IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 February 2023, accepted 16 March 2023, date of publication 23 March 2023, date of current version 29 March 2023.
Digital Object Identifier 10.1109/ACCESS.2023.3261139

RESEARCH ARTICLE

A Two-Stage Algorithm Based on 12 Priority Rules for the Stochastic Distributed Resource-Constrained Multi-Project Scheduling Problem With Multi-Skilled Staff

YINING YU, 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 In practical multi-skilled resource-constrained multi-project management, the activity duration is often affected by some factors (e.g., rework, increased workload), leading to uncertainty. Moreover, multiple projects are often managed under a distributed decision-making environment. To deal with uncertain activity durations in distributed multi-project management with multi-skilled staff, this paper studies a stochastic distributed resource-constrained multi-project scheduling problem with multi-skilled staff (MS-SDRCMPSP). In a distributed decision-making environment, a two-stage model with local scheduling and global coordination stages is established to describe MS-SDRCMPSP. A two-stage algorithm with 12 priority rules (TSA-12PRs) is proposed, these 12 priority rules are composed of 4 activity priority rules and 3 resource priority rules. In the local schedule stage, 4 activity priority rules (PRs) are applied to obtain the local schedule plan. In the global coordination phase, we develop 3 resource PRs based on variable neighborhood search (VNS), of which VNS is used to solve the execution order of conflicting projects, and 3 resource PRs are developed to formulate multi-skilled resource assignment strategies. Based on the multi-skilled instances adapted from benchmark instances, we evaluate the performance of the 12 PRs on different instances. The experiment results show that two PRs among 12 PRs perform better than other PRs in all-size instances. Comparing the two-stage algorithm with better two PRs with other approaches in literatures, we find that our method performs better than other approaches, especially in large-size instances. In addition, further experiments show that our method is more conducive to shortening the CPU runtime on distributed problems than centralized methods.

INDEX TERMS Multi-project scheduling, multi-skilled staff, uncertain activity durations, stochastic scheduling priority rules.

I. INTRODUCTION

Resource-constrained multi-project scheduling problem (RCMPSP) contains a series of projects, each of which contains activities that satisfy the resource availability and priority relationship constraints when obtaining a multi-project scheduling plan [1]. An extension to RCMPSP is the multi-skilled resource-constrained multi-project scheduling problem (MS-RCMPSP) [2]. A classic practical scenario about MS-RCMPSP can be found in software development [3]. There exists a multi-project management environment in MS-RCMPSP. Multiple projects compete for limited multi-skilled staff that primary several skills. The assignment of multi-skilled resources involves the matching relationship of "activity-skill-resource"; i.e., implementing activities in the project requires the completion of multi-skilled staff who primary specific skills. The case of the "activity-skill-resource" is shown in Fig. 1.

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

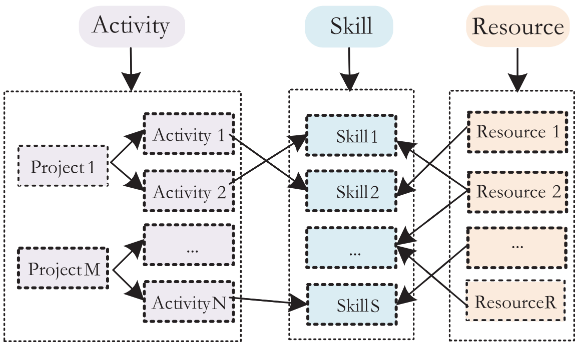


FIGURE 1. The relationship of the "activity-skill-resource".

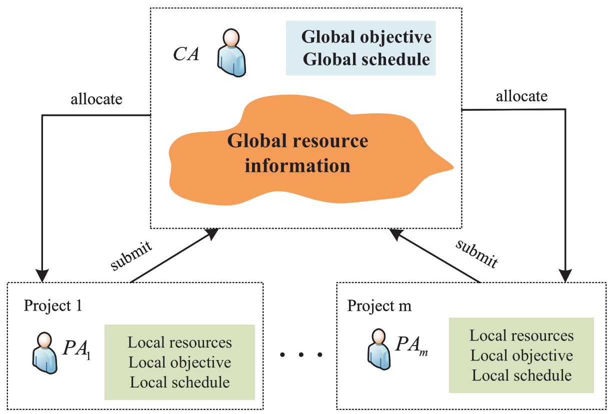


FIGURE 2. The function of MAS in DRCMPSP.

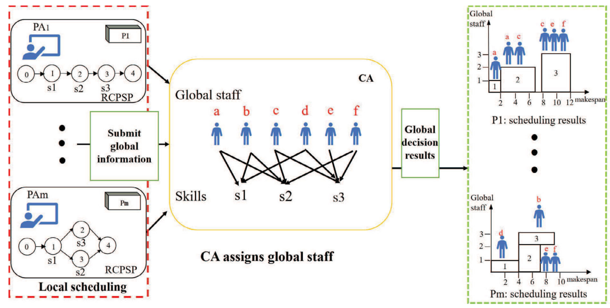


FIGURE 3. The case of MS-RCMPSP.

The projects undertaken by enterprises become larger and more complex, so enterprises often choose to undertake multiple projects at the same time. At the same time, organizational management is also gradually distributed (distributed decision-making environment). There are multiple project decision-makers, and all projects are independent. The only connection among projects is to share limited resources. When multiple projects compete for shared resources, shared resources are prone to conflict due to resource limitations and project independence, which forms a distributed resource-constrained multi-project scheduling problem (DRCMPSP) [4]. The solution to DRCMPSP involves the independent scheduling of multiple single-project executed simultaneously and the reasonable allocation of shared resources among projects. Generally, a multi-agent system (MAS) is used for solving the DRCMPSP, which includes multiple project agents (PA) and one coordinating agent (CA). Each PA has local resources to pursue its local interest objective and does not disclose local information to each other. CA coordinates shared resources (global resources) for each PA according to the global objective. The function of MAS in DRCMPSP is shown in Fig. 2.

In the distributed decision-making environment, it is more and more common to share resources with multi-skilled staff based on multiple projects. For instance, a software development company has two project teams that undertake an online office platform and an e-commerce project. The two project teams share the multi-skilled staff with coding, testing, and other skills. Each project needs limited local resources managed by the project manager and shared global staff managed by the coordinating manager. This problem needs to solve the independent scheduling of each project and the allocation of global multi-skilled staff. Therefore, when facing such distributed problems, it is necessary to study the distributed resource-constrained multi-project scheduling problem with multi-skilled staff (MS-DRCMPSP) [5]. In MS-DRCMPSP, each PA submits the global information to CA, and then CA assigns the multi-skilled staff by considering the characteristics of multi-skilled staff heterogeneity and the matching relationship of "activity-skill-resource". Further, CA solves the global multi-skilled staff conflicts by developing an effective coordination mechanism, which includes determining the start time of the activity and assigning multi-skilled staff. When multi-skilled staff primary several skills and the skill level is different (multi-skilled heterogeneity), the actual duration of the activity will be also changed. These further increase the difficulty of solving MS-DRCMPSP. As an extension of DRCMPSP, MS-DRCMPSP belongs to NP-hard. The case of the MS-DRCMPSP is shown in Fig. 3.

In the existing literature, scholars only study deterministic problems where all parameters are deterministic. However, during the practical implementation of the project, there are often uncertain situations, the most common of which are uncertain activity durations. When uncertainty occurs, activities are interrupted, resulting in delays and losses in practical project management. For example, uncertain activity durations make the baseline scheduling plan and the resource assignment strategy impossible. Stochastic scheduling is often used to solve the uncertainty in project scheduling. In stochastic scheduling, a scheduling policy is obtained instead of a specific scheduling plan [6]. In addition, stochastic scheduling is a common method to solve the problem of uncertain activity duration, in which the activity duration usually follows a known distribution. There is no literature that studies the multi-skilled resource-constrained multi-project scheduling problem with uncertain activity durations by stochastic scheduling.

Based on the MS-DRCMPSP and uncertain activity durations, we proposed the stochastic distributed resource-constrained multi-project scheduling problem with multi-skilled staff (MS-SDRCMPSP). Each project is scheduled by its PA to optimize the expected project makespan (local objective) in this problem. CA assigns multi-skilled staff for each project depending on minimizing the expected total tardiness costs (global objective). Further, heterogeneous characteristics of multi-skilled staff and uncertain activity durations significantly increase the difficulty of modeling and solving MS-SDRCMPSP. We attempt to develop a two-stage algorithm with 12 priority rules (TSA-12PRs) and variable neighborhood search algorithm (VNS) by multi-skilled staff heterogeneous characteristics for the MS-SDRCMPSP, especially designing some effective multi-skilled staff assignment strategies.

The main contributions of this article can be described as follows:

Firstly, multi-skilled staff are considered shared resources in a distributed multi-project environment with uncertain activity durations.

Secondly, a two-stage algorithm with 12 priority rules (TSA-12PRs) is proposed to solve this problem, among which 4 activity PRs for solving the local schedule stage and variable neighborhood search (VNS) with 3 resource PRs for solving the global coordination stage.

Thirdly, the TSA-12PRs are evaluated on different size problems, and the two best PRs are selected on all size instances. Further experiments compared a distributed approach (SGNM) and a centralized method (BRKGA).

The rest of this article is structured as follows. A summary of existing publications in this field is presented in Section II. Section III is devoted to the problem description and mathematical formulations. In Section IV, the 12PRs and the proposed algorithms are used in this paper to solve the MS-SDRCMPSP. A comprehensive experimental analysis is provided in Section V. Finally, Section VI summarizes the article.

II. LITERATURE REVIEW

In this section, a brief review of the existing literature relevant to this article is conducted. It includes the distributed resource-constrained multi-project scheduling problem, the multi-skilled resources-constrained multi-project scheduling problem, and the stochastic project scheduling problem.

A. MULTI-SKILLED RESOURCE-CONSTRAINED MULTI-PROJECT SCHEDULING PROBLEM

Hegazy et al. were the first to propose the multi-skilled project scheduling problem [7]. The Bellenguez team considered the characterize of multi-skilled resources and defined the MS-RCPSP in the current paper [8], [9]. Now the researches about multi-skilled staff mainly concentrate on a single-project environment. They are mainly divided into multi-skilled resource-heterogeneous project scheduling problems [10], [11], multi-mode multi-skilled resource-constrained project scheduling problems [12], [13], [14], multi-objective multi-skilled resource-constrained project scheduling problems [15], [16].

There are few related types of research on the multi-skilled resource-constrained multi-project scheduling problem. The leading research is a centralized scheduling method, that is, there is only one decision-maker. Heimerl and Kolisch [17] studied the MS-RCMPSP in the centralized decision-making environment of IT enterprises. Considering multi-skilled outsourced resources, the meta-heuristic algorithm was used to solve MS-RCMPSP [17]. Walter and Zimmermann [18] studied MS-RCMPSP from the perspective of the team size. Chen et al. [19] studied the learning effect of MS-RCMPSP, established a multi-objective constrained model of MS-RCMPSP, and used a non-dominated genetic algorithm (NSGAII) to solve the problem; Some studies (Wu and Sun [20]; Gutjahr et al. [21]) also studied the centralized MS-RCMPSP [20], [21], [22]. In summary, a centralized approach with only one decision-maker is unsuitable for multiple decision-makers. It lacks a certain degree of flexibility and can not only partially satisfy all projects.

Conversely, the distributed approach is more suitable for management environments with multiple decision-makers. So far, only scholar Yu et al. [5]studied the multi-skilled distributed resource-constrained multi-project scheduling problem, established a two-stage decision-making model, and provided practical management suggestions. Therefore, there are more possibilities for the MS-DRCMPSP, especially under uncertain situations.

B. DISTRIBUTED RESOURCE CONSTRAINED MULTI-PROJECT SCHEDULING PROBLEM

Generally, the distributed scheduling problem is solved by multi-agent systems (MAS). In the MAS, there are multiple Project Agents (PAs) as the project managers and a Coordinating Agent (CA) as the coordinating manager [23], [24], [25], [26]. Homberger solved DRCMPSP through an electronic iterative negotiation mechanism based on the mediation protocol [27]; Subsequently, Homberger continued to expand the DRCMPSP model, proposed the restart evolutionary algorithm, and solved the large-size example problem [28]. Zheng et al. [29]solved the DRCMPSP by a critical-chain method. Wang et al. [30] researched a resource-constrained project scheduling problem with a fractional shared resource by a column-generation-based algorithm. Rostami et al. [31] designed a lagrangian relaxation algorithm for DRCMPSP and verified the effectiveness of the method.

Recently, the distributed approach has also been applied in different areas. For example, Zhao and Xu [32] discussed the distributed multi-project scheduling problem with the transfer time. Kosztyan [33]studied the flexible multi-level project scheduling problem by a matrix-based multi-level multi-mode project scheduling algorithm. Other scholars (Babaee Tirkolaee et al. [34]; Tirkolaee et al. [35]; Mahdavi et al. [36]) also studied the multi-trip scheduling problem in different fields, such as waste management, heat storage systems, and two-echelon supply chain management.

C. STOCHASTIC PROJECT SCHEDULING PROBLEM

In stochastic scheduling, project activity duration is often known and assumed to follow a distribution; the objective is to minimize the expected makespan of the project. Given the randomness of the activity duration, the solution of SRCPSP is a scheduling policy rather than a deterministic schedule. For solving the SRCPSP, there are two main methods. One of them is a meta-heuristic to solve the SRCPSP. For example, genetic algorithms (Chen et al. [37]; Zaman et al. [38]), tabu search (Servranckx and Vanhoucke [39]). The other is the priority rules-based heuristic [37], [38], [39].

In contrast to meta-heuristics, the priority rules-based heuristic relies on creating a priority rule instead of multiple iterations. It includes two types of elementary policies, which can be subdivided into static policies and dynamic policies. The static policy means that the strategy is given before the project is executed. During the execution process, the strategy will not change. Some static policies are often used in the literature, such as resource-based policy class, activity-based policy class, earliest-start policy class, preselective policy class, preprocessor policy class, and generalized preprocessor policy class [40], [41], [42]. The dynamic policy seeks the best policy step by step in a dynamic way. Its solutions must be constantly updated according to new information, such as the activity start time. Thus, dynamic policies usually need more CPU runtime.

Scholars Thomas F, et al. studied uncertain problems by stochastic scheduling under the centralized environment. The research expanded the deterministic model and solved the uncertain MS-RCMPSP through the Frank Wolfe algorithm [6]. However, under the distributed decision-making environment, there is no literature that studies the multi-skilled resource-constrained multi-project scheduling problem with uncertain activity durations by stochastic scheduling.

Considering the complexity of the distributed environment and multi-skilled heterogeneity, a dynamic strategy will significantly increase the CPU runtime of the stochastic scheduling. It is more realistic that a decision-maker needs to make decisions as quickly as possible with a static strategy. Therefore, it is more important to study the stochastic distributed multi-skilled resource-constrained multi-project scheduling problem by the static policy.

III. PROBLEM MODELING

In this section, stochastic distributed resource-constrained multi-project scheduling problem with multi-skilled staff is introduced in detail, and a two-stage model is established.

A. PROBLEM DESCRIPTION

In this problem, M projects are scheduled simultaneously. Each project $t \in 1, 2, \ldots, M$ has an arrival date $w_i \geq 0$ denoting its earliest possible start time. In project i, there are a set of activities noted as $a_{ij} \in V_i(V_i = \{a_{i0}, \ldots, a_{iJ_i}\})$, where activity a_{i0} and activity a_{iJ_i} represent two dummy activities at the beginning and the end. E_{ij} denotes the predecessor activity set for a_{ij}, that is, a_{ij} can be started only after all activities in E_{ij} are completed. A dummy activity has a duration of zero and no resource usage under any probability distribution; oppositely, the duration of a non-dummy activity is a random variable and obeys a known probability distribution. Non-dummy activities also require several types of local resources and at most one type of global resource. The local resource $k(k = 1, 2, \ldots, K_i)$ stands for the ordinary staff who has only one skill. The global resource $g(g = 1, 2, \ldots, G)$ refers to the multi-skilled staff who primary several skills.

When multiple PAs conflict with limited global resources, some activities can not get global resources in time and will cause their projects to be postponed. Each project has a unit delay cost tc_i and a completion time $B_i(B_i = t \cdot x_{iJ_it})$. CA designs a coordination mechanism to assign global staff to each PA. There is no connection among projects except for sharing limited global staff. During all project execution, one staff can only use one skill to perform one activity simultaneously. At any moment, the skill demands for all activities being performed in the multi-project cannot exceed the total available global resources in providing skills at current moment. Similarly, the local resources also have a limited quantity R_{ik}.

Each PA performs local scheduling of one project to minimize the expected makespan of this project, which is described as the following (1) and (2).

$$\min E(f_p) \tag{1}$$

$$f_p = \sum_{t=0}^{T} t \cdot x_{iJ_it}. \tag{2}$$

As a global decision-maker, the objective of CA is to find a feasible schedule and global resources assignment plan with the goal of minimizing the expected total tardiness costs (TTC) denoted by (1) and (4).

$$\min E(TTC) \tag{3}$$

$$TTC = \sum_{i=1}^{M} tc_i \cdot (B_i - w_i - CPD_i) \tag{4}$$

This objective refers to the sum of the delay costs caused by the actual makespan $(B_i - w_i)$ of each project exceeding the critical path duration (CPD_i).

The following particular assumptions are considered in this article:

- An activity can only be executed after all the resources required by the activity have arrived.
- All multi-skilled staff have the characteristics of heterogeneity. In other words, each resource may have a different level for each skill $s(s = 1, 2, \ldots, S)$. The higher the skill level, the shorter the actual duration.

B. NOTATION

1) PARAMETERS

l_{gs} - the level of skill s primaried by the resource g.

r_{ij}^s - the skill s demand of the activity a_{ij}.

v_{ij}^s - a Boolean variable indicating when skill s is required by activity a_{ij} that equals 1 and equals 0 otherwise.

$\bar{d}_{ij}$ - the planned duration of activity a_{ij}, that is, the staff with $l_{gs} = 1$ performs the activity, $\bar{d}_{ij}$ follows a random variable with a known probability distribution.

d_{ij} - the actual duration of activity a_{ij}, the actual duration is effected by the level of skill, see (5).

2) DECISION VARIABLE

x_{iJ_it}: 1, if the activity a_{ij} starts at time t; 0, otherwise;

y_{ijt}^{gs}: 1, if the resource g with the skill s perform activity a_{ij} at time t; 0, otherwise;

$$d_{ij} = \left\lceil r_{ij}^s \cdot \bar{d}_{ij} / \sum_{g=1} l_{gs} \cdot y_{ijt}^{gs} \cdot v_{ij}^s \right\rceil \tag{5}$$

IV. TWO-STAGE ALGORITHM BASED ON 12 PRIORITY RULES

In order to obtain a complete multi-project scheduling plan in the distributed environment, a two-stage coordination process is designed at each conflicting time. Here, conflicting time refers to when more than two projects require global resources, and these two projects are also called conflicting projects. MAS comprises PA as a project manager and CA as a coordinating manager.

In stage one, the local scheduling is a statistic scheduling approach with the goal of the expected project makespan by each PA. Each PA solves the local scheduling based on activity priority rules (in Section IV-A1). In stage two, the global coordination process not only needs to determine the execution order of conflicting projects (in Section IV-B) at each conflicting time, but also needs to assign multi-skilled resources (in Section IV-A2) reasonably. CA solves the global coordination process to minimize the expected total delay costs of multiple projects. This paper combines the characteristics of multi-skilled resources and designes a heuristic strategy based on 12 priority rules.

Fig. 4 presents an overview of the global coordination press.

Step1:From the time 0, after all PAs finish the local scheduling by activity PRs (in Section IV-A1), they submit the global resources demand information to the CA.

Step2:At this time, if there is only one project requiring global resources, CA assigns global resources for this activity according to the resource PRs (in Section IV-A2); Suppose there are more than two projects requiring global resources, CA ranks conflicting projects by Variable Neighborhood Search (VNS, in Section IV-B) and coordinates global resource assignments by resource PRs (in Section IV-B).

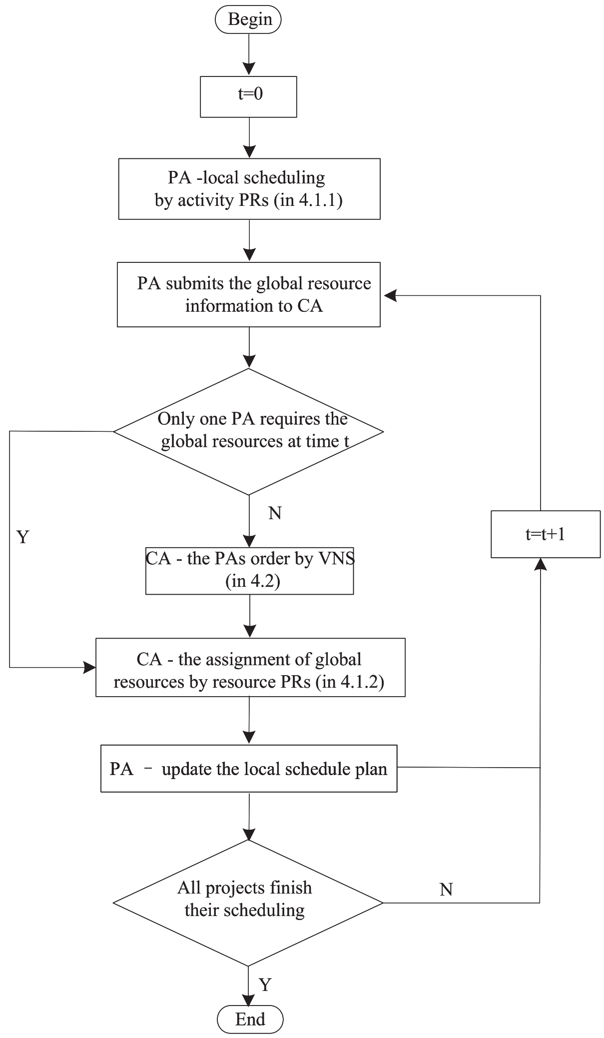


FIGURE 4. An overview of the global coordination press.

A. PRIORITY RULE BASE HEURISTIC POLICY

The local scheduling is managed by each PA, which is to minimize the expected makespan of this project. In the local scheduling, each PA obtains the local scheduling by activity priority rules. At the conflicting time, CA assigns the global resources for a selected project by resource priority rules.

1) ACTIVITY PRIORITY RULES

This section exhibits the 4 activity priority rules for the local SRCPSP. These rules perform better than other static priority rule heuristic policies for SRCPSP [43], [44], [45]. Then we test the suitability of these 4 activity priority rules on MS-SDRCMPSP.

This section exhibits the 4 activity priority rules (in Table 1) for the local SRCPSP. These rules perform better than other static priority rule heuristic policies for SRCPSP. Then we will test the suitability of these 4 activity priority rules on MS-SDRCMPSP.

TABLE 1. Four activity priority rules for the local SRCPSP.

Priority rules	Extreme	Calculation formula	Comments
Latest start time(LST)	MIN	LS_{ij}	LS_{ij} is the latest start time of activity.
Latest finish time(LFT)	MIN	LF_{ij}	LF_{ij} is the latest finish time of activity, α is the index of the simulation, ns is the total number of simulations.
Statistical latest start time (SLST)	MIN	$1/ns \cdot \sum_{\alpha=1}^{ns}(LS_{ij}^{\alpha})$	LS_{ij} is the latest start time calculated by the critical path method CPM in simulation α.
Statistical latest finish time (SLFT)	MIN	$1/ns \cdot \sum_{\alpha=1}^{ns}(LF_{ij}^{\alpha})$	LF_{ij} is the latest finish time calculated by the critical path method CPM in simulation α.

2) RESOURCE PRIORITY RULES

We choose three resource priority rules with good performance from the existing literature (Snauwaert and Vanhoucke [48]); that is, Highest Average Level (HAL), Lowest Average Level (LAL), Lowest Number & Highest Level (LN&HL). Then we design a new resource priority rule called Highest Level &Lowest Number (HL&LN). This section explains that 4 resource priority rules are used at the conflicting time. Some of them are based on the skill-level or the skill-number.

a: HIGHEST AVERAGE LEVEL (HAL)

This rule uses the skill level to assign multi-skilled staff. Resources are ranked based on the average level of their mastered skills, then activities with HAL are selected first, which indicates that the most efficient resources will be prioritized.

b: LOWEST AVERAGE LEVEL (LAL)

This priority rule is the opposite of the previous rule. Resources will be ranked from the lowest to the highest average depth, indicating that the least efficient resources will be prioritized. Similar to Highest Breadth First, this rule adds diversity to the set of priority rules.

c: HIGHEST LEVEL &LOWEST NUMBER (HL&LN)

The rule considers both skill-level and skill-number of resources. It gives priority to resources with the highest skill-level and selects resources with the lowest skill-number as tie-breakers. The objective of this rule is to minimize the makespan of the activity by utilizing resources with the highest skill-level.

d: LOWEST NUMBER & HIGHEST LEVEL (LN&HL)

This rule also considers the skill-level and the skill-number. In this case, the lowest skill-number resources are prioritized, and the highest skill-level resources are used as tie-breakers. This rule aims to keep the most skill-number resources available while assigning the most efficient resources.

In the local scheduling, 4 activity priority rules are chosen. For each activity priority rule, there are 4 resource assignment rules. By Combining activity priority rules and resource assignment rules, we obtain 16 priority rules based on heuristic strategies. They are LST-HAL, LST-LAL, LST-HL&LN, LST-LN&HL; LFT-HAL, LFT-LAL, LFT-HL&LN, LFT-LN&HL; SLST-HAL, SLST-LAL, SLST-HL&LN, SLST-LN&HL; SLFT-HAL, SLFT-LAL, SLFT-HL&LN, SLFT-LN&HL.

B. VARIABLE NEIGHBORHOOD SEARCH DESIGN

To minimize the global objective, CA determines an execution order of all conflicting projects at each conflicting time.

Firstly, an initial order L2 is selected from highest to lowest unit cost; projects with higher unit costs are prioritized for execution, which helps narrow the search to find a reasonable solution. Then CA assigns the corresponding global resources for projects in L2 according to the resource PRs (in Section IV-A2), and calculates the total delay cost (an initial solution S_{init}) of multi-projects in L2. Finally, starting from position 1 (the first project at the current order i_{pro}), exchanging the order with the following projects as a new order. CA assigns the global resources based on the resource PRs (in Section IV-A2) and calculates the new solution S_{new} in the current order. If the new solution S_{new} is larger than the initial solution S_{init}, the order will be updated again; otherwise, replace the initial solution and return to position 1. We continue to perform the variable neighborhood search until the best solution All_{delay} is obtained at the current moment.

Algorithm 1 The Postcode of Variable Neighborhood Search (VNS)

Require:
LP: the set of all conflicting projects at the current conflicting time;
$L1$: the set of projects at least one activities meet the skill availability for each project in LP.

```
if L1 ≠ then
    Obtain initial order L2; %selected in order by highest
    unit deferred cost first
    Calculate S_init ; %the initial solution.
    Record i_pro = 1; Count = 0.
    if length(L1(1)) = 1 then
        L1_delay = S_init;
    else
        %change the order of adjacent projects
        while i_pro < allocate_pro do
            Count = Count + 1; 0
            a = L2(1, i_pro);
            L2(1, i_pro) = L2(1, i_pro + 1);
            L2(1, i_pro + 1) = a;
            Calculate S_new; % the new solution
            if S_new ≥ S_init then
                i_pro = i_pro + 1;
            else
                i_pro = 1;
                S_init = S_new;
            end if
            L1_delay = S_init;
        end while
    end if
end if
```

Ensure:
$All_{delay} = L1_{delay}$.

TABLE 2. Five distributions for activity durations.

Distribution type	Distribution code	Range	Shape parameter	Variance
Uniform	U1	$U(\bar{d}_{ij} - \sqrt[2]{\bar{d}_{ij}}, \bar{d}_{ij} + \sqrt[2]{\bar{d}_{ij}})$	-	$\bar{d}_{ij}/3$
distribution	U2	$U(0, 2\bar{d}_{ij})$	-	$\bar{d}_{ij}^2/3$
Beta	B1	$B(\bar{d}_{ij}/2, 2\bar{d}_{ij})$	$\alpha = \bar{d}_{ij}/2 - 1/3, \beta = 2\alpha$	$\bar{d}_{ij}/3$
distribution	B2	$B(\bar{d}_{ij}/2, 2\bar{d}_{ij})$	$\alpha = 1/6, \beta = 2\alpha$	$\bar{d}_{ij}^2/3$
Exponential distribution	EXP	$E(\bar{d}_{ij})$	- -	$\bar{d}_{ij}^2$

V. COMPUTATIONAL STUDY

A series of experiments are carried out to conduct the computational study. All designed test problems are solved in Matlab R2018b, with a core i7 CPU and 16 GB memory. This section contains four parts: Section V-A introduce the problem instances; Section V-B analyzes the impact of simulated times on the priority rules; Section V-C verifies the performance of the TSA-12 PRs on different distributions. Section V-D analysis the performance of the TSA-12 PRs compared with other algorithms (including the distributed method and centralized method).

In order to be able to compare the PR results with previous works on the SRCPSP, we first follow the existing literature in line with their probability distribution types and parameters [41], [46]. Assuming that the activity duration is a random variable, the mean value of the duration is equal to the deterministic duration of the MPSPLIB data set. The five distributions and their variances are shown in Table 2. Variances represent different degrees of uncertainty: the variance of distribution U1 and B1 is the smallest, the variance of U2 and B2 is the middle, and the variance of Exp is the largest. Additionally, in each beta distribution, the shape of the distribution is determined by two main shape parameters, α and β. For subsequent experiments, we used the two most commonly used parameters as reported in the literature, and the specific values of the shape parameters are shown in Table 2.

A. PROBLEM INSTANCES

Yu et al. [5] provided problem instances for the MS-DRCMPSP under certainty. In this paper, we introduce uncertainty in activity duration to the MS-DRCMPSP, which increases the problem complexity in terms of solution times and solving difficulty for the MS-SDRCMPSP. Wang et al. [47] demonstrated that the J30 dataset is already representative in the multi-skilled project scheduling problem under uncertainty. Therefore, it is reasonable that we selected 20 instances from the literature [5], including the J30 and J90 datasets generated from the MPSPLIB.

The problem instances are shown in Table 3. These instances are classified into 4 subsets. Each problem subset is named as MPJ_i_m (MP subset), where the number of activities J_i per project m is 30, 90, and the number of projects is 2, 5, *NOI* denotes the number of instances primaried by each problem subset. According to the problem size, MP90_5 is called a large-size instance, and the other three problem sets are called small-size instances.

Further multi-skilled information required to generate instances are:

- Each problem instance is provided with at most one type of global resource and three types of local resources.
- The types of skills with project numbers of 2, 5 are set to 3, 5, respectively.
- The value of the parameters r_{ij}^s are generated in the range of [1] and [3] uniformly.
- Each staff primaries the types of skills are generated in the range of [2] and [3] uniformly.
- The level of skill is randomly generated in the range of 0.6, 0.8, and 1, respectively. Such skill levels are valid and guarantee the speed at which the resource executes the activity [48], so staff with lower skill levels are not considered.

TABLE 3. Problem instances of MS-SDRCMPSP.

Problem subsets	NOI	m	J_i	Probelm size	Skill types
$MP30\_2$	5	2	30	60	3
$MP90\_2$	5	2	90	180	5
$MP30\_5$	5	5	30	150	3
$MP90\_5$	5	5	90	450	5

B. THE INFLUENCE OF SIMULATION TIMES AND SELECTION OF PRIORITY RULES

*MP*30_2 and *MP*90_2 problem sets are used for pre-experiment when selecting simulated times. In order to test the significance of the results among the three simulated times, we first perform the paired samples Wilcoxon signed-rank test for 12 PRs of each problem set. The significance level is set to 5%. Table 4 and 5 show the statistical test results on *MP*30_2 and *MP*90_2, respectively.

TABLE 4. Wilcoxon signed rank test results for 12 PRs on *MP*30_2.

Priority rules	10vs30	10vs50	30vs50
LST-HL&LN	0.000	0.034	0.228
LST-HAL	0.000	0.000	0.297
LST-LN&HL	0.000	0.000	0.284
LST-LAL	0.000	0.000	0.345
LFT-HL&LN	0.004	0.001	0.149
LFT-HAL	0.001	0.000	0.647
LFT-LN&HL	0.001	0.039	0.334
LFT-LAL	0.001	0.000	0.647
SLST-HL&LN	0.001	0.009	0.905
SLST-HAL	0.017	0.000	0.512
SLST-LN&HL	0.004	0.037	0.098
SLST-LAL	0.005	0.045	0.145
SLFT-HL&LN	0.007	0.000	0.547
SLFT-HAL	0.041	0.000	0.07
SLFT-LN&HL	0.003	0.002	0.653
SLFT-LAL	0.004	0.023	0.489

The results indicate significant differences in the quality of solutions obtained for 12 PRs with 10 vs 30 simulated times and 10 vs 50 simulated trajectories, but no significant differences between 30 and 50 simulated trajectories. Comparing the solutions, we find the results of 30 and 50 simulations are statistically better than that of 10 simulations. To balance solution quality and computational runtime, we set the number of simulated times to 30 in this paper.

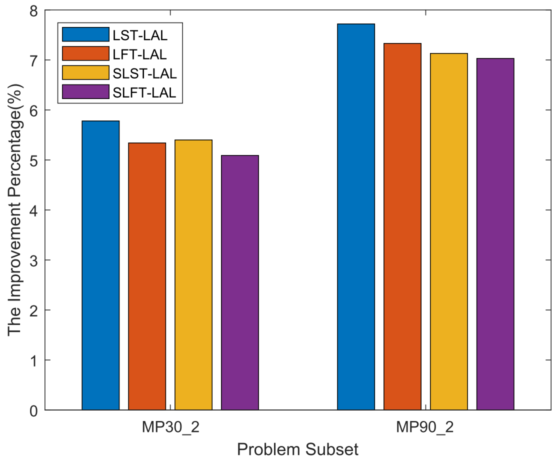


FIGURE 5. The deviation of the combination of the four worst heuristic rules.

TABLE 5. Wilcoxon signed rank test results for *MP*90_2.

Priority rules	10vs30	10vs50	30vs50
LST-HL&LN	0.000	0.001	0.134
LST-HAL	0.011	0.022	0.073
LST-LN&HL	0.007	0.015	0.201
LST-LAL	0.000	0.000	0.06
LFT-HL&LN	0.000	0.000	0.06
LFT-HAL	0.000	0.002	0.102
LFT-LN&HL	0.016	0.002	0.560
LFT-LAL	0.003	0.017	0.199
SLST-HL&LN	0.023	0.012	0.072
SLST-HAL	0.003	0.017	0.199
SLST-LN&HL	0.010	0.000	0.063
SLST-LAL	0.000	0.000	0.08
SLFT-HL&LN	0.037	0.02	0.082
SLFT-HAL	0.000	0.000	0.480
SLFT-LN&HL	0.000	0.000	0.08
SLFT-LAL	0.023	0.032	0.527

To choose better combinations of priority rules, we test the results of different priority rules with *MP*30_2 and *MP*90_2 under the U1 distribution. We sort 16 rule combinations and find that the last four combinations ranked had poor results: LST-LAL, LFT-LAL, SLST-LAL, and SLFT-LAL. After analyzing the deviation between the four poorly ranked combinations and the 12th combination, we found that the deviation was greater than 5%. As a result, we excluded the four underperforming combinations and selected the top 12 combinations for further analysis. Fig. 5 shows the deviation of the combination of the last four worst heuristic rules on *MP*30_2 and *MP*90_2.

C. PERFORMANCE OF THE 12 PRIORITY RULES UNDER DIFFERENT DISTRIBUTIONS

Table 6, 7, 8 and 9 show the average tardiness cost of 30 runs for *MP*30_2,*MP*90_2, *MP*30_5 and *MP*90_5 under different distributions. The bold values in each table stand for the minimum average E(TTC) under the same activity priority rule. The underlined values in each column are the best results obtained by all precedence rules under the same distribution. According to the bold and underlined values, the combination of several priority rules that perform better in each problem set is shown in bold.

TABLE 6. Average $E(TTC)$ of *MP*30_2 obtained by 12 priority rules.

Priority rules	U1	U2	β1	β 2	EXP
LST-HL&LN	**3023.67**	**3042.29**	**3035.76**	**3061**	**3324.47**
LST-HAL	3154.35	3184.23	3160.86	3194.96	3423.76
LST-LN&HL	3320.17	3345.09	3322.71	3371.85	3604.32
LFT-HL&LN	**3016.71**	**3068.62**	**3006.84**	**3059.47**	**3302.14**
LFT-HAL	3142.02	3212.6	3136.42	3193.65	3419.34
LFT-LN&HL	3305.43	3356.41	3296.85	3341.8	3589.17
SLST-HL&LN	**3036.23**	**3062.8**	**3009.77**	**3100.61**	**3402.15**
SLST-HAL	3154.23	3248.29	3160.95	3255.71	3578.86
SLST-LN&HL	3321.27	3368.41	3288.57	3375.45	3600.16
SLFT-HL&LN	**3010.11**	**3075.64**	**3027.03**	**3075.32**	**3329.27**
SLFT-HAL	3150.98	3220.71	3169.29	3222.07	3550.47
SLFT-LN&HL	3303.45	3358.58	3289	3361.13	3579.88

TABLE 7. Average $E(TTC)$ of *MP*90_2 obtained by 12 PRs.

Priority rules	U1	U2	β1	β2	EXP
LST-HL&LN	**11039.85**	**11220.66**	**11008.83**	**11281.79**	**11678.12**
LST-HAL	11633.53	11758.55	11690.36	11759.57	12179.34
LST-LN&HL	12212.18	12417.89	12235.78	12456.94	12873.27
LFT-HL&LN	**10896.86**	**11198.75**	**10943.73**	**11258.27**	**11609.73**
LFT-HAL	11603.21	11639.97	11659.89	11730.65	12289.85
LFT-LN&HL	12199.64	12374.59	12195.97	12376.92	12875.18
SLST-HL&LN	**11002.09**	**11265.67**	**11030.38**	**11306.71**	**11885.24**
SLST-HAL	11559.88	11793.7	11613.51	11779.71	12278.12
SLST-LN&HL	12190.56	12456.2	12223.1	12490.03	12991.23
SLFT-HL&LN	**10933.46**	**11240.59**	**10915.85**	**11232.69**	**11683.83**
SLFT-HAL	11491.68	11753.73	11528.47	11774.15	12293.39
SLFT-LN&HL	12165.7	12404.97	12185.37	12350.18	12843.46

TABLE 8. Average $E(TTC)$ of *MP*30_5 obtained by 12 PRs.

Priority rules	U1	U2	β1	β2	EXP
LST-HL&LN	**5812.9**	**6052.59**	**5888.8**	**6121.14**	**6447.23**
LST-HAL	6206.45	6473.87	6282.32	6570.65	6901.57
LST-LN&HL	6168.32	6427.13	6192.69	6365.18	6689.12
LFT-HL&LN	**5795.91**	**6256.48**	**5980.19**	**6091.33**	**6243.89**
LFT-HAL	6213.62	6668.51	6410.05	6511.97	6834.87
LFT-LN&HL	6146.85	6593.17	6335.89	6418.27	6739.25
SLST-HL&LN	**5728.53**	**5825.32**	**5808.91**	**5904.13**	**6399.37**
SLST-HAL	6164.77	6290.81	6205.88	6413.35	6735.86
SLST-LN&HL	6089.65	6310.05	6056.58	6321.77	6634.19
SLFT-HL&LN	**5695.69**	**5995.13**	**5773.83**	**5917.97**	**6283.29**
SLFT-HAL	6176.07	6408.43	6233.65	6485.75	6789.52
SLFT-LN&HL	6012.35	6315.64	6160.71	6187.11	6502.13

TABLE 9. Average $E(TTC)$ of *MP*90_5 obtained by 12 PRs.

Priority rules	U1	U2	β1	β2	EXP
LST-HL&LN	**33884.29**	**34588.26**	**33674.91**	**34717.12**	**37350.72**
LST-HAL	36037.3	37006.7	36251.89	37289.69	40378.67
LST-LN&HL	36372.31	37595.55	36643.44	37530.03	40549.04
LFT-HL&LN	**33690.7**	**34586.91**	**33525.71**	**34684.8**	**37130.17**
LFT-HAL	36330.81	37018.39	36175.43	37188.39	40123.56
LFT-LN&HL	36218.95	37469.55	36291.29	37000.73	39235.29
SLST-HL&LN	**33735.25**	**34945.97**	**33671.58**	**34776.01**	**37298.12**
SLST-HAL	35851.27	37318.76	36381.55	37490.25	40245.72
SLST-LN&HL	36178.25	37563.86	36516.92	37708.65	40456.53
SLFT-HL&LN	**33623.21**	**34649.37**	**33516.02**	**34694.93**	**37100.56**
SLFT-HAL	35720.23	37243.63	36192	37396.55	39328.17
SLFT-LN&HL	36021.69	37474.67	36310.65	37683.72	40145.29

In Table 6, for *MP*30_2 problem set, LFT-HL&LN has the best performance under B1, B2, and EXP, and SLFT-HL&LN and LST-HL&LN perform best under U1 and U2, respectively. In Table 7, for *MP*90_2 problem set, LFT-HL&LN has the best performance under U1, U2, and EXP, and SLFT-HL&LN performs best under B1 and B2. In Table 8, for *MP*30_5 problem set, SLST-HL&LN and SLFT-HL&LN are the best compared with other priority rules under U2, B2 and U1, B1, respectively. However, LFT-HL&LN has the best performance under EXP. While in Table 9, for *MP*90_5 problem set, LFT-HL&LN performs better than other priority rules under U2 and B2. However, SLFT-HL&LN has the best performance under U1, B1 and EXP.

Additionally, in all Tables, HL&LN is beneficial in reducing the objective under all distributions. Since resources with the highest skill-level and the lowest skill-number can perform activities as soon as possible, effectively reducing actual activity durations and the single project makespan. From the global perspective, when the resource priority rule (HL&LN) is determined, LFT-HL&LN and SLFT-HL&LN have the best performance for all-size instances, especially for large-size instances. However, LST-HL&LN and SLST-HL&LN only perform well for small-size instances. Therefore, as a global decision-maker, it is better to choose the LFT or SLFT as the activity priority rules and the HL&LN as the resource priority rules, respectively.

D. PERFORMANCE ON THE TSA-12 PRs COMPARED WITH OTHER ALGORITHMS

In order to verify the performance of TSA-12 PRs, our comparative experiments mainly focus on two aspects: (1) comparing the distributed methods on 12 heuristic priority rules; (2) comparing the distributed and centralized methods on different size instances.

The first comparative experiment aims to evaluate the effectiveness of the VNS compared to other distributed methods for 12 heuristic priority rules, specifically the sequential game-based negotiation mechanism (SGNM).

SGNM is a traversal search method where all conflicting projects are arranged, and the best project sequence is selected as the execution order at the conflicting time. Li and Xu [49] introduced that SGNM had an excellent performance in solving general distributed scheduling problems. There is no literature on MS-SDRCMPSP with uncertain activity duration, so we applied SGNM to MS-SDRCMPSP with certain representativeness. The detailed SGNM is described as follows:

SGNM: A sequential game-based negotiation mechanism based on the distributed multi-agent system, introduced in [49]. The sequential game is defined as a game consisting of finite and at least two players where each player takes actions at different times or in turn. CA as a coordinator organizes sequential games for PAs. After several sequential game negotiations, CA determines the best subgame perfect Nash equilibrium. Then multiple PAs resolve their local schedules with the allocated global resources from CA independently.

Table 10, 11, 12, and 13 denote the comparison results between our method and SGNM on 12 priority rules for different instances. Since SGNM is also combined with a two-stage distributed approach, its role as VNS is to decide the project execution order in the global coordination press. In order to test the impact of VNS and SGNM on results, we ensure that each activity PR and resource PR as a combination and 12 PRs are tested. The gap means the improved percentage of VNS than SGNM in each priority rule. "+" denotes that VNS is better than SGNM. "−" means that SGNM is better than VNS.

TABLE 10. Comparisons of $E(TTC)$ with the distributed approach of $MP30\_2$.

Priority rules	VNS	SGNM	Gap
LST-HL&LN	3023.67	2987.45	-1.21%
LST-HAL	3154.35	3129.34	-0.80%
LST-LN&HL	3320.17	3412.17	+2.77%
LFT-HL&LN	3016.71	3055.65	+1.29%
LFT-HAL	3142.02	3085.68	-1.83%
LFT-LN&HL	3305.43	3351.26	+1.39%
SLST-HL&LN	3036.23	3096.52	+2.0%
SLST-HAL	3154.23	3116.59	-1.21%
SLST-LN&HL	3321.27	3356.36	+1.06%
SLFT-HL&LN	3010.11	2960.61	-1.67%
SLFT-HAL	3150.98	3054.19	-3.17%
SLFT-LN&HL	3303.45	3345.38	+1.27%

TABLE 11. Comparisons of $E(TTC)$ with the distributed approach of $MP90\_2$.

Priority rules	VNS	SGNM	Gap
LST-HL&LN	11039.85	10788.72	-2.33%
LST-HAL	11633.53	12158.23	+4.51%
LST-LN&HL	12212.18	12732.01	+4.26%
LFT-HL&LN	10896.86	10419.79	-4.58%
LFT-HAL	11603.21	11419.45	-1.61%
LFT-LN&HL	12199.64	12519.25	+2.62%
SLST-HL&LN	11002.09	11358.57	+3.24%
SLST-HAL	11559.88	11038.18	-4.73%
SLST-LN&HL	12190.56	12301.42	+0.91%
SLFT-HL&LN	10933.46	10621.91	-2.93%
SLFT-HAL	11491.68	10981.35	-4.65%
SLFT-LN&HL	12165.7	12561.54	+3.25%

TABLE 12. Comparisons of $E(TTC)$ with the distributed approach of $MP30\_5$.

Priority rules	VNS	SGNM	Gap
LST-HL&LN	5812.9	6462.83	10.56%
LST-HAL	6206.45	6850.23	9.40%
LST-LN&HL	6168.32	6797.36	9.25%
LFT-HL&LN	5795.91	6425.35	9.80%
LFT-HAL	6213.62	6868.51	9.53%
LFT-LN&HL	6146.85	6758.39	9.05%
SLST-HL&LN	5728.53	6358.09	9.90%
SLST-HAL	6164.77	6778.46	9.05%
SLST-LN&HL	6089.65	6658.05	8.54%
SLFT-HL&LN	5695.69	6313.95	9.79%
SLFT-HAL	6176.07	6792.63	9.08%
SLFT-LN&HL	6012.35	6591.28	8.78%

TABLE 13. Comparisons of $E(TTC)$ with the distributed approach of $MP90\_5$.

Priority rules	VNS	SGNM	Gap
LST-HL&LN	33884.29	39559.09	14.35%
LST-HAL	36037.3	41079.32	12.27%
LST-LN&HL	36372.31	41359.05	12.06%
LFT-HL&LN	33690.7	39259.93	14.19%
LFT-HAL	36330.81	41319.67	12.07%
LFT-LN&HL	36218.95	41286.28	12.27%
SLST-HL&LN	33735.25	39359.27	14.29%
SLST-HAL	35851.27	41229.25	13.04%
SLST-LN&HL	36178.25	42758.25	15.39%
SLFT-HL&LN	33623.21	39108.93	14.03%
SLFT-HAL	35720.23	41108.37	13.11%
SLFT-LN&HL	36021.69	42017.82	14.27%

Table 10 and 11 show no apparent difference between the two projects for our method and SGNM. When there are only two projects, there are at most two sequences

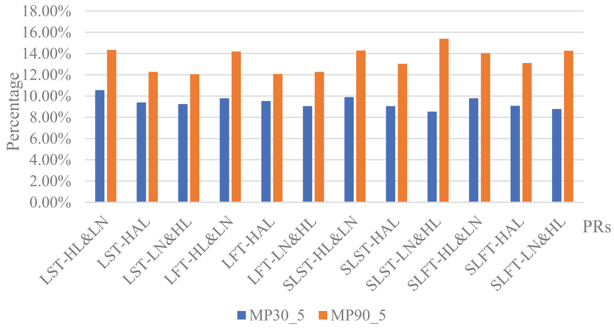


FIGURE 6. The improved percentage of the solutions on different instances.

at each moment, such as project 1-project 2 or project 2-project 1. If the sequence of random selection by SGNM is the same as the sequence finally selected by VNS at the decision time, the obtained objective refers to the same. If the sequence of random selection by SGNM is opposite to the VNS, the results of the two methods are different at this moment. We also find that although the results from VNS and SGNM are not the same, the deviation is controlled within 5%, which is possible. Based on the comparison in Fig. 6, it appears that the VNS method performs better than SGNM, with an advantage of 8% on $MP30_5$and over 12% on $MP90\_5$. After analyzing the priority rules of VNS in each table, it was found that LFT-HL&LN performed best in $MP30\_2$ and $MP90\_2$, while SLFT-HL&LN performed best in $MP30\_5$ and $MP90\_5$.

The second comparative experiment aims to compare the performance of the distributed methods with a centralized method, which in this case is the BRKGA algorithm. The BRKGA algorithm was originally proposed by Almeida et al. [50] for the project scheduling problem with flexible resources. Therefore, it is reasonable to test the BRKGA algorithm on our instances. The detailed BRKGA is as follows:

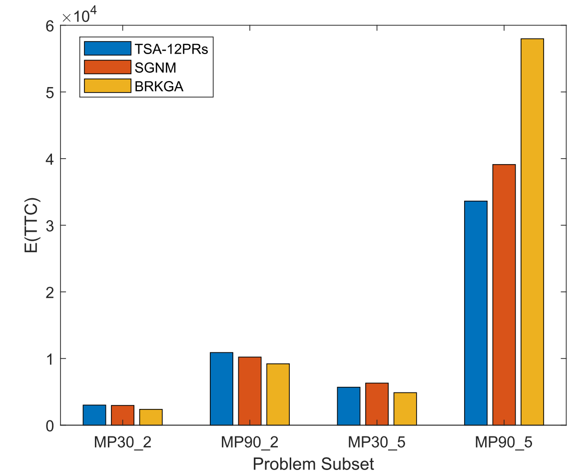
BRKGA: This centralized approach is a based random-key genetic algorithm (BRKGA) for the project scheduling problem with flexible resources [50]. In BRKGA, there are three key parameters: the population size (pop), the crossover rate(rc) and the mutation rate(rm). MP90_2 is a medium-sized example in this paper, so we take it as an example to set up a pre-experiment and select appropriate parameters for BRKGA. According to the $pop = 60/100$, $rc = 0.7/0.8/0.9$, $rm = 0.05/0.1/0.15$, *different objetives are obtained. Firstly, the Kolmogorov-Smirnov(K-S) test is used to verify the sample data* $P = 0.000(< 0.05)$, *and that does not obey the normal distributionso this article uses a non-parametric test. Then, the Wilcoxon signed rank-sum test is applied since there are two kinds of population sizes. The result is* $P = 0.000(<0.05)$. *Therefore, at a significance level of 5%, the value of population size has a significant impact. Finally, this article applies the Friedman test to multiple related samples of mutation probability between the different population sizes, and the results are shown in Table 14.*

TABLE 14. Friedman test under *pm* grouping.

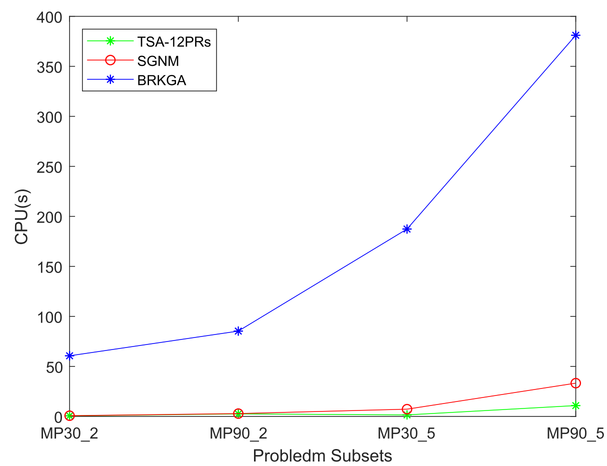
rm	$pop = 60$	$pop = 100$
0.05	0.004(<0.05)	0.217
0.10	0.069	0.026(< 0.05)
0.15	0.34	0.124

TABLE 15. Comparisons of $E(TTC)$ for best priority rules among different approaches.

Problem	TSA-12PRs		SGNM		BRKGA	
Subsets	$E(TTC_1)$	CPU_1(s)	$E(TTC_2)$	CPU_2(s)	$E(TTC_3)$	CPU_3(s)
$MP30\_2$	3016.71	0.586	2960.61	0.694	**2374.52**	60.597
$MP90\_2$	10896.86	2.495	10219.79	2.795	**9209.72**	85.287
$MP30\_5$	5695.69	1.495	6313.95	7.232	**4880.64**	187.331
$MP90\_5$	**33623.21**	10.824	39108.93	33.234	57978.48	381.055



(a) Average global objective



(b) Average CPU time

FIGURE 7. Average CPU time obtained by three approaches.

The bolded part of Table 14 indicates that there is a significant impact when pop $= 60$, *rm* $= 0.05$ *and pop* $= 100$, *rm* $= 0.1$ *are at the 5% significance level. Therefore, this article compares the target values under the three crossover probabilities and selects the combination with the smallest target value as pop* $= 100$, *rc* $= 0.8$, *and rm* $= 0.1$. *Combined with the CPU runtime, the maximum number of iterations as Gen* $= 100$.

SGNM and BRKGA stand for the distributed method and the centralized algorithm, respectively. Table 15 shows the compared results among three algorithms on best priority rules for different problem subsets.

Table 15 and Fig 7 (a) shows that TSA-12PRs has greater difference than other two approaches on *MP*90_5. And we also find that BRKGA performs well on *MP*30_2, *MP*90_2 and *MP*30_5, but the advantages are not obvious. Therefore, TSA-12PRs is more suitable for large-size instances, while BRKGA is more suitable for small-size instances. Then SGNM performs well on *MP*30_2 and *MP*90_2, but SGNM has only a slight advantage over TSA-12PRs and only performs better on the instances with 2 projects. We can see that the number of projects is more important to SGNM than the problem size. Fig.7(b) shows that BRKGA needs more CPU runtime, which is not applicable to the actual situation. When there are more projects, SGNM has no advantage, such as five projects.

It appears that the distributed method, particularly TSA-12PRs, outperforms the centralized method for large-size instances, while the centralized method is more suitable for small-size instances. Additionally, the proposed TSA-12PRs takes less time than the centralized method, making it a promising approach for addressing the MS-SDRCMPSP with uncertain activity duration.

VI. CONCLUSION

This article investigates the stochastic distributed resource-constrained multi-project scheduling problem with the multi-skilled staff. A two-stage approach with 12 priority rules is developed for this problem. In the local scheduling stage, 4 activity priority rules are applied to optimize the expected project makespan; in the global decision stage, 3 resource priority rules are designed to achieve the expected total tardiness cost. In order to confirm the performance of our approach, different size multi-skilled instances are solved. The experimental results show that the two best PRs, including LFT-HL&LN and SLFT-HL&LN, perform better than other PRs on all-size instances. When the two best PRs with the two-stage algorithm are selected on all-size instances, our approach performs better than others(SGNM), especially for large-size instances. Additionally, further experiments show that the centralized approach (BRKGA) is suitable for small-size instances, but the CPU runtime of our method is within the controllable range. Therefore, managers can consider our method if there are more projects in practical project management and the processing time is limited.

Since our method is only applicable to large instances, we plan to design a more effective distributed coordination method that can be applied to instances of all sizes in the future. Additionally, since it is inevitable that multi-skilled staff members will leave and return during a project, managers may need to reallocate these staff members to work on multiple projects. Therefore, future research aims to solve the distributed resource-constrained multi-project scheduling with multi-skilled staff through rescheduling in an uncertain environment.

ACKNOWLEDGMENT

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

REFERENCES

[1] A. Lova and P. Tormos, "Analysis of scheduling schemes and heuristic rules performance in resource-constrained multiproject scheduling," *Ann. Oper. Res.*, vol. 102, no. 1, pp. 263–286, 2001.
[2] M. Hendriks, B. Voeten, and L. Kroep, "Human resource allocation in a multi-project R&D environment: Resource capacity allocation and project portfolio planning in practice," *Int. J. Project Manag.*, vol. 17, no. 3, pp. 181–188, 1999.
[3] X. Shen, L. L. Minku, R. Bahsoon, and X. Yao, "Dynamic software project scheduling through a proactive-rescheduling method," *IEEE Trans. Softw. Eng.*, vol. 42, no. 7, pp. 658–686, Jul. 2016.
[4] 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.
[5] 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.
[6] 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.
[7] 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.
[8] O. Bellenguez and E. Neron, "Lower bounds for the multi-skill project scheduling problem with hierarchical levels of skills," in *Proc. Int. Conf. Pract. Theory Automated Timetabling*. Cham, Switzerland: Springer, 2004, pp. 229–243.
[9] 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, Apr. 2007.
[10] 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.
[11] P. B. Myszkowski, M. E. Skowroński, Ł. P. Olech, and P. O. Oślizlo, "Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem," *Soft Comput.*, vol. 19, no. 12, pp. 3599–3619, 2015.
[12] Y. Kadrou and N. Najid, "Tabu search for the multi-mode resource constarined project scheduling problem whith resource flexibility," in *Proc. 19th Int. Conf. Prod. Res.*, 2007, pp. 1–7.
[13] H. Kazemipoor, R. Tavakkoli-Moghaddam, and P. Shahnazari-Shahrezaei, "Solving a mixed-integer linear programming model for a multi-skilled project scheduling problem by simulated annealing," *Manag. Sci. Lett.*, vol. 2, no. 2, pp. 681–688, Apr. 2012.
[14] H. Kazemipoor, P. Shahnazari-Shahrezaei, and R. Tavakkoli-Moghaddam, "Solving a novel multi-skilled project scheduling model by scatter search," *South Afr. J. Ind. Eng.*, vol. 24, no. 1, pp. 121–135, 2013.
[15] L. L. Minku and X. Yao, "Software effort estimation as a multiobjective learning problem," *ACM Trans. Softw. Eng. Methodol.*, vol. 22, no. 4, pp. 1–32, Oct. 2013.
[16] H. Maghsoudlou, B. Afshar-Nadjafi, and S. T. A. Niaki, "Multi-skilled project scheduling with level-dependent rework risk; Three multi-objective mechanisms based on cuckoo search," *Appl. Soft Comput.*, vol. 54, pp. 46–61, May 2017.
[17] C. Heimerl and R. Kolisch, "Work assignment to and qualification of multi-skilled human resources under knowledge depreciation and company skill level targets," *Int. J. Prod. Res.*, vol. 48, no. 13, pp. 3759–3781, Jul. 2010.
[18] 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.
[19] R. Chen, C. Liang, D. Gu, and J. Y.-T. Leung, "A multi-objective model for multi-project scheduling and multi-skilled staff assignment for IT product development considering competency evolution," *Int. J. Prod. Res.*, vol. 55, no. 21, pp. 6207–6234, Nov. 2017.

[20] M.-C. Wu and S.-H. Sun, "A project scheduling and staff assignment model considering learning effect," *Int. J. Adv. Manuf. Technol.*, vol. 28, nos. 11–12, pp. 1190–1195, May 2006.
[21] W. J. Gutjahr, S. Katzensteiner, P. Reiter, C. Stummer, and M. Denk, "Competence-driven project portfolio selection, scheduling and staff assignment," *Central Eur. J. Oper. Res.*, vol. 16, pp. 281–306, Apr. 2008.
[22] W. J. Gutjahr, S. Katzensteiner, P. Reiter, C. Stummer, and M. Denk, "Multi-objective decision analysis for competence-oriented project portfolio selection," *Eur. J. Oper. Res.*, vol. 205, no. 3, pp. 670–679, Sep. 2010.
[23] Y. H. Lee, S. R. T. 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.
[24] 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.
[25] J. S. K. Lau, G. Q. Huang, K. L. Mak, and L. Liang, "Distributed project scheduling with information sharing in supply chains: Part II—Theoretical analysis and computational study," *Int. J. Prod. Res.*, vol. 43, no. 23, pp. 4899–4927, Dec. 2005.
[26] 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.
[27] 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, Nov. 2007.
[28] J. Homberger, "A (μ, λ)-coordination mechanism for agent-based multi-project scheduling," *OR Spectr.*, vol. 34, no. 1, pp. 107–132, Jan. 2012.
[29] 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.
[30] Q. Wang, C. Liu, and L. Zheng, "A column-generation-based algorithm for a resource-constrained project scheduling problem with a fractional shared resource," *Eng. Optim.*, vol. 52, no. 5, pp. 798–816, May 2020.
[31] M. Rostami and M. Bagherpour, "A Lagrangian relaxation algorithm for facility location of resource-constrained decentralized multi-project scheduling problems," *Oper. Res.*, vol. 20, no. 2, pp. 857–897, Jun. 2020.
[32] 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.
[33] Z. T. Kosztyán, "An exact algorithm for the flexible multilevel project scheduling problem," *Exp. Syst. Appl.*, vol. 158, Nov. 2020, Art. no. 113485.
[34] E. B. Tirkolaee, I. Mahdavi, M. M. S. Esfahani, and G.-W. Weber, "A hybrid augmented ant colony optimization for the multi-trip capacitated arc routing problem under fuzzy demands for urban solid waste management," *Waste Manag. Res.*, vol. 38, no. 2, pp. 156–172, Feb. 2020.
[35] E. B. Tirkolaee, A. Mardani, Z. Dashtian, M. Soltani, and G.-W. Weber, "A novel hybrid method using fuzzy decision making and multi-objective programming for sustainable-reliable supplier selection in two-echelon supply chain design," *J. Cleaner Prod.*, vol. 250, Mar. 2020, Art. no. 119517.
[36] A. Mahdavi, M. A. E. Moghaddam, and A. Mahmoudi, "Simultaneous charging and discharging of multi-tube heat storage systems using copper fins and Cu nanoparticles," *Case Stud. Thermal Eng.*, vol. 27, Oct. 2021, Art. no. 101343.
[37] H. Chen, G. Ding, S. Qin, and J. Zhang, "A hyper-heuristic based ensemble genetic programming approach for stochastic resource constrained project scheduling problem," *Exp. Syst. Appl.*, vol. 167, Apr. 2021, Art. no. 114174.
[38] F. Zaman, S. Elsayed, R. Sarker, D. Essam, and C. A. C. Coello, "An evolutionary approach for resource constrained project scheduling with uncertain changes," *Comput. Oper. Res.*, vol. 125, Jan. 2021, Art. no. 105104.
[39] 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.
[40] B. Ashtiani, R. Leus, and M.-B. Aryanezhad, "New competitive results for the stochastic resource-constrained project scheduling problem: Exploring the benefits of pre-processing," *J. Scheduling*, vol. 14, no. 2, pp. 157–171, Apr. 2011.
[41] F. Ballestín, "When it is worthwhile to work with the stochastic RCPSP?" *J. Scheduling*, vol. 10, no. 3, pp. 153–166, Jun. 2007.
[42] S. Rostami, S. Creemers, and R. Leus, "New strategies for stochastic resource-constrained project scheduling," *J. Scheduling*, vol. 21, no. 3, pp. 349–365, Jun. 2018.
[43] F. F. Boctor, "Some efficient multi-heuristic procedures for resource-constrained project scheduling," *Eur. J. Oper. Res.*, vol. 49, no. 1, pp. 3–13, Nov. 1990.
[44] R. Kolisch, "Efficient priority rules for the resource-constrained project scheduling problem," *J. Oper. Manag.*, vol. 14, no. 3, pp. 179–192, Sep. 1996.
[45] Z. Chen, E. Demeulemeester, S. Bai, and Y. Guo, "Efficient priority rules for the stochastic resource-constrained project scheduling problem," *Eur. J. Oper. Res.*, vol. 270, no. 3, pp. 957–967, Apr. 2018.
[46] C. Fang, R. Kolisch, L. Wang, and C. Mu, "An estimation of distribution algorithm and new computational results for the stochastic resource-constrained project scheduling problem," *Flexible Services Manuf. J.*, vol. 27, no. 4, pp. 585–605, Dec. 2015.
[47] M. Wang, G. Liu, and X. Lin, "Dynamic optimization of the multi-skilled resource-constrained project scheduling problem with uncertainty in resource availability," *Mathematics*, vol. 10, no. 17, p. 3070, Aug. 2022.
[48] 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.
[49] 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.
[50] 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, Jul. 2018.

YINING YU was born in Qingdao, Shandong, China, in 1994. She received the B.S. degree in mathematics from Qufu Normal University, Shandong, in 2016, and the M.S. degree in mathematics from Beihang University, Beijing, China, in 2018, where she is currently pursuing the Ph.D. degree. Her research interests include project scheduling, optimization algorithm, and risk management.

ZHE XU received the Ph.D. degree from the Department of Economics and Management, Beihang University, Beijing, China, in 2006. Since 2009, she has been a Professor with the Department of Economics and Management, Beihang University. Her current 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 the China University of Mining and Technology, Beijing, China, in 2014, and the M.S. degree in mathematics from Beihang University, Beijing, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include project scheduling, optimization algorithm, and risk management.