IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 19 January 2024, accepted 16 February 2024, date of publication 23 February 2024, date of current version 29 February 2024.
Digital Object Identifier 10.1109/ACCESS.2024.3369177

RESEARCH ARTICLE

A Two-Layer Approach for the Decentralized Multi-Project Scheduling Problem Sharing Multi-Skilled Staff

WEIBAO YOU, ZHE XU, AND SONG ZHAO
School of Economics and Management, Beihang University, Beijing 100191, China
Corresponding author: Zhe Xu (zhexubuaa@163.com)

This work was supported by the National Natural Science Foundation of China under Grant 72271012.

ABSTRACT With the rapid development of economic globalization and information technology, the projects undertaken by enterprises are gradually becoming larger and more complex. The multi-project management model has become the norm in enterprise management, and it shows a trend of decentralization in terms of geographical distribution and management environment. This decentralization mainly manifests in the fact that each project has its own private information and benefit objectives and shares limited global resources with other projects during the execution process, forming a decentralized decision-making environment. As an extension of the resource-constrained project scheduling problem, the decentralized resource-constrained multi-project scheduling problem integrates single-project scheduling and global resource coordination allocation in a decentralized decision-making environment with multiple independent decision-makers. Furthermore, when considering global resources as multi-skilled staff, the decentralized multi-project scheduling problem sharing multi-skilled staff studied in this paper is proposed. A two-layer model containing local scheduling and global coordination decision-making is established to describe this problem. A two-layer approach (TLA) is proposed to solve this problem. In the local scheduling layer, a bat algorithm based on forward-backward scheduling (BAFBS) is developed to generate local baseline schedules to minimize the single-project completion time. In the global coordination decision-making layer, a variable neighborhood tabu search algorithm with greedy assignment strategy is designed to resolve global resource conflicts to minimize the multi-project total tardiness cost. Computational experiments are conducted based on the Multi-Project Scheduling Problem LIBrary dataset. The results show that the BAFBS can obtain high-quality local baseline schedules. Compared to the existing decentralized and centralized methods, our proposed TLA can get better solutions on most problem subsets, which proves that our approach can effectively coordinate the allocation of multi-skilled staff among multiple projects.

INDEX TERMS Decentralized multi-project scheduling, multi-skilled staff, two-layer approach, tabu search, bat algorithm.

I. INTRODUCTION

Project scheduling, as an important component of project management, can provide scientific decision-making support for project managers [1]. The resource-constrained project scheduling problem (RCPSP) studies the reasonable scheduling of the start time of all activities while satisfying the priority relationship constraints between activities and resource constraints to achieve the goals of project managers, such as minimizing project completion time or cost [2], [3], [4]. With the development of the global economy and the intensification of market competition, the scale and quantity of projects that enterprises need to manage simultaneously have also increased. The multi-project operation mode has become the primary way to manage complex tasks efficiently for many enterprises. As an extension of the RCPSP,

The associate editor coordinating the review of this manuscript and approving it for publication was Xueqin Jiang.

the resource-constrained multi-project scheduling problem (RCMPSP) has been widely studied in recent years [5], [6], [7], [8]. The RCMPSP not only needs to satisfy the priority relationship constraints and local resource constraints, but also involves the reasonable allocation of global resources among projects. In the case of limited global resource availability, it is necessary to meet the requirements of all project activities for global resources to construct a reasonable multi-project schedule. The centralized management method is the most commonly used to tackle RCMPSP [9], i.e., integrating all the activities of multiple projects into a super- or meta-network, and then a unified decision-maker with complete information on multiple projects makes a scheduling plan and allocates resources.

With further globalization and the rapid development of information technology, the dominant trend in multi-project management is toward decentralization [10], such as some large or multinational companies have branches in geographically distributed areas, each managed by its own manager. Based on the practical background of decentralized management, the decentralized resource-constrained multi-project scheduling problem (DRCMPSP) is proposed, which has been verified to be an NP-hard problem [11] and has received extensive attention from many scholars [12], [13], [14], [15]. In the DRCMPSP, multiple projects are managed in a decentralized decision-making environment. Specifically, the execution of each project requires local resources and global resources with limited availability. Local resources are only available to a single project, and global resources are shared among multiple projects. Each single project is directed by an individual project manager, i.e., the local decision-maker. Project managers do not exchange information with each other. Their only connection is to compete for global resources. Global resources are managed and allocated by a senior manager, i.e., the global decision-maker. Since the information is not shared among projects, it often leads to conflicts in using global resources. The senior manager needs to coordinate global resource allocation to eliminate conflicts. How to allocate global resources reasonably and reduce the adverse impact of limited resources on multi-project is the key of DRCMPSP research. When global resources are multi-skilled staff, the coordination of resources becomes more complex, such as the matching relationship of "activity-skill-resource" needs to be considered, i.e., the execution of activities requires specific skills, and skills are mastered by multi-skilled staff. Thereby, the decentralized resource-constrained multi-project scheduling problem sharing multi-skilled staff (DRCMPSP-MS) is proposed [16].

The application scenarios of the DRCMPSP-MS can be found in many real-life practices. Taking the software development projects presented by Yu et al. [16] as an example, introduce the scenarios involving project scheduling and human resource management and explain how these scenarios relate to DRCMPSP-MS. In order to satisfy user demands, software development enterprises often execute multiple software development projects in parallel, such as e-commerce projects, network office platform projects, etc. Each software development project is in charge of different project teams. The internal staff of each project team only execute their own projects. The leader of each project team (i.e., the project manager) is responsible for managing internal staff and making independent scheduling decisions for their own projects, including determining the allocation of internal staff and the start time of project tasks. Multi-skilled staff with coding, testing, and other development skills are shared among multiple software development projects and managed by the senior manager of multiple projects. Executing each project may require both internal staff and shared multi-skilled staff. Due to the non-sharing of scheduling information among project managers and the limited number of staff, it often leads to conflicts in the use of shared multi-skilled staff. The senior manager needs to coordinate the allocation of multi-skilled staff to resolve conflicts. Parallel execution of multiple software development projects is equivalent to scheduling multiple projects simultaneously in the DRCMPSP-MS. The internal staff of project teams and the multi-skilled staff shared among multiple projects are equivalent to the local resources and global resources in the DRCMPSP-MS. The project managers act as the local decision-makers of the DRCMPSP-MS, who master all information about internal activities and local resources, and determine the local scheduling plans and local resource allocation schemes. The senior manager corresponds to the global decision-maker in the DRCMPSP-MS, who masters global resource information, identifies resource conflicts among multiple projects, and coordinates global resource allocation. Project scheduling and human resource management are equivalent to determining the start time of all multi-project activities and developing allocation schemes for local and global resources in the DRCMPSP-MS.

The main reasons for solving the DRCMPSP-MS are as follows: on the one hand, the DRCMPSP-MS involving decentralized multi-project scheduling and multi-skilled staff assignment is relatively common in practical applications. The research in this paper can provide some methodological support and decision-making reference for project managers to solve such problems. Based on the reliable multi-project schedules and global resource allocation schemes, project managers can allocate resources to project activities effectively and facilitate the optimization of project performance objectives. On the other hand, as an extension of the DRCMPSP, the DRCMPSP-MS also belongs to NP-hard. Related studies on DRCMPSP-MS have been carried out in the literature [16], and the research in this paper has a certain theoretical foundation. Based on existing research, this paper designs a new approach to solve the DRCMPSP-MS in order to obtain better solutions and further enrich the theory and methods of decentralized multi-project scheduling.

In the DRCMPSP-MS, multiple projects are scheduled simultaneously. The execution of activities requires local and global resources. The local resources refer to the ordinary staff who possess only one skill and are managed within a single project. The global resources represent the multi-skilled staff who master several skills and are shared among multiple projects. Each single project is directed by a local decision-maker, which makes scheduling decisions according to the local information to minimize the project completion time (local objective). The global decision-maker is required to allocate global resources to minimize the multi-project total tardiness costs (global objective). In addition, the heterogeneous characteristics of multi-skilled staff and the decentralized management environment undoubtedly increase the complexity of the scheduling problem and make it more challenging to solve. We attempt to develop a new approach to tackle the DRCMPSP-MS. The main contributions are described as follows:

- Our work considers multi-skilled staff as shared resources in a decentralized multi-project scheduling environment. Moreover, a two-layer model containing local scheduling and global coordination decision-making is established to formulate the research problem.
- A two-layer approach (TLA) is developed to solve the DRCMPSP-MS. In the local layer, a bat algorithm based on forward-backward scheduling (BAFBS) is designed to handle the local scheduling problem. In the global layer, a variable neighborhood tabu search algorithm with greedy assignment strategy (VNTS-GAS) is proposed to resolve global resource conflicts, where the variable neighborhood tabu search identifies the execution order of conflicting activities and the greedy assignment strategy coordinates multi-skilled staff allocation.
- The performance of BAFBS is evaluated based on different-size problem instances. The experimental results demonstrate that the BAFBS can obtain high-quality local scheduling solutions. In addition, comparing the results with that of three state-of-the-art methods (TSA-SSM, PSGSMINSLK, and BRKGA) in the existing literature, the effectiveness of the proposed TLA is verified. When managers encounter such problems in practice, they can use our approach to make quick and effective decisions.

The remainder of the paper is organized as follows. Section II reviews the relevant literature. Section III provides the problem description and mathematical model. The proposed two-layer approach is presented in Section IV. Section V presents the computational experiments and results. Section VI draws conclusions and discusses future research directions.

II. LITERATURE REVIEW

The literature review contains the following parts: the decentralized multi-project scheduling problem and the multi-skilled project scheduling problem.

A. DECENTRALIZED MULTI-PROJECT SCHEDULING PROBLEM

The existing literature on the DRCMPSP focuses on developing coordinated approaches to solve global resource conflicts. Two main methods have been proposed: the auction-based mechanism and the negotiation-based mechanism.

In the auction-based mechanism, Lee et al. [17] found that large firms tended to manage multiple projects in a decentralized way and coordinated shared resource allocation in decentralized environments using an auction-based approach. Confessore et al. [11] introduced the DRCMPSP with local and global renewable resources and proposed the iterative combinatorial auction mechanisms to tackle the problem. Adhau et al. [12] developed a multi-unit combinatorial auction-based approach to minimize the average project delay. The local decision-makers submitted a bid consisting of the resource requirement of the eligible activity and the bid price, and the global decision-maker determined the winner of each auction round via a heuristic procedure. Liu et al. [18] proposed a combinatorial auction mechanism to solve the decentralized surgical scheduling problem with conflicting multiple renewable resource requirements. Other researchers [13], [19] have also used the auction-based mechanisms to deal with the DRCMPSP. In the negotiation-based mechanisms, Lau et al. [20] developed an agent-based model and negotiation-based algorithm for the DRCMPSP in supply chains. During the negotiation process, project agents and contractor agents proposed and counter-proposed operation start times iteratively until an acceptable agreement was achieved. Homberger [21] presented a restart evolution strategy for allocating global resources. Subsequently, Homberger and Fink [22] developed two generic negotiation mechanisms with side payments to address the DRCMPSP. One mechanism randomly generates high-quality solutions and divides monetary surpluses at the same stage. The other identified a set of Pareto optimal solutions, and the decision-maker requested a monetary surplus to choose the final solution. Li and Xu [23] proposed a two-stage decomposition approach to solve the DRCMPSP. In stage one, the initial local schedules were generated. In stage two, a sequential game-based negotiation mechanism was developed to coordinate shared resource conflicts. Some studies [24], [25] also designed different negotiation mechanisms to handle the DRCMPSP.

Some other project scheduling problems have been extended based on the DRCMPSP. Adhau et al. [26] and Zhao and Xu [27] further investigated the DRCMPSP with transfer times and transfer costs of global resources between multiple projects. Li et al. [9] addressed a new decentralized multi-project time-cost trade-off problem, where multiple projects competed for the limited global budget. Fu and Zhou [28] considered information asymmetry and resource heterogeneity based on the DRCMPSP. Liu and Xu [15] proposed a multi-PR heuristic for DRCMPSP without local resource constraints, in which activity durations were modeled as stochastic variables. Liu et al. [10] studied the DRCMPSP subject to global resource disruption. A three-stage algorithm with a task-scoring mechanism was proposed to solve the problem, which was generic for scheduling generation and repair processes. Zhao and Xu [14] designed a rollout policy-based approximate dynamic programming algorithm for solving the DRCMPSP with resource transfers and uncertain activity durations. The primary relation between the above studies and this paper is that they are all extension problems of DRCMPSP. The solving process of these extensions involves local project scheduling and global resource coordination allocation. The main difference is that different extension perspectives form different types of project scheduling problems, and the above studies further consider scenarios such as resource transfer, time-cost trade-off, and global resource disruption based on the DRCMPSP. In contrast, this paper focuses on global resources as multi-skilled staff on the basis of the DRCMPSP, which is not considered in the abovementioned studies. The main contributions of these studies are to further enrich the research content of the decentralized multi-project scheduling problem and design some effective decentralized scheduling methods, which provide method support and decision-making reference for project managers to solve related practical problems.

So far, only one paper has solved the DRCMPSP-MS [16]. The main relation between literature [16] and this paper is that both papers study the same scheduling problem. The main difference is that different approaches are designed to solve the DRCMPSP-MS. Literature [16] proposes a two-stage approach with softmax scoring mechanism to solve the problem. The local scheduling plans are obtained by the forward-backward scheduling genetic algorithm in the local scheduling stage, and the softmax scoring mechanism solves the global multi-skilled staff conflicts in the global decision stage. This paper develops a new TLA to solve the DRCMPSP-MS. Specifically, the BAFBS is designed to handle the local scheduling problem in the local layer. In the global layer, the VNTS-GAS is proposed to resolve global resource conflicts. The contributions of this paper are presented in the introduction section. The main contributions of literature [16] are to propose the DRCMPSP-MS in a decentralized multi-project scheduling environment and develop a two-stage approach with softmax scoring mechanism for solving the problem.

In addition, Yu et al. [29] studied the stochastic decentralized resource-constrained multi-project scheduling problem with multi-skilled staff (SDRCMPSP-MS). The main relation is that the SDRCMPSP-MS is the further extension of the DRCMPSP-MS studied in this paper, i.e., based on the DRCMPSP-MS, literature [29] further considers that the activity durations are uncertain and uses stochastic scheduling optimization methods to solve the problem. The main differences are that this paper and literature [29] studied the different problems of DRCMPSP-MS in deterministic environments and duration-uncertain environments, respectively. And different methods are designed to solve the corresponding problems. Literature [29] proposes a two-stage algorithm based on 12 priority rules to tackle the SDRCMPSP-MS. The main contribution of literature [29] is using the static stochastic scheduling optimization method based on priority rules to solve the DRCMPSP-MS under activity duration uncertainty.

Moreover, You et al. [30] studied the robust decentralized resource-constrained multi-project scheduling problem with multi-skilled staff (RDRCMPSP-MS). The main relation is that the RDRCMPSP-MS is also an extension of the DRCMPSP-MS. Both literatures [29] and [30] further consider the uncertainty of activity durations based on the DRCMPSP-MS. This paper and literature [30] studied the different problems of DRCMPSP-MS in deterministic environments and uncertain environments, respectively. The main differences between literatures [29] and [30] are that literature [30] employs the robust project scheduling to tackle activity duration uncertainty, generating robust baseline scheduling plans without considering that the durations obey a specific probability distribution. In contrast, literature [29] solves the problem using the stochastic project scheduling, in which activity durations are assumed to obey known distributions and scheduling strategies are obtained instead of baseline schedules. The main contribution of literature [30] is using the robust scheduling optimization method combining time buffer addition and robust resource allocation to solve the DRCMPSP-MS under activity duration uncertainty.

B. MULTI-SKILLED PROJECT SCHEDULING PROBLEM

Research on the multi-skilled resource-constrained project scheduling problem (MS-RCPSP) primarily focuses on single-project environments. Hegazy et al. [31] were the first to propose the MS-RCPSP, and they implemented resource substitution rules in order to include resources that can handle multiple skills. Néron [32] proposed the skillsets to characterize multi-skilled resources instead of resource substitution rules and presented two possible lower bounds for the problem. Then, some exact approaches and heuristic algorithms have been developed for solving the MS-RCPSP, including the branch-and-bound algorithm [33], hybrid Benders decomposition approach [34], branch-and-price algorithm [35], pareto-based grey wolf optimizer algorithm [36], and genetic algorithm [37], etc. Furthermore, Almeida et al. [38] and Snauwaert and Vanhoucke [39] studied different integer and mixed-integer linear programming formulations for the MS-RCPSP. Myszkowski et al. [40] and Snauwaert and Vanhoucke [41] generated new datasets based on multi-skilled resource parameters for the MS-RCPSP. More recently, research on MS-RCPSP has been extended to multi-skilled resource-heterogeneous project scheduling problems [42], [43], and multi-objective multi-skilled project scheduling problems [44], [45].

Compared to single-project environments, studies about the MS-RCPSP in the multi-project environment are relatively less. Kolisch and Heimerl [46] considered scheduling

IT projects and assigning multi-skilled resources simultaneously. They developed a meta-heuristic algorithm consisting of the genetic algorithm and the tabu search postprocessor to solve this problem. Walter and Zimmermann [47] constructed a mixed-integer linear programming model to minimize the multi-project team size and applied centralized scheduling to assign multi-skilled heterogeneous staff. Felberbauer et al. [48] extended the deterministic model developed by Kolisch and Heimerl [46]. They presented two stochastic optimization approaches to solve the multi-skilled multi-project problem in an uncertain environment. Cui et al. [6] developed a variable neighborhood search algorithm to solve multi-model multi-skilled multi-project scheduling problems. Haroune et al. [49] proposed a local search and a tabu search algorithm to tackle the multi-project scheduling and multi-skilled employees assignment problem with hard and soft constraints. Some studies [50], [51] have proposed different approaches for solving the multi-skilled multi-project scheduling problem. The above literature mainly employs centralized scheduling to solve MS-RCPSP in multi-project environments. Since information is not shared between decision-makers in a decentralized environment, the centralized scheduling methods with only one decision-maker on MS-RCPSP are unsuitable for solving the decentralized problem with multiple decision-makers.

To our knowledge, only one paper [16] addressed the DRCMPSP-MS using a two-stage approach with softmax scoring mechanism. Our primary work in this paper is to develop a new approach expecting to obtain better solutions for the DRCMPSP-MS.

III. MODEL FORMULATION

In this section, the DRCMPSP-MS is introduced in detail, and a two-layer model is established.

A. PROBLEM DESCRIPTION

The DRCMPSP-MS considers n projects that need to be scheduled simultaneously. Each project i has a release time $ad_i \geq 0$ denoting its earliest possible start time. And each project i consists of J_i non-dummy activities where activity a_{ij} indicates the jth activity of project i. F_{ij} represents the predecessor activity set of activity a_{ij}, and any activity cannot start until all its predecessor activities have been completed. Activities a_{i0} and $a_{i(J_i+1)}$ are two dummy activities that represent the beginning and completion of project i, respectively. The dummy activities have a duration of zero and no resource demand. Each non-dummy activity requires several types of local resources and at most one type of global resource. The local resource stands for the ordinary staff who has only one skill. The global resource refers to the multi-skilled staff. The multi-skilled staff has the characteristics of heterogeneity, i.e., each staff may master several skills and have different levels for each skill. Moreover, each multi-skilled staff can use at most one skill to perform an activity at a time. Local resources are only managed within a single project, and all projects share global resources. There is no connection among projects except for shared global resources.

After each single project arrives, project managers make local decisions without considering global resource constraints to obtain local baseline schedules and submit global resource requirement information to the senior manager based on the local scheduling results. Then, under satisfying the activity-skill-resource matching relationship, the senior manager allocates global resources to each project according to the global resource information and the global objective. The coordination results are fed back to the project managers and each project manager adjusts their respective local baseline schedules. Through repeated communication between senior manager and project managers, until all activities are scheduled, the multi-project baseline scheduling plan and resource allocation scheme are obtained. The scheduling framework of the DRCMPSP-MS is shown in Fig. 1. The notations are summarized in Table 1.

B. TWO-LAYER MODEL

The two-layer model integrating local scheduling and global coordination decision-making is formulated based on the multi-agent system (MAS). A MAS is a decentralized system that includes a set of independent, autonomous, and self-interested agents [52]. Here, the MAS consists of several project agents (PAs) and a coordinating agent (CA). PAs correspond to project managers, and CA represents the senior manager.

1) LOCAL SCHEDULING MODEL

Each PA generates an local baseline schedule to minimize the project completion time. The local model for project $i (i = 1, 2, \ldots, n)$ is the following.

$$\min \quad \sum_{t=0}^{T} t \cdot x_{i(J_i+1)t} \tag{1}$$

$$\text{s.t.} \quad \sum_{t=0}^{T} x_{ijt} = 1, \ \forall j \in V_i \tag{2}$$

$$\sum_{j \in V_i} \sum_{q'=t}^{t+\bar{d}_{ij}-1} x_{ijq'} \cdot w_{ij}^{r} \leq W_{ir}, \quad \forall r \in R_i, \ \forall t \in [0, T] \tag{3}$$

$$\sum_{t=0}^{T} \left(t - \bar{d}_{ij}\right) \cdot x_{ijt} \geq \sum_{t=0}^{T} t \cdot x_{iht}, \quad \forall a_{ih} \in F_{ij}, \ \forall j \in V_i \tag{4}$$

$$\sum_{t=0}^{T} t \cdot x_{i0t} \geq ad_i \tag{5}$$

$$x_{ijt} \in \{0, 1\} \tag{6}$$

The local objective function (1) minimizes the project completion time, i.e., minimizes the end time of dummy finish activity. $x_{i(J_i+1)t}$ denotes the binary decision variable,

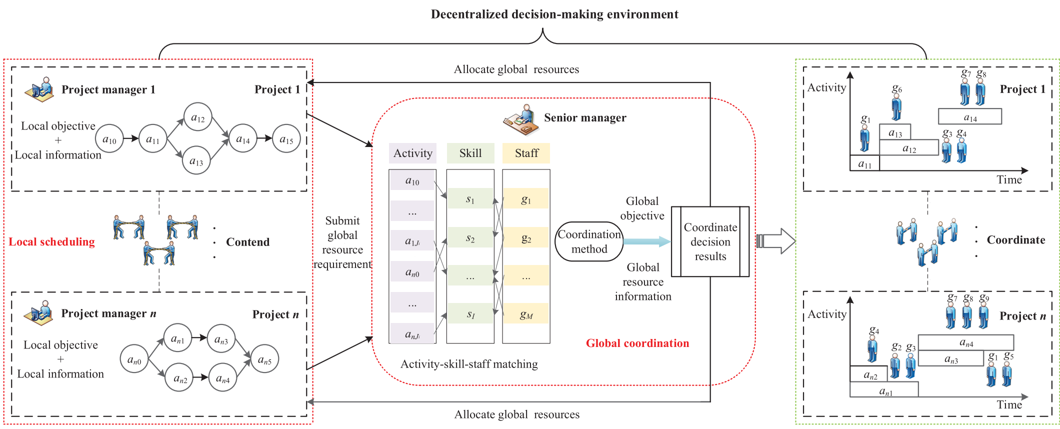

FIGURE 1. The scheduling framework of the DRCMPSP-MS.

$x_{i(J_i+1)t}$ equals 1 if dummy finish activity $a_{i(J_i+1)}$ ends at time t and equals 0 otherwise. Constraint (2) ensures that each activity is non-preempt, i.e., each activity is assigned exactly one end time. Constraint (3) represents the local resource constraint to ensure that at any time point t, the local resource r required by all activities does not exceed the available capacity of resource r. Constraint (4) ensures that the finish-to-start precedence relationships are fulfilled, i.e., any activity cannot start until all its predecessor activities have been completed. Constraint (5) indicates that the start time of project i cannot be earlier than its release time ad_i, where x_{i0t} equals 1 if dummy start activity a_{i0} ends at time t, and equals 0 otherwise. Constraint (6) denotes the decision variable. The meaning of x_{ijt} is shown in Table 1.

2) GLOBAL COORDINATION DECISION-MAKING MODEL

The CA allocates global resources with the optimization objective of minimizing the multi-project total tardiness costs. The global model can be formulated in the following.

$$\min \sum_{i=1}^{n}\sum_{t=0}^{T} tc_i \cdot \left(t \cdot x_{i(J_i+1)t} - ad_i - CPD_i\right) \tag{7}$$

$$\text{s.t.} \sum_{i\in N}\sum_{j\in V_i}\sum_{s\in S} e_{ijt}^{gs} \le 1, \ \forall g \in G, \forall t \in [0,T] \tag{8}$$

$$\sum_{i\in N}\sum_{j\in V_i} y_{ijt} \cdot p_{ij}^{s} \le \sum_{i\in N}\sum_{j\in V_i}\sum_{g\in G}\left(b_{gs}\cdot\left(1-e_{ijt}^{gs}\right)\right), \quad \forall s \in S, \forall t \in [0,T] \tag{9}$$

$$\sum_{s\in S} z_{gst} \le 1, \ \forall g \in G, \forall t \in [0,T] \tag{10}$$

$$d_{ij} = \left\lceil \frac{p_{ij}^{s} \cdot \bar{d}_{ij}}{\sum_{g\in G} m_{gs} \cdot e_{ijt}^{gs} \cdot o_{ij}^{s}} \right\rceil, \quad \forall i \in N, \ \forall j \in V_i, \ \forall s \in S, \ \forall t \in [0,T] \tag{11}$$

$$x_{ijt}, q_{ij}^{gt}, y_{ijt}, z_{gst}, e_{ijt}^{gs} \in \{0,1\} \tag{12}$$

The global objective function (7) minimizes the multi-project total tardiness costs, where $\sum_{t=0}^{T}\left(t \cdot x_{i(J_i+1)t} - ad_i\right)$ denotes the actual makespan of project i, CPD_i denotes the critical path length of project i, tc_i represents the unit tardiness cost of project i, and $\sum_{t=0}^{T} tc_i \cdot \left(t \cdot x_{i(J_i+1)t} - ad_i - CPD_i\right)$ stands for the tardiness cost incurred as the delay penalty of project i. Constraint (8) denotes the skill usage constraint for multi-skilled staff, meaning that each multi-skilled staff can only use one skill to perform one activity at any time. Constraint (9) indicates the skill availability constraint. In other words, at any time, the skill requirement for all activities being performed cannot exceed the total amount of available global resources in providing skills at that time. $\sum_{i\in N}\sum_{j\in V_i} y_{ijt} \cdot p_{ij}^{s}$ represents the requirement for skill s from all activities being performed at time t. $\sum_{i\in N}\sum_{j\in V_i}\sum_{g\in G}\left(b_{gs}\cdot\left(1-e_{ijt}^{gs}\right)\right)$ denotes total amount of available global resources in providing skill s at time t. To ensure that each multi-skilled staff can only use one skill at any time, constraint (10) is implemented. Equation (11) calculates the actual duration d_{ij} of activity a_{ij}. The actual duration is affected by the skill level of assigned multi-skilled staff. The higher the skill level of the multi-skilled staff assigned to activity a_{ij}, the shorter the actual duration of activity a_{ij}. If the calculated result of actual duration is not an integer, round it up. Constraint (12) defines the decision variables. The specific meanings of these decision variables are shown in Table 1.

IV. TWO-LAYER APPROACH

This section proposes a two-layer approach to solve the DRCMPSP-MS. In the local scheduling layer (Section IV-A), each PA uses the bat algorithm based on forward-backward scheduling to generate local baseline schedules. In the global coordination decision-making layer (Section IV-B), beginning at the time $t = 0$, each PA determines the activities that start at the current time and require global resources based on the local baseline schedule and submits the global resource requirement information of the relevant activities to CA. The CA adopts the variable neighborhood tabu search algorithm with greedy assignment strategy to determine the global resource assignment scheme. According

TABLE 1. Notation and description.

	Notation	Description
Parameters	i	The index of projects, $i = 1, 2, \ldots, n$
	j	The index of activities, $j = 0, 1, \ldots, J_i, J_i + 1$
	a_{ij}	The jth activity of project i
	a_{i0}	The dummy start of project i
	$a_{i(J_i+1)}$	The dummy end of project i
	r	The index of local resources types, $r = 1, 2, \ldots, E$
	g	The index of global resources, $g = 1, 2, \ldots, M$
	s	The index of skills types, $s = 1, 2, \ldots, I$
	T	The whole planning horizon
	t	The time point, $t = 0, 1, \ldots, T$
	CPD_i	The critical path length of project i
	ad_i	The release time of project i, $ad_i \geq 0$
	$\bar{d}_{ij}$	The planned duration of activity a_{ij}
	d_{ij}	The actual duration of activity a_{ij}
	m_{gs}	The level of skill s mastered by global resource g
	p_{ij}^{s}	The skill requirement of activity a_{ij} for skill s
	w_{ij}^{r}	The local resource requirement of activity a_{ij} for local resource r
	W_{ir}	The availability of local resource r in project i
	W_s	The availability of skill s
	tc_i	The unit tardiness cost of project i
	o_{ij}^{s}	A given parameter indicates when skill s is required by activity a_{ij} that equals 1 and equals 0 otherwise
	b_{gs}	A given parameter indicates when global resource g masters skill s that equals 1 and equals 0 otherwise
Sets	N	The set of projects, $N = \{1, 2, .., n\}, \forall i \in N$
	V_i	The activity set of project i, $V_i = \{0, 1, \ldots, J_i, J_i + 1\}, \forall j \in V_i$
	R_i	The set of local resources of project i, $R_i = \{1, 2, \ldots, E\}, \forall r \in R_i$
	G	The set of global resources, $G = \{1, 2, \ldots, M\}, \forall g \in G$
	S	The set of skills, $S = \{1, 2, \ldots, I\}, \forall s \in S$
	F_{ij}	The predecessor activity set of activity a_{ij}
Decision variables	e_{ijt}^{gs}	Equals 1 if resource g with skill s performs activity a_{ij} at time t, and equals 0 otherwise
	x_{ijt}	Equals 1 if activity a_{ij} ends at time t, and equals 0 otherwise
	q_{ij}^{gt}	Equals 1 if resource g starts performing activity a_{ij} at time t, and equals 0 otherwise
	y_{ijt}	Equals 1 if activity a_{ij} is performed at time t, and equals 0 otherwise
	z_{gst}	Equals 1 if resource g uses the skill s at time t, and equals 0 otherwise

to the coordination results at the current time, the relevant PAs adjust their local baseline schedules. Time advances to the next time, and the iterative process continues until all activities are scheduled.

A. LOCAL SCHEDULING LAYER

The local scheduling problem does not consider global resource constraints and belongs to the RCPSP, which is known as NP-hard [5]. For small-scale problems, several exact algorithms can be used to solve them. However, for large-scale problems, exact algorithms usually cannot obtain optimal or even feasible solutions in polynomial time. In recent years, many heuristic and intelligent optimization algorithms have been developed to solve the RCPSP [1]. Inspired by bat predation, the bat algorithm (BA) was introduced by Yang [53]. The BA has been successfully applied in job shop scheduling [54], streaming feature selection [55], and other fields [56] due to its easy implementation and good optimization performance. In this paper, the BA is applied in the field of project scheduling and improved according to the characteristics of the research problem. In addition, the forward-backward scheduling algorithm [57] is used to optimize algorithm results further. Therefore, a bat algorithm based on forward-backward scheduling (BAFBS) is developed to solve the local scheduling problem.

In the BAFBS, the position of each bat represents a candidate solution (scheduling strategy). The position of prey is equivalent to the optimal solution. Bats use echolocation to change their flight positions to keep approaching the prey. The BAFBS mainly includes the following steps: population initialization, forward-backward scheduling, position update, and population update. The pseudocode of the BAFBS is described in Algorithm 1. Fig. 2 shows the flowchart of the BAFBS procedure.

1) POPULATION INITIALIZATION

The initial population is randomly generated, and the population size is POP. Each bat B_k $(k = 1, 2, \ldots, POP)$ in the population can be described by a five-tuple $B_k = (l_k, v_k, f_k, h_k, A_k)$. The details are as follows: (1) $J+2$ dimensions random key vector $l_k = \left(l_{k0}, \ldots, l_{kj}, \ldots, l_{k(J+1)}\right)$ $(l_{kj} \in [0, 1], j = 0, 1, \ldots, J + 1)$ represents the position of the bat B_k, J indicates the number of non-dummy activities in one project. The initial value of l_k is generated randomly. (2) $J + 2$ dimensions vector v_k denotes the velocity of the bat B_k, and the initial value of v_k is set to 0. (3) The pulse frequency of the bat B_k is represented by scalar f_k, $f_k \in [f_{min}, f_{max}]$, and the initial value of f_k adopts f_{max}. (4) Scalar h_k indicates the pulse rate of the bat B_k, $h_k \in [h_{min}, h_{max}]$, the initial value of h_k uses h_{min}. (5) Scalar A_k represents the ultrasonic loudness of bat B_k, $A_k \in [A_{min}, A_{max}]$, the initial value of A_k is A_{max}. We set $f_{min} = h_{min} = A_{min} = 0$, and the values of the remaining parameters $(f_{max}, h_{max}, A_{max})$ are determined by experiments (Section V-B1).

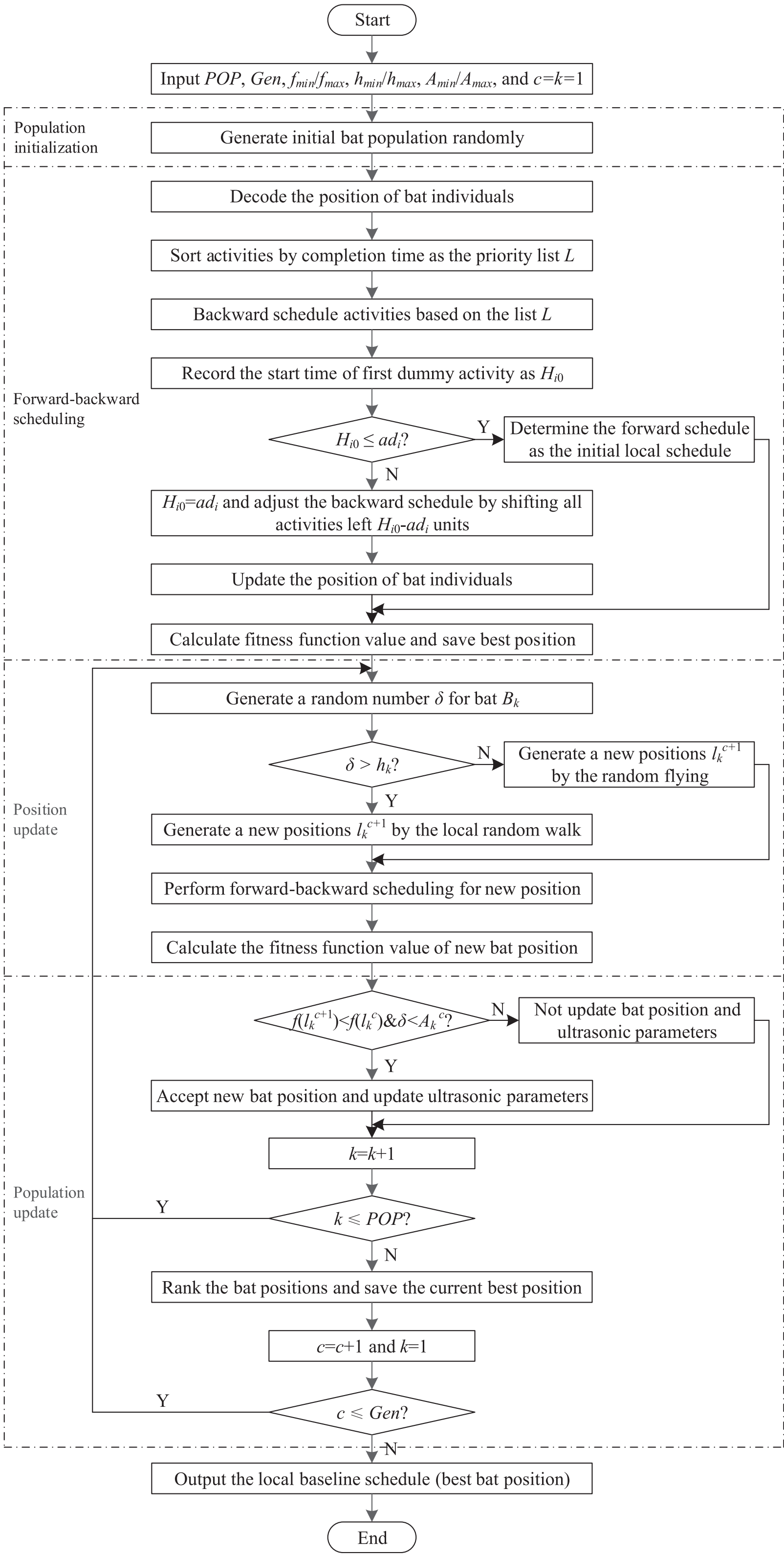

FIGURE 2. The flowchart of the BAFBS procedure.

2) POSITION UPDATE

There are two ways to update the position of bats: the local random walk and the random flying. Given a bat B_k, a random number $\delta \in [0, 1]$ is generated accordingly. Compare the random number δ with the pulse rate h_k, if $\delta > h_k$, the position of the bat B_k will be updated using the local random

Algorithm 1 Bat Algorithm Based on Forward-Backward Scheduling for PA

Input: population size (POP); total count (Gen); pulse frequency (f_{min}/f_{max}); pulse rate (h_{min}/h_{max}); ultrasonic loudness (A_{min}/A_{max}).
Output: local baseline schedule.

```
Generate initial bat population (POP individuals) randomly.
Decode the position of bat individuals.
Arrange activities in descending order of completion time as the priority list L.
Backward schedule activities according to the priority list L.
Record the start time of the first dummy activity as H_i0.
if H_i0 ≤ ad_i then
    Determine the forward schedule as the initial local schedule of project.
else
    H_i0 ← ad_i
    Adjust the backward schedule by shifting all the activities left H_i0 − ad_i units.
    Update the position of bat individuals.
end if
Calculate the fitness function value (local objective function (1)) of bat positions
and save the current best position.
Set the generation counter c ← 1; the total count Gen ← number of generations.
while c ≤ Gen do
    for each bat B_k (k = 1, 2, ..., POP) do
        Generate a random number δ.
        if δ > h_k then
            Generate a new positions l_k^{c+1} by the local random walk.
        else
            Generate a new positions l_k^{c+1} by the random flying.
        end if
        Perform forward-backward scheduling for the new bat position.
        Calculate the fitness function value of the new bat position.
        if f(l_k^{c+1}) < f(l_k^c) &δ < A_k^c then
            Accept the new bat position.
            Update ultrasonic parameters.
        end if
    end for
    Rank the bat positions and save the current best position.
    c = c + 1
end while
Return the local baseline schedule (best bat position).
```

walk. Otherwise, the position will be updated by the random flying.

In the random flying, the position is updated by using the following formulas:

$$f_k = f_{min} + (f_{max} - f_{min}) \cdot \delta \tag{13}$$

$$v_k^{c+1} = v_k^c + \left(l_k^c - l^*\right) \cdot f_k \tag{14}$$

$$l_k^{c+1} = l_k^c + v_k^{c+1} \tag{15}$$

The formula (13) determines the pulse frequency f_k of bat B_k based on the generated random number δ. Given the pulse frequency f_k, the formula (14) calculates the updated velocity v_k^{c+1} of bat B_k, where c ($c = 0, 1, \ldots, Gen$) denotes the number of iterations and l^* represents the current best bat position. v_k^c and l_k^c denote the velocity and position of bat B_k at the cth iteration, respectively. Based on the cth iteration position l_k^c and updated velocity v_k^{c+1}, the formula (15) determines the updated position l_k^{c+1} of bat B_k. If the value of any element in the newly generated l_k^{c+1} is outside the interval [0, 1], the element with a value less than 0 will be reset to 0, and the element with a value greater than 1 will be reset to 1. The mentioned rules are also applied to the local random walk.

In the local random walk, the position is updated by the following formula:

$$l_k^{c+1} = l^* + \varepsilon \cdot \bar{A}^c \tag{16}$$

The formula (16) implies that the updated position l_k^{c+1} of bat B_k is generated based on the current best bat position l^*, where $\varepsilon \in [-1, 1]$ denotes a random number and $\bar{A}^c$ represents the average loudness of all bats in the cth iteration.

3) POPULATION UPDATE

The previous position l_k^c can be replaced by the new position l_k^{c+1}, which needs to meet the following two conditions simultaneously: $f\left(l_k^{c+1}\right) < f\left(l_k^c\right)$ and $\delta < A_k^c$, where $f\left(l_k^{c+1}\right)$ and $f\left(l_k^c\right)$ denote the fitness function values of l_k^{c+1} and l_k^c calculated according to formula (1), respectively.

If the new position of the bat B_k is accepted, the corresponding ultrasonic parameters are updated according to the following formulas:

$$h_k^{c+1} = h_{max} \cdot [1 - exp(-\chi \cdot c)] \tag{17}$$

$$A_k^{c+1} = \beta \cdot A_k^c \tag{18}$$

The formula (17) determines the updated pulse rate h_k^{c+1} of bat B_k, where h_{max} represents the maximum value of the pulse rate, exp represents the exponential function operations, and c represents the number of iterations. The formula (18) calculates the updated ultrasonic loudness A_k^{c+1} of bat B_k, where A_k^c denotes the ultrasonic loudness of bat B_k at the cth iteration. χ and β are the adaptation parameters, which are set to $\chi = \beta = 0.9$ [53].

B. GLOBAL COORDINATION DECISION-MAKING LAYER

The global coordination decision-making layer mainly addresses the global resources assignment. When multiple activities require global resources simultaneously, the CA needs to determine an activity sequence that indicates which activity is assigned resources in priority. Moreover, considering that global resources master multiple skills and skill levels are heterogeneous, the CA needs to further specify the priority order of resource assignment. Tabu search algorithm (TS) was first developed by F. Glover based on artificial intelligence systems [58]. TS is an iterative search algorithm that searches the solution space by performing the neighborhood search on the current solution. Due to its strong exploring ability, TS has been widely applied in many areas such as project scheduling [59], job shop scheduling [60], and incremental graph drawing [61]. In this subsection, a variable neighborhood tabu search algorithm (VNTS) is designed to select the execution order of activities with resource conflicts. Embedding variable neighborhood search into TS can fully utilize the search advantages of both to find better solutions. In addition, the greedy assignment strategy (GAS) is adopted to assign global resources. The VNTS and the GAS are combined to solve the assignment problem at each decision point. The pseudocode of the VNTS-GAS is shown in Algorithm 2. Fig. 3 shows the flowchart of the VNTS-GAS procedure. The coordination process is mainly divided into the following three steps.

Step 1: Identify the decision point.

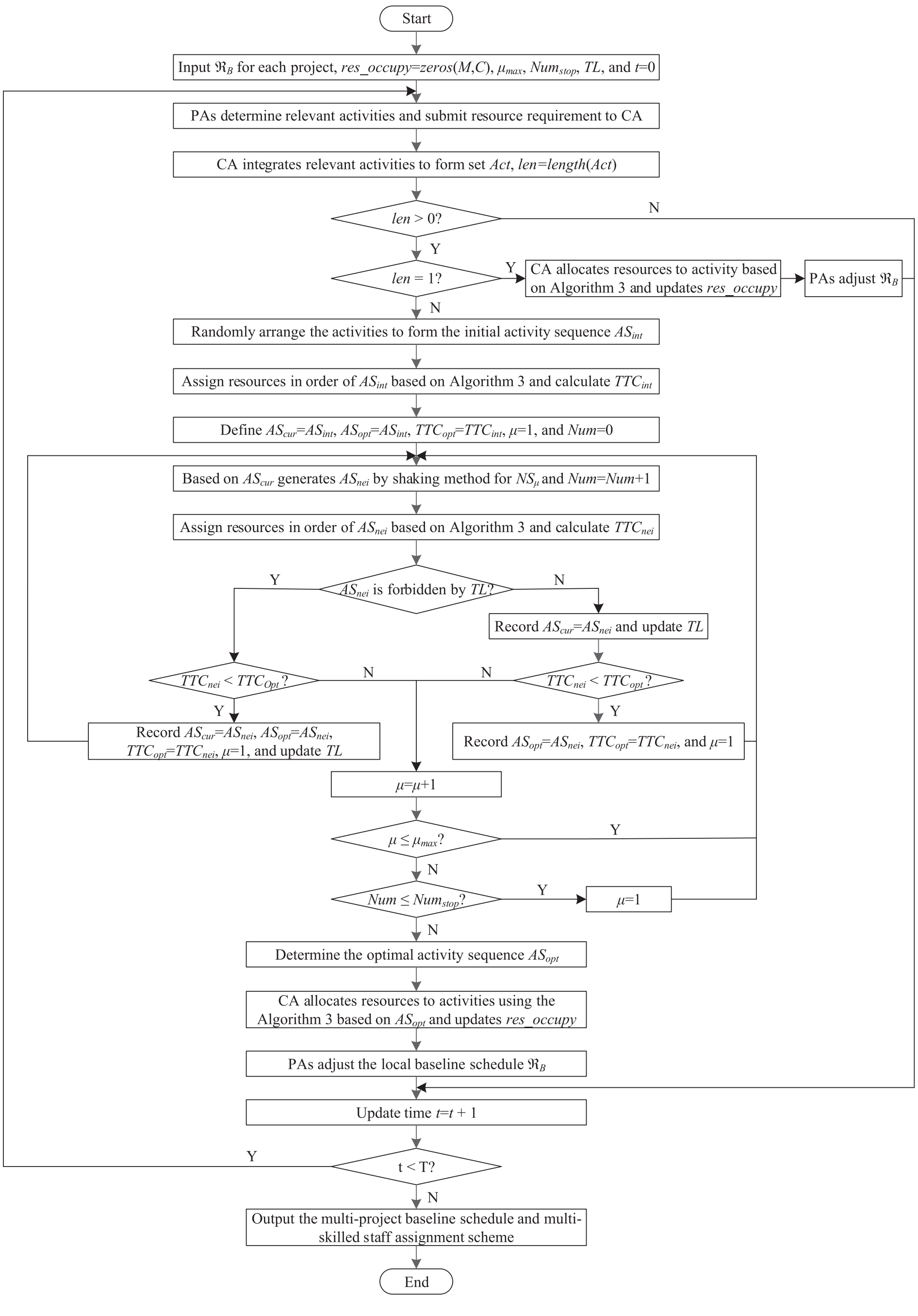

FIGURE 3. The flowchart of the VNTS-GAS procedure.

After all PAs complete the local scheduling according to Algorithm 1, they submit global resource requirement information to the CA based on the generated local baseline schedules. The resource requirement information includes the type and number of skills required for activities and which activities start at the current time. Starting from the time $t = 0$, the decision point is defined as the time when activities require global resources.

Step 2: The CA assigns global resources.

At the decision point, if only one activity requires global resources, the CA assigns global resources to the activity according to the greedy assignment strategy (Algorithm 3). When two or more activities compete for global resources, these activities are called conflicting activities. The CA performs the variable neighborhood tabu search for conflicting activities and selects the optimal conflicting activity sequence at each decision point. Moreover, the CA assigns global resources to conflicting activities using the Algorithm 3. The resource assignment results are fed back to the PAs.

Step 3: The PAs adjust the local baseline schedules.

Firstly, the PAs update the start time of conflicting activities. Secondly, the PAs adjust the original local schedule

Algorithm 2 Variable Neighborhood Tabu Search Algorithm With Greedy Assignment Strategy for CA

Input: local baseline schedule $\Re_B$ for each project; multi-skilled staff occupancy matrix $res\_occupy = zeros(M, C)$, M is the number of multi-skilled staff, C is the number of non-dummy activities in multi-project; number of neighborhood structure μ_{max}; number of activity sequence Num_{stop}; tabu list TL; initialize $t = 0$.
Output: multi-project baseline schedule and multi-skilled staff assignment scheme.

1: **while** $t < T$ **do**
2: PAs determine the activities that start at time t and require global resources based on the $\Re_B$, and submit global resource requirement to CA.
3: CA integrates all related activities to form activity set Act, $len = length\,(Act)$.
4: **if** $len > 0$ **then**
5: **if** $len == 1$ **then**
6: CA allocates global resources to the activity according to the greedy assignment strategy (Algorithm 3), and updates the res_occupy.
7: PAs adjusts the local baseline schedule $\Re_B$.
8: **else**
9: Randomly arrange the activities in set Act to form the initial activity sequence AS_{int}.
10: Assign global resources in order of AS_{int} according to Algorithm 3, and calculate the total tardiness costs, noted as TTC_{int}.
11: Define current activity sequence $AS_{cur} = AS_{int}$, Optimal activity sequence $AS_{opt} = AS_{int}$, Optimal tardiness costs value $TTC_{opt} = TTC_{int}$, $\mu = 1$, and $Num = 0$.
12: Based on AS_{cur} generates a neighborhood activity sequence AS_{nei} by the shaking method for NS_μ, and $Num = Num + 1$.
13: Assign global resources in order of AS_{nei} according to Algorithm 3, and calculate the total tardiness costs, noted as TTC_{nei}.
14: **if** AS_{nei} is forbidden by the tabu list TL **then**
15: **if** $TTC_{nei} < TTC_{opt}$ **then**
16: Record $AS_{cur} = AS_{nei}$, $AS_{opt} = AS_{nei}$, $TTC_{opt} = TTC_{nei}$, $\mu = 1$, and update TL.
17: Return to step 12.
18: **else**
19: $\mu = \mu + 1$
20: **if** $\mu \leq \mu_{max}$ **then**
21: Return to step 12.
22: **else**
23: **if** $Num \leq Num_{stop}$ **then**
24: $\mu = 1$
25: Return to step 12.
26: **end if**
27: **end if**
28: **end if**
29: **else**
30: Record $AS_{cur} = AS_{nei}$ and update TL.
31: **if** $TTC_{nei} < TTC_{opt}$ **then**
32: Record $AS_{opt} = AS_{nei}$, $TTC_{opt} = TTC_{nei}$, and $\mu = 1$.
33: Return to step 12.
34: **else**
35: Return steps 19.
36: **end if**
37: **end if**
38: Determine the optimal activity sequence AS_{opt}.
39: CA allocates global resources to activities using the Algorithm 3 according to AS_{opt}, and updates the res_occupy.
40: PAs adjusts the local baseline schedule $\Re_B$.
41: **end if**
42: **end if**
43: Update time $t = t + 1$.
44: **end while**

for the activities whose start time is later than the decision time by using the translation strategy, i.e. the start time of activities is moved to the right to satisfy the priority relationship and resource availability constraints. Finally, the PAs provide new global resource requirement information to the CA based on the adjusted local baseline schedules. Continue looping the above process until all activities are scheduled.

1) VARIABLE NEIGHBORHOOD TABU SEARCH

The following features are involved in the VNTS:

a: INITIAL SOLUTION

Each solution is represented by a sequence of conflicting activities in the VNTS. The activity sequence is randomly generated in the initial solution AS_{int}.

b: NEIGHBOURHOOD STRUCTURE

The CA uses the VNTS, which combines deterministic and stochastic changes to the neighborhood, to search for desirable solutions. Considering the representation of solutions, the neighborhood structure NS_μ ($\mu = 1, 2, \ldots, \mu_{max}$) is defined as the set of all neighborhood activity sequences AS_{nei} in which only μ pairs of elements are different from the corresponding elements in the current activity sequence AS_{cur}.

c: SHAKING METHOD

Based on the above definition of the neighborhood structure, the shaking method is presented as follows: for the current activity sequence, select μ pairs of activities randomly and swap their positions, thus obtaining a new neighborhood activity sequence.

d: TABU LIST

The tabu list TL is managed according to the first-in-first-out (FIFO) rule [62]. Whenever a neighboring solution is obtained, the corresponding reverse move is stored in the TL, and the oldest existing move is deleted. Normally, all the moves in the TL are forbidden. However, if the solution generated by a tabu move is better than the best solution obtained so far, then its tabu status may be canceled, and the move will be accepted (aspiration criterion). Reference [63], the size of tabu list is defined as $\sqrt{\Omega}$, where Ω denotes the number of conflicting activities at each decision point.

e: STOPPING CRITERION

The stopping criterion of the VNTS is defined as an assumed number of activity sequences visited, denoted as Num_{stop}. We describe the optimal activity sequence found and the number of activity sequences visited during the searching process as AS_{opt} and Num, respectively.

2) GREEDY ASSIGNMENT STRATEGY

The global resources considered in this paper have the characteristic of multi-skilled heterogeneity, i.e., each resource masters multiple skills and has different skill levels. Therefore, when assigning global resources to activities, the CA needs to design relevant assignment strategies that take into account the characteristics of multi-skilled resources. The GAS is designed based on the skill level and the number of skills mastered by global resources, and each allocation is locally optimal. This strategy indicates that resources with high skill levels are assigned priority. When skill levels are the same, priority should be given to assigning resources with the lowest skill numbers, and the minimum resource serial number is used to break the tie.

Algorithm 3 Greedy Assignment Strategy

Input: set of activities requiring global resources at time t, noted as Act^t; multi-skilled staff occupancy matrix at time t, noted as $res\_occupy\,(t)_{temp}$.
Output: global resource assignment scheme at time t.

Determine the sequence of conflicting activities AS^t
Based on $res\_occupy\,(t)_{temp}$, record the resource-skill set RS and the available of skill AOS.
for $\alpha = 1 : length\left(AS^t\right)$ **do**
  $a_{ij} = AS^t\,(\alpha)$
  if $p_{ij}^s \leq AOS\,(s)$ **then**
    Global resources that master skill s are selected in RS to form resource set RS_{temp}.
    Record the skill level $\left(m_{gs}\right)$ and number of skills $\left(ns_g\right)$ mastered by each global resource g in RS_{temp}.
    Sort the resources in RS_{temp} in decreasing order of m_{gs}, break the tie with ns_g ascending order.
    Select the first p_{ij}^s resources to form resource set RS_{fi}, and assign them to a_{ij}.
    Calculate duration d_{ij} based on formula (11).
    Record $res\_occupy\,(t)_{temp}$.
    Update $RS = RS \setminus RS_{fi}$, $AOS\,(s) = AOS\,(s) - p_{ij}^s$.
  else
    $Act^{t+1} = Act^{t+1} \bigcup AS^t\,(\alpha)$
  end if
end for

a: HIGHEST SKILL LEVEL FIRST

This priority rule assigns resources based on the skill level. Global resources are sorted based on the level of skills mastered. Resources with the highest skill level will receive the highest priority. This rule aims to prioritize the allocation of the most efficient resources, effectively reducing the actual duration of activities and the total project completion time.

b: LOWEST SKILL NUMBERS FIRST

This rule uses the number of mastered skills to allocate global resources. Each global resource can only use one skill to perform an activity. Prioritize the allocation of resources with fewer skills, and resources with more skills are more available to be allocated to other activities. Therefore, this rule increases the usage of resources and improves the flexibility of assignments.

When a conflicting activity sequence is determined, the CA assigns global resources to activities based on the GAS mentioned above. Calculate the actual duration d_{ij} of activities, and update the multi-skilled staff occupancy matrix $res\_occupy\,(t)_{temp}$, the resource-skill set RS and the available of skill AOS. The pseudocode of the GAS is described in Algorithm 3. Fig. 4 shows the flowchart of the GAS procedure.

V. COMPUTATIONAL EXPERIMENTS

A series of computational experiments are conducted to study the performance of proposed approach. All program codes are written using Matlab R2018b, such as the BAFBS program, the VNTS-GAS program, reading programs for instances, etc. All experiments are performed on an Intel Core i7-8700K processor computer with 3.70 GHz clock speed and 16 GB RAM. Section V-A presents a experimental design. Section V-B discusses the parameter settings of the BAFBS and evaluates its performance. Section V-C analyzes the impact of problem size and skill utilization factor on the global coordination results. Section V-D verifies the performance of the proposed TLA by comparing it with distributed and centralized methods.

A. EXPERIMENTAL DESIGN

The proposed approach is tested on 60 problem instances taken from the Multi-Project Scheduling Problem LIBrary (MPSPLIB) (http://www.mpsplib.com). These instances are classified into 12 problem subsets. Each problem subset is named "MPJ_i_n", where J_i is the number of non-dummy activities in each project and n is the number of projects in each instance. Specifically, the number of projects includes 2, 5, 10, and 20, where every single project consists of 30, 90, or 120 activities. Each problem subset contains 5 instances. Further information required to generate instances are:

- Each problem instance is provided with four types of resources, including three types of local resources and one type of global resource.
- The total skill demand types with activity numbers of 30, 90, and 120 are set to 3, 5, and 7, respectively.
- The types of skills mastered by each multi-skilled staff are generated in the range of [2,3] uniformly.
- The skill requirements p_{ij}^s of each activity are generated in the range of [1,3] uniformly
- The level of each skill mastered by multi-skilled staff is randomly generated in 0.6, 0.8, and 1, respectively.

The problem subsets and parameters are shown in Table 2. Problem size is the total number of activities in each instance. The average skill utilization factor ($ASUF$) is the average of the skill utilization factor (SUF) of 5 instances in each problem subset. SUF is calculated as formula (19). It denotes the maximum degree of restriction on each skill requirement. The larger the SUF, the scarcer the global resources with relevant skills and the stronger the resource conflicts among projects [16].

$$SUF = \max_{s \in S} SUF_s,\ SUF_s = \frac{\sum_{i=1}^{n} \sum_{j=1}^{J_i} P_{ij}^s}{W_s \times GCPD} \tag{19}$$

where SUF_s denotes the utilization factor of skill s. $GCPD$ represents the global critical path length without considering resource constraints. P_{ij}^s indicates the skill requirement of activity a_{ij} for skill s within the whole duration.

B. ANALYSIS OF LOCAL SCHEDULING RESULTS

1) PARAMETER SETTINGS OF BAFBS

This subsection uses the Taguchi's design of experiment (DOE) technique [64] to determine the parameter settings of BAFBS. The MP30_20 problem subset is used to carry out the test. This subset contains 100 single project. The BAFBS involves four key parameters: the population size (POP), the maximum pulse frequency (f_{max}), the maximum pulse rate (h_{max}), and the maximum ultrasonic loudness (A_{max}). We have set 5 different level values for each parameter, as shown in Table 3.

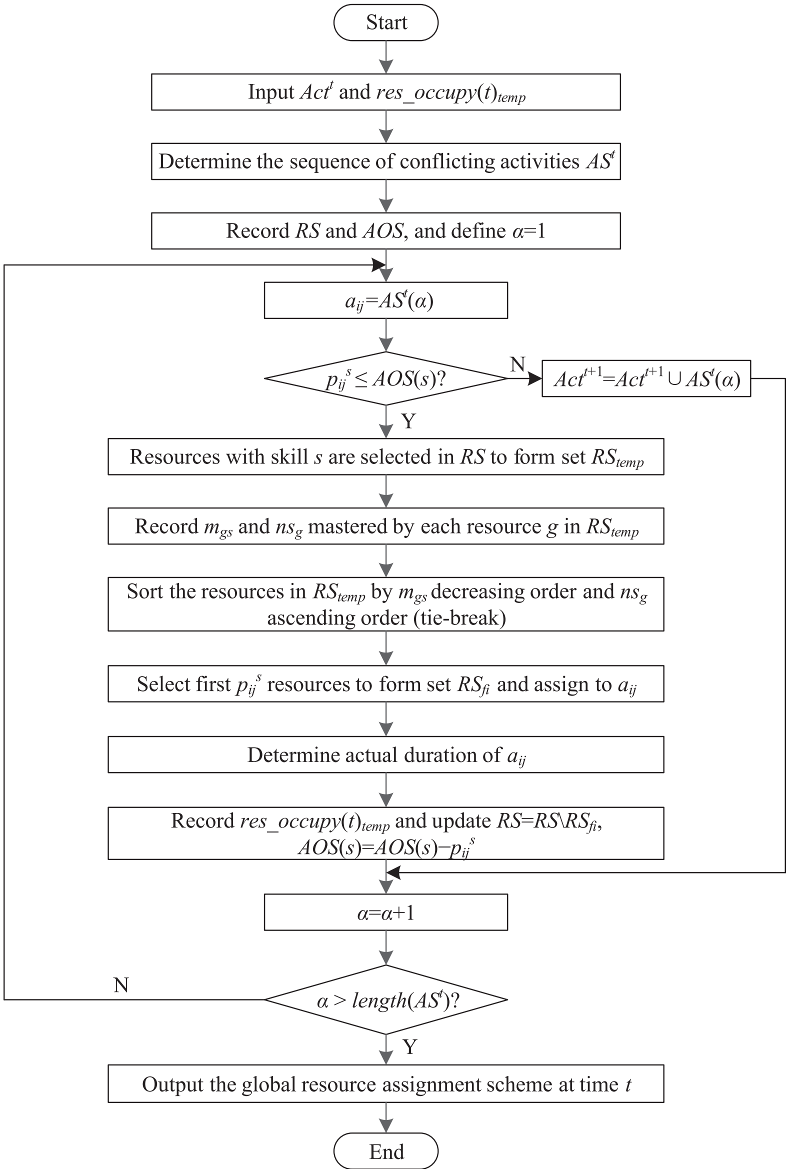

FIGURE 4. The flowchart of the GAS procedure.

TABLE 2. The problem subsets and parameters.

Problem subsets	Number of instances	Number of projects	Number of activities	Problem size	*ASUF*
MP30_2	5	2	30	60	0.84
MP90_2	5	2	90	180	0.79
MP120_2	5	2	120	240	1.02
MP30_5	5	5	30	150	0.91
MP90_5	5	5	90	450	1.07
MP120_5	5	5	120	600	1.29
MP30_10	5	10	30	300	1.40
MP90_10	5	10	90	900	1.02
MP120_10	5	10	120	1200	1.42
MP30_20	5	20	30	600	0.99
MP90_20	5	20	90	1800	0.85
MP120_20	5	20	120	2400	1.05

Firstly, average response variable (ARV) is calculated by the formula (20).

$$ARV = \frac{\sum_{u=1}^{num} \left[(obj_u - opt_u) \big/ opt_u \right]}{num} \tag{20}$$

where num represents the total number of project, i.e., $num = 100$. obj_u and opt_u are the local objective value (minimizing the project completion time) corresponding to the uth project obtained by the BAFBS and the exact branch-and-bound (B&B) algorithm, respectively.

TABLE 3. Parameter levels.

Parameter levels	Parameters			
	POP	f_{max}	h_{max}	A_{max}
1	30	0.00001	0.1	0.1
2	60	0.0001	0.3	0.3
3	90	0.001	0.5	0.5
4	120	0.01	0.7	0.7
5	150	0.1	0.9	0.9

TABLE 4. Response table for ***ARV*** and rank for each parameter.

Levels	POP	f_{max}	h_{max}	A_{max}
1	0.010804939	0.00873643	0.008389639	0.00858522
2	0.008273216	0.007684334	0.007874996	0.008731633
3	0.007129625	**0.007580789**	0.00768346	0.007978499
4	0.007000679	0.007699153	**0.007668276**	0.007618427
5	**0.006171729**	0.007679481	0.007763816	**0.006466409**
Range	0.00463321	0.001155641	0.000721362	0.002265224
Rank	1	3	4	2

Next, we conduct the DOE test on the scale of $L_{25}\left(5^4\right)$, i.e., we have a total of 25 treatments and 5 levels for each of the 4 parameters. The ARV values are obtained by running the BAFBS with various combinations of the parameter settings given in Table 3. The termination criterion of BAFBS is to reach the maximum number of generations $Gen = 100$. The response table for ARV and rank for each parameter are presented in Table 4. The second to the sixth row of Table 4 shows the average value of the ARV for a parameter with different levels, and the best parameter setting is bold. The row 'Range' represents the range of the average ARV of each parameter. The significance rank of each parameter is shown on the row 'Rank'.

It can be seen from Table 4 that POP has the largest impact on the ARV and h_{max} has the least significant impact. We select the level that results in the smallest ARV value for each parameter. Therefore, the parameters for the BAFBS are set as $POP = 150$ (level 5), $f_{max} = 0.001$ (level 3), $h_{max} = 0.7$ (level 4), $A_{max} = 0.9$ (level 5), and $Gen = 100$.

2) EFFECTIVENESS ANALYSIS OF BAFBS

We compare the project completion time of each local schedule obtained by the BAFBS with the results obtained by the B&B algorithm embedded in RESCON software [65] (all solutions obtained by the exact B&B algorithm are optimal results and if RESCON runs out of memory, no solutions can be found). Both of them are performed on 555 single projects from the 12 problem subsets. The BAFBS is only compared with the B&B algorithm for the single project that can be solved exactly by the B&B algorithm. Table 5 shows the comparison results of the two algorithms. The table includes the average project completion time ($APCT$), the number of projects with solvable solutions ($NPSS$), and the number of projects with optimal solutions ($NPOS$), respectively. The second column shows the total number of projects (TNP) in each problem subset. The last column represents the average relative deviation (ARD) of the solutions solved by the BAFBS and the optimal solutions for each subset obtained by the B&B algorithm.

The results show that out of the 555 single projects, the B&B algorithm can obtain optimal solutions for 352 projects and 89.77% of which are also obtained by the BAFBS. Moreover, the more activities involved in a project, the less likely it is for the B&B algorithm to obtain a feasible solution before running out of memory, especially for projects with 120 activities. However, the BAFBS can find a feasible solution efficiently for any size project. In addition, for each problem subset, the results for the $APCT$ obtained by BAFBS and B&B are very close. The maximum average relative deviation is 2.15% (MP30_2), while the others are less than 1%. The average relative deviation of all 12 problem subsets is 0.42%. It fully turned out that the BAFBS performs well on local scheduling problems and can provide high-quality solutions (satisfactory local baseline schedules).

C. ANALYSIS OF GLOBAL COORDINATION RESULTS

The global coordination results for each problem subset are evaluated using the multi-project total tardiness cost (TTC), as presented in Table 6. Considering the CPU running time, the result is the average of 10 runs of each instance. Moreover, similar to the pre-experiment in Section V-B1, the parameters of the VNTS-GAS are configured as follows: number of neighborhood structure $\mu_{max} = 3$, number of activity sequence $Num_{stop} = 100$. Subsequently, the impact of problem size and $ASUF$ on global coordination results is analyzed.

The problem size is expressed as multiplying the number of projects and activities in each instance. Fig. 5 shows the bubble chart of TTC changing with problem size. It can be seen from Fig. 5 that when the number of projects is fixed, the more activities of a single project, the greater the TTC value. Similarly, in the case of a certain number of activities, the more the number of projects, the higher the TTC value. All problem subsets are divided into two types according to $ASUF< 1$ and $ASUF> 1$. $ASUF>1$ indicates a relatively strong resource conflict, while the opposite indicates a weak one. The comparison bar chart of the average TTC for two types of problem subsets under different $ASUF$ is presented in Fig. 6. As revealed in Fig. 6, the stronger the resource conflict degree, the higher the multi-project total tardiness cost. In addition, Fig. 7 illustrates bubble chart of TTC under different problem size and $ASUF$. The size of the bubbles is proportional to the TTC value. Comparing Area A (with smaller problem sizes and $ASUF$) and Area B (with larger problem sizes and $ASUF$), the results show that the larger the problem size and the stronger the resource conflicts degree, the higher the TTC value.

To sum up, the problem size and resource conflicts degree have an impact on TTC. Managers can adjust the total number of resources appropriately to prevent excessive resource conflicts from delaying the project for too long, thereby increasing the cost of the delay.

D. PERFORMANCE OF TWO-LAYER APPROACH

So far, only one paper [16] proposed a two-stage approach with softmax scoring mechanism (TSA-SSM) to solve the

TABLE 5. Comparison of results obtained by B&B and BAFBS.

Problem subsets	TNP	B&B $APCT_1$	B&B $NPSS_1$	B&B $NPOS_1$	BAFBS $APCT_2$	BAFBS $NPSS_2$	BAFBS $NPOS_2$	ARD (%)
MP30_2	10	55.8	10	10	57	10	5	2.15
MP90_2	10	93.5	8	8	93.5	10	8	0
MP120_2	10	98	1	1	98	10	1	0
MP30_5	25	59.8	25	25	60.2	25	18	0.67
MP90_5	25	89.1	20	20	89.1	25	20	0
MP120_5	25	109	2	2	109	25	2	0
MP30_10	50	52.9	50	50	53.4	50	41	0.95
MP90_10	50	88.36	36	36	88.4	50	36	0.05
MP120_10	50	106.5	10	10	107	50	10	0.47
MP30_20	100	57.82	100	100	58.2	100	87	0.66
MP90_20	100	92.84	85	85	92.9	100	83	0.06
MP120_20	100	109	5	5	109	100	5	0
Total	555	84.39*	352	352	84.64*	555	316	0.42*

* Represents the average of all problem subsets.

TABLE 6. Global coordination results.

Problem subsets	Problem size	$ASUF$	TTC
MP30_2	60	0.84	2762.2
MP90_2	180	0.79	7979
MP120_2	240	1.02	21794.6
MP30_5	150	0.91	5708.6
MP90_5	450	1.07	40705.6
MP120_5	600	1.29	42810.8
MP30_10	300	1.40	14567
MP90_10	900	1.02	50240.4
MP120_10	1200	1.42	98254.6
MP30_20	600	0.99	28926.4
MP90_20	1800	0.85	79168
MP120_20	2400	1.05	160373.8

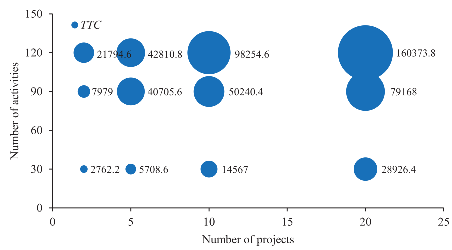

FIGURE 5. Bubble chart of *TTC* changing with problem size.

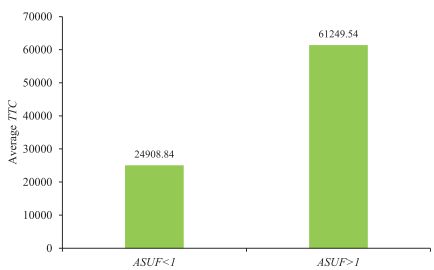

FIGURE 6. Average *TTC* under different *ASUF*.

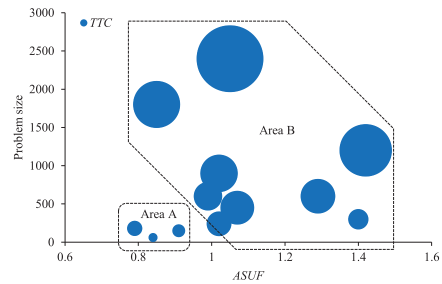

FIGURE 7. Bubble chart of *TTC* under different problem size and *ASUF*.

DRCMPSP-MS. In this subsection, we compare the TLA with TSA-SSM to verify the performance of the proposed approach. In addition, to further evaluate the effectiveness of the TLA, we further compared it with two centralized methods (i.e., non-two-stage methods): PSGSMINSLK and BRKGA. The centralized method is to integrate all the activities of multiple projects into a super- or meta-network, and then a unified decision-maker with complete information on multiple projects makes a scheduling plan and allocates resources. The algorithms used for comparison are introduced as follows.

1) TSA-SSM

A two-stage approach with softmax scoring mechanism is introduced by Yu et al. [16]. TSA-SSM belongs to a decentralized method (i.e., a two-stage method) that includes local scheduling stage and global coordination decision stage. According to the local scheduling plan obtained by the genetic algorithm, the softmax scoring mechanism is presented to resolve global resource conflicts.

2) PSGSMINSLK

A heuristic approach PSGSMINSLK is introduced by Villafáñez et al. [66]. PSGSMINSLK belongs to a centralized method, i.e., a non-two-stage method. The main reason for comparing algorithms is that the PSGSMINSLK performs well, and this algorithm outperforms other algorithms published in the multi-project scheduling problem library in 16% of the cases and holds the best result in 27% of the cases. The PSGSMINSLK combines the priority rule minimum total slack (MIN-SLK) and the parallel schedule generation scheme (P-SGS) and adopts the same strategy for assigning multi-skilled resources as in this paper.

TABLE 7. Comparison results of TLA and other algorithms.

Problem subsets	TLA		TSA-SSM		PSGSMINSLK		BRKGA		
	TTC_1	$CPU_1(s)$	TTC_2	$CPU_2(s)$	TTC_3	$CPU_3(s)$	TTC_4	$CPU_4(s)$	$Gap(\%)$
MP30_2	2762.2	29.96	2998.4	15.6	3004.8	0.21	**2248.6**	45.49	22.84
MP90_2	7979	75.35	9177.8	43.36	9282.2	0.29	**6929.2**	102.4	15.15
MP120_2	**21794.6**	119.98	24912.4	65.97	24429.2	0.61	24256.8	159.7	0
MP30_5	5708.6	69.39	5415.8	36.56	6406.8	0.29	**4802.2**	95.84	18.87
MP90_5	40705.6	189.95	**39105.4**	102.3	45401.4	0.75	56160	249.27	4.09
MP120_5	**42810.8**	290.97	44067.2	181.36	53849	1.75	69112.6	370.99	0
MP30_10	14567	137.89	**12008**	72.47	16269.6	0.56	13626	183.43	21.31
MP90_10	**50240.4**	406.04	53094.4	233.12	77087.8	1.91	106793.2	520.6	0
MP120_10	**98254.6**	786.44	114009.4	484.39	137325.8	4.02	171791.8	992.23	0
MP30_20	**28926.4**	292.75	29963.4	146.76	45526.2	1.16	78134.4	383.43	0
MP90_20	**79168**	1085.74	84601	482.01	112460.2	4.24	228195	1420.49	0
MP120_20	**160373.8**	1882.58	170741.4	963.22	223151	5.42	225123.8	2421.33	0

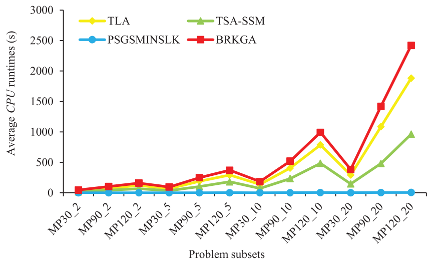

FIGURE 8. Comparison of *CPU* runtimes for different algorithms.

3) BRKGA

A biased random-key genetic algorithm is developed by Almeida et al. [67]. BRKGA also belongs to a centralized method, i.e., a non-two-stage method. BRKGA is more used for the project scheduling problem with flexible resources and proved to have good performance. In this algorithm, a population of chromosomes evolves over a number of generations until the defined stopping criteria are met. The relevant parameters are configured as follows: population size = 100, number of generations = 100, crossover rate = 0.8, mutation rate = 0.1.

Table 7 shows the comparison results of different algorithms, including the *TTC* and *CPU* runtimes. The results in the table denote the average of the 10 runs of each algorithm, and the best-found result of each problem subset is marked in bold. The *Gap* indicate the relative deviations between the results of *TLA* and the best-found result. The calculation formula is shown in equation (21), where *Best* is the best-found result of each problem subset. Fig. 8 shows a line chart comparing the *CPU* runtimes of different algorithms, where the runtimes represent the average time of each instance in each problem subset.

$$Gap = \frac{TLA - Best}{Best} \times 100\% \tag{21}$$

It can be seen from Table 7 that the TLA obtained the best-found results in 7 out of 12 problem subsets. Compared to other algorithms, the solution results of the TLA are slightly worse on specific problem subsets (e.g., MP30_2, MP90_2). However, the TLA can get better results on most problem subsets, and the average relative deviation of all problem subsets is 6.86%, which reflects the effectiveness of the TLA. In terms of *CPU* runtimes, Table 7 and Fig. 8 show that although the PSGSMINSLK has the shortest runtimes, it has poor performance compared with the other algorithms. The TLA takes slightly more runtime than the TSA-SSM, but less than the BRKGA. The TLA can update the found-best results on most problem subsets within a reasonable runtime. Therefore, when managers encounter such problems in practice, they can use our approach to make quick and effective decisions

VI. CONCLUSION AND FUTURE RESEARCH

This paper studies the DRCMPSP-MS. We formulate this problem as a two-layer model based on the MAS and propose the TLA to solve this problem. In the TLA, the BAFBS is introduced to deal with the local scheduling problem in layer one. The VNTS-GAS is developed to resolve global resource conflicts in layer two. Different size instances are solved to examine the performance of the proposed TLA. Computational results show that the BAFBS performs well on the local scheduling problem. High-quality solutions can be obtained by the BAFBS for all problem instances. It is further verified that the problem size and resource conflicts degree have an impact on global coordination results. In addition, compared with a decentralized method (TSA-SSM) and two centralized methods (PSGSMINSLK and BRKGA), the proposed TLA can obtain lower total tardiness cost on most problem subsets. This proves that TLA is suitable for solving DRCMPSP-MS and can effectively allocate multi-skilled staff shared among multiple projects.

The shortcomings of this paper and possible future research directions are described as follows. One shortcoming is that the initial solutions of both BAFBS and VNTS-GAS are randomly generated, and the quality of the initial solutions has a certain impact on the performance of the algorithms. Therefore, more effective methods for generating initial solutions can be designed in the future to further improve the approach proposed in this paper. Another shortcoming is that TLA involves some algorithm parameters, such as population size, number of neighborhood structures, etc. The setting of these parameters also affects the solving performance of the algorithm. Therefore, in future research, more efficient and fewer parameters new approaches can be developed to solve the DRCMPSP-MS. Moreover, the DRCMPSP-MS studied in this paper only considers the deterministic environment. However, resource availability may change in practice due to staff leave, resignation, or recruitment. Therefore, it will be a promising research direction further to consider the staff availability uncertainty in the DRCMPSP-MS. In addition, this paper assumes that the skill level of staff remains unchanged during project execution. Since the skill level of staff tends to be a very dynamic concept, it can also be a promising addition to the DRCMPSP-MS to incorporate an adjustable skill level that is affected by the learning or forgetting effect.

ACKNOWLEDGMENT

The authors thank their team for excellent technical support and assisting in preparation of this manuscript.

REFERENCES

[1] S. Hartmann and D. Briskorn, "An updated survey of variants and extensions of the resource-constrained project scheduling problem," *Eur. J. Oper. Res.*, vol. 297, no. 1, pp. 1–14, Feb. 2022.

[2] J. Blazewicz, J. K. Lenstra, and A. H. G. R. Kan, "Scheduling subject to resource constraints: Classification and complexity," *Discrete Appl. Math.*, vol. 5, no. 1, pp. 11–24, Jan. 1983.

[3] H. M. H. Saad, R. K. Chakrabortty, S. Elsayed, and M. J. Ryan, "Quantum-inspired genetic algorithm for resource-constrained project-scheduling," *IEEE Access*, vol. 9, pp. 38488–38502, 2021.

[4] Z. Ma, M. Ning, and Y. Wang, "Proactive project scheduling with activity splitting and resource transfer times under uncertain environments," *IEEE Access*, vol. 10, pp. 87490–87499, 2022.

[5] M. Tian, R. J. Liu, and G. J. Zhang, "Solving the resource-constrained multi-project scheduling problem with an improved critical chain method," *J. Oper. Res. Soc.*, vol. 71, no. 8, pp. 1243–1258, Aug. 2020.

[6] L. Cui, X. Liu, S. Lu, and Z. Jia, "A variable neighborhood search approach for the resource-constrained multi-project collaborative scheduling problem," *Appl. Soft Comput.*, vol. 107, Aug. 2021, Art. no. 107480.

[7] N. He, D. Z. Zhang, and B. Yuce, "Integrated multi-project planning and scheduling—A multiagent approach," *Eur. J. Oper. Res.*, vol. 302, no. 2, pp. 688–699, Oct. 2022.

[8] W. Peng, D. Yu, and J. Lin, "Resource-constrained multi-project reactive scheduling problem with new project arrival," *IEEE Access*, vol. 11, pp. 64370–64382, 2023.

[9] F. Li, Z. Xu, and H. Li, "A multi-agent based cooperative approach to decentralized multi-project scheduling and resource allocation," *Comput. Ind. Eng.*, vol. 151, Jan. 2021, Art. no. 106961.

[10] D. Liu, Z. Xu, and F. Li, "A three-stage decomposition algorithm for decentralized multi-project scheduling under uncertainty," *Comput. Ind. Eng.*, vol. 160, Oct. 2021, Art. no. 107553.

[11] G. Confessore, S. Giordani, and S. Rismondo, "A market-based multi-agent system model for decentralized multi-project scheduling," *Ann. Oper. Res.*, vol. 150, no. 1, pp. 115–135, Feb. 2007.

[12] S. Adhau, M. L. Mittal, and A. Mittal, "A multi-agent system for distributed multi-project scheduling: An auction-based negotiation approach," *Eng. Appl. Artif. Intell.*, vol. 25, no. 8, pp. 1738–1751, Dec. 2012.

[13] W. Song, D. Kang, J. Zhang, and H. Xi, "A multi-unit combinatorial auction based approach for decentralized multi-project scheduling," *Auto. Agents Multi-Agent Syst.*, vol. 31, no. 6, pp. 1548–1577, Jun. 2017.

[14] S. Zhao and Z. Xu, "New closed-loop approximate dynamic programming for solving stochastic decentralized multi-project scheduling problem with resource transfers," *Exp. Syst. Appl.*, vol. 185, Dec. 2021, Art. no. 115593.

[15] D. Liu and Z. Xu, "A multi-PR heuristic for distributed multi-project scheduling with uncertain duration," *IEEE Access*, vol. 8, pp. 227780–227792, 2020.

[16] Y. Yu, Z. Xu, D. Liu, and S. Zhao, "A two-stage approach with softmax scoring mechanism for a multi-project scheduling problem sharing multi-skilled staff," *Exp. Syst. Appl.*, vol. 203, Oct. 2022, Art. no. 117385.

[17] Y. H. Lee, S. R. Kumara, and K. Chatterjee, "Multiagent based dynamic resource scheduling for distributed multiple projects using a market mechanism," *J. Intell. Manuf.*, vol. 14, no. 5, pp. 471–484, Oct. 2003.

[18] L. Liu, C. Wang, J.-J. Wang, and A. M. F. Crespo, "An iterative auction for resource-constrained surgical scheduling," *J. Oper. Res. Soc.*, vol. 74, no. 3, pp. 968–978, Mar. 2023.

[19] J. A. Araúzo, J. Pajares, and A. Lopez-Paredes, "Simulating the dynamic scheduling of project portfolios," *Simul. Model. Pract. Theory*, vol. 18, no. 10, pp. 1428–1441, Nov. 2010.

[20] J. S. K. Lau, G. Q. Huang, K. L. Mak, and L. Liang, "Distributed project scheduling with information sharing in supply chains: Part I—An agent-based negotiation model," *Int. J. Prod. Res.*, vol. 43, no. 22, pp. 4813–4838, Nov. 2005.

[21] J. Homberger, "A multi-agent system for the decentralized resource-constrained multi-project scheduling problem," *Int. Trans. Oper. Res.*, vol. 14, no. 6, pp. 565–589, Oct. 2007.

[22] J. Homberger and A. Fink, "Generic negotiation mechanisms with side payments—Design, analysis and application for decentralized resource-constrained multi-project scheduling problems," *Eur. J. Oper. Res.*, vol. 261, no. 3, pp. 1001–1012, Sep. 2017.

[23] F. Li and Z. Xu, "A multi-agent system for distributed multi-project scheduling with two-stage decomposition," *PLoS ONE*, vol. 13, no. 10, Oct. 2018, Art. no. e0205445.

[24] Z. Zheng, Z. Guo, Y. Zhu, and X. Zhang, "A critical chains based distributed multi-project scheduling approach," *Neurocomputing*, vol. 143, pp. 282–293, Nov. 2014.

[25] T. Wauters, K. Verbeeck, P. De Causmaecker, and G. V. Berghe, "A learning-based optimization approach to multi-project scheduling," *J. Scheduling*, vol. 18, no. 1, pp. 61–74, Feb. 2015.

[26] S. Adhau, M. L. Mittal, and A. Mittal, "A multi-agent system for decentralized multi-project scheduling with resource transfers," *Int. J. Prod. Econ.*, vol. 146, no. 2, pp. 646–661, Dec. 2013.

[27] S. Zhao and Z. Xu, "A sealed bid auction-based two-stage approach for a decentralized multiproject scheduling problem with resource transfers," *Appl. Intell.*, vol. 52, no. 15, pp. 18081–18100, Apr. 2022.

[28] F. Fu and H. Zhou, "A combined multi-agent system for distributed multi-project scheduling problems," *Appl. Soft Comput.*, vol. 107, Aug. 2021, Art. no. 107402.

[29] Y. Yu, Z. Xu, and S. Zhao, "A two-stage algorithm based on 12 priority rules for the stochastic distributed resource-constrained multi-project scheduling problem with multi-skilled staff," *IEEE Access*, vol. 11, pp. 29554–29565, 2023.

[30] W. You, Z. Xu, Y. Yu, and S. Zhao, "A two-layer approach for solving robust decentralized multiproject scheduling problem with multi-skilled staff," *Int. Trans. Oper. Res.*, vol. 31, no. 3, pp. 1631–1670, Nov. 2023.

[31] T. Hegazy, A. K. Shabeeb, E. Elbeltagi, and T. Cheema, "Algorithm for scheduling with multiskilled constrained resources," *J. Construct. Eng. Manag.*, vol. 126, no. 6, pp. 414–421, Dec. 2000.

[32] E. Neron, "Lower bounds for the multi-skill project scheduling problem," in *Proc. 8th Int. Workshop Proj. Manag. Scheduling*, Apr. 2002, pp. 274–277.

[33] O. Bellenguez-Morineau and E. Néron, "A branch-and-bound method for solving multi-skill project scheduling problem," *RAIRO-Oper. Res.*, vol. 41, no. 2, pp. 155–170, Jun. 2007.

[34] H. Li and K. Womer, "Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm," *J. Scheduling*, vol. 12, no. 3, pp. 281–298, Jun. 2009.

[35] C. Montoya, O. Bellenguez-Morineau, E. Pinson, and D. Rivreau, "Branch-and-price approach for the multi-skill project scheduling problem," *Optim. Lett.*, vol. 8, no. 5, pp. 1721–1734, Jun. 2014.

[36] A. H. Hosseinian and V. Baradaran, "P-GWO and MOFA: Two new algorithms for the MSRCPSP with the deterioration effect and financial constraints (case study of a gas treating company)," *Int. J. Speech Technol.*, vol. 50, no. 7, pp. 2151–2176, Mar. 2020.

[37] J. Snauwaert and M. Vanhoucke, "A new algorithm for resource-constrained project scheduling with breadth and depth of skills," *Eur. J. Oper. Res.*, vol. 292, no. 1, pp. 43–59, Jul. 2021.

[38] B. F. Almeida, I. Correia, and F. Saldanha-da-Gama, "Modeling frameworks for the multi-skill resource-constrained project scheduling problem: A theoretical and empirical comparison," *Int. Trans. Oper. Res.*, vol. 26, no. 3, pp. 946–967, May 2019.

[39] J. Snauwaert and M. Vanhoucke, "Mathematical formulations for project scheduling problems with categorical and hierarchical skills," *Comput. Ind. Eng.*, vol. 169, Jul. 2022, Art. no. 108147.

[40] P. B. Myszkowski, M. E. Skowronski, and K. Sikora, "A new benchmark dataset for multi-skill resource-constrained project scheduling problem," in *Proc. Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Sep. 2015, pp. 129–138.

[41] J. Snauwaert and M. Vanhoucke, "A classification and new benchmark instances for the multi-skilled resource-constrained project scheduling problem," *Eur. J. Oper. Res.*, vol. 307, no. 1, pp. 1–19, May 2023.

[42] M. Firat and C. A. J. Hurkens, "An improved MIP-based approach for a multi-skill workforce scheduling problem," *J. Scheduling*, vol. 15, no. 3, pp. 363–380, Jun. 2012.

[43] P. B. Myszkowski, M. E. Skowronski, L. P. Olech, and K. Oslizlo, "Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem," *Soft Comput.*, vol. 19, no. 12, pp. 3599–3619, Dec. 2015.

[44] L. Zhu, J. Lin, Y.-Y. Li, and Z.-J. Wang, "A decomposition-based multi-objective genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem," *Knowl.-Based Syst.*, vol. 225, Aug. 2021, Art. no. 107099.

[45] Y.-Y. Li, J. Lin, and Z.-J. Wang, "Multi-skill resource constrained project scheduling using a multi-objective discrete Jaya algorithm," *Int. J. Speech Technol.*, vol. 52, no. 5, pp. 5718–5738, Mar. 2022.

[46] R. Kolisch and C. Heimerl, "An efficient metaheuristic for integrated scheduling and staffing IT projects based on a generalized minimum cost flow network," *Nav. Res. Logistics (NRL)*, vol. 59, no. 2, pp. 111–127, Feb. 2012.

[47] M. Walter and J. Zimmermann, "Minimizing average project team size given multi-skilled workers with heterogeneous skill levels," *Comput. Oper. Res.*, vol. 70, pp. 163–179, Jun. 2016.

[48] T. Felberbauer, W. J. Gutjahr, and K. F. Doerner, "Stochastic project management: Multiple projects with multi-skilled human resources," *J. Scheduling*, vol. 22, no. 3, pp. 271–288, Jun. 2019.

[49] M. Haroune, C. Dhib, E. Neron, A. Soukhal, H. M. Babou, and M. F. Nanne, "Multi-project scheduling problem under shared multi-skill resource constraints," *TOP*, vol. 31, no. 1, pp. 194–235, Apr. 2023.

[50] R. Chen, C. Liang, D. Gu, and H. Zhao, "A competence-time-quality scheduling model of multi-skilled staff for IT project portfolio," *Comput. Ind. Eng.*, vol. 139, Jan. 2020, Art. no. 106183.

[51] A. H. Hosseinian and V. Baradaran, "A two-phase approach for solving the multi-skill resource-constrained multi-project scheduling problem: A case study in construction industry," *Eng., Construct. Architectural Manag.*, vol. 30, no. 1, pp. 321–363, Feb. 2023.

[52] N. R. Jennings and M. Wooldridge, "Applying agent technology," *Appl. Artif. Intell.*, vol. 9, no. 4, pp. 357–369, 1995.

[53] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. Berlin, Germany: Springer, 2010, pp. 65–74.

[54] T.-K. Dao, T.-S. Pan, T.-T. Nguyen, and J.-S. Pan, "Parallel bat algorithm for optimizing makespan in job shop scheduling problems," *J. Intell. Manuf.*, vol. 29, no. 2, pp. 451–462, Feb. 2018.

[55] S. Eskandari and M. Seifaddini, "Online and offline streaming feature selection methods with bat algorithm for redundancy analysis," *Pattern Recognit.*, vol. 133, Jan. 2023, Art. no. 109007.

[56] J. Dong, L. Wu, X. Liu, Z. Li, Y. Gao, Y. Zhang, and Q. Yang, "Estimation of daily dew point temperature by using bat algorithm optimization based extreme learning machine," *Appl. Thermal Eng.*, vol. 165, Jan. 2020, Art. no. 114569.

[57] K. Y. Li and R. J. Willis, "An iterative scheduling technique for resource-constrained project scheduling," *Eur. J. Oper. Res.*, vol. 56, no. 3, pp. 370–379, Feb. 1992.

[58] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, Jan. 1986.

[59] T. Servranckx and M. Vanhoucke, "A Tabu search procedure for the resource-constrained project scheduling problem with alternative subgraphs," *Eur. J. Oper. Res.*, vol. 273, no. 3, pp. 841–860, Mar. 2019.

[60] S. Mahmud, A. Abbasi, R. K. Chakrabortty, and M. J. Ryan, "Multi-operator communication based differential evolution with sequential Tabu search approach for job shop scheduling problems," *Appl. Soft Comput.*, vol. 108, Sep. 2021, Art. no. 107470.

[61] B. Peng, S. Wang, D. Liu, Z. Su, Z. Lu, and F. Glover, "Solving the incremental graph drawing problem by multiple neighborhood solution-based Tabu search algorithm," *Exp. Syst. Appl.*, vol. 237, Mar. 2024, Art. no. 121477.

[62] N.-H. Pan, P.-W. Hsaio, and K.-Y. Chen, "A study of project scheduling optimization using Tabu search algorithm," *Eng. Appl. Artif. Intell.*, vol. 21, no. 7, pp. 1101–1112, Oct. 2008.

[63] O. Atli, "Tabu search and an exact algorithm for the solutions of resource-constrained project scheduling problems," *Int. J. Comput. Intell. Syst.*, vol. 4, no. 2, pp. 255–267, 2011.

[64] D. C. Montgomery, "Introduction to factorial designs," in *Design and Analysis of Experiments*. New York, NY, USA: Wiley, 2017, pp. 179–220.

[65] F. Deblaere, E. Demeulemeester, and W. Herroelen, "RESCON: Educational project scheduling software," *Comput. Appl. Eng. Educ.*, vol. 19, no. 2, pp. 327–336, Apr. 2011.

[66] F. Villafáñez, D. Poza, A. López-Paredes, J. Pajares, and R. D. Olmo, "A generic heuristic for multi-project scheduling problems with global and local resource constraints (RCMPSP)," *Soft Comput.*, vol. 23, no. 10, pp. 3465–3479, May 2019.

[67] B. F. Almeida, I. Correia, and F. Saldanha-da-Gama, "A biased random-key genetic algorithm for the project scheduling problem with flexible resources," *TOP*, vol. 26, no. 2, pp. 283–308, Apr. 2018.

WEIBAO YOU was born in Taian, Shandong, China, in 1994. He received the B.S. degree in engineering management and the M.S. degree in architecture and civil engineering from Qingdao University of Technology, Shandong, in 2016 and 2019, respectively. He is currently pursuing the Ph.D. degree in management science and engineering with Beihang University, Beijing, China. His research interests include project scheduling, optimization algorithms, and risk management.

ZHE XU received the Ph.D. degree from the School of Economics and Management, Beihang University, Beijing, China, in 2006. Since 2009, she has been a Professor with the School of Economics and Management, Beihang University. Her research interests include project management, quality management, and risk management.

SONG ZHAO was born in Chengde, Hebei, China, in 1991. He received the B.S. degree in mathematics from China University of Mining and Technology, Beijing, China, in 2014, and the M.S. degree in mathematics and the Ph.D. degree in management science and engineering from Beihang University, Beijing, in 2017 and 2022, respectively. His research interests include project scheduling, optimization algorithms, and risk management.