Expert Systems With Applications 140 (2020) 112844

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A bi-objective dynamic collaborative task assignment under uncertainty using modified MOEA/D with heuristic initialization

Wenqin Xu[a,b], Chen Chen[a,b,*], Shuxin Ding[c], Panos M. Pardalos[d]

[a] School of Automation, Beijing Institute of Technology, Beijing, 100081, China
[b] State Key Laboratory of Intelligent Control and Decision of Complex System, Beijing, 100081, China
[c] Signal and Communication Research Institute, China Academy of Railway Sciences, Beijing, 100081, China
[d] Center for Applied Optimization, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, 32611, USA

ARTICLE INFO

Article history:
Received 22 April 2019
Revised 8 July 2019
Accepted 25 July 2019
Available online 26 July 2019

Keywords:
Task assignment
Uncertainty
MOEA/D
Heuristic
Taguchi method

ABSTRACT

The collaborative task assignment involved in Command and Control Systems is a key problem to be solved. The existing researches have their limitations to the natures of dynamic, uncertainty, flexibility and cooperation in a defensive scenario. Aiming at these, we formulate a bi-objective multi-stage task assignment model. The cooperation between sensor platforms and weapon platforms is considered. Also a Soyster robust model is introduced to handle uncertainty in a real time assignment process. Multi-objective evolutionary algorithm based on decomposition (MOEA/D) is adopted for the purpose of command flexibility. Currently, research focusing on multi-objective heuristics is relatively lacking. In this paper, we present a novel constructive heuristic for initializing the population. It successively adds quaternions into the assignment scheme to construct a solution set along the Pareto front, which is an interesting heuristic framework for multi-objective problems. We have also modified MOEA/D with nadir-based Tchebycheff and utilized the proposed neighbor matching strategy to gain better performance. Since algorithms are sensitive to their parameters, the Taguchi method with a novel response metric is utilized to calibrate the parameters. Numerical experiments demonstrate the superiority of the proposed algorithm and the necessity of a robust model.

1. Introduction

The revolution of network-centric warfare has systematically organized the originally separated combat platforms, thereby achieving a high level of information sharing and increasing the chance of more efficient operations. However, the collaborative task assignment of multi-platforms remains an urgent problem that needs to be solved. Generally, in a Command & Control System (CCS), combat platforms can be divided into three categories: sensor platforms (SP), weapon platforms (WP), and Command & Control platforms (CCP). Each part plays an essential role in the completion of the overall combat mission. From the perspective of the classic 'Observe-Orient-Decide-Act' (OODA) loop developed by Boyd, SP plays the role of 'Observe'. It acquires battlefield information to support the decision making of CCP and provide fire guidance for WP. CCP is the controller of the system, and it plays the roles of 'Orient' and 'Decide' in the OODA loop. WP plays the role of 'Act'. Under the command of CCP and the guidance information of the target provided by SP, WP performs the task of interception. Information is exchanged continuously between different platforms and forms a complex feedback control chain. Fig. 1 shows a typical informational combat scenario. The platforms are organized in a distributed network, and the information of sub-platforms is shared among the network centers. The platforms can exchange information with a network center to obtain the overall battlefield situation; thus, any platform is capable of forming effective cooperation with others.

Under the background of network-centric warfare, uncertainty is an important characteristic, and it exists throughout the combat process. In general, the uncertainty in CCS is mainly from two layers:

(1) **Bottom execution layer**. In this layer, the SP and WP execute the detection and interception tasks, respectively. The actual performance of the platforms is influenced by many factors, such as ambient noise, target interference, and stability of the platform itself. The effect of execution may be varied as we have predicted, and this causes uncertainty. The

* Corresponding author at: School of Automation, Beijing Institute of Technology, Beijing, 100081, China.
E-mail addresses: 2120170967@bit.edu.cn (W. Xu), xiaofan@bit.edu.cn (C. Chen), jackietindd@gmail.com (S. Ding), pardalos@ufl.edu (P.M. Pardalos).

https://doi.org/10.1016/j.eswa.2019.112844
0957-4174/

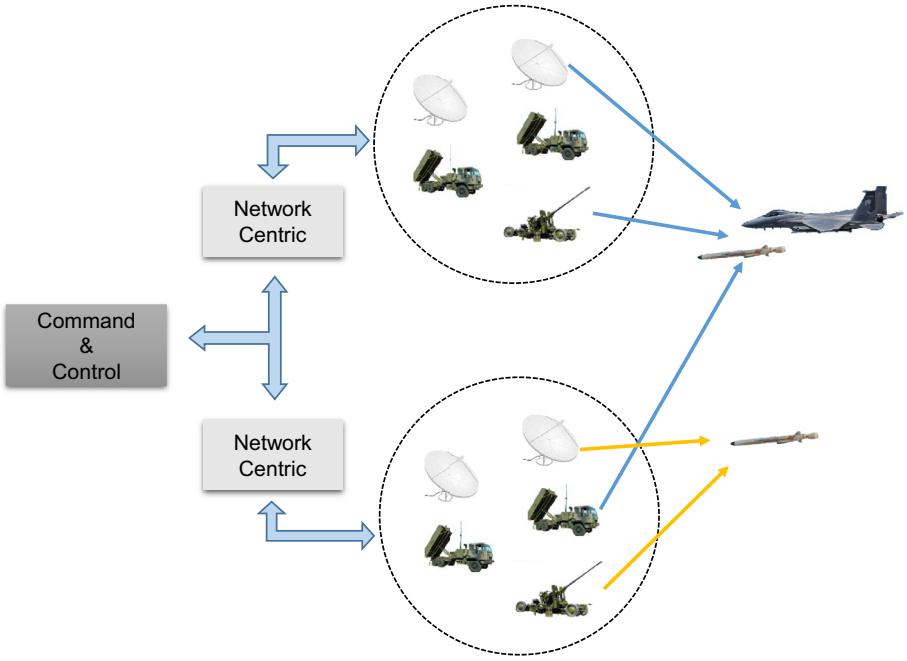

Fig. 1. Typical informational combat scenario.

uncertainty of the SP is manifested in the capturing probability, tracking accuracy, etc. The WP is uncertain mainly in the interception probability.

(2) **Situational decision layer**. The situational decision layer fuses the information from the bottom execution layer to form a battlefield situation and performs the task assignment based on it. On the one hand, the information from the bottom execution layer is nondeterministic, inconsistent, and incomplete; therefore, uncertainty should be an important feature of fused battlefield situations. On the other hand, when making assignment decisions, the trails and intents of the enemy should be inferred. Due to the intelligence and synergy of the enemy, the accuracy of our inferred results can be affected, thus introducing uncertainty.

The aim of this paper is to solve the dynamic task assignment under uncertainty in a defensive scenario. The main contributions of this paper can be summarized as follows. Firstly, a bi-objective dynamic collaborative task assignment model under uncertainty is formulated, which has considered the cooperation between SPs and WPs. Secondly, a novel multi-objective constructive heuristic based on efficiency cost ratio is proposed. The infeasible quaternions are deleted based on rules, and the crowding-distance-based deleting of heuristic individuals maintains the diversity of population. Thirdly, several modifications are made on MOEA/D to enhance the searching performance during the evolutionary process. Finally, the Taguchi method with a novel response metric is applied to calibrate the parameters.

The outline of the paper is as follows: Section 2 will briefly review previous work. In Section 3, we formulate the problem. Section 4 presents the proposed solution algorithm, MMOEA/D-Heuristic. Some numerical experiment is carried in Section 5. The conclusion and future work are presented in Section 6.

2. Related work

The problem of multi-platform collaborative task assignment has been studied since the 1950s, and this problem is termed as the classical weapon target assignment (WTA). Lloyd proved that WTA is an NP-hard problem (Lloyd & Witsenhausen, 1986). Recently, Kline, Ahner, and Hill (2018) reviewed the research on WTA. Most of the researchers have neglected the influence of SP on WP. This is applicable when the SP resources are sufficient. However, the scale of the battlefield keeps expanding, and the generic SP, which is independent of WP, appears, thus making it necessary to combine SP and WP by performing their task assignment in a collaborative manner. Bogdanowicz (2007) developed a model seeking to maximize the sum of benefits of assigning each sensor to each target and each weapon to each target. Jian and Chen (2015) modeled the damage probability of an interceptor as the probability that a sensor will identify the missile and the destructive capacity of the paired weapon. Xin, Wang, and Chen (2018) modeled the probability of successful engagement as the product of the interceptor's probability of kill and the sensor's probability of detection.

The collaborative task assignment consists of two versions: static and dynamic. It was firstly proposed by Hosein and Athans (1990b) during his research on WTA. The static version usually refers to the models that finish the assignment in a single step, while the dynamic version considers the dynamic adjustment during the assignment process, which can be considered as a repetition of the OODA loop. Previous studies have focused on the static version (Ahuja, Kumar, Jha, & Orlin, 2007; Hosein & Athans, 1990a; Kwon, Kang, Lee, & Park, 1999; Kwon, Lee, Kang, & Park, 2007; Manne, 1958; Shang, Zaiyue, Xiaoru, & Cungen, 2007); however, recently, dynamic models have attracted considerable attention (Ahner & Parson, 2015; Hosein & Athans, 1990b; Khosla, 2001; Li, Chen, & Xin, 2015; Li, Chen, Xin, Dou, & Peng, 2016; Xin, Chen, Zhang, Dou, & Peng, 2010). A dynamic assignment model is more than just multiple repetitions of a static assignment model. It considers the time window and aims to find the global optimal assignment scheme. The introduction of time window makes this problem much more complicated. Khosla (2001) considered the time window in the study of firepower allocation, and he took launch time as a variable to be optimized. A hybrid genetic algorithm combining the simulated annealing was proposed to solve this problem. However, the time window was divided into very short pieces, resulting in large optimization spaces, and hence it is not suitable for practical use. The multi-stage model is another

way to consider the time window. Hosein and Athans (1990b) conducted early research and proposed a two-stage dynamic assignment model. Xin et al. (2010) proposed a multi-stage asset-based assignment model. It divides the combat process into several independent combat stages and optimizes the remaining stages. This kind of dynamic model is close to reality and simpler than the model proposed by Khosla.

Robust and stochastic optimization is a technology developed to reduce the influence of uncertainty. It has been applied in some task assignment problems. The uncertainty of the interception probability in WTA was studied by Krokhmal, Murphey, Pardalos, Uryasev, and Zrazhevski (2003) utilizing the CVaR constraint. The uncertainty of the target number was modeled as a two-stage stochastic assignment process. Ahner and Parson (2015) considered the possible number of targets in the second stage, and the expected value of the second stage was included in the objective. A robust optimization model for the uncertainty of interception probability was proposed in Li et al. (2016). Most of these robust assignment models are scenario-based. The robustness relies on the number of scenarios, which result in more computational burden.

As for the study of MOEAs, Schaffer (1985) designed VEGA during the mid-1980s and it is considered to be the first MOEA. After that, many representative MOEAs have been proposed. NSGA-II (Deb, Agrawal, Pratap, & Meyarivan, 2000) proposed by Deb is one of the most successful MOEAs that use fast non-dominated sorting and the crowding distance. Recently, a branch of MOEA based on decomposition has become increasingly popular. It decomposes a multi-objective problem into a set of scale subproblems and optimizes them simultaneously. Zhang and Li (2007) proposed the earliest version of MOEA/D. Many researchers have focused on (1) decomposition method (Cai, Mei, Fan, & Zhang, 2018; Chen, Li, & Xin, 2017; Yang, Li, Liu, & Zheng, 2013; Zhang, Li, Maringer, & Tsang, 2010), (2) adaptive mechanism (Chiang & Lai, 2011; Qi et al., 2014; Wang, Zhang, Zhou, Gong, & Jiao, 2016; Zhang, Liu, & Li, 2009), and (3) combination with other state-of-art algorithms or local search methods (Alhindi & Zhang, 2014; Ke, Zhang, & Battiti, 2013; Tan, Jiao, Li, & Wang, 2012; Wang & Cai, 2015; Zapotecas-Martínez et al., 2015). Some researchers have used MOEAs to solve task assignment problems (Li et al., 2015; Li, Kou, & Li, 2018; Li, Kou, Li, Xu, & Chang, 2017).

In MOEAs, many parameters are involved. The performance of the algorithm is very sensitive to these parameters, and they need be fined-tuned for optimal performance. Xu and Mei (2018) did a full factorial experiment to the sensitive analysis of parameters. However, It is an onerous work for researchers to test all the combinations of parameters. Taguchi method is a fractional factorial experiment, which was proposed by Taguchi as an efficient alternative for the full factorial experiment. It has been a common approach for calibration of MOEA recently (Ding, Chen, Xin, & Pardalos, 2018; Fattahi, Hajipour, & Nobari, 2015; Mousavi, Sadeghi, Niaki, & Tavana, 2016).

Due to the difficulty of efficiently solving large scale task assignment problems, many heuristic algorithms have been designed. Madni and Andrecut (2012) presented two heuristic algorithms, namely simulated annealing and threshold accepting, to solve WTA. They were compared with the solution of relaxed WTA using maximal marginal return algorithm. Ahuja et al. (2007) proposed a network-flow-based construction heuristic and a VLSN search algorithm to solve the WTA problem. Xin et al. (2010) designed a virtual permutation and tabu search heuristics. Xin et al. (2018) proposed a marginal-return-based constructive heuristic to solve the sensor-weapon-target assignment problem. Chang, Kong, Hao, Xu, and Yang (2018) used an improved artificial bee colony algorithm with a heuristic factor initialization to solve WTA.

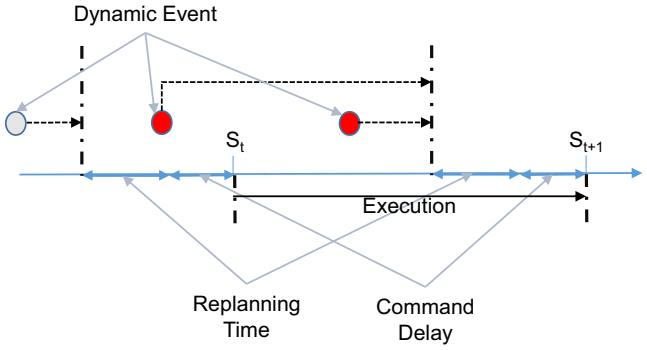

Fig. 2. Dynamic process of collaborative task assignment.

3. Problem formulation

The combat scenario considered in this paper is narrated as follows. The defender has Q sensor platforms and W weapon platforms to defend an asset. They are connected by networks and can combine with each other to finish a task collaboratively. At a certain time, T incoming targets appear and aim at the asset. Before the enemy finishes the attack, the defender has a time interval to intercept these targets, which can be divided into several stages with a fixed length. A stage is the minimum combat time unit. Assume that there are S stages. As shown in Fig. 2, S_t and S_{t+1} are the beginning times of stages t and $t+1$, respectively. During stage t, the WPs and SPs accept the assignment scheme from CCP at S_t, and execute their task from S_t to S_{t+1}. The CCP should monitor dynamic events, such as new targets appearing and old targets being destroyed, and adjust its action accordingly. There is a certain time delay between the occurrence of dynamic events and the response of WPs and SPs. The red points represent the dynamic events that were responded to in stage $t+1$, and the gray point represents the event that was responded to in the previous stage. CCP collectively processes the dynamic events before a certain time of the next stage and sends the command to the execution platforms. From the figure, it can be seen that the factors that affect the time delay of the response to the dynamic events are the replanning time, the command delay, and the length of each stage.

The scheme of WPs is denoted by $X = [x_{sik}]_{S\times W\times T}$, where $x_{sik} = 1$ if the ith WP is allocated to the kth target at stage s, and 0 otherwise. The scheme of SPs is denoted by $Y = [y_{sjk}]_{S\times Q\times T}$, where $y_{sjk} = 1$ if the jth SP is allocated to the kth target at stage k, and 0 otherwise. For ease of reading, Table 1 summarizes all the notations that are used throughout this paper.

3.1. Deterministic model

(1) Objective

$q_{jk}(s)$ is used to denote the tracking performance of the jth SP to the kth target at stage s, and $p_{ik}(s)$ is used to denote the probability that the kth target is destroyed by the ith WP at stage s under the condition that the kth target gets a high quality of tracking. Thus, the tracking performance to the kth target at stage s is

$$Q_k(s) = 1 - \prod_{j=1}^{Q}(1 - q_{jk}(s))^{y_{sjk}}. \tag{1}$$

The destroying probability of the kth target at stage s with high quality of tracking is

$$P_k(s) = 1 - \prod_{i=1}^{W}(1 - p_{ik}(s))^{x_{sik}}. \tag{2}$$

Table 1
Notations.

S, W, Q, T	the number of stages, WPs, SPs and targets respectively
$q_{jk}(s)$	the tracking performance of jth SP to kth target at stage s
$p_{ik}(s)$	the destroying probability of ith WP to kth target at stage s under effective guidance.
$Q_k(s)$	the comprehensive tracking performance to kth target at stage s
$P_k(s)$	the comprehensive destroying probability to kth target at stage s under effective guidance.
v_k	the threat value of kth target
c_i	the cost of ith WP when operating in a single stage
d_j	the cost of jth SP when operating in a single stage
f_{sik}	$f_{sik} = 1$ if kth target can assigned to ith WP at stage s, 0 otherwise
f_{sjk}	$f_{sjk} = 1$ if kth target can assigned to jth SP at stage s, 0 otherwise
n_i	the maximum number of targets that can be assigned to ith WP at each stage.
m_j	the maximum number of targets that can be assigned to jth SP at each stage.
F_i	the amount of remaining ammunition of ith WP
N_{sik}	the required number of continuous stages of guidance to target k when it is assigned to ith weapon at stage s
γ_{iks}	the uncertain degree of destroying probability for ith WP to intercept kth target at stage s
γ_{jks}	the uncertain degree of tracking performance for jth SP to track kth target at stage s
ξ	the uncertain parameter which is non-determinate before actual happening
σ	the robust regulatory factor
$X = [x_{sik}]$	the decision matrix of WPs. $x_{sik} = 1$ when target k is assigned to ith WP at stage s, 0 otherwise
$Y = [y_{sjk}]$	the decision matrix of SPs. $y_{sjk} = 1$ when target k is assigned to jth SP at stage s, 0 otherwise

Influenced by the tracking performance, the destroying probability of the kth target at stage s is $P_k(s)Q_k(s)$. The threat value of the kth target is denoted by v_k, which is mainly composed of the capability threat and the intent threat and can be evaluated according to the battlefield situation. Thus, the total threat elimination from stage t to the final stage S is

$$F_1(t) = \sum_{k=1}^{T} v_k \left(1 - \prod_{s=t}^{S} (1 - P_k(s)Q_k(s)) \right). \tag{3}$$

Assume that each platform possesses only one type of combat resource. The ith WP and the jth SP cost c_i and d_j to operate in a single stage, respectively. Thus, the total cost from stage t to the final stage S is given by

$$F_2(t) = \sum_{s=t}^{S} \sum_{k=1}^{T} \left(\sum_{i=1}^{W} c_i x_{sik} + \sum_{j=1}^{Q} d_j y_{sjk} \right). \tag{4}$$

Through the above analysis, we get two optimization objectives:

$$\max F_1(t), \min F_2(t). \tag{5}$$

(2) Constraints

$$x_{sik} \le f_{sik} \ \forall s \in I_S, i \in I_W, k \in I_T, \tag{6}$$

$$y_{sjk} \le f_{sjk} \ \forall s \in I_S, j \in I_Q, k \in I_T, \tag{7}$$

$$\sum_{k=1}^{T} x_{sik} \le n_i \ \forall s \in I_S, i \in I_W, \tag{8}$$

$$\sum_{k=1}^{T} y_{sjk} \le m_j \ \forall s \in I_S, j \in I_Q, \tag{9}$$

$$\sum_{s=t}^{S} \sum_{k=1}^{T} x_{sik} \le F_i \ \forall i \in I_W, \tag{10}$$

$$\sum_{t=s}^{\min(S, s+N_{sik}-1)} I\left(\sum_{j=1}^{Q} y_{tjk} > 0 \right) = N_{sik} x_{sik}, \quad \forall s \in I_S, i \in I_W, \quad k \in I_T, \tag{11}$$

$$I_W = \{1, 2, \ldots, W\}, I_Q = \{1, 2, \ldots, Q\},$$

$$I_T = \{1, 2, \ldots, T\}, I_S = \{1, 2, \ldots, S\}.$$

The constraint sets (6) and (7) represent the feasibility of WP and SP. Here, $f_{sik} = 1$ if the ith WP can engage with the kth target at stage s, and 0 otherwise. $f_{sjk} = 1$ if the jth SP can track the kth target at stage s, and 0 otherwise. The constraint sets (8) and (9) limit the maximum number of targets WP and SP can engage or track in a single stage respectively. It depends on the number of channels they have. A platform with multiple channels can be regarded as multi platforms with a single channel; therefore, n_i and m_j are set to 1 in this paper. The constraint set (10) consists of the ammunition constraints. F_i is the remaining ammunition of the ith WP. The constraint set (11) is the continuous guidance constraint set, which shows that when the WP is intercepting a target, we should guarantee several stages of tracking or illumination to the target in order to observe whether it is destroyed or to guide weapon towards it. N_{sik} is the number of stages that the kth target needs to be tracked since stage k, if it is assigned to the ith WP at this stage. N_{sik} is determined by the time interval between the launching of a weapon and the interception with the aiming target. Function $I(\cdot) = 1$ if the inequality in the parentheses is true, and 0 otherwise. In most studies, the tracking task of the SP and the launching task of the WP are thought to be completed in a single stage. However, the launching time of the WP is relatively fixed while the tracking time of the SP varies significantly, especially for missile weapons with a wide range of defenses, and the relay guidance is needed sometimes.

(3) Model

The deterministic model is given as follows:

$$\begin{aligned} &\max \ F_1(t), \min F_2(t), \\ &\text{s.t.} \ \ (6), (7), (8), (9), (10), (11). \end{aligned} \tag{12}$$

Remark 1. The proposed model divides the defense time interval into several fixed stages. The length of each stage is flexible and can be determined according to the actual situation. A minimum length is required to ensure that the platforms can operate effectively at each stage. With the increase in the stage length, the complexity of our model decreases, and when the length of a stage is equal to the defense time interval, the model becomes a static version of the assignment model.

Remark 2. The above model holds an important assumption. That is, the destroying probability is not influenced by the SPs used to track or illuminate in the following stages after the weapon is launched. This is reasonable since missiles usually have terminal active guidance or the purpose of observation does not require good tracking performance.

3.2. Robust model

The traditional collaborative task assignment model requires the parameters to be accurately known. However, due to the interference effects of many factors in the real world, errors are inevitable from both modeling and parameter acquisition, and the deterministic assumption is no longer valid. Therefore, this paper considers a robust model, which mainly focuses on the uncertainty from the bottom execution layer as mentioned before.

The execution uncertainty of WPs and SPs is influenced by many kinds of stochastic or non-stochastic factors. Rapid changing environments such as wind and humidity are the main stochastic factors. Different targets have different maneuverability and reflection cross-sectional areas, which makes the execution performance deviate from the theoretical value. The target and the execution platform inherently determine this deviation, which is a kind of non-stochastic factor. An assumption on execution uncertainty is made here: The real performance of the SP and WP occur within a certain range of theoretical value due to uncertainty factors.

$$p_{ik}(s,\xi) \in [(1-\gamma_{iks})p_{ik}(s), (1+\gamma_{iks})p_{ik}(s)],$$

$$q_{jk}(s,\xi) \in [(1-\gamma_{jks})q_{jk}(s), (1+\gamma_{jks})q_{jk}(s)],$$

where γ_{iks} represents the degree of uncertainty for the ith WP to intercept target k at stage s. γ_{jks} represents the degree of uncertainty for the jth SP to track target k at stage s. The range can be estimated from historical data, current environmental condition, etc., and the basic idea of robust optimization techniques is making the best decision under the worst condition. Thus, the objective function F_1 with uncertain parameters becomes:

$$F_1'(t) = \min_{\xi} \sum_{k=1}^{T} v_k \left(1 - \prod_{s=t}^{S} (1 - P_k(s,\xi)Q_k(s,\xi))\right), \tag{13}$$

$$P_k(s,\xi) = 1 - \prod_{i=1}^{W} (1 - p_{ik}(s,\xi))^{x_{sik}}, \tag{14}$$

$$Q_k(s,\xi) = 1 - \prod_{j=1}^{Q} (1 - q_{jk}(s,\xi))^{y_{sjk}}, \tag{15}$$

The function above is scenario-based. A good approximation of the uncertainty requires a large number of scenarios, which increase the computational burden. A Soyster robust (Soyster, 1973) model was first proposed to solve the box uncertain interval in uncertain linear programming. A modified Soyster model is given in Bertuccelli, Alighanbari, and How (2004), which is more flexible than the original. By utilizing this, F_1 becomes

$$F_1''(t) = \sum_{k=1}^{T} v_k \left(1 - \prod_{s=t}^{S} \left(1 - P_k''(s)Q_k''(s)\right)\right), \tag{16}$$

$$P_k''(s) = 1 - \prod_{i=1}^{W} (1 - (1-\sigma\gamma_{iks})p_{ik}(s))^{x_{sik}}, \tag{17}$$

$$Q_k''(s) = 1 - \prod_{j=1}^{Q} (1 - (1-\sigma\gamma_{jks})q_{jk}(s))^{y_{sjk}}, \tag{18}$$

where $\sigma \in [0, 1]$ is the robust regulatory factor. The proposed robust model is given as follows:

$$\max F_1''(t), \min F_2(t), \text{s.t. } (6), (7), (8), (9), (10), (11). \tag{19}$$

Remark 3. A necessary condition of the proposed model is that the objective function F_1 is monotonic with respect to each parameter. The model becomes more flexible with parameter σ. When $\sigma = 1$, the proposed model is equivalent to the max-min robust model, and when $\sigma = 0$, it becomes a deterministic model. As σ increases from 0 to 1, the robustness is generally enhanced with the price of decreasing expectation.

3.3. NP-Hardness

Theorem 1. *The proposed bi-objective problem is NP-hard.*

Proof. The proof details are given in Appendix A. □

4. MOEA/D optimizer

Since the problem is NP-hard, there is no polynomial-time algorithm to obtain the exact solution. MOEA/D (Zhang & Li, 2007), proposed by Zhang et al., has attracted considerable attention recently. This framework aims to decompose a multi-objective problem into a set of scalar subproblems and optimizes them simultaneously. MOEA/D offers a clear and extensible framework that utilizes the neighbor structure. Hence, we mainly focus on studying this framework. It should be noted that the solving efficiency is very important in a real battlefield situation. For this purpose, several efforts are made to enhance the performance of MOEA/D. First, a solution representation based on the coding table is designed. Second, a multi-objective constructive heuristic initialization based on efficiency-cost ratio is proposed. Third, two main modifications are made on the general MOEA/D framework, which are the nadir-based Tchebycheff approaches and the neighbor matching strategy (NMS), to enhance the process of generating and selecting of offspring. The framework of the modified MOEA/D is given in Algorithm 1. We will discuss in detail in the following subsections.

4.1. Solution representation

Decimal coding is employed in this paper. We apply a bit more effort and ensure that the solution satisfies the constraint sets (6)–(9) naturally. To achieve this, two coding tables are first built based on the model parameters.

Weapon-stage table (WS): $WS_{is} = \{t_1, t_2, \ldots t_{l_{is}}\}, i \in I_W, s \in I_S$ enumerates all the targets that can be assigned to the ith WP at stage s.

Sensor-stage table (QS): $QS_{js} = \{t_1, t_2, \ldots t_{l_{js}}\}, j \in I_Q, s \in I_S$ enumerates all the targets that can be assigned to the jth SP at stage s.

The conditions that restrict a target from being assigned to a platform can be summarized as follows:

Condition 1: Both the WS table and the QS table should satisfy the feasibility constraint sets (6) and (7). For the WS table, target k can be included in WS_{is} only when $f_{sik} = 1$. For the QS table, target k can be included in QS_{js} only when $f_{sjk} = 1$.

Condition 2: The WS table should satisfy the ammunition constraint set (10). When $F_i = 0$, no target will be included in WS_{is} for any stage s.

Condition 3: We should guarantee the feasibility of the continuous guidance constraint before a target is added to the WS table. $Tr(s, k)$ is used to denote whether there is any sensor platform that

Algorithm 1 The framework of the modified MOEA/D .

Initialize the Population $Pop = \{I_1, \ldots, I_P\}$ using Algorithm 4, where P is the population size. Evaluate the fitness $F = \{F_1, \ldots, F_P\}$ of Pop.
Generate P evenly distributed weight vectors $\lambda = \{\lambda_1, \ldots, \lambda_P\}$. Obtain T neighbors of each subproblem using the Euclidean distances between weight vectors. Denote by $B(i) = \{i_1, \ldots, i_T\}$ the neighbor set of the ith subproblem.
$z_i = \min_j(F_j^i), z_i^{nad} = \max_j(F_j^i), i = 1, \ldots, m.$
Initialize R_{ij} and Inc_{ij}.
while maximum generation G is not reached **do**
  $R_{ij} = R_{ij} Inc_{ij}, i = 1, \ldots, N, j = 1, \ldots, N.$
  for each subproblem i **do**
    Two parents p_1, p_2 are selected by Algorithm 5.
    An offspring o is generated using Algorithm 6. And we evaluate its fitness f.
    $z_i = \min(z_i, f_i), z_i^{nad} = max(z_i^{nad}, f_i), i = 1, \ldots, m$
    For all individuals, $\tilde{F}^i = \frac{F^i - z_i}{z_i^{nad} - z_i}, i = 1, \ldots, m.$
    for each subproblem $j \in B(i)$ **do**
      if $g^{nte}(\tilde{F}_j|\lambda_j, 0) < g^{nte}(\tilde{f}|\lambda_j, 0)$ **then**
        $I_j = o, F_j = f.$
      end if
    end for
  end for
end while
Output the final population P and its Fitness F.

can track target k at stage s.

$$Tr(s,k) = I\left(\sum_{j=1}^{Q} f_{sjk}\right). \tag{20}$$

Target k is an element of WS_{is} table when the following equation holds.

$$\sum_{t=s}^{\min(S, s+N_{sik}-1)} Tr(t,k) = N_{sik}. \tag{21}$$

Based on the above three conditions, WS and QS tables are built according to Algorithm 2.

Algorithm 2 Build coding table.

Input: $\{f_{sik}\}$, $\{f_{sjk}\}$, $\{F_i\}$, $\{N_{sik}\}$.
Output: WS, QS.
Calculate Tr by Eq. (20).
Set $WS_{is} = \emptyset$ and $QS_{js} = \emptyset$ for all i, j, k.
for each quaternion (s, i, j, k) **do**
  if (s, i, k) satisfy Condition 1, 2, and 3 **then**
    $WS_{is} = WS_{is} \cup k.$
  end if
  if (s, j, k) satisfy Condition 1 **then**
    $QS_{js} = QS_{js} \cup k.$
  end if
end for

The WS table and QS table eliminate the impossible allocation cases to the maximum extent. In an actual combat situation, most targets cannot be allocated to a platform in all stages. Therefore, the searching space is limited.

By utilizing the coding tables, the solution representation is presented in Fig. 3. The value at each locus represents the index of the target at the corresponding coding table, and its value ranges from 0 to the size of the coding table. As shown in Fig. 3, the WS table corresponding to the Wth weapon platform at stage 1 is {1, 3, 6, 7}. Thus, at this locus, the integer value varies from 0 to $|WS_{W1}| = 4$. If this value is 2, the assigned target can be found by looking up WS_{W1} and it is found that target 3 is assigned to the Wth weapon platform at stage 1. If the value at each locus is 0, then no target is assigned. In this way, the constraint sets (6)–(9) are always satisfied, and some impossible allocation case is avoided.

4.2. Heuristic initialization based on efficiency-cost ratio

The initial population is very important for any swarm optimization algorithm. Both quality and diversity should be guaranteed. An efficient way to initialize the population is by utilizing the domain knowledge. Most of the current researches on heuristic strategies for solving task assignment problems focus on a single objective. In Xin et al. (2018), a marginal-return-based constructive heuristic is proposed to solve a static sensor-weapon-target assignment problem, and it obtains very good performance. We extend this framework to a multi-objective version and propose a novel heuristic initialization strategy to construct a hybrid population.

For simplicity, we neglect the continuous guidance constraint when constructing an initial heuristic population. It will be fixed further using a repairing technique introduced in the following subsection. In this manner, at a certain stage, it will be pointless to assign one type of platform to a target. When only one type of platform, SP or WP, is assigned to target k at stage s, then we have $P_k(s) = 0$ or $Q_k(s) = 0$; thus, $P_k(s)Q_k(s) = 0$. The assigned platform has no contribution to the improvement of the total efficiency but adds to the costs. In view of these facts, the quaternion (s, i, j, k) is introduced to show that weapon platform i and sensor platform j are assigned to target k at stage s simultaneously if it is added to the assignment scheme X and Y. All the possible quaternions will be enumerated and considered. In this way, the case that only one type of platform is assigned to a target can be avoided. We denote the set of all available quaternions as AQS. We can further initialize AQS using the WS table and QS table shown in Algorithm 3.

Algorithm 3 Initialize AQS.

Input: WS, QS.
Output: AQS.
$AQS = \emptyset.$
for each quaternion (s, i, j, k) **do**
  if k is in WS_{is} and QS_{js} **then**
    $AQS = AQS \cup (s, i, j, k).$
  end if
end for

The basic idea of our proposed heuristic initialization strategy is to add these quaternions into an empty assignment schemes X and Y successively in order to iteratively construct a limited number of individuals for the initial population. There are two main problems to be solved.

(1) How to select a quaternion from AQS to add it into the assignment scheme and then delete the invalid quaternions in AQS to accelerate the process.
(2) If the number of constructed individuals exceeds our limitation, how to delete some of them while maintaining diversity.

A direct method for solving problem (1) is to select one quaternion with the optimal efficiency-cost ratio (ECR). For the empty assignment schemes X and Y, where all elements x_{sik} and y_{sjk} are set to 0, and AQS, we denote $P_{m,k}(s) = 1 - P_k(s)$, $Q_{m,k}(s) = 1 - Q_k(s)$. It is obvious that $P_{m,k}(s) = 1$ and $Q_{m,k}(s) = 1$ initially. If quaternion $(s, i, j, k) \in AQS$ is to be added into current assignment schemes

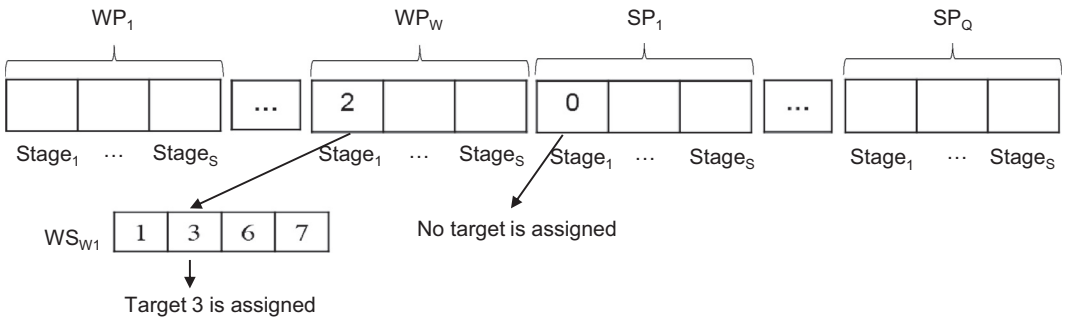

Fig. 3. Solution representation.

X and Y, then $P_{m,k}(s)$ and $Q_{m,k}(s)$ will become

$$P'_{m,k}(s) = P_{m,k}(s)(1 - p_{ik}(s))^{1-x_{sik}}, \tag{22}$$

$$Q'_{m,k}(s) = Q_{m,k}(s)(1 - q_{jk}(s))^{1-y_{sjk}}. \tag{23}$$

It is clear that there is a gain for both the objectives of the model after this quaternion is added. The gain is termed as marginal return in Xin et al. (2018). We also accept this concept in this paper, and the efficiency-cost ratio δ_{sijk} of this quaternion can be given as

$$\delta_{sijk} = \frac{MR(F_1)}{MR(F_2)}, \tag{24}$$

where $MR(F_1)$ and $MR(F_2)$ are the marginal returns of the first and second objectives, respectively. They are calculated using Eq. (25) and Eq. (26).

$$MR(F_1) = v_k\left(\prod_{s=t}^{S}\left(1-(1-P_{m,k}(s))(1-Q_{m,k}(s))\right) - \prod_{s=t}^{S}\left(1-\left(1-P'_{m,k}(s)\right)\left(1-Q'_{m,k}(s)\right)\right)\right), \tag{25}$$

$$MR(F_2) = c_i(1 - x_{sik}) + d_j(1 - y_{sjk}). \tag{26}$$

This ratio represents the marginal return of unit cost for each quaternion. We obtain the efficiency-cost ratios of all quaternions in AQS. Then the optimal quaternion is selected as follows:

$$(s^*, i^*, j^*, k^*) = \arg\max_{(s,i,j,k)\in AQS}\{\delta_{sijk}\}. \tag{27}$$

This selected quaternion is added to the assignment schemes X and Y. We realize this by setting $x_{s^*i^*k^*} = 1$ and $y_{s^*j^*k^*} = 1$. After the optimal quaternion is added, some quaternions in AQS will become infeasible considering the constraints. We should find out and delete them to accelerate our algorithm. Generally, We have the following three rules to update AQS after (s^*, i^*, j^*, k^*) is added.

Rule 1: Weapon platform i^* cannot be assigned to any other target except k^* at stage s^*.

Rule 2: Sensor platform j^* cannot be assigned to any other target except k^* at stage s^*.

Rule 3: The remaining ammunition of the i^*th weapon platform is denoted as F'_{i^*}. If $F'_{i^*} = 0$, it cannot be assigned to any other target.

The process of adding a quaternion and updating AQS loops until AQS becomes empty. For the given assignment schemes X, Y and the updated AQS, the assignment scheme after $(s, i, j, k) \in AQS$ is denoted as $\{X, Y\}^{(s,i,j,k)}$. $\{X, Y\}^{(s^*,i^*,j^*,k^*)}$ is always non-dominated compared to other $\{X, Y\}^{(s,i,j,k)}$. This property guarantees that adding the quaternion with the maximum efficiency-cost ratio will generate a local optimal scheme in our multi-objective problem.

It should be noted that when calculating the marginal return of the two objectives, we do not have to calculate the whole objective value of the assignment scheme before and after a quaternion is added. We need to calculate only the incremental part. $P_{m,k}(s)$ and $Q_{m,k}(s)$ will be updated when a new quaternion is added. Therefore, the previously added quaternions do not have to be recalculated. These designs reduce the computational burden needed to calculate the efficiency-cost ratios and speed up the initialization process.

Each time a new quaternion is added, a new scheme with different costs is generated. We can convert it into the form of solution representation and add it into the heuristic population. If the size of the heuristic population exceeds our limit, then problem (2) occurs. To maintain diversity, we borrow the concept of crowding distance in NSGA-II, where the position of each individual is defined as the number of added quaternions in the scheme. The closer these numbers are, the more similar they are. We delete the individual with minimum crowding distance to keep the size of the heuristic population within our limit. The details of constructing the initial population, which is a hybrid of the heuristic and the randomly generated population, are given in Algorithm 4.

The time complexity of the heuristic initialization at each loop mainly depends on the size of AQS, which is L. The worst time complexity happens when the WPs have sufficient ammunition and the feasibility constraints are always satisfied. Thus, at the beginning, the size of AQS is $(S \cdot W \cdot Q \cdot T)$. Since AQS is determined, the time complexity mainly depends on the number of deleted quaternions in AQS in each loop. The less we delete, the more repetitions will run. There are $(W \cdot Q \cdot T)$ quaternions related to stage s. If the first quaternion related to stage s is added, $A_1 = Q(T - 1) + (W - 1)(T - 1) + 1$ quaternions will be deleted from AQS. The minimum number of deletions for the following $(Q - 1)$ times of the loop is $A_2 = (W - 1)(T - 1) + 1$. For the remaining $(W - 1)Q$ quaternions related to stage s, we can only delete one in each loop. Thus, the maximum number of loops is $(S \cdot W \cdot Q)$. In the worst condition, L at mth loop can be calculated as follows:

$$L = \begin{cases} \alpha & \beta = 0, \\ \alpha - A_1 & \beta = 1, \\ \alpha - A_1 - A_2(\beta - 1) & \beta = 2, \ldots, Q, \\ \alpha - A_1 - A_2(Q - 1) - \beta - Q & \beta = Q + 1, \ldots, W \cdot Q - 1, \end{cases}$$

$$\alpha = (S - mod(m - 1, W \cdot Q)) \cdot W \cdot Q \cdot T,$$

$$\beta = (m - 1)/(W \cdot Q).$$

By summing up L of all loops, the time complexity of the heuristic initialization is $O(S^2W^2Q^2T)$. Actually, the worst condition is unlikely to happen, and the algorithm can achieve better performance than the theoretical value.

Algorithm 4 Heuristic initialization based on ECR.

Input: WS, QS, $\{F_i\}$, $\{p_{ik}(s)\}$, $\{q_{jk}(s)\}$, Maximum size of heuristic population (U), Size of hybrid population (P).
Output: Hybrid population (HP).

$HP = \emptyset$, $Position = \emptyset$.
Initialize AQS using Algorithm 3.
$x_{sik} = 0, y_{sjk} = 0$, for all s, i, j, k.
$sol = zeros(1, S \cdot (W + Q))$//an empty solution represented by a one-row zero vector.
$count = 0$.
$L = |AQS|$.
$F_i^{'} = F_i, i = 1, \ldots, W$.
$P_{m,k}(s) = 1, Q_{m,k}(s) = 1$, $k = 1, \ldots T$, $s = 1, \ldots, S$.
while $L > 0$ **do**
  for l from 1 to L **do**
    $(s_l, i_l, j_l, k_l) = AQS(l, :)$.
    $\delta_{s_l i_l j_l k_l} = MR(F_1)/MR(F_2)$.
  end for
  $l^* = \arg\max_l \{\delta_{s_l i_l j_l k_l}\}$.
  (s^*, i^*, j^*, k^*) = the l^*th quaternion in AQS.
  $F_{i^*}^{'} = F_{i^*}^{'} - (1 - x_{s^* i^* k^*})$.
  for l from 1 to L **do**
    $(s_l, i_l, j_l, k_l) = AQS(l, :)$.
    if $s_l = s^*$ and $i_l = i^*$ and $k_l \neq k^*$ **then**
      delete the lth quaternion in AQS(Rule 1).
    else if $s_l = s^*$ and $j_l = j^*$ and $k_l \neq k^*$ **then**
      delete the lth quaternion in AQS(Rule 2).
    else if $F_{i_l}^{'} = 0$ and $x_{s_l i_l k_l} = 0$ **then**
      delete the lth quaternion in AQS(Rule 3).
    end if
  end for
  Delete the l^*th quaternion in AQS.
  $L = |AQS|$.
  $P_{m,k^*}(s^*) = P_{m,k^*}(s^*)(1 - p_{i^* k^*}(s^*))^{1 - x_{s^* i^* k^*}}$.
  $Q_{m,k^*}(s^*) = Q_{m,k^*}(s^*)(1 - q_{j^* k^*}(s^*))^{1 - y_{s^* j^* k^*}}$.
  $x_{s^* i^* k^*} = 1, y_{s^* j^* k^*} = 1$.
  $count = count + 1$.
  Add (s^*, i^*, j^*, k^*) into sol to construct a new solution.
  $HP(end + 1, :) = sol$.
  $Position(end + 1) = count$.
  if $|HP| > U$ **then**
    $CD(1) = Inf, CD(|HP|) = Inf$.
    for n from 2 to $|HP| - 1$ **do**
      $CD(n) = Position(n + 1) - Position(n - 1)$.
    end for
    $n^* = \arg\min_n CD(n)$.
    delete n^*th element in HP and $Position$.
  end if
end while
Randomly insert $P - |HP|$ individuals that are randomly generated into HP.

4.3. Modification of MOEA/D framework

In this subsection, we modify the general MOEA/D framework with nadir-based Tchebycheff approach and propose a novel neighbor matching strategy for the purpose of generating and selecting more promising individuals during the evolutionary process.

(1) Ideal-based and Nadir-based Tchebycheff approach
As MOEA/D is widely studied, many decomposition methods have been proposed, such as weighted sum, Tchebycheff approach, PBI approach (Zhang & Li, 2007), NPI-style Tchebycheff approach (Zhang et al., 2010), ε-constraint approach (Chen et al., 2017), grid-based approach (Cai et al., 2018), etc. They all have their own advantages and disadvantages and need to be chosen according to the specific problem. Here, we mainly discuss the Tchebycheff approach.
The original Tchebycheff approach is ideal-based, which means that the scalar subproblems are optimized towards the ideal point. Refer (Zhang & Li, 2007) for more details about this approach. There are two main drawbacks of the ideal-based Tchebycheff approach.
  (1) The population is crowded in a small area of the PF for a large searching space. This can be explained by its selection preference as shown in the left part of Fig. 4. The red curve is the true Pareto front. The gray area is the replacement region of the new solution for each subproblem. Point z is a transient ideal point, which is iteratively updated during the evolutionary process. It is apparent that a solution closer to z is preferred. This limits the spread of the population to the boundary of PF. Furthermore, if the weight vectors of subproblems λ_1 and λ_N are set to (0,1) and (1,0), then subproblem λ_1 aims to minimize the ordinate value, while subproblem λ_N aims to minimize the abscissa value. This will cause the solution of subproblem λ_1 to be closer to that of subproblem λ_{N-1}, and the solution of subproblem λ_N to be closer to that of subproblem λ_2. The neighbor structure is destructed in a sense.
  (2) The population does not distribute evenly along a convex PF. When the PF is convex as shown in Fig. 4, the evenly distributed weight vectors will not generate evenly distributed solutions along the Pareto front due to the peaking and tailing at the two sides of the PF. This problem has also been reported in the literature (Cai et al., 2018).

Prior knowledge is that the true PF of our model is convex. For better performance, an inverted form of Tchebycheff proposed in Jiang and Yang (2017) is utilized. In this paper, we term it as nadir-based Tchebycheff (NTE). Unlike the ideal-based Tchebycheff, NTE takes the nadir point as the reference point and the optimization objective of each subproblem becomes far away from this worst case. The ith subproblem is as follows:

$$\max g^{nte}(F|\lambda, z^{nad}) = \min_{1 \le i \le m} \{\lambda^i (z_i^{nad} - F^i)\}, \tag{28}$$

where F is the objective function vector of a solution. Its selection region is shown in the right part of Fig. 4. To avoid abnormal selection, the weight vectors of λ_1 and λ_N are set to (ϵ, 1-ϵ) and (1-ϵ, ϵ), where ϵ is a very small number. Instead of trending to one point, NTE guides the population to all directions, resulting in a better sparsity. Moreover, the solutions are distributed more evenly for convex PF. To avoid the influence of different dimensions, we normalize the objective value.

$$\tilde{F}^i = \frac{z_i^{nad} - F^i}{z_i^{nad} - z_i}, i = 1, \ldots, m, \tag{29}$$

where F_i is the ith objective value, z_i is the ith element of the ideal point z, z_i^{nad} is the ith element of the nadir point z^{nad}, and $\tilde{F}^i$ is the normalized ith objective value. Thus, the optimization goal becomes

$$\max g^{nte}(\tilde{F}|\lambda, 0). \tag{30}$$

(2) Neighbor matching strategy
As an important feature of MOEA/D, the neighbor structure plays a vitally important role in the evolution process. An assumption in MOEA/D is that the solutions of neighbor subproblems are similar to each other. A further consideration

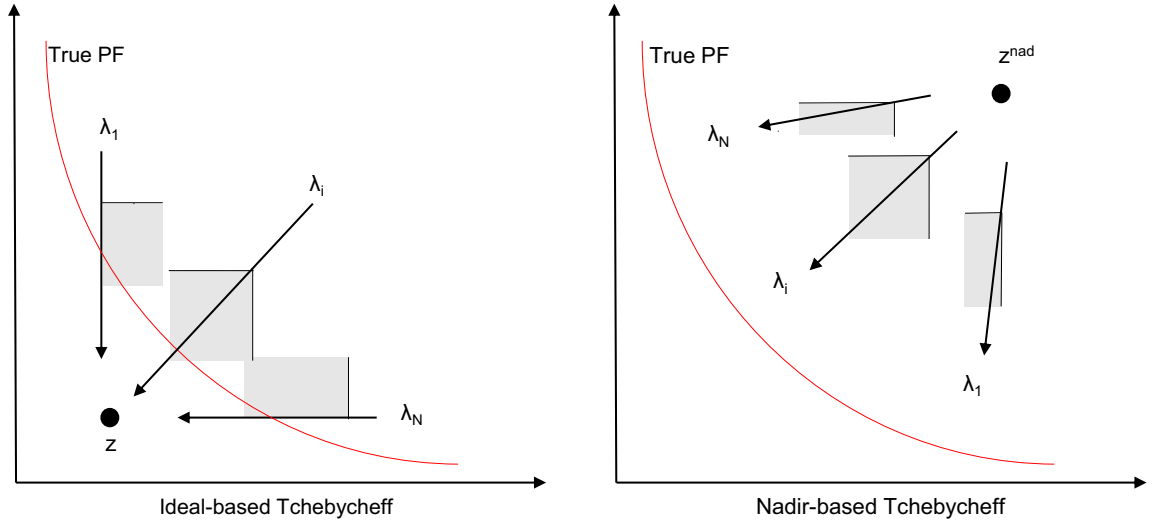

Fig. 4. Selection preference of two approaches.

is that the closer the two subproblems are, the more similar their solutions are. Based on this idea, a neighbor matching strategy (NMS) is proposed here. In the original version of MOEA/D, a subproblem and its neighbor have the same equivalence when mating them to generate a new solution. In the proposed NMS, we focus on improving the subproblem itself, and the improvement of its neighbors can be thought of as a carried interest. The relationship between subproblem i and subproblem j is defined as

$$R_{ij} = exp(-\rho||\lambda_i - \lambda_j||^2), \tag{31}$$

where λ_i and λ_j are the weight vectors of subproblems i and j, respectively. ρ is a parameter used to determine the strength of the relationship. When optimizing subproblem i, the relative relationship between other subproblems and subproblem i can be defined as

$$r_{ij} = \frac{R_{ij}}{\sum_j R_{ij}}, j = 1, \ldots, N. \tag{32}$$

The relative relationship provides the key parameter of NMS. Fig. 5 shows the relative relationship between subproblem 1 and other subproblems with respect to different ρ. With an increase in ρ, the relative relationship between subproblem closer and subproblem i increases, while the farther subproblem decreases.

Two different subproblems are chosen and crossed according to the relative relationship. The details are given in Algorithm 5 and Algorithm 6. The subproblem with a bigger value of the relative relationship is considered more promising for generating high-quality offspring for the current subproblem. It has the superiority of being selected and takes up a larger proportion in the generated solution after the crossover. Besides, in order to ensure diversity, initially, ρ is set to a small value $\rho_{\min}$. The relative relationship is not significantly different among subproblems at this time. As the evolution proceeds, ρ is increased to enhance the convergence. After the generation, R_{ij} is updated by

$$R_{ij} = R_{ij} Inc_{ij}, \tag{33}$$

$$Inc_{ij} = exp\left(-\frac{\rho_{max} - \rho_{min}}{G}||\lambda_i - \lambda_j||^2\right), \tag{34}$$

where Inc_{ij} is a constant incremental factor. It is calculated using Eq. (34). $\rho_{\max}$ and $\rho_{\min}$ are the maximum and minimum values of ρ, respectively, and G is the maximum generation of the evolutionary algorithms.

Algorithm 5 Select parents.

Input: $\{R_{ij}\}$, $j = 1, \ldots, N$.
Output: p_1, p_2.
1: $r_j = R_{ij}$, for $j = 1, \ldots, N$.
2: **for** m = 1:2 **do**
3: $r_j = \frac{r_j}{\sum_j r_j}$, for $j = 1, \ldots, N$.
4: Choose one subproblem $p_m \in \{1, \ldots, N\}$ based on the probability distribution $\{r_j\}$.
5: Set $r_{p_m} = 0$.
6: **end for**

Algorithm 6 Genetic operator.

Input: $I_{p_1}, I_{p_2}, R_{ip_1}, R_{ip_2}$.
Output: Offspring o.
1: $p_c = \frac{R_{ip_1}}{R_{ip_1} + R_{ip_2}}$.
2: **for** each genetic locus l **do**
3: **if** $rand < p_c$ **then**
4: $o(l) = I^l_{p_1}$.
5: **else**
6: $o(l) = I^l_{p_2}$.
7: **end if**
8: **end for**
9: Perform mutation on offspring o.

4.4. Repairing technique

Since the constraint sets (6)–(9) are always satisfied using the proposed solution representation, here we focus on handling the ammunition constraints and the continuous guidance constraints. For the ammunition constraints, if the consumption of a WP exceeds its remaining ammunition, we randomly eliminate some assigned targets and set the corresponding locus to 0 until the constraint is satisfied. For the continuous guidance constraints, we use a mask-based technique and aim to find all the valid loci in an individuals. A invalid locus is defined as a locus where the assigned target can't be engaged if this locus corresponding to a WP or has no weapon guiding to it at that stage if this locus corresponding to

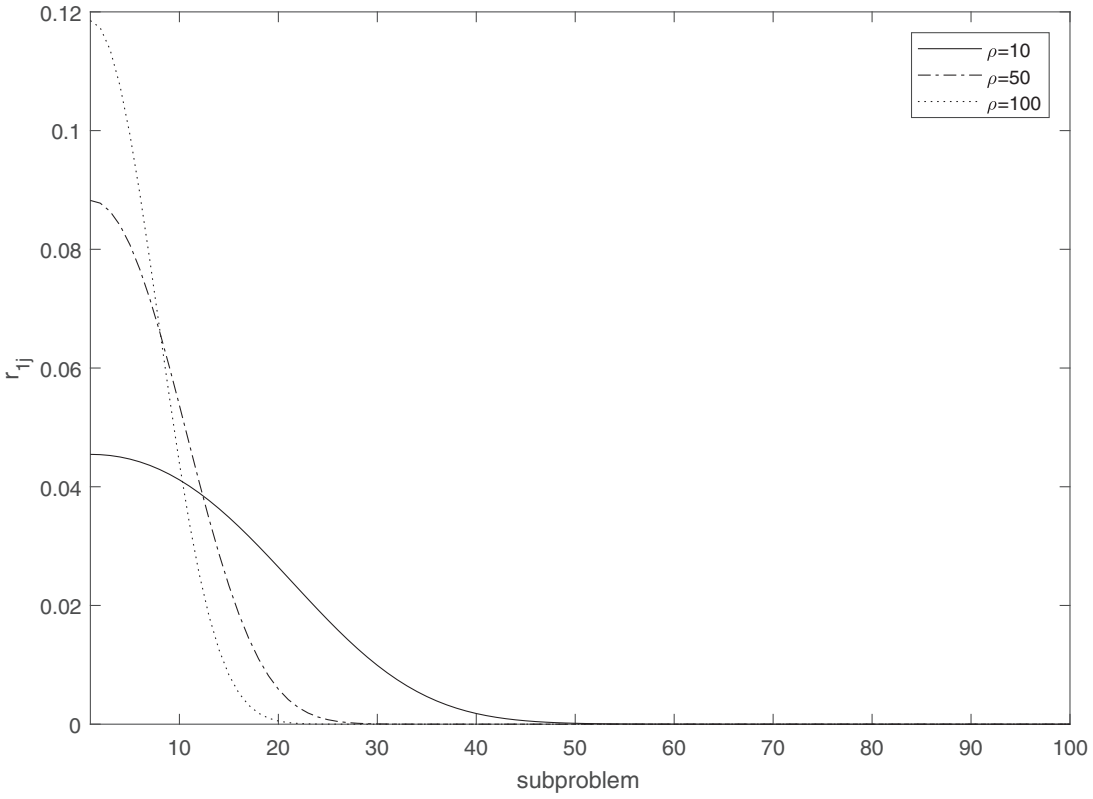

Fig. 5. Relative relationship with respect to different ρ.

Table 2
f_{sik}, $p_{ik}(s)$ and N_{sik} related to WP and f_{sjk} and $q_{jk}(s)$ related to SP of illustrative example.

	WP		SP	
	$k = 1$	$k = 2$	$k = 1$	$k = 2$
s=1	1, 0.5813, 2	1, 0.3738, 1	0, 0.4254	1, 0.5490
s=2	1, 0.5223, 1	1, 0.3848, 1	1, 0.3665	1, 0.6952

a SP. For example, if T_1 is assigned to WP_1 at stage 1, and no SP is assigned to this target at this stage, then this assignment is invalid. We find all these valid loci and evaluate the fitness of them.

4.5. Illustrative example

For a better understanding of the procedure of our proposed algorithm, we present a typical numerical example in this subsection. Assume that a WP and a SP have 2 stages to defense 2 targets in a combat scenario. The threat values of 2 targets are 98.45 and 46.20. The costs of WP and SP are 37.36 and 15.06 respectively. WP has enough ammunition resources. Robust parameters are set to 0. Other parameters are given in Table 2.

To the first beginning, we build the coding tables WS and QS. As we can see in Fig. 6, although T_1 can be engaged by WP at first stage, it is not included in WS_{11} since it require 2 stages of continuous guidance however there are no SP can finish this task at first stage.

Heuristic initialization is followed in the next phase. We find all the valid quaternions and put them into AQS. Initially, there are three valid ones, which are (1,1,1,2), (2,1,1,1) and (2,1,1,2). We evaluate the ECR of each quaternion as show in figure and then the quaternion (2,1,1,1) with the maximum ECR is selected as the candidate to add into the scheme. it's clear that the empty solution would change to [0, 1, 0, 1] by referring the solution representation. At this time, the quaternion (2,1,1,2) would become infeasible since both WP and SP are assigned to T_1 at stage 2 and they can not be assigned to T_2 further at this stage. In the next step, the quaternion (1,1,1,2) is added into the scheme. There are three heuristic individuals at this time and it exceeds our limit if we set $U = 2$, the crowding-distance-based deleting is employed and the second individual with the minimum CD will be deleted. Finally, since the size of population is $P = 4$, we randomly insert other 2 individuals that are randomly generated into the population to finish the construction of the initial population.

At the evolutionary process, for each subproblem i, we select two different individuals according the probability distribution $p_j = r_{ij}$. In our example, we assume that I_1 and I_3 are selected. Then we perform the crossover operator with the probability P_1 of selecting locus from I_1. Then the mutation operator changes the third locus to 1. The newly generated solution is used to update the neighbors of subproblem i using the nadir-based Tchebycheff. This process continued until the stopping criteria satisfied.

5. Experiment and result analysis

5.1. Comparison algorithm

To prove the effectiveness of the modified MOEA/D with heuristic initialization, we take another famous multi-objective optimization framework, NSGA-II (Deb et al., 2000), as the main comparison algorithm. Following six algorithms are designed to compare: NSGA-II with and without heuristic initialization (NSGA-II-Heuristic and NSGA-II), MOEA/D with and without heuristic initialization (MOEA/D-Heuristic and MOEA/D), and modified MOEA/D with and without heuristic initialization (MMOEA/D-Heuristic and MMOEA/D). Uniform crossover and random mutation operators are applied to the algorithms without the proposed modifications. Random initialization is used for those without heuristic.

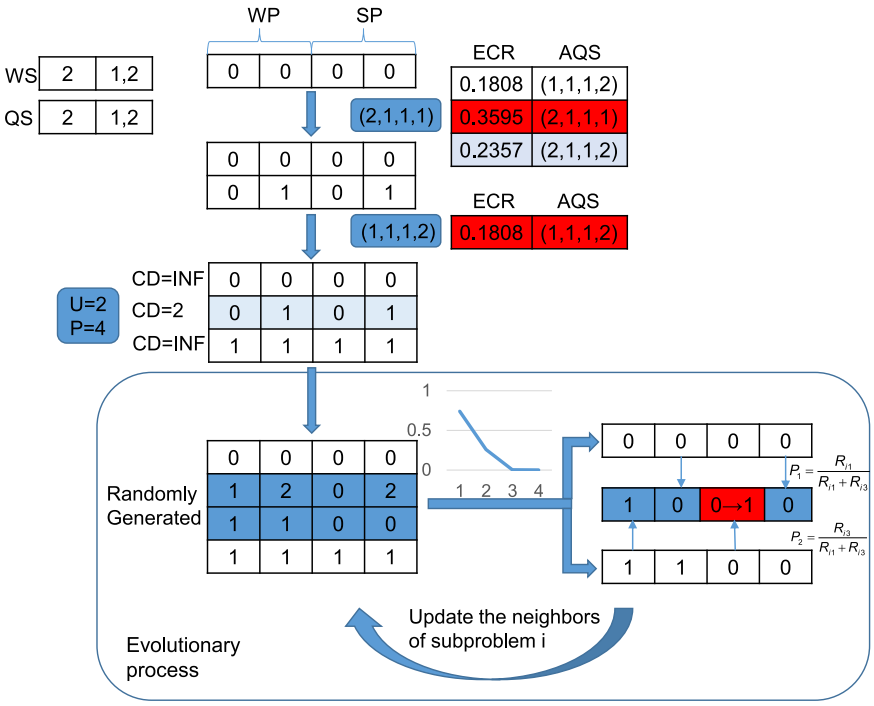

Fig. 6. An example to illustrate the procedure of the proposed algorithm.

5.2. Performance metrics

Several popular metrics are introduced to compare the performance of the different algorithms.

Inverted generational distance (IGD) (Ishibuchi, Masuda, Tanigaki, & Nojima, 2015): This is a metric that can measure both diversity and convergence. It represents the average distance of the given set of non-dominated solutions to the true Pareto front. The formulation is as follows:

$$IGD(A, P^*) = \frac{\sum_{v \in P^*} d(v, A)}{|P^*|}, \tag{35}$$

where P^* is the true Pareto front, $d(v, A)$ is the minimum distance between v and the points in A. It is hard to obtain the true Pareto front. In this paper, the true Pareto front is set as the set of non-dominated solutions among all solutions obtained by different algorithms.

Generational distance (GD) (Ishibuchi et al., 2015): Unlike the IGD metric, the GD metric measures the average distance from an inverted perspective. This metric puts more emphasis on convergence. Its formulation is as follows:

$$GD(A, P^*) = \frac{\sum_{v \in A} \tilde{d}(v, P^*)}{|A|}, \tag{36}$$

where $\tilde{d}(v, P^*)$ is the minimum distance between v and the points in P^*.

Diversification metric (DM) (Afzalirad & Rezaeian, 2017): This metric measures the spread of non-dominated solutions on the Pareto front. It is calculated as follows:

$$DM = \sqrt{(\max f_{1i} - \min f_{1i})^2 + (\max f_{2i} - \min f_{2i})^2}, \tag{37}$$

where f_{1i} and f_{2i} are the first and second objective values of ith solution, respectively.

5.3. Case generator

Since it is difficult to obtain real battlefield data, we develop a test case generator to generate several instances with different scales to compare the performance of different algorithms (Xin et al., 2010). Given the parameters S, W, Q, and T, the other parameters are generated as follows.

(1) $\{v_k\}$

$$v_k = 10 + 90 \cdot rand. \tag{38}$$

The threat value of the kth target is randomly generated in the interval of [10,100].

(2) $\{p_{ik}(s)\}$, $\{q_{jk}(s)\}$

$$p_{ik}(s) = P_L + rand \cdot (P_H - P_L), \tag{39}$$

$$q_{ik}(s) = Q_L + rand \cdot (Q_H - Q_L), \tag{40}$$

where P_H and P_L are the upper and lower bounds of the destroying probability of WP, respectively. Q_H and Q_L are the upper and lower bounds of the tracking performance of SP, respectively. *rand* generates a uniform random number within the range of [0,1]. $p_{ik}(s)$, $q_{jk}(s)$ are randomly generated in these two ranges. In the following, P_H and Q_H are set to 0.7, while P_L and Q_L are set to 0.3.

(3) $\{c_i\}$, $\{d_j\}$

Similar to 2), c_i and d_j are randomly generated in the intervals [10,50] and [5,30], respectively.

(4) $\{f_{sik}\}$, $\{f_{sjk}\}$

$$f_{sik} = (sgn(rand - f_{r1}) + 1)/2, \tag{41}$$

$$f_{sjk} = (sgn(rand - f_{r2} + 1)/2, \tag{42}$$

where $sgn(\cdot)$ is equal to 1 if its argument is positive, and -1 otherwise, while f_{r1} and f_{r2} denote the probability that the parameter is equal to 0. We set them to 0.5.

(5) $\{F_i\}$

$$F_i = \lfloor (F_{max} + 1) \cdot rand \rfloor. \tag{43}$$

This will randomly generate an integer ranging from 0 to $F_{\max}$. In our implementation, $F_{\max}$ is set to 4.

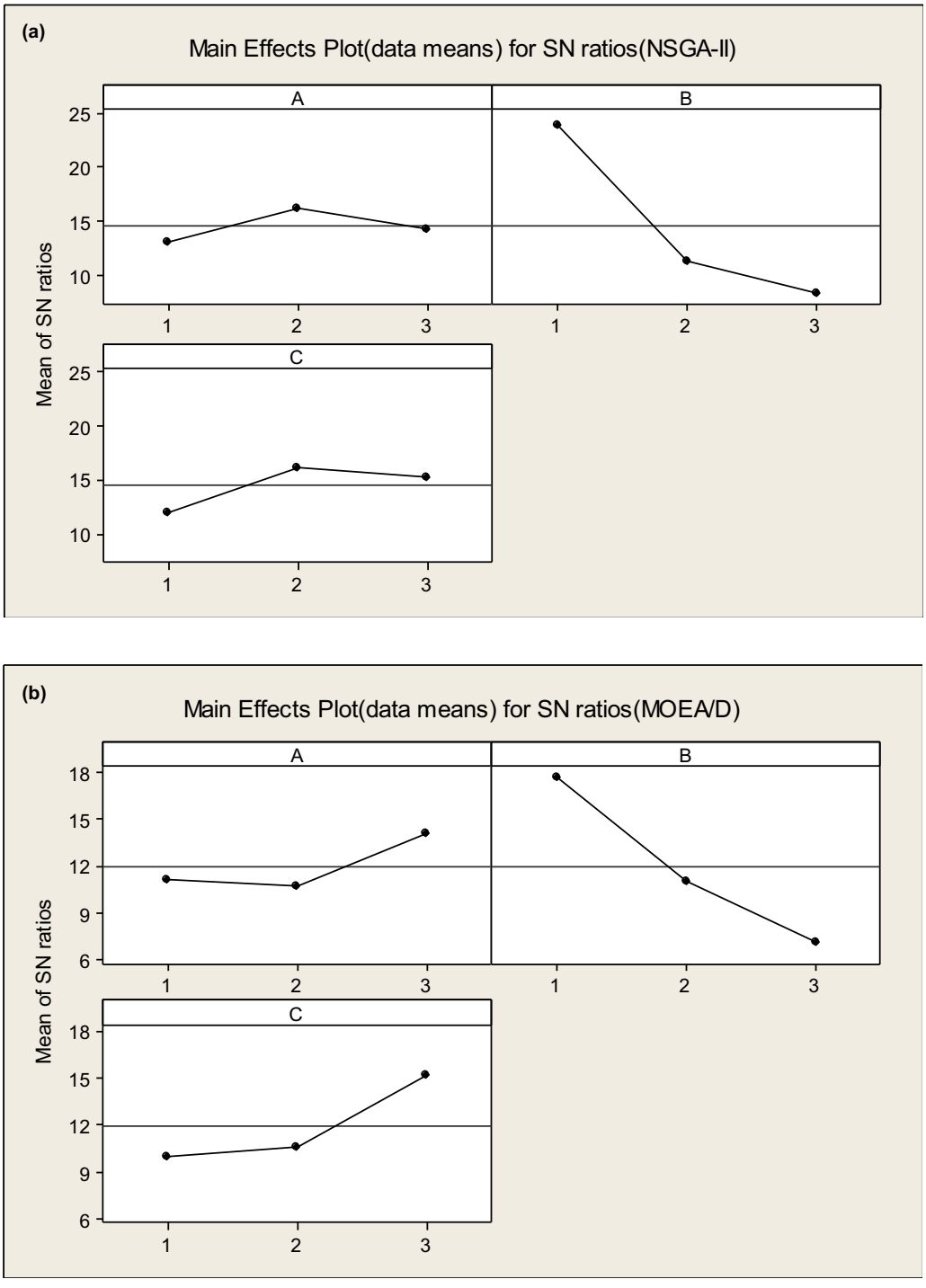

Fig. 7. (a)-(f): Taguchi ratios for all algorithms.

(6) $\{N_{sik}\}$
Similar to 5), N_{sik} is set to a randomly generated integer ranging from 1 to $N_{\max}$. In our experiment, $N_{\max} = 2$.
(7) $\{\gamma_{iks}\}$, $\{\gamma_{jks}\}$

Similar to 2), the uncertain parameters of the combat platforms are randomly generated in the interval [0.1,0.5].

In the computational experiment, several instances with different scales are randomly generated using the case generator. In order to illustrate the necessity of a robust model, the robust regulator parameter σ is set to three different levels: L1: $\sigma = 0$, L2: $\sigma = 0.5$, and L3: $\sigma = 1$. Thus, 12 different instances are given in Table 3, where S, W, Q, T represent the number of stages, WPs, SPs, targets respectively. The instances under the same scenario share the same parameters except for different robust level. This is for the comparison of robust model in following subsection.

5.4. Parameter tuning

As discussed earlier, in order to calibrate the parameters, Taguchi method is applied. It uses orthogonal arrays to study a group of factors. Factors that influence the performance are categorized into two groups: 1) controllable or signal factors and 2) noise factors. The Taguchi method seeks to find the optimal

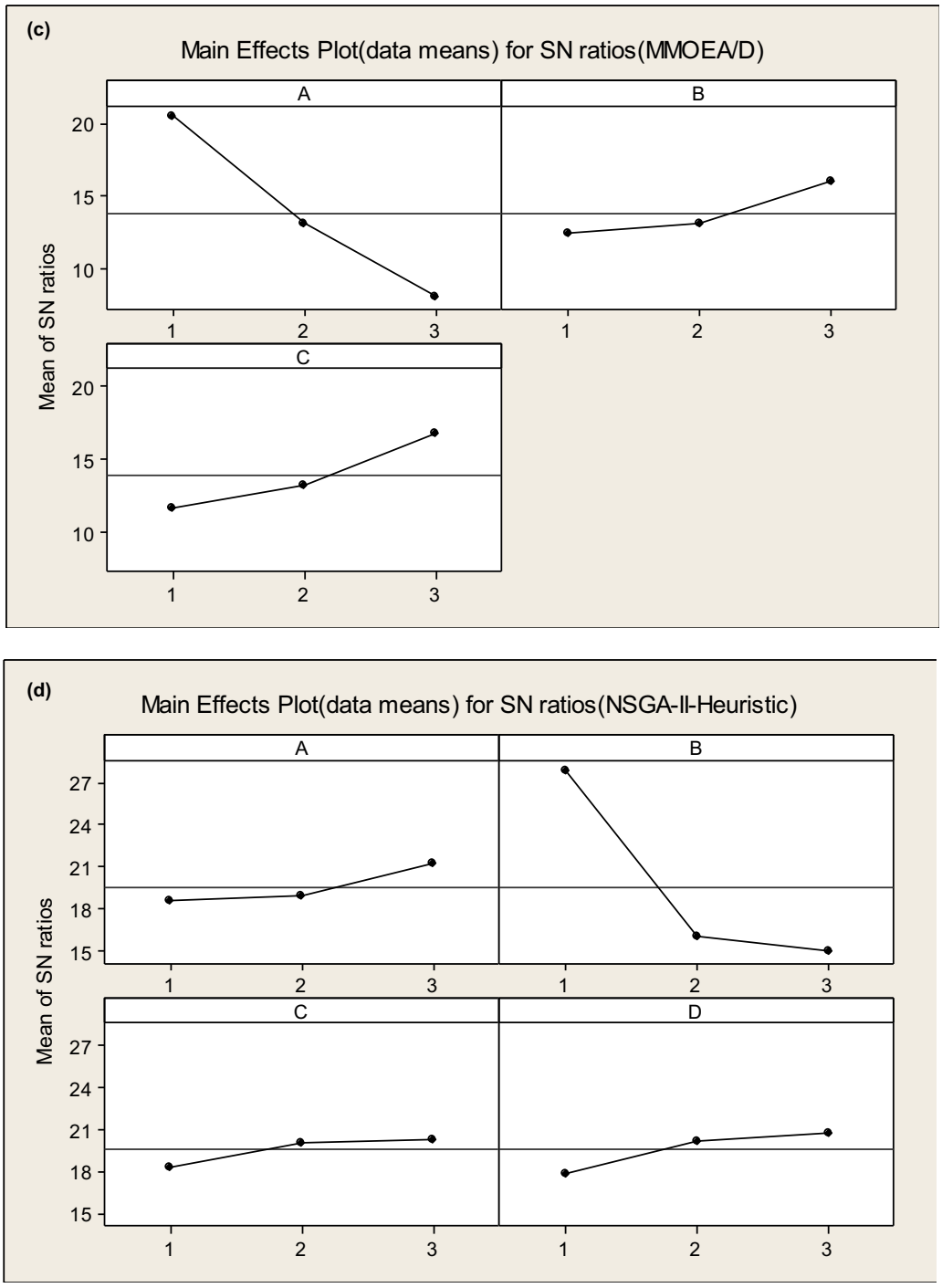

Fig. 7. Continued

combination of signal factor levels that minimizes the effects of noise factors in the response. It is a robust process design for noise factors. The concept of signal to noise ratio(S/N) is introduced as the goal of calibration. In this paper, the smaller-the-better type of response is adopted. The S/N is given as follows:

$$S/N = -10 \times \log\left(\frac{S(Y^2)}{n}\right), \tag{44}$$

where Y and n are the response and the number of orthogonal arrays respectively. $S(Y^2)$ is the summation of responses under one combination of signal factors. In this paper, a new metric called combinatorial ratio (C.R.) is introduced as the response of the Taguchi method. It is given by Eq. (45). GD and DM are metrics discussed in the previous subsection. It is clear that a smaller value of C.R. is better. This metric requires that the final non-dominated solutions converge closely to the true Pareto front and can approximate the whole Pareto front.

$$C.R. = \frac{GD}{DM}. \tag{45}$$

For a fair comparison, the size of the population (PS) and the number of generations (NOG) are set the same for different algorithms as shown in Table 4. The other parameters to be calibrated are given in Table 5, where CR stands for cross rate, MR stands for mutation rate, TS stands for tournament size, U/P stands

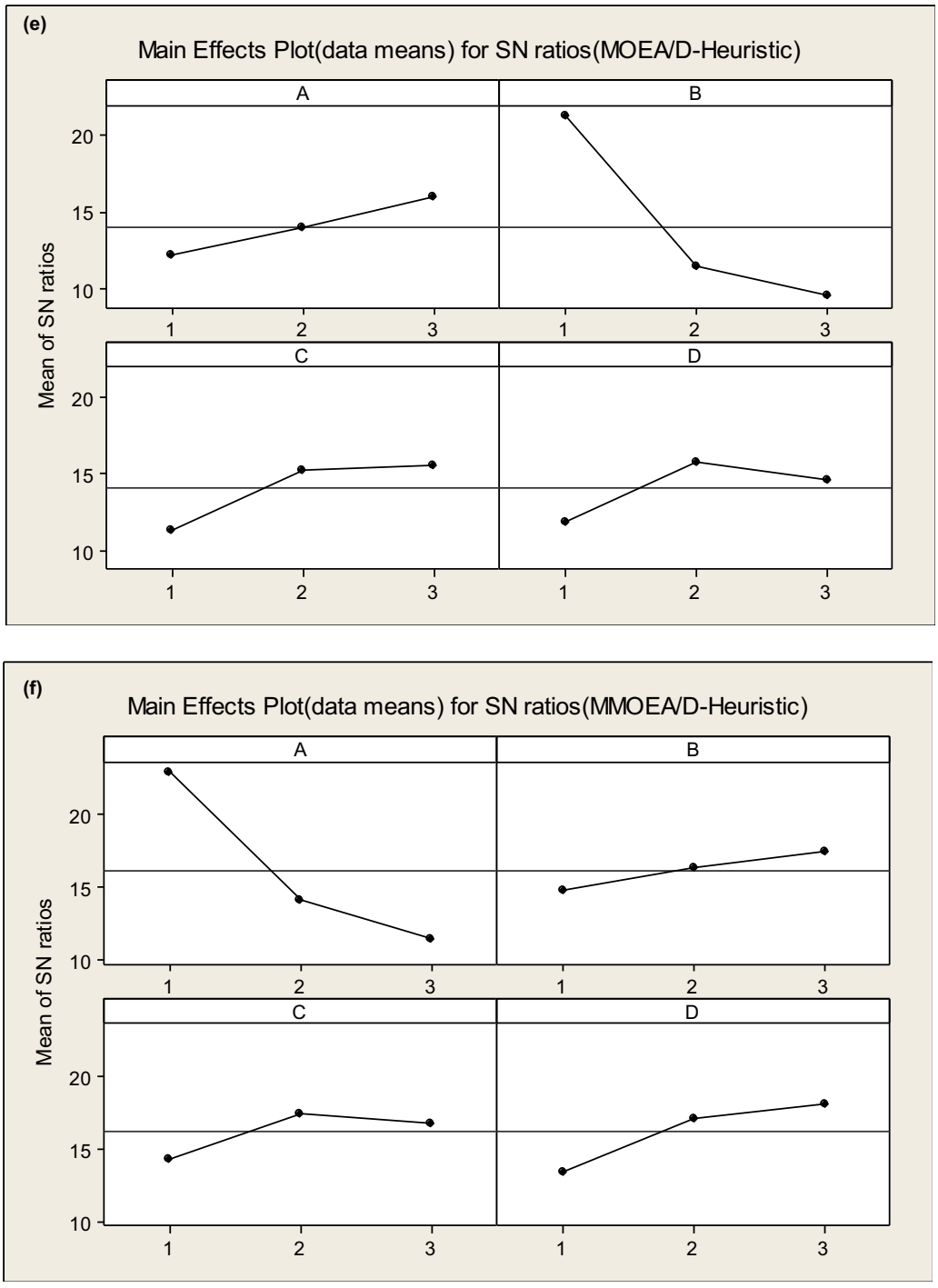

Fig. 7. Continued

for the ratio of the size of heuristic individuals to the size of the population. The ranges of these parameters are choosing via trials and errors to a less sensitive sections around the optimal values for ease of tuning. In the range of each parameter, its value is divided into three levels: Low, Medium, and High. Instances 1, 4, 7, and 10 with different scales are chosen as representatives to perform the Taguchi tests. The tuned parameters are expected to be feasible in all instances. For algorithms with 3–4 parameters, the L9 design is employed. The effect plots of S/N ratios are presented in Fig. 7. The parameter level with the maximum mean of S/N is chosen as the optimal level. The calibration result is highlighted in Table 5.

5.5. Result analysis

In this subsection, we perform numerical experiments to compare the 6 different algorithms on 12 different instances and analyze the results. The parameters are set the same as the calibration result in the previous subsection. Independent runs are performed on each instance 30 times for all algorithms. The program is implemented using MATLAB software and run on a laptop with 2.6 GHz Core i5 CPU and 4.00 GB RAM. The statistical results for the metrics IGD, GD, and DM are presented in Tables 6–8, respectively. For each instance, we give the rank of the mean value with respect to each metric. The Wilixon's rank sum test at a 5% significance

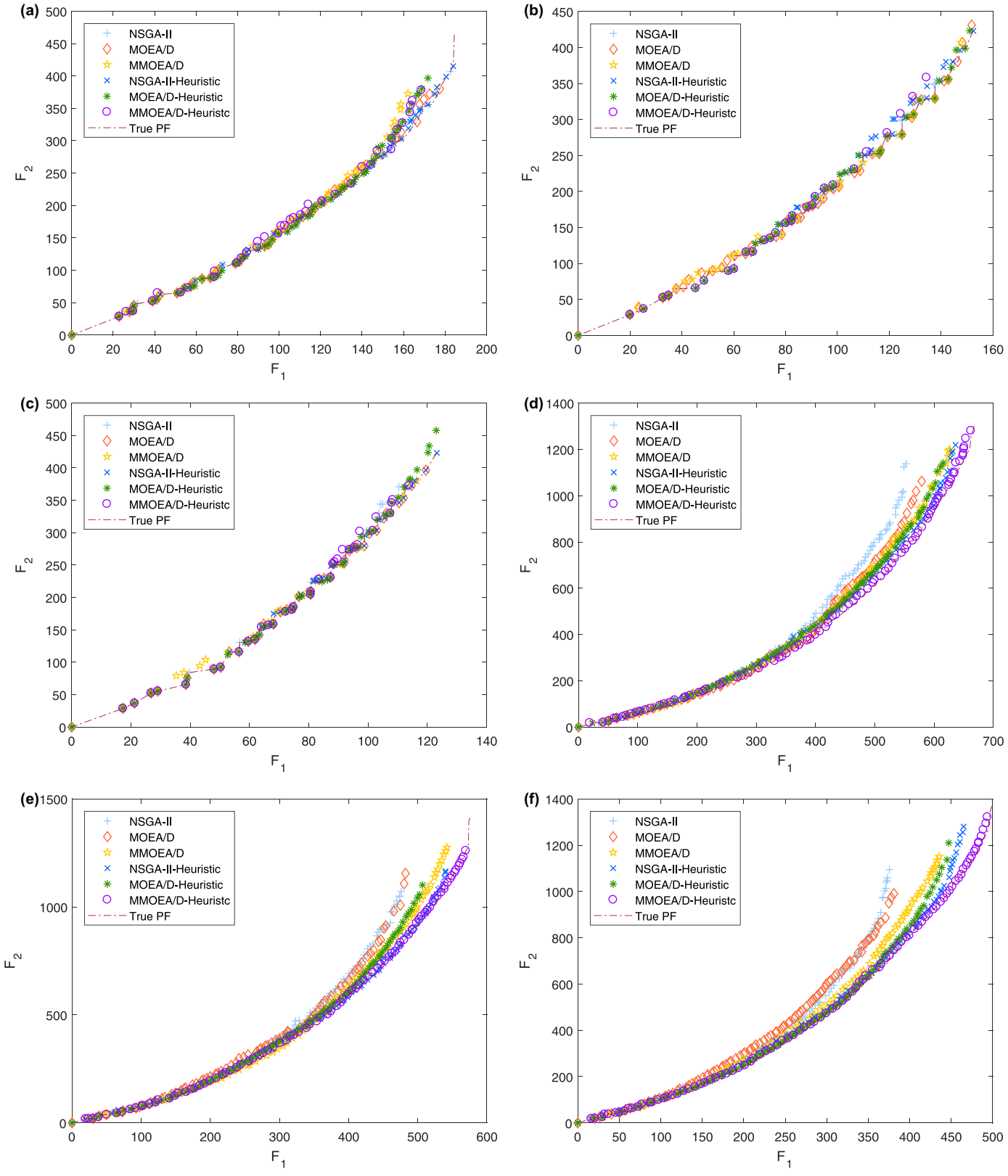

Fig. 8. (a)–(l): PF of a single run with minimum IGD value on 12 instances.

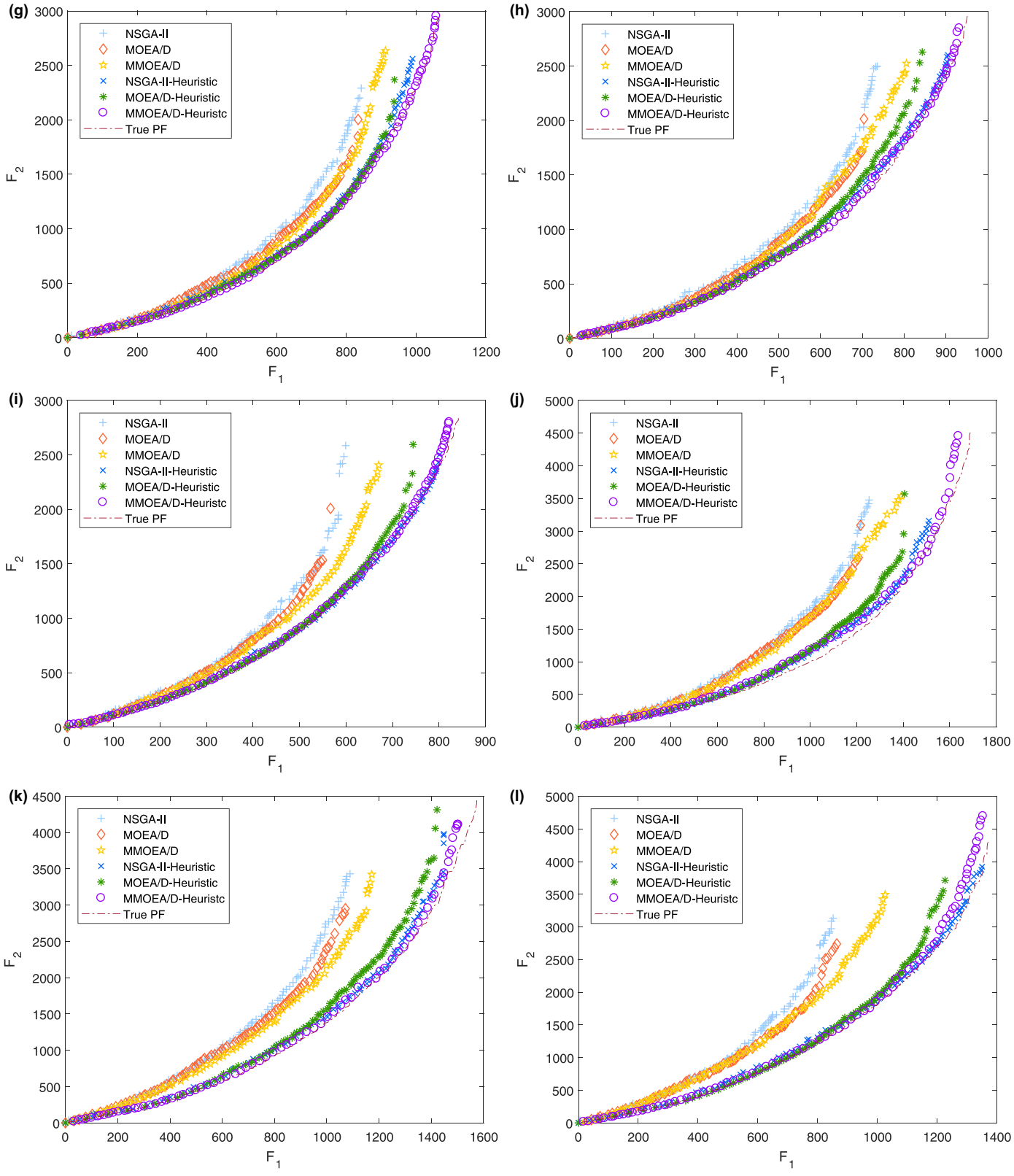

Fig. 8. Continued

level is conducted to test the significance of the differences between the mean metric values yielded by MMOEA/D-Heuristic and the other comparison algorithms. The symbols †, §, and ≈ indicate that the performance of MMOEA/D-Heuristic is better than, worse than, and similar to that of the comparison algorithm according to Wilixon's rank sum test, respectively. The bold data in the table are the best mean metric values for each instance. The data in parentheses are the standard deviations. Table 9 summarizes the overall performance of the mean rank and the statistical result obtained via the Wilixon's rank sum test. Fig. 8 shows the PFs of a

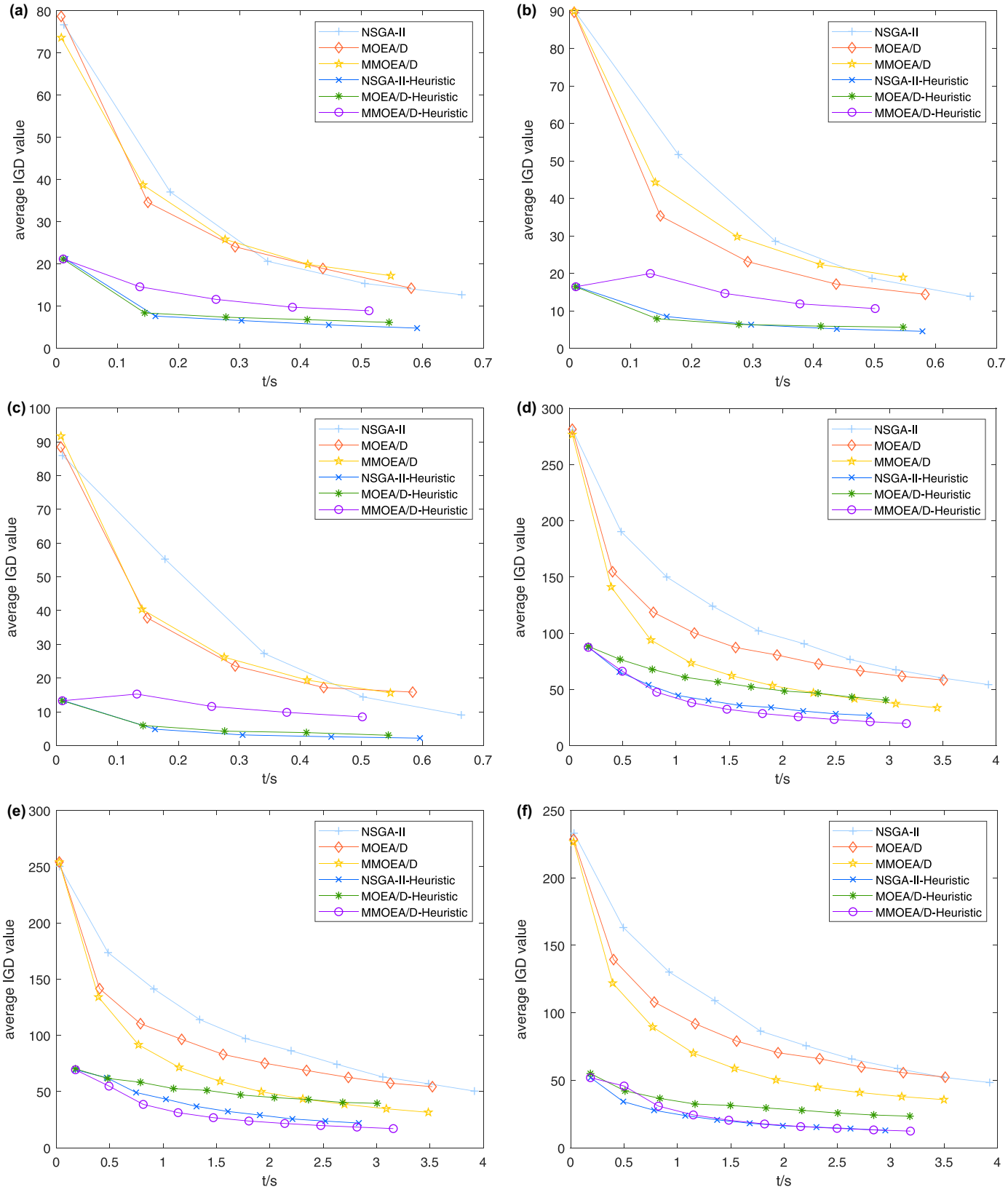

Fig. 9. (a)–(l): The convergence curves of IGD with respect to time on 12 instances.

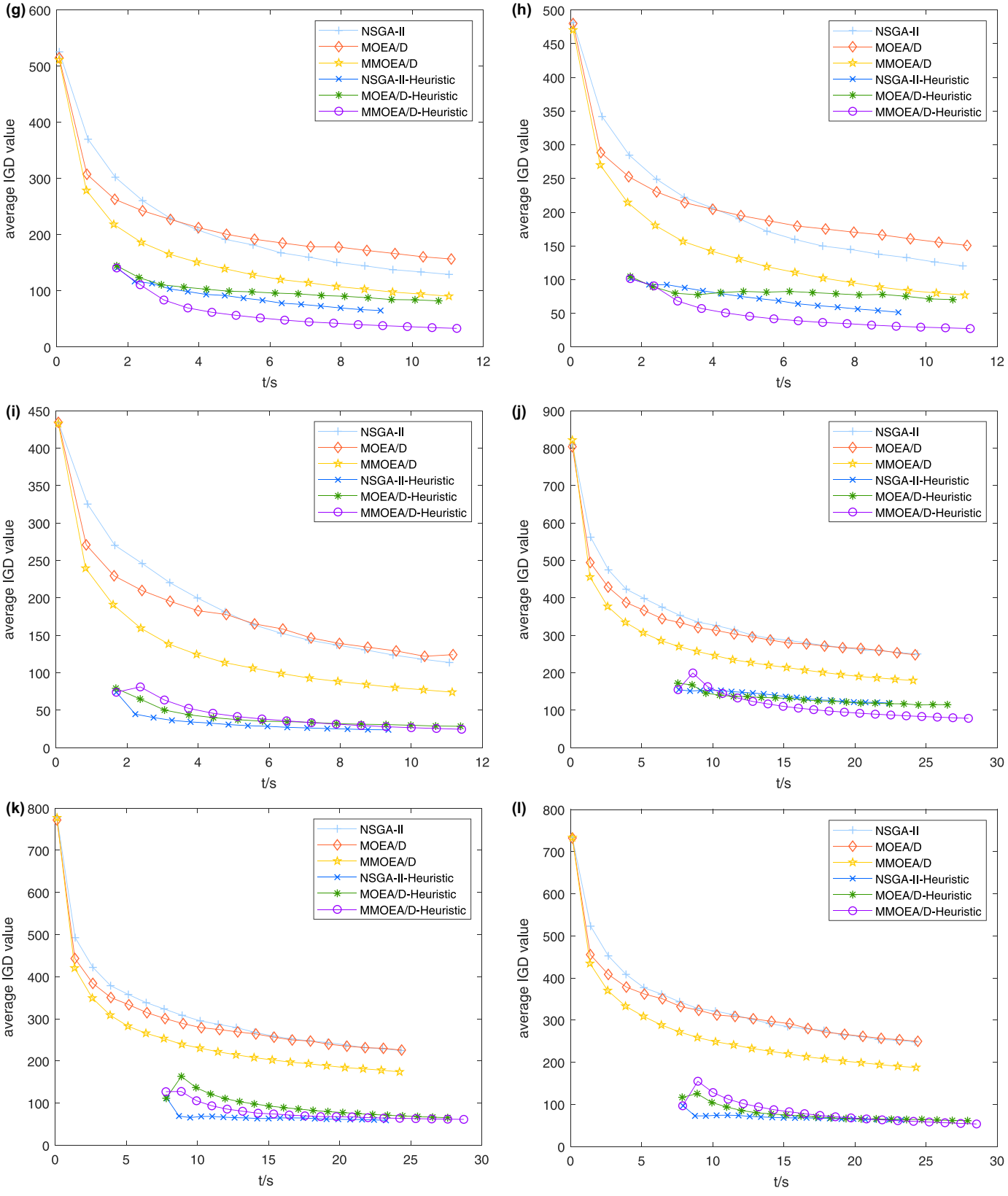

Fig. 9. Continued

Table 3
The setting of test instances.

Instance	Robust Level	Scenario
No. 1	L1	(Sce 1)S5W4Q4T10
No. 2	L2	
No. 3	L3	
No. 4	L1	(Sce 2)S8W10Q15T20
No. 5	L2	
No. 6	L3	
No. 7	L1	(Sce 3)S12W25Q20T30
No. 8	L2	
No. 9	L3	
No. 10	L1	(Sce 4)S15W30Q30T40
No. 11	L2	
No. 12	L3	

Table 4
Basic parameter setting.

Instances	1,2,3	4,5,6	7,8,9	10,11,12
PS	100	100	100	100
NOG	100	200	300	400

single run with minimum IGD value for different algorithms on 12 instances.

(1) The effectiveness of modifications and heuristic initialization As shown by the statistical results, the modifications of MOEA/D performs relatively poor in the small-scale instances 1, 2, and 3. The main reason is that in the small case, the solutions of different subproblems do not have significant differences. Therefore, the newly generated solution does not obtain sufficient variance using the neighbor structure and can easily be trapped into the local optimal. NSGA-II using the n-tournament selection and randomly matching has more advantages in the small-scale case. As the scale of the problem increases, the superiority of MMOEA/D becomes apparent. On the one hand, compared to NSGA-II and MOEA/D, MMOEA/D shows the best performance in terms of the DM metric. The nadir-based Tchbycheff approach gives a better approximation of the whole Pareto front than the ideal-based. On the other hand, the convergence of MMOEA/D is enhanced with the help of NMS. It should be noted that there is a contradiction between the DM and GD metrics. The smaller the DM metric is, the better the GD metric will be since the computation is emphasized on a smaller region. To illustrate the effectiveness of NMS considering this contradiction, I^{gd} and I^{dm} are given as follows to compare MOEA/D and MMOEA/D:

$$I^{gd} = \frac{GD_m - GD}{GD} \tag{46}$$

$$I^{dm} = \frac{DM_m - DM}{DM} \tag{47}$$

where GD and DM are the mean GD and DM values of MOEA/D on one instance, respectively. GD_m and DM_m are the mean GD and DM values of MMOEA/D on the same instance, respectively. I^{gd} and I^{dm} measure the relative increase in the GD and DM metrics caused by the modified part of MOEA/D. These two values on 12 instances are given in Table 10. It is clear that I^{dm} is bigger than the I^{gd} metric for most instances. A better approximation of the whole Pareto front with a relatively lower decrease in the performance of the GD metric illustrates the effectiveness of NMS, especially in large scale cases. Since IGD is a comprehensive metric, it also illustrates the superiority of MMOEA/D compared to NSGA-II and MOEA/D.

After the heuristic initialization is incorporated into the algorithms, NSGA-II-Heuristic achieves a better performance than MOEA/D-Heuristic. However, we can still see that MMOEA/D-Heuristic performs better on IGD and DM metrics and remain almost even on the GD compared with NSGA-II-Heuristic. A comparison between MOEA/D-Heuristic and MMOEA/D-Heuristic shows that the modified strategy can greatly improve the performance of MOEA/D under heuristic initialization.

As for the initialization strategy based on efficiency-cost ratio, from all the compared metrics, its necessity is apparent. All the algorithms with heuristic initialization are superior to those without it.

The mean rank presented in Table 9 shows that MMOEA/D-Heuristic ranks first among all, and MMOEA/D ranks first among the three algorithms without heuristic initialization.

Table 5
Algorithm parameter ranges along with their levels.

Multi-objective algorithms	Algorithm parameters	Parameter range	Low(1)	Medium(2)	Large(3)
NSGA-II	CR(A)	0.6–0.8	0.6	0.7	0.8
	MR(B)	0.01–0.05	0.01	0.03	0.05
	TS(C)	1–3	1	2	3
MOEA/D	CR(A)	0.6–1	0.6	0.8	1
	MR(B)	0.01–0.05	0.01	0.03	0.05
	T(C)	5–15	5	10	15
MMOEA/D	MR(A)	0.01–0.05	0.01	0.03	0.05
	$\rho_{\min}$(B)	25–75	25	50	75
	$\rho_{\max}$(C)	100 200	100	150	200
NSGA-II-Heuristic	CR(A)	0.6–0.8	0.6	0.7	0.8
	MR(B)	0.01–0.05	0.01	0.03	0.05
	TS(C)	1–3	1	2	3
	U/P(D)	0.1–0.4	0.1	0.25	0.4
MOEA/D-Heuristic	CR(A)	0.6–1	0.6	0.8	1
	MR(B)	0.01–0.05	0.01	0.03	0.05
	T(C)	5–15	5	10	15
	U/P(D)	0.1–0.4	0.1	0.25	0.4
MMOEA/D-Heuristic	MR(A)	0.01–0.05	0.01	0.03	0.05
	$\rho_{\min}$(B)	25–75	25	50	75
	$\rho_{\max}$(C)	100–200	100	150	200
	U/P(D)	0.1–0.4	0.1	0.25	0.4

Table 6
Comparison of IGD.

	NSGA-II	MOEA/D	MMOEA/D	NSGA-II-Heuristic	MOEA/D-Heuristic	MMOEA/D-Heuristic
No. 1	8.1897§[3](4.0216)	11.2333≈[5](4.6026)	13.5177†[6](4.3067)	**4.7752**§[1](**1.4657**)	6.1208§[2](1.212)	10.8284[4](3.3159)
No. 2	6.7144§[3](4.0643)	11.8985§[4](6.722)	15.3401≈[6](7.4336)	**3.5721**§[1](**0.79585**)	5.0046§[2](1.5217)	15.165[5](5.7316)
No. 3	7.0085§[3](3.5371)	11.3758≈[5](5.6998)	15.4622†[6](8.1054)	**2.1865**§[1](**1.2767**)	2.7318§[2](1.126)	11.0289[4](4.0486)
No. 4	46.4218†[5](6.6109)	48.9877†[6](10.5628)	33.4408†[3](8.6916)	21.896†[2](3.6471)	33.4497†[4](9.6625)	**12.6786**[1](**2.8647**)
No. 5	48.294†[6](8.6999)	47.5322†[5](9.7378)	26.5508†[3](7.5571)	19.1953†[2](5.4501)	37.4899†[4](10.8499)	**10.7277**[1](**2.4062**)
No. 6	44.4059†[5](7.8744)	54.5964†[6](6.5979)	31.5698†[4](7.6475)	12.1979†[2](2.6255)	22.9316†[3](6.8973)	**7.9872**[1](**1.657**)
No. 7	120.3177†[5](11.1618)	129.3656†[6](13.5988)	80.8877†[4](9.6519)	49.0242†[2](12.1271)	69.5727†[3](16.4761)	**21.0255**[1](**4.8404**)
No. 8	114.0853†[5](19.458)	142.8034†[6](17.5629)	73.4541†[4](9.3825)	47.0074†[2](13.8484)	64.8361†[3](20.8623)	**17.884**[1](**3.7479**)
No. 9	107.7719†[5](9.819)	124.2339†[6](13.89)	73.2567†[4](8.0731)	20.9982†[2](3.4971)	29.2277†[3](7.5018)	**14.0544**[1](**2.4736**)
No. 10	249.5716†[5](19.5241)	249.6157†[6](20.0271)	189.936†[4](10.5346)	119.402†[2](21.0057)	136.2349†[3](25.787)	**46.1347**[1](**5.0993**)
No. 11	229.5058†[6](18.1755)	226.6189†[5](14.328)	175.7186†[4](9.6031)	56.7808†[2](9.1682)	67.3445†[3](10.2782)	**31.2276**[1](**4.9239**)
No. 12	231.9612†[5](16.9235)	236.8257†[6](20.0957)	179.6928†[4](11.208)	43.266†[2](7.5725)	51.0314†[3](9.9346)	**26.2692**[1](**4.4915**)
†/§/≈	9/3/0	9/1/2	11/0/1	9/3/0	9/3/0	

Table 7
Comparison of GD.

	NSGA-II	MOEA/D	MMOEA/D	NSGA-II-Heuristic	MOEA/D-Heuristic	MMOEA/D-Heuristic
No. 1	**3.3632**§[1](**2.5105**)	5.5596≈[4](3.9349)	6.6961≈[6](4.4706)	4.2172§[2](2.1684)	5.3217≈[3](1.7564)	6.1848[5](2.2348)
No. 2	2.7484†[4](2.3277)	6.1322†[6](4.4711)	5.52†[5](4.3112)	2.2833†[3](0.23461)	**1.628**≈[1](**0.52999**)	1.6886[2](0.76828)
No. 3	1.8318†[4](2.1392)	6.013†[5](4.4143)	7.6568†[6](4.4371)	0.82282†[3](0.20685)	0.78735†[2](0.29576)	**0.63239**[1](**0.8053**)
No. 4	41.3634†[6](9.2589)	26.9482†[4](7.8578)	35.21†[5](8.7313)	16.5687†[3](2.0239)	13.0505≈[2](3.6877)	**11.6893**[1](**3.233**)
No. 5	33.9842†[6](9.7405)	24.6953†[4](9.2815)	26.4347†[5](7.9665)	10.8574†[2](2.6145)	11.4434†[3](3.5005)	**8.8855**[1](**2.6259**)
No. 6	32.367†[5](6.6827)	29.3642†[4](7.3415)	32.4438†[6](8.3488)	8.2414†[3](1.6271)	7.9304≈[2](3.6489)	**6.3738**[1](**2.3157**)
No. 7	98.6666†[6](13.9295)	57.5787†[4](13.327)	76.3171†[5](9.3622)	18.9955≈[2](4.1967)	**15.795**§[1](**7.8556**)	20.0001[3](6.5517)
No. 8	91.7119†[6](11.2963)	54.1882†[4](13.3446)	73.8322†[5](10.0883)	18.0584≈[3](4.1385)	**15.9309**≈[1](**5.667**)	16.4357[2](5.4308)
No. 9	90.8852†[6](11.1888)	59.6647†[4](9.347)	74.0202†[5](9.1025)	**12.0513**≈[1](**2.903**)	14.2417≈[3](5.1848)	12.675[2](3.4597)
No. 10	246.5104†[6](20.0577)	170.0311†[4](20.5599)	186.8477†[5](11.9188)	**33.2194**§[1](**7.0312**)	61.9192†[3](26.7459)	42.956[2](6.9723)
No. 11	251.8857†[6](17.2594)	175.4591†[4](21.9591)	199.5874†[5](16.1971)	**25.3147**§[1](**4.5829**)	54.7884†[3](21.9265)	32.1558[2](7.6379)
No. 12	231.6516†[6](21.5368)	155.5378†[4](23.273)	199.9317†[5](15.5215)	**19.9941**§[1](**4.8203**)	41.4174†[3](14.3707)	24.9027[2](6.6515)
†/§/≈	11/1/0	11/0/1	11/0/1	5/4/3	5/1/6	

Table 8
Comparison of DM.

	NSGA-II	MOEA/D	MMOEA/D	NSGA-II-Heuristic	MOEA/D-Heuristic	MMOEA/D-Heuristic
No. 1	389.592§[3](37.4239)	382.4988§[4](31.4648)	339.9919†[6](32.6963)	**444.2962**§[1](**26.8517**)	444.0833§[2](17.7894)	355.5651[5](25.4693)
No. 2	393.9062§[4](35.7881)	394.1848§[3](51.027)	325.5311§[5](47.2558)	**416.4242**§[1](**16.683**)	402.0545§[2](23.5178)	298.6246[6](24.2296)
No. 3	362.8592§[4](36.1275)	372.8791§[3](47.8951)	320.6564§[5](50.9549)	404.9622§[2](23.0615)	**406.5821**§[1](**29.8917**)	296.8157[6](21.5294)
No. 4	1234.3957†[4](94.6385)	1102.084†[6](83.906)	1388.0034≈[2](67.1363)	1276.2175†[3](77.2405)	1125.9806†[5](77.0893)	**1404.0331**[1](**35.9819**)
No. 5	1133.7655†[4](83.8843)	1091.6675†[5](94.2224)	1369.0141≈[2](68.4812)	1244.848†[3](81.9575)	1070.6588†[6](77.9422)	**1386.5861**[1](**36.5755**)
No. 6	1096.2139†[4](96.2958)	978.2114†[6](64.3693)	1324.4972≈[2](59.7193)	1266.3439†[3](83.2416)	1091.2242†[5](77.3163)	**1339.5139**[1](**53.66**)
No. 7	2262.5516†[4](175.2617)	1909.5924†[6](128.9347)	2632.2703†[2](88.169)	2358.4352†[3](176.5866)	2115.5118†[5](188.474)	**3159.4834**[1](**59.6493**)
No. 8	2280.2155†[4](216.7015)	1852.4035†[6](120.2123)	2622.4711†[2](109.0001)	2388.7434†[3](176.5225)	2196.2312†[5](189.9181)	**2977.7155**[1](**66.7503**)
No. 9	2140.5727†[5](183.1792)	1774.7104†[6](112.8489)	2530.3564†[2](101.3092)	2524.9726†[3](128.6506)	2369.4886†[4](175.1347)	**2893.9271**[1](**63.786**)
No. 10	3304.0783†[3](252.5444)	2760.8892†[6](194.6916)	3659.7427†[2](106.9569)	3072.4658†[5](202.6642)	3092.973†[4](295.756)	**4649.8304**[1](**152.8444**)
No. 11	3296.9752†[5](219.7755)	2775.6826†[6](207.296)	3623.2878†[3](106.0351)	3557.3529†[4](207.8268)	3772.5925†[2](371.8444)	**4624.7691**[1](**170.2016**)
No. 12	3074.7199†[5](253.0552)	2522.5279†[6](203.9411)	3606.5167†[3](129.1448)	3535.2622†[4](224.7087)	3666.3555†[2](306.3085)	**4420.9737**[1](**188.863**)
†/§/≈	9/3/0	9/3/0	7/2/3	9/3/0	9/3/0	

Table 9
Overall performance of four algorithms on the 12 instances in terms of IGD, GD, and DM metrics.

	Mean Rank	Total †/§/≈
NSGA-II	4.6389	29/7/0
MOEA/D	5	29/4/3
MMOEA/D	4.1944	29/2/5
NSGA-II-Heuristic	2.25	23/10/3
MOEA/D-Heuristic	2.9167	23/7/6
MMOEA/D-Heuristic	2	–

In addition, the total count of †/§/≈ shows that MMOEA/D-Heuristic outperforms all other algorithms. Fig. 8 gives an intuitive vision to the obtained Pareto fronts by all algorithms on 12 instances. This also confirms that our proposed strategy is effective in solving the model.

(2) The convergence curve with respect to time

Table 10
Comparison of MOEA/D and MMOEA/D on I^{gd} and I^{dm}.

Instance	I^{gd}	I^{dm}
1	0.2044	−0.1111
2	−0.0998	−0.1742
3	0.2734	−0.1742
4	0.3066	0.2594
5	0.0704	0.2541
6	0.1049	0.3540
7	0.3254	0.3784
8	0.3625	0.4157
9	0.2406	0.4258
10	0.0989	0.3256
11	0.1375	0.3054
12	0.2854	0.4297

In this part, we analysis the convergence curve of IGD with respect to time on 12 instances, as shown in Fig. 9. We

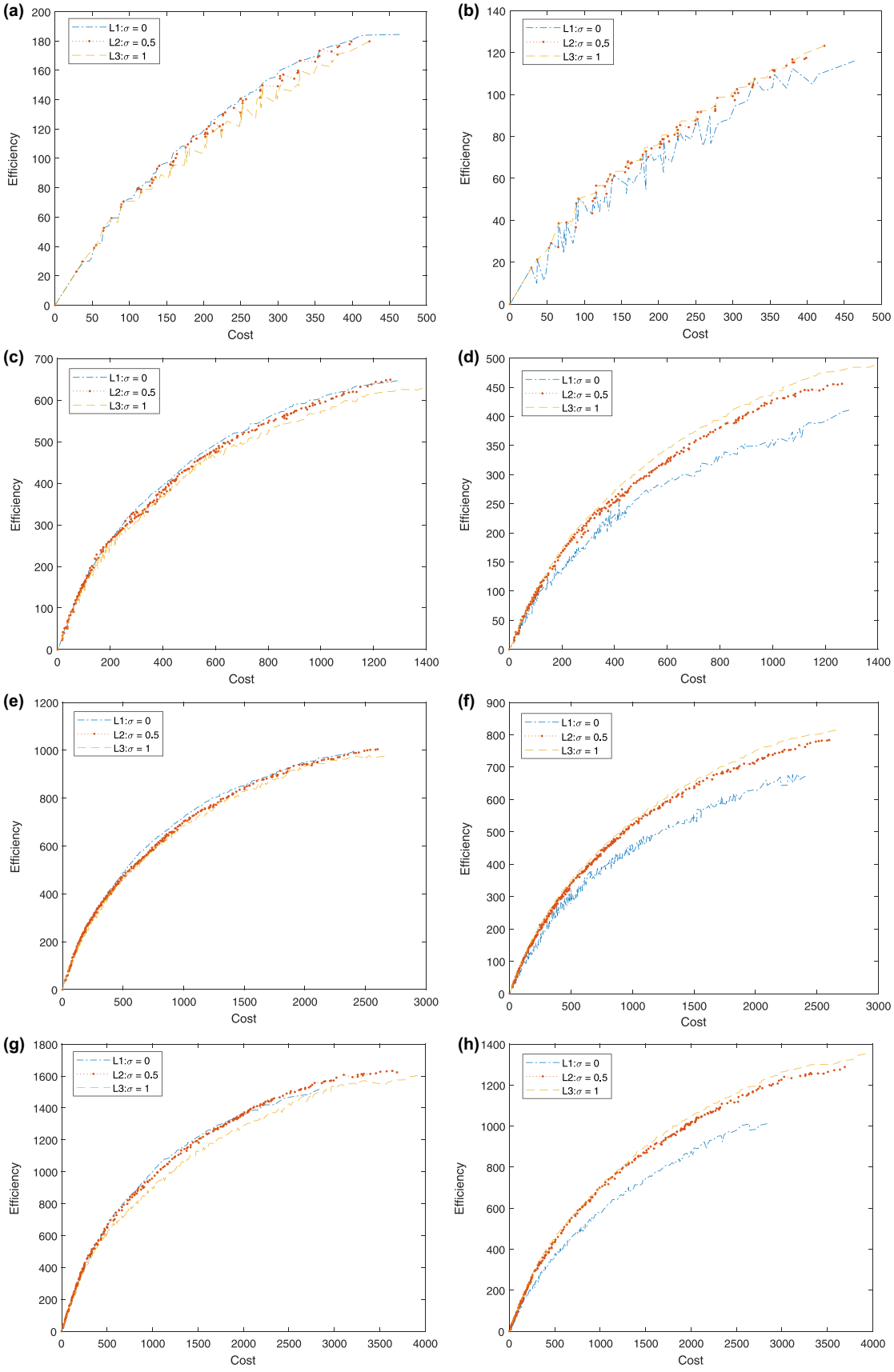

Fig. 10. (a)-(h): The performance under nominal and worst conditions for different robust levels. (a) Sce 1 nominal; (b) Sce 1 worst; (c) Sce 2 nominal; (d) Sce 2 worst; (e) Sce 3 nominal; (f) Sce 3 worst; (g) Sce 4 nominal; (h) Sce4 worst.

Table 11
Time consuming of random and heuristic initialization.

Instance	1	4	7	10
Random/s	0.0002	0.0003	0.0006	0.0010
Heuristic/s	0.0022	0.1562	1.6255	7.4255

sample the population each 20 generations, and record the IGD value of the non-dominated solutions in population and the corresponding time starting from the beginning of the algorithm. 30 independent runs are carried, and the average time versus the average IGD curves are as plotted.

As we stated earlier, the performance of decomposition-based algorithm are worse than NSGA-II in the small scale instances 1, 2 and 3. MMOEA/D and MMOEA/D-Heuristic have relatively poor performance than others in these instances. In other instances, the ranking of these curves from the bottom up is MMOEA/D-Heuristic, NSGA-II-Heuristic, MOEA/D-Heuristic, MMOEA/D, MOEA/D and NSGA-II. It's apparent that the modifications have enhanced the searching efficiency under the limit of time.

We further consider the time consuming of initialization. The heuristic initialization can give a better hybrid population at the cost of more time consuming. The time consuming of several instances is shown in Table 11 and they are also reflected in Fig. 9. Although it costs more time to do initialization, we can start from a better initial state, and Fig. 9 shows that this is much more efficient than evolving the population from a randomly initialized state. Besides, there is an interesting phenomenon that the algorithms with heuristic initialization cost less time evolving the same number of generations than those without it. This might be explained as that they need less repairing or replacing operations. In a whole, the heuristic initialization seems to be a very efficient way in our optimization problem. In fact, there are many tricks can be further utilized to accelerate the initialization process. For example, the calculation of ECR and the deleting of infeasible quaternions can be processed in parallel, or we can select one among several quaternions in AQS instead of selecting from the whole AQS. The time consuming is thus acceptable in our real dynamic problem.

(3) The necessity of robust model

To illustrate the necessity of the proposed robust model under uncertainty, we compare the non-dominated solutions of the same scales with different robust levels. There are four scenarios in our test instances as shown in Table 3. We study two different conditions for each scenario to test the performance of the robust model with different robust levels. They are the nominal and the worst conditions. Under the nominal condition, the uncertain parameters $p_{ik}(s, \xi)$ and $q_{jk}(s, \xi)$ take the nominal values $p_{ik}(s)$ and $q_{jk}(s)$, similar to the deterministic model. Under the worst conditions, the uncertain parameters take the worst values $(1-\gamma_{iks})p_{ik}(s)$ and $(1-\gamma_{jks})q_{jk}(s)$, respectively. The non-dominated solutions in the true Pareto front displayed in Fig. 8 are evaluated under these two conditions, and the cost-efficiency curve for each robust level in each scenario is plotted in Fig. 10.

From Fig. 10, it can be seen that under Sce 1, the performance of different robust levels under the nominal and the worst conditions do not have significant differences. However, with the increase in the scales, the advantage of the robust model becomes apparent. It is clear that under the worst condition of Sce 2, 3, and 4, the robust model has better efficiency than the deterministic model while keeping the costs the same. That is, under the worst condition, the assignment scheme obtained by the robust model achieves a greater threat elimination when consuming the same amount of combat resources as that obtained by the deterministic model. It is also seen that under the nominal condition, the cost-efficiency curves of the three robust levels almost overlap with each other. Thus, we can conclude that the robust model can effectively improve the performance under the worst condition while keeping almost the same performance as the deterministic model under the nominal condition.

6. Conclusion and future work

In this paper, we focus on modeling the dynamic collaborative task assignment. A bi-objective dynamic assignment model is established. To solve the model efficiently, a modified MOEA/D with heuristic initialization is proposed. The solution representation is designed. A novel constructive heuristic initialization based on efficiency-cost ratio is proposed to generate an initial hybrid population. The nadir-based Tchebycheff approach is employed to obtain a better approximation to the whole Pareto front. A neighbor matching strategy with an adaptive adjustment mechanism is proposed for better utilization of the neighbor information. Finally, we apply the Taguchi method to tune the parameters, and numerical experiments are carried out.

Dynamic, uncertain, flexible and cooperative natures in real battlefield are the main bottlenecks to achieve the automation of task assignment. According to the current research status in this field, these aspects has not been satisfied solved. This paper addresses these problems. Firstly, we specify the multi-stage dynamic adjustment mechanism which has not been clearly pointed out by others. Secondly, we first model the cooperative behavior of SP and WP in a multi stage assignment process. Thirdly, the Soyster robust model is introduced to replace those scenario-based model for real time purpose. The model is an extension of the classical WTA problem and its NP-hardness has also been proved. In algorithm aspect, the multi-objective heuristic has not been widely studied currently. We present a novel and efficient heuristic framework based on the concept of ECR for multi-objective problems and improve MOEA/D with several modifications.

Some works need to be further studied in the future. We pay less attention to the uncertainty from the situational decision layer in this paper. This involves the uncertainty of the time window, which increases the complexity of this problem. It should be studied in future works. And the proposed heuristic initialization is very effective. Local selection of quaternions or parallel computing maybe studied to further decrease the time consumption. Since NMS can enhance the convergence, we could design feedback mechanisms to obtain better performance.

Declaration of Competing Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Credit authorship contribution statement

Wenqin Xu: Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **Chen Chen:** Conceptualization, Funding acquisition, Investigation, Methodology, Project administration, Supervision, Writing - review & editing. **Shuxin Ding:** Conceptualization, Formal analysis, Investigation, Methodology, Validation, Supervision, Writing - review & editing. **Panos M. Pardalos:**

Conceptualization, Funding acquisition, Investigation, Supervision, Writing - review & editing.

Acknowledgments

We thank the editors and anonymous reviewers for their helpful comments and suggestions on improving the presentation of this paper. This work was supported by the National Natural Science Foundation of China (NSFC) with Grant No. 61773066 and a provincial and ministerial project with Grant No. 61403120401. P. M. Pardalos is partially supported by the Paul and Heidi Brown Preeminent Professorship at ISE (University of Florida, USA) and a Humboldt Research Award (Germany).

Appendix A. Proof of Theorem 1

Proof. To prove it, We first convert our model into a single-objective formulation by limiting the maximum total costs using the ε-constraint method (Miettinen, 2012):

$$\begin{aligned} &\max \ \ F_1''(t), \\ &\text{s.t.} \ \ F_2(t) < C, (6), (7), (8), (9), (10), (11). \end{aligned} \tag{A.1}$$

By setting $Q_k''(s) = 1$, $d_j = 0$, and $N_{sik} = 0$ for all s, i, j, k, Problem (A.1) is reduced to the general WTA, which is a well-known NP-hard problem (Lloyd & Witsenhausen, 1986). Hence, WTA is a special case of Problem (A.1). When Problem (A.1) is solved, WTA is also solved, which illustrates that Problem (A.1) is also NP-hard.

With this prior knowledge, the proof is clear. If the bi-objective problem is not NP-hard, we can obtain a solution of each ε-constraint version in polynomial time, which contradicts with the observation that Problem (A.1) is NP-hard. Thus, the NP-hardness is proved. □

References

Afzalirad, M., & Rezaeian, J. (2017). A realistic variant of bi-objective unrelated parallel machine scheduling problem: NSGA-II and MOACO approaches. *Applied Soft Computing, 50*, 109–123.

Ahner, D. K., & Parson, C. R. (2015). Optimal multi-stage allocation of weapons to targets using adaptive dynamic programming. *Optimization Letters, 9*(8), 1689–1701.

Ahuja, R. K., Kumar, A., Jha, K. C., & Orlin, J. B. (2007). Exact and heuristic algorithms for the weapon-target assignment problem. *Operations Research, 55*(6), 1136–1146.

Alhindi, A., & Zhang, Q. (2014). MOEA/D with Tabu Search for multiobjective permutation flow shop scheduling problems. In *Evolutionary computation (CEC), 2014 ieee congress on* (pp. 1155–1164). IEEE.

Bertuccelli, L., Alighanbari, M., & How, J. (2004). Robust planning for coupled cooperative UAV missions. In *Decision and control, 2004. CDC. 43rd IEEE conference: 3* (pp. 2917–2922). IEEE.

Cai, X., Mei, Z., Fan, Z., & Zhang, Q. (2018). A constrained decomposition approach with grids for evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation, 22*(4), 564–577.

Chang, T., Kong, D., Hao, N., Xu, K., & Yang, G. (2018). Solving the dynamic weapon target assignment problem by an improved artificial bee colony algorithm with heuristic factor initialization. *Applied Soft Computing, 70*, 845–863.

Chen, J., Li, J., & Xin, B. (2017). DMOEA-εC: Decomposition-based multi-objective evolutionary algorithm with the ε-constraint framework. *IEEE Transactions on Evolutionary Computation*.

Chiang, T.-C., & Lai, Y.-P. (2011). MOEA/D-AMS: Improving MOEA/D by an adaptive mating selection mechanism. In *Evolutionary computation (CEC), 2011 ieee congress on* (pp. 1473–1480). IEEE.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *International conference on parallel problem solving from nature* (pp. 849–858). Springer.

Ding, S., Chen, C., Xin, B., & Pardalos, P. M. (2018). A bi-objective load balancing model in a distributed simulation system using NSGA-II and MOPSO approaches. *Applied Soft Computing, 63*, 249–267.

Fattahi, P., Hajipour, V., & Nobari, A. (2015). A bi-objective continuous review inventory control model: Pareto-based meta-heuristic algorithms. *Applied Soft Computing, 32*, 211–223.

Hosein, P. A., & Athans, M. (1990a). Preferential defense strategies. Part I: The static case. *Mit laboratory for information and decision systems with partial support*.

Hosein, P. A., & Athans, M. (1990b). Preferential defense strategies. Part II: The dynamic case. *Mit laboratory for information and decision systems with partial support*.

Ishibuchi, H., Masuda, H., Tanigaki, Y., & Nojima, Y. (2015). Modified distance calculation in generational distance and inverted generational distance. In *International conference on evolutionary multi-criterion optimization* (pp. 110–125). Springer.

Jian, W., & Chen, C. (2015). Sensor-weapon joint management based on improved genetic algorithm. In *Control conference (CCC), 2015 34th Chinese* (pp. 2738–2742). IEEE.

Jiang, S., & Yang, S. (2017). An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts. *IEEE Transactions on Cybernetics, 46*(2), 421–437.

Ke, L., Zhang, Q., & Battiti, R. (2013). MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and antcolony. *IEEE Transactions on Cybernetics, 43*(6), 1845–1859.

Khosla, D. (2001). Hybrid genetic approach for the dynamic weapon-target allocation problem. In *Battlespace digitization and network-centric warfare: 4396* (pp. 244–260). International Society for Optics and Photonics.

Kline, A., Ahner, D., & Hill, R. (2018). The weapon-target assignment problem. *Computers and Operations Research*.

Krokhmal, P., Murphey, R., Pardalos, P., Uryasev, S., & Zrazhevski, G. (2003). Robust decision making: Addressing uncertainties in distributions. In *Cooperative control: Models, applications and algorithms* (pp. 165–185). Springer.

Kwon, O., Kang, D., Lee, K., & Park, S. (1999). Lagrangian relaxation approach to the targeting problem. *Naval Research Logistics, 46*(6), 640–653.

Kwon, O., Lee, K., Kang, D., & Park, S. (2007). A branch-and-price algorithm for a targeting problem. *Naval Research Logistics, 54*(7), 732–741.

Li, J., Chen, J., & Xin, B. (2015). Efficiently solving multi-objective dynamic weapon–target assignment problems by NSGA-II. In *Control conference (CCC), 2015 34th Chinese* (pp. 2556–2561). IEEE.

Li, J., Chen, J., Xin, B., Dou, L., & Peng, Z. (2016). Solving the uncertain multi-objective multi-stage weapon target assignment problem via MOEA/D-AWA. In *Evolutionary computation (CEC), 2016 ieee congress on* (pp. 4934–4941). IEEE.

Li, Y., Kou, Y., & Li, Z. (2018). An improved nondominated sorting genetic algorithm III method for solving multiobjective weapon-target assignment part I: The value of fighter combat. *International Journal of Aerospace Engineering, 2018*.

Li, Y., Kou, Y., Li, Z., Xu, A., & Chang, Y. (2017). A modified Pareto ant colony optimization approach to solve biobjective weapon-target assignment problem. *International Journal of Aerospace Engineering, 2017*.

Lloyd, S., & Witsenhausen, H. (1986). Weapons allocation is NP-complete.. In *1986 summer computer simulation conference* (pp. 1054–1058).

Madni, A. M., & Andrecut, M. (2012). Efficient heuristic approach to the weapon-target assignment problem. *Journal of Aerospace Computing Information and Communication, 6*(6), 1–10.

Manne, A. S. (1958). A target-assignment problem. *Operations Research, 6*(3), 346–351.

Miettinen, K. (2012). Nonnonlinear multiobjective optimization. *Springer Science and Business Media, 12*.

Mousavi, S. M., Sadeghi, J., Niaki, S. T. A., & Tavana, M. (2016). A bi-objective inventory optimization model under inflation and discount using tuned Pareto-based algorithms: NSGA-II, NRGA, and MOPSO. *Applied Soft Computing, 43*, 57–72.

Qi, Y., Ma, X., Liu, F., Jiao, L., Sun, J., & Wu, J. (2014). MOEA/D with adaptive weight adjustment. *Evolutionary Computation, 22*(2), 231–264.

Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the first international conference on genetic algorithms and their applications, 1985*. Lawrence Erlbaum Associates. Inc., Publishers.

Shang, G., Zaiyue, Z., Xiaoru, Z., & Cungen, C. (2007). Immune genetic algorithm for weapon-target assignment problem. In *Intelligent information technology application, workshop on* (pp. 145–148). IEEE.

Soyster, A. L. (1973). Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research, 21*(5), 1154–1157.

Tan, Y.-Y., Jiao, Y.-C., Li, H., & Wang, X.-K. (2012). MOEA/D-SQA: A multi-objective memetic algorithm based on decomposition. *Engineering Optimization, 44*(9), 1095–1115.

Wang, J., & Cai, Y. (2015). Multiobjective evolutionary algorithm for frequency assignment problem in satellite communications. *Soft Computing, 19*(5), 1229–1253.

Wang, Z., Zhang, Q., Zhou, A., Gong, M., & Jiao, L. (2016). Adaptive replacement strategies for MOEA/D. *IEEE Transactions on Cybernetics, 46*(2), 474–486.

Xin, B., Chen, J., Zhang, J., Dou, L., & Peng, Z. (2010). Efficient decision makings for dynamic weapon-target assignment by virtual permutation and tabu search heuristics. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 40*(6), 649–662.

Xin, B., Wang, Y., & Chen, J. (2018). An efficient marginal-Return-Based constructive heuristic to solve the sensor-weapon-target assignment problem. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, (99), 1–12.

Xu, Y., & Mei, Y. (2018). A modified water cycle algorithm for long-term multi-reservoir optimization. *Applied Soft Computing, 71*, 317–332.

Yang, S., Li, M., Liu, X., & Zheng, J. (2013). A grid-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation, 17*(5), 721–736.

Bogdanowicz, Z. R., & Coleman, N. P. (2007). Sensor-target and weapon-target pairings based on auction algorithm. In *Proceedings of the 11th WSEAS international conference on applied mathematics* (pp. 92–96). World Scientific and Engineering Academy and Soc.(WSEAS) Stevens Point, WI.

Zapotecas-Martínez, S., Derbel, B., Liefooghe, A., Brockhoff, D., Aguirre, H. E., & Tanaka, K. (2015). Injecting CMA-ES into MOEA/D. In *Proceedings of the 2015 annual conference on genetic and evolutionary computation* (pp. 783–790). ACM.

Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation, 11*(6), 712–731.

Zhang, Q., Li, H., Maringer, D., & Tsang, E. (2010). MOEA/D with NBI-style Tchebycheff approach for portfolio management. In *Evolutionary computation (CEC), 2010 ieee congress on* (pp. 1–8). IEEE.

Zhang, Q., Liu, W., & Li, H. (2009). The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In *Evolutionary computation, 2009. cec'09. ieee congress on* (pp. 203–208). IEEE.