



An improved estimation of distribution algorithm for rescue task emergency scheduling considering stochastic deterioration of the injured

Ying Xu^{1,2} · Xiaobo Li³ · Qian Li² · Weipeng Zhang²

Received: 25 February 2023 / Accepted: 27 May 2023 / Published online: 25 July 2023
© The Author(s) 2023

Abstract

Efficient allocating and scheduling emergency rescue tasks are a primary issue for emergency management. This paper considers emergency scheduling of rescue tasks under stochastic deterioration of the injured. First, a mathematical model is established to minimize the average mathematical expectation of all tasks' completion time and casualty loss. Second, an improved multi-objective estimation of distribution algorithm (IMEDA) is proposed to solve this problem. In the IMEDA, an effective initialization strategy is designed for obtaining a superior population. Then, three statistical models are constructed, which include two tasks existing in the same rescue team, the probability of first task being processed by a rescue team, and the adjacency between two tasks. Afterward, an improved sampling method based on referenced sequence is employed to efficiently generate offspring population. Three multi-objective local search methods are presented to improve the exploitation in promising areas around elite individuals. Furthermore, the parameter calibration and effectiveness of components of IMEDA are tested through experiments. Finally, the comprehensive comparison with state-of-the-art multi-objective algorithms demonstrates that IMEDA is a high-performing approach for the considered problem.

Keywords Allocating · Scheduling · Stochastic deterioration · Estimation of distribution algorithm · Local search

Introduction

Natural and man-made disasters have significantly threatened people's production and life. Therefore, efficiently implementing rescue work is important to mitigate casualties and property losses. Related work demonstrates that rescue forces are insufficient in most disaster rescue sites [1]. Consequently, making full use of limited rescue resources to maximize relief efficiency and minimize damage is an essential issue for emergency response.

Here, we present a simple case of rescue task emergency scheduling problem. As illustrated in Fig. 1a, there are eight

rescue tasks marked in yellow circles and a workstation marked in green circle denoting the starting point of two rescue teams. The number on each edge indicates traveling time between two tasks, and the triple on each yellow circle denotes the initial injured state, number of injured, and task processing time (in minute) for its initial injured state. The initial injured state is graded on levels 1, 2, and 3, representing mild degree, moderate degree, and severe degree, respectively. Because the injured state may deteriorate over time, the worse injured state, the longer task processing time, and the higher treatment cost. In consequence, our objective is to work out task assignment and scheduling scheme by minimizing average mathematical expectation of all tasks' completion time and casualty loss simultaneously. The mathematical expectation of a task's completion time not only depends on the completion time of its preceding tasks, but also depends on the probability distribution of this task's injured state when a rescue team arrives. The mathematical expectation of a task's casualty loss, which indicates the cost for rescuing injured, is determined by the number of injured and the probability distribution of this task's injured state when a rescue team arrives.

✉ Xiaobo Li
lxb@zjnu.edu.cn

¹ Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, Zhejiang, China

² College of Digital Technology and Engineer, Ningbo University of Finance & Economics, Ningbo 315175, Zhejiang, China

³ School of Computer Science and Technology, Zhejiang Normal University, Jinhua 321004, Zhejiang, China

The traditional multi-objective programming algorithm could work out a feasible solution for this simple instance, as shown in Fig. 1b, where a task sequence 1, 6, 7, and 8 is assigned to a rescue team, and a sequence 3, 2, 5, and 4 is assigned to another team. However, the rescue operation usually involves a large number of tasks and response time is quite limited. Additionally, rescuers have a fatigue effect under the high load of rescue work and pressure, which may lead to an extension of processing time for the subsequent tasks. Therefore, it is necessary to conduct in-depth study on practical problem in the real world.

It is recognized that rescue task scheduling problem is NP-hard [2]; therefore, it takes a long time to solve this problem especially for high-dimensional instances with multiple objectives. Scholars established reasonable mathematical models and developed some multi-objective combinatorial methods to obtain high-quality solutions for disaster management and relief planning. In the following, we review some related studies for this multi-objective combinatorial optimization problem in rescue planning field.

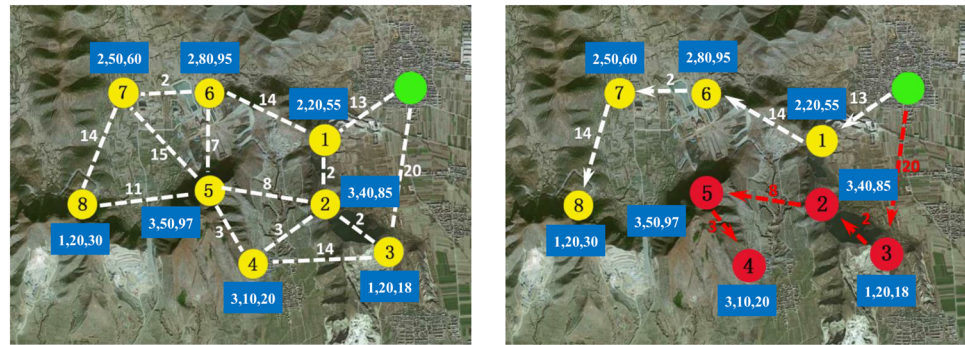
Because of good stability and superior performance, NSGA-II has been widely used and extended in disaster relief operation. For example, Zhang et al. [3] presented a multistage assignment model and three priority scheduling approaches for the emergency resource allocation and scheduling problem. The authors improved NSGA-II algorithm with an elitism strategy and preserving population diversity to adaptively solve the problem and obtain satisfactory results. Addressing task scheduling problem for multiple targets in disaster management, Niu et al. [4] employed NSGA-II algorithm to solve mathematical model. In this algorithm, the authors designed efficient initialization method, novel individual selection, and evolution strategy to evolve solutions. To enhance the efficiency of rescue attachments in rescue operations, Wang et al. [5] utilized NSGA-II algorithm by designing new crossover and mutation operators to optimize multi-objective of attachment, which achieves good convergence and population diversity in some test instances.

Due to the rapid development of meta-heuristic algorithms, the combination of NSGA-II and other algorithms has attracted great attention. Focusing on the emergency logistics scheduling problem, Chang et al. [6] proposed a greedy-search-based NSGA-II algorithm, in which the greedy method accelerates the convergence of algorithm, while NSGA-II guarantees the population diversity. The algorithm dynamically regulated the schedules according to the requirements aiming at minimizing unsatisfied demand for resources, delivery time, and transportation cost simultaneously. Addressing the multi-depot dynamic emergency scheduling problem in relief operation, Wan et al. [7] constructed a bi-objective dynamic vehicle routing model under practical constraints, and proposed a hybrid ant colony opti-

mization (ACO) algorithm to obtain candidate solutions. In the following study, the authors [8] established a multi-objective emergency material scheduling model based on the characteristics of problem, and combined ACO algorithm with NSGA-II to obtain solutions in a reasonable time. In the proposed algorithm, polygon circumcenter was adopted to ensure that neighboring tasks be grouped together, and variable neighborhood search method was employed to avoid trapping into local optima. Nayeri et al. [9] constructed an integer linear model to formulate rescue planning problem. This model considered rescue capability and fatigue effect of rescue team to make it more in accordance with the practical scenarios. Moreover, the author developed three heuristics to tackle this problem in a reasonable time. The authors [10] studied a multi-objective mathematical model in the following study to better simulate the real world and incorporated NSGA-II with particle swarm optimization (PSO) algorithm to obtain candidate solutions.

Some studies developed classical meta-heuristic algorithms, i.g. PSO, ACO, differential evolution (DE), Tabu, etc. and their integrations to solve multi-objective problems in this field. For example, considering multi-objective emergency shelter allocation problem under three different scenarios, Zhao et al. [11] improved the multi-objective PSO algorithm by applying different structure in two loops and obtained high-quality solutions in a real-world case study. Addressing location-routing problem in emergency management, Yan et al. [12] proposed multi-objective discrete particle swarm and Harris hawks optimization algorithm that includes novel learning and evolution strategies to pursue high-quality solutions. Wu et al. [13] presented a capacitated vehicle routing scheduling model and an improved hybrid algorithm for emergency material delivery problem, which considers timeliness and fairness simultaneously. The proposed algorithm utilized multi-objective ACO algorithm to pursue high-quality initial solutions and applied variable neighborhood search strategy to perform local search for obtaining optimal solutions. Considering unmanned aerial vehicle path planning problem in rescue operations, Yu et al. [14] modeled a multi-objective optimization formulation with some constraints, and proposed a knee point-based DE algorithm. The knee solution guided search direction and provided satisfied solutions for decision-makers. Wang et al. [15] constructed a dynamic multi-objective model for multi-period material emergency supply problem. The authors presented a dynamic multi-objective DE algorithm with a detect operator to response the constantly changing environment. In addition, an adaptive mutation method is adopted to enhance the exploitation and exploration ability. Li et al. [16] concentrated on the mobile emergency material allocation problem in disaster management, and employed a hybrid leapfrog algorithm to solve this problem, which utilizes the advantages of GA and PSO to strike a balance between

Fig. 1 A simple case of the considered problem



(a) Eight rescue tasks are distributed in disaster areas.

(b) A feasible solution for this case.

exploration and exploitation and efficiently evolve solutions toward Pareto front. Addressing the dynamic multi-objective material distribution in a post-disaster emergency scenario, Wan et al. [17] presented a hybrid approach which combines multi-objective salp swarm and sine–cosine algorithm. The proposed algorithm exhibited superior performance in solving some disaster events. Tian et al. [18] established a multi-objective model to formulate the vehicles scheduling in forest fire emergency services under certain constraints. The authors proposed a multi-objective discrete gravitational search approach with greedy selection strategy to obtain satisfied solutions in an actual forest fire rescue case. Addressing multi-objective rescue task emergency scheduling in relief operation, Zheng et al. [2] established a model considering the importance of tasks and fuzzy processing time, designed a multi-objective biogeography-based optimization algorithm with effective operators to evolve solutions toward Pareto front, which exhibits superior performance on a set of actual rescue cases. Zheng et al. [19] concentrated on an emergency equipment maintenance scheduling problem aiming at balancing multiple objectives related to cost and maintenance risks, etc., and developed an efficient solution encoding method and multi-objective Tabu algorithm to solve this problem.

Moreover, multi-objective evolutionary algorithm based on decomposition (MOEA/D) has been employed for the rescue work in recent years. Gharib et al. [20] applied MOEA/D and NSGA-II for planning the material delivery in a post-disaster reconstruction project. The experiment showed that both algorithms have their own advantages. MOEA/D achieved higher accuracy, while NSGA-II took shorter time. This study could provide an integrated scheme for decision-makers to formulate a detailed rescue strategy. Zhou et al. [21] improved MOEA/D for the multi-period dynamic emergency resource scheduling problem. In this algorithm, the authors designed efficient population initialization method, and developed new operators and weight neighborhood selection strategy based on characteristics of

this problem. The experiment illustrated that the proposed algorithm outperforms NSGA-II for some instances.

Estimation of distribution algorithm (EDA) is an optimization heuristic based on statistical theory, which extracts information from superior individuals to establish explicit probabilistic model [22]. Unlike swarm intelligence optimization algorithm, which uses interaction or mutation operator to generate offspring individuals. EDA produces offspring through sampling from probabilistic models. Then, these offspring individuals are incorporated into the current population. This process continues until stop criterion is satisfied. The general framework of basic EDA is shown in Pseudocode 1. Because EDA is good at mining the regularities of problems, and has good global exploration ability, it has been employed in real-world scenarios, such as engineering design [23–25], financial optimization [26, 27], network optimization [28–30], and clustering [31, 32], and has exhibited superior performance.

Algorithm 1 Basic EDA

Input: initial population

Output: the best individual

```

1: while the stop criterion is not satisfied do
2:   Select superior individuals from current population.
3:   Establish explicit probabilistic model from the selected superior individuals.
4:   Sample probabilistic model to generate offspring individuals.
5:   Incorporate offspring individuals into current population.
6: end while

```

It is noted that EDA has also been applied in multi-objective combinatorial optimization problem in recent years. The essential idea is to design reasonable representation of solutions first, then construct probability models based on the intrinsic properties of considered problem, generate offspring individuals through sampling from these models, and utilize multi-objective individual selection strategy to select superior individuals to update probability models. For example, Wang et al. [33] presented a multi-objective esti-

mation of distribution algorithm for solving mixed-variable problems. The algorithm utilized index coding method to deal with discrete variables, constructed probability model, and population diversity maintenance to enhance search ability. Jiang et al. [34] proposed a domain adaptation and non-parametric estimation-based estimation of distribution algorithm for solving dynamic multi-objective optimization problem, which makes full use of critical features to construct probability model and sampling method. Moreover, the utilization of Monte Carlo method and transfer learning algorithm were used to enhance the searching ability.

Since our study focuses on the rescue task allocation and scheduling problem, here we describe some researches of EDA be applied to multi-objective scheduling problems. Shao et al. [35] presented a Pareto-based EDA for multi-objective distributed no-wait flow-shop scheduling problem, which builds probabilistic models to describe positional relationship between jobs and factories, and designs effective local search methods to enhance exploitation ability. Focusing on the multi-objective flexible job-shop problem, Wang et al. [36] developed a Pareto-based EDA that constructs a probability model to estimate probability distribution of solution space, and divides the population into two sub-population for generating neighbor individuals by applying different operators, designs a local search strategy based on the critical path to optimize quality of individuals. Considering a three-stage multi-objective integrated scheduling problem, Deng et al. [37] developed a hybrid EDA with variable neighborhood search that includes an effective encoding/decoding strategy and speed scheduling scheme, and designed two local search methods to improve the quality of solutions. Shao et al. [38] developed EDA to solve blocking flow-shop scheduling problem. This algorithm constructed a probability model to describe the distribution of superior solutions, applied path relinking strategy to accelerate convergence of algorithm, and introduced local search as well as diversity maintenance technique to enhance search efficiency.

From the aforementioned investigation, we consider that some studies can be conducted in the following directions:

- (1) Although related researches have established mathematical models for describing rescue operations. A few studies considered the dynamic evolution of rescue tasks as well as the geographical proximity of tasks processed by a rescue team in their proposed algorithms. In fact, the traumatic condition of injured might deteriorate due to the untimely rescue in the practical scenario, which leads to an aggravation of casualty loss to a certain degree. The severity of a rescue task is closely related to its geographical location due to characteristics of natural disasters, and tasks rescued by a team are most likely to be adjacent. It might be a beneficial attempt to emphasize the stochastic deterioration of the injured and utilize the geographical

relevance of tasks to make this study more consistent with the practical situation of disaster relief.

- (2) Some multi-objective combinatorial optimization algorithms have been proposed to improve the quality of solutions or reduce response time. However, more studies need to obtain satisfactory solutions for different type of rescue operation problems. EDA has a characteristic of no algorithm-specific parameters, which facilitates its adaptation for solving combinatorial optimization problems. The key idea is to construct probability models that describe task scheduling order, sampling method, and evolutionary mechanism without designing calculation form of parameters. In addition, EDA is easy to combine with other strategies to improve its performance, which is an effective strategy in adaptation process.

Inspired by above studies on practical rescue operation, this paper proposes a rescue task emergency scheduling problem considering stochastic deterioration of the injured. The main contributions and originalities are twofold:

- (1) A mathematical model considering stochastic deterioration of the injured is constructed according to the practical scenario of task allocation and scheduling in a rescue operation, whose objectives are minimization of expected task completion time and casualty loss simultaneously.
- (2) An improved multi-objective estimation of distribution algorithm is proposed, which includes a novelty initialization strategy, three statistical models and an efficient sampling approach to generate offspring population, three multi-objective local search methods to enhance the quality of elite individuals. Our proposed algorithm obtains better results than other considered multi-objective optimization algorithms for this problem.

In the rest of the paper, Sect. [The rescue task emergency scheduling problem considering stochastic deterioration of the injured](#) formulates problem description and modeling. Section [Proposed algorithm](#) presents the proposed algorithm. Section [Computational experiment](#) calibrates parameters of IMEDA and presents computational experiments. Section [Conclusion](#) concludes this paper.

The rescue task emergency scheduling problem considering stochastic deterioration of the injured

Problem description

As illustrated in Zheng et al. [2], multiple rescue tasks are distributed in disaster area and waiting to be rescued by engineering rescue teams. The goal is to determine task allocation and scheduling scheme with the aim of minimizing both the

average mathematical expectation of all tasks' completion time and casualty loss. Table 1 lists the notations of considered problem.

Generally speaking, when a disaster occurs, rescue teams start from workstation and drive to affected areas to implement rescue activities. The capabilities of rescue teams are different for performing a task. Under the continuous high-intensity working condition, rescuers are tired and require more time for subsequent tasks, which can be known as the fatigue effect of rescue team [9]. In addition, the injured state may deteriorate over time before well-treatment, which leads to an extension of task rescue time. Based on these assumptions of this problem, the mathematical model is formulated as follows:

$$\min(f_1, f_2) \quad f_1 = \frac{1}{m} \sum_{j=1}^m E(c_j) \quad f_2 = \frac{1}{m} \sum_{j=1}^m E(loss_j) \quad (1)$$

$$\text{subject to: } \pi_i \neq \emptyset, \forall i \quad (2)$$

$$\sum_{l=1}^{n_i} \sum_{i=1}^n X_{j,l,i} = 1, \forall j \quad (3)$$

$$\left\{ \begin{array}{ll} p_{j4t_j^{arr}} = 1 & \text{if } \sum_{\lambda=s_j^{ini}}^3 t_{\lambda(\lambda+1)}^{swi} - t_j^{arr} < 0 \\ p_{j\lambda^* t_j^{arr}} = 1 - \frac{t_j^{arr}}{t_{\lambda^*(\lambda^*+1)}^{swi}}, \quad p_{j(\lambda^*+1)t_j^{arr}} = 1 - p_{j\lambda^* t_j^{arr}} & \text{else if } \sum_{\lambda=s_j^{ini}}^{\lambda'} t_{\lambda(\lambda+1)}^{swi} - t_j^{arr} \geq 0, \text{ and } \lambda^* = s_j^{ini} \\ p_{j\lambda^* t_j^{arr}} = 1 - \frac{t_j^{arr} - \sum_{\lambda=s_j^{ini}}^{\lambda^*-1} t_{\lambda(\lambda+1)}^{swi}}{t_{\lambda^*(\lambda^*+1)}^{swi}}, \quad p_{j(\lambda^*+1)t_j^{arr}} = 1 - p_{j\lambda^* t_j^{arr}} & \text{else if } \sum_{\lambda=s_j^{ini}}^{\lambda'} t_{\lambda(\lambda+1)}^{swi} - t_j^{arr} \geq 0, \text{ and } \lambda^* > s_j^{ini} \end{array} \right. \quad (4)$$

where λ^* be the minimum λ' satisfying the inequality $\sum_{\lambda=s_j^{ini}}^{\lambda'} t_{\lambda(\lambda+1)}^{swi} - t_j^{arr} \geq 0$, where $\lambda' \leq 3$

$$E(c_j) = \sum_{\lambda=s_j^{ini}}^4 t_{ij}^{pro} (1 + \alpha_j)^{\lambda-s_j^{ini}} (1 + \delta_i)^{r_j-1} p_{j\lambda t_j^{arr}}, \forall j, \quad (5)$$

$$\text{where } t_j^{arr} = \begin{cases} t_{0j}^{tra} & r_j = 1 \\ E(c_{\pi_i(r_j-1)}) + t_{\pi_i(r_j-1)j}^{tra} & r_j = 2, 3, \dots, n_i \end{cases}$$

$$E(loss_j) = \sum_{\lambda=s_j^{ini}}^4 \theta_j \psi_\lambda p_{j\lambda t_j^{arr}}, \forall j, \quad (6)$$

$$\text{where } t_j^{arr} = \begin{cases} t_{0j}^{tra} & r_j = 1 \\ E(c_{\pi_i(r_j-1)}) + t_{\pi_i(r_j-1)j}^{tra} & r_j = 2, 3, \dots, n_i \end{cases}$$

This model is formulated as a bi-objective optimization problem, which aims to minimize average mathematical expectation of task completion time as well as casualty loss,



Fig. 2 The deterioration process of injured state

as shown in objective function (1). It is widely known that efficient completion of rescue work is important for saving lives and mitigating damage. Meanwhile, the second objective reflects both the cost of rescue operation and trauma sustained by injured. Equations (2) ensure that each team should implement rescue tasks. Equations (3) guarantee that each task should be performed by only one rescue team. The state of injured may deteriorate over time before rescuers arrive at the disaster site. This deterioration process conforms to the no aftereffect of Markov process, which is separated into four finite state inhomogeneous Markov processes: minor injury (MI), moderate injury (MO), severe injury (SE), and death (DE), as shown in Fig. 2. The deterioration process ends when rescue team arrives at the task and starts the rescue work. Chu et al. [39] assumed that the severity of injured is linearly related to the delay time before they are well treated. When a rescue team reaches the task j , the injured state probability distribution of this task is calculated as Eq. (4), which is decided by the initial state of injured, arriving time of rescue team and total transition time of two adjacent injured states. The first branch of Eq. (4) represents that the injured are in the last state of Markov processes; the second branch illustrates that when rescue team arrives, the injured state of task j is in the transition process from its initial state to the next one; The third branch denotes that when rescue team arrives, the injured state of task j has completely transitioned from its initial state to subsequent state. Equation (5) defines mathematical expectation for the completion time of task j . Taking into account the extension of working hours caused by the fatigue factor of rescue team, as well as task processing time in the different injured states, the processing time of task j with injured state λ for rescue team i is computed as $t_{ij}^{pro} (1 + \alpha_j)^{\lambda-s_j^{ini}} (1 + \delta_i)^{r_j-1}$. Equation (6) formulates the mathematical expectation for casualty loss of the task j , where $\theta_j \psi_\lambda$ denotes the cost of rescuing task j with the injured state λ .

Multi-objective optimization problem

The multi-objective optimization problem aims to optimize multiple objectives simultaneously through searching for solutions satisfying equality and inequality constraints. The general mathematical model for minimization problem is stated as follows:

$$\min f(x) = \min(f_1(x), f_2(x), \dots, f_\eta(x)) \quad (7)$$

Table 1 Notations and variables

Notation	Description
Indexes	
n	The number of rescue teams
m	The number of rescue tasks
λ	The index for the injured state where $\lambda = 1, 2, 3, 4$ representing minor injury, moderate injury, severe injury and death, respectively
n_i	The number of rescue tasks processed by the team i
r_j	The position of the task j in the schedule of team i , where $r_j = 1, 2, \dots, n_i$
Parameters	
s_j^{ini}	The initial injured state of the task j
θ_j	The number of injured in the task j
ψ_λ	The cost of rescuing per injured in the state λ
$t_{\lambda(\lambda+1)}^{swi}$	The time it takes for the injured to transit from state λ to $\lambda+1$ completely
t_{0j}^{tra}	The travel time from workstation to task j , workstation is the starting point of rescue teams
$t_{jj'}^{tra}$	The travel time from the task j to j'
t_{ij}^{pro}	The processing time of task j in its initial injured state for the rescue team i
δ_i	The fatigue factor for the rescue team i
α_j	The state coefficient of rescue task j
Decision variables	
π_i	The task scheduling sequence of team i , where $\pi_i = \{\pi_i(1), \pi_i(2), \dots, \pi_i(n_i)\}$
π	A solution $\pi = \{\pi_1, \pi_2, \dots, \pi_i, \dots, \pi_n\}$, where $\pi_i \in \pi$
$X_{j,l,i}$	If $\pi_i(l) = j$, then $X_{j,l,i} = 1$; otherwise $X_{j,l,i} = 0$
t_j^{arr}	The time when rescue team arrives at the task j
$p_{j\lambda t_j^{arr}}$	The probability that the rescue task j is in the injured state λ when rescue team reaches the task at time t_j^{arr}
$E(loss_j)$	The mathematical expectation for the casualty loss of task j
$E(c_j)$	The mathematical expectation for the completion time of task j

$$\text{s.t. } \begin{cases} g_i(x) \geq 0 & i = 1, 2, \dots, \theta \\ h_j(x) = 0 & j = 1, 2, \dots, p, \end{cases} \quad (8)$$

where $f_l(x)$ is the l -th subobjective function, $x \in \Omega$ is a solution, and Ω denotes the domain of definition. Moreover, some essential concepts about Pareto optimality are explained here [40].

- (1) Pareto dominance: A feasible solution x is said to dominate a feasible solution y (denoted as $x < y$) if and only if $f_i(x) \leq f_i(y)$, $\forall i \in \{1, 2, \dots, \eta\} \wedge f_j(x) < f_j(y)$, $\exists j \in \{1, 2, \dots, \eta\}$.
- (2) Pareto-optimal solution: A solution x is said to be a Pareto optimal solution if there is no solution that dominates x .

A set including all Pareto-optimal solutions is known as Pareto-optimal set; the corresponding objective function of Pareto-optimal set is defined as Pareto front.

Proposed algorithm

In our proposed algorithm, an initialization method is designed considering both the geographic proximity of tasks and casualty loss. Then, three probabilistic models are established, where a probabilistic model is used for determining the team that a task is allocated to, and other two models are used for determining the position of this task in the team. Next, a referenced sequence-based sampling method is applied to generate offspring. Since Pan and Ruiz [41] have pointed out the local exploitation ability of EDA is limited, we develop three multi-objective local search method to improve local exploitation further.

Solution representation and initialization

The solution representation includes n lists $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$, and each list $\pi_i = \{\pi_i(1), \pi_i(2), \dots, \pi_i(n_i)\}$ has a partial permutation denoting the processing sequence of tasks for a rescue team.

It is known that the quality of initial solutions has a profound effect on the effectiveness of evolutionary algorithm for solving the scheduling problem. An efficient initialization strategy helps the algorithm start searching from multiple promising solutions and improves the search efficiency of subsequent programs to a certain extent [42]. In this paper, the initialization has two procedures: The first step is determining the first task for each rescue team. A task sequence σ^{L_1} is generated by sorting all tasks in descending order according to the casualty loss value, and then, the first n tasks are successively inserted into the team that can make them complete the earliest. In the second step, the earliest completion team i^* and the completion time t^* of its current task are computed. Afterward, a sequence σ^D is produced by sorting the unscheduled tasks in ascending distance from the current position of team i^* , and a sequence σ^{L_2} is generated by sorting the unscheduled tasks in descending casualty loss at the time t^* . Next, weighted sum of unscheduled task position value $\varphi_j = w_1 \times \tau_j(\sigma^D) + w_2 \times \tau_j(\sigma^{L_2})$, $j \in \{\text{unscheduled tasks}\}$ are calculated. $\tau_j(\sigma^D)$ and $\tau_j(\sigma^{L_2})$ represent position values of task j in σ^D and σ^{L_2} , respectively. (w_1, w_2) is weighted vector for the l -th solution, i.e., $(\frac{l-1}{N-1}, \frac{N-l}{N-1})$, where N is population size. Finally, the task with the minimum value φ_j is extracted and inserted into the end of team i^* . The second step is repeated until all tasks have been assigned. The pseudo-code of initialization is detailed in Pseudocode 2.

Algorithm 2 Initialization

Input: population size N , parameters in Table 1

Output: The initial population POP

```

1: Let  $l \leftarrow 1$ .
2: while  $l \leq N$  do
3:   Let  $\pi \leftarrow \emptyset$ .
4:   Generate a task sequence  $\sigma^{L_1}$  by sorting tasks in descending
   casualty loss value at the initial state.
5:   The first  $n$  tasks of  $\sigma^{L_1}$  are successively inserted into the team of
    $\pi$  that can make them complete the earliest.
6:   while it exists unscheduled tasks do
7:     Compute the earliest completion team  $i^*$  and the completion
     time  $t^*$  of its current task.
8:     Generate a sequence  $\sigma^D$  by sorting the unscheduled tasks in
     ascending distance from the current position of team  $i^*$ .
9:     Generate a sequence  $\sigma^{L_2}$  by sorting the unscheduled tasks in
     descending casualty loss at the time  $t^*$ .
10:    Set a weight vector  $(w_1, w_2) \leftarrow (\frac{l-1}{N-1}, \frac{N-l}{N-1})$ .
11:    Calculate  $\varphi_j \leftarrow w_1 \times \tau_j(\sigma^D) + w_2 \times \tau_j(\sigma^{L_2})$ , where  $j \in$ 
     $\{\text{unscheduled tasks}\}$ ,  $\tau_j(\sigma^D)$  and  $\tau_j(\sigma^{L_2})$  represent the position of
    task  $j$  in  $\sigma^D$  and  $\sigma^{L_2}$ .
12:    Take the task with the minimum value  $\varphi_j$  and insert it into
    the team  $i^*$  of  $\pi$ .
13:  end while
14:   $POP(l) \leftarrow \pi, l \leftarrow l + 1$ .
15: end while

```

Probabilistic model

EDA generates offspring population via sampling from probabilistic models. Therefore, it is important to establish reasonable probabilistic models and updating mechanism to guide exploration in the decision space. According to the characteristics of problem and sampling method, we need to calculate the team that a task is allocated to and its position in this team. Consequently, we design three probabilistic models to describe the position relationships between rescue teams and tasks, i.e., two tasks processed by same team PS , the first task of each team PF , and two adjacent tasks PA . The following paragraph details the probabilistic models.

- (1) The probability PS that two tasks are processed by the same team, denoted as (9), where $ps_{j,j'}$ is the probability of task j and j' in the same team, computed as (10). In this equation, $\sum_{l=1}^{SSize} IS_{j,j'}(l) / \sum_{j'=1}^m \sum_{l=1}^{SSize} IS_{j,j'}(l)$ gathers the frequency information that task j and j' belong to the same team for all superior solutions, and $(1 - dis_{j,j'} / \sum_{j'=1, j' \neq j}^m dis_{j,j'})$ considers the distance between two tasks, that is, the closer two tasks are, the more likely they are to be processed by the same team. $IS_{j,j'}(l)$ is an indicator function representing whether the task j and j' are belonging to the same team, and $dis_{j,j'}$ is the distance between two tasks

$$PS = \begin{pmatrix} ps_{1,1} & ps_{1,2} & \dots & ps_{1,m} \\ ps_{2,1} & ps_{2,2} & \dots & ps_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ ps_{m,1} & ps_{m,2} & \dots & ps_{m,m} \end{pmatrix} \quad (9)$$

$$ps_{j,j'} = \begin{cases} w_{ps} \frac{\sum_{l=1}^{SSize} IS_{j,j'}(l)}{\sum_{j'=1}^m \sum_{l=1}^{SSize} IS_{j,j'}(l)} + \frac{(1-w_{ps})}{m-2} \left(1 - \frac{dis_{j,j'}}{\sum_{j'=1, j' \neq j}^m dis_{j,j'}} \right) & \text{if } j \neq j' \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

$$IS_{j,j'}(l) = \begin{cases} 1, & \text{the task } j \text{ and } j' \text{ are in the same team for the } l - \text{th solution in } SS \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

- (2) The probability PF of first task in each team, denoted as (12), where $pfi_{i,j}$ is the probability that task j is in the first position of team i , calculated as (13). $\sum_{l=1}^{SSize} IF_{i,j}(l) / SSize$ denotes the frequency of task j appears in the first position of team i for all superior solutions, and $s_j^{ini} / \sum_{j=1}^m s_j^{ini}$ considers the initial injured state of each rescue task, because tasks with more seriously injured are more likely to be prioritized. $IF_{i,j}(l)$ is an indicator function indicating whether the task j is the first one to be rescued in team i for the l -th solution, and

$SSize$ represents the size of superior solutions (denoted as SS) used for statistics

$$PF = \begin{pmatrix} pf_{1,1} & pf_{1,2} & \cdots & pf_{1,m} \\ pf_{2,1} & pf_{2,2} & \cdots & pf_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ pf_{n,1} & pf_{n,2} & \cdots & pf_{n,m} \end{pmatrix} \quad (12)$$

$$pf_{i,j} = w_{pf} \frac{\sum_{l=1}^{SSize} IF_{i,j}(l)}{SSize} + (1 - w_{pf}) \frac{s_j^{ini}}{\sum_{j=1}^m s_j^{ini}} \quad (13)$$

$$IF_{i,j}(l) = \begin{cases} 1, & j \text{ is the first task of team } i \text{ for the } l\text{-th solution in } SS \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

- (3) The probability PA that two tasks are next to each other in a team, denoted as (15), where $pa_{j,j'}$ is the probability of task j' appearing immediately after task j , computed as (16). Similarly, $\sum_{l=1}^{SSize} IA_{j,j'}(l) / \sum_{j'=1}^m \sum_{l=1}^{SSize} IA_{j,j'}(l)$ represents the statistical information obtained from superior solutions, describing the adjacent frequency of task j and j' . $(1 - dis_{j,j'} / \sum_{j'=1}^m dis_{j,j'})$ takes account of the

distance between two tasks as well. Generally, two tasks performed next to each other are geographically adjacent. $IA_{j,j'}(l)$ is an indicator function denoting whether the task j' follows task j immediately:

$$PA = \begin{pmatrix} pa_{1,1} & pa_{1,2} & \cdots & pa_{1,m} \\ pa_{2,1} & pa_{2,2} & \cdots & pa_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ pa_{m,1} & pa_{m,2} & \cdots & pa_{m,m} \end{pmatrix} \quad (15)$$

$$pa_{j,j'} = \begin{cases} w_{pa} \frac{\sum_{l=1}^{SSize} IA_{j,j'}(l)}{\sum_{j'=1}^m \sum_{l=1}^{SSize} IA_{j,j'}(l)} + \frac{(1-w_{pa})}{m-2} \left(1 - \frac{dis_{j,j'}}{\sum_{j'=1}^m dis_{j,j'}} \right) & \text{if } j \neq j' \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

$$IA_{j,j'}(l) = \begin{cases} 1, & \text{the task } j' \text{ appears immediately after the task } j \\ & \text{for the } l\text{-th solution in } SS \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

PS , PF , and PA are adjusted in each generation to search for the promising areas. Here, Heb rule is adopted to update probabilistic models, described as (18)–(20), where γ is used to control the learning rate

$$PS(t+1) = (1 - \gamma)PS(t-1) + \gamma PS(t) \quad (18)$$

$$PF(t+1) = (1 - \gamma)PF(t-1) + \gamma PF(t) \quad (19)$$

$$PA(t+1) = (1 - \gamma)PA(t-1) + \gamma PA(t). \quad (20)$$

Sampling method

Sampling from probabilistic models is crucial to generating high-quality solutions. Salhi [43] proposed that occasionally the solution directly samples from the models is not a completely representation of the current probabilistic distribution, which results in the low sampling efficiency. To overcome this drawback, we develop a sampling method referring to literature [44–46]. The core components are detailed as follows:

- (1) A high-quality solution is selected as the referenced sequence individual.
- (2) d tasks are extracted by roulette wheel selection method according to the $E(loss_j)(j = 1, 2, \dots, m)$. The task with higher expected casualty loss is more likely to be selected, and then, the rest tasks form a partial sequence.
- (3) These selected tasks are successively inserted into the partial sequence based on the probabilistic models. More precisely, PS is utilized to identify the team that a task is allocated to, and then, PF and PA are applied to determine the position of this task in the assigned team.

The pseudo-code of sampling with referenced sequence is shown in Pseudocode 3.

Algorithm 3 Sampling with referenced sequence

Input: d , a referenced sequence solution π_r , PF , PS , PA

Output: a new solution π_{new}

- 1: Select d tasks (denoted as $ta_1, ta_2, \dots, ta_j, \dots, ta_d$) using a roulette wheel selection method according to the $E(loss_j)$.
- 2: $\pi \leftarrow$ remove ta_j ($j = 1, 2, \dots, d$) from π_r .
- 3: **for** each task ta_j **do**
- 4: **for** each team i **do** // Lines 4–8 compute the team which ta_j is allocated to.
- 5: $avg \leftarrow \sum_{l=1}^{n_i} PS(ta_j, \pi_i(l)) / n_i$.
- 6: Calculate the probability of ta_j in the team i , $p_{team}(i) \leftarrow avg^{n_i}$.
- 7: **end for**
- 8: $i^* \leftarrow \arg \max_i p_{team}(i)$. // The task ta_j is allocated to the team i^* .
- 9: Let $p_{position}(1) \leftarrow PF(i^*, ta_j)$. // Lines 9–13 compute the position of ta_j in team i^* .
- 10: **for** $l = 2$ to $n_{i^*} + 1$ **do**
- 11: $p_{position}(l) \leftarrow PA(\pi_{i^*}(l-1), ta_j)$
- 12: **end for**
- 13: $\pi_{i^*}(l^*) \leftarrow ta_j$, where $l^* \leftarrow \arg \max_l p_{position}(l)$. // Insert the task ta_j into the position with the maximum $p_{position}$ of the team i^* in π .
- 14: **end for**

To better explain the sampling phase, a simple example is illustrated as follows. Suppose there are three rescue teams and ten rescue tasks. Five high-quality solutions in Table 2 are selected to generate probability models. The normalized

Table 2 High-quality solutions

$\pi_1 = \{\{6, 3, 7\}, \{8, 5, 1\}, \{2, 4, 9, 10\}\}$
$\pi_2 = \{\{6, 1, 7, 4\}, \{9, 5, 8\}, \{3, 10, 2\}\}$
$\pi_3 = \{\{2, 10, 8\}, \{9, 1, 5\}, \{7, 6, 3, 4\}\}$
$\pi_4 = \{\{2, 10, 4, 5\}, \{3, 8, 7\}, \{1, 6, 9\}\}$
$\pi_5 = \{\{5, 2, 1\}, \{7, 6, 8, 9\}, \{10, 3, 4\}\}$

matrix *inistatus* denotes initial injured state of each rescue task. The matrix *dis* represents the distance between two tasks. Without loss of generality, parameters w_{ps} , w_{pf} , and w_{pa} are set to 0.5. Then, the probability models *PS*, *PF*, and *PA* are calculated according to Eqs. (10, 13, 16), respectively.

$inistatus = [0.11, 0.01, 0.14, 0.16, 0.12, 0.13, 0.13, 0.07, 0.11, 0.03]$

$dis =$	0	0.82	1.06	0.34	0.67	0.72	0.60	0.80	0.65	0.82
	0.82	0	0.78	0.49	0.32	1.16	0.83	0.36	0.19	0.06
	1.06	0.78	0	0.92	0.53	0.82	0.56	0.42	0.85	0.84
	0.34	0.49	0.92	0	0.42	0.89	0.64	0.57	0.31	0.48
	0.67	0.32	0.53	0.42	0	0.85	0.52	0.14	0.33	0.37
	0.72	1.16	0.82	0.89	0.85	0	0.33	0.89	1.08	1.19
	0.60	0.83	0.56	0.64	0.52	0.33	0	0.56	0.77	0.87
	0.80	0.36	0.42	0.57	0.14	0.89	0.56	0	0.43	0.42
	0.65	0.19	0.85	0.31	0.33	1.08	0.77	0.43	0	0.17
	0.82	0.06	0.84	0.48	0.37	1.19	0.87	0.42	0.17	0

$PS =$	0	0.10	0.05	0.10	0.19	0.15	0.10	0.10	0.15	0.05
	0.09	0	0.09	0.14	0.14	0.05	0.05	0.10	0.10	0.23
	0.05	0.10	0	0.14	0.06	0.15	0.19	0.10	0.05	0.15
	0.09	0.13	0.12	0	0.09	0.12	0.13	0.06	0.09	0.16
	0.19	0.15	0.05	0.10	0	0.05	0.05	0.15	0.15	0.10
	0.13	0.05	0.13	0.13	0.06	0	0.21	0.09	0.13	0.05
	0.09	0.05	0.17	0.13	0.05	0.21	0	0.13	0.09	0.05
	0.10	0.10	0.10	0.05	0.15	0.10	0.15	0	0.15	0.10
	0.14	0.10	0.05	0.10	0.14	0.13	0.09	0.14	0	0.10
	0.05	0.23	0.14	0.18	0.10	0.05	0.05	0.10	0.10	0

$PF =$	0.06	0.20	0.07	0.08	0.16	0.26	0.06	0.03	0.06	0.01
	0.06	0.00	0.17	0.08	0.06	0.06	0.16	0.13	0.26	0.01
	0.16	0.10	0.17	0.08	0.06	0.06	0.16	0.03	0.06	0.11

$PA =$	0	0.05	0.05	0.06	0.22	0.22	0.22	0.05	0.06	0.05
	0.18	0	0.05	0.18	0.06	0.05	0.05	0.06	0.06	0.31
	0.05	0.06	0	0.25	0.06	0.06	0.16	0.16	0.05	0.15
	0.06	0.06	0.05	0	0.31	0.05	0.05	0.06	0.31	0.06
	0.22	0.22	0.05	0.06	0	0.05	0.05	0.23	0.06	0.06
	0.16	0.05	0.26	0.06	0.06	0	0.06	0.16	0.15	0.05
	0.06	0.05	0.06	0.22	0.06	0.39	0	0.06	0.05	0.05
	0.05	0.05	0.06	0.05	0.23	0.05	0.22	0	0.22	0.06
	0.22	0.06	0.05	0.06	0.22	0.05	0.05	0.06	0	0.23
	0.05	0.19	0.18	0.18	0.06	0.05	0.05	0.18	0.06	0

Then, the sampling procedure is described as follows:

- (1) Suppose that π_1 is the referenced sequence.
- (2) For simplicity, let $d = 1$ and the task 9 is extracted from π_1 using roulette wheel selection method based on expected casualty loss. Then the partial sequence is $\{\{6, 3, 7\}, \{8, 5, 1\}, \{2, 4, 10\}\}$.
- (3) This step is computing the team that task 9 is allocated to. According to the Pseudocode 3, we have

$$p_{\text{team}}(1) = ((PS(9, 2) + PS(9, 3) + PS(9, 7))/3)^3 \\ = (0.13 + 0.05 + 0.09)^3 = 0.0197$$

$$p_{\text{team}}(2) = ((PS(9, 8) + PS(9, 5) + PS(9, 1))/3)^3 \\ = (0.14 + 0.14 + 0.14)^3 = 0.0741$$

$$p_{\text{team}}(3) = ((PS(9, 2) + PS(9, 4) + PS(9, 10))/3)^3 \\ = (0.1 + 0.1 + 0.1)^3 = 0.027.$$

Since $p_{\text{team}}(2) > p_{\text{team}}(1)$ and $p_{\text{team}}(2) > p_{\text{team}}(3)$, the task 9 is allocated to the team 2.

- (4) This step is computing the position of task 9 in team 2

$$p_{\text{position}}(1) = PF(2, 9) = 0.26$$

$$p_{\text{position}}(2) = PA(8, 9) = 0.22$$

$$p_{\text{position}}(3) = PA(5, 9) = 0.06$$

$$p_{\text{position}}(4) = PA(1, 9) = 0.06.$$

Because $p_{\text{position}}(1) > p_{\text{position}}(i), i \neq 1$, the task 9 is inserted into the first position of team 2.

We can obtain a new solution $\pi_{\text{new}} = \{\{6, 3, 7\}, \{9, 8, 5, 1\}, \{2, 4, 10\}\}$.

Local search

It is recognized that efficient local search method is conducive to enhancing the exploitation of evolutionary algorithm. In this section, we present three property-based multi-objective local search methods for improving the quality of solutions, including insert and swap operators to adjust the team assignment and scheduling sequence in a team. For a solution that requires local intensification, the search strategies are described as follows:

- (1) *task_insert* ($LS_1(\pi_s, j)$): The task j is extracted and re-inserted into all positions of its team in π_s , excluding the original position. During the insertion procedure, all generated solutions are returned.
- (2) *team_insert* ($LS_2(\pi_s, j)$): The task j is extracted and re-inserted into the forward and backward of its k -nearest neighborhood tasks in other teams of π_s . During the insertion procedure, all generated solutions are returned.
- (3) *team_swap* ($LS_3(\pi_s, j)$): The task j is swapped with its k -nearest neighborhood tasks in π_s . During the swap procedure, all generated solutions are returned.

The aforementioned local search methods are applied to this multi-objective evolutionary algorithm to optimize elite solutions. To be specific, two stages are involved in the local intensification procedure. In the first stage, the top ζ tasks with the largest $E(loss_j)(j = 1, 2, \dots, m)$ in the current solution π_s are selected. Afterward, each selected task is performed $LS_i(i = 1, 2, 3)$ to obtain a set of neighboring

solutions, denoted as set_i , if there exists any solution $\pi_{s'}$ in set_i that dominates π_s , then π_s is replaced by $\pi_{s'}$. This process continues until each task is examined, and all new generated solutions are stored into a set LWS . The second stage is similar to the first one; the difference is that the top ζ tasks of $E(c_j)(j = 1, 2, \dots, m)$ are extracted to implement the local search in this stage. Finally, the non-dominated set of LWS is returned. The pseudo-code of multi-objective local search strategy is detailed in Pseudocode 4.

Algorithm 4 Multi-objective local search strategy

Input: A solution π_s, ζ
Output: the solutions $LPOP$ generated by local search

- 1: Select the top ζ tasks according to $E(loss_j), j = 1, 2, \dots, m$ in π_s , denoted as $[\sigma_1^l, \sigma_2^l, \dots, \sigma_\zeta^l]$.
- 2: $LWS \leftarrow []$.
- 3: **for** each task σ_j^l **do**
- 4: **for** $i \leftarrow 1$ to 3 **do**
- 5: $set_i \leftarrow LS_i(\pi_s, \sigma_j^l)$ // Perform i -th local search method for π_s
- 6: **if** $\exists \pi_{s'} \in set_i \wedge \pi_{s'} < \pi_s$ **then**
- 7: $\pi_s \leftarrow \pi_{s'}$.
- 8: **end if**
- 9: $LWS \leftarrow LWS \cup set_i$.
- 10: **end for**
- 11: **end for**
- 12: Select the top ζ tasks according to $E(c_j), j = 1, 2, \dots, m$ in π_s , denoted as $[\sigma_1^c, \sigma_2^c, \dots, \sigma_\zeta^c]$.
- 13: **for** each task σ_j^c **do**
- 14: **for** $i \leftarrow 1$ to 3 **do**
- 15: $set_i \leftarrow LS_i(\pi_s, \sigma_j^c)$ // Perform i -th local search method for π_s
- 16: **if** $\exists \pi_{s'} \in set_i \wedge \pi_{s'} < \pi_s$ **then**
- 17: $\pi_s \leftarrow \pi_{s'}$.
- 18: **end if**
- 19: $LWS \leftarrow LWS \cup set_i$.
- 20: **end for**
- 21: **end for**
- 22: $LPOP \leftarrow$ Pareto-optimal set of LWS .

Overview of IMEDA

As mentioned above, the complete procedure of IMEDA mainly involves four components, i.e., initialization, construction of probabilistic models, sampling methods, and multi-objective local search. First, a two-stage initialization method is proposed to generate initial solutions. Then, three probabilistic models considering the position relationships between rescue teams and tasks are established according to the distribution of superior solutions. Next, a sampling approach based on the referenced sequence is developed to produce offspring. Finally, a multi-objective local search strategy is applied to several specific superior solutions. The procedure continues until stop criterion is satisfied. Pseudocode 5 outlines the framework of IMEDA and Fig. 3 shows the relationship between main solution sets.

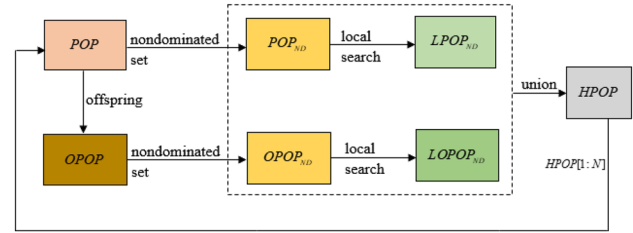


Fig. 3 The relationship of main solution sets

Algorithm 5 The framework of IMEDA

Input: the parameters in Table 1, the population size N
Output: a solution set AS

- 1: Initialize the population POP . // See Pseudocode 2
- 2: $POP_{ND} \leftarrow$ the non-dominated set of POP .
- 3: Compute the probabilistic matrix PF, PS and PA based on POP_{ND} .
- 4: **while** the stop criterion is not satisfied **do**
- 5: Produce an offspring for each solution in POP by sampling from the probabilistic matrices, and denote the offspring set as $OPOP$. // See Pseudocode 3
- 6: $OPOP_{ND} \leftarrow$ the non-dominated set of $OPOP$.
- 7: Perform multi-objective local search for each solution in POP_{ND} , store all solution sets generated by local search into $LPOP_{ND}$. // See Pseudocode 4
- 8: Perform multi-objective local search for each solution in $OPOP_{ND}$, store all solution sets generated by local search into $LOPOP_{ND}$. // See Pseudocode 4
- 9: Let $HPOP \leftarrow POP_{ND} \cup OPOP_{ND} \cup LPOP_{ND} \cup LOPOP_{ND}$.
- 10: Implement fast non-dominated sorting on $HPOP$.
- 11: $AS \leftarrow$ the non-dominated set of $HPOP$.
- 12: $POP \leftarrow HPOP[1:N]$.
- 13: $POP_{ND} \leftarrow$ the non-dominated set of POP .
- 14: Compute the probabilistic matrix PF, PS and PA based on the distribution of POP_{ND} .
- 15: Update PF, PS and PA .
- 16: **end while**

Computational experiment

Test instances

Based on the public statistical data from disaster operations of the 2008 Wenchuan earthquake (08W), the 2010 Yushu earthquake (10Y), the 2010 Zhouqu mudslides (10Z), the 2011 Yingjiang earthquake (11Y), the 2012 Ninglang earthquake (12N), the 2013 Ya'an earthquake (13Y), the 2016 Wuhan flood (16W), the 2017 Jiuzhaigou earthquake (17J), and the 2021 Zhengzhou rainstorm (21Z), respectively, we construct nine instances for the considered problem. According to the number of rescue teams, 08W, 13Y, 16W, and 21Z are considered as high-dimensional instances, while the remaining are regarded as low-dimensional instances. Table 3 shows the range of values for generated parameters which

Table 3 Parameters and distribution

Parameters	Distribution
$t_{\lambda(\lambda+1)}^{swi}$: the time it takes for state λ to completely transition to state $\lambda + 1$	[12, 24] hours
t_{0j}^{tra} : the travel time from workstation to task j	$dis(loc_{workstation}, loc_j)/v + \varepsilon$ $v \in [40, 65]$ (km/h), $\varepsilon \in N(0, 25)$ minutes
$t_{jj'}^{tra}$: the travel time from task j to task j'	$dis(loc_j, loc_{j'})/v + \varepsilon$ $v \in [40, 65]$ (km/h), $\varepsilon \in N(0, 25)$ minutes
δ_i : fatigue factor	0.05
α_j : state coefficient of rescue task j	0.05

Table 4 The MRT for each instance

Name	n	m	MRT (m)
08W	58	626	30
10Y	15	67	15
10Z	16	39	10
11Y	12	55	15
12N	9	32	10
13Y	21	102	20
16W	26	113	20
17J	11	43	10
21Z	47	293	25

are in line with real rescue situation, where $loc_{workstation}$ and loc_j represent the location of workstation and rescue task j , respectively, $dis(loc_j, loc_{j'})$ represents the distance between them. Due to urgency requirement, a maximum response time (MRT) shown in Table 4 is set for each instance, which indicates the upper limit of time for algorithm to solve this problem.

All experiments are performed on a PC with Inter(R) Core(TM) i7-6500U CPU and 8 G RAM and all algorithms are implemented with the same language. In the following, we calibrate parameter settings, and analyze the critical components of IMEDA, and then provide a comparative test with the existing algorithms in detail.

Performance measure indicators

Convergence and distribution are two aspects for validating the performance of a multi-objective algorithm. In our experiment, two metrics are employed to measure the convergence and distribution of founded non-dominated solutions, which are illustrated as follows:

(1) Hypervolume (I_H) [47]: The indicator I_H measures the hypervolume enclosed by the solution space dominated by a set A of non-dominated solutions and the referenced

Table 5 Combinations of parameter values

Parameter	Factor level		
	Level1	Level2	Level3
N	10	20	30
w_{pf}	0.5	0.6	0.7
w_{ps}	0.5	0.6	0.7
w_{pa}	0.5	0.6	0.7
ρ	0.01	0.015	0.02
μ	0.15	0.2	0.25
ϑ	0.03	0.04	0.05
γ	0.1	0.2	0.3

point $z^* = (z_1, z_2)$. Each objective value is normalized before calculating the I_H , as shown in (21)

$$f'_i(x) = \frac{f_i(x) - \min f_i}{\max f_i - \min f_i}, \quad (21)$$

where $f_i(x)$ is the i -th objective value of the solution x , and $f'_i(x)$ denotes the normalized $f_i(x)$, and $\min f_i$ and $\max f_i$ represent the minimum and maximum values of i -th objective from all Pareto fronts of all algorithms [48]. Then, a two-objective $I_H(A)$ can be computed as (22)

$$I_H(A) = VOL \left(\bigcup_{x \in A} |f'_1(x), z_1| \times |f'_2(x), z_2| \right), \quad (22)$$

where $VOL(\cdot)$ is the Lebesgue measurement. As shown in [49, 50], the reference point z^* is assigned as (1, 2, 1.2), which guarantees that the contributions of extreme points (0, 1) and (1, 0) for the hypervolume are not neglected. For comparing two fronts, a larger $I_H(A)$ indicates a better one.

(2) Unary Epsilon (I_ε^1) [51]: The indicator I_ε^1 calculates the distance between a given Pareto front A and referenced Pareto front R . Since R is unknown, we construct a referenced set by collecting all non-dominated solutions obtained from all considered algorithms. Each objective value is normalized into [1, 2], as shown in (23)

$$f'_i(x) = \frac{f_i(x) - \min f_i}{\max f_i - \min f_i} + 1, \quad (23)$$

where $f_i(x)$, $f'_i(x)$ and $\min f_i$, $\max f_i$ have the same meanings as (21). Then, a two-objective $I_\varepsilon^1(A, R)$ can be computed as (24)

$$I_\varepsilon^1(A, R) = \max_{y \in R} \min_{x \in A} \max_{1 \leq i \leq 2} \frac{f'_i(x)}{f'_i(y)}. \quad (24)$$

The range of $I_\varepsilon^1(A, R)$ is [1, 2], and a value close to 1 denotes a better approximation to the referenced Pareto front.

Table 6 Orthogonal array and ARV

NO.	Parameters								Δ_{avg}					ARV
	N	w_{pf}	w_{ps}	w_{pa}	ρ	μ	ϑ	γ	08W	21Z	16W	10Z	12N	
1	2	2	3	1	2	3	3	3	0.943	1.363	1.361	1.342	1.332	1.268
2	3	1	3	2	3	3	1	1	1.023	1.382	1.376	1.320	1.332	1.286
3	3	3	2	1	3	2	2	2	1.036	1.392	1.375	1.325	1.255	1.277
4	3	2	1	3	3	1	3	3	1.009	1.386	1.359	1.326	1.271	1.270
5	2	1	2	1	1	1	3	2	1.030	1.356	1.347	1.364	1.028	1.225
6	3	2	1	2	1	2	1	2	1.029	1.381	1.379	1.350	1.028	1.234
7	1	3	3	3	1	3	2	2	0.848	1.284	1.334	1.335	1.386	1.237
8	2	3	1	3	1	3	1	3	0.949	1.350	1.345	1.347	1.332	1.265
9	3	3	2	2	2	1	1	3	0.952	1.380	1.352	1.358	1.028	1.214
10	3	2	1	1	2	3	2	1	1.027	1.380	1.364	1.325	1.332	1.285
11	1	2	2	1	2	3	1	2	0.851	1.308	1.305	1.299	1.386	1.230
12	2	2	3	3	3	1	1	2	0.996	1.342	1.346	1.353	1.333	1.274
13	1	1	1	3	2	2	2	3	0.903	1.307	1.281	1.314	1.299	1.221
14	1	2	2	3	3	1	2	1	0.911	1.312	1.298	1.305	1.280	1.221
15	2	1	2	2	3	3	2	3	0.949	1.347	1.351	1.347	1.303	1.259
16	3	3	2	3	1	3	3	1	1.005	1.403	1.372	1.326	1.332	1.288
17	2	1	2	3	2	2	1	1	1.010	1.358	1.332	1.346	1.028	1.215
18	2	2	3	2	1	2	2	1	1.023	1.356	1.364	1.347	1.028	1.224
19	1	1	1	1	1	1	1	1	0.922	1.292	1.341	1.338	1.299	1.238
20	2	3	1	1	3	2	3	1	1.011	1.360	1.349	1.317	1.292	1.266
21	1	3	3	1	3	2	1	3	0.857	1.283	1.315	1.301	1.293	1.210
22	2	3	1	2	2	1	2	2	1.013	1.346	1.346	1.353	1.028	1.217
23	3	1	3	1	1	1	2	3	0.964	1.361	1.376	1.355	1.028	1.217
24	1	2	2	2	1	2	3	3	0.849	1.269	1.332	1.328	1.299	1.215
25	3	1	3	3	2	2	3	2	1.037	1.371	1.382	1.337	1.028	1.231
26	1	1	1	2	3	3	3	2	0.814	1.287	1.320	1.338	1.221	1.196
27	1	3	3	2	2	1	3	1	0.900	1.323	1.310	1.338	1.299	1.234

Table 7 Average response value and rank of each factor

Level	Parameters							
	N	w_{pf}	w_{ps}	w_{pa}	ρ	μ	ϑ	γ
1	1.223	1.232	1.244	1.246	1.238	1.235	1.241	1.251
2	1.246	1.247	1.238	1.231	1.235	1.232	1.240	1.236
3	1.256	1.245	1.242	1.247	1.251	1.257	1.244	1.238
Extreme difference	0.033	0.015	0.005	0.016	0.016	0.025	0.004	0.015
Rank	1	6	7	4	3	2	8	5

Calibration of IMEDA

As mentioned above, our proposed algorithm includes eight parameters to be calibrated: N , w_{pf} , w_{ps} , w_{pa} , ρ ($\rho = d/m$ denotes the proportion of selected tasks in the referenced sequence-based sampling method), μ ($\mu = \zeta/m$ denotes the proportion of selected tasks to undergo local search), and ϑ ($\vartheta = k/m$ denotes the proportion of tasks regarded as the neighborhood), γ . A Taguchi calibration approach is performed on five instances to determine appropriate parameter

settings. For a set of parameter combination, each instance is performed five replications and average I_H is recorded. Table 5 displays factor levels for each parameter. The orthogonal array $L_{27}(3^8)$ and total average Δ_{avg} are illustrated in Table 6. The total average Δ_{avg} is used to calculate average response variable (ARV). Based on extreme difference, the relative importance of parameters is listed in Table 7. Figure 4 describes the I_H value of a factor at different levels according to Table 7, which helps to determine a level for each factor.

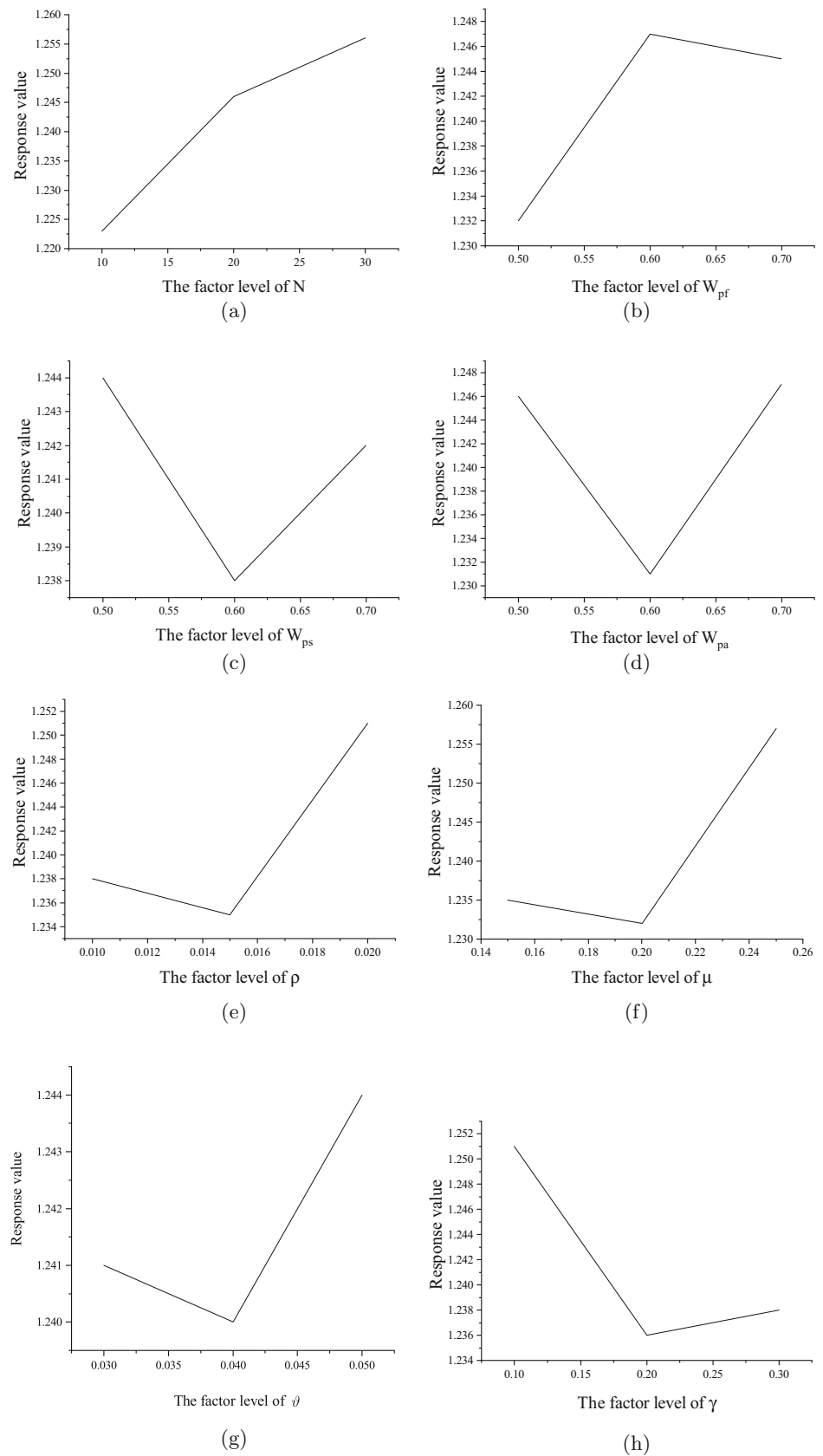
Fig. 4 Factor level trend of parameters

Table 7 demonstrates that the population size N has the most significant influence on performance of IMEDA. A bigger N makes more contribution in obtaining high-quality solutions. μ is the second parameter, which controls the number of tasks performing local intensification. We can see that the exploitation ability of IMEDA is enhanced with the increase of μ . Regarding the parameter ρ , the trend curve illustrates that bigger value leads to better results, because a bigger ρ is of benefit to the diversity of solutions. w_{pa} places the fourth position, which shows that the adjacent probability of two tasks is more derived from the superior solutions than the geographical relationship. According to the above analysis, the level of rest parameters can be seen from Fig. 4. When $N = 30$, $w_{pf} = 0.6$, $w_{ps} = 0.5$, $w_{pa} = 0.7$, $\rho = 0.02$, $\mu = 0.25$, $\vartheta = 0.05$, and $\gamma = 0.1$, IMEDA can achieve the best performance.

Performance analysis of each part of IMEDA

As mentioned in Section [Proposed algorithm](#), IMEDA involves three critical components: initialization, sampling method based on referenced sequence, and multi-objective local search strategy. To validate the effectiveness of these components, we test and analyze three variants of IMEDA, including IMEDA_I, IMEDA_S, and IMEDA_L. Each variant modifies only one component of IMEDA. IMEDA_I denotes a randomly generated initial population for the IMEDA. In IMEDA_S, an offspring individual is generated by sampling from probabilistic models. Specifically, the first task of each team should be determined by sampling from probability matrix PF . Then, we successively assign each unscheduled task to the corresponding team through sampling from probability matrix PS of two tasks processed by the same team, and decide its position in the team via sampling from probability matrix PA of two adjacent tasks. IMEDA_L removes the local search methods applied to the POP_{ND} and $OPOP_{ND}$. All variants adopt the MRT as stop criterion. Each variant is performed 30 replications on each instance. The median, worst, best, and standard deviation (std.) of obtained values in hypervolume and unary epsilon indices are recorded in Table 8, where the better values of best and median metrics are shown in boldface. Furthermore, non-parametric Wilcoxon rank sum test is conducted to make a comparison between IMEDA and three variants on two indicators for each instance. Table 9 displays the results, where italic p value shows that performance of IMEDA and its variant has statistically significantly difference at a confidence level of 95%.

From Table 8, it can be observed that IMEDA_L has the worst result, which means that the multi-objective local search strategy influences the performance of IMEDA to a large extent. Efficient local search method improves the quality of superior non-dominated solutions, and exploita-

tion ability of IMEDA is reduced without the effective local search mechanism. IMEDA_I yields worse results except for 10Y, 10Z, and 11Y, which emphasizes the critical role of initialization for algorithm solving the considered problem, especially for high-dimensional instances. Our proposed initialization mechanism makes IMEDA start searching from a promising area, which significantly benefits obtaining high-quality solutions. The variant IMEDA_S is also inferior to IMEDA, which implies that the offspring population could make full use of the inherited information from their parents through the referenced sequence-based sampling method, that ensures the elite solutions are well evolved toward the Pareto front. To validate the conclusions mentioned above, we conduct the box plots of IMEDA and each variant on two indicators for all instances, as illustrated in Figs. 5 and 6 (Here I, I1, I2, and I3 denotes IMEDA, IMEDA_I, IMEDA_S, and IMEDA_L, respectively). It can be observed that the difference between IMEDA and its variants in high-dimensional cases is more significant than that in low-dimensional cases, which reveals that IMEDA achieves more non-dominated solutions in reference Pareto front for the high-dimensional instances. Additionally, the result of non-parametric Wilcoxon rank sum test in Table 9 highlights the performance of IMEDA outperforms its variants from a statistical perspective except IMEDA_I applying to 10Y and IMEDA_S applying to 16 W, which further confirms the effectiveness of all employed components.

Comparison of IMEDA with other algorithms

To evaluate the effectiveness of IMEDA, we compare it with nine multi-objective evolutionary algorithms. As stated in Section [The rescue task emergency scheduling problem considering stochastic deterioration of the injured](#), two algorithms are presented to solve the rescue task allocation and scheduling problem, that is, EMOBBO of Zheng et al. [2] and MOTS of Zheng et al. [19]. Hence we adopt these two algorithms to make a comparison with IMEDA. Since PSO and HSP of Nayeri et al. [9] are presented for the problem with a single objective, we integrate the multi-objective concept into the algorithm to make it suitable for the considered problem, that is, applying fast non-dominated sorting method to sort the solutions and store non-dominated solutions. Moreover, MOEALS of Shao et al. [42] and MGVNS of Shao et al. [52] proposed in these 2 years for distributed job scheduling problem are compared with IMEDA. For an adequate comparison, three general multi-objective optimization algorithms, i.e., NSGAII of Deb et al. [53], SPEAII of Zitzler et al. [54], and PESAI of Corne et al. [55], are participated in solving this problem. Note that these optimization algorithms are not initially designed for the rescue task emergency scheduling problem, we maintain the framework of algorithm and employ crossover and mutation operators proposed in liter-

Table 8 Statistical results of hypervolume and unary epsilon for the variants of IMEDA

	Metric	IMEDA_I		IMEDA_S		IMEDA_L		IMEDA	
		I_H	I_ϵ^1	I_H	I_ϵ^1	I_H	I_ϵ^1	I_H	I_ϵ^1
08W	Median	0.555	1.612	1.265	1.027	1.119	1.107	1.270	1.025
	Worst	0.263	1.904	1.264	1.028	1.107	1.108	1.268	1.027
	Best	0.608	1.574	1.266	1.026	1.124	1.101	1.278	1.021
	Std	0.125	0.112	0.001	0.001	0.005	0.002	0.003	0.002
10Y	Median	1.216	1.117	1.181	1.171	0.717	1.584	1.244	1.111
	Worst	1.122	1.207	1.181	1.199	0.622	1.669	1.181	1.170
	Best	1.322	1.044	1.182	1.169	0.853	1.469	1.339	1.044
	Std	0.062	0.043	0.000	0.001	0.057	0.049	0.046	0.037
10Z	Median	1.412	1.019	1.421	1.017	1.372	1.056	1.430	1.010
	Worst	1.345	1.047	1.406	1.028	1.369	1.059	1.419	1.017
	Best	1.431	1.007	1.431	1.006	1.382	1.048	1.438	1.004
	Std	0.016	0.007	0.009	0.006	0.004	0.003	0.006	0.004
11Y	Median	1.188	1.109	1.157	1.130	1.033	1.222	1.223	1.089
	Worst	1.106	1.185	1.155	1.131	1.008	1.251	1.178	1.133
	Best	1.280	1.048	1.160	1.129	1.076	1.179	1.303	1.046
	Std	0.032	0.027	0.001	0.001	0.016	0.016	0.028	0.020
12N	Median	1.192	1.170	1.272	1.116	0.784	1.525	1.295	1.104
	Worst	1.060	1.296	1.099	1.234	0.667	1.641	1.222	1.146
	Best	1.271	1.108	1.384	1.034	0.846	1.434	1.374	1.047
	Std	0.041	0.038	0.067	0.047	0.033	0.043	0.039	0.026
13Y	Median	1.204	1.138	1.234	1.147	1.035	1.332	1.272	1.105
	Worst	1.139	1.205	1.231	1.148	0.982	1.377	1.239	1.135
	Best	1.308	1.065	1.237	1.146	1.078	1.292	1.380	1.010
	Std	0.042	0.035	0.002	0.001	0.021	0.019	0.029	0.027
16W	Median	1.224	1.138	1.322	1.080	1.096	1.260	1.320	1.078
	Worst	0.991	1.302	1.322	1.149	1.051	1.301	1.262	1.078
	Best	1.349	1.075	1.323	1.025	1.150	1.203	1.401	1.072
	Std	0.070	0.048	0.000	0.033	0.023	0.023	0.038	0.003
17J	Median	1.211	1.153	1.237	1.127	0.571	1.713	1.259	1.119
	Worst	1.125	1.234	1.184	1.128	0.530	1.756	1.235	1.128
	Best	1.339	1.060	1.238	1.126	0.669	1.597	1.386	1.021
	Std	0.055	0.045	0.001	0.001	0.041	0.046	0.047	0.040
21Z	Median	0.878	1.443	1.203	1.196	1.171	1.220	1.206	1.195
	Worst	0.737	1.557	1.202	1.197	1.168	1.223	1.196	1.203
	Best	1.099	1.214	1.204	1.195	1.176	1.216	1.216	1.186
	Std	0.103	0.102	0.001	0.001	0.002	0.002	0.004	0.003

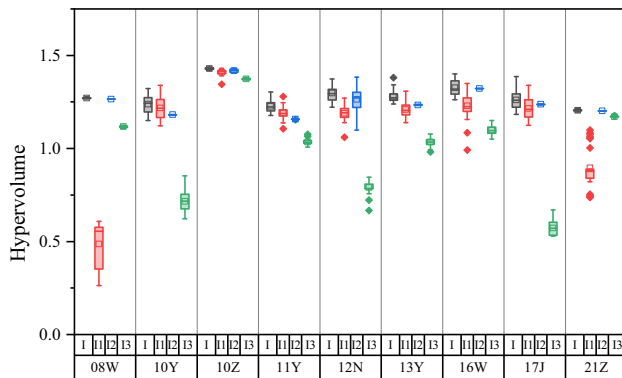
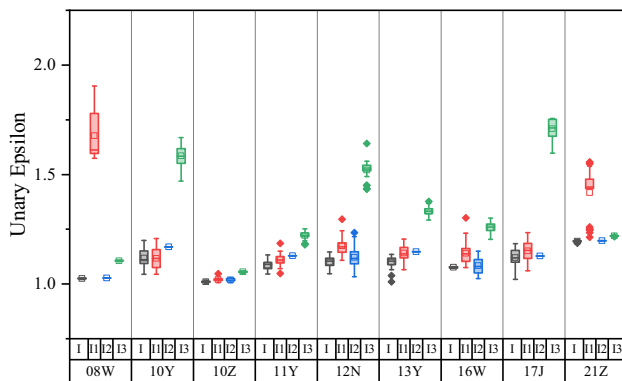
ature [56]. Suppose s_1 and s_2 are two feasible solutions. In crossover operation, a task is randomly selected from each team of s_2 for dividing s_1 , tasks that are on the right side of each dividing point of s_2 are removed from s_1 , then the rest partial sequence of s_1 is inherited into the child s' , and finally, the right-side tasks of s_2 are inserted into the corresponding team of s' . In mutation operation, a task pair is randomly selected to swap positions, and the number of task pair is $m/4$. It is noted that we re-implement the compared algorithms strictly based on the explanations of the origi-

nal literature, and all parameters are calibrated by Taguchi method. To make a fair comparison, all algorithms have the same experimental conditions and take MRT as the stopping criterion.

The computational results on hypervolume and unary epsilon indices are summarized in Table 10, where better results in best and median metrics are marked in boldface. From Table 10, we can see that the proposed IMEDA is superior to its counterparts on hypervolume and unary epsilon indicators. The I_H and I_ϵ^1 value of IMEDA are much close to

Table 9 p Values of the non-parametric Wilcoxon rank sum test between IMEDA and variants

Instance	Indicator	IMEDA_I	IMEDA_S	IMEDA_L
08W	I_H	0.000	0.000	0.000
	I_ε^1	0.000	0.000	0.000
10Y	I_H	0.114	0.000	0.000
	I_ε^1	0.965	0.000	0.000
10Z	I_H	0.000	0.000	0.000
	I_ε^1	0.000	0.000	0.000
11Y	I_H	0.000	0.000	0.000
	I_ε^1	0.000	0.000	0.000
12N	I_H	0.000	0.027	0.000
	I_ε^1	0.000	0.071	0.000
13Y	I_H	0.000	0.000	0.000
	I_ε^1	0.000	0.000	0.000
16W	I_H	0.000	0.650	0.000
	I_ε^1	0.000	0.188	0.000
17J	I_H	0.002	0.003	0.000
	I_ε^1	0.014	0.343	0.000
21Z	I_H	0.000	0.001	0.000
	I_ε^1	0.000	0.000	0.000

**Fig. 5** Box plot of hypervolume for all variants of IMEDA**Fig. 6** Box plot of unary epsilon for all variants of IMEDA

the best values for most cases. The result of MOTS is close to IMEDA for the low-dimensional instances. However, the performance differences of two algorithms are enlarged with the increase of instance size, which verifies the efficiency and stability of IMEDA in solving the high-dimensional instances. Meanwhile, MOEALS and MGVNS exhibit worse value of indices for the considered problem. The reason may be that these algorithms take quite a long time to obtain efficient initial solutions, resulting in inadequate evolution and local search for these solutions in limited response time. Additionally, HSP and PSO are not doing well in this problem, because these two algorithms are initially designed for the single-objective rescue unit allocation and scheduling problem, and the adaptability improvement of two algorithms is not adequate for designing the overall framework for solving this multi-objective scheduling problem. NSGAI, SPEAI, PESAI, and EMOBBO are better than HSP and PSO, which demonstrates the effectiveness of implementing Pareto-based local search in searching process.

Figures 7, 8 show the box plots of the union I_H and I_ε^1 value which includes 270 results combining 9 instances and 30 replicates per instance. As seen in these figures, there is a slight overlapping interval between IMEDA and MOTS, and no overlapping interval between IMEDA and other algorithms. The reason for the slight overlapping interval between IMEDA and MOTS is that the performance of two algorithms are close to each other for low-dimensional instances.

The aforementioned analysis has demonstrated the effectiveness of IMEDA for the considered problem. Nevertheless, the computational results have stochastic characteristics to a certain extent. Therefore, we perform a non-parametric post hoc Holm's procedure on I_H and I_ε^1 value, which is suitable for testing the computational results of a group of evolutionary algorithms. IMEDA is regarded as the control algorithm, then κ hypotheses are generated, denoted as $H_1, H_2, \dots, H_\kappa$. H_i ($i = 1, 2, \dots, \kappa$) represents that the i -th compared algorithm is equal to the control algorithm. For each $i = 1, 2, \dots, \kappa$, p_i denotes the corresponding p -value of testing H_i . Afterward, κ p -values are sorted in ascending order, $p_{1'} \leq p_{2'} \leq \dots \leq p_{\kappa'}$. Suppose that i' is the smallest integer satisfying $p_{i'} \geq \alpha/(\kappa - i' + 1)$, then the Holm's procedure rejects $H_{1'}, H_{2'}, \dots, H_{i'-1}$ and accept the remaining hypotheses at a $(1 - \alpha)$ confidence level. The results of Holm's procedure are illustrated in Table 11, where $\alpha = 0.5$. It is observed that each hypothesis is rejected due to $p_i \leq \alpha/(\kappa - i + 1)$ on two indicators, indicating that IMEDA outperforms those nine algorithms from a statistical perspective.

Figure 9 illustrates the Pareto front of each instance obtained by compared algorithms. The solutions yielded by PSO and HSP are not displayed due to their inferior performance. From these figures, we can see that MOEALS and MGVNS exhibits good convergence while bad diver-

Table 10 Statistical results of hypervolume and unary epsilon for compared algorithms

Metric	NSGAI		SPEAI		PESAI		MOEALS		PSO		HSP		MOTS		EMOBBO		MGVNS		IMEDA	
	I_H	I_ϵ^1	I_H	I_ϵ^1	I_H	I_ϵ^1	I_H	I_ϵ^1	I_H	I_ϵ^1	I_H	I_ϵ^1	I_H	I_ϵ^1	I_H	I_ϵ^1	I_H	I_ϵ^1	I_H	I_ϵ^1
08W	Median	1.269	1.134	1.260	1.142	1.339	1.081	0.433	1.835	1.092	1.263	1.022	1.293	1.421	1.329	1.071	0.624	1.679	1.427	1.013
	Worst	1.256	1.146	1.245	1.153	1.327	1.089	0.433	1.835	1.029	1.296	0.948	1.346	1.410	1.324	1.076	0.617	1.684	1.419	1.019
	Best	1.300	1.119	1.284	1.119	1.347	1.076	0.443	1.821	1.297	1.118	1.109	1.244	1.437	1.006	1.061	0.682	1.630	1.434	1.006
	Std	0.009	0.007	0.008	0.007	0.005	0.004	0.004	0.006	0.075	0.050	0.036	0.023	0.006	0.005	0.003	0.027	0.023	0.004	0.003
10Y	Median	1.051	1.273	1.033	1.281	1.165	1.188	0.317	1.918	0.833	1.448	0.571	1.542	1.392	1.202	1.119	0.756	1.548	1.420	1.017
	Worst	1.035	1.285	1.008	1.301	1.134	1.213	0.317	1.903	0.545	1.587	0.480	1.616	1.380	1.187	1.125	0.740	1.566	1.404	1.029
	Best	1.077	1.246	1.078	1.251	1.198	1.166	0.321	1.919	1.211	1.181	0.663	1.455	1.408	1.230	1.114	0.840	1.478	1.434	1.006
	Std	0.010	0.010	0.017	0.015	0.016	0.012	0.002	0.007	0.192	0.109	0.049	0.036	0.008	0.011	0.003	0.040	0.034	0.008	0.005
10Z	Median	1.241	1.144	1.141	1.197	1.336	1.077	0.615	1.666	0.992	1.274	0.938	1.316	1.402	1.258	1.089	0.724	1.591	1.419	1.016
	Worst	1.224	1.152	1.111	1.220	1.284	1.114	0.609	1.671	0.957	1.302	0.842	1.349	1.382	1.238	1.098	0.704	1.609	1.405	1.026
	Best	1.276	1.117	1.200	1.157	1.360	1.053	0.624	1.666	1.060	1.204	1.020	1.266	1.414	1.021	1.272	1.075	1.568	1.426	1.011
	Std	0.010	0.008	0.018	0.013	0.019	0.014	0.006	0.003	0.026	0.019	0.033	0.019	0.010	0.007	0.008	0.005	0.019	0.016	0.003
11Y	Median	1.088	1.193	1.074	1.196	1.229	1.111	0.468	1.584	0.928	1.368	0.558	1.484	1.374	1.111	1.221	1.045	1.179	1.391	1.022
	Worst	1.071	1.207	1.041	1.212	1.196	1.137	0.467	1.585	0.651	1.473	0.371	1.615	1.356	1.092	1.240	1.034	1.193	1.375	1.038
	Best	1.141	1.167	1.105	1.169	1.260	1.091	0.537	1.436	1.119	1.255	0.896	1.305	1.414	1.025	1.176	1.172	1.067	1.145	1.408
	Std	0.016	0.010	0.014	0.011	0.015	0.012	0.034	0.074	0.160	0.064	0.094	0.054	0.013	0.007	0.014	0.012	0.019	0.008	0.005
12N	Median	1.291	1.118	1.282	1.124	1.363	1.061	0.360	1.894	1.075	1.282	0.992	1.336	1.376	1.050	1.366	1.053	1.540	1.426	1.013
	Worst	1.274	1.129	1.267	1.139	1.346	1.072	0.361	1.904	1.005	1.341	0.899	1.409	1.351	1.070	1.357	1.057	1.550	1.418	1.024
	Best	1.310	1.106	1.306	1.101	1.382	1.051	0.362	1.894	1.325	1.096	1.089	1.252	1.391	1.042	1.373	1.049	1.490	1.435	1.007
	Std	0.009	0.006	0.010	0.008	0.010	0.006	0.001	0.004	0.100	0.076	0.045	0.040	0.008	0.006	0.004	0.029	0.024	0.005	0.004
13Y	Median	1.125	1.219	1.104	1.228	1.197	1.169	0.808	1.521	0.819	1.403	0.667	1.460	1.377	1.042	1.213	1.114	1.517	1.419	1.015
	Worst	1.111	1.231	1.086	1.240	1.172	1.186	0.787	1.539	0.712	1.449	0.609	1.539	1.367	1.050	1.198	1.127	1.523	1.414	1.022
	Best	1.137	1.210	1.130	1.214	1.229	1.146	0.808	1.521	1.082	1.274	0.784	1.413	1.385	1.035	1.236	1.092	1.497	1.436	1.003
	Std	0.007	0.005	0.011	0.007	0.012	0.009	0.009	0.008	0.093	0.039	0.046	0.030	0.005	0.003	0.008	0.007	0.011	0.010	0.004
16W	Median	1.011	1.281	0.975	1.305	1.110	1.221	0.685	1.572	0.584	1.528	0.497	1.576	1.372	1.046	1.173	1.138	1.395	1.424	1.015
	Worst	0.999	1.288	0.945	1.319	1.075	1.237	0.685	1.573	0.527	1.575	0.426	1.655	1.360	1.058	1.153	1.154	1.428	1.415	1.027
	Best	1.036	1.253	1.006	1.275	1.148	1.192	0.688	1.572	0.821	1.431	0.577	1.530	1.383	1.036	1.199	1.128	1.339	1.435	1.004
	Std	0.008	0.007	0.014	0.012	0.019	0.012	0.002	0.000	0.081	0.038	0.037	0.031	0.006	0.005	0.013	0.007	0.041	0.037	0.005
17J	Median	1.269	1.134	1.260	1.142	1.339	1.081	0.433	1.835	1.092	1.263	1.022	1.293	1.421	1.016	1.329	1.071	1.679	1.427	1.013
	Worst	1.256	1.146	1.245	1.153	1.327	1.089	0.433	1.835	1.029	1.296	0.948	1.346	1.410	1.023	1.324	1.076	1.684	1.419	1.019
	Best	1.300	1.119	1.284	1.119	1.347	1.076	0.443	1.821	1.297	1.118	1.109	1.244	1.437	1.006	1.061	0.682	1.630	1.434	1.006
	Std	0.009	0.007	0.008	0.007	0.005	0.004	0.004	0.006	0.075	0.050	0.036	0.023	0.006	0.004	0.005	0.027	0.023	0.004	0.003
21Z	Median	0.912	1.295	0.855	1.313	1.025	1.243	1.012	1.205	0.501	1.574	0.407	1.646	1.214	1.155	1.114	1.215	1.093	1.437	1.003
	Worst	0.899	1.303	0.832	1.327	0.994	1.254	0.991	1.205	0.429	1.625	0.309	1.713	1.188	1.173	1.082	1.238	1.084	1.203	1.434
	Best	0.935	1.283	0.893	1.291	1.059	1.222	1.077	1.205	0.651	1.505	0.571	1.505	1.249	1.129	1.142	1.189	1.106	1.440	1.002
	Std	0.009	0.005	0.018	0.008	0.014	0.009	0.039	0.000	0.057	0.031	0.058	0.048	0.012	0.010	0.015	0.012	0.009	0.007	0.001

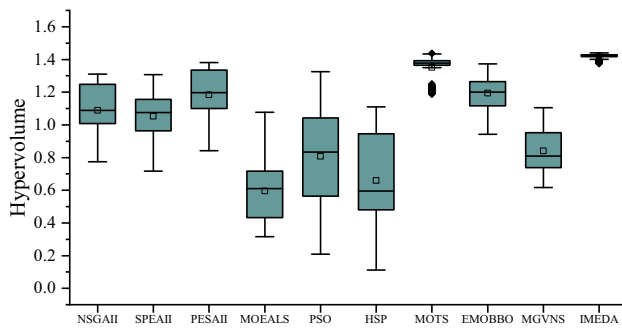


Fig. 7 Box plot of hypervolume for compared algorithms

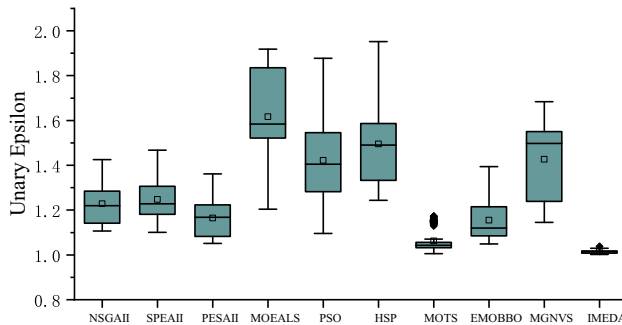


Fig. 8 Box plot of unary epsilon for compared algorithms

sity, which can explain their poor results on hypervolume and unary epsilon indices. However, these two algorithms still obtain some high-quality solutions for high-dimensional instances. Because the detailed initialization method and local search strategy ensure that solutions evolve toward the Pareto front. Furthermore, it can be observed that for low-dimensional instances, the Pareto fronts achieved by IMEDA and MOTS approach to the bottom-left corner of coordinate axis, which exhibits their good searching ability compared to other algorithms. With the increase of instance scale, the performance differences between compared algorithms are obviously enlarged. Especially, the non-dominated solutions obtained by IMEDA can dominate those of other consid-

ered algorithms. In other words, IMEDA achieves reference Pareto front for high-dimensional instances. IMEDA shows stable performance on various scale instances, demonstrating that IMEDA is high performing for solving the considered problem.

Conclusion

In this paper, we investigate a multi-objective rescue task allocation and scheduling problem considering the stochastic deterioration of injured with the objective of minimizing both average mathematical expectation of all tasks' completion time and casualty loss. This problem is important in planning the rescue work, because there are a large number of tasks in disaster areas and the response time for decision-makers is quite limited. The study will help improve the efficiency of rescue operations in disaster management.

Our proposed algorithm is innovative in some ways. First, an initialization method is designed to generate high-quality individuals in promising areas, which takes into account the geographical proximity between tasks and casualty loss simultaneously. Second, three probabilistic models considering the proximity of rescue tasks are established to gather advanced statistical information from superior individuals, which makes the algorithm have a strong global exploration ability. Third, a multi-objective local search strategy is performed on elite individuals to enhance the local exploitation.

Several issues are worth more consideration in modeling. First, rescue work has various uncertainties, e.g., rescue tasks' processing time, transition time between two adjacent injured states, which are difficult to estimate. In addition, rescuers may be injured in their work, which will lead to an extension of task processing time if rescue forces are not replenished in time. How to represent these uncertainties to match actual rescue situation should be further studied. Second, rescuers have been in a high-risk working environment

Table 11 Results of Holm's procedure

i	$\alpha/(\kappa - i + 1)$	Hypervolume			Unary epsilon		
		H_i	p -value	R/A	H_i	p -value	R/A
1	0.0056	IMEDA=MOEALS	0.0000	R^2	IMEDA=MOEALS	0.0000	R^2
2	0.0063	IMEDA=MGVNS	0.0000	R^2	IMEDA=MGVNS	0.0000	R^2
3	0.0071	IMEDA=HSP	0.0000	R^2	IMEDA=HSP	0.0000	R^2
4	0.0083	IMEDA= PSO	0.0000	R^2	IMEDA= PSO	0.0000	R^2
5	0.0100	IMEDA= SPEAI	0.0000	R^2	IMEDA= SPEAI	0.0000	R^2
6	0.0125	IMEDA= NSGAI	0.0000	R^2	IMEDA= NSGAI	0.0000	R^2
7	0.0167	IMEDA=PESAI	0.0000	R^2	IMEDA=PESAI	0.0000	R^2
8	0.0250	IMEDA=EMOBBO	0.0000	R^2	IMEDA=EMOBBO	0.0000	R^2
9	0.0500	IMEDA=MOTS	0.0000	R^2	IMEDA=MOTS	0.0000	R^2

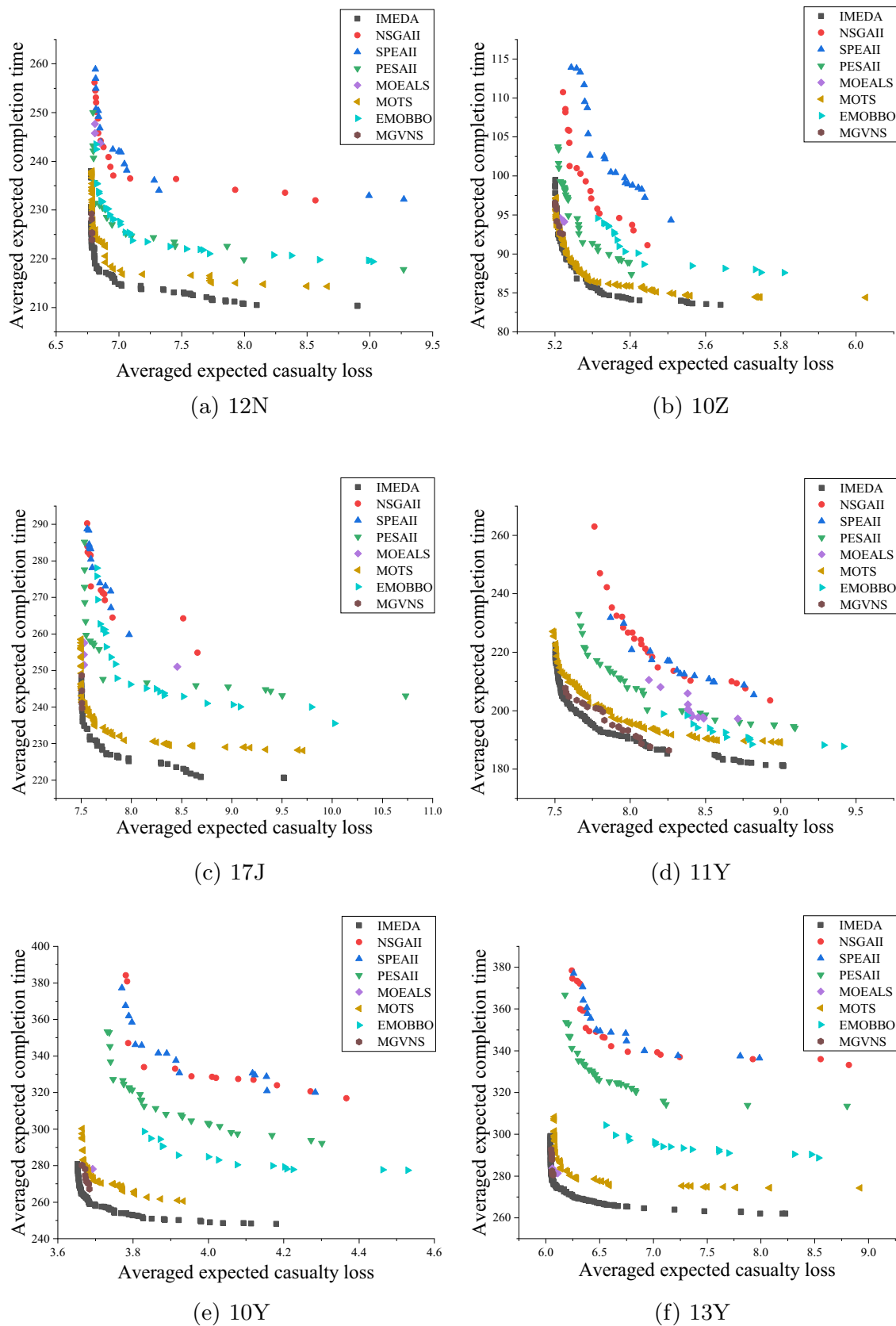


Fig. 9 Pareto fronts of compared algorithms for the test instances

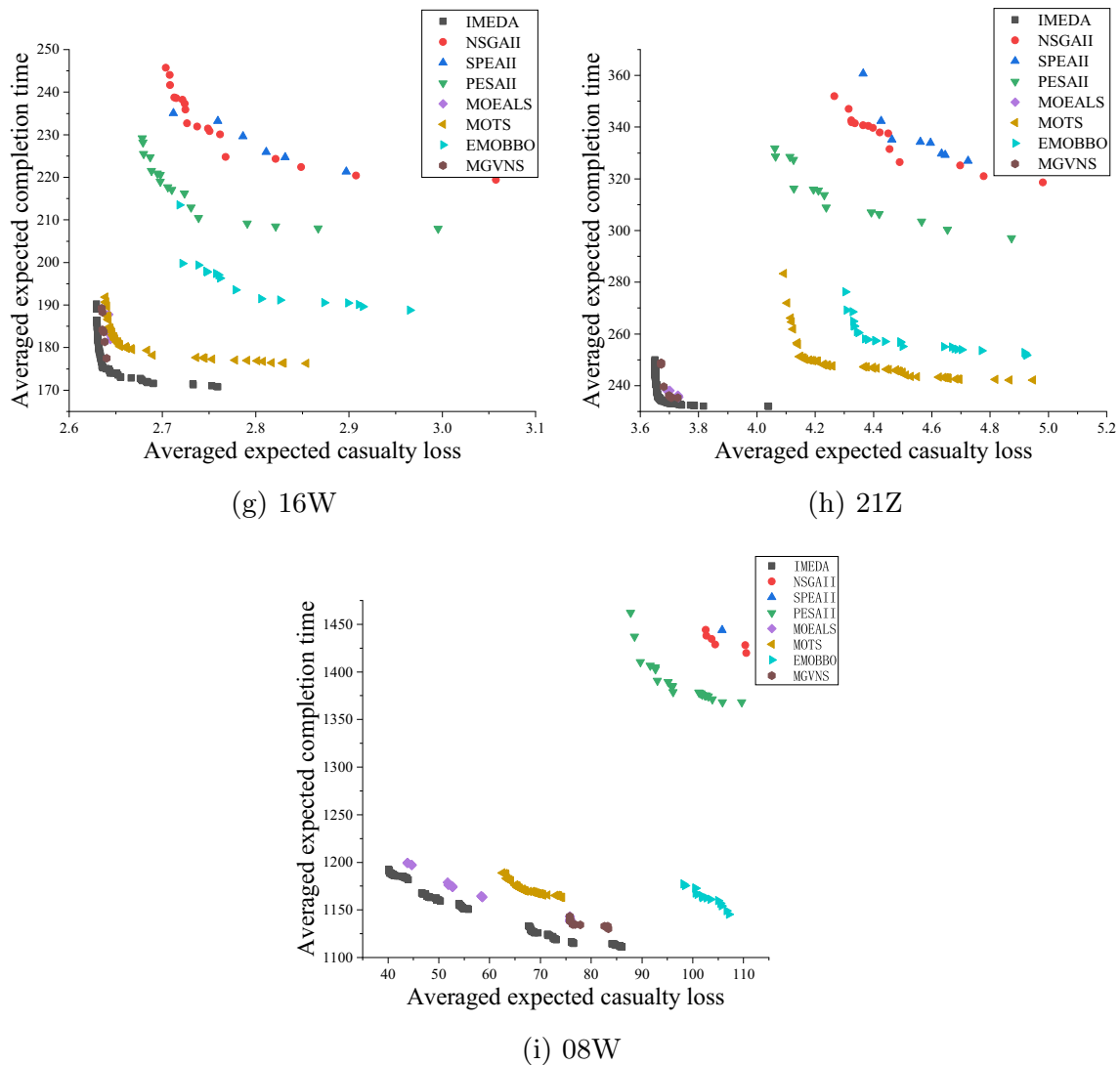


Fig. 9 continued

for a long time. How to assess and mitigate the operation risk for rescue teams is also a focus of research.

Our research provides an effective method for rescue task emergency scheduling problem. The scenarios widely exist in the real world, e.g., container allocation, maintenance task scheduling, etc. Consequently, further study of this algorithm will be helpful to solve this kind of practical problem efficiently.

For future work, a research direction may consider the collaboration of multiple types of rescue teams. Specially, at a rescue site after an earthquake, several rescue teams are responsible for clearing the outside barriers, while others are responsible for lifting the covers, and then, medical teams provide first aid treatment for the injured. Efficient

cooperation between different rescue teams is an important issue worthy of in-depth study. In our study, the decision space is task allocation and scheduling sequence, and objective function is the average of expected task completion time and casualty loss. Although the state of injured varies over time, neither the decision space nor objective function changes with time. However, various emergencies occur during rescue process, such as secondary disaster, breakdown of rescue machinery, etc., which requires re-planning of rescue tasks under the current condition. In this case, the decision space and objective function change with time, and evolutionary algorithms mentioned in our study are not suitable for this dynamic multi-objective optimization problem. Consequently, establishing a more realistic rescue operation model and developing efficient dynamic multi-objective optimization algorithm are an important direction in future work.

Acknowledgements This research was supported by the National Natural Science Foundation of China [grant number 61373057]; Zhejiang Natural Science Foundation [grant number LQ20F020025]; Ningbo Natural Science Foundation [grant number 202003N4073]; Public Welfare project of Ningbo City [grant numbers 202002N3139, 2019C10051].

Declarations

Conflict of interest On behalf of all the authors, the corresponding author states that there is no conflict of interest. On behalf of all the authors, the corresponding author states that there is no financial or non-financial interests that are directly or indirectly related to the work submitted for publication.

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

References

- Zhou L, Wu XH, Xu ZS, Fujita H (2018) Emergency decision making for natural disasters: an overview. *Int J Disast Risk Re* 27:567–576
- Zheng YJ, Ling HF, Xu XL, Chen SY (2015) Emergency scheduling of engineering rescue tasks in disaster relief operations and its application in China. *Int Trans Oper Res* 22:503–518
- Zhang SW, Guo H, Zhu K, Yu S, Li J (2017) Multistage assignment optimization for emergency rescue teams in the disaster chain. *Knowl Based Syst* 137:123–137
- Xiu XN, Tang H, Wu LX (2018) Satellite scheduling of large areal tasks for rapid response to natural disaster using a multi-objective genetic algorithm. *Int J Disast Risk* 28:813–825
- Wang CR, Zhao J, Xia E (2018) Multi-objective optimal design of a novel multi-function rescue attachment based on improved NSGA-II. *J Braz Soc Mech Sci* 40:344
- Chang FS, Wu JS, Lee CN, Shen HC (2014) Greedy-search-based multi-objective genetic algorithm for emergency logistics scheduling. *Expert Syst Appl* 41:2947–2956
- Wan F, Guo HX, Li JL, Gu MY, Pan WW (2021) Multi-objective emergency scheduling for geological disasters. *Nat Hazards* 105:1323–1358
- Wan F, Guo HX, Li JL, Gu MY, Pan WW, Ying YJ (2021) A scheduling and planning method for geological disasters. *Appl Soft Comput* 111:107712
- Nayeri S, Asadi-Gangraj E, Emami S (2019) Metaheuristic algorithms to allocate and schedule of the rescue units in the natural disaster with fatigue effect. *Eur J Comput Appl* 31:7517–7537
- Nayeri S, Gangraj EA, Emami S, Rezaeian J (2021) Designing a bi-objective decision support model for the disaster management. *Rairo-Oper Res* 55:3399–3426
- Zhao XJ, Xu W, Ma YJ, Hu FY (2015) Scenario-based multi-objective optimum allocation model for earthquake emergency shelters using a modified particle swarm optimization algorithm: a case study in Chaoyang district, Beijing, China. *PLoS ONE* 10:1–16
- Yan TR, Lu FQ, Wang SX, Wang LZ, Bi HL (2023) A hybrid metaheuristic algorithm for the multi-objective location-routing problem in the early post-disaster stage. *J Ind Manag Optim* 19:4663–4691
- Wu YF, Pan F, Li SX, Chen Z, Dong M (2020) Peer-induced fairness capacitated vehicle routing scheduling using a hybrid optimization ACO-VNS algorithm. *Soft Comput* 24:2201–2213
- Yu XB, Li CL, Yen GG (2021) A knee-guided differential evolution algorithm for unmanned aerial vehicle path planning in disaster management. *Appl Soft Comput* 98:106857
- Wang YD, Shi Q, Hu QW (2019) Dynamic multi-objective optimization for multi-period emergency logistics network. *J Intell Fuzzy Syst* 37:8471–8481
- Li JX, Fu HX, Lai KK, Ram B (2022) Optimization of multi-objective mobile emergency material allocation for sudden disasters. *Front Public Health* 10:927241
- Wan MR, Ye CM, Peng DJ (2023) Multi-period dynamic multi-objective emergency material distribution model under uncertain demand. *Eng Appl Artif Intel* 117:105530
- Tian GD, Fathollahi-Fard AM, Ren YP, Li ZW, Jiang XY (2022) Multi-objective scheduling of priority-based rescue vehicles to extinguish forest fires using a multi-objective discrete gravitational search algorithm. *Inform Sci* 608:578–596
- Zheng YJ, Chen SY, Ling HF (2013) Efficient multi-objective tabu search for emergency equipment maintenance scheduling in disaster rescue. *Optim Lett* 7:89–100
- Gharib Z, Yazdani M, Bozorgi-Amiri A, Tavakkoli-Moghaddam R, Taghipourian MJ (2022) Developing an integrated model for planning the delivery of construction materials to post-disaster reconstruction projects. *J Comput Des Eng* 9:1135–1156
- Zhou YW, Liu J, Zhang YT, Gan XH (2017) A multi-objective evolutionary algorithm for multi-period dynamic emergency resource scheduling problems. *Transport Res E-Log* 99:77–95
- Mark H, Martin P (2011) An introduction and survey of estimation of distribution algorithms. *Swarm. Evol Comput* 1:111–128
- Yang YH, Cheng X, Jiang JR, Jiang D, Li SJ (2018) Improved Alopex-based evolutionary algorithm by Gaussian copula estimation of distribution algorithm and its application to the Butterworth filter design. *Int J Syst Sci* 49:160–178
- Li YQ, Zhang GX, Cheng JX (2012) A modified estimation of distribution algorithm for digital filter design. *Rom J Inf Sci Tech* 15:50–62
- Ji SF, Luo RJ (2017) A hybrid estimation of distribution algorithm for multi-objective multi-sourcing intermodal transportation network design problem considering carbon emissions. *Sustainability* 9:1133–1156
- Shi W, Chen WN, Lin Y, Gu TL, Kwong S, Zhang J (2019) An adaptive estimation of distribution algorithm for multipolicy insurance investment planning. *IEEE Trans Evol Comput* 23:1–14
- Wang F, Li Y, Zhou A, Tang K (2020) An estimation of distribution algorithm for mixed-variable newsvendor problems. *IEEE Trans Evol Comput* 24:479–493
- Qi XW, Li K, Potter WD (2016) Estimation of distribution algorithm enhanced particle swarm optimization for water distribution network optimization. *Front Env Sci Eng* 10:341–351
- Chen ZG, Lin Y, Gong YJ, Zhan ZH, Zhang J (2021) Maximizing lifetime of range-adjustable wireless sensor networks: a neighborhood-based estimation of distribution algorithm. *IEEE Trans Cybern* 51:5433–5444
- Jiao DB, Ke LJ, Yang WB, Li J (2017) An estimation of distribution algorithm based dynamic clustering approach for wireless sensor networks. *Wireless Pers Commun* 97:4697–4727

31. Pena J, Lozano J, Larranaga P (2004) Unsupervised learning of Bayesian networks via estimation of distribution algorithms: an application to gene expression data clustering. *Int J Uncertain Fuzz* 12:63–82
32. Gao HJ, Li YT, Petr K, Xu H, Rodrigo MB (2020) A novel hybrid PSO-K-means clustering algorithm using Gaussian estimation of distribution method and levy flight. *IEEE Access* 8:122848–122863
33. Wang WX, Li KS, Jalil H, Wang H (2022) An improved estimation of distribution algorithm for multi-objective optimization problems with mixed-variable. *Neural Comput Appl* 34:19703–19721
34. Jiang QY, Cui JN, Wang L, Lin YY, Wu YL, Hei XH (2023) A regularity model-based multi-objective estimation of distribution memetic algorithm with auto-controllable population diversity. *Memet Comput* 15:45–70
35. Shao WS, Pi DC, Shao ZS (2019) A pareto-based estimation of distribution algorithm for solving multiobjective distributed no-wait flow-shop scheduling problem with sequence-dependent setup time. *IEEE Trans Autom Sci Eng* 16:1344–1360
36. Wang L, Wang S, Liu M (2013) A Pareto-based estimation of distribution algorithm for the multi-objective flexible job-shop scheduling problem. *Int J Prod Res* 51:3574–3592
37. Deng C, Hu R, Qian B, Jin HP (2021) Hybrid estimation of distribution algorithm for solving three-stage multiobjective integrated scheduling problem. *Complexity* 2021:5558949
38. Shao ZS, Pi DC, Shao WS (2018) Estimation of distribution algorithm with path relinking for the blocking flow-shop scheduling problem. *Eng Optim* 50:894–916
39. Chu X, Zhong QY, Khokhar SG (2015) Triage scheduling optimization for mass casualty and disaster response. *Asia Pac J Oper Res* 32:1550041
40. Bosman PAN, Thierens D (2003) The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans Evol Comput* 7:174–188
41. Pan QK, Ruiz R (2012) An estimation of distribution algorithm for lot-streaming flow shop problems with setup times. *Omega* 40:166–180
42. Shao WS, Shao ZS, Pi DC (2021) Multi-objective evolutionary algorithm based on multiple neighborhoods local search for multi-objective distributed hybrid flow shop scheduling problem. *Expert Syst Appl* 183:115453
43. Salhi A, Rodríguez JAV, Zhang Q (2007) An estimation of distribution algorithm with guided mutation for a complex flow shop scheduling problem, in *Proc. 9th Annu. Conf. Genetic Evol. Comput.*, 570–576
44. Chen SH, Chang PC, Cheng TCE, Zhang QF (2012) A self-guided genetic algorithm for permutation flowshop scheduling problems. *Comput Oper Res* 39:1450–1457
45. Chen SH, Chen MC (2013) Addressing the advantages of using ensemble probabilistic models in estimation of distribution algorithms for scheduling problems. *Int J Prod Econ* 141:24–33
46. Zhang Q, Sun J, Tsang E (2005) An evolutionary algorithm with guided mutation for the maximum clique problem. *IEEE Trans Evol Comput* 9:192–200
47. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans Evol Comput* 3:257–271
48. Ciavotta M, Minella G, Ruiz R (2013) Multi-objective sequence dependent setup times permutation flowshop: a new algorithm and a comprehensive study. *European. J Oper Res* 227:301–313
49. Minella G, Ruiz R, Ciavotta M (2008) A review and evaluation of multi-objective algorithms for the flowshop scheduling problem. *INFORMS J Comput* 20:451–471
50. Minella G, Ruiz R, Ciavotta M (2011) Restarted Iterated Pareto Greedy algorithm for multi-objective flowshop scheduling problems. *Comput Oper Res* 38:1521–1533
51. Zitzler E, Thiele L, Laumanns M, Fonseca CM, Fonseca VGD (2003) Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans Evol Comput* 7:117–132
52. Shao WS, Shao ZS, Pi DC (2022) Multi-local search-based general variable neighborhood search for distributed flow shop scheduling in heterogeneous multi-factories. *Appl Soft Comput* 125:109138
53. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGAII. *IEEE Trans Evol Comput* 6:182–197
54. Zitzler E, Laumanns M, Thiele L (2001) SPEA2: Improving the performance of the strength Pareto evolutionary algorithm, *Comput. Eng. Commun. Netw. Lab, Swiss Federal Inst. Technol., Zurich, Switzerland, Tech. Rep.* 103
55. Corne DW, Jerram NR, Knowles JD, Oates MJ, Martin J (2001) PEDDA-II: Region-based selection in evolutionary multi-objective optimization, in *Proc. 3rd Annu. Conf. Genet. Evol. Comput. (GECCO)*. San Francisco: Morgan Kaufmann, 283–290
56. Gao J, Chen R (2011) A hybrid genetic algorithm for the distributed permutation flowshop scheduling problem. *Int J Comput Intell Syst* 4:497–508

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