

## Research Article

# Robust Optimization for Integrated Construction Scheduling and Multiscale Resource Allocation

Qian Li <sup>1</sup>, Sha Tao <sup>1</sup>, Heap-Yih Chong,<sup>2</sup> and Zhijie Sasha Dong<sup>3</sup>

<sup>1</sup>School of Management & Engineering, Nanjing University, Nanjing 210093, China

<sup>2</sup>School of Built Environment, Curtin University, Perth, WA 6102, Australia

<sup>3</sup>Ingram School of Engineering, Texas State University, San Marcos, TX 78666, USA

Correspondence should be addressed to Sha Tao; [shirley\\_tao211@126.com](mailto:shirley_tao211@126.com)

Received 17 April 2018; Revised 6 June 2018; Accepted 13 June 2018; Published 24 July 2018

Academic Editor: Shoujun Huang

Copyright © 2018 Qian Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This research investigates an integrated problem of construction scheduling and resource allocation. Inspired by complex construction practices, multi-time scale resources are considered for different length of terms, such as permanent staff and temporary workers. Differing from the common stochastic optimization problems, the resource price is supposed to be an uncertain parameter of which probability distribution is unknown, but observed data is given. Hence, the problem here is called Data-Driven Construction Scheduling and Multiscale Resource Allocation Problem (DD-CS&MRAP). Based on likelihood robust optimization, a multiobjective programming is developed where project completion time and expected resource cost are minimized simultaneously. To solve the problem efficiently, a double-layer metaheuristic comprised of Multiple Objective Particle Swarm Optimization (MOPSO) and interior point method named MOPSO-interior point algorithm is designed. The new solution presentation scheme and decoding process are developed. Finally, a construction case is used to validate the proposed method. The experimental results indicate that the MOPSO-interior point algorithm can reduce resource cost and improve the efficiency of resource utilization.

## 1. Introduction

Construction project scheduling needs allocating time and resource to each construction activity subject to precedence relations, deadline, and other constraints. Poor scheduling is one of the common issues in the field of construction management [1]. Different project schedules lead to different resource demand curves over time. Take a simple instance in Figure 1, for example, two possible schedules that lead to different resource demand curves as represented by the blue dotted lines. Resource allocation plan would also lay restraints on the realization of project schedule [2, 3]. If some required resources are not available at that time, activities would fail to be completed on time.

In a construction practice, the required resources, such as manpower and equipment, often have two features. First, these resources could be available in multiple periods that are different in length called multi-time scale periods. For example, there are permanent staff and temporary

employees. The former, also called long-term employees, are available at all times during the project. By contrast, the latter can be hired and only available in some short term. Similarly, some equipment is purchased, but some other is rented on a short-term basis. The second feature describes that the price of resources often fluctuates with the market, such as the price of equipment renting and human hiring.

This research investigates the problem called Data-Driven Construction Scheduling and Multiscale Resource Allocation Problem (DD-CS&MRAP). A construction project consists of several activities with precedence relations. Execution of each activity requires a certain amount of time and resources. Resources can be available in multiple lengths of terms, which are called multi-(time) scale resources. Moreover, the resource price is a stochastic variable of which the distribution is unknown, only the observed data is given. The research aims to determine the start time of each activity and make a combined strategy of resource allocation in multiscale terms for minimizing two objectives, namely,

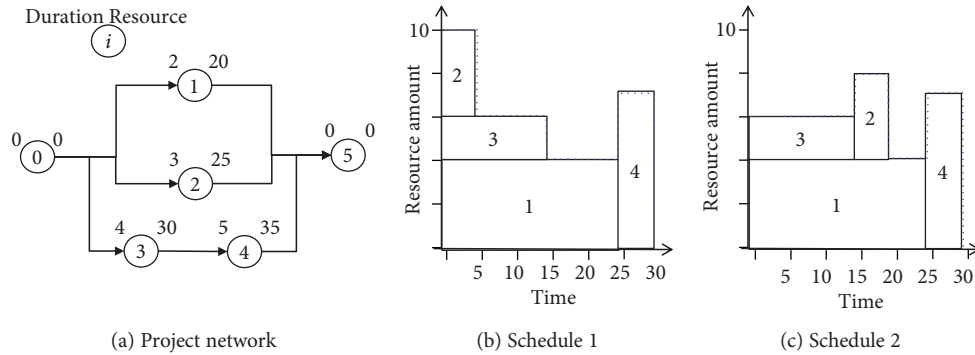


FIGURE 1: Simple instance.

(a) project completion time and (b) expected total resource cost. A double-layer metaheuristic consisted of Multiple Objective Particle Swarm Optimization (MOPSO), and interior point method is proposed to solve DD-CS&MRAP efficiently. The MOPSO is used at the external layer for generating a set of feasible solution about project scheduling plan and multiscale resource allocation plan, while the interior point method is used at the internal layer to determine the worst probability distribution of resource price. Subsequently, a complex construction case is used to validate the proposed metaheuristic.

The remainder is organized as follows. Section 2 articulates the related studies and concepts. Section 3 introduces the DD-CS&MRAP problem and the mathematical model. Section 4 describes MOPSO-interior point metaheuristic. Section 5 explains the validation of the proposed method. Section 6 discusses and concludes the findings.

## 2. Related Studies

**2.1. Construction Scheduling and Resource Allocation Problem.** Critical path method (CPM) is a widely used method. It is a duration-driven method to cope with scheduling problems where the required resources are abundant [4]. However, the limitation of the available resource makes it difficult for CPM. In a Resource-Constrained Project Scheduling Problem (RCPSP), the total available resource is known and fixed, namely, the optimal project scheduling plan is solved under the limited resources to achieve the shortest project goal [5]. Some project scheduling problems aim at optimizing resource-related objectives, such as Resource Levelling Problem (RLP) and Resource Investment Problem (RIP). RLP takes into account the balance of resource usage and makes the use of resources as balanced as possible within the plan during project scheduling, which means that resource requirements fluctuate as little as possible [6–8]. RIP performs project scheduling under the premise of meeting deadline to achieve the minimum total resource cost [9, 10], recently, towards multiskilled workers who have different skills at different speeds and costs. Javanmard et al. [11] then combined the multiskilled project scheduling problem with the RIP, which aims to obtain a parallel optimization strategy of project scheduling and skill's recruitment. Shahsavari et al. [12] combined RIP with the quantity discount problem to reduce the resource costs.

The decision problem of resource optimization aims to select and allocate resources rationally for completing the project on schedule and within the budget. The project resource optimization problem considers known conditions such as schedule, time window, and space location to formulate a resource optimization configuration plan, such as equipment rental, equipment selection, and equipment scheduling. Yeoh and Chua [13] investigated the crane selection and locating plans in the multistage construction and developed a four-dimensional scheduling model. K. Kim and KJ. Kim [14] used a multiagent system simulation method to simulate the project site and evaluate the equipment scheduling solution. Guillén-Burguete et al. [15] used the location information of active tasks and the transfer of work equipment to develop an optimal equipment scheduling model. Al Hattab et al. [16] investigated the near-real-time optimization scheduling problem of tower cranes and used the simulation model to perform metaheuristic algorithms which identified possible collisions.

In fact, the resource allocation plan often affects the project's progress. Unreasonable allocation of resources often results in resource bottlenecks, making it difficult to implement project activities as planned, which eventually leads to project delays. Project schedule also determines the resource allocation plan. A reasonable project scheduling plan can shorten the construction period and often improve the efficiency of resource utilization to some extents. Therefore, resource allocation problem and project scheduling problem are closely related to each other. Some scholars have integrated these two problems. Afshar-Nadjafi [17] investigated the multimode project scheduling problem under the condition of decision making of recruitment and release for resources. Tao et al. [18] investigated coupling project scheduling and resource flow allocation in repetitive projects considering site location and workspace congestion. Dong et al. [19] took the feature of flexible resource renting and leasing into project scheduling problem. The resource renting/leasing price is effected by resource scarcity in the market. In this research, we also investigate the integrated project scheduling and resource allocation problem.

In construction practices, some huge equipment, such as tower cranes, is frequently used. In order to avoid frequent rent relinquishment, equipment suppliers often adjust the lease price appropriately according to the length of the lease period. For example, the longer the lease period, the lower

the rental cost will be [20]. For human resource, a construction often involves two types of workforce, namely, permanent staff and temporary staff. Similarly, the prices of these two types of employees are different. These phenomena indicate that multiscale resources are common in the construction field. However, most of the current related studies are mostly resource allocation optimization problems on a certain time scale and do not consider resources of multiple time scales and the corresponding different prices. Based on this realistic background, this paper will study the resource allocation problem with multiscale resources named Multiscale Resource Allocation Problem (MRAP). It is worth noting that the major difference between MRAP and the classic resource allocation problems is that resources (even if they are the same type) can be available in multiple lengths of terms, such as short term, middle term, and long term. The goal is to jointly allocate multiscale resources in different time periods to meet resource requirements and minimize costs.

Many studies have investigated project scheduling problems in deterministic environments. The parameters, such as activity duration, human resource price, and resource availability, have been determined in advance instead of fluctuating. However, this is not realistic [21]. In order to get closer to the reality, some scholars proposed stochastic optimization or robust optimization for solving project scheduling problems in uncertain environments. Recently, Zheng et al. [22] dealt with the project scheduling problem with stochastic duration from the two sides of the contract to maximize net present value. A tabu search approach is developed to solve the problem. Ji and Yao [23] proposed a multiobjective project scheduling problem that considers the uncertainty of activity duration and resource allocation time. Genetic algorithm was used to solve the uncertain project scheduling model to meet the minimization of total cost. Tao et al. [24] investigated resource-constrained project scheduling problem with hierarchical alternatives considering stochastic activity durations. A stochastic chance constraint was developed. The sampling average approximation (SAA) and the population-based evolutionary artificial algae algorithm (AAA) are integrated to solve the problem [24]. Robust RCPSP considers a variety of uncertainties such as overtime or under expiration of activity duration and temporary shortage of resources [25]. Robustness refers to the ability of the system to maintain normal operation when there are uncertainties in the system, which is a solution evaluated using the realization of the uncertainty that is most unfavorable [26]. For the uncertainty of the activity time, Bruni et al. [27] proposed an adaptive robust optimization model and deduced the decision to minimize the worst case resource allocation under the polyhedral uncertainty sets. In this research, the resource price is supposed to be stochastic variable. Different from major existing literatures, here, we suppose that the only observed data of resource price is known. The likelihood robust optimization is used to cope with the situation.

*2.2. Related Concepts of Likelihood Robust Optimization.* The likelihood robust optimization is to cope with optimization

problems in an uncertain environment [28]. The distribution of the input parameter is unknown, but some observed/historical data is given. The decision is then made to optimize the expected objective value under the worst-case distribution which makes the observed/historical data to achieve a certain level of likelihood  $\gamma$ . The likelihood robust optimization can maintain robustness of the solution for any statistically likely outcomes and also avoid the overconservativeness.

Suppose  $\xi$  is a random variable taking values in a given set  $\Xi = (\hat{\xi}_1, \hat{\xi}_2, \dots, \hat{\xi}_L)$ . Independent  $\Phi$  samples of  $\xi$  is observed with  $\varphi_l$  occurrences of  $\hat{\xi}_l$ . The likelihood robust distribution set  $\mathfrak{D}(\gamma)$  is defined as

$$\mathfrak{D}(\gamma) = \mathbf{p} = (p_1, \dots, p_L) \mid \sum_{l=1}^L \varphi_l \log p_l \geq \gamma, \quad (1)$$

$$\sum_{l=1}^L p_l = 1, p_l \geq 0 \forall l = 1, \dots, L,$$

where  $\mathbf{p} = (p_1, \dots, p_L)$  represents the probability distribution of  $\xi$  and  $p_l$  is the probability of  $\xi = \hat{\xi}_l$ .  $\mathfrak{D}(\gamma)$  contains all probability distributions which make the observed data achieves an empirical likelihood of at least  $\exp(\gamma)$ , called likelihood robust distribution set with likelihood level  $\gamma$ .

Suppose that we attend to minimize an objective function  $f(\mathbf{x}, \xi)$  where  $\mathbf{x}$  is the vector of decision variable with feasible set  $\Psi$ . The general likelihood robust optimization model can be formulated.

$$\min_{\mathbf{x} \in \Psi} \left\{ \max_{\mathbf{p} \in \mathfrak{D}(\gamma)} \sum_{l=1}^L p_l f(\mathbf{x}, \hat{\xi}_l) \right\}, \quad (2)$$

where  $\sum_{l=1}^L p_l f(\mathbf{x}, \hat{\xi}_l)$  is the expected value of objective function. The likelihood robust optimization model makes decision on  $\mathbf{x}$  to minimize the expected function under the worst-case distribution in  $\mathfrak{D}(\gamma)$ .

According to the asymptotic analysis in the literature, an appropriate likelihood threshold  $\gamma$  can be computed by following (3) such that the likelihood robust distribution set has a confidence level of  $1 - \alpha$ .

$$\gamma = \sum_{l=1}^L \varphi_l \log \frac{\varphi_l}{\Phi} - \frac{1}{2} \chi_{n-1, 1-\alpha}^2, \quad (3)$$

where  $\chi_{n-1, 1-\alpha}^2$  is the  $1 - \alpha$  quantile of a  $\chi^2$  distribution with  $1 - n$  degrees of freedom.

### 3. Problem Statement and Formulation

*3.1. Problem Statement.* In the DD-CS&MRAP, a construction project consists of a group of activities with precedence relations. An AON (activity-on-node) network presentation of the project is given, denoted as  $G = \{\mathcal{N}, \mathbf{A}\}$  where  $\mathcal{N} = \{0, 1, 2, \dots, n, n+1\}$  represents the set of all activities and 0/1 matrix  $\{\mathbf{A}_{ij}\}$  represents the finish-to-start precedence. Let the activity 0 and  $n+1$  be the dummy start and finish activities of the project, respectively. Suppose that several

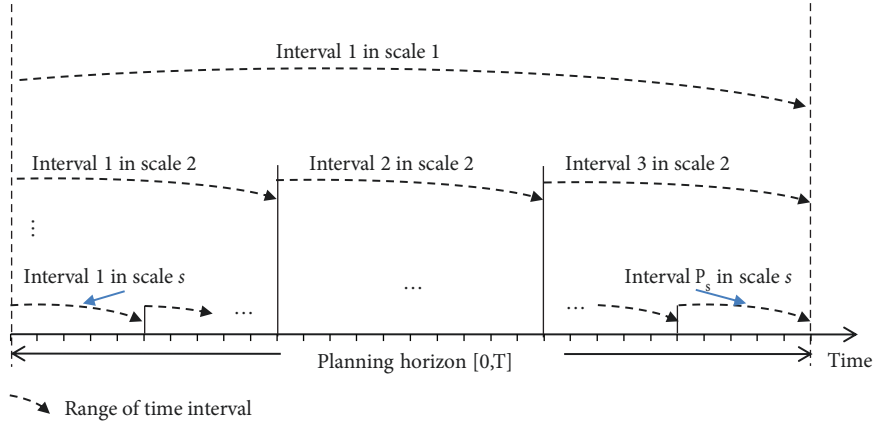


FIGURE 2: Intervals of multiple time scale.

types of limited resources are available during the planning horizon, such as labor crew and equipment. Execution of each activity needs certain duration  $D_i$  and resources  $R_{ir}$ . The dummy start and finish activities consume no resource or time. In order to model the problem as an integer programming, the planning horizon  $[0, T]$  is divided into a set of regular time slots. The length of each slot is the smallest unit of time, such as day and week. To allocate multiscale resources as the distinct feature of DD-CS&MRAP problem, the whole planning horizon is broken into several regular time intervals based on each type of time scale. There is a set of time scales  $S = \{1, \dots, S\}$  which is numbered based on the time interval length from the smallest to the largest. For any one time scale  $s$ , there is a set of time intervals  $\mathcal{P}_s$  that forms the whole planning horizon. For example, Figure 2 illustrates different time scales and the corresponding intervals. For the time scale 1, there is only one interval, namely, planning horizon. For the time scale 2, there are three intervals, which are numbered by 1, 2, and 3 in chronological order. In DD-CS&MRAP problem, how to schedule activities (determine start time of each activity) and allocate resources for each interval in each time scale should be determined. The objective of project scheduling is minimizing the project completion time as well as resource cost. Since the prices of different time scale resources are supposed to be uncertain, the multiscale resource allocation plan should be robust to minimize the expected total cost.

**3.2. Notations.** Table 1 shows all related notations of the model.

**3.3. Mathematical Model.** First, the uncertainty of the resource price is not considered. Let  $\xi_{rs}$  be deterministic scalar. The deterministic problem can be formulated as the following biobjectives Linear Integer Programming (LIP) [M1].

$$\begin{aligned}
 \text{[M1] min} \quad & \mathbf{F} = (f^T, f^C) \\
 \text{s.t.} \quad & f^T = \sum_{t \in T} t \cdot y_{n+1,t}, \quad (4)
 \end{aligned}$$

$$f^C = \sum_{s \in S, r \in R} \left( \xi_{rs} \cdot \sum_{k \in P_s} z_{skr} \right), \quad (5)$$

$$\mathbf{A}_{ij} \left( \sum_{t \in T} t \cdot y_{it} \right) \leq \sum_{t \in T} t \cdot y_{jt} - D_j \quad \forall i, j, \quad (6)$$

$$Q_{rt} = \sum_{s \in S, k \in P_s} B_{skt} \cdot z_{skr} \quad \forall r, t, \quad (7)$$

$$\sum_{i \in N} \sum_{\tau=t}^{t+D_i-1} R_{ir} \cdot y_{i\tau} \leq Q_{rt} \quad \forall r, t, \quad (8)$$

$$Q_{rt} \leq \bar{R}_r \quad \forall r, t, \quad (9)$$

$$f^T \leq T, \quad (10)$$

$$y_{it} \in \{0, 1\} \quad \forall i, t, \quad (11)$$

$$z_{skr} \in \mathbb{Z}^+ \quad \forall s, k, r. \quad (12)$$

The model is to minimize two objectives simultaneously: project completion time  $f^T$  and the total resource cost  $f^C$ , as presented in objective vector  $\mathbf{F}$ . Equation (4) describes that the project completion time is the finish time of the finish activity.  $\sum_{t \in T} t \cdot y_{it}$  calculates the finish time of activity  $i$ . Equation (5) shows that the total cost is the summation of the cost of allocation resources in all time scale periods. Constraint (6) presents finish-start precedence relations among the activities. Any one activity must not be started until all of its predecessors have been finished. Equation (7) calculates the total amount of resources allocated in time slot  $t$ , that is, the sum of resource allocated in different time scale intervals which contain time slot  $t$ . Constraint (8) means that in any time slot  $t$ , the total amount of resources allocated should be no less than the requirement. Constraint (9) makes sure that the allocated resource does not exceed the capacity  $\bar{R}_r$ . Constraint (10) guarantees that the process of the project should be within the whole planning horizon. Constraints (11) and (12) specify the feasible ranges for the decision variables.

Further,  $\xi_{rs}$  is assumed to be random variable. The probability distribution of  $\xi_{rs}$  is unknown, but there is some

TABLE 1: Notations.

Set and indices	
$\mathcal{N} = \{0, 1, \dots, n+1\}$	Set of activities topologically ordered and indexed by $i, j$
$\mathcal{R} = \{1, \dots, R\}$	Set of resource types indexed by $r$
$\mathcal{T} = \{0, 1, \dots, T\}$	Set of time slots indexed by $t$
$\mathcal{S} = \{1, \dots, S\}$	Set of time scales indexed by $s$
$\mathcal{P}_s = \{1, \dots, P_s\}$	Set of time intervals of scale $s$ indexed by $k$
Parameters	
$\mathbf{A}_{ij}$	Adjacency matrix: if activity $i$ is activity $j$ 's immediate predecessor, $\mathbf{A}_{ij} = 1$ ; otherwise, $\mathbf{A}_{ij} = 0$
$D_i$	Duration of completing activity $i$
$R_{ir}$	Quantity of resource $r$ required when executing activity $i$
$\bar{R}_r$	Capacity of resource $r$
$B_{skt} \in \{0, 1\}$	If time slot $t$ be in period $k$ of scale $s$ , $B_{skt} = 1$ ; otherwise, $B_{skt} = 0$
$\xi_{rs}$	The cost of input one unit of resource $r$ for a period of time scale $s$
$\Xi_{rs} = (\hat{\xi}_{rs}^1, \dots, \hat{\xi}_{rs}^L)$	Vector of possible values taken by $\xi_{rs}$ , $l \in \{1, \dots, L\}$
$\Phi_{rs}$	The number of total independent samples of $\xi_{rs}$
$\varphi_{rs}^l$	$\hat{\xi}_{rs}^l$ is observed with $\varphi_{rs}^l$ occurrences
$M$	An arbitrary big scalar
Variables	
$\mathbf{F}$	Objective vector of the two objectives
$f^T$	Project completion time
$f^C$	Resource cost
$\mathbf{y} = \{y_{it}\}, y_{it} \in \{0, 1\}$	If activity $i$ is completed at the end of time slot $t$ , $y_{it} = 1$ ; otherwise, 0
$\mathbf{z} = \{z_{skr}\}, z_{skr} \in \mathbb{Z}^+$	$z_{skr}$ is the amount of resource $r$ allocated in interval $k$ of time scale $s$
$\mathbf{p}_{rs} = \{p_{rs}^1, \dots, p_{rs}^L\}$	$p_{rs}^l$ is the probability of $\xi_{rs}$ being $\hat{\xi}_{rs}^l$ , $0 \leq p_{rs}^l \leq 1$
$\boldsymbol{\theta} = \{\theta_i\}, \theta_i \in \mathcal{T}$	$\theta_i$ is the start time of activity $i$
$Q_{rt}$	The total amount of resource $r$ allocated in time slot $t$

observed data drawn from the distribution.  $\xi_{rs}$  takes values in set  $\Xi_{rs} = (\hat{\xi}_{rs}^1, \dots, \hat{\xi}_{rs}^L)$ .  $\hat{\xi}_{rs}^l$  is observed with  $\varphi_{rs}^l$  occurrences. There are  $\Phi_{rs}$  independent samples. In [M1], random variable  $\xi_{rs}$  only appears in objective function  $f^C$ ; therefore, based on the general likelihood robust optimization model introduced in Section 2, the whole likelihood robust optimization model for the DD-CS&MRAP problem is formulated as following [M2].

$$\begin{aligned}
 \text{[M2] } \min \quad & \mathbf{F} = (f^T, \hat{f}^C) \\
 \text{s.t. } \quad & \hat{f}^C = \max \left\{ \sum_{s \in \mathcal{S}, r \in \mathcal{R}} \left( \left( \sum_{l \in \{1, \dots, L\}} p_{rs}^l \hat{\xi}_{rs}^l \right) \cdot \left( \sum_{p \in \mathcal{P}_s} z_{spr} \right) \right) \right\},
 \end{aligned} \tag{13}$$

$$\sum_{l=1}^L \varphi_{rs}^l \log p_{rs}^l \geq \gamma_{rs} \quad \forall s, r, \tag{14}$$

$$\gamma_{rs} = \sum_{l=1}^L \varphi_{rs}^l \log \frac{\varphi_{rs}^l}{\Phi_{rs}} - \frac{1}{2} \chi_{n-1, 1-\alpha}^2 \quad \forall s, r, \tag{15}$$

$$\sum_{l=1}^L p_{rs}^l = 1 \quad \forall s, r, \tag{16}$$

$$0 \leq p_{rs}^l \leq 1 \quad \forall l, s, r, \tag{17}$$

subject to constraints (4), (6), (7), (8), (9), (10), (11), and (12).

In [M2],  $\mathbf{p}_{rs} = \{p_{rs}^1, \dots, p_{rs}^L\}$  represents the distribution of random variable  $\xi_{rs}$ . Each element  $p_{rs}^l$  is the probability of  $\xi_{rs}$  being  $\hat{\xi}_{rs}^l$ .  $\mathbf{p}_{rs}$  is the decision variable that should be determined to make the expected cost maximized (the worst) as in objective (13). The likelihood robust distribution set of each random variable  $\xi_{rs}$  is defined by constraints (14), (15), (16), and (17). Constraint (14) makes the distribution achieve a certain level of likelihood  $\gamma_{rs}$ . The value of  $\gamma_{rs}$  is calculated based on (15), where  $1 - \alpha$  is the confidence level and

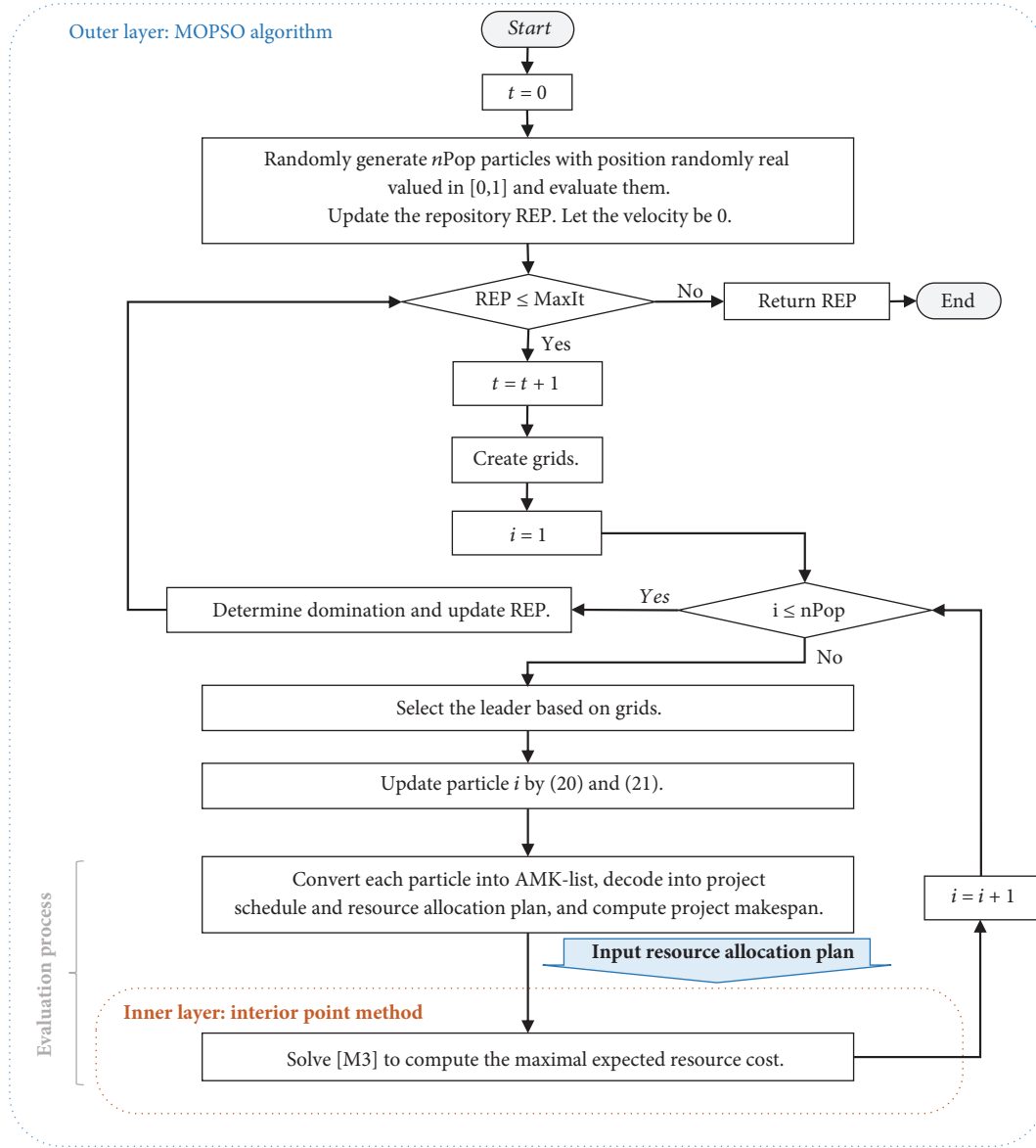


FIGURE 3: Flowchart of MOPSO-interior point metaheuristic.

$\chi_{n-1,1-\alpha}^2$  is the  $1 - \alpha$  quantile of a  $\chi^2$  distribution with  $1 - n$  degrees of freedom. Constraints (16) and (17) specify the feasible ranges for the distribution variables.

#### 4. MOPSO-Interior Point Metaheuristic

**4.1. Flowchart of MOPSO-Interior Point Algorithm.** The likelihood robust optimization model [M2] of DD-CS&MRAP is hard to solve directly due to the NP hardness as well as the nonlinear constraint (14). So in this section, a metaheuristic is proposed to cope with the problem. To lay the groundwork, the model [M2] is analyzed further here. In fact, [M2] consists of two nested submodels, the first model is to make decision on project scheduling and resource allocation, that is, determine decision variable  $(\mathbf{y}, \mathbf{z})$ , and the second one is to decide the worst-case distribution  $\mathbf{p}_{rs}$ . Suppose that a feasible solution  $(\mathbf{y}, \mathbf{z})$ , satisfying constraints (6), (7), (8),

(9), (10), (11), and (12), is given. The value of time objective  $f^T$  can be calculated easily. The rest problem is to solve the inner optimization model composed by (13), (14), (15), (16), and (17), which is defined as model [M3]. Model [M3] is a convex continuous optimization thus can be solved by some proven methods, such as interior point method, gradient method, and SDP.

Following the above logic, a double-layer metaheuristic is designed to solve DD-CS&MRAP problem as shown in Figure 3. The MOPSO is used at the external layer to generate a set of feasible solution consisting of two parts: project scheduling plan and multiscale resource allocation plan. Project completion time can also be computed. MOPSO is selected because it is easy to implement and a large number of experiments have proven that its performance is highly competitive when comparing it against other multiobjective evolutionary algorithms [29]. Then at the internal layer,

interior point method is used to solve convex continuous optimization model [M3] based on the resource allocation plan and calculate the worst distribution and the maximal expectation of resource cost. This metaheuristic is named as the MOPSO-interior point method. The whole metaheuristic returns a set of nondominated solutions stored in the final repository REP.

The interior point method has been proven mature for solving convex programming and can be realized directly using some software system, such as MATLAB. Therefore, it will not be discussed in detail in this article. Next, the solution representation and decoding method in MOPSO are introduced, respectively.

## 4.2. MOPSO

**4.2.1. Multiobjective Optimization.** A general multiobjective optimization problem can be defined in the following format.

Objective values  $\mathbf{f}(\mathbf{x}) = \min \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})\}$  ( $k \geq 2$ ),

$$\text{s.t. } g_i(\mathbf{x}) \geq 0 \quad \text{for } i = 1, \dots, m, \quad (18)$$

$$h_i(\mathbf{x}) = 0 \quad \text{for } i = 1, \dots, p, \quad (19)$$

where the constraints given by (18) and (19) define the feasible region  $S \in \mathbb{R}^n$  and the vector of decision variables  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  belongs to  $S$ . The above objective values involve  $k$  conflicting functions  $f_i: \mathbb{R}^n \rightarrow \mathbb{R}$  that we want to minimize simultaneously.

**Definition 1** (Pareto dominance [29]). A solution  $\mathbf{x}_1 \in S$  dominates a solution  $\mathbf{x}_2 \in S$  (denoted by  $\mathbf{x}_1 \prec \mathbf{x}_2$ ) if and only if  $f_i(\mathbf{x}_1)$  is partially less than  $f_i(\mathbf{x}_2)$ .

$$f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2) \quad \forall i = 1, \dots, k \cap f_i(\mathbf{x}_1) < f_i(\mathbf{x}_2), \exists i \in \{1, \dots, k\}. \quad (20)$$

**Definition 2** (Pareto optimality). A solution  $\mathbf{x}^* \in S$  is Pareto optimal if there exists no feasible solution  $\mathbf{x} \in S$  such that  $\mathbf{x} \prec \mathbf{x}^*$ .

Without information of project manager's preference, all the Pareto optimal solutions are equally good. The purpose of multiobjective optimization is to achieve the Pareto optimal solution set rather than finding a single solution.

**4.2.2. Particle Swarm Optimization.** The particle swarm optimization (PSO) algorithm is a population-based search algorithm based on the simulation of the social behavior of birds in a flock [30]. A particle is treated as a point in an  $n$ -dimension space, and the status of a particle is characterized by its position and velocity. The  $\text{POP}_t$  is the population of particles in  $t$  generation. The  $n$ -dimension position of the  $i$ th particle in the  $t$ th iteration can be denoted as  $\mathbf{x}^i = \{x_t^{i1}, x_t^{i2}, \dots, x_t^{in}\}$ . Similarly, the velocity  $\mathbf{v}^i = \{v_t^{i1}, v_t^{i2}, \dots, v_t^{in}\}$ . The particle-updating mechanism can be formulated as below.

$$\begin{aligned} \mathbf{v}_{t+1}^i &= w\mathbf{v}_t^i + c_1 r_1 (\mathbf{p}_t^i - \mathbf{x}_t^i) + c_2 r_2 (\mathbf{p}_t^{i,g} - \mathbf{x}_t^i), \\ \mathbf{x}_{t+1}^i &= \mathbf{x}_t^i + \mathbf{v}_t^i, \end{aligned} \quad (21)$$

where  $w$  is the inertia weight of the particle influencing the trade-off between the global and local experiences,  $c_1$  and  $c_2$  are the learning factors, and  $r_1, r_2 \in [0, 1]$  are random values.  $\mathbf{p}_t^{i,g}$  represents the global best position among all particles achieved so far.  $\mathbf{p}_t^i$  is the local best position of the  $i$ th particle encountered after  $t-1$  iterations. In the PSO, the performance of each particle is measured by a predefined fitness function, which is related to the problem to be solved.

**4.2.3. Multiobjective Particle Swarm Optimization.** The PSO can optimize the single-objective problem in a high speed of convergence, which is very suitable for multiobjective optimization problems. However, it is difficult to define the local optimal solution and the global optimal solution in each generation when there is no absolute global optimization. Therefore, the algorithm of the PSO is extended to handle the multiobjective optimization problem by incorporating a Pareto ranking scheme. The implementation steps of the MOPSO are shown in Algorithm 1. First, we introduce symbols and parameters involved in the algorithm:  $t$  represents generation index; MaxIt denotes the maximum number of iterations; POP denotes the population;  $n\text{Pop}$  denotes the population size; REP represents the repository which stores the nondominated particles' position;  $n\text{Rep}$  denotes the repository size;  $n\text{Grid}$  means the number of Grids; alpha means inflation rate; beta represents leader selection pressure; gamma represents deletion selection pressure; and Var min, Varmax defines the upper and lower limits of variables.

In the initial stage of the MOPSO, the initial population is randomly generated and evaluated. The repository REP is updated by function Update(POP, REP), which selects and stores nondominated particles from current population  $\text{POP}_t$  as well as REP. After that, MOPSO implements an iterative process until the maximum number of iterations has been reached. In each iteration, first of all, the grid is created. In the repository, objective function space is divided into many hypercubes and particles are located depending on its objective function values by using these hypercubes as a coordinate system. Then, population is updated based on above particle-updating mechanism (Steps 9-10), where the global best position  $\mathbf{p}_t^{i,g}$ /leader is obtained using the hypercubes and applying roulette wheel [29]. After that, the updated particle is bounded between the lower bound and upper bound, in order to maintain the particles within the search space (Steps 11-12). Then in Steps 14-18,  $\mathbf{p}_t^i$  which represents the historic best position of particle  $i$  is updated depending on the new position of particle  $\mathbf{x}_{t+1}^i$  and the last nondominated (best) position  $\mathbf{p}_t^i$ . Finally, all particles in the new population are evaluated and the repository REP is updated. Since the size of the repository is limited, whenever it gets full, a secondary criterion will be applied, those particles located in less populated areas of objective space are retained preferentially over those lying in more densely populated regions. After all

**Input:**  $MaxIt, nPop, nRep, w, c_1, c_2, nGrid, alpha, beta, gamma$ .  
**Output:** Non-dominated solutions in repository ( $REP$ ).  
**Initialization:**

1.  $REP = \emptyset; t = 0; POP_t = \{\mathbf{x}_t^i \mid i = \{1, \dots, nPop\}\}; \mathbf{p}_t^i = \mathbf{x}_t^i, \mathbf{v}_t^i = \mathbf{0}, \forall i = \{1, \dots, nPop\}$ ;
2. Evaluate  $POP_t$
3.  $REP = Update(POP_t, REP)$
4. While  $t \leq MaxIt$
5.  $t = t + 1$
6.  $G_t = CreateGrid(REP, nGrid, alpha)$
7. For  $i = 1 : nPOP$
8.  $\mathbf{p}_t^{i,g} = SelecLeader(REP, beta)$
9.  $\mathbf{v}_{t+1}^i = w\mathbf{v}_t^i + c_1r_1(\mathbf{p}_t^i - \mathbf{x}_t^i) + c_2r_2(\mathbf{p}_t^{i,g} - \mathbf{x}_t^i)$
10.  $\mathbf{x}_{t+1}^i = \mathbf{x}_t^i + \mathbf{v}_t^i$
11.  $\mathbf{x}_{t+1}^i = \max(\mathbf{x}_{t+1}^i, Varmin)$
12.  $\mathbf{x}_{t+1}^i = \min(\mathbf{x}_{t+1}^i, Varmax)$
13. Evaluate  $\mathbf{x}_{t+1}^i$
14. If  $\mathbf{x}_{t+1}^i$  dominates  $\mathbf{p}_t^i$
15.  $\mathbf{p}_{t+1}^i = \mathbf{x}_{t+1}^i$
16. Else
17.  $\mathbf{p}_{t+1}^i = \mathbf{p}_t^i$
18. End If
19. End For
20. Evaluate  $POP_t$
21.  $REP = Update(POP_t, REP)$
26. End While
27. Return  $REP$

ALGORITHM 1: MOPSO.

iterations are finished, the final output is the set of nondominated solutions stored in the final repository.

**4.3. Solution Representation and Decoding Method.** The solution representation consists of three parts: (1) activity list (A-list),  $AL = \{\mathbf{A}_1, \dots, \mathbf{A}_i, \dots, \mathbf{A}_n\}$ , where any element  $\mathbf{A}_i$  represents the order of activity  $i$ ,  $\mathbf{A}_i \in \{1, \dots, n\}$ ; (2) list of maximal resource supplied during the project process (M-list),  $ML = \{M_1, \dots, M_r, \dots, M_R\}, 0 \leq M_r \leq \bar{R}_r$  where any element  $M_r$  represents the maximum of resource  $r$  supplied; and (3) list of ratio of resource amount (K-list)  $KL = \{K_{1,1,1}, \dots, K_{skr}, \dots, K_{S-1, P_{S-1}, R}\}$ , where  $0 \leq K_{skr} \leq 1$  can be decoded in to the amount of resource  $r$  allocated in interval  $k$  of time scale  $s$ . It will be described later in decoding process. It is worth noting that the last/smallest time scale  $S$  and the corresponding intervals are not in the K-list, because the amount of resource allocated in the  $S$  time scale  $S$  intervals can be easily calculated when the amount of resources allocated in 1 to  $S-1$  time scale periods are determined. As a result, the solution representation is the connection of above three types of lists, denoted by AMK-list. Obviously, the length of the AMK-list is  $Length_{AMK} = n + R + (\sum_{s \in \mathcal{S}} P_s) \cdot R$ .

However, since MOPSO is originally designed to solve continuous optimization problem, we should map the real-valued particles to the AMK-list at first. Each particle of MOPSO is represented by vector  $\mathbf{x} = (x_1, \dots, x_{Length_{AMK}})$ ,  $x_i \in [0, 1]$ . Then, we need to convert each part of real-valued elements into the A-list, M-list, and K-list, respectively. As

for the first  $n$  elements,  $(x_1, \dots, x_n)$ , the smallest position value (SPV) rule [31] is utilized by sorting  $x_1, \dots, x_n$  from the smallest to the largest and then the corresponding rank is assigned for each activity according to its order. As for  $(x_{n+1}, \dots, x_{n+R})$ , the maximum of resource  $r$  supplied during the project process can be calculated by formula  $M_r = ROUND(\underline{R}_r + x_{r+n+2} \cdot (\bar{R}_r - \underline{R}_r))$ , where  $(\underline{R}_r, \bar{R}_r)$  are lower-upper bound of available resource and function  $ROUND()$  rounds real number to the nearest integer. As for remaining part of particle, since each element of K-list is real number in  $[0, 1]$ , let the corresponding part in  $\mathbf{x}$  equals to K-list. Thus, any real-valued particle  $\mathbf{x}$  can be mapped to the ACM list.

An example is illustrated in Figure 4. There are four activities which should be scheduled in planning horizon  $[0, 30]$ . There are three time scales as shown in Figure 2 in Section 3. The whole horizon  $[0, 30]$  is of time scale 1 (long term). There are three intervals of time scale 2 (middle term) and 6 intervals of time scale 3 (short term). The problem involves only one type of resource with a capacity of 10. Let  $(\underline{R}_r, \bar{R}_r) = (0, 10)$ . The real-valued list is converted into A-list = (1,3,2,4), M-list = (8), and K-list = (0.48, 0.95, 0.61, 0.13), respectively.

In order to translate the AMK-list into a feasible solution, a decoding process is developed. The decoding process contains two parts, activity schedule and multiscale resource allocation plan, which should be realized sequentially. Notations that appears in the text below can be referred to Table 1.



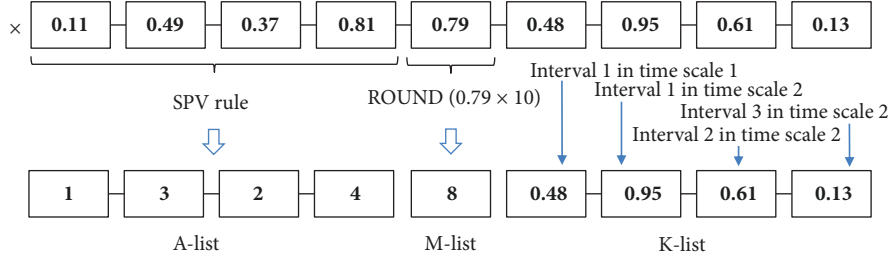


FIGURE 4: Example of converting a particle into AMK-list.

```

Input:  $\theta$ ,  $KL = \{K_{skr}\}$ ,  $R_{ir}$ ,  $D_i$ ,  $B_{skt}$ 
Output:  $\mathbf{z} = \{z_{skr}\}$ ,  $z_{skr} \in \mathbb{Z}^+$ , amount of resource  $r$  allocated in interval  $k$  of time scale  $s$ 
Initialization:  $\lambda_{rt} = 0, \forall r, t, z_{skr} = 0, \forall s, k, r$ 
1: For  $r = 1 : R$  Do
2:   For  $t = 1 : T$  Do
3:     For  $i = 1 : n$  Do
4:       If  $\theta_i < t$  and  $\theta_i + D_i \geq t$ 
5:          $\lambda_{rt} = \lambda_{rt} + R_{ir}$ 
6:       End If
7:     End For
8:   End For
9: End For
10:  $\bar{\lambda}_{rt} = \lambda_{rt}, \forall r, t$ 
11: For  $s = 1 : (S - 1)$  Do
12:   For  $k = 1 : P_s$  Do
13:      $z_{skr} = K_{skr} \cdot \max_{t \in \{1, \dots, T\}} \{\bar{\lambda}_{rt} \cdot B_{skt}\}, \forall r$ 
14:      $\{\bar{\lambda}_{rt} = \bar{\lambda}_{rt} - z_{skr} \mid B_{skt} = 1\}, \forall r$ 
15:   End For
13: End For
14:  $z_{skr} = \max_{t \in \{1, \dots, T\}} \{\bar{\lambda}_{rt} \cdot B_{skt}\}, \forall k \in \{1, \dots, P_s\}, r$ 
15: Return  $\mathbf{z} = \{z_{skr}\}$ 

```

ALGORITHM 2: Pseudocode of Stage 2.

*Stage 1 : Project schedule scheme.* First of all, we schedule activities with resource limitation and precedence constraints. The limitation of all types of resource is the M-list, which is the maximal amount of resource supplied. Then, the serial scheduling scheme (SSS) is applied as the common decoding method for the RCPSP problem. The SSS assigns the start time to each activity one by one with the principle that activity with higher rank in the A-list be scheduled earlier. For each activity, the start time is assigned without exceeding the resource limitation. Otherwise, the start time is pending until resource limitation is satisfied. Finally, stage 1 returns the start time of any activity  $i$ , denoted by  $\theta = \{\theta_i\}$ . Since the SSS has been extensively studied, we will not describe it in detail. Interested readers are referred to related literature of Zhang et al. [32].

*Stage 2 . Multiscale resource allocation scheme.* Based on the project schedule  $\theta$  generated in Stage 1 and the K-list, a plan of resource allocation in multiple time scales, denoted by  $\mathbf{z} = \{z_{skr}\}$ ,  $z_{skr} \in \mathbb{Z}^+$ , can be generated. The pseudocode of Stage 2 is presented in Algorithm 2. Let  $\lambda_{rt}$  be the total amount of resource  $r$  required in time slot  $t$ . First, we

calculate  $\lambda_{rt}$  by implementing Step 1 to Step 9 based on project schedule  $\theta$  as well as the given parameter  $R_{ir}$ . If  $\theta_i < t$  and  $\theta_i + D_i \geq t$ , activity  $i$  is executing in time slot  $t$  and requires  $R_{ir}$  resource. Second,  $S - 1$  cycles are conducted. In each cycle  $s$ , the resource allocation plan for time scale  $s$ , that is, the amount resource be allocated in each interval of time scale  $s$ , is determined. Define variable  $\bar{\lambda}_{rt}$  as the remaining resources that need to be allocated after  $s - 1$  cycles having being conducted.  $\bar{\lambda}_{rt}$  is initialized to be  $\lambda_{rt}$ . In any cycle  $s$ , for each interval  $k$ ,  $z_{skr} = K_{skr} \cdot \max_{t \in \{1, \dots, T\}} \{\bar{\lambda}_{rt} \cdot B_{skt}\}$ , resource  $r$  should be allocated. Then, update  $\bar{\lambda}_{rt}$  by abstracting the amount of allocated resource,  $\bar{\lambda}_{rt} = \bar{\lambda}_{rt} - z_{skr}$ . Finally, for the last time scale  $S$ , let  $z_{skr}$  be  $\max_{t \in \{1, \dots, T\}} \{\bar{\lambda}_{rt} \cdot B_{skt}\}$  to meet the remaining resource requirement.

In conclusion, by implementing the above two stages, the AMK-list can be finally decoded into a feasible solution for project scheduling and multiscale resource allocation.

Take the project instance in Figure 1, for example, to demonstrate the decoding process. Time scales are set as in Figure 2, and solution representation is the AMK-list in

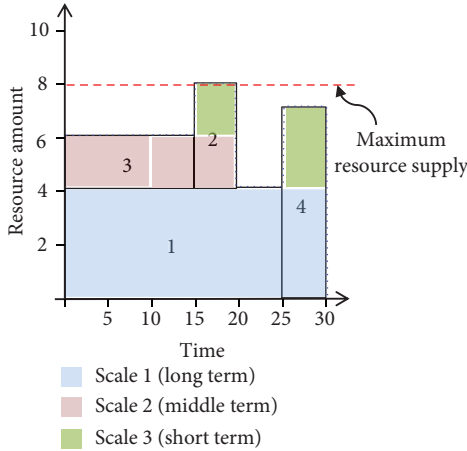


FIGURE 5: Feasible solution of DD-CS&MRAP.

Figure 4. In Stage 1, the SSS translate the A-list into a schedule by 4 cycles with resource capacity of 8 (M-list). Firstly, activity 1 is selected because its rank is number 1. Let activity 1 start at time 0 for the duration of 25 and resource usage of 4. Then, activity 3 is selected and scheduled at time 0 for the duration of 20, because resource supply is adequate. After that, let activity 2 start at time 15, because its resource requirement is 4 but the available resource in interval  $[0,15]$  is inadequate,  $8-4-2=2 < 4$ . Finally, let activity 4 start at time 25. So far, the project schedule is determined. The  $\lambda_t$  is also determined as the resource requirement histogram as represented by the blue dotted line in Figure 5. Then in Stage 2, the long-term (scale 1) interval  $[0,30]$  is firstly allocated with  $\text{ROUND}(0.48 \times 8) = 4$  units of resource. The remaining unallocated resources are updated by  $\bar{\lambda}_t = \lambda_t - 4\forall t$ . For time scale 2, allocate  $\text{ROUND}(0.95 \times 2) = 2$ ,  $\text{ROUND}(0.61 \times 4) = 2$ , and  $\text{ROUND}(0.13 \times 3) = 0$  units of resource into the three middle term intervals,  $[0,10]$ ,  $[10,20]$ , and  $[20,30]$ , respectively. Update  $\bar{\lambda}_{t \in [15,20]} = 2$ ,  $\bar{\lambda}_{t \in [25,30]} = 3$ . At last, for the short terms, resource should be allocated to meet make up for resource shortage (scale 3).

In sum, the feasible solution of DD-CS&MRAP including project schedule and multiscale resource allocation plan is obtained, as shown in Figure 5. The start time of activity 1, 2, 3, and 4 are 0, 15, 0, and 25, respectively. 4 units of resource are allocated in the long-term period  $[0,30]$ , 2 units of resource are allocated in the middle-term period  $[0,10]$  and  $[10,20]$ , respectively, and 2 and 3 units of resource are allocated in the short-term period  $[15, 20]$  and  $[25, 30]$ , respectively.

## 5. Case Study

5.1. *Computational Setup.* A construction case is used to validate the proposed metaheuristic. The project case consists of 33 activities (except dummy start and finish activity) with precedence relations. The project network is shown in Figure 6. The expected duration and resource requirement (human resource) of each activity are given

and also marked near the corresponding activity in the network. There were two types of human resource to comply with a more flexible approach of employment in the project, namely, permanent staff and temporary staff. Every permanent staff should participate the construction from its start to finish (long term). The temporary staff could be hired for day's work (short term). The permanent staff was paid by the month. Figure 7(a) illustrates the data set of monthly salary for permanent staff. Temporary staff was paid by the day, and the data set of daily salary was shown in Figure 7(b). The integrated problem of construction scheduling and human resource allocation (including long-term permanent staff allocation and short-term temporary staff allocation) needs to be solved. The planning horizon is estimated at 800 days, and human resource capacity is approximately 500.

All algorithms are coded and compiled in the MATLAB R2016a. Experiments are conducted on a PC with Intel® Core™ i5-5200U CPU @2.20 GHz processor and 8GB RAM. The double-layer metaheuristic is performed based on the parameter setting as shown in Table 2.

### 5.2. Results Analysis

5.2.1. *Pareto Analysis.* By conducting the double-layer metaheuristic five times, five sets of nondominated solutions are obtained. The average computing time is 1145.97 seconds. By gathering the nondominated solutions and omitting the dominated, repeated, and closer solutions from it, eight solutions are selected to approximate the Pareto solution set. The Pareto optimal front is drawn in Figure 8. It can be inferred that the project completion time and resource cost are conflicting objectives. With the project completion time increasing, the resource cost tends to decrease. In particular, when the project completion time increases from 722 to 758, the resource cost has a large slump. Each solution contains two plans. One is a schedule of all activities represented by Gantt chart. As shown in Figure 9, each grey bar represents an activity process from the start to finish time. The other one is a resource allocation plan as shown in Figure 10. The black edge curve of the grey area represents the resource demand curve over time, dotted blue curve represents the total amount of labor allocated in every day, and broken red line represents the amount of permanent worker allocated in the project. Thus, the amount of temporary worker allocated in each day equals to the total amount of labor minus the amount of permanent worker. That is the gap between the broken red curve and the dotted blue curve in Figure 10.

Any two solutions in the Pareto solution set are equally good, because the multiobjective optimization takes no regard of preference decision maker. The metaheuristic proposed here just provides solution set, construction manager can select a proper solution from it based on his/her preference further. Among these trade-off solutions, solution with objective value 675, 44.638 (named solution 1) and solution with objective value 758, 39.004 (named solution 2) are selected and a comparison was made.

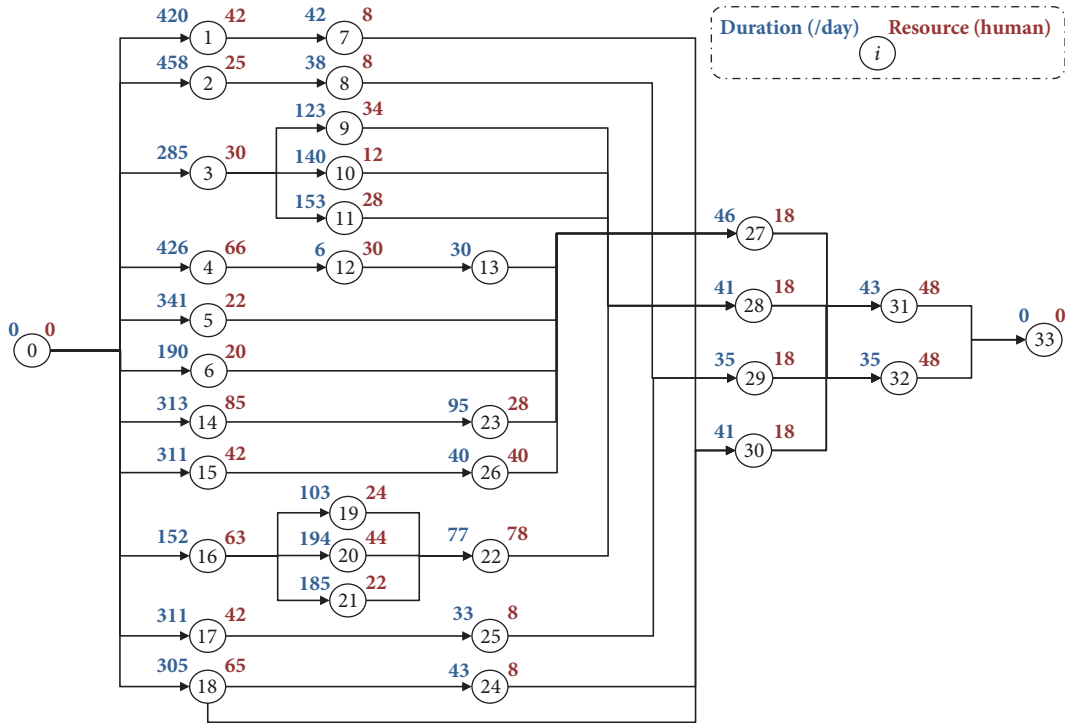


FIGURE 6: Network of construction project.

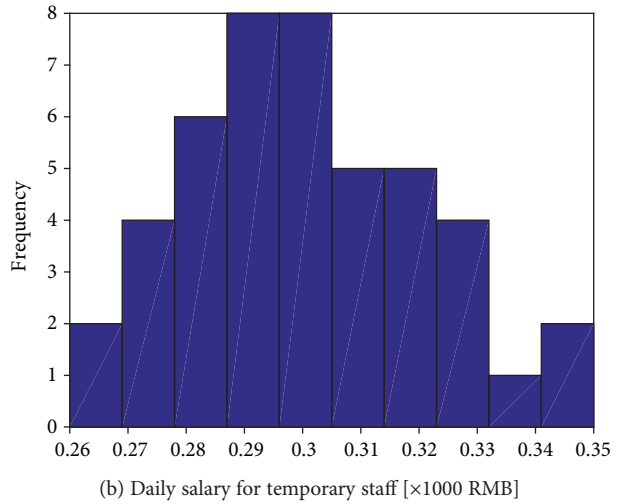
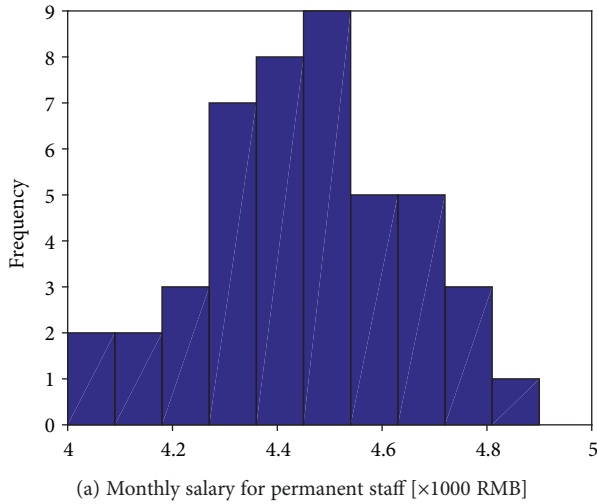


FIGURE 7: Histogram for the price data.

By referring to Figure 9, we can see that the activity progress bars of solution 2 are distributed evenly over time than those of solution 1. The solution 1's activity progress bars are mainly distributed before 500 days. Solution 2 then has a longer completion time, but with a smoother demand curve than in solution 1 as illustrated in Figure 10. It can also be observed that Figure 10(a) has a higher broken red line, which means more permanent workers are allocated in solution 1. Besides, the area between dotted blue curve and broken red line is smaller, which means more temporary

workers are hired in solution 1. Since solution 1 requires more manpower (permanent workers as well as temporary workers), the total labor cost of solution 1 is higher. In fact, there was no significant difference on cost of permanent worker between solution 1 and solution 2, which are 37.682 and 37.010, respectively. It is because, although more permanent worker inputting would lead to higher cost per month, the project duration would be shortened which helps in reducing the total cost of hiring permanent workers. However, the cost of temporary worker of solution 1 (6.956) is much more

TABLE 2: Parameter setting of metaheuristic.

Number	Symbol	Description	Value
1	MaxIt	Maximum number of iterations	500
2	$nPop$	Population size	20
3	$nRep$	Repository size	50
4	$w$	Inertia weight	0.5
5	$c_1$	Personal learning coefficient	1
6	$c_2$	Global learning coefficient	2
7	$nGrid$	Number of grids per dimension	7
8	alpha	Inflation rate	0.1
9	beta	Leader selection pressure	2
10	gamma	Deletion selection pressure	2
11	$(\underline{R}_r, \bar{R}_r)$	Upper-lower limits of resource	(100, 500)
12	$1 - \alpha$	Confidence level	0.9

than solution 2 (1.994). Therefore, the temporary worker allocation is the primary reason for cost gap between the two solutions.

*5.2.2. Comparison Multiscale Resource Allocation with Single-Scale Resource Allocation.* This comparison is conducted to validate the multiscale resource allocation strategy proposed in this study. The multiscale resource allocation strategy is named as strategy 1, which is to allocate permanent workers as well as temporary workers for the project. There are two kinds of single-scale resource allocation strategies. One is to allocate permanent workers only, which is named as strategy 2. The other is to allocate temporary workers only, which is named as strategy 3. We make a comparison of these three strategies.

We fix the project schedule plans of eight optimal solutions in Pareto set. It means that the resource demand curve over time for each solution is known. Then, we apply the three resource allocation strategies to each project schedule. We compare the three strategies from two aspects, resource cost and resource utilization rate. Table 3 reports the total cost of three strategies for each solution, the cost gaps, and the average values. For each solution, the total labor cost of strategy 1 is much less than that of strategy 2 and considerably less than that of strategy 3. The average gap between the strategy 1 and strategy 2 is 1.72 million yuan; however, the average gap between strategy 1 and strategy 3 reaches as high as 21.37 million yuan. Table 4 reports the resource utilization rate of strategy 1 and strategy 2 and their gaps. Resource utilization rate is the average percentage of labors that are working in total allocated labors during the whole construction. Obviously, strategy 3 must achieve 100% resource utilization rate and is therefore omitted from the table. It can be observed that the utilization rate of strategy 1 (average value is 76.0%) is higher than that of strategy 2 (average value is 73.1%). The average gap is 2.9%.

According to Tables 3 and 4, we can conclude that strategy 1 is more flexible than strategy 2. By combining

permanent labor assignment and temporary labor recruitment, strategy 1 can save costs and improve the efficiency of resource utilization. In addition, strategy 3 is quite flexible that can eliminate idle labor, but its labor cost is too high. Therefore, the strategy of multiscale resource allocation is able to reduce cost and guarantee the resource utilization efficiency to some extents.

*5.2.3. Comparison of Different Confidence Level.* This comparison is to explore the influence of confidence level  $1 - \alpha$ . Let confidence level be 0.900, 0.950, 0.975, 0.990, and 0.995. Take solution 1 as the example, the model [M3] is solved for achieving the probability distribution of labor price under the worst-case distribution with each confidence level. The distributions of permanent and temporary labor price under different confidence levels are shown in Figures 11(a) and 11(b), respectively. The probability distribution of price gets worse when the confidence level increases. For example, the probability of permanent labor salary being the most expensive (i.e., 4.9) equals to 0.022, 0.119, 0.133, 0.147, 0.174, and 0.188 when confidence level is at 0.900, 0.950, 0.975, 0.990, and 0.995, respectively. The probability of the occurrence of worst situation grows markedly when the confidence level increases. The expected cost of labor employment increases with confidence level increase as shown in Figure 12. Besides, the expected cost under frequency distribution is much less. It means that solutions under the higher confidence level could deal with the worse situation of higher resource cost. In other words, the higher the confidence level, the worse will be the case selected and the more robust the solution will be. Therefore, decision makers can adjust the confidence level, which is the parameter of double-layer metaheuristic, based on the risk preference to obtain the optimal solution with different levels of robustness. For example, the higher confidence level should be considered for more robust solutions, especially under conservative or risk adverse situations.

## 6. Discussions and Conclusion

The paper focuses on construction scheduling and resource allocation in an integrated framework. Multiscale resources and uncertainty in resource price are two distinct features in the problem. A multiobjective programming based on likelihood robust optimization is modeled to optimize the project completion time as well as the expected resource cost. A double-layer MOPSO-interior point metaheuristic is designed to solve the multiobjective optimization model. The metaheuristic provides a set of optimal solutions that make a trade-off between time and cost. It can be utilized by decision makers based on their objective preferences in all circumstances. Moreover, project managers also can formulate project schedule and resource allocation plans based on the project's risk preference by tuning parameter of confidence level in the metaheuristic. Finally, since the problem's distinct features are all inspired from practical constructions, it would be very promising in construction applications. The results of the case study have indicated that the method

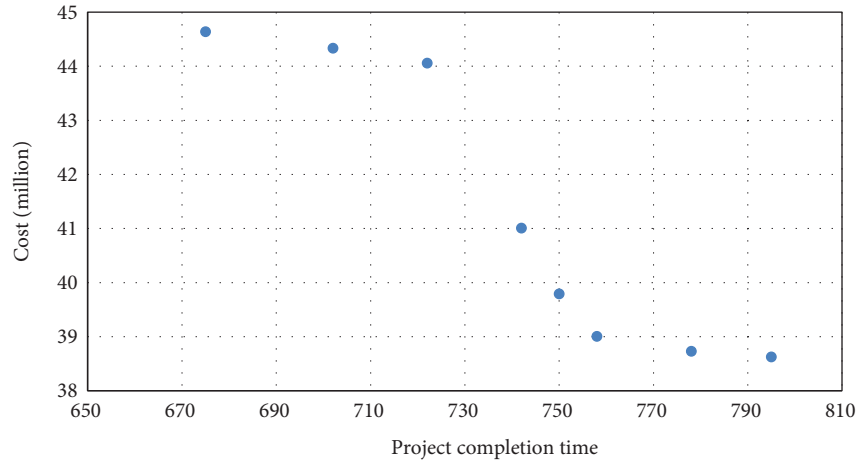
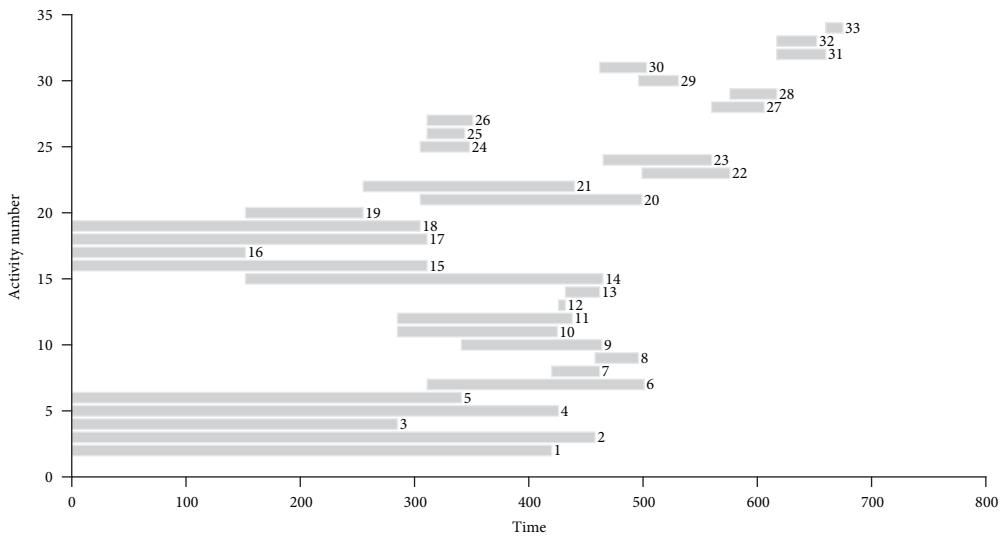
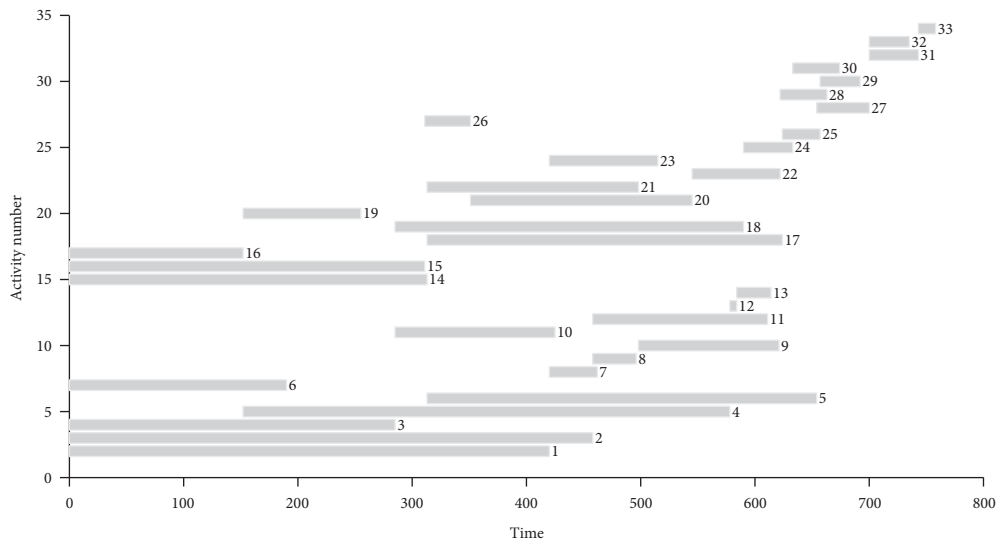


FIGURE 8: Pareto optimal front.

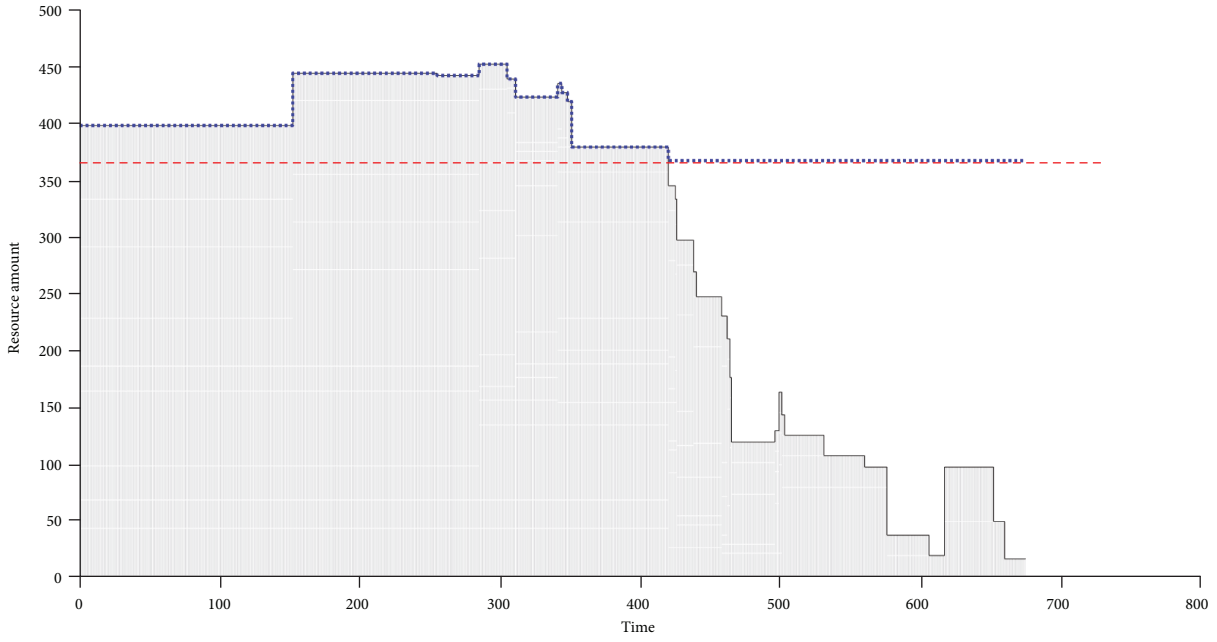


(a) Gantt chart of solution 1

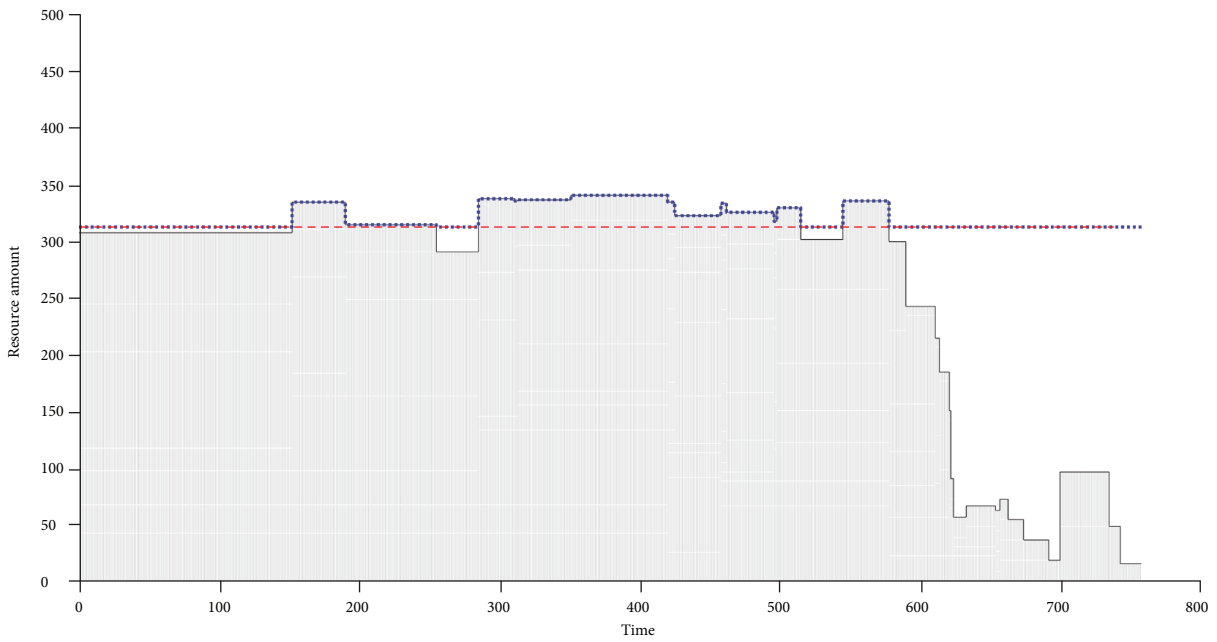


(b) Gantt chart of solution 2

FIGURE 9: Gantt chart of solution 1 and solution 2.



(a) Resource profile of solution 1



(b) Resource profile of solution 2

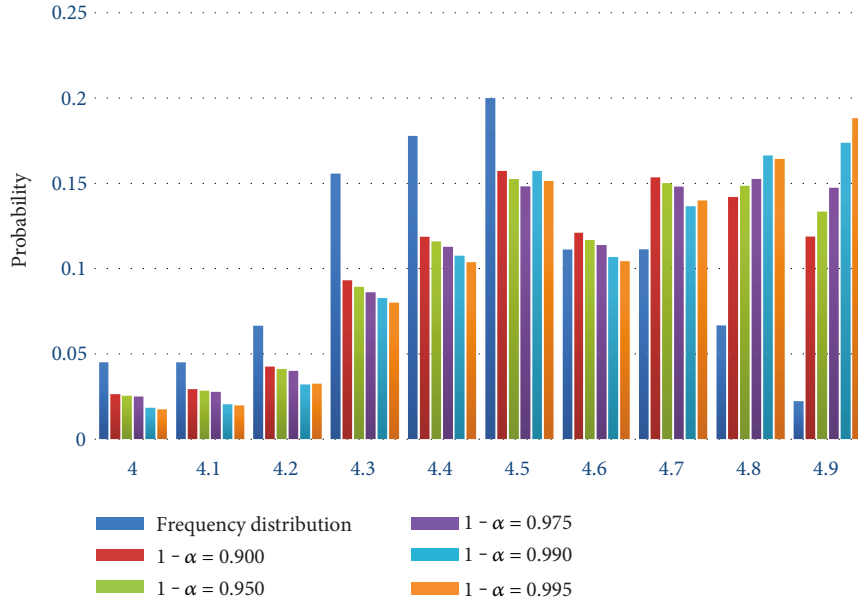
FIGURE 10: Resource profile of solution 1 and solution 2.

TABLE 3: Comparison of cost between multiscale and single-scale resource allocation strategy.

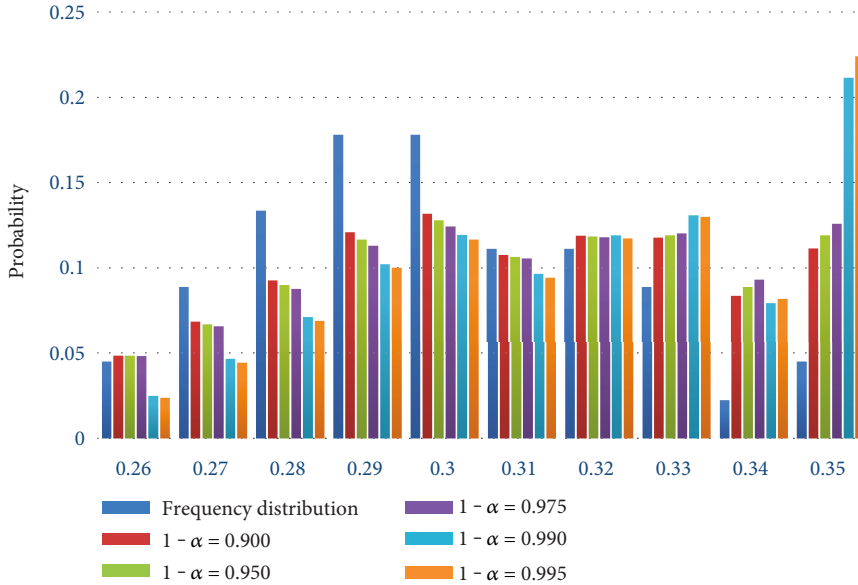
Solution number	1	2	3	4	5	6	7	8	Avg.
Strategy 1: allocate permanent and temporary workers	44.64	39.00	44.06	38.62	38.73	39.79	41.01	44.33	41.27
Strategy 2: allocate permanent workers only	47.34	40.33	45.86	40.40	40.57	40.95	42.45	46.01	42.99
Strategy 3: allocate temporary workers only	62.97	62.25	63.46	62.43	62.11	62.57	61.91	63.43	62.64
Gap between strategy 1 and strategy 2	2.71	1.33	1.81	1.78	1.85	1.16	1.45	1.67	1.72
Gap between strategy 1 and strategy 3	18.34	23.24	19.41	23.80	23.38	22.78	20.90	19.10	21.37

TABLE 4: Comparison of resource utilization rate between multiscale and single-scale resource allocation strategy.

Solution number	1	2	3	4	5	6	7	8	Avg.
Strategy 1: allocate permanent and temporary workers	0.728	0.811	0.720	0.827	0.812	0.807	0.760	0.714	0.760
Strategy 2: allocate permanent workers only	0.655	0.769	0.678	0.767	0.764	0.757	0.731	0.676	0.731
Gap between strategy 1 and strategy 2	0.073	0.042	0.042	0.06	0.048	0.05	0.029	0.038	0.029



(a) Monthly salary for permanent staff [×1000 RMB]



(b) Daily salary for temporary staff [×1000 RMB]

FIGURE 11: Distributions under different confidence levels.

proposed here can reduce resource cost as well as improve efficiency of resource utilization.

Certain limitations need to be considered in using the research findings. The proposed model would have different

outcomes under strict contractual requirements in traditional procurement system in terms of their fixed price provisions, which do not depend on resources or time used in the project. Moreover, uncertain market conditions would

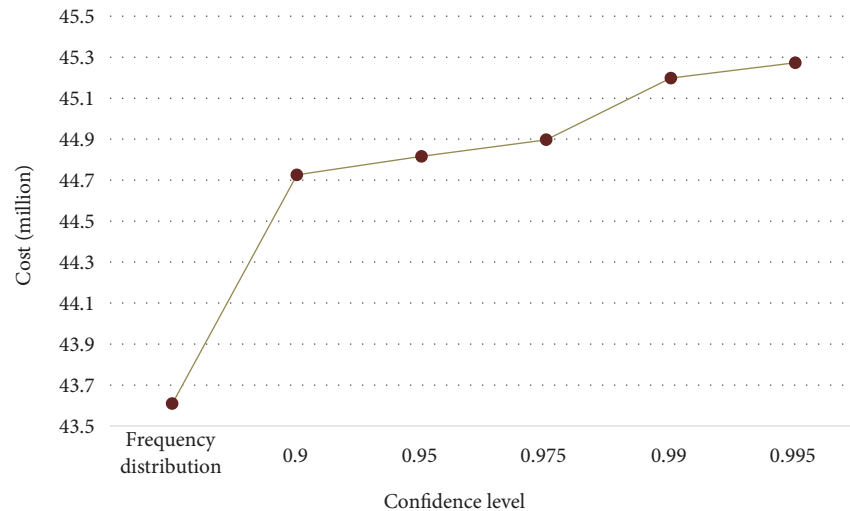


FIGURE 12: Expected total cost of labor employment under different confidence levels.

also affect the research findings. Future studies could be more automatic and dynamic for resource allocation by capturing the real-time data from the social-economic environment.

### Data Availability

The data used to support the findings of this study are included within the article.

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

### Acknowledgments

This work was jointly supported by the National Natural Science Foundation of China (71571098, 71390520, 71390521, 61473326, 71501102, 71732003, and 71701090), the Fundamental Research Funds for the Central Universities (14380023), and the Lab for Public Engineering Audit of Jiangsu Province, Nanjing Audit University (GGSS2016-01).

### References

- [1] S. S. Leu and C. H. Yang, "GA-based multicriteria optimal model for construction scheduling," *Journal of Construction Engineering and Management*, vol. 125, no. 6, pp. 420–427, 1999.
- [2] D. F. Cooper, "Heuristics for scheduling resource-constrained projects: an experimental investigation," *Management Science*, vol. 22, no. 11, pp. 1186–1194, 1976.
- [3] S. Tao and Z. S. Dong, "Scheduling resource-constrained project problem with alternative activity chains," *Computers & Industrial Engineering*, vol. 114, pp. 288–296, 2017.
- [4] P. A. Hofmann, "Critical path method: an important tool for coordinating clinical care," *Joint Commission Journal on Quality Improvement*, vol. 19, no. 7, pp. 235–246, 1993.
- [5] V. van Peteghem and M. Vanhoucke, "An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances," *European Journal of Operational Research*, vol. 235, no. 1, pp. 62–72, 2014.
- [6] E. T. Coughlan, M. E. Lübbecke, and J. Schulz, "A branch-price-and-cut algorithm for multi-mode resource leveling," *European Journal of Operational Research*, vol. 245, no. 1, pp. 70–80, 2015.
- [7] L. Bianco, M. Caramia, and S. Giordani, "Resource levelling in project scheduling with generalized precedence relationships and variable execution intensities," *Or Spectrum*, vol. 38, no. 2, pp. 405–425, 2016.
- [8] H. Li and E. Demeulemeester, "A genetic algorithm for the robust resource leveling problem," *Journal of Scheduling*, vol. 19, no. 1, pp. 43–60, 2016.
- [9] J. J. Qi, Y. J. Liu, P. Jiang, and B. Guo, "Schedule generation scheme for solving multi-mode resource availability cost problem by modified particle swarm optimization," *Journal of Scheduling*, vol. 18, no. 3, pp. 285–298, 2015.
- [10] A. Drexl and A. Kimms, "Optimization guided lower and upper bounds for the resource investment problem," *Journal of the Operational Research Society*, vol. 52, no. 3, pp. 340–351, 2001.
- [11] S. Javanmard, B. Afshar-Nadjafi, and S. T. A. Niaki, "Preemptive multi-skilled resource investment project scheduling problem: mathematical modelling and solution approaches," *Computers & Chemical Engineering*, vol. 96, pp. 55–68, 2017.
- [12] A. Shahsavar, N. Zoraghi, and B. Abbasi, "Integration of resource investment problem with quantity discount problem in material ordering for minimizing resource costs of projects," *Operational Research*, vol. 18, no. 2, pp. 315–342, 2018.
- [13] J. K. W. Yeoh and D. K. H. Chua, "Optimizing crane selection and location for multistage construction using a four-dimensional set cover approach," *Journal of Construction Engineering and Management*, vol. 143, no. 8, 2017.
- [14] K. Kim and K. J. Kim, "Multi-agent-based simulation system for construction operations with congested flows," *Automation in Construction*, vol. 19, no. 7, pp. 867–874, 2010.
- [15] S. Guillén-Burguete, H. Sánchez-Larios, and J. G. Vázquez-Vázquez, "An optimal transportation schedule of mobile equipment," *Journal of Applied Research and Technology*, vol. 10, no. 5, pp. 713–723, 2012.



- [16] M. Al Hattab, E. Zankoul, and F. R. Hamzeh, "Near-real-time optimization of overlapping tower crane operations: a model and case study," *Journal of Computing in Civil Engineering*, vol. 31, no. 4, 2017.
- [17] B. Afshar-Nadjafi, "Multi-mode resource availability cost problem with recruitment and release dates for resources," *Applied Mathematical Modelling*, vol. 38, no. 21-22, pp. 5347-5355, 2014.
- [18] S. Tao, C. Wu, Z. Sheng, and X. Wang, "Space-time repetitive project scheduling considering location and congestion," *Journal of Computing in Civil Engineering*, vol. 32, no. 3, 2018.
- [19] Z. S. Dong, S. Tao, and Q. Shi, "Coupling resource allocation and project scheduling considering resource scarcity," in *Proceeding of the 2018 IISE Annual Conference*, Orlando, FL, USA, 2018Forthcoming.
- [20] T. M. Simatupang and R. Sridharan, "A critical analysis of supply chain issues in construction heavy equipment," *International Journal of Construction Management*, vol. 16, no. 4, pp. 326-338, 2016.
- [21] S. Creemers, "Minimizing the expected makespan of a project with stochastic activity durations under resource constraints," *Journal of Scheduling*, vol. 18, no. 3, pp. 263-273, 2015.
- [22] W. B. Zheng, H. E. Zheng-Wen, and R. J. Liu, "Renewable resource constraint based max-npv project robust scheduling optimization with stochastic activity duration," *Journal of Industrial Engineering & Engineering Management*, no. 4, pp. 186-193, 2017.
- [23] X. Ji and K. Yao, *Uncertain Project Scheduling Problem with Resource Constraints*, Springer-Verlag New York, Inc., 2017.
- [24] S. Tao, C. Wu, Z. Sheng, and X. Wang, "Stochastic project scheduling with hierarchical alternatives," *Applied Mathematical Modelling*, vol. 58, pp. 181-202, 2018.
- [25] J. Wang, "Constraint-based schedule repair for product development projects with time-limited constraints," *International Journal of Production Economics*, vol. 95, no. 3, pp. 399-414, 2005.
- [26] V. Gabrel, C. Murat, and A. Thiele, "Recent advances in robust optimization: an overview," *European Journal of Operational Research*, vol. 235, no. 3, pp. 471-483, 2014.
- [27] M. E. Bruni, L. di Puglia Pugliese, P. Beraldi, and F. Guerriero, "An adjustable robust optimization model for the resource-constrained project scheduling problem with uncertain activity durations," *Omega*, vol. 71, pp. 66-84, 2017.
- [28] Z. Wang, P. W. Glynn, and Y. Ye, "Likelihood robust optimization for data-driven problems," *Computational Management Science*, vol. 13, no. 2, pp. 241-261, 2016.
- [29] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256-279, 2004.
- [30] M. R. Sierra and C. A. Coello Coello, "Improving PSO-based multi-objective optimization using crowding, mutation and  $\epsilon$ -dominance," in *Third International Conference on Evolutionary Multi-Criterion Optimization*, pp. 505-519, Berlin, Heidelberg, 2005.
- [31] H. Zhang, H. Li, and C. M. Tam, "Particle swarm optimization for resource-constrained project scheduling," *International Journal of Project Management*, vol. 24, no. 1, pp. 83-92, 2006.
- [32] H. Zhang, X. Li, H. Li, and F. Huang, "Particle swarm optimization-based schemes for resource-constrained project scheduling," *Automation in Construction*, vol. 14, no. 3, pp. 393-404, 2005.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

