

Received April 18, 2022, accepted May 4, 2022, date of publication May 16, 2022, date of current version May 20, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3175312

# Bi-Objective Reactive Project Scheduling Problem Under Resource Uncertainty and Its Heuristic Solution Based on Priority Rules

HAN SONG<sup>1</sup>, GUANGMIN JIA<sup>2</sup>, AND WULIANG PENG<sup>1</sup> 

<sup>1</sup>School of Economics and Management, Yantai University, Yantai 264005, China

<sup>2</sup>School of Foreign Languages, Yantai University, Yantai 264005, China

Corresponding author: Wuliang Peng (wuliang.p@ytu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 71671117.


**ABSTRACT** In the process of project implementation, the deviation usually exists between the actual scheduling process and the predefined schedule. A typical situation is that the project cannot be executed according to the generated schedule due to the uncertainty of resource supply. In this case, it is necessary to adjust the schedule to achieve the lowest possible cost and the shortest possible duration. This paper studies the reactive scheduling problem under the condition of resource uncertainty. Under the constraints of precedence relationship and resources, a novel bi-objective reactive project scheduling problem with the objectives of minimizing project duration and minimizing adjustment cost is proposed. Besides, the regret-based biased random sampling heuristic algorithm (RBRSA), a heuristic algorithm based on multiple priority rules, is presented to solve the presented bi-objective reactive scheduling problem. In addition, the advantages and disadvantages of the priority rules thus retained are compared. Moreover, we further investigate the combinations of different parameter values of RBRSA and schedule generating schemes (SGS). When the parameter value is set at an appropriate value, the priority rules LST and LFT are the best two in RBRSA if the serial scheduling generation scheme (SSGS) is adopted. When the priority rules of LST or LFT are adopted in RBRSA, the parameter values of RBRSA should be set at other values if the parallel scheduling generation scheme (PSGS) is adopted. Computational experiments show that the presented bi-objective reactive project scheduling method is efficient and practical.

**INDEX TERMS** Project scheduling, reactive scheduling, heuristic algorithm, multi-objective optimization.

## I. INTRODUCTION

In the process of project execution, the actual implementation is hardly consistent with the baseline schedule generated before the project starts. Due to the interference of various external uncertain factors, deviations frequently occur between the actual execution processes and the baseline schedule. With the development of project management, we have found that the proposed method can be more practical and closer to the actual situation of project execution when considering deviations. In recent years, therefore, the uncertain resource-constrained project scheduling problem has attracted widespread attention [1], which usually includes two types: 1) the resource-constrained project

proactive scheduling problem and 2) the resource-constrained project reactive scheduling problem (RCPRSP). To address the former, a robust baseline schedule is generated prior to the start of the project, taking into account the numerous uncertain factors that may arise. Therefore, the baseline schedule does not need to be modified or amended as little as possible when a disturbance occurs. But for the latter, the baseline schedule that has been generated by the project managers must be adjusted in the implementation with some nonnegligible interference factors [2]. Consequently, proactive and reactive scheduling occur at different stages of the project. Proactive scheduling occurs before the project starts to obtain a more stable baseline schedule. Reactive scheduling occurs in the process of project implementation because of its quick response to interference. However, in the practice of project management, even if the proactive scheduling generates a

The associate editor coordinating the review of this manuscript and approving it for publication was Shih-Wei Lin .

highly robust baseline schedule considering various possible uncertainties in advance, project leaders may still be unable to implement the baseline schedule as expected due to various disturbances. In this paper, the above dilemma can be solved by reactive scheduling because of its quick response ability. Therefore, as an important control tool in project implementation, reactive scheduling has broad application prospects.

At present, the reactive scheduling can be divided into two categories. One is the repair scheduling, which refers to the method that can quickly restore scheduling in the case of external interference. Although the right shift method is the simplest repair scheduling method for its intuition, disadvantages still can be found to some extent, and thus the right shift method cannot meet the expectations. The other is the pure rescheduling, that is, when the external disturbance factors make the baseline schedule infeasible, the remaining unfinished activities after the disturbance time will be entirely rescheduled. Compared to proactive scheduling, there are relatively few studies on reactive scheduling. Nevertheless, in recent years, a host of pure rescheduling strategies or methods have been proposed, such as the minimum interference strategy, the scheduling method based on priority rules, and the minimum delay costs [3]. Furthermore, reactive scheduling can be noticed in many scheduling territories, such as manufacturing [4], machine scheduling [5], and virtual cloud scheduling [6]. With the increasing development of management, a variety of fundamental methods for RCPRSP have been put forward by researchers. From the available literature, the studies of RCPRSP concentrate on constructing models for different application scenarios and developing an efficient solution. In the existing studies on RCPRSP, many algorithms have been designed by researchers to solve the reactive scheduling models which have been constructed to reach a maximum or minimum objective function under the condition of satisfying the constraints of resources and precedence relationship, including accurate algorithms [7], heuristic algorithms [8], and intelligent optimization algorithms [9], [10].

The minimum interference strategy indicates that the key of RCPRSP is how to achieve “ex-post stability” of the scheduling. That is, the total deviation between the regenerated schedule and the baseline schedule is the lowest [3]. Most existing literature employs the adjustment cost of schedule start time ( $\sum_{i \in N} \omega_i |s_i^q - s_i^b|$ ) to represent this total deviation, for an activity  $i \in N$ ,  $\omega_i$  is the weight,  $s_i^b$  is the start time in the baseline schedule,  $s_i^q$  is the start time in the schedule regenerated after time  $q$  when the availability of resources varies. Van de Vonder *et al.* [10], [11] studied the single-mode proactive-reactive scheduling problem with the objective of minimizing the total weighted deviation of the activity start time between the baseline schedule and the regenerated schedule when activities were interrupted. Besides, the robust parallel and serial scheduling generation schemes were proposed to repair or re-optimize the baseline scheduling. In order to maximize the stability of baseline scheduling under the continuous change of activity duration,

Deblaere *et al.* [12] proposed an effective schedule repair procedure to minimize the cost of schedule stability (defined as the adjustment cost of schedule start time), which saves various types of cost produced by the delay in the start time of each activity. Lambrechts *et al.* [2] focused on the reactive scheduling problem in the case of resource interruption interference with the objective of minimizing the adjustment cost, and they argued that the combination of the protection of the baseline scheduling and intelligent repair scheduling can effectively improve the scheduling performance. Delgoshaei *et al.* [13] concentrated on multi-mode resource constraint project scheduling problems (MRCPPSPs) with preemptive resources to maximize the net present value of the MRCPPSPs, and they presented a new forward programming heuristic for modifying an over-allocated schedule. Although most objectives of RCPRSP are to minimize the adjustment cost of the activity times, there are some objectives of RCPRSP that consider both the adjustment cost of the activity times and other costs. Kuster *et al.* [14] put forward a new local rescheduling method for scheduling suspended in actual management by considering the weighted sum of project delay cost, activity execution cost, and adjustment cost of the activity times. Considering the project duration in scheduling targets, Chakraborty *et al.* [15] restored interrupted schedules by exploring the RCPRSP in two utility scenarios: “preemption-repetition” and “preemption-recovery”. Wang *et al.* [9] constructed a reactive scheduling sub-model for the reactive scheduling problem of resource disruption that included the additional cost and decreased cost caused by the variation of activity duration in the scheduling objective. Considering the activity modes, time buffers, and activity start times, Ning *et al.* [16] aimed to minimize the contractor’s maximal cumulative gap between cash outflows and cash inflows to generate a robust baseline schedule. In addition, Elloumi *et al.* [17] addressed the multi-objective reactive project scheduling problem under different resource requirements and activity durations and took the minimum project duration and the lowest adjustment cost as the two scheduling objectives. Tirkolaee *et al.* [18] developed a non-linear programming model to address the multi-objective multi-mode resource-constrained project scheduling problem in payment planning. Also, it adopted NPV maximization and project duration minimization as the scheduling aims, considering renewable resources and non-renewable resources. From the above mentioned, the current studies of RCPRSP mainly focus on the single-objective problem when considering the change in activity duration, while studies on the multi-objective RCPRSP considering resource uncertainty is still relatively scarce.

To solve RCPRSP, many algorithms have been proposed in the existing literature, considering the above uncertainties and deficiencies. These algorithms include exact algorithms, heuristic algorithms, intelligent optimization algorithms, and so on. Besides, extensive studies on the algorithms has been conducted for solving the single-objective RCPRSP with uncertainty of activity duration in the process of project

execution. Deblaere *et al.* [12] proposed three heuristic algorithms based on integer programming which can be used to obtain the lower bound of scheduling stability cost and provide an effective progress recovery schedule. Later, Deblaere *et al.* [7] introduced the branch and bound algorithm and the iterative deepening algorithm (IDA). The IDA is a tree-based optimal search technology related to optimal priority search. These two algorithms improved the stability of the baseline schedule by creating a new reactive schedule for which the authors combined the advantages of IDA and other algorithms. Elloumi *et al.* [8] designed level heuristic (LH), hurried slipping window method (HSWM) and lexicographic regret based on heuristic (LRBH) algorithms to provide a feasible schedule. It is proved that the LRBH algorithm is the best than the LH and HSWM in the aspects of generating feasible solutions and reducing deviations. In an effort to select and determine the optimal scheduling strategy, Godinho and Branco [19] developed an electromagnetic heuristic algorithm by putting forward a self-adaptive model. Furthermore, Rahman *et al.* [20] presented a real-time reactive meta-heuristic algorithm equipped with IGFBI (defined as IGFBI for this algorithm uses a forward-backward improvement strategy to improve the computing ability of an intelligent genetic algorithm), which can be conducive to conducting the posterior evaluation of the project progress. And it has been verified that this method is superior to the right-shift method. In order to solve the RCPRSP under resource uncertainty in project progress, Lambrechts *et al.* [2] constructed the model by transforming the uncertainty into the resource availability affected by unforeseen faults. And they put forward a tabu search heuristic algorithm based on priority rules to minimize the instability cost and generate a stable schedule. Chakraborty *et al.* [21] suggested two algorithms in the scenarios of “preemption-recovery” and “preemption-repeat” for dealing with the mixed integer programming models of RCPRSP. Chakraborty *et al.* [22] proposed an enhanced iterative greedy algorithm that can effectively solve benchmark instances with more activities, and they developed an event based reactive approach (EBRA), which has proved effective in reducing computational complexity. In response to other uncertainties, Suwa *et al.* [23] presented a “degree of project progress” reactive project scheduling method. And it is shown that the generation and implementation of the revision schedule are more effective than both the event-driven decision method and the periodic rescheduling method from the perspectives of performance measurement and revision cost. In multi-objective scheduling, Zhang *et al.* [24] concentrated on the uncertainty of scheduling plan adjustment caused by the failure of renewable resources, duration delay, and random insertion of new projects. Furthermore, to address the bi-objective reactive scheduling problem, the multi-objective particle swarm optimization algorithm was presented with two objectives, namely, minimizing the total loss caused by the start time delay and minimizing the sum of the weighted project duration. Elloumi *et al.* [17] proposed the careful slipping window method (CSWM) and the multi-objective evolu-

tionary algorithm (MOEA) to solve the bi-objective RCPRSP for minimizing both the project duration and the interruption measure in the situation of mode modification or interruption. Delgoshaei *et al.* [25] developed a multi-objective non-linear mixed integer programming model and designed a multi-objective weighting genetic algorithm with scarce resources to maximize the profit, minimize scheduling costs, and minimize the project duration. From the existing studies on RCPRSP, it can be concluded that all the meta-heuristic algorithms have been employed to solve RCPRSPs since they belong to the NP-Hard problem.

Extensive studies on RCPRSP have been conducted by researchers. The traditional scheduling objective of minimizing project duration has been changed to minimize the adjustment cost of the schedule, and the adjustment cost gradually includes several aspects. Additionally, in order to meet managerial expectations, the multi-objective RCPRSPs have attracted great interest. In terms of algorithms, for solving RCPRSP, a variety of intelligent optimization algorithms have been effectively applied. In the current studies on RCPRSPs, however, there are the following problems: 1) The existing studies mainly consider the uncertainty of project duration while the studies on resource uncertainty are still relatively scarce; 2) Although the existing studies have transformed to the single objective of the scheduling adjustment cost, the original objectives of baseline scheduling, i.e., minimizing project duration, are often ignored. Besides, the research on bi-objective RCPRSP that minimizes project duration and adjustment cost has not been reported; 3) In the available algorithm research, meta-heuristic algorithms play a dominant role, but the research on more practical heuristic algorithms based on priority rules is relatively deficient.

In summary, resource uncertainty is one of the main risks in the process of project execution. Uncertainties in core staff, tool damage, equipment failure, and other factors may make it impossible to implement the baseline schedule in accordance with expectations. In this case, it is a key issue to improve project scheduling performance to maintain the optimal or near optimal project scheduling objective with the lowest adjustment cost. In this paper, a novel bi-objective reactive project scheduling problem considering resource uncertainty is proposed to address the resource uncertainty. It aims to achieve the goals of the minimum adjustment cost and the minimum project duration under the constraints of the precedence relationships and resources. A regret-based biased random sampling heuristic algorithm, which employs 19 priority rules, is adopted to solve the problem proposed in this paper.

## II. PROBLEM DESCRIPTION AND MODEL CONSTRUCTION

### A. PROBLEM DESCRIPTION

The single-mode resource constrained project scheduling provides the baseline schedule for RCPRSP. The project is represented by Activity-on-Arrows (AOA), namely  $G = (N, A)$ , where  $N = \{0, 1, 2, 3, \dots, n, n + 1\}$  is a set of activities. The dummy activities are 0 and  $n + 1$  in  $N$ , which do

not have duration or resources, representing the start and end of the project.  $A$  is a set of arcs that indicate the finish-start (FS) constraint relations between activities.  $(i, j) \in A$  denotes that  $j$  can only be started after an activity  $i$  is finished. The preceding activity set, belonging to the activity  $i$ , is indicated by  $P_i$ . In terms of resources, this paper exclusively considers the constraints of renewable resources, with  $K$  denoting the set of different types of renewable resources. After that,  $k = 1, 2, \dots, |K| - 1, |K|, k \in K$ . Furthermore,  $R_k$  is the amount of renewable resource  $k$ . In the traditional RCPS, the availability of any resource does not vary during the project execution. While in RCPRSP, each resource amount ( $R_k$ ) will change. Besides, the activity  $i$ , with  $d_i$  as the duration, requires  $r_{ik}$  for renewable resource  $k$ . In this paper, the baseline schedule  $\Pi^b = (S^b, F^b)$  is based on the assumption that all project parameters are constant and that all activities are subject to the constraints of resources and precedence relationship. In addition, in the baseline schedule, the set of start times for all activities is  $S^b = (s_0^b, s_1^b, \dots, s_{n+1}^b)$ , and the set of finish times for all activities is  $F^b = (f_0^b, f_1^b, \dots, f_{n+1}^b)$ . The upper bound of the project duration is  $\bar{D}$ . Chen et al. [26] have verified that the LFT is one of the best priority rules to solve resource-constrained random project scheduling problems. Therefore, LFT is selected to generate the baseline schedule by the authors.

Because the project execution environment is believed to be deterministic in the traditional RCPS, most studies aim to minimize the project duration under the constraints of resources and precedence relationship. The research on RCPRSP, on the other hand, considers that the project execution environment contains uncertain factors. For example, the variety of activity durations and resource supply. Therefore, this paper focuses on the reactive scheduling problem when each resource amount changes after time  $q$ . That is, managers need to reschedule the activities that have not started after time  $q$  according to the baseline schedule for each resource amount that varies after time  $q$ . Furthermore, due to the varying resource amounts after time  $q$ , the baseline schedule must be adjusted to make it feasible, or even optimal. At the same time, to avoid meaningless scheduling costs in reactive scheduling, this paper assumes that the activities, finished before time  $q$  or executed at time  $q$ , are implemented according to the baseline schedule. While other activities that have not started until time  $q$  are rescheduled in accordance with the reactive scheduling method. Then, a new rescheduling schedule is produced by adjusting the baseline scheduling. At time  $q$ , we note that the new schedule is  $\Pi^q = (S^q, F^q)$ , the start time set is  $S^q = (s_0^q, s_1^q, \dots, s_{n+1}^q)$  and the finish time set is  $F^q = (f_0^q, f_1^q, \dots, f_{n+1}^q)$ , the completed activity set is denoted by  $C_q$ , the ongoing activity set is denoted by  $A_q$ , the eligible (candidate) activity set is denoted by  $D_q$ , and the uncompleted activity set is denoted by  $U_q$ . If activity  $i \in U_q$ ,  $s_i^b = s_i^q$  and  $f_i^b = f_i^q$ .

Maintaining the stability of the baseline schedule is an important goal for RCPRSP. Therefore, the regenerated schedule must be as close to the baseline schedule as possible.

The rescheduling cost (an adjustment cost) is established to quantify the deviation between the regenerated schedules and the baseline schedule. Besides, we take the sum deviation of the activity start times between the  $\Pi^b$  and the  $\Pi^q$  as the rescheduling cost. To avoid producing repetitive scheduling cost, it is only the unscheduled activities after time  $q$  that can generate rescheduling cost. That is, when  $i \in U_q$ , the rescheduling cost is  $\sum_{i \in U_q} |s_i^q - s_i^b|$ .

In addition, although the study of RCPRSP is multifaceted, minimizing project duration is still the popular objective in most project scheduling. As a result, RCPRSP in this paper refers to unscheduled activities until time  $q$  is rescheduled so as to minimize not only the project duration but also the adjustment cost under the constraints of precedence relationship and resource.

### B. MODEL CONSTRUCTION

According to the description of the problem in the previous section, we construct a bi-objective reactive scheduling problem model for RCPRSP as follows.

$$\text{Min } f_{n+1}^q \tag{1}$$

$$\text{Min } \sum_{i=0}^{n+1} |s_i^q - s_i^b| \tag{2}$$

s.t.

$$s_i^q = s_i^b, f_i^q = f_i^b, \quad i \notin U^q \tag{3}$$

$$s_i^q \geq q, \quad i \in U^q \tag{4}$$

$$f_i^q = s_i^q + d_i, \quad i = 0, 1, \dots, n, n+1 \tag{5}$$

$$s_j^q + d_j \leq s_i^q, \quad i = 0, 1, \dots, n, n+1, \quad j \in P_i \tag{6}$$

$$\sum_{i \in A_t} r_{ik} \leq R_k, \quad k = 1, 2, \dots, |K|, \quad t = 1, 2, \dots, \bar{D} \tag{7}$$

(1) is to minimize the project duration, and (2) is to minimize the adjustment cost, where  $|s_i^q - s_i^b|$  indicates the deviation of the start time of the activity  $i$ . (3) shows that the activity  $i$  is not in the uncompleted activity set and is carried out according to the baseline schedule. (4) indicates that the start time of activity  $i$  in the uncompleted activity set should not be earlier than  $q$ . (5) represents the finish time of activity  $i$ . (6) signifies the constraints of precedence relationship, and (7) is the series of resource constraints, in (7),  $A_t$  denotes a set of activities being executed at time  $t$ , and the total amount of resource  $k$  occupied by the activities in  $A_t$  should be less than the total amount of the resource  $k$ .

In the above model,  $s_i^q, s_i^b, f_i^q$ , and  $f_i^b$  are positive integer decision variables. Furthermore, the reactive scheduling time  $q$  is a positive integer that must be pre-determined.

### III. ALGORITHM DESIGN

The heuristic algorithm based on priority rules (Pr) is widely used in RCPRSP due to its intuitive logic, easy understanding, and fast calculation speed [26], [27]. It combines the SGS, which mainly includes the PSGS and the SSGS, with priority rules to generate feasible project schedules. The difference between the above PSGS and SSGS lies in that PSGS takes time as a stage variable, while SSGS takes activity



as a stage variable [28], [29]. In addition, priority rules are generally used to determine the priority values of activities. Besides, the activity with the higher priority value will be scheduled first when resource conflicts occur. The combination of SGS and Pr mainly constructs two algorithms: a single-channel algorithm and a multi-channel algorithm. The difference between these two algorithms is whether the new schedule can be obtained by repeating the same algorithm. Furthermore, the single-channel algorithm combines one SGS that is equipped with one of the Pr, so as to generate a deterministic project schedule. Even if repeated, no new schedule will be generated. While the multi-channel algorithms mainly include four types: 1) multiple priority rule heuristic algorithm, 2) forward-backward scheduling heuristic algorithm, 3) sampling heuristic algorithm, and 4) adaptive heuristic algorithm [30]. Additionally, the sampling heuristic algorithms will generate a new schedule and select activities from the feasible activity set according to probability. What is more, the RBRSA falls under the category of sampling heuristic algorithms. To solve the RCPRSP proposed in this paper, RBRSA is utilized to merge the two scheduling generation schemes with the Pr. The 19 priority rules in Pr are shown in the Table 1 [28]–[31].

**A. REGRET-BASED BIASED RANDOM SAMPLING ALGORITHM**

The RBRSA is an effective method to solve RCPRSP [32]. In RBRSA, an activity belonging to the eligible activity set is equipped with a selected probability, which is calculated according to the priority value computed by the priority rule. In the traditional random scheduling solution, an activity is selected from the eligible activity set according to the priority value when resource conflicts occur. While in RBRSA, an activity  $j$  is selected from the eligible activity set by the probability. To be more specific, RBRSA uses the following probability assignment function to calculate the selected probability of an activity  $j$ .

$$p(j) = \frac{v''(j)}{\sum_{j' \in D_n} v''(j')} \tag{8}$$

$v'(j)$  is the amended to  $v''(j)$ .

$$v''(j) = (v'(j) + \varepsilon)^\alpha \tag{9}$$

Parameter  $\varepsilon$  ensures that each activity has a probability of being selected, and parameter  $\alpha$  is used to control the randomness of the scheduling process.  $v'(j)$  is the maximum distance between the priority value of activity  $j$  and the maximal or minimal priority value of an activity in the eligible activity set of each scheduling stage (noted as  $D_g$ ).

$$v'(j) = \begin{cases} \max V(D_g) - v(j), & \text{if extr} = \min; \\ v(j) - \min V(D_g), & \text{if extr} = \max. \end{cases} \tag{10}$$

$v(j)$  is the priority value of the activity  $j$ ,  $\max V(D_g)$  is the maximal one of the priority values in  $D_g$ , and  $\min V(D_g)$  is

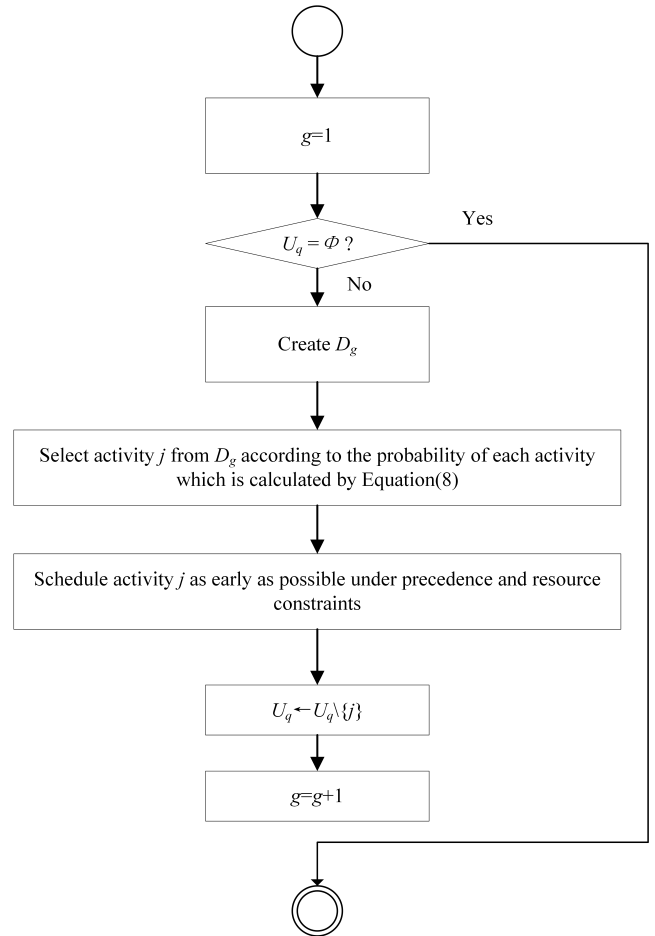


FIGURE 1. RBRSA based on SSGS.

the minimal one. If the adopted priority rule stipulates that the activity with the lower priority value should be preferred,  $v'(j)$  is valued by  $\max V(D_n) - v(j)$ . Otherwise,  $v'(j)$  is valued by  $v(j) - \min V(D_n)$ . Furthermore, the activity  $j$  can be selected from  $D_g$  according to the selected probability of each activity in  $D_g$ , which can be calculated by the above formula (8).

In the RBRSA,  $\varepsilon$  and  $\alpha$  have an impact on the amended value of  $v''(j)$ , which can alter the probability of each activity in  $D_g$ . For parameter  $\varepsilon$ , this paper follows previous studies, and values  $\varepsilon$  by 1 [33]. For parameter  $\alpha$ , Türkakın et al. [30] and Dek [33] found that  $\alpha$  has a significant influence on the performance of heuristic priority rules. Therefore, in the following section of this paper, for parameter  $\alpha$ , we will value  $\alpha$  by different values to investigate the influence of  $\alpha$  on the performance of RBRSA for the presented bi-objective RCPRSP.

The process of RBRSA under SSGS is shown in Fig. 1.

Apart from adopting both SSGS and PSGS to generate schedules, we also take SSGS as an example to illustrate the flow of RBRSA. As is shown in Fig. 1, after setting the priority rule, the number of iterations  $g$  is valued by 1 and  $D_g$  is determined based on the activities that have been scheduled. The activity  $j$  is selected from  $D_g$  according to

TABLE 1. Priority rules.

No.	Priority rule	Formula	Extreme value
(0)	EST(Earliest start time)	$ES_j$	min
(1)	EST(Earliest start time)	$EF_j$	min
(2)	LST(Latest start time)	$LS_j$	min
(3)	LFT(Latest finish time)	$LF_j$	min
(4)	MSLK(Minimum slack)	$LS_j - ES_j$	min
(5)	MFF(Minimum free float)	$\min\{ES_h\} - EF_j, h \in S_j$	min
(6)	MSF(Minimum safety float)	$LS_j - \max\{LF_h\}, h \in P_j$	min
(7)	MIS(Maximum immediate tasks)	$ S_j $	max
(8)	SRD(Smallest total resource demand)	$d_j \cdot \sum_{k=1}^K r_{jk}$	min
(9)	GRD(Great total resource demand)	$d_j \cdot \sum_{k=1}^K r_{jk}$	max
(10)	GRU(Great resource utilization)	$\sum_{k=1}^K r_{jk}$	max
(11)	GRPW(Greatest rank positional weight)	$d_j + \sum_{h \in S_j} d_h$	max
(12)	LFS(Least average float time of successive tasks)	$(LS_j - ES_j)/ S_j^T $	min
(13)	WRUP(Weighted resource utilization and precedence)	$0.7 S_j  + 0.3 \sum_{k=1}^K (r_{jk}/R_k)$	max
(14)	MTS(Most total successors)	$S_j^*$	max
(15)	TRS(Total resource scarcity)	$\sum_{k=1}^K (r_{jk}/R_k)$	min
(16)	SPT(Smallest project time)	$d_j$	min
(17)	LPT(Longest project time)	$d_j$	max
(18)	WACRU(Amend WRUP)	$\omega G_j + (1 - \omega) \sum_{k=1}^K (r_{jk}/R_k)$	max

<sup>1</sup>  $ES_j, EF_j, LS_j, LF_j$  — Activity  $j$  earliest start time, earliest finish time, latest start time, latest finish time.

<sup>2</sup>  $|S_j|$  — Number of all direct immediate activities of activity  $j$ .

<sup>3</sup>  $|S_j^T|$  — Number of all follow-up activities for activity  $j$ .

the selected probability that is obtained by the priority rule. After the activity  $j$  is scheduled,  $g = g + 1$ . The process is repeated until all activities in the  $U_q$  are scheduled. Finally, the project duration is obtained. If the PSGS is adopted in the RBRSA, multiple activities are sequentially selected from  $D_g$  according to selected probability and scheduled by the algorithm. In each iteration, the scheduled activity will be deleted from  $D_g$  in turn until  $D_g$  is empty.

**B. BI-OBJECTIVE RBRSA**

In this paper, we propose a bi-objective RBRSA for solving the bi-objective RCPRSP. Fig. 2 illustrates the overall flow of the bi-objective RBRSA.

Before the scheduling starts, the critical path times of each activity are acquired by the algorithm based on the critical path method (CPM), such as the earliest start time, the latest start time, the earliest finish time, and the latest finish time. All of them are used to calculate the priority values. Then, the baseline schedule is generated by SGS and LFT priority rules. The time  $q$  will be set when each resource amount is changed, and thereby the following  $C_q, A_q,$  and  $U_q$  are obtained. The activity  $i$  is added to  $C_q$  if  $s_i^b < q$ ; to  $A_q$  if  $s_i^b < q < f_i^b$ ; and to  $U_q$  if  $q < s_i^b$ . Afterwards, the maximum number of generation, denoted by Gen, is set. And the set of reactive scheduling result, denoted by RS, is initiated as  $\emptyset$ . Then,  $gen$  is valued by 1, and the RBRSA is called to schedule the activities in  $U_q$ , with the SGS,  $\alpha$ , and priority rule until all the activities in  $U_q$  are scheduled to acquire reactive schedule and the project duration. Following the preceding steps, the rescheduling cost  $\sum_{i=0}^{n+1} (|s_i^q - s_i^b|)$  is calculated in accordance to the reactive schedule, and the baseline schedule. The reactive schedule and the result (duration, cost) are added to RS. Then, the  $gen = gen + 1$  is executed, and the RBRSA is repeated to be called to generate new schedules. Once  $gen$  is

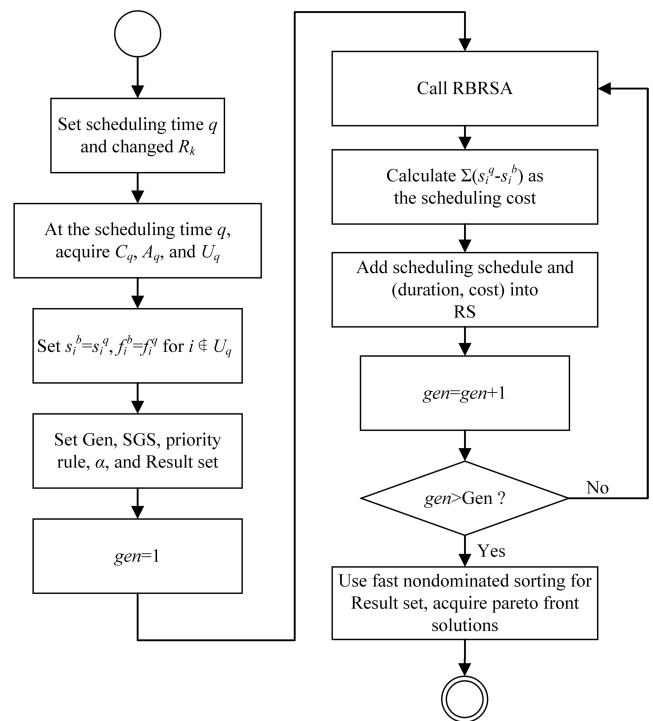


FIGURE 2. RBRSA based on SSGS.

greater than Gen, the new schedule is no longer generated. Next, the fast nondominated sorting [31] is used to acquire the pareto front solutions of RS.

**IV. COMPUTATIONAL EXPERIMENT**

**A. EXPERIMENTAL DESIGN**

In this study, all benchmark instances of J60.sm and J120.sm from the project scheduling problem library (PSPLIB) [34] are tested. Each instance includes 4 types of renewable

resources. Besides this paper studies the RCPRSP, which is under the assumption that each resource amount varies after a random time  $q$ . In the experiment, the random scheduling time  $q$  for each instance in J60.sm is arbitrarily set at 25. For each instance in J120.sm, the reactive scheduling time  $q$  is set at 45. To ensure that each resource amount can still meet the minimal resource demand of each activity after the initial resource amount decreases by 10%, the initial resource amount of every instance increased by 20% will be regarded as the new  $R_k$ . Therefore, each resource amount varies after the time  $q$  is based on the new  $R_k$ . Besides, the bi-objective RBRSA is used to solve the bi-objective RCPRSP, in which each renewable resource  $R_k$  decreases or increases by 10% after the time  $q$ . Furthermore, 19 popular priority rules, as described in the Table 1, are applied in the RBRSA with SSGS/PSGS. In addition, the algorithm is written in *Python*. The experiments are conducted on a PC with an Intel (R) Core (TM) i5-10210U CPU @ 1.60GHz, 8.00 GB of RAM, and the Windows 10 operating system.

Solving the bi-objective optimization problem is a process that gradually approaches the pareto front, which consists of a set of pareto optimal solutions. However, it is difficult to acquire the Pareto front in real-life large-scale problems. As a result, only the best solutions we found can be considered pareto front solutions. Likewise, the performance of the algorithm is further evaluated by judging the quality of these best solutions (approximate solutions). And researchers proposed various evaluation methods [31], [33], [35]–[37] to evaluate the quality of these approximate solutions. The bi-objective RBRSA with each priority rule generates a different pareto front, and thereby, the 19 priority rules employed in RBRSA separately will generate the 19 pareto fronts. The final pareto front, which is also generated by the fast non-dominated sorting procedure, is defined as the pareto front of the union set of these 19 pareto fronts. Based on the final pareto front, we select the first criterion, coverage of the final set (final pareto front), referring to the coverage of two sets, denoted by CS [38], to analyze and evaluate the performance of RBRSA, which has a special priority rule, as follows:

$$CS_{pr} = \frac{(|a \in P_{pr}; \exists b \in P_{final} : b \succcurlyeq a|)}{|P_{pr}|} \quad (11)$$

In the above formula, CS measures the convergence of the algorithm,  $b \succcurlyeq a$  represents that the solution  $a$  is dominated by or equals to  $b$ ,  $P_{final}$  is the set of the final pareto front which is obtained from all RS with 19 priority rules,  $P_{pr}$  is the set of pareto front obtained by RBRSA with a specified priority rule, and  $|a \in P_{pr}; \exists b \in P_{final} : b \succcurlyeq a|$  the number of solutions in  $P_{pr}$  that are dominated by the solutions in  $P_{final}$ . The lower the  $CS_{pr}$  value is, the better the convergence of the algorithm is, and RBRSA with the specified priority rule performs well. The second criterion is the diversity metric ( $\Delta$ ) [35], which evaluates the diversity of solutions in the

TABLE 2. Performance of algorithm with SSGS and different  $\alpha$ .

SSGS	CS		$\Delta$	
	$R_k \times 0.9$	$R_k \times 1.1$	$R_k \times 0.9$	$R_k \times 1.1$
$\alpha=1$	0.21349	0.24511	0.95688	0.92839
$\alpha=2$	0.17790	0.23616	0.94391	0.92963
$\alpha=3$	0.16729	0.21114	0.93611	0.93691

TABLE 3. Performance of algorithm with PSGS and different  $\alpha$ .

SSGS	CS		$\Delta$	
	$R_k \times 0.9$	$R_k \times 1.1$	$R_k \times 0.9$	$R_k \times 1.1$
$\alpha=1$	0.30903	0.25309	0.94118	0.83118
$\alpha=2$	0.30934	0.27192	0.93442	0.84031
$\alpha=3$	0.34121	0.33761	0.93730	0.85955

pareto front algorithms.

$$\Delta = \frac{h_f + h_l + \sum_{i=0}^{n-1} |h_i - \bar{h}|}{h_f + h_l + (n - 1)\bar{h}} \quad (12)$$

In the formula,  $h_i$  is the distance between adjacent two points, and  $\bar{h}$  is the mean value of  $h_i$ .  $h_f$  is the distance between the right extreme point (rightmost solution) in the  $P_{final}$  and the rightmost solution in the  $P_{pr}$ .  $h_l$  is the distance between the right extreme point (leftmost solution) in the  $P_{final}$  and the leftmost solution in the  $P_{pr}$ . In an ideal situation,  $h_f = h_l = 0$ ,  $h_i = \bar{h}$  for each  $h_i$  so as to  $\Delta = 0$ . The lower the value of  $\Delta$  is, the wider the distribution of the approximate solutions is. Specifically, the better the diversity of approximate solutions is. A lower value of  $\Delta$  indicates that the specified priority rule adopted in the algorithm has better performance.

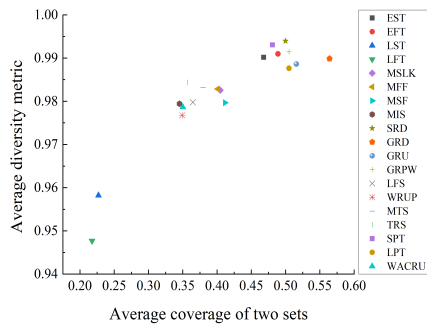
### B. ANALYSIS OF THE EFFECT OF A COMBINATION OF DIFFERENT PARAMETERS $\alpha$ AND A SCHEDULING GENERATION SCHEME

Section III-A has illustrated that the algorithm performance is affected by the parameter  $\alpha$  of the Pr used in the RBRSA. In this paper, according to Türkakın *et al.* [30] and Dek [33], the parameter  $\alpha$  is valued by 1, 2 or 3, respectively. Also, the performance of RBRSA with different schedule generation schemes are explored. Besides, 480 benchmark instances in J60.sm are tested to obtain the pareto front of every priority rule and the final pareto front under different values of  $\alpha$ . Then, considering the different SGS and  $\alpha$ , the average values of CS and  $\Delta$  can be calculated, respectively.

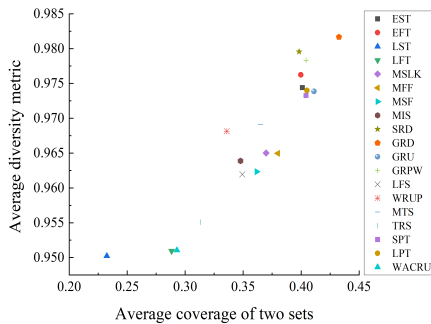
For the SSGS, the results with different  $\alpha$  are shown in Table 2.

For the PSGS, the results with different  $\alpha$  are shown in Table 3.

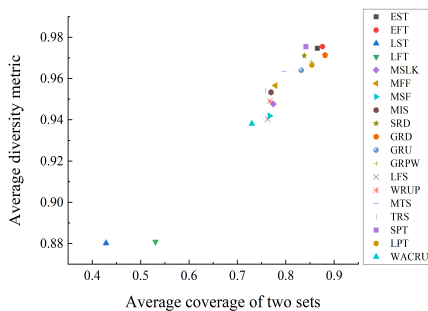
It is shown from Table 2 that for RBRSA with SSGS,  $\alpha = 3$  obtains the best value of CS when each  $R_k$  decreases or increases by 10%, and  $\alpha = 2$  acquires the second-best value. For the diversity metric ( $\Delta$ ), when the  $R_k$  decreases by 10% and the parameter  $\alpha = 3$ , it is the lowest. However, the  $\Delta$  value is the lowest when  $\alpha = 1$ , which is relatively close to the  $\Delta$  value when  $\alpha = 3$  with each  $R_k$  increasing by 10%.  $\Delta$



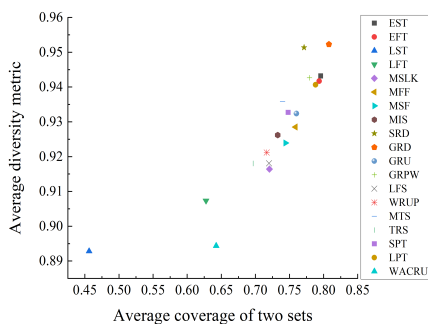
(a) J60,  $\alpha=3$ , Resource amount\*0.9



(b) J60,  $\alpha=3$ , Resource amount\*1.1



(c) J120,  $\alpha=3$ , Resource amount\*0.9



(d) J120,  $\alpha=3$ , Resource amount\*1.1

FIGURE 3. Performance of RBRSA with SSGS and different priority rules.

values are ranked second when  $\alpha = 2$  with each  $R_k$  increasing or decreasing by 10%.

The results for PSGS are shown in Table 3. For the CS, it can be seen that the CS values are the lowest when  $\alpha = 1$  with each  $R_k$  increasing or decreasing by 10%, and the CS values are the second lowest when  $\alpha = 2$  in the same resource case. With each  $R_k$  decreasing by 10%, the  $\Delta$  value is the lowest when  $\alpha = 2$  which is close to the  $\Delta$  value when  $\alpha = 1$ . In the case of each  $R_k$  increasing by 10%, the  $\Delta$  value is lowest when  $\alpha = 1$ , and the  $\Delta$  value ranks second lowest when  $\alpha = 2$ .

Based on the above analysis, it can be concluded that the RBRSA with both SSGS and  $\alpha = 3$  achieves the best performance, followed by  $\alpha = 2$ . For RBRSA with PSGS,  $\alpha = 1$  is the best, and  $\alpha = 2$  ranks second. Consequently, in the following study,  $\alpha$  is valued by 3 in the RBRSA with SSGS, and  $\alpha$  is valued by 1 in the RBRSA with PSGS.

### C. COMPARISON OF PRIORITY RULES

In this section, the performance of priority rules are explored for solving the presented bi-objective RCPRSP by RBRSA. Based on the conclusions of section IV-B, the  $\alpha$  is valued by 3 for RBRSA with SSGS, and  $\alpha$  is valued by 1 for RBRSA with PSGS. Consequently, the results of average values of CS and  $\Delta$ , denoted by (average value of CS, average value of  $\Delta$ ), are calculated based on the pareto fronts which are obtained by testing all benchmark instances in J60.sm and J120.sm.

#### 1) RBRSA WITH SSGS

In this part, the performance RBRSA with SSGS with different priority rules are tested when each  $R_k$  increases or decreases by 10% after the time  $q$ , and  $\alpha$  is valued by 3. In Fig. 3, the average values of both CS and  $\Delta$  of all the RBRSA with different priority rules are shown.

According to the definitions of CS and  $\Delta$ , the lower the values of these two criteria are, the better the performance of the priority rules are. Fig. 3 (a) and Fig. 3 (c) shows that the LST and LFT have the best performance because the values of CS and  $\Delta$  of these two points are relatively low, with each  $R_k$  decreasing by 10%. In Fig. 3 (b) and Fig. 3 (d), the (average value of CS, average value of  $\Delta$ )s obtained by LST, LFT, WACRU, and TRS are all close to the origin when each  $R_k$  increases by 10%. This means that the LST, LFT, WACRU, and TRS priority rules with SSGS have relatively better performance.

We randomly employ a project instance, j605\_7 in J60.sm, to compare the performance of the mentioned priority rules. The pareto fronts obtained by RBRSA with LST, LFT, WACRU, and TRS are shown in Fig. 4, in which the solution is denoted by (duration, cost) in RS. In Fig. 4, the solutions acquired by LST and LFT have pareto fronts that are closer to the origin, implying that the solutions obtained by LST and LFT can dominate the majority of the solutions obtained by



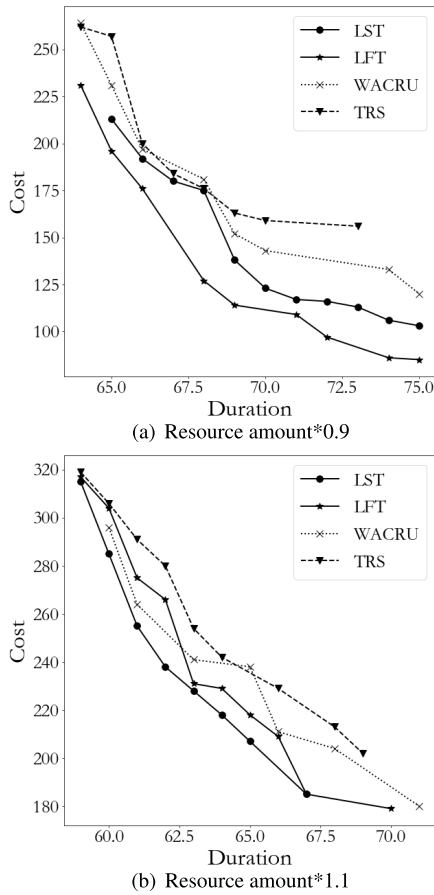


FIGURE 4. Pareto fronts of four priority rules under SSGS.

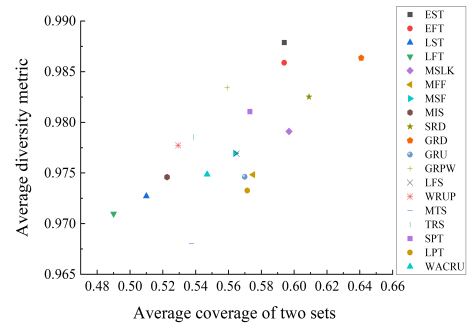
other priority rules. It can be concluded that the LFT/LST employed in the RBRSA with SSGS shows performs better than the other two.

2) RBRSA WITH PSGS

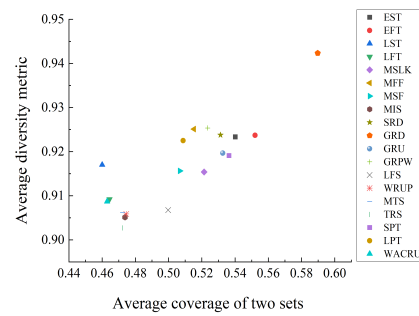
In this section, the performance of RBRSA with PSGS and different priority rules are investigated by a computational experiment. Similarly, each resource amount will increase or decrease by 10% after time  $q$ . The results, denoted by the (average value of CS, average value of  $\Delta$ )s and obtained by RBRSA with PSGS and different priority rules, are shown in Fig. 5.

As shown in Fig. 5 (a) and Fig. 5 (d), when each  $R_k$  decreases or increases by 10%, the (average value of CS, average value of  $\Delta$ )s, obtained by LST, LFT, MTS, MIS, WRUP, and WACRU, are relatively close to the origin. In Fig. 5 (b) and Fig. 5 (d), the values of CS and  $\Delta$  at points LST and LFT are the lowest ones with each  $R_k$  decreasing or increasing by 10%. Overall, it can be deduced that the RBRSA with these 6 priority rules have better performance.

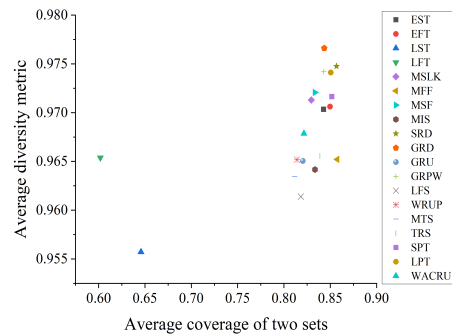
Randomly selected from J120.sm, a project instance j12022\_1, is solved by RBRSA with LST, LFT, MTS, MIS, WRUP, and WACRU, respectively, to compare the



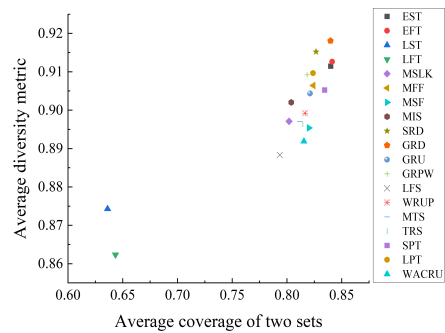
(a) J60,  $\alpha=1$ , Resource amount\*0.9



(b) J60,  $\alpha=1$ , Resource amount\*1.1

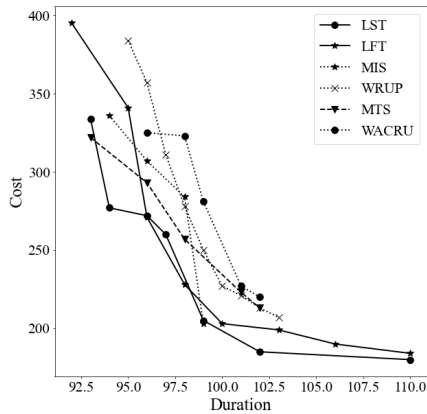


(c) J120,  $\alpha=1$ , Resource amount\*0.9

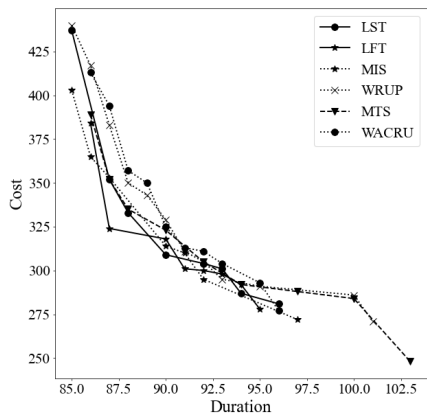


(d) J120,  $\alpha=1$ , Resource amount\*1.1

FIGURE 5. Performance of RBRSA with PSGS and different priority rules.



(a) Resource amount\*0.9



(b) Resource amount\*1.1

FIGURE 6. Pareto fronts of five priority rules under PSGS.

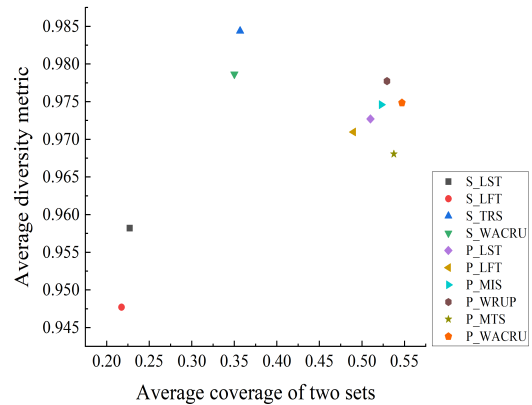
TABLE 4. Better priority rules under different scheduling generation schemes.

SSGS( $\alpha=3$ )	LST, LFT
PSGS( $\alpha=1$ )	LST, LFT

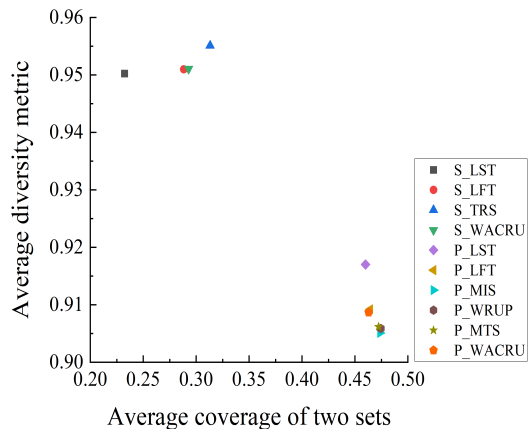
performance of these 6 priority rules. The results in Fig. 6 show that the pareto fronts of LST and LFT are closer to the origin, which means the pareto fronts of LST and LFT can dominate other pareto fronts. Therefore, compared to other priority rules, the priority rules that include LST and LFT have the best performance in the RBRSA.

Finally, the priority rules with better performance are shown in the Table 4.

Based on the conclusions from Fig. 4, Fig. 6, and Table 4, we further compare the performance of the excellent priority rules in RBRSA with different SGS by test all instances inJ60.sm. Meanwhile, the  $\alpha$  is valued at 3 in the algorithm with SSGS, while the  $\alpha$  is valued at 1 in the algorithm with PSGS. Then, the different  $\alpha$ , SGS, and priority rules in Table 4 are employed in RBRSA to acquire the (average value of CS, average value of  $\Delta$ )s. Besides, as is shown in Fig. 7, these points are denoted by S\_priority rule or P\_priority rule, which indicate the combination of the SSGS/PSGS and different priority rules. For example, S\_LST



(a) Resource amount\*0.9



(b) Resource amount\*1.1

FIGURE 7. Performance of RBRSA with different SGS and priority rules.

signifies that the SSGS and priority rule LST are adopted in RBRSA, and P\_LFT signifies that the PSGS and priority rule LFT are employed in RBRSA.

As is shown in Fig. 7 (a), S\_LST and S\_LFT are close to the origin, which means the average values of CS and  $\Delta$  are the lowest for these two points, with each  $R_k$  decreasing by 10%. Furthermore, in Fig. 7 (b), when each  $R_k$  increases by 10%, the CS values of S\_LST and S\_LFT are less than others because the maximum deviation of the CS value of all points is about 0.25 (by 0.475, subtracting 0.225). Whereas the  $\Delta$  values of all points have a little difference because the maximum deviation of  $\Delta$  values is about 0.045 (by 0.95, subtracting 0.905). Therefore, the paper concluded that the priority rules LST and LFT in RBRSA with SSGS and  $\alpha = 3$  have better performance than the other combinations.

## V. CONCLUSION

During the execution processes of projects, there may be some uncertain factors that lead to the predefined schedules being unfeasible. Reactive project scheduling problem is one of the important project scheduling problems to address the uncertainty of projects in progress. This paper studies the RCPRSP under resource uncertainty. To better balance the requirements of managers in terms of project duration and adjustment cost under the constraints of new resource amounts and precedence relations, we propose a

new bi-objective reactive project scheduling model to minimize both project duration and schedule adjustment cost. The RBRSA, which combines two scheduling generation schemes with different priority rules, is designed to solve the presented problem. It is explored how the key parameter of RBRSA affects the performance of RBRSA with different schedule generation schemes. Finally, computational experiments are used to compare the performance of different priority rules in two popular criteria: coverage of two sets and diversity metric. Some excellent combinations of parameter value, schedule generation scheme, and priority rules are explored in RBRSA for the presented bi-objective RCPRSP. The main contributions to this paper are as follows: (1) Considering the uncertainty of resource amounts, we propose a new bi-objective RCPRSP. Other than the existing research, which only minimizes the adjustment cost, the problem is a bi-objective scheduling problem aiming to achieve a balance between schedule adjustment cost and project duration. (2) To solve the bi-objective RCPRSP, we design a new heuristic algorithm based on priority rules. Although there is a lot of studies on heuristics based on priority rules for single objective project scheduling problems, to our best knowledge, there has been no similar heuristic based on priority rules for solving bi-objective project scheduling so far. The study of the presented RCPRSP provides the project leader with more decision-making ability to control the project since it can balance the two core objectives, duration and cost. The priority rule based on heuristic is an intuitive and efficient solution for the bi-objective RCPRSP, and it is also a promising alternative to solve other multi-objective project scheduling problems. How to further improve the performance of the RBRSA and compare it with other evolutionary multi-objective algorithms may be a future research direction.

## REFERENCES

- [1] Ö. Hazir and G. Ulusoy, "A classification and review of approaches and methods for modeling uncertainty in projects," *Int. J. Prod. Econ.*, vol. 223, May 2020, Art. no. 107522.
- [2] L. Olivier, D. Erik, and H. Willy, "Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities," *J. Scheduling*, vol. 11, no. 2, pp. 121–136, Apr. 2008.
- [3] Z. W. He, M. J. Ning, and Y. Xu, "A survey of proactive and reactive project scheduling methods," *Oper. Res. Manage. Sci.*, vol. 25, no. 5, pp. 278–287, 2016.
- [4] T. Tanikawa, Y. Tanimizu, and H. Nonomiya, "Expansion of a diffuse co-evolutionary genetic algorithm in reactive scheduling," in *Proc. Mech. Eng. Congr. Jpn.*, 2016, Art. no. S1420204.
- [5] R. Yonemoto and H. Suwa, "Reactive scheduling based on adaptive manipulator operations in a job shop configuration with two machines," *Int. J. Autom. Technol.*, vol. 15, no. 5, pp. 661–668, Sep. 2021.
- [6] H. Chen, G. Liu, S. Yin, X. Liu, and D. Qiu, "ERECT: Energy-efficient reactive scheduling for real-time tasks in heterogeneous virtualized clouds," *J. Comput. Sci.*, vol. 28, pp. 416–425, Sep. 2018.
- [7] F. Deblaere, E. Demeulemeester, and W. Herroelen, "Reactive scheduling in the multi-mode RCPRSP," *Comput. Oper. Res.*, vol. 38, no. 1, pp. 63–74, 2011.
- [8] S. Elloumi, T. Loukil, and P. Fortemps, "Reactive heuristics for disrupted multi-mode resource-constrained project scheduling problem," *Expert Syst. Appl.*, vol. 167, Apr. 2021, Art. no. 114132.
- [9] W. Wang, X. Ge, L. Li, and J. Su, "Proactive and reactive multi-project scheduling in uncertain environment," *IEEE Access*, vol. 7, pp. 88986–88997, 2019.
- [10] S. Van de Vonder, E. Demeulemeester, R. Leus, and W. Herroelen, "Proactive-reactive project scheduling trade-offs and procedures," in *Perspectives in Modern Project Scheduling*. New York, NY, USA: Springer, 2006, pp. 25–51.
- [11] S. Van De Vonder, F. Ballestín, E. Demeulemeester, and W. Herroelen, "Heuristic procedures for reactive project scheduling," *Comput. Ind. Eng.*, vol. 52, no. 1, pp. 11–28, Feb. 2007.
- [12] F. Deblaere, E. Demeulemeester, W. Herroelen, and S. Van De Vonder, "Robust resource allocation decisions in resource-constrained projects," *Decis. Sci.*, vol. 38, no. 1, pp. 5–37, Feb. 2007.
- [13] A. Delgoshaei, T. Rabczuk, A. Ali, and M. K. A. Ariffin, "An applicable method for modifying over-allocated multi-mode resource constraint schedules in the presence of preemptive resources," *Ann. Operations Res.*, vol. 259, nos. 1–2, pp. 85–117, Dec. 2017.
- [14] J. Kuster, D. Jannach, and G. Friedrich, "Applying local rescheduling in response to schedule disruptions," *Ann. Oper. Res.*, vol. 180, no. 1, pp. 265–282, Nov. 2010.
- [15] R. K. Chakraborty, R. A. Sarker, and D. L. Essam, "Multi-mode resource constrained project scheduling under resource disruptions," *Comput. Chem. Eng.*, vol. 88, pp. 13–29, May 2016.
- [16] M. Ning, Z. He, T. Jia, and N. Wang, "Metaheuristics for multi-mode cash flow balanced project scheduling with stochastic duration of activities," *Autom. Construct.*, vol. 81, pp. 224–233, Sep. 2017.
- [17] S. Elloumi, P. Fortemps, and T. Loukil, "Multi-objective algorithms to multi-mode resource-constrained projects under mode change disruption," *Comput. Ind. Eng.*, vol. 106, pp. 161–173, Apr. 2017.
- [18] E. B. Tirkolaei, A. Goli, M. Hematian, A. K. Sangaiah, and T. Han, "Multi-objective multi-mode resource constrained project scheduling problem using Pareto-based algorithms," *Computing*, vol. 101, no. 6, pp. 547–570, Jun. 2019.
- [19] P. Godinho and F. G. Branco, "Adaptive policies for multi-mode project scheduling under uncertainty," *Eur. J. Oper. Res.*, vol. 216, no. 3, pp. 553–562, Feb. 2012.
- [20] M. H. F. Rahman, R. K. Chakraborty, and M. J. Ryan, "Managing uncertainty and disruptions in resource constrained project scheduling problems: A real-time reactive approach," *IEEE Access*, vol. 9, pp. 45562–45586, 2021.
- [21] R. K. Chakraborty, R. A. Sarker, and D. L. Essam, "Single mode resource constrained project scheduling with unreliable resources," *Oper. Res.*, vol. 20, no. 3, pp. 1369–1403, Sep. 2020.
- [22] R. K. Chakraborty, H. F. Rahman, K. M. A. Haque, S. K. Paul, and M. J. Ryan, "An event-based reactive scheduling approach for the resource constrained project scheduling problem with unreliable resources," *Comput. Ind. Eng.*, vol. 151, Jan. 2021, Art. no. 106981.
- [23] H. Suwa and D. Morita, "Reactive project scheduling method to enhance project progress under uncertainty," *J. Adv. Mech. Des., Syst., Manuf.*, vol. 10, no. 3, 2016, Art. no. JAMDSM0051.
- [24] S. Q. Zhang, X. D. Chen, Q. X. Chen, and X. Chen, "Reactive scheduling algorithm for multiple mould and die projects based on optimized resource flow constraints," *Syst. Eng. Theory Pract.*, vol. 31, no. 8, pp. 1571–1580, 2011.
- [25] A. Delgoshaei, A. Aram, V. Mantegh, S. Hanjani, A. H. Nasiri, and F. Shirmohamadi, "A multi-objectives weighting genetic algorithm for scheduling resource-constrained project problem in the presence of resource uncertainty," *Int. J. Supply Oper. Manage.*, vol. 6, no. 3, pp. 213–230, 2019.
- [26] Z. Chen, "Efficient priority rules for the stochastic resource-constrained project scheduling problem," *Eur. J. Oper. Res.*, vol. 270, no. 3, pp. 957–967, Apr. 2018.
- [27] W. Peng, M. Huang, and H. Yongping, "A multi-mode critical chain scheduling method based on priority rules," *Prod. Planning Control*, vol. 26, no. 12, pp. 1011–1024, Sep. 2015.
- [28] M. Dumic and D. Jakobović, "Ensembles of priority rules for resource constrained project scheduling problem," *Appl. Soft Comput.*, vol. 110, Oct. 2021, Art. no. 107606.
- [29] T. R. Browning and A. A. Yassine, "Resource-constrained multi-project scheduling: Priority rule performance revisited," *Int. J. Prod. Econ.*, vol. 126, no. 2, pp. 212–228, Aug. 2010.
- [30] O. H. Türkakın, D. Arditi, and E. Manisali, "Comparison of heuristic priority rules in the solution of the resource-constrained project scheduling problem," *Sustainability*, vol. 13, no. 17, p. 9956, Sep. 2021.
- [31] P. C. Roy, K. Deb, and M. M. Islam, "An efficient nondominated sorting algorithm for large number of fronts," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 859–869, Mar. 2019.

[32] Y. Y. Shou, "A multi-pass biased random sampling method for scheduling multiple projects," *J. Ind. Eng. Eng. Manage.*, vol. 19, no. 3, pp. 32–35, 2005.

[33] K. Deb, *Multiobjective Optimization Using Evolutionary Algorithms*. Hoboken, NJ, USA: Wiley, 2001.

[34] R.-M. Chen, C.-L. Wu, C.-M. Wang, and S.-T. Lo, "Using novel particle swarm optimization scheme to solve resource-constrained scheduling problem in PSPLIB," *Expert Syst. Appl.*, vol. 37, no. 3, pp. 1899–1910, Mar. 2010.

[35] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Aug. 2002.

[36] X. Cai, H. Sun, and Z. Fan, "A diversity indicator based on reference vectors for many-objective optimization," *Inf. Sci.*, vols. 430–431, pp. 467–486, Mar. 2018.

[37] P. M. Pradhan and G. Panda, "Pareto optimization of cognitive radio parameters using multiobjective evolutionary algorithms and fuzzy decision making," *Swarm Evol. Comput.*, vol. 7, pp. 7–20, Dec. 2012.

[38] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion," in *Proc. IEEE Int. Conf. Evol. Comput.*, Jul. 2006, pp. 892–899.



**GUANGMIN JIA** was born in Weifang, Shandong, China, in 1998. He received the B.A. degree in English language and literature from the Wenjing College, Yantai University, Yantai, in 2021, where he is currently pursuing the master's degree in translation and interpreting.



**HAN SONG** received the B.S. degree in engineering from the Wuhan University of Science and Technology (WUST), Wuhan, China, in 2018. She is currently pursuing the master's degree in business management with Yantai University. Her research interests include project management and technology economics and management.



**WULIANG PENG** received the Ph.D. degree in industrial engineering from Northeastern University, in 2008. He is currently a Full Professor at Yantai University. He has published articles in various journals such as the *International Journal of Project Management*, the *International Journal of Production Economics*, and the *International Journal of Production Research*. He is a reviewer for several international journals.

...