\in\mathcal{J} \tag{16}$$

$$C'_{ij}-max\{S'_{ij},\ B^e_i\}-p^a_{ij}+L(1-h_{ij3})\geq 0\, i=5;j\in\mathcal{J} \tag{17}$$

$$C'_{ij}-max\{S'_{ij},\ B^e_i\}-p^a_{ij}-L(1-h_{ij3})\leq 0\, i=5;j\in\mathcal{J} \tag{18}$$

$$C_{max}\geq S'_{ij}+p^a_{ij}+t_{jiw}\, i\in\{1,2,\ldots,5\};j\in\mathcal{J} \tag{19}$$

$$x_{ji'i}=\{0,1\},\ y_{ij'j}=\{0,1\},\ h_{ij1},\ h_{ij2},\ h_{ij3}=\{0,1\},\ N_{ij}\in N^+,\ i\in\{1,2,\ldots,5\};j\in\mathcal{J} \tag{20}$$

Constraints (4) ensure that a job can be processed on at least one machine at each stage. Constraints (5) restrict that one and only one machine can process the jobs at encapsulation stage (i.e., stage 4).

Constraints (6) ensure that the number of available machines to process each operation is within its range. Constraints (7) guarantee that the job cannot be started before it is released and then is transported from the warehouse to the stage conducting its first operation. Constraints (8) impose that the first operation of the job at each stage cannot start until its setup time is completed. Constraints (9)–(10) guarantee that each job cannot be started until its preceding operations are completed and the job is transported to stage i. Constraints (11)–(12) ensure that each job cannot be started before its preceding jobs at the same stage are completed and its setup time is finished. Constraints (13)–(14) guarantee the first situation, which is for the operations that have finished before the machine breakdown. Constraints (15)–(16) ensure the second condition, is for the operations that overlap with the machine breakdown. Constraints (17)–(18) ensure the third condition, which is for the operations start after the machine breakdown. Constraints (19) specify that the makespan is equal to the time that the last job is completed and then transported to the warehouse. Constraints (20) restrict that decision variables x, y, and h are binary, and N_{ij} is a positive integer number.

Due to the NP-hard characteristic of the problem, large scale problems cannot be addressed within a reasonable computational time by using deterministic algorithms. The metaheuristic can provide an optimal or near-optimal solution with acceptable time consumption for scheduling problems. One of the most famous metaheuristics for MOP is NSGA-II which has been effective in solving combinatorial optimization. However, as mentioned previously, no free lunch theorem implies that different search methods may generate different results on a particular problem, but they are indistinguishable on all problems. Recently, a new trend of hybridization of different search operators has appeared. An appropriate combination of different techniques can improve the performance of algorithm (Tang and Wang, 2013). More precisely, hybrid MOEAs with other EAs can provide more powerful search ability than MOEAs with the single strategy, which inspires us to incorporate several simple but efficient operators to improve search diversity and robustness of algorithms. GWO is a simple yet effective metaheuristic due to its high convergence speed, few parameters and ease of implementation. Therefore, we should propose a more effective HMOGWO than NSGA-II and other high-performing MOEAs to solve such large scale problems.

3. The proposed HMOGWO for MODWSP

In this section, firstly we describe the original GWO (Mirjalili et al., 2014), and then state the basic background on multi-objective optimization. Finally the details of proposed HMOGWO for the MODWSP are elaborated.

3.1. The canonical GWO

GWO (Mirjalili et al., 2014) is a new metaheuristic inspired by the grey wolf hunting for the prey in nature. The main steps of GWO are provided in the following parts.

To establish a social hierarchy of wolves when developing GWO, all the grey wolves are classified into four kinds of the wolf according to the 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 mainly guided by α, β and δ. The ω wolves obey these three wolves.

Grey wolves tend to encircle the prey. To model encircling behavior, the equation is defined as follows (Mirjalili et al., 2014):

$$\boldsymbol{D}=\boldsymbol{C}\circ\boldsymbol{X}_p(t)-\boldsymbol{X}(t) \tag{21}$$

$$\boldsymbol{X}(t+1)=\boldsymbol{X}_p(t)-\boldsymbol{A}\circ\boldsymbol{D} \tag{22}$$

where t represents the current iteration, $\circ$ represents the hadamard product operation, $\boldsymbol{X}_p$ denotes the position vector of the prey and $\boldsymbol{X}$ denotes the position vector of a grey wolf. For $\boldsymbol{C}$, $\boldsymbol{X}_p$, $\boldsymbol{A}$ and $\boldsymbol{D}$, they have the same dimension. The vector $\boldsymbol{A}$ and $\boldsymbol{C}$ are formulated as follows:

$$\boldsymbol{A}=2\boldsymbol{a}\circ\boldsymbol{r}_1-\boldsymbol{a} \tag{23}$$

$$\boldsymbol{C}=2\boldsymbol{r}_2 \tag{24}$$

where $\boldsymbol{a}$ is linearly decreased from 2 to 0 during the search process, and it is used to emphasize exploration and exploitation respectively, $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are random vectors in range [0,1]. The hunt is often guided by the alpha. The beta and delta might also join in on the hunting. However, the position of the 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 (Mirjalili et al., 2014).

$$\boldsymbol{D}_\alpha=\boldsymbol{C}_1\circ\boldsymbol{X}_\alpha-\boldsymbol{X},\ \boldsymbol{D}_\beta=\boldsymbol{C}_2\circ\boldsymbol{X}_\beta-\boldsymbol{X},\ \boldsymbol{D}_\delta=\boldsymbol{C}_3\circ\boldsymbol{X}_\delta-\boldsymbol{X} \tag{25}$$

$$\boldsymbol{X}_1=\boldsymbol{X}_\alpha-\boldsymbol{A}_1\circ\boldsymbol{D}_\alpha,\ \boldsymbol{X}_2=\boldsymbol{X}_\beta-\boldsymbol{A}_2\circ\boldsymbol{D}_\beta,\ \boldsymbol{X}_3=\boldsymbol{X}_\delta-\boldsymbol{A}_3\circ\boldsymbol{D}_\delta \tag{26}$$

$$\boldsymbol{X}(t+1)=\frac{\boldsymbol{X}_1+\boldsymbol{X}_2+\boldsymbol{X}_3}{3} \tag{27}$$

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 $|\boldsymbol{A}|>1$ and converge to attack prey when $|\boldsymbol{A}|<1$. Finally the optimal solution (i.e., the prey) is output if the stop criterion is satisfied.

3.2. Description of multi-objective optimization

To better understand the proposed HMOGWO for solving the above problem, we begin with a brief introduction of basic concept of MOP. Without loss of generality, a MOP can be defined as follows:

$$\min f(\boldsymbol{X})=\min[f_1(\boldsymbol{X}),\ f_2(\boldsymbol{X}),\ldots f_m(\boldsymbol{X})] \tag{28}$$

$$\boldsymbol{X}=(x_1,\ x_2,\ldots x_n)\in R^n$$

$$s.\ t.\begin{cases} g_i(\boldsymbol{X})\geq 0, & i=1,\ldots,k \\ h_j(\boldsymbol{X})=0, & j=1,\ldots,p \end{cases}$$

where $f_m(\boldsymbol{X})$ indicates the m-th sub-objective function; $\boldsymbol{X}$ is a vector of the solution, which should satisfy the above constraints. R^n is the decision variable space.

Let $\boldsymbol{a}$ and $\boldsymbol{b}\in R^n$, a vector $\boldsymbol{a}=[a_1,\ a_2,\ldots,\ a_n]^T$ is said to dominate another vector $\boldsymbol{b}=[b_1,\ b_2,\ldots,\ b_n]^T$ (denoted by $\boldsymbol{a}\prec\boldsymbol{b}$) if and only if $f_i(\boldsymbol{a})\leq f_i(\boldsymbol{b})$ for each $i\in\{1,\ldots,m\}$ and $f_l(\boldsymbol{a})<f_l(\boldsymbol{b})$ for at least one index $l\in\{1,\ldots,m\}$. A solution $\boldsymbol{X}^*\in R^n$ is a Pareto optimal solution if there is not any a solution $\boldsymbol{X}\in R^n$ that dominates $\boldsymbol{X}^*$. The corresponding objective function is called the Pareto optimal front vector $f(\boldsymbol{X}^*)$. For a Pareto optimal solution, the improvement of any objective must cause the deterioration of at least another objective. The set of all Pareto optimal solutions is called Pareto set (PS*), while the set of all Pareto optimal front vectors is called the Pareto optimal front (PF*) (Deb et al., 2002). The main goal of multi-objective optimization is to find PF*. However, a Pareto front usually consists of a large number of points. Therefore, a good Pareto front contains a limited number of points which should be as close as possible to the PF* and uniformly spread as well.

3.3. Proposed HMOGWO

We propose a HMOGWO to tackle the above MODWSP. A basic flowchart of HMOGWO is shown in Fig. 3. These improvement procedures mainly include: encoding and decoding schemes, initialization, social hierarchy, and hybrid search strategy consisting of grey wolf search and genetic operator. The following subsections give a detailed

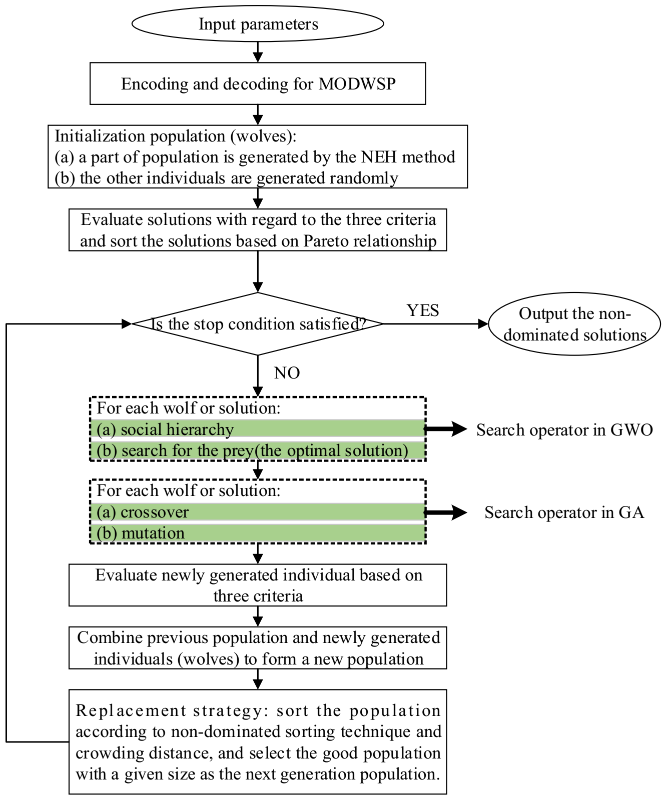

Fig. 3. A flowchart of the proposed HMOGWO.

explanation about the improvement steps of HMOGWO.

3.3.1. Encoding scheme

The encoding scheme is an important procedure before applying the proposed HMOGWO to a particular problem. Unlike the 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 two-part encoding scheme, namely, $\boldsymbol{X}=[\pi;N]$. The first part is the job permutation denoted by notation $\boldsymbol{X}(\pi)$. The second part is the machine assignment matrix denoted by symbol $\boldsymbol{X}(N)$. To illustrate this encoding scheme, we give a solution representation with 3 jobs and 2 stages as follows.

$\boldsymbol{X}(\pi)=[3,1,2]$

$$\boldsymbol{X}(N)=\begin{bmatrix}3,1,2\\1,1,1\end{bmatrix}$$

In this solution representation, the job sequence is job 3, job 1 and job 2. The row and column of $\boldsymbol{X}(N)$ represents the stage $i\in\{1,2,\ldots,m\}$ and job $j\in\{1,2,\ldots,n\}$, respectively. The element $\boldsymbol{X}(N)_{ij}$ in the machine assignment matrix denotes the number of the machine assigned to the operation O_{ij}. For example, $\boldsymbol{X}(N)_{13}$ represents that the number of machines that are assigned to job 3 at stage 1 (i.e., O_{13}) is 2.

3.3.2. Rescheduling decoding method

In this paper, three types of disruptions are considered simultaneously, i.e., machine breakdown, job repaired time, and job release delay. Therefore, dynamic event processing methods contain the following conditions.

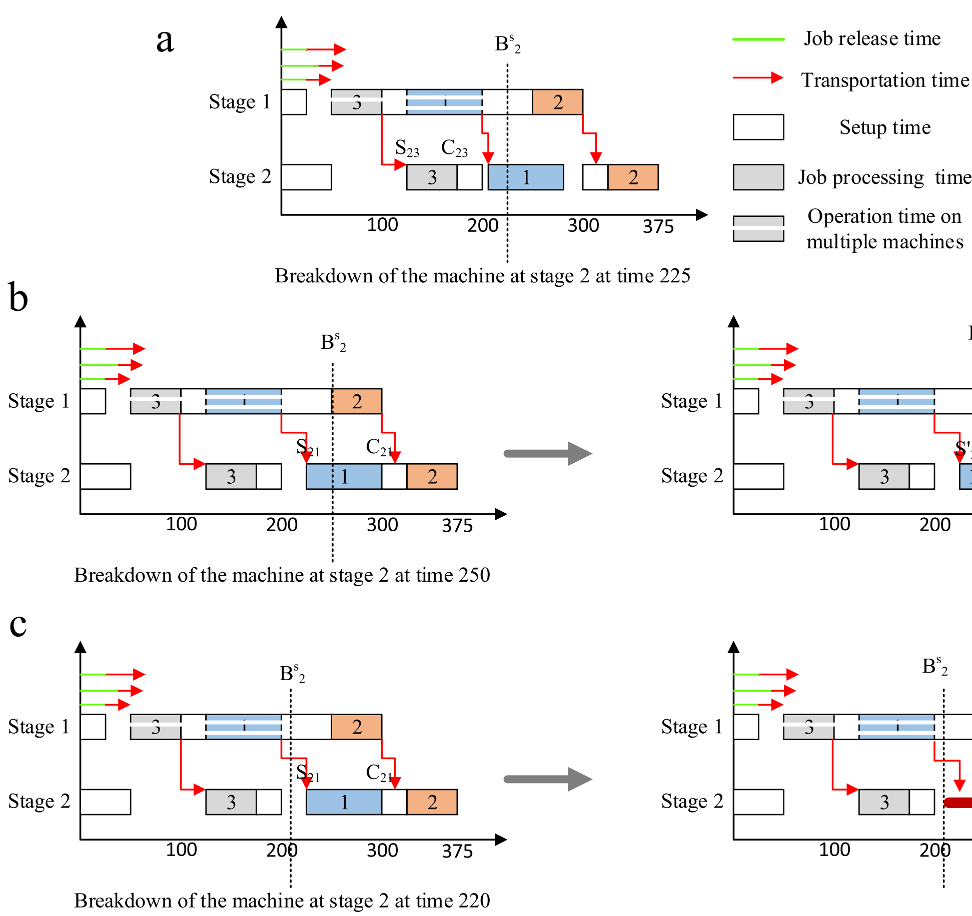

Fig. 4. Machine breakdown example (a) the first status (b) the second status and (c) the third status. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- Machine breakdowns: machine breakdowns happen only on busy machines in this study, e.g., machine at encapsulation stage. Machine breakdown may occur in one of three statuses based on its progress (Li et al., 2015): (a) the first status, in which the operation has been completed before the disruption. (b) The second status, in which the operation is overlapped with the disruption. (c) The third status, in which the operation starts to process after the disruption. Fig. 4 presents the decoding procedure for the above three statues. Assume that only one machine is available at stage 2 and thus machine breakdown only occurs at stage 2. Fig. 4(a) illustrates the first status, which is defined by constraints (13) and (14). In this status, the operation of job 3 at stage 2 (i.e., O_{23}) is completed before the machine breakdown, and thus it will keep the start and completion time unchanged. Fig. 4(b) presents the second status, which is defined by constraints (15) and (16). In this status, the machine breakdown time overlaps with the operation of job 1 at stage 2 (i.e., O_{21}). In this status, the operation O_{21} is interrupted at the machine breakdown time point. At the same time, the follow-up work is delayed until the recovery of the machine. Therefore, in the second status, the interrupted operation O_{21} will keep its start time as in the on-going schedule whilst delaying its completion time. Fig. 4(c) gives the third status, which is provided by constraints (17) and (18). In this status, operation O_{21} starts its task after the machine breakdown disruption. We should guarantee that the start time of the affected operation is not earlier than the recovery time of the breakdown machine.

- Job with poor quality: the processing time of the operation should be increased when the operation is required to repair due to the poor quality. The repaired operation will affect the follow-up operations. Fig. 5 gives an example of the job repaired, where job 2 at stage 1 postpones its processing time. Poor quality of the operation is a small probability event. This uncertain event usually only occurs with a probability of 0.05. The reschedule for this uncertain event can perform a right shift on the affected operation.

- Job release delays: In this status, the machine at the stage should wait for the job's arrival until the operation can start. Fig. 6 illustrates an example of the job release time variation, where the release time of job 3 is delayed at time t. it will affect the follow-up operations. The reschedule for this uncertain event can perform a right shift on the affected operation.

- Multiple interference between disruptions: the three types of disruptions can occur together, i.e., a machine breakdown, job with poor quality and job with release delay happen in scheduling system simultaneously.

In the above Gantt charts (Figs. 4–6), the color box represents the actual processing time, where the number of the white bar denotes the number of the additional machines assigned to the corresponding operation. The white box represents the setup time. The red arrow denotes the transportation time. The other information is given in the above Gantt charts.

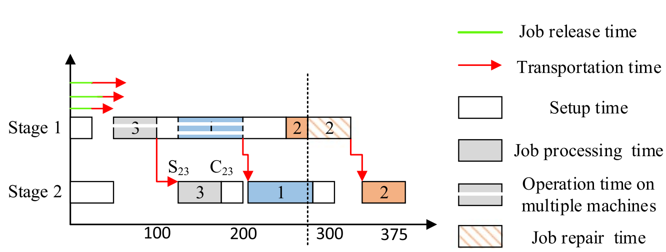

Fig. 5. Job with poor quality. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

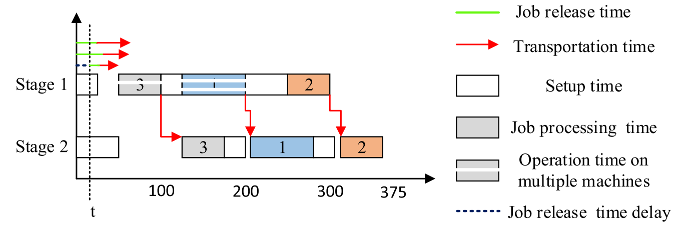

Fig. 6. Job release delay.(For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.3.3. Initialization

Based on the above encoding and decoding scheme, each solution 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 the NEH method (Nawaz et al., 1983). The other initial individuals are randomly generated based on the proposed encoding scheme.

3.3.4. 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 the MOP. That is, for a trade-off solution, the improvement of any 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 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.

3.3.5. Search for the prey

The search operator of the GWO is originally developed for solving continuous optimization problems (Mirjalili et al., 2014). Thus, it cannot be directly used to deal with combinatorial optimization 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:

$$X(\pi)_i^{t+1} = \begin{cases} shift\,(X(\pi)_i^t, C\bullet(X(\pi)_\alpha^t - X(\pi)_i^t)) & \text{if rand} < \frac{1}{3} \\ shift\,(X(\pi)_i^t, C\bullet(X(\pi)_\beta^t - X(\pi)_i^t)) & \text{else if rand} < \frac{2}{3} \\ shift\,(X(\pi)_i^t, C\bullet(X(\pi)_\delta^t - X(\pi)_i^t)) & \text{otherwise} \end{cases}$$

where $shift\,(x, d)$ 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] and C is the control factor. In this paper, C is set to 1.0. Fig. 7 gives an example of this search operator.

For the second part, namely, the machine assignment matrix:

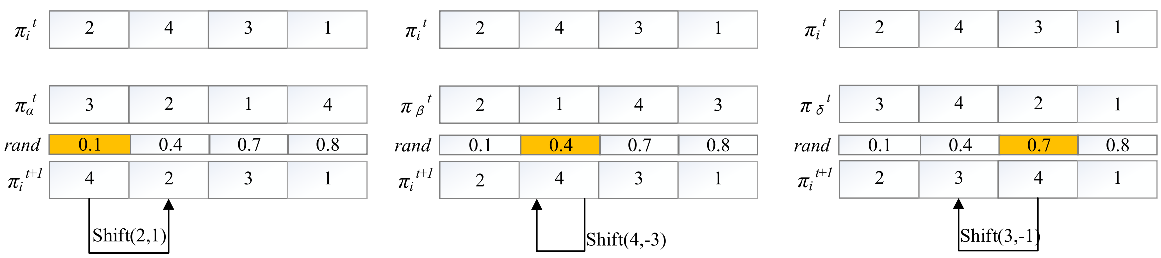

Fig. 7. An example of search for the prey.

Table 1
Data set distribution.

Input variables	Distribution
Number of jobs (n)	20, 40, 60, 80, 100
Number of stage (m)	4, 5, 6, 7, 8
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 [10, 30]
Transportation time (t)	discrete uniform [10, 20]
Coefficient of penalty for machine usage (u)	continuous uniform [1, 5]
Time to repair jobs with poor quality	discrete uniform [10, 25]
The probability for poor quality of each operation	0.05
Machine breakdown time	discrete uniform $[p_{11},\ n{\bullet}m{\bullet}p_{11}/2]$
Time to repair machine breakdown	discrete uniform [10,100]
Release time for each job	discrete uniform [5, 15]
Release delay for each job	discrete uniform [0, 10]

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

where the new machine assignment element $\boldsymbol{X}(N)_{i,j}^{t+1}$ 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, $\boldsymbol{X}(N)_{i,j}^{t+1}$ can be any feasible value not limited to the first rule with the probability of $(1-a)$.

3.3.6. Crossover and mutation operators

Crossover can explore unvisited areas of solution search space with a probability Pc. For the first part (i.e., π), partially mapping crossover (PMX) (Goldberg, 1989) is adopted to update the permutation part. For the second part, two-point crossover (John, 1992) is used to update machine assign matrix part in this paper.

Mutation operator assists the algorithm to escape from local optima, and occurs after crossover in order to randomly change elements of a solution. The well-known swap mutation (Pan et al., 2011), performed by randomly exchanging two elements of a solution, is applied to two parts with a probability P_m.

3.3.7. Replacement procedure

This phase of the HMOGWO is different from that of the original GWO when solving MOPs, the newly generated solutions are evaluated in terms of three fitness values considered (i.e., makespan, total penalty of machine load and instability). 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 (Deb et al., 2002). To achieve wide-spread Pareto fronts (solutions differently balancing makespan, total penalty of machine load and instability), 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 the ranking and crowding distance for the next generation.

4. Experimental studies and case study

This section is devoted to measuring the performance of the proposed algorithm HMOGWO for the addressed problems. In the following subsections, test problems, performance metrics and parameter setting are described at first, and then the experimental studies are further investigated step by step. All algorithms are coded in Java through the jMetal software (Durillo and Nebro, 2011). The experimental tests are implemented on a computer with Intel Core i5, 2.39 GHz, 4 GB RAM, with a Windows 8 operating system.

4.1. MODWSP instances

The instances randomly generated are defined in Table 1. The schedule disruption is usually simulated by generating random machine breakdowns at time t, $0 \le t \le C_{\max}(B)$, where $C_{\max}(B)$ denotes the makespan of baseline B and this baseline can be calculated according to the literature (Katragjini et al., 2013; Li et al., 2015; Ruiz and Stützle, 2007). But this MODWSP is different from traditional scheduling problems. For the sake of simplicity of computation, $t \in [p_{11},\ n{\bullet}m{\bullet}p_{11}/2]$ in this experiment, where p_{11} is the normal processing time of the operation O_{11}, n and m represent the number of jobs and stages. In addition, machine breakdowns only occur on the busy machine at certain stages, we assume that the machine breakdown disruption only happens at stage 3 in this experiment. Each instance can be labeled in the form of "n_m". For example, "10_5" represents that the problem is characterized as 10 jobs with 5 stages.

4.2. Performance metrics

To measure the PF obtained by these MOEAs, three metrics including Spread (Deb et al., 2002), GD (Veldhuizen and Lamont, 1998), and IGD (Zitzler and Thiele, 1999) should be employed below.

(1) Spread (Δ). The metric is a diversity indicator that measures the extent of spread achieved among the front found. The metric is defined as:

$$\Delta = \frac{\sum_{j=1}^{n_o} d_j^e + \sum_{i=1}^{|PF|} |d_i - \bar{d}|}{\sum_{j=1}^{n_o} d_j^e + |PF|{\bullet}\bar{d}} \tag{29}$$

where d_i denotes the Euclidean distance of each point in Pareto front (PF) to its closest point in its neighbor in 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, $|PF|$ is the number of points in PF, and n_o is the number of objectives. A lower spread value indicates a good distribution of solutions in PF.

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

$$\mathrm{GD} = \frac{\sqrt{\sum_{i=1}^{|PF|} d_i^2}}{|PF|} \tag{30}$$

where $|PF|$ is the number of points in the PF, and d_i is the Euclidean distance between each point of that front and the nearest member of the 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). This is a variant of the GD but represents a combined or comprehensive indicator. It measures both the diversity and convergence of PF in a sense. IGD can be defined as follows:

$$\mathrm{IGD} = \frac{\sum_{x \in PF^*} dist(x,\ PF)}{|PF^*|} \tag{31}$$

where $|PF^*|$ is the number of all elements in PF^*, and $dist(x,\ PF)$ is the minimum Euclidean distance between x and the nearest member in PF. 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 all MOEAs for each instance in all the independent runs are regarded as PF* (reference points) on that instance.

4.3. Parameters settings

It is well known that the parameter settings can affect the behavior of the algorithm (Katunin et al., 2014). Due to the space limitation, we conducted an experiment for one case (e.g., problem 40_4) with different parameter settings. In this experiment, only the comprehensive metric IGD is used to assess the behavior of different methods with different parameter combinations through using the graphical view. Here, the experiments focus on the main parameters such as crossover rate, mutation rate, population size, and the maximum number of function evaluations (NFEs). First, we test the crossover and mutation rate combinations, which affect the performance of methods. In this experiment, crossover rate is from the set [0.6, 0.7, 0.8, 0.9]; mutation rate is from the set [0.1, 0.2, 0.3, 0.4]; population size is set to 100; the maximum NFEs is set to 100,000. The statistical results of the IGD metric over 30 independent runs are presented in Fig. 8. The horizontal X-axis of each plot represents the set of mutation rate. The horizontal Y-axis of each plot represents the set of crossover rate. The vertical Z-axis of each plot denotes the mean IGD value over 30 independent runs. The lower IGD value is desirable. Note that we only study on HMOGWO, NSGA-II and SPEA2 except for MOGWO since MOGWO does not include crossover and mutation rate.

We also study the effect of population size on the behavior of each algorithm, where population size is updated from 50 to 200 with step 50, the maximum NFEs is set to 100,000, crossover rate is set to 0.9, and mutation rate is set to 0.2. The statistical results over 30 independent runs are shown as boxplots in Fig. 9, with the red circle denoting the average IGD value. The vertical axis of each plot represents the IGD and the horizontal axis indicates the variation of the population size.

We further investigate the effect of the maximum NFEs on the performance of each algorithm, where the maximum NFEs is updated from 20,000 to 100,000 with step 20,000, the population size is set to 100, crossover rate is set to 0.9, and mutation rate is set to 0.2. The statistical results of the IGD metric over 30 independent runs are presented as boxplots in Fig. 10, with the red circle denoting the average IGD value. The vertical axis of each plot denotes the IGD and the horizontal axis indicates the variation of the population size.

From the above figures and boxplots (Figs. 8–10), we can see that the parameter configuration with a larger crossover rate, a smaller mutation rate, a larger population and the maximum NFEs has a positive effect on the performance of the algorithm. To make a fair comparison, all MOEAs use the same maximum NFEs and the proposed encoding scheme. More precisely, the parameters are summarized in Table 2. These parameter settings are reasonable. Each experiment is conducted 30 independent times on each test problem for each algorithm.

Due to the stochastic characteristic of all MOEAs, a Wilcoxon sign rank test with the significance level of 0.05 is adopted to measure the significant difference between different MOEAs. The sign "+" indicates that the proposed HMOGWO algorithm performs significantly better than the second best algorithm. While "−" represents that the HMOGWO algorithm is significantly worse than the best algorithm. The sign "=" denotes that there is no significant difference between HMOGWO and the best or second best MOEA. The optimal results are highlighted with **bold** in the tables.

4.4. Efficiency of initialization strategy in HMOGWO

To evaluate the efficiency of the improved initialization strategy in HMOGWO, we investigate the effect of initialization strategy on all test functions. In this experiment, $\mathrm{HMOGWO_n}$ represents the HMOGWO without NEH strategy (i.e., random initialization), and HMOGWO includes NEH strategy. The other parameters of HMOGWO and $\mathrm{HMOGWO_n}$ are the same for a fair comparison. Table 3 shows mean

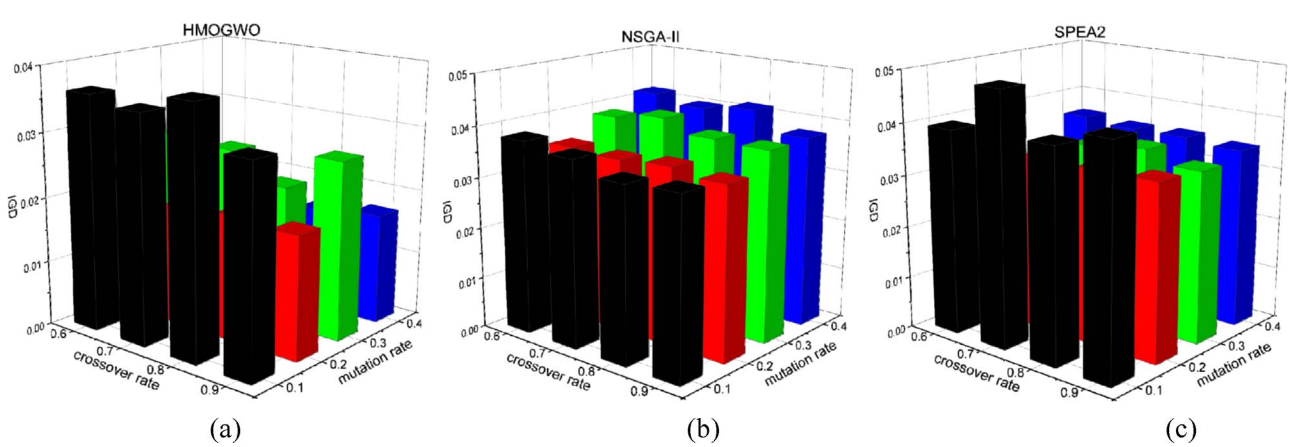

Fig. 8. Sensitivity study of crossover and mutation rate, (a) HMOGWO, (b) NSGA-II, (c) SPEA2.

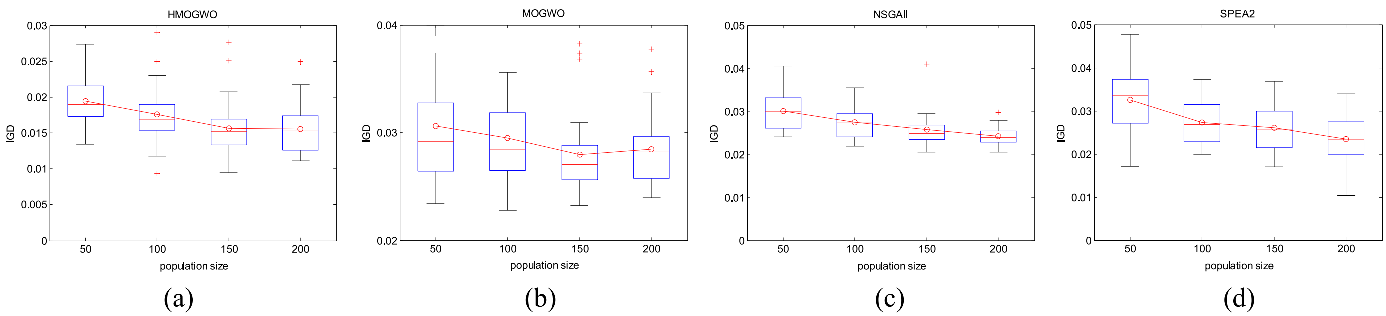

Fig. 9. Sensitivity study of population size, (a) HMOGWO, (b) MOGWO, (c) NSGA-II, (d) SPEA2. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

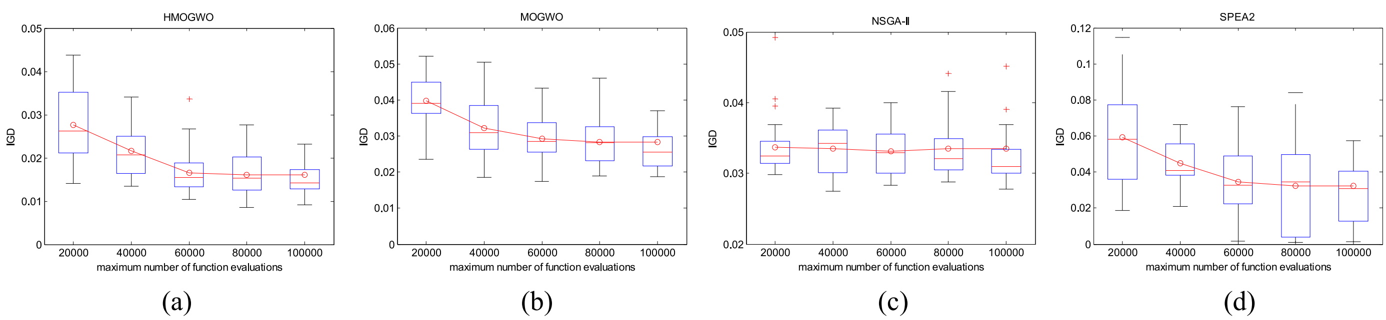

Fig. 10. Sensitivity study of the maximum NFEs, (a) HMOGWO, (b) MOGWO, (c) NSGA-II, (d) SPEA2. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

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

HMOGWO	MOGWO	NSGA-II	SPEA2
Population size: 80 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: 80 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: 80 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: 80 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 (near optimal solution)	Search for the prey (near optimal solution)	PMX and insert operation Crossover rate: 0.9 swap operation mutation rate: 0.2	PMX and insert operation Crossover rate: 0.9 swap operation mutation rate: 0.2
PMX and insert operation Crossover rate: 0.9 Swap operation Mutation rate: 0.2			

and standard deviation metrics on two algorithms over 30 runs.

Table 3 reveals that the HMOGWO performs better than the HMOGWO$_n$ for the GD metric since the HMOGWO achieves the optimal metric results on 15 out of 25 test instances. This means that the HMOGWO has a better convergence toward the Pareto optimal solution than the HMOGWO$_n$ on most instances. Concerning the Spread metric, the HMOGWO is competitive with the HMOGWO$_n$ because the HMOGWO can obtain the best values on 14 out of 25 test problems. This implies that results obtained by the HMOGWO have a more uniform distribution than those by the HMOGWO$_n$ on some problems. As mentioned above, IGD is a comprehensive performance metric. It should be pointed out that IGD is a priority indicator compared to the other metrics. Taking the problem 20_7 for example, the HMOGWO is better than the HMOGWO$_n$ for the Spread metric, and it is worse than the HMOGWO$_n$ for the GD metric. However, the HMOGWO outperforms the HMOGWO$_n$ in terms of the IGD metric. Thus, the HMOGWO outperforms the HMOGWO$_n$ on the overall performance for this problem. Furthermore, HMOGWO is significantly better than HMOGWO$_n$ on most instances in terms of the IGD metric. Therefore, it can be concluded that the HMOGWO using a NEH strategy can effectively improve the quality of solutions.

4.5. Efficiency of the improved social hierarchy strategy in HMOGWO

To test the effectiveness of the improved social hierarchy strategy, HMOGWO is compared to its variation in which three best solutions are selected using a binary tournament selection. In this experiment, this variation is denoted as HMOGWO$_{bts}$. Each experiment was conducted 30 independent times on each test problem for each algorithm. The statistic results of three metrics are summarized in Table 4.

We can clearly observe from Table 4 that the performance of the proposed algorithm can be greatly improved by the incorporation of improvement social hierarchy strategy. Concerning the comprehensive metric IGD, the performance of HMOGWO is overwhelming except for several instances like problem 20_7, 20_8, 40_6, 60_5 and 80_8. For problems 20_7, 20_8 and 60_5, the HMOGWO has a good convergence performance since it can obtain better GD metric values than its competitors on these problems, but the Spread metric obtained by the HMOGWO becomes worse. For problems 80_8, the HMOGWO can obtain a good Spread metric but it fails to provide a good GD metric. Such a phenomenon also happens for the problem 40_6. The HMOGWO gives a good Spread metric but it fails to generate a good

Table 3
Mean and standard deviation value of all metrics between HMOGWO$_n$ and HMOGWO.

Problem	GD (Mean/std)		Spread (Mean/std)		IGD (Mean/std)	
	HMOGWO$_n$	HMOGWO	HMOGWO$_n$	HMOGWO	HMOGWO$_n$	HMOGWO
20_4	**3.80e − 02/1.8e−02**	4.09e − 02/1.6e−02^{-}	**1.30e + 00/1.1e−01**	1.32e + 00/1.3e−01$^{=}$	2.78e − 02/7.8e−03	**2.77e − 02/6.2e−03$^{=}$**
20_5	**2.24e − 02/1.2e−02**	2.71e − 02/1.3e−02^{-}	9.14e − 01/9.9e−02	**8.64e − 01/1.2e−01^{+}**	2.26e − 02/6.7e−03	**1.96e − 02/5.7e−03^{+}**
20_6	1.37e − 02/3.1e−03	**1.33e − 02/3.4e−03$^{=}$**	6.20e − 01/4.4e−02	**6.13e − 01/4.8e−02^{+}**	9.33e − 03/1.2e−03	**6.98e − 03/1.0e−03^{+}**
20_7	**1.02e − 02/3.0e−03**	1.22e − 02/2.6e−03^{-}	7.84e − 01/8.3e−02	**7.44e − 01/6.0e−02^{+}**	1.18e − 02/1.1e−03	**1.01e − 02/1.5e−03^{+}**
20_8	2.36e − 02/1.2e−02	**2.26e − 02/9.3e−03^{+}**	8.41e − 01/8.4e−02	**8.11e − 01/8.1e−02^{+}**	1.71e − 02/8.4e−03	**1.36e − 02/6.9e−03^{+}**
40_4	**4.73e − 02/1.8e−02**	5.08e − 02/3.1e−02^{-}	1.08e + 00/1.7e−01	**1.05e + 00/2.1e−01$^{=}$**	3.30e − 02/5.2e−03	**1.97e − 02/4.8e−03^{+}**
40_5	4.61e − 02/1.6e−02	**3.53e − 02/1.2e−02^{+}**	**7.16e − 01/7.3e−02**	1.05e + 00/2.1e−01^{-}	4.11e − 02/9.4e−03	**1.96e − 02/5.2e−03^{+}**
40_6	3.16e − 02/9.1e−03	**1.23e − 02/9.0e−03^{+}**	**7.48e − 01/9.0e−02**	8.15e − 01/1.1e−01^{-}	2.86e − 02/6.5e−03	**1.36e − 02/2.8e−03^{+}**
40_7	**1.70e − 02/5.8e−03**	2.03e − 02/6.6e−03^{-}	6.54e − 01/4.5e−02	**6.27e − 01/5.7e−02^{+}**	1.75e − 02/2.6e−03	**1.37e − 02/2.6e−03^{+}**
40_8	2.25e − 02/5.7e−03	**9.32e − 03/4.7e−03^{+}**	7.05e − 01/5.8e−02	**6.73e − 01/6.8e−02^{+}**	1.88e − 02/4.1e−03	**7.76e − 03/1.9e−03^{+}**
60_4	4.74e − 02/6.6e−03	**2.08e − 02/1.4e−02^{+}**	**6.48e − 01/6.8e−02**	7.78e − 01/7.3e−02^{-}	3.27e − 02/5.0e−03	**1.09e − 02/2.9e−03^{+}**
60_5	4.42e − 02/1.1e−02	**2.38e − 02/1.2e−02^{+}**	**6.40e − 01/5.7e−02**	7.15e − 01/8.4e−02^{-}	3.89e − 02/7.7e−03	**1.65e − 02/6.4e−03^{+}**
60_6	3.69e − 02/8.2e−03	**1.59e − 02/8.1e−03^{+}**	**6.42e − 01/5.0e−02**	7.09e − 01/7.3e−02^{-}	3.65e − 02/6.9e−03	**1.49e − 02/4.5e−03^{+}**
60_7	**1.77e − 02/5.5e−03**	2.18e − 02/5.5e−03^{-}	6.99e − 01/5.7e−02	**6.70e − 01/5.3e−02$^{=}$**	1.82e − 02/1.9e−03	**1.09e − 02/2.4e−03^{+}**
60_8	**1.20e − 02/3.8e−03**	1.57e − 02/3.7e−03^{-}	7.19e − 01/7.2e−02	**6.43e − 01/5.4e−02^{+}**	1.51e − 02/1.2e−03	**1.01e − 02/1.9e−03^{+}**
80_4	4.50e − 02/7.1e−03	**2.42e − 02/1.4e−02^{+}**	**6.45e − 01/6.7e−02**	7.66e − 01/8.5e−02^{-}	5.76e − 02/6.7e−03	**1.65e − 02/4.1e−03^{+}**
80_5	**3.12e − 02/7.8e−03**	3.66e − 02/1.6e−02^{-}	**6.23e − 01/4.8e−02**	6.63e − 01/6.2e−02^{-}	3.46e − 02/3.3e−03	**2.38e − 02/8.3e−03^{+}**
80_6	1.94e − 02/6.6e−03	**1.78e − 02/5.0e−03^{+}**	6.62e − 01/5.8e−02	**6.22e − 01/3.9e−02^{+}**	2.11e − 02/2.7e−03	**1.28e − 02/2.4e−03^{+}**
80_7	**2.67e − 02/8.5e−03**	2.86e − 02/9.3e−03^{-}	6.58e − 01/5.5e−02	**6.51e − 01/4.7e−02^{+}**	2.57e − 02/2.6e−03	**2.02e − 02/6.1e−03^{+}**
80_8	**1.46e − 02/5.0e−03**	1.90e − 02/5.3e−03^{-}	7.03e − 01/7.7e−02	**6.37e − 01/4.8e−02^{+}**	1.90e − 02/1.8e−03	**1.07e − 02/2.0e−03^{+}**
100_4	2.16e − 02/3.4e−03	**1.85e − 02/5.2e−03^{+}**	**6.72e − 01/4.1e−02**	7.03e − 01/6.6e−02^{-}	3.89e − 02/2.1e−03	**1.76e − 02/3.9e−03^{+}**
100_5	6.58e − 01/1.4e−01	**6.42e − 02/5.1e−02^{+}**	**8.62e − 01/6.6e−02**	9.03e − 01/9.1e−02^{-}	3.55e − 01/4.7e−02	**2.51e − 02/9.2e−03^{+}**
100_6	9.63e − 03/2.4e−03	**8.09e − 03/1.9e−03^{+}**	6.89e − 01/6.2e−02	**6.58e − 01/4.6e−02^{+}**	1.92e − 02/1.4e−03	**6.04e − 03/7.9e−04^{+}**
100_7	6.95e − 01/2.4e−01	**5.24e − 02/3.7e−02^{+}**	9.05e − 01/6.5e−02	**8.65e − 01/1.4e−01^{+}**	5.11e − 01/8.8e−02	**3.90e − 02/9.7e−03^{+}**
100_8	2.42e − 02/4.2e−03	**9.69e − 03/5.3e−03^{+}**	**7.22e − 01/6.1e−02**	7.62e − 01/5.7e−02^{-}	5.12e − 02/5.3e−03	**1.50e − 02/3.3e−03^{+}**

GD metric. The IGD generated by the HMOGWO is worse than its rivals on these five problems, which implies the HMOGWO cannot show a good performance on the above problems. However, the performance of HMOGWO is consistent and good for most instances in terms of the IGD metric. The outperformance of HMOGWO can be attributed to an effective social hierarchy scheme, by which the balance between exploration and exploitation can be further improved to some extent since solutions selected are from different ranks. However, the HMOGWO$_{bts}$ emphasize exploitation by using a binary tournament selection to choose three good solutions. Thus, HMOGWO$_{bts}$ cannot completely search unknown areas of the search space. Therefore, the improvement strategy can help to enhance the behavior of the proposed algorithm.

4.6. Efficiency of the hybrid strategy in HMOGWO

To test the effectiveness of hybrid strategy, HMOGWO is compared to the multi-objective grey wolf optimizer with only single search

Table 4
Mean and standard deviation value of all metrics between HMOGWO$_{bts}$ and HMOGWO.

Problem	GD (Mean/std)		Spread (Mean/std)		IGD (Mean/std)	
	HMOGWO$_{bts}$	HMOGWO	HMOGWO$_{bts}$	HMOGWO	HMOGWO$_{bts}$	HMOGWO
20_4	6.66e − 02/2.7e−02	**6.33e − 02/2.6e−02^{+}**	**1.26e + 00/1.9e−01**	1.27e + 00/2.0e−01$^{=}$	3.57e − 02/1.2e−02	**3.57e − 02/1.1e−02$^{=}$**
20_5	1.95e − 02/7.9e−03	**1.65e − 02/6.8e−03^{+}**	8.35e − 01/1.1e−01	**8.21e − 01/9.6e−02^{+}**	1.43e − 02/2.1e−03	**1.39e − 02/1.9e−03$^{=}$**
20_6	1.18e − 02/3.7e−03	**1.11e − 02/3.0e−03^{+}**	6.14e − 01/4.9e−02	**6.01e − 01/5.9e−02^{+}**	8.66e − 03/1.7e−03	**7.99e − 03/1.0e−03^{+}**
20_7	**1.74e − 02/4.2e−03**	1.84e − 02/6.5e−03^{-}	6.76e − 01/5.7e−02	**6.69e − 01/5.6e−02$^{=}$**	**1.19e − 02/1.8e−03**	1.25e − 02/2.5e−03^{-}
20_8	**1.56e − 02/8.4e−03**	1.61e − 02/7.4e−03^{-}	7.82e − 01/6.9e−02	**7.48e − 01/6.9e−02^{+}**	**1.14e − 02/4.8e−03**	1.16e − 02/4.1e−03$^{=}$
40_4	2.77e − 02/1.3e−02	**2.20e − 02/1.1e−02^{+}**	**9.33e − 01/1.3e−01**	9.74e − 01/1.4e−01^{-}	1.47e − 02/2.9e−03	**1.42e − 02/3.5e−03^{+}**
40_5	3.31e − 02/9.5e−03	**3.24e − 02/1.1e−02^{+}**	6.57e − 01/5.9e−02	**6.54e − 01/5.6e−02$^{=}$**	2.23e − 02/5.0e−03	**2.21e − 02/6.4e−03$^{=}$**
40_6	1.07e − 02/6.3e−03	**1.04e − 02/6.2e−03$^{=}$**	8.15e − 01/1.1e−01	**7.71e − 01/1.2e−01^{+}**	**7.03e − 03/1.9e−03**	8.07e − 03/1.4e−03^{-}
40_7	1.91e − 02/6.1e−03	**1.73e − 02/5.9e−03^{+}**	6.20e − 01/5.5e−02	**5.99e − 01/4.4e−02^{+}**	1.39e − 02/3.6e−03	**1.33e − 02/3.2e−03$^{=}$**
40_8	**2.61e − 02/8.4e−03**	2.74e − 02/7.3e−03^{-}	6.99e − 01/6.2e−02	**6.72e − 01/6.9e−02^{+}**	2.33e − 02/6.7e−03	**2.31e − 02/4.2e−03$^{=}$**
60_4	4.60e − 02/3.0e−02	**3.26e − 02/1.7e−02^{+}**	7.99e − 01/9.3e−02	**7.53e − 01/6.4e−02^{+}**	1.77e − 02/6.8e−03	**1.22e − 02/4.2e−03^{+}**
60_5	**3.13e − 02/1.8e−02**	3.15e − 02/1.5e−02$^{=}$	7.41e − 01/8.2e−02	**7.29e − 01/6.9e−02^{+}**	**1.61e − 02/5.2e−03**	2.08e − 02/9.6e−03^{-}
60_6	2.56e − 02/9.5e−03	**2.23e − 02/1.0e−02^{+}**	7.00e − 01/7.0e−02	**6.87e − 01/6.3e−02^{+}**	2.13e − 02/6.2e−03	**1.93e − 02/8.5e−03^{+}**
60_7	2.50e − 02/1.1e−02	**2.34e − 02/7.5e−03^{+}**	**6.15e − 01/3.7e−02**	6.18e − 01/4.6e−02$^{=}$	1.68e − 02/4.5e−03	**1.55e − 02/4.3e−03^{+}**
60_8	1.92e − 02/6.4e−03	**1.81e − 02/4.7e−03^{+}**	6.23e − 01/4.0e−02	**6.17e − 01/5.5e−02^{+}**	1.39e − 02/3.1e−03	**1.35e − 02/3.1e−03$^{=}$**
80_4	**7.10e − 02/6.5e−02**	7.37e − 02/3.1e−02^{-}	**7.97e − 01/6.9e−02**	8.21e − 01/8.9e−02^{-}	2.94e − 02/9.2e−03	**2.77e − 02/9.9e−03^{+}**
80_5	4.51e − 02/1.6e−02	**3.86e − 02/1.7e−02^{+}**	**6.73e − 01/7.2e−02**	6.80e − 01/5.2e−02^{-}	2.86e − 02/9.7e−03	**2.37e − 02/8.8e−03^{+}**
80_6	2.58e − 02/5.6e−03	**1.93e − 02/6.2e−03^{+}**	**6.15e − 01/4.1e−02**	6.24e − 01/5.5e−02^{-}	1.79e − 02/4.4e−03	**1.46e − 02/4.0e−03^{+}**
80_7	1.87e − 02/7.6e−03	**1.58e − 02/6.9e−03^{+}**	7.22e − 01/7.4e−02	**6.81e − 01/5.6e−02^{+}**	1.43e − 02/4.1e−03	**1.35e − 02/5.0e−03^{+}**
80_8	1.93e − 02/5.5e−03	**1.86e − 02/4.8e−03^{+}**	**6.20e − 01/4.0e−02**	6.27e − 01/3.7e−02^{-}	**1.56e − 02/3.0e−03**	1.61e − 02/3.3e−03^{-}
100_4	1.88e − 02/4.2e−03	**1.57e − 02/5.1e−03^{+}**	7.07e − 01/5.8e−02	**6.89e − 01/5.4e−02^{+}**	1.52e − 02/4.7e−03	**1.40e − 02/4.0e−03^{+}**
100_5	8.21e − 02/5.2e−02	**7.67e − 02/5.3e−02^{+}**	9.71e − 01/1.3e−01	**9.54e − 01/1.2e−01^{+}**	3.74e − 02/1.5e−02	**3.67e − 02/1.2e−02^{+}**
100_6	1.30e − 02/3.5e−03	**1.24e − 02/3.4e−03$^{=}$**	6.77e − 01/6.0e−02	**6.55e − 01/3.7e−02^{+}**	1.11e − 02/2.3e−03	**1.08e − 02/2.0e−03$^{=}$**
100_7	1.15e − 01/6.7e−02	**1.01e − 01/7.5e−02^{+}**	**9.81e − 01/1.4e−01**	9.85e − 01/1.5e−01$^{=}$	7.38e − 02/1.9e−02	**6.93e − 02/1.7e−02^{+}**
100_8	1.81e − 02/1.0e−02	**1.44e − 02/7.8e−03^{+}**	7.32e − 01/6.4e−02	**6.86e − 01/5.7e−02^{+}**	1.63e − 02/4.9e−03	**1.45e − 02/4.0e−03^{+}**

Table 5
Mean and standard deviation value of all metrics between MOGWO1 and HMOGWO.

Problem	GD (Mean/std)		Spread (Mean/std)		IGD (Mean/std)	
	MOGWO1	HMOGWO	MOGWO1	HMOGWO	MOGWO1	HMOGWO
20_4	**2.64e − 02/1.1e−02**	3.49e − 02/1.3e−02^{-}	1.39e + 00/1.8e−01	**1.24e + 00/1.7e−01^{+}**	2.05e − 02/3.6e−03	**1.80e − 02/4.9e−03^{+}**
20_5	2.62e − 02/1.0e−02	**1.86e − 02/9.1e−03^{+}**	1.29e + 00/1.3e−01	**8.97e − 01/1.3e−01^{+}**	2.03e − 02/2.1e−03	**1.93e − 02/4.0e−03^{+}**
20_6	1.53e − 02/3.6e−03	**1.50e − 02/5.4e−03$^{=}$**	8.62e − 01/1.1e−01	**6.59e − 01/5.8e−02^{+}**	1.76e − 02/2.0e−03	**8.73e − 03/1.6e−03^{+}**
20_7	1.13e − 02/3.0e−03	**1.02e − 02/3.2e−03^{+}**	8.46e − 01/1.1e−01	**6.99e − 01/6.0e−02^{+}**	1.61e − 02/1.4e−03	**9.60e − 03/1.1e−03^{+}**
20_8	2.62e − 02/7.8e−03	**1.36e − 02/7.0e−03^{+}**	9.88e − 01/1.0e−01	**7.86e − 01/7.0e−02^{+}**	3.84e − 02/3.0e−03	**1.17e − 02/3.1e−03^{+}**
40_4	**1.55e − 02/4.7e−03**	1.73e − 02/6.3e−03^{-}	1.28e + 00/1.1e−01	**9.91e − 01/1.8e−01^{+}**	2.13e − 02/2.1e−03	**1.75e − 02/3.2e−03^{+}**
40_5	**1.71e − 02/7.8e−03**	3.43e − 02/1.4e−02^{-}	1.14e + 00/1.6e−01	**7.22e − 01/6.6e−02^{+}**	2.10e − 02/2.0e−03	**1.72e − 02/2.2e−03^{+}**
40_6	3.29e − 02/1.3e−02	**2.55e − 02/2.0e−02^{+}**	1.04e + 00/1.4e−01	**8.94e − 01/9.9e−02^{+}**	3.19e − 02/1.7e−03	**1.58e − 02/2.6e−03^{+}**
40_7	**1.79e − 02/5.9e−03**	1.87e − 02/5.4e−03^{-}	7.86e − 01/7.7e−02	**5.89e − 01/4.3e−02^{+}**	1.91e − 02/2.1e−03	**9.65e − 03/1.4e−03^{+}**
40_8	1.77e − 02/4.0e−03	**1.13e − 02/5.7e−03^{+}**	1.01e + 00/1.3e−01	**7.15e − 01/6.4e−02^{+}**	2.48e − 02/1.5e−03	**1.50e − 02/1.6e−03^{+}**
60_4	1.55e − 02/6.8e−03	**8.12e − 03/3.6e−03^{+}**	9.31e − 01/1.0e−01	**8.06e − 01/7.5e−02^{+}**	3.09e − 02/1.5e−03	**1.41e − 02/2.5e−03^{+}**
60_5	**1.67e − 02/1.1e−02**	2.36e − 02/1.4e−02^{-}	9.07e − 01/9.2e−02	**7.98e − 01/7.6e−02^{+}**	2.89e − 02/1.5e−03	**2.20e − 02/3.0e−03^{+}**
60_6	2.60e − 02/7.0e−03	**1.26e − 02/9.2e−03^{+}**	8.40e − 01/1.0e−01	**7.16e − 01/6.1e−02^{+}**	2.33e − 02/1.3e−03	**1.31e − 02/2.3e−03^{+}**
60_7	**1.02e − 02/3.7e−03**	1.34e − 02/5.1e−03^{-}	8.92e − 01/9.7e−02	**6.48e − 01/6.0e−02^{+}**	1.84e − 02/1.5e−03	**9.73e − 03/1.4e−03^{+}**
60_8	**1.14e − 02/3.8e−03**	1.29e − 02/2.9e−03^{-}	8.49e − 01/1.3e−01	**6.24e − 01/4.1e−02^{+}**	2.28e − 02/1.8e−03	**1.17e − 02/1.4e−03^{+}**
80_4	1.34e − 02/7.1e−03	**9.12e − 03/4.8e−03^{+}**	9.15e − 01/9.2e−02	**7.41e − 01/7.1e−02^{+}**	2.61e − 02/1.3e−03	**1.04e − 02/2.0e−03^{+}**
80_5	**1.50e − 02/7.4e−03**	3.59e − 02/1.5e−02^{-}	8.41e − 01/7.7e−02	**6.85e − 01/8.9e−02^{+}**	**2.44e − 02/1.8e−03**	3.10e − 02/2.6e−03^{-}
80_6	1.75e − 02/8.5e−03	**1.18e − 02/5.7e−03^{+}**	9.02e − 01/1.3e−01	**6.40e − 01/4.2e−02^{+}**	2.54e − 02/1.3e−03	**1.69e − 02/1.8e−03^{+}**
80_7	**1.65e − 02/8.1e−03**	3.52e − 02/1.5e−02^{-}	8.31e − 01/7.6e−02	**6.83e − 01/7.0e−02^{+}**	**1.80e − 02/9.4e−04**	2.04e − 02/3.2e−03^{-}
80_8	1.46e − 02/5.4e−03	**1.24e − 02/3.3e−03^{+}**	7.31e − 01/6.3e−02	**5.99e − 01/3.4e−02^{+}**	1.96e − 02/1.5e−03	**8.77e − 03/1.0e−03^{+}**
100_4	**9.21e − 03/3.1e−03**	1.90e − 02/4.1e−03^{-}	8.71e − 01/1.1e−01	**6.81e − 01/4.2e−02^{+}**	1.69e − 02/1.1e−03	**1.53e − 02/1.4e−03^{+}**
100_5	8.44e − 02/3.4e−02	**5.79e − 02/9.0e−02^{+}**	1.14e + 00/1.0e−01	**1.05e + 00/1.1e−01^{+}**	6.15e − 02/7.9e−03	**3.23e − 02/1.3e−02^{+}**
100_6	**8.40e − 03/2.5e−03**	8.62e − 03/1.9e−03^{-}	8.16e − 01/7.6e−02	**6.45e − 01/3.8e−02^{+}**	1.47e − 02/7.4e−04	**6.83e − 03/9.4e−04^{+}**
100_7	6.50e − 02/2.7e−02	**3.05e − 02/2.5e−02^{+}**	9.88e − 01/6.3e−02	**9.14e − 01/1.1e−01^{+}**	5.57e − 02/2.9e−03	**2.45e − 02/4.6e−03^{+}**
100_8	5.10e − 02/1.2e−02	**3.64e − 02/2.9e−02^{+}**	9.64e − 01/1.2e−01	**7.97e − 01/8.7e−02^{+}**	5.15e − 02/3.5e−03	**3.72e − 02/7.0e−03^{+}**

Table 6
Mean and standard deviation value of all metrics between MOGWO2 and HMOGWO.

Problem	GD (Mean/std)		Spread (Mean/std)		IGD (Mean/std)	
	MOGWO2	HMOGWO	MOGWO2	HMOGWO	MOGWO2	HMOGWO
20_4	1.02e − 01/3.0e−02	**3.49e − 02/1.3e−02^{+}**	1.26e + 00/1.9e−01	**1.24e + 00/1.7e−01^{+}**	5.06e − 02/1.1e−02	**1.80e − 02/4.9e−03^{+}**
20_5	2.34e − 02/8.9e−03	**1.86e − 02/9.1e−03^{+}**	9.34e − 01/1.5e−01	**8.97e − 01/1.3e−01^{+}**	2.70e − 02/6.5e−03	**1.93e − 02/4.0e−03^{+}**
20_6	**1.40e − 02/5.8e−03**	1.50e − 02/5.4e−03^{-}	**6.49e − 01/6.0e−02**	6.59e − 01/5.8e−02$^{=}$	1.04e − 02/1.4e−03	**8.73e − 03/1.6e−03^{+}**
20_7	1.24e − 02/4.6e−03	**1.02e − 02/3.2e−03^{+}**	8.15e − 01/7.8e−02	**6.99e − 01/6.0e−02^{+}**	1.14e − 02/8.7e−04	**9.60e − 03/1.1e−03^{+}**
20_8	2.13e − 02/6.9e−03	**1.36e − 02/7.0e−03^{+}**	**7.58e − 01/7.6e−02**	7.86e − 01/7.0e−02^{-}	1.92e − 02/4.4e−03	**1.17e − 02/3.1e−03^{+}**
40_4	4.60e − 02/2.2e−02	**1.73e − 02/6.3e−03^{+}**	1.20e + 00/1.7e−01	**9.91e − 01/1.8e−01^{+}**	2.48e − 02/3.8e−03	**1.75e − 02/3.2e−03^{+}**
40_5	**1.83e − 02/1.0e−02**	3.43e − 02/1.4e−02^{-}	9.50e − 01/1.3e−01	**7.22e − 01/6.6e−02^{+}**	2.39e − 02/2.5e−03	**1.72e − 02/2.2e−03^{+}**
40_6	**2.31e − 02/2.0e−02**	2.55e − 02/2.0e−02^{-}	9.64e − 01/9.7e−02	**8.94e − 01/9.9e−02^{+}**	1.71e − 02/3.2e−03	**1.58e − 02/2.6e−03^{+}**
40_7	**1.73e − 02/5.6e−03**	1.87e − 02/5.4e−03^{-}	6.35e − 01/5.2e−02	**5.89e − 01/4.3e−02^{+}**	1.24e − 02/2.6e−03	**9.65e − 03/1.4e−03^{+}**
40_8	1.42e − 02/4.1e−03	**1.13e − 02/5.7e−03^{+}**	7.96e − 01/7.2e−02	**7.15e − 01/6.4e−02^{+}**	2.04e − 02/4.5e−03	**1.50e − 02/1.6e−03^{+}**
60_4	1.06e − 02/4.3e−03	**8.12e − 03/3.6e−03^{+}**	**8.05e − 01/9.4e−02**	8.06e − 01/7.5e−02$^{=}$	1.42e − 02/3.0e−03	**1.41e − 02/2.5e−03$^{=}$**
60_5	**1.30e − 02/8.5e−03**	2.36e − 02/1.4e−02^{-}	9.07e − 01/9.2e−02	**7.98e − 01/7.6e−02^{+}**	2.83e − 02/3.1e−03	**2.20e − 02/3.0e−03^{+}**
60_6	**1.16e − 02/5.2e−03**	1.26e − 02/9.2e−03^{-}	7.55e − 01/8.1e−02	**7.16e − 01/6.1e−02^{+}**	1.71e − 02/1.7e−03	**1.31e − 02/2.3e−03^{+}**
60_7	**1.22e − 02/3.5e−03**	1.34e − 02/5.1e−03^{-}	6.85e − 01/5.7e−02	**6.48e − 01/6.0e−02^{+}**	1.43e − 02/1.8e−03	**9.73e − 03/1.4e−03^{+}**
60_8	1.36e − 02/3.2e−03	**1.29e − 02/2.9e−03^{+}**	6.85e − 01/5.2e−02	**6.24e − 01/4.1e−02^{+}**	1.59e − 02/1.7e−03	**1.17e − 02/1.4e−03^{+}**
80_4	9.14e − 03/3.6e−03	**9.12e − 03/4.8e−03$^{=}$**	8.32e − 01/1.1e−01	**7.41e − 01/7.1e−02^{+}**	1.39e − 02/2.1e−03	**1.04e − 02/2.0e−03^{+}**
80_5	**2.97e − 02/1.5e−02**	3.59e − 02/1.5e−02^{-}	9.95e − 01/1.0e−01	**6.85e − 01/8.9e−02^{+}**	4.17e − 02/4.4e−03	**3.10e − 02/2.6e−03^{+}**
80_6	1.85e − 02/5.5e−03	**1.18e − 02/5.7e−03^{+}**	7.81e − 01/7.7e−02	**6.40e − 01/4.2e−02^{+}**	2.39e − 02/1.5e−03	**1.69e − 02/1.8e−03^{+}**
80_7	**2.95e − 02/8.4e−03**	3.52e − 02/1.5e−02^{-}	8.22e − 01/8.3e−02	**6.83e − 01/7.0e−02^{+}**	2.77e − 02/2.4e−03	**2.04e − 02/3.2e−03^{+}**
80_8	**1.08e − 02/2.4e−03**	1.24e − 02/3.3e−03^{-}	6.79e − 01/7.2e−02	**5.99e − 01/3.4e−02^{+}**	1.25e − 02/1.3e−03	**8.77e − 03/1.0e−03^{+}**
100_4	**1.91e − 02/8.7e−03**	1.90e − 02/4.1e−03$^{=}$	9.49e − 01/1.2e−01	**6.81e − 01/4.2e−02^{+}**	2.12e − 02/2.2e−03	**1.53e − 02/1.4e−03^{+}**
100_5	**3.82e − 02/3.3e−02**	5.79e − 02/9.0e−02^{-}	**9.97e − 01/1.5e−01**	1.05e + 00/1.1e−01^{-}	4.17e − 02/1.2e−02	**3.23e − 02/1.3e−02^{+}**
100_6	**1.06e − 02/6.1e−03**	8.62e − 03/1.9e−03^{-}	7.13e − 01/6.1e−02	**6.45e − 01/3.8e−02^{+}**	1.14e − 02/9.5e−04	**6.83e − 03/9.4e−04^{+}**
100_7	2.63e − 02/2.2e−02	**3.05e − 02/2.5e−02^{+}**	**8.36e − 01/1.2e−01**	9.14e − 01/1.1e−01^{-}	2.60e − 02/4.0e−03	**2.45e − 02/4.6e−03^{+}**
100_8	2.89e − 02/2.8e−02	**3.64e − 02/2.9e−02^{+}**	**7.47e − 01/9.1e−02**	7.97e − 01/8.7e−02^{-}	4.67e − 02/1.1e−02	**3.72e − 02/7.0e−03^{+}**

operator. In this experiment, the MOGWO1 denotes the algorithm with only grey wolf search operator. MOGWO2 represents the algorithm using only the genetic operator. Note that MOGWO2 only selects two good solutions to update information of population based on the improved social hierarchy. Obviously, the proposed HMOGWO contains the two above search strategies. Therefore, their combination should be separated during the test. Each experiment was conducted 30 independent times on each test problem for each algorithm. All the experimental results are presented in Tables 5 and 6.

From Tables 5 and 6, it can be observed that HMOGWO has a better performance with comparison to MOGWO1 and MOGWO2 with regard to all metrics, especially for IGD and Spread metrics. IGD metric is a priority indicator. Taking a problem 100_7 in Table 6 for example, HMOGWO has dominated MOGWO2 in terms of GD metric, but MOGWO2 can provide a better result for Spread metric. When it occurs, the overall performance of this algorithm is determined by the IGD metric. That is, HMOGWO outperforms MOGWO2 on this problem since HOMGWO is better than its compared algorithms for

Table 7
Mean and standard deviation of GD metric by NSGA-II, SPEA2, MOGWO, and HMOGWO.

problem	NSGA-II (Mean/std)	SPEA2 (Mean/std)	MOGWO (Mean/std)	HMOGWO (Mean/std)	
20_4	2.61e − 02/8.3e−03	**2.18e − 02/5.8e−03**	8.32e − 02/2.5e−02	3.74e − 02/1.6e−02	−
20_5	2.97e − 02/9.4e−03	**2.20e − 02/1.1e−02**	3.92e − 02/2.2e−02	2.67e − 02/1.2e−02	−
20_6	1.39e − 02/3.2e−03	**1.07e − 02/3.2e−03**	1.25e − 02/3.9e−03	1.15e − 02/2.7e−03	−
20_7	2.44e − 02/5.0e−03	1.99e − 02/3.2e−03	1.52e − 02/5.0e−03	**1.37e − 02/3.4e−03**	+
20_8	2.51e − 02/9.5e−03	1.92e − 02/6.4e−03	1.83e − 02/7.0e−03	**1.62e − 02/5.7e−03**	+
40_4	4.42e − 02/1.5e−02	3.66e − 02/2.6e−01	6.11e − 02/3.4e−02	**2.28e − 02/1.1e−02**	+
40_5	3.00e − 02/8.5e−03	2.75e − 02/9.0e−03	**2.33e − 02/1.2e−02**	3.32e − 02/1.2e−02	−
40_6	1.36e − 02/4.3e−03	**1.16e − 02/4.4e−03**	8.25e − 03/3.7e−03	7.22e − 03/2.9e−03	−
40_7	2.36e − 02/9.0e−03	**1.68e − 02/8.3e−03**	2.69e − 02/7.6e−03	2.80e − 02/9.6e−03	−
40_8	2.01e − 02/9.2e−03	1.62e − 02/6.9e−03	1.33e − 02/8.7e−03	**1.13e − 02/4.5e−03**	+
60_4	2.44e − 02/5.9e−03	2.11e − 02/3.9e−03	1.92e − 02/1.6e−03	**1.42e − 02/5.1e−02**	+
60_5	2.18e − 02/7.6e−03	**1.69e − 02/6.7e−03**	1.99e − 02/1.7e−02	3.77e − 02/1.5e−02	−
60_6	3.60e − 02/6.6e−03	2.48e − 02/8.1e−03	**1.85e − 02/7.1e−03**	2.21e − 02/1.0e−02	−
60_7	1.98e − 02/7.8e−03	**1.26e − 02/4.5e−03**	1.66e − 02/6.0e−03	1.45e − 02/5.0e−03	−
60_8	1.12e − 02/3.7e−03	6.98e − 03/2.8e−03	1.58e − 02/3.8e−03	**1.21e − 02/1.9e−03**	+
80_4	1.79e − 02/3.8e−03	1.30e − 02/4.1e−03	**9.35e − 03/4.0e−03**	1.04e − 02/3.5e−03	−
80_5	3.62e − 02/9.0e−03	**2.80e − 02/6.0e−03**	3.90e − 02/2.0e−03	4.51e − 02/9.6e−03	−
80_6	1.62e − 02/5.3e−03	**1.25e − 02/4.8e−03**	2.70e − 02/9.3e−03	2.67e − 02/5.4e−03	−
80_7	1.87e − 02/6.2e−03	**9.48e − 03/4.2e−03**	2.94e − 02/1.0e−02	3.04e − 02/9.2e−03	−
80_8	1.82e − 02/4.4e−03	**1.27e − 02/3.4e−03**	1.88e − 02/4.7e−03	1.77e − 02/4.0e−03	−
100_4	1.18e − 02/1.5e−03	**1.03e − 02/2.0e−03**	3.13e − 02/1.4e−02	2.74e − 02/7.7e−03	−
100_5	7.12e − 02/2.8e−02	**5.59e − 02/1.4e−02**	6.72e − 02/8.1e−02	8.46e − 02/5.4e−02	−
100_6	8.53e − 03/2.9e−03	**5.29e − 03/1.7e−03**	1.09e − 02/2.9e−03	9.53e − 03/2.4e−03	−
100_7	7.30e − 02/2.0e−02	5.05e − 02/1.1e−02	**2.64e − 02/1.7e−02**	4.19e − 02/3.4e−02	−
100_8	4.74e − 02/7.5e−03	4.63e − 02/7.9e−03	**2.24e − 02/8.0e−03**	3.13e − 02/1.7e−02	−

Table 8
Mean and standard deviation of Spread metric (Δ) obtained by NSGA-II, SPEA2, MOGWO, and HMOGWO.

problem	NSGA-II (Mean/std)	SPEA2 (Mean/std)	MOGWO (Mean/std)	HMOGWO (Mean/std)	
20_4	1.45e + 00/1.2e−01	1.43e + 00/9.3e−02	1.30e + 00/1.0e−01	**1.28e + 00/1.4e−01**	+
20_5	1.25e + 00/1.9e−01	1.34e + 00/1.1e−01	9.96e − 01/1.5e−01	**8.37e − 01/1.3e−01**	+
20_6	8.75e − 01/1.0e−01	7.63e − 01/1.5e−01	6.50e − 01/4.7e−02	**6.34e − 01/4.3e−02**	+
20_7	8.26e − 01/8.6e−02	7.48e − 01/1.3e−01	8.29e − 01/9.8e−02	**7.15e − 01/4.7e−02**	+
20_8	1.03e + 00/1.3e−01	9.99e − 01/1.7e−01	**7.54e − 01/8.5e−02**	8.27e − 01/8.0e−02	−
40_4	1.32e + 00/1.1e−01	1.27e + 00/1.6e−01	1.27e + 00/1.5e−01	**9.53e − 01/1.2e−01**	+
40_5	1.07e + 00/1.6e−01	1.08e + 00/1.9e−01	9.51e − 01/1.5e−01	**6.85e − 01/5.7e−02**	+
40_6	1.02e + 00/1.4e−01	9.03e − 01/1.8e−01	8.92e − 01/9.2e−02	**8.13e − 01/1.0e−01**	+
40_7	8.18e − 01/1.0e−01	7.40e − 01/9.6e−02	**6.60e − 01/8.4e−02**	6.62e − 01/5.7e−02	=
40_8	1.02e + 01/1.1e−01	9.44e − 01/1.8e−01	7.94e − 01/7.5e−02	**6.67e − 01/5.8e−02**	+
60_4	1.04e + 00/1.1e−01	8.83e − 01/2.2e−01	**7.88e − 01/8.0e−02**	8.85e − 01/7.7e−02	−
60_5	8.55e − 01/8.6e−02	8.22e − 01/1.8e−01	9.36e − 01/1.1e−01	**8.15e − 01/8.3e−02**	+
60_6	8.74e − 01/1.2e−01	8.23e − 01/1.6e−01	7.90e − 01/8.3e−02	**7.82e − 01/7.9e−02**	+
60_7	8.73e − 01/9.0e−02	8.36e − 01/1.2e−01	6.89e − 01/7.4e−02	**6.52e − 01/5.4e−02**	+
60_8	8.86e − 01/1.2e−01	8.26e − 01/1.1e−01	6.83e − 01/4.5e−02	**6.38e − 01/4.0e−02**	+
80_4	9.16e − 01/1.1e−03	8.34e − 01/1.3e−01	8.64e − 01/1.0e−01	**7.55e − 01/7.0e−02**	+
80_5	8.46e − 01/7.5e−02	7.54e − 01/7.8e−02	9.88e − 01/1.0e−01	**7.44e − 01/8.4e−02**	+
80_6	9.41e − 01/1.3e−01	8.85e − 01/1.6e−01	8.21e − 01/9.3e−02	**6.72e − 01/4.0e−02**	+
80_7	7.81e − 01/7.0e−02	7.25e − 01/9.2e−02	8.31e − 01/9.0e−02	**6.97e − 01/6.8e−02**	+
80_8	7.88e − 01/7.9e−02	7.05e − 01/7.8e−02	6.89e − 01/6.6e−02	**6.32e − 01/5.4e−02**	+
100_4	8.13e − 01/7.6e−02	6.96e − 01/1.3e−01	9.34e − 01/9.4e−02	**6.91e − 01/5.0e−02**	+
100_5	1.10e + 00/1.1e−01	1.05e + 00/1.7e−01	**9.41e − 01/1.3e−01**	1.01e + 00/8.4e−02	−
100_6	8.63e − 01/9.0e−02	7.19e − 01/1.0e−01	7.20e − 01/4.6e−02	**6.27e − 01/5.5e−02**	+
100_7	1.06e + 00/1.1e−01	9.91e − 01/1.3e−01	**8.72e − 01/8.7e−02**	9.02e − 01/1.3e−01	−
100_8	9.41e − 01/1.1e−01	8.60e − 01/1.7e−01	**7.08e − 01/7.3e−02**	7.39e − 01/9.1e−02	−

IGD metric. When the grey wolf search operator and genetic operator are combined, we can observe that the proposed HMOGWO wins the best IGD metric value on 23 out of 25 test problems and with statistical confidence on 22 problems. This means that the hybrid strategy can help to enhance comprehensive performance (convergence and diversity). Concerning the Spread metric, the HMOGWO obtains the best value on 19 out of 25 test problems and with statistical confidence on 19 problems, which denotes our proposal can generate more uniform PF than the others with only single operator. The proposed HMOGWO with the hybrid scheme is also competitive or superior to its two variations in terms of GD metric. Besides, the results computed by the proposed HMOGWO are more stable, which indicates that the hybrid strategy can strengthen the stability of the HMOGWO.

4.7. Performance comparison with other MOEAs

To further validate the effectiveness of the proposed HMOGWO rescheduling approach in MODWSP, we compared it with two well-known MOEAs: NSGA-II (Deb et al., 2002) and SPEA2 (Zitzler et al., 2001b). They are high-performing algorithms in the multi-objective evolutionary community, which have been successfully applied to many engineering applications. These MOEAs are usually used as the compared algorithms since they are very competitive or superior to other existing MOEAs when solving scheduling problems. NSGA-II is one of the most popular MOEAs. The characteristic of NSGA-II is that it uses a fast non-dominated sorting and crowding distance estimation procedure. A fast non-dominated sorting technique is used to assign the parent and offspring population to different levels of non-dominated solution fronts. A crowding distance strategy is employed to maintain diversity of the population. SPEA2 is another famous MOEA. The main characteristic of SPEA2 is the fitness assignment scheme and the use of an external archive method. The fitness of each solution is the sum of its strength raw fitness and a density indicator. An offspring population is composed of the non-dominated solutions in both the

Table 9
Mean and standard deviation of IGD metric obtained by NSGA-II, SPEA2, MOGWO, and HMOGWO.

problem	NSGA-II (Mean/std)	SPEA2 (Mean/std)	MOGWO (Mean/std)	HMOGWO (Mean/std)	
20_4	**2.38e – 02/3.7e–03**	2.40e – 02/4.9e–03	5.84e – 02/9.9e–03	2.77e – 02/3.9e–03	–
20_5	**1.67e – 02/3.2e–03**	1.74e – 02/2.2e–03	3.13e – 02/9.4e–03	2.01e – 02/4.5e–03	–
20_6	1.25e – 02/1.9e–03	1.26e – 02/1.6e–03	7.99e – 03/1.5e–03	**6.69e – 03/7.7e–04**	+
20_7	2.47e – 02/1.7e–03	2.40e – 02/1.9e–03	1.50e – 02/1.1e–03	**1.23e – 02/1.8e–03**	+
20_8	3.44e – 02/3.4e–03	3.47e – 02/2.8e–03	2.34e – 02/6.2e–03	**1.45e – 02/4.6e–03**	+
40_4	3.34e – 02/4.6e–03	3.26e – 02/4.9e–03	3.01e – 02/5.8e–03	**1.98e – 02/5.3e–03**	+
40_5	2.47e – 02/1.5e–03	2.44e – 02/1.6e–03	2.43e – 02/3.4e–03	**1.67e – 02/1.9e–03**	+
40_6	2.98e – 02/1.9e–03	3.03e – 02/1.7e–03	**1.16e – 02/2.3e–03**	1.17e – 02/2.4e–03	=
40_7	1.45e – 02/1.8e–03	1.41e – 02/1.3e–03	1.73e – 02/3.4e–03	**1.14e – 02/2.6e–03**	+
40_8	2.37e – 02/2.2e–03	2.36e – 02/1.9e–03	1.61e – 02/3.1e–03	**1.22e – 02/2.2e–03**	+
60_4	3.42e – 02/1.4e–03	3.33e – 02/1.4e–03	2.00e – 02/3.4e–03	**1.74e – 02/2.6e–03**	+
60_5	1.62e – 02/1.1e–03	**1.59e – 02/1.1e–03**	2.17e – 02/2.5e–03	1.63e – 02/1.8e–03	–
60_6	3.67e – 02/2.5e–03	3.55e – 02/2.5e–03	3.41e – 02/4.6e–03	**2.68e – 02/5.0e–03**	+
60_7	1.67e – 02/1.0e–03	1.74e – 02/1.0e–03	1.32e – 02/3.1e–03	**9.16e – 03/1.4e–03**	+
60_8	1.41e – 02/9.0e–04	1.44e – 02/1.0e–03	1.54e – 02/1.7e–03	**9.15e – 03/1.3e–03**	+
80_4	2.73e – 02/1.0e–03	2.73e – 02/1.4e–03	1.49e – 02/2.4e–03	**1.24e – 02/2.5e–03**	+
80_5	**2.60e – 02/2.4e–03**	2.63e – 02/1.9e–03	4.55e – 02/4.2e–03	3.38e – 02/2.5e–03	–
80_6	1.77e – 02/1.2e–03	**1.74e – 02/9.4e–04**	2.74e – 02/2.2e–03	2.18e – 02/1.7e–03	–
80_7	1.72e – 02/1.5e–03	**1.58e – 02/1.0e–03**	2.83e – 02/2.5e–03	2.10e – 02/2.8e–03	–
80_8	1.41e – 02/1.0e–03	1.43e – 02/9.5e–04	1.83e – 02/2.0e–03	**1.15e – 02/1.2e–03**	+
100_4	**1.29e – 02/8.8e–04**	1.30e – 02/6.9e–04	2.33e – 02/2.0e–03	1.76e – 02/1.7e–03	–
100_5	5.31e – 02/4.7e–03	5.39e – 02/4.4e–03	3.58e – 02/6.6e–03	**3.13e – 02/7.2e–03**	+
100_6	1.25e – 02/6.9e–04	1.35e – 02/8.6e–04	1.29e – 02/1.5e–03	**8.63e – 03/9.5e–04**	+
100_7	3.86e – 02/3.1e–03	3.68e – 02/2.5e–03	2.22e – 02/3.8e–03	**1.87e – 02/2.4e–03**	+
100_8	7.08e – 02/3.8e–03	6.93e – 02/3.9e–03	4.42e – 02/1.0e–03	**3.51e – 02/7.9e–03**	+

parent population and the external archive. The archive approach is used to determine whether a non-dominated solution should be stored or not based on the distance to its kth nearest neighbor. In addition, HMOGWO is also compared with the multi-objective grey wolf optimizer without any improved strategy (i.e., MOGWO). MOGWO is a new MOEA based on GWO. Choosing the MOGWO as the compared algorithm is to further demonstrate the effectiveness of the improvement strategy in the HMOGWO. They use the same initialization strategy according to the proposed encoding scheme. The other parameters of MOEAs are set in Section 4.3. Each experiment was conducted 30 independent times on each test problem for each algorithm. All the experimental results are presented in Tables 7–9.

Table 10
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
Submerged arc automatic welder	3	Fillet welding

It can be observed that the HMOGWO is significantly better than other MOEAs on most instances in terms of Spread (Table 8) and IGD (Table 9) metrics. Concerning the GD (Table 7) metric, the HMOGWO is better than NSGA-II and MOGWO, and is also competitive with SPEA2. These results indicate that the HMOGWO has the best or competitive performance among these four MOEAs with regard to convergence, diversity and coverage. The reasons for superiority of HMOGWO can be ascribed to the following facts: First, it adopts the social hierarchy scheme to improve the quality of population since the search process is guided by three good solutions, and thus the HMOGWO can quickly converge to the optimal solutions. Second, to maintain diversity of population, the three good solutions may be from different non-dominated levels, which can improve search diversity to some extent. This viewpoint can be supported by the fact that HMOGWO outperforms the other MOEAs in terms of Spread metric. Third, to enhance exploration, the hybrid search scheme that combines the search operator of GWO and the genetic operator is used to explore new unvisited areas of the search space, as different search operators may have different search directions.

To visualize the stastical results of these MOEAs, Fig. 11 presents three boxplots of IGD with three level scale instances (i.e., 40_4, 60_4 and 100_4) over 30 runs. These figures confirm some conclusions derived from the above numerical analysis. This indicates that the HMOGWO significantly ouperforms its rivals in terms of the IGD. Therefore, the proposed HMOGWO is a promising method for this dynamic scheduling problem.

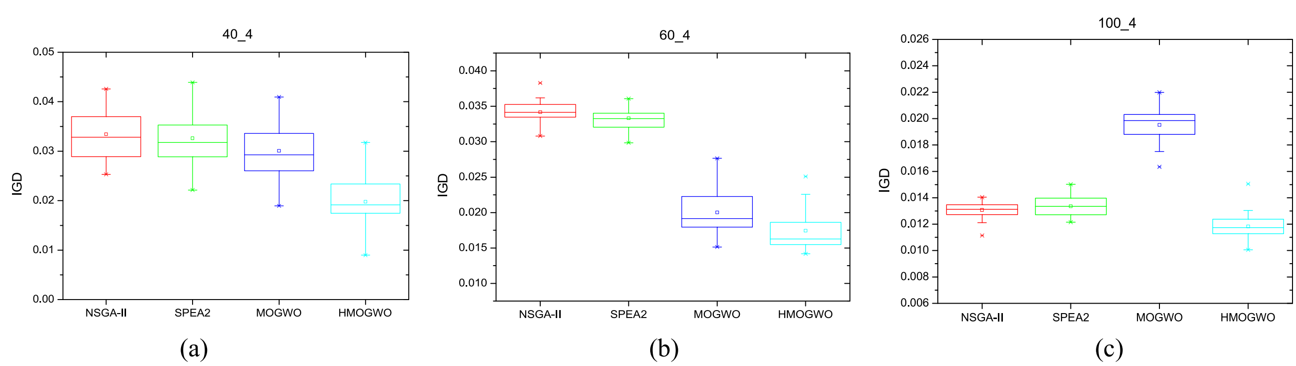

Fig. 11. Boxplot of IGD obtained by different MOEAs on (a) 40_4 problem, (b) 60_4 problem, and (c) 100_4 problem.

Table 11
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 12
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 13
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 14
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 15
The coefficient of penalty for machine usage at each stage.

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

4.8. Case study

The following case is from a real-world welding workshop. This welding shop is required to process 10 types of jobs that should be passed through 5 stages in the same flow. Assume that the machine breakdown time is at 250 min and the time to repair machine is 20 min. The poor quality are assumed to follow a Gaussian distribution with the mean of zero. The probability for poor quality of each operation is 0.05. The job release time delay vary uniformly in [0,10]. The other relative data is provided in Tables 10–17.

The above MOEAs are also applied to this real-world case. The parameters of MOEAs are set according to Section 4.3. 30 independent runs of each MOEA were conducted. Statistic results are summarized in Table 18. It can be clearly observed that the HMOGWO is better than the other MOEAs in terms of all metrics on this case. It implies that HMOGWO is very suitable for solving this type of problem. Thus, we adopted HMOGWO to optimize this case.

Fig. 12 shows the PFs with the best IGD metric obtained by different algorithms under different perspectives. It can be observed that the HMOGWO can approach closer to the true PF whereas MOGWO, NSGA-II and SPEA2 fail to reach the true PF (they trap into a local optimal point in this real-world case). Based on this figure, the HMOGWO is superior to its compared algorithms in the terms of the convergence performance. It also implies that the proposed HMOGWO is capable of providing better solutions than NSGA-II,

Table 16
The normal release time.

Job release time	job 1	job 2	Job3	job 4	job 5	job 6	Job 7	Job 8	Job 9	Job 10
Release time	14	10	12	15	5	5	6	8	7	3

Table 17
The repair times of each operation when job with poor quality occurs.

Setup time (min)	Job1	Job2	Job3	Job4	Job5	Job6	Job7	Job8	Job9	Job10
Slicing	10	15	20	10	5	9	13	12	15	10
Web grouping	21	20	18	15	14	10	8	12	24	10
Internal seam weld	15	10	25	16	5	8	10	14	15	5
Assembling	10	12	14	5	10	5	8	15	20	25
Fillet welding	12	15	24	10	4	6	9	10	5	6

Table 18
Mean and standard deviation value by NSGA-II, SPEA2, MOGWO and HMOGWO on a real-world case.

Case	GD Mean (std)	Δ Mean (std)	IGD Mean (std)
NSGA-II	1.55e − 02/6.2e−03	8.60e − 01/5.5e−02	5.84e − 03/6.6e−04
SPEA2	9.95e − 03/3.0e−03	7.01e − 01/9.9e−02	5.60e − 03/8.3e−04
MOGWO	1.44e − 02/8.2e−03	7.14e − 01/5.7e−02	5.93e − 03/1.0e−03
HMOGWO	**7.25e − 03/ 4.3e−03⁺**	**7.00e − 01/ 8.2e−02⁼**	**3.78e − 03/ 1.1e−03⁺**

MOGWO and SPEA2 with regard to quality of solutions. Fig. 12 (a) presents a 3-dimensional view with three criteria. It can be observed from Fig. 12 (b) that there is a conflicting relationship between the makespan and machine load. For example, the minimum makespan (i.e., 447.5 min) can be found at the point A in which the machine load (i.e., 17.0) is maximum. On the contrary, the minimum machine load (i.e., 10) occurs at point B where the makespan (i.e., 659 min) is maximum. Fig. 12 (c) presents a graphical view only considering makespan and instability. It is also can be seen that the two criteria have a conflicting trend. Similarly, Fig. 12 (d) gives a 2-dimensional view involving the machine load and instability. These observations are consistent with our views that the three objectives of the optimization problem are usually in conflict with each other. The point D in Fig. 12 (a) (makespan is 690.5 min, machine load is 17.2, and instability is 455) represents the original solution obtained by the First-In and First-Out strategy (FIFO, it is a commonly adopted approach in this practical welding scheduling production), which is dominated by solutions found by the HMOGWO. The makespan at point A is decreased by 54.3% compared with the one at point D. The machine load at point B is reduced by 72% compared with the point at D. The stability at point C is significantly superior to that at the point D. In addition, the three PFs and the corresponding solutions at the point A, B and C are given below.

The PF is (447.5, 17.0, 210.0) and solution at point A is

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

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

The PF is (659.0, 10.0, 62.0) and solution at point B is

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

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

The PF is (532.5, 14.85, 0) and solution at point C is

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

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

Point A, B and C correspond to three extreme objectives of this case, respectively. More specifically, point A corresponds to the minimum value of makespan. B corresponds to the minimum value of machine load. C corresponds to the minimum value of instability. These solutions of point A, B and C are all non-dominated. Decision-maker can select an appropriated solution from non-dominated solution according to the requirements of real-world production. For instance, point A can be regarded as the final result in the practical sense if executors emphasize time cost or production efficiency. B will be chosen as the final result in the practical sense if executors emphasize the stability of production systems. Similarity, point C will be chosen as the final result in the practical sense if works are more interested in reduction of machine load. In this real-world welding shop, workers emphasize the makespan, thus, A is chosen as the final result in the practical sense. The Gantt charts of the solutions of point A, B and C are plotted in Figs. 13–15, respectively. Note that the length of the color box represents the actual processing time, where the number of the white bar denotes the number of the additional machines assigned to the corresponding operation. The part in blue denotes the time interval of machine breakdown. The part in red represents the repaired time interval due to the poor quality of operation.

5. Conclusions and future work

In this study, a real-world dynamic welding scheduling problem is investigated from the theory and practical application perspectives. First, we formulate a multi-objective mathematical model which considered three dynamic events consisting of machine breakdown, job with release time delay and job with poor quality simultaneously. This model also involves sequence dependent setup time, job dependent transportation times and controllable processing times. Then, we develop a hybrid multi-objective grey wolf optimizer (HMOGWO) to address this dynamic problem with the objective to minimize the makespan, machine load, and instability simultaneously. The proposed HMOGWO contains three major improvement strategies. The effi-

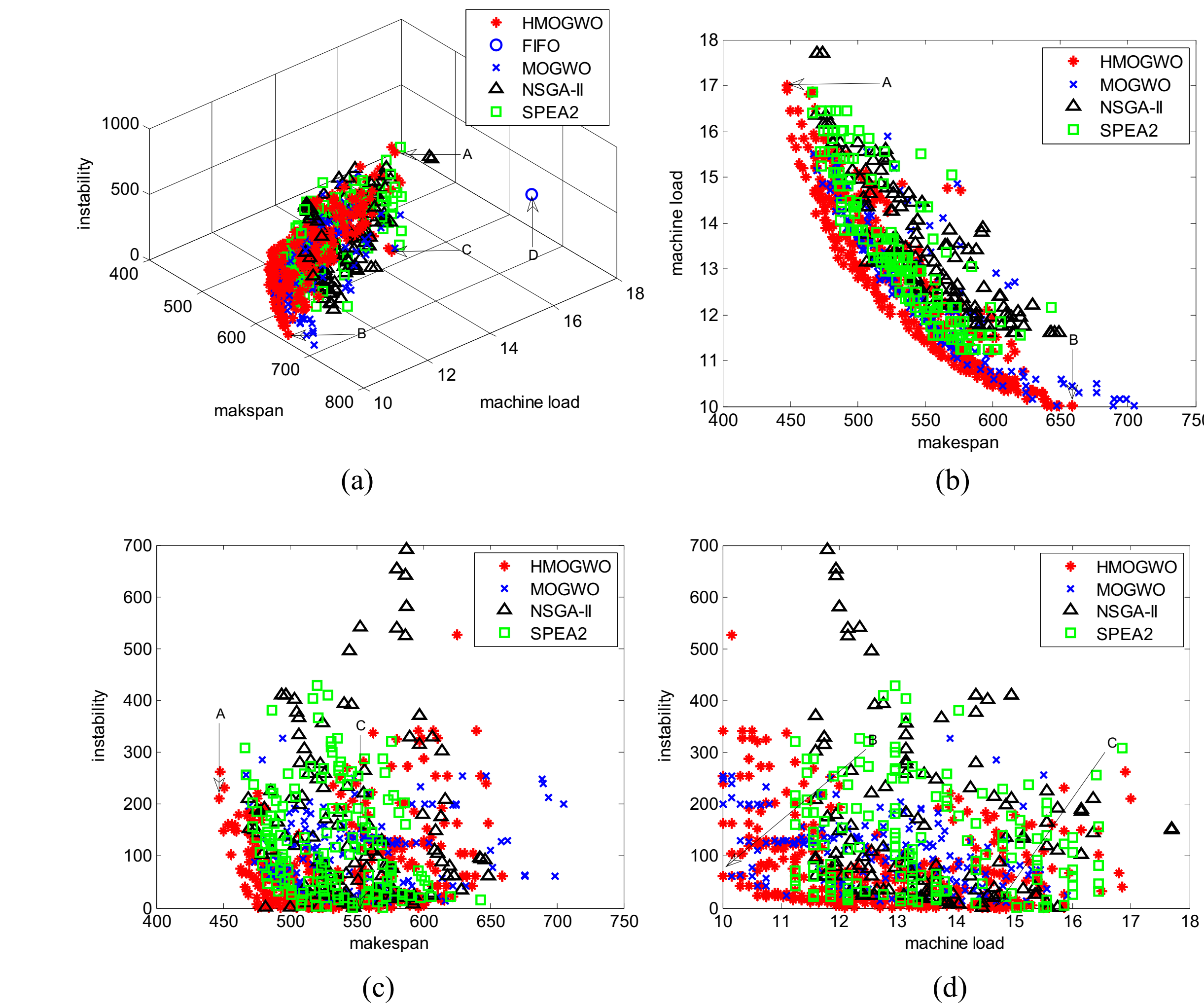

Fig. 12. Pareto fronts obtained by different MOEAs under different angles, (a)PF by different MOEAs and FIFO with three criteria, (b) PF with makspan and machine load, (c) PF with makespan and instability, (d) PF with machine load and instability.

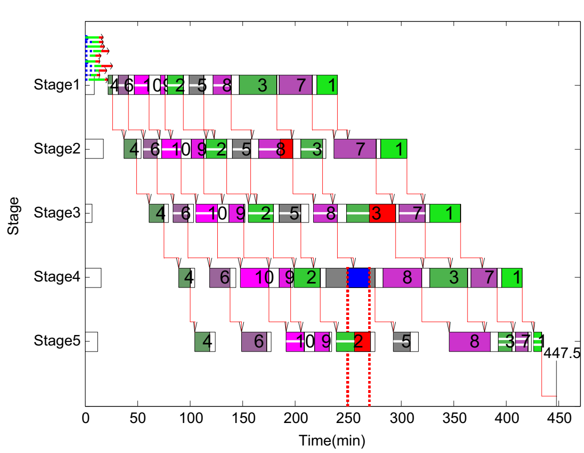

Fig. 13. The Gantt chart of the point A obtained by the proposed HMOGWO. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

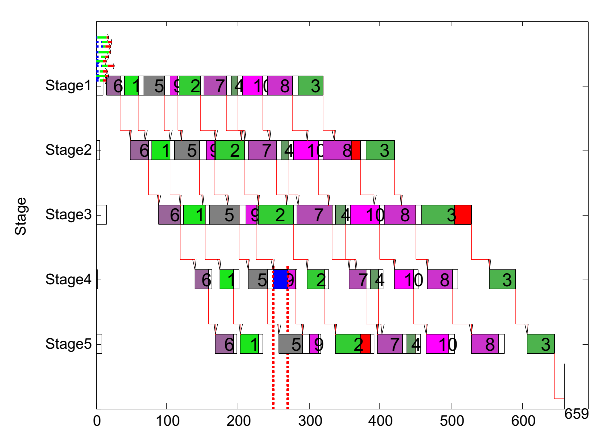

Fig. 14. The Gantt chart of the point B obtained by the proposed HMOGWO. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

ciency of each improvement strategy was verified through conducting experiments on randomly generated instances with regard to convergence, spread and comprehensive metrics. Furthermore, we compare the proposed HMOGWO with other MOEAs including NSGA-II, SPEA2, and MOGWO. Experimental results show that HMOGWO outperformed the other MOEAs for most instances in terms of three metrics. Finally, the HMOGWO is successfully applied to a real-world case, and the result has considerable instructive value in real-world production. The main contributions of this work are as follows:

(1) A new dynamic scheduling model for welding process is formulated, which is easily extended to other types of scheduling problems.

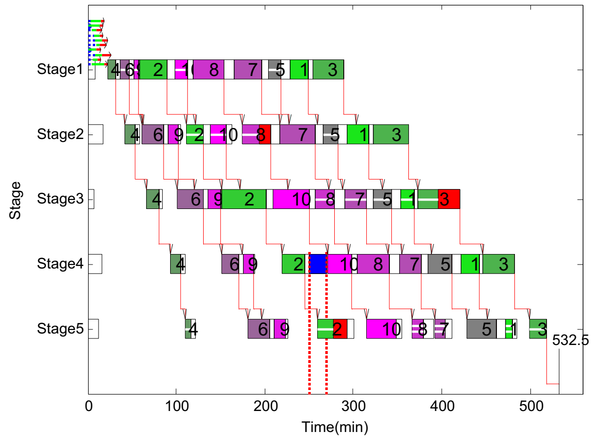

Fig. 15. The Gantt chart of the point C obtained by the proposed HMOGWO. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

(2) A hybrid multi-objective grey optimizer is proposed to deal with this MODWSP.
(3) A new encoding scheme is designed to accommodate the characteristic of the MODWSP.
(4) This study can provide new insights on modelling and addressing practical welding scheduling problems.

Although the proposed HMOGWO shows a good performance compared to other MOEAs in solving MODWSP, there are still some limitations for HMOGWO. One limitation of this method is that it does not use information of problem properties to improve the behavior of the proposal for the MODWSP. The hybridization of GWO and GA may give rise to another issue, which makes the method consume a long time to obtain a set of non-dominated solutions. With respect to future work, one research direction may extend the method for addressing dynamic scheduling problems in other types of scheduling problems. Another interesting research direction is to formulate a new mathematical model for the welding scheduling problem from a new aspect of green manufacturing.

Acknowledgments

This research work was supported by the National Natural Science Foundation of China (NSFC) under Grant nos. 51375004 and 51435009, National Key Technology Support Program under Grant no. 2015BAF01B04, and Youth Science & Technology Chenguang Program of Wuhan under Grant no. 2015070404010187.

References

Ahmadizar, F., Shahmaleki, P., 2014. Group-shop scheduling with sequence-dependent set-up and transportation times. Appl. Math. Model. 38, 5080–5091.
Ciavotta, M., Minella, G., Ruiz, R., 2013. Multi-objective sequence dependent setup times permutation flowshop: a new algorithm and a comprehensive study. Eur. J. Oper. Res. 227, 301–313.
Cowling, P., Johansson, M., 2002. Using real time information for effective dynamic scheduling. Eur. J. Oper. Res. 139, 230–244.
Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: nsga-ii. IEEE Trans. Evol. Comput. 6, 182–197.
Durillo, J.J., Nebro, A.J., 2011. jMetal: a Java framework for multi-objective optimization. Adv. Eng. Softw. 42, 760–771.
El Sehiemy, R., El-Ela, A.A., Shaheen, A., 2013. Multi-objective fuzzy-based procedure for enhancing reactive power management. IET Gener., Transm. Distrib. 7, 1453–1460.
Emary, E., Zawbaa, H.M., Hassanien, A.E., 2016. Binary grey wolf optimization approaches for feature selection. Neurocomputing 172, 371–381.
Gao, L., Li, X., Wen, X., Lu, C., Wen, F., 2015. A hybrid algorithm based on a new neighborhood structure evaluation method for job shop scheduling problem. Comput. Ind. Eng. 88, 417–429.
Garey, M.R., Johnson, D.S., Sethi, R., 1976. The complexity of flowshop and jobshop scheduling. Math. Oper. Res. 1, 117–129.
Garg, A., Garg, A., Tai, K., Sreedeep, S., 2014. An integrated SRM-multi-gene genetic programming approach for prediction of factor of safety of 3-D soil nailed slopes. Eng. Appl. Artif. Intell. 30, 30–40.
Gen, M., Lin, L., 2014. Multiobjective evolutionary algorithm for manufacturing scheduling problems: state-of-the-art survey. J. Intell. Manuf. 25, 849–866.
Goldberg, D.E., 1989. Genetic algorithms in search optimization and machine learning. Addison-wesley Reading, Menlo Park.
Holland, John H., 1992. Adaptation in natural and artificial systems. MIT Press, Cambridge, MA.
Jalilvand-Nejad, A., Fattahi, P., 2015. A mathematical model and genetic algorithm to cyclic flexible job shop scheduling problem. J. Intell. Manuf. 26, 1085–1098.
Jayakumar, N., Subramanian, S., Ganesan, S., Elanchezhian, E.B., 2016. Grey wolf optimization for combined heat and power dispatch with cogeneration systems. Int. J. Electr. Power Energy Syst. 74, 252–264.
Katragjini, K., Vallada, E., Ruiz, R., 2013. Flow shop rescheduling under different types of disruption. Int. J. Prod. Res. 51, 780–797.
Katunin, A., Przystalka, P.Ieee, 2014. Meta-optimization method for wavelet-based damage identification in composite structures, Federated Conference on Computer Science and Information Systems, 2014, pp. 429-438.
Kazakov, A.L., Lempert, A.A., 2015. On mathematical models for optimization problem of logistics infrastructure. Int. J. Artif. Intell.™ 13, 200–210.
Li, J.-q., Pan, Q.-k., Mao, K., 2015. A discrete teaching-learning-based optimisation algorithm for realistic flowshop rescheduling problems. Eng. Appl. Artif. Intell. 37, 279–292.
Li, X., Shao, X., Gao, L., Qian, W., 2010. An effective hybrid algorithm for integrated process planning and scheduling. Int. J. Prod. Econ. 126, 289–298.
Lin, D., Lee, C.K.M., Ho, W., 2013. Multi-level genetic algorithm for the resource-constrained re-entrant scheduling problem in the flow shop. Eng. Appl. Artif. Intell. 26, 1282–1290.
Lin, Q., Gao, L., Li, X.Y., Zhang, C.J., 2015. A hybrid backtracking search algorithm for permutation flow-shop scheduling problem. Comput. Ind. Eng. 85, 437–446.
Loubière, P., Jourdan, A., Siarry, P., Chelouah, R., 2016. A sensitivity analysis method for driving the Artificial Bee Colony algorithm's search process. Appl. Soft Comput. 41, 515–531.
Lu, C., Xiao, S., Li, X., Gao, L., 2016. An effective multi-objective discrete grey wolf optimizer for a real-world scheduling problem in welding production. Adv. Eng. Softw. 99, 161–176.
Ma, H., Simon, D., Fei, M., Shu, X., Chen, Z., 2014. Hybrid biogeography-based evolutionary algorithms. Eng. Appl. Artif. Intell. 30, 213–224.
Manjarres, D., Landa-Torres, I., Gil-Lopez, S., Del Ser, J., Bilbao, M.N., Salcedo-Sanz, S., Geem, Z.W., 2013. A survey on applications of the harmony search algorithm. Eng. Appl. Artif. Intell. 26, 1818–1831.
Mendes, N., Neto, P., Loureiro, A., Moreira, A.P., 2016. Machines and control systems for friction stir welding: a review. Mater. Des. 90, 256–265.
Mirjalili, S., Mirjalili, S.M., Lewis, A., 2014. Grey wolf optimizer. Adv. Eng. Softw. 69, 46–61.
Mirjalili, S., Saremi, S., Mirjalili, S.M., Coelho, Ld.S., 2016. Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. Expert Syst. Appl. 47, 106–119.
Mou, J., Gao, L., Li, X., Lu, C., Hu, H., 2014. Optimisation of the reverse scheduling problem by a modified genetic algorithm. Int. J. Prod. Res., 1–14.
Nawaz, M., Enscore, E.E., Ham, I., 1983. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega 11, 91–95.
Nie, L., Gao, L., Li, P.G., Li, X.Y., 2013. A GEP-based reactive scheduling policies constructing approach for dynamic flexible job shop scheduling problem with job release dates. J. Intell. Manuf. 24, 763–774.
Niu, S.H., Ong, S.K., Nee, A.Y.C., 2013. An improved intelligent water drops algorithm for solving multi-objective job shop scheduling. Eng. Appl. Artif. Intell. 26, 2431–2442.
Noshadi, A., Shi, J., Lee, W.S., Shi, P., Kalam, A., 2015. Optimal PID-type fuzzy logic controller for a multi-input multi-output active magnetic bearing system. Neural Comput. Appl., 1–16.
Ouelhadj, D., Petrovic, S., 2009. A survey of dynamic scheduling in manufacturing systems. J. Sched. 12, 417–431.
Pan, Q.K., Wang, L., Gao, L., Li, W.D., 2011. An effective hybrid discrete differential evolution algorithm for the flow shop scheduling with intermediate buffers. Inf. Sci. 181, 668–685.
Panda, M., Abraham, A., 2015. Hybrid evolutionary algorithms for classification data mining. Neural Comput. Appl. 26, 507–523.
Precup, R.E., David, R.C., Petriu, E.M., 2016b. Grey wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity. IEEE Trans. Ind. Electron., [1].
Precup, R.-E., David, R.-C., Petriu, E.M., Preitl, S., Radac, M.-B., 2013. Fuzzy logic-based adaptive gravitational search algorithm for optimal tuning of fuzzy-controlled servo systems. IET Control Theory Appl. 7, 99–107.
Precup, R.-E., David, R.-C., Petriu, E.M., Szedlak-Stinean, A.-I., Bojan-Dragos, C.-A., 2016a. Grey wolf optimizer-based approach to the tuning of Pi-fuzzy controllers with a reduced process parametric sensitivity. IFAC-Pap. 49, 55–60.
Priore, P., Gómez, A., Pino, R., Rosillo, R., 2014. Dynamic scheduling of manufacturing systems using machine learning: an updated review. Artif. Intell. Eng. Des., Anal. Manuf. 28, 83–97.
Rangsaritratsamee, R., Ferrell, W.G., Kurz, M.B., 2004. Dynamic rescheduling that simultaneously considers efficiency and stability. Comput. Ind. Eng. 46, 1–15.
Ruiz, R., Stützle, T., 2007. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. Eur. J. Oper. Res. 177, 2033–2049.
Ruiz, R., Maroto, C., Alcaraz, J., 2006. Two new robust genetic algorithms for the

flowshop scheduling problem. Omega 34, 461–476.
Saremi, S., Mirjalili, S.Z., Mirjalili, S.M., 2015. Evolutionary population dynamics and grey wolf optimizer. Neural Comput. & Appl. 26, 1257–1263.
Shen, X.-N., Yao, X., 2015. Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems. Inf. Sci. 298, 198–224.
Stoop, P.P., Wiers, V.C., 1996. The complexity of scheduling in practice. Int. J. Oper. Prod. Manag. 16, 37–53.
Suresh, V., Chaudhuri, D., 1993. Dynamic scheduling—a survey of research. Int. J. Prod. Econ. 32, 53–63.
Tang, L., Wang, X., 2013. A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems. IEEE Trans. Evol. Comput. 17, 20–45.
Tang, L., Zhao, Y., Liu, J., 2014. An improved differential evolution algorithm for practical dynamic scheduling in steelmaking-continuous casting production. IEEE Trans. Evol. Comput. 18, 209–225.
Veldhuizen, D.A.V., Lamont, G.B., 1998. Lamont multiobjective evolutionary algorithm research: a history and analysis, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Ohio.
Yaochu, J., Branke, J., 2005. Evolutionary optimization in uncertain environments-a survey. IEEE Trans. Evol. Comput. 9, 303–317.
Zeng, B., Dong, Y., 2016. An improved harmony search based energy-efficient routing algorithm for wireless sensor networks. Appl. Soft Comput. 41, 135–147.
Zhang, G., Gao, L., Shi, Y., 2011. An effective genetic algorithm for the flexible job-shop scheduling problem. Expert Syst. Appl. 38, 3563–3573.
Zitzler, E., Thiele, L., 1999. Multiobjective evolutionary algorithms: a comparative case study and the Strength Pareto approach. IEEE Trans. Evol. Comput. 3, 257–271.
Zitzler, E., Laumanns, M., Thiele, L., 2001a. SPEA2: improving the strength Pareto evolutionary algorithm, Eurogen, pp. 95–100.
Zitzler, E., Laumanns, M., Thiele, L., 2001b. SPEA2: improving the strength Pareto evolutionary algorithm, Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK).