

Article

# Dynamic Optimization of the Multi-Skilled Resource-Constrained Project Scheduling Problem with Uncertainty in Resource Availability

Min Wang<sup>1</sup>, Guoshan Liu<sup>2</sup> and Xinyu Lin<sup>2,\*</sup><sup>1</sup> College of Business Administration, Fujian Jiangxia University, Fuzhou 350108, China<sup>2</sup> Business School, Renmin University of China, Beijing 100872, China

\* Correspondence: 2020000696@ruc.edu.cn

**Abstract:** Multi-skilled resources have brought more flexibility to resource scheduling and have been a key factor in the research of resource-constrained project scheduling problems. However, existing studies are mainly limited to deterministic problems and neglect some uncertainties such as resource breakdowns, while resource availability may change over time due to unexpected risks such as the COVID-19 pandemic. Therefore, this paper focuses on the multi-skilled project scheduling problem with uncertainty in resource availability. Different from previous assumptions, multi-skilled resources are allowed a switch in their skills, which we call dynamic skill assignment. For this complex problem, a nested dynamic scheduling algorithm called GA-PR is proposed, which includes three new priority rules to improve the solving efficiency. Moreover, the algorithm's effectiveness is verified by an example, and the modified Project Scheduling Problem Library (PSPLIB) is used for numerical experimental analysis. Numerical experiments show that when the uncertainty in resource availability is considered, the more skills the resource has and the more resources are supplied, the better the dynamic scheduling method performs; on the other hand, the higher the probability of resource unavailability and the more skills are required, the worse the dynamic scheduling method performs. The results are helpful for improved decision making.

**Keywords:** project scheduling; uncertainty in resource availability; multi-skilled resource; dynamic skill assignment

**MSC:** 90B36

**Citation:** Wang, M.; Liu, G.; Lin, X. Dynamic Optimization of the Multi-Skilled Resource-Constrained Project Scheduling Problem with Uncertainty in Resource Availability. *Mathematics* **2022**, *10*, 3070. <https://doi.org/10.3390/math10173070>

Academic Editors: Antonin Ponsich, Mariona Vila Bonilla and Bruno Domenech

Received: 12 July 2022

Accepted: 22 August 2022

Published: 25 August 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The resource-constrained project scheduling problem (RCPSP) has been an important topic within project management over the past few decades. Extensive focus has been placed on single-skilled resources, while multi-skilled resources are becoming increasingly common with the development of the economy [1]. This extension of the RCPSP is known as the multi-skilled resource-constrained project scheduling problem (MSRCPSPP). It was inspired by a problem in the software development industry, where employees had several skills relating to programming, data analysis, debugging and so on [2]. MSRCPSPP is suitable in projects with multi-skilled human resources or multi-functional machines. It has been a prevalent topic in recent years and has been gradually applied in production scheduling [3], research and development [4], construction engineering [5] and other projects [6,7]. Although multi-skilled resource increases scheduling flexibility and expands alternatives for project scheduling, it makes the problem more challenging. One needs to decide not only resource scheduling matters but also skill assignments.

In practice, resource unavailability is a frequent occurrence, especially in the wake of COVID-19, such as staff turnover, equipment maintenance, and transportation interruption. In this situation, project managers are forced to take a series of measures to make the project

scheduling more efficient and to adapt quickly as possible to uncertainties [8]. Therefore, we focus on the MSRCPSP with uncertainty in resource availability in this paper. Moreover, distinct from previous assumptions, multi-skilled resources are allowed to switch their skills, which we call dynamic skill assignment. This means that when some resources with one skill are unavailable, the impact of resource shortage can be alleviated by skill switching from other idle resources with the same skill. If dynamic skill assignment still fails to make up for the shortage of resources, additional resources will be considered under the constraint of deadlines, such as the recruitment of temporary workers, equipment renting and so on. According to WEC (World Employment Confederation), temporary employment accounts for 70 percent of the global HR market, which is worth nearly USD 4 hundred billion. It plays an important role in reducing the cost and relieving the shortage of resources. Methods for optimizing project scheduling and dynamic skill assignment with uncertainty in resource availability so as to achieve the goal of minimum additional resource costs are the focus of this paper.

This paper has the following three contributions. First, we extend the MSRCPSP with uncertainty in resource availability, and the uncertainty is described by the Markov process. Second, dynamic skill assignment is proposed, which allows multi-skilled resources to switch skills. Third, a nested dynamic scheduling algorithm called GA-PR is proposed, which includes three new priority rules to improve the solving efficiency, and the effectiveness of the algorithm is proved by comparing the existing static and random scheduling method.

The remainder of this paper is organized as follows. A literature review is presented in Section 2. Definitions of the MSRCPSP with uncertainty in resource availability are discussed in Section 3. The nested dynamic scheduling algorithm is explained in detail in Section 4. A numerical example is provided in Section 5 to illustrate the new model and the new algorithm. The computational experiments and results analysis are shown in Section 6. Section 7 is the conclusion.

## 2. Literature Review

Although multi-skilled resource make solving MSRCPSP more flexible, it also renders the scheduling procedure more complex and difficult; thus, modern methods and tools are usually used to improve scheduling processes [9]. Methods for designing more effective scheduling procedures with modern methods have become important topics in the MSRCPSP. Bellenguez and Néron (2007) [2] proposed that each activity needs a specific set of skills, and the resources are staff members who possess fixed skill(s). Moreover, these staff members have unavailable periods. To minimize the makespan, a Branch-and-Bound method is proposed. Generally, the more skills a staff member possesses, the more costs are incurred. To minimize the total costs for multi-skilled personnel, Li and Womer (2009) [10] develop a hybrid Benders decomposition (HBD) algorithm that combines the complementary strengths of mixed-integer linear programming and constraint programming. Correia and Saldanha-da-Gama (2014) [11] consider that the costs associated with resources include fixed and variable costs. The fixed costs are incurred simply by using resources, while variable costs depend on the final makespan of the project. For this problem, a mathematical programming modeling framework is proposed, and a non-linear objective function is included, which can be linearized at the expense of an additional set of continuous variables. For resources considering skills, in addition to the cost, the skill level also directly affects the project scheduling scheme such that the higher the skill, the shorter the task duration. Heimerl and Kolisch (2010) [4] consider the MSRCPSP in a multi-project-environment (i.e., the processing of the projects' external and internal resources with different skills and at different performance levels). Thus, the question is how projects are scheduled and how resources are assigned to a project such that different requirements are met, keeping the costs minimal. To address the complex project-scheduling problem, a mixed-integer linear program with a tight LP-relaxation, which makes solving real-world problems possible, is proposed. A related problem was examined by Firat and Hurkens (2012) [12]. The

authors consider a mixed-integer-based approach for a multi-skill work-load problem and where skill levels are not homogeneous. Each activity has requirements for each skill-level combination. The goal is to maximize the number of tasks processed in each workday. Snauwaert and Vanhoucke (2021) [13] addressed an MSRCPSP with breadth and depth of skills, where the breadth of a resource is perceived as the amount of skills an employee masters and the depth of a skill is the efficiency level at which work can be performed by a resource that masters that skill. After that, in 2022, they studied how hierarchical skills (depth of skills) affect project scheduling from aspects of efficiency, cost, and quality [14].

Although the above studies provide references for the model and algorithmic design of MSRCPSP, they assume that the availability of renewable resources remains constant over time and rarely consider uncertainties in resource availability [1]. This assumption may be too strict. Resource availability might change in response to the availability of labor due to vacation days or varying availability of equipment due to maintenance [15]. A relatively common type of research is the project scheduling problem under uncertainties in project duration, including proactive scheduling [16] and reactive scheduling [17]. Once resources are unavailable (staff turnover, machine failure, etc.), the original scheduling is no longer feasible, especially for key resources. Therefore, the uncertainty in resource availability has gradually become an important and difficult point in project scheduling. Lambrechts et al. (2008) [18] introduced a variant of the RCPSP, for which the uncertainty in resource availability is considered. The objective is to find a robust schedule that minimizes the schedule's instability cost. The schedule's instability cost is the expected weighted sum of the absolute deviations between the planned and the actual starting times of the activity during the execution of proposed proactive and reactive strategies. Furthermore, to determine the impact of unexpected resource breakdowns on activity durations, they developed an approach for inserting explicit idle time into the project schedule. This was also implemented to protect it from disruptions caused by resource unavailability [19].

The literature cited above indicates that regardless of whether project scheduling considers uncertainties in terms of duration or resource availability, the idea is to set a buffer, namely time or resource buffer. The buffer can effectively protect the scheduling benchmark and improve the robustness of the solution. However, considering multi-skilled resources, effectively using this attribute to deal with the disturbance caused by uncertainties in resource availability has become a noteworthy problem. To our knowledge, there are only a few authors that incorporate uncertainty in resource availability in MSRCPSP, thus making this topic an interesting and novel path for research.

Ahmadpour and Ghezavati (2019) [20] provide a fuzzy scheduling model for the RCPSP, which considers fuzzy conditions for the calendar of the project. Multi-skilled human resources are also being considered to cope with the risk of resource shortages and delays in project completion. The results obtained from the fuzzy model for the value of objective function were evaluated under the influence of the resource calendar, consequently showing its benefits. The results provide a research idea for the MSRCPSP with resources uncertainty. However, this study assumes that once resources are assigned to a specific skill, they will be completely unchangeable until the end of the project. However, when a resource is unavailable due to resource uncertainty, other resources with the same skill are often recruited to continue the activity and avoid delay. When the resource is available again, the resource may be required to use other skills and perform other tasks to avoid information asymmetry caused by the resource transition. This multi-skilled resource dynamic skill assignment realizes the rotation of different resources with different skills among different tasks and alleviates the disturbance caused by the uncertainty in resource availability [21]. Compared with the situation that skills cannot be changed, this scheduling is closer to real-world issues, thus making resource scheduling more flexible and effective in dealing with absenteeism [22]. Moreover, as multi-skilled resources are often acquired through cross-training, a worker who does not frequently practice one skill may tend to forget it [23,24]. In the long run, multi-skill can be easily transformed into single-skill. Therefore, in this study, we relax the assumption that resources are allowed to change skills.

Compared to the existing literature, uncertainties in resource availability and dynamic skill assignment of multi-skilled resources are considered on the basis of MSRCPSP in this paper. When a resource is unavailable, it can be replaced by skill switching relative to other multi-skilled resources; alternatively, the external supplement of resources is another method. The uncertainty in resource availability is described by the Markov process. This is mainly because the evolution of availability or unavailability of resources is a discrete stochastic process that evolves with time. In other words, only the present determines the future, and the past is irrelevant. This corresponds to the Markov chain [25]. Then, we designed a nested scheduling algorithm combined with three new priority rules to solve this complex problem to minimize additional resource costs. Finally, the effectiveness of the proposed algorithm is verified by experimental analysis. The results of this research are helpful in the decision making of multi-skilled project scheduling. It is also valuable for expanding the project scheduling research in an uncertain environment, especially since uncertainty appears to be an extremely difficult element to deal with. Although most researchers recognize the importance and ubiquity of uncertainty, it remains the most popular topic of future research sections in many papers [26].

### 3. Problem Description

In the MSRCPSP, we employ the activity-on-node (AoN) representation and assume a zero time lag for precedence relations. The project consists of  $n + 2$  activities. The duration of activity  $j (j \in V, V = \{0, 1, 2, \dots, j, \dots, n + 1\})$  is  $d_j (d_j \in N_0)$  and  $d_0 = d_{n+1} = 0$ . We assume that the project needs  $K$  types of renewable resources, the set is  $R (R = \{1, 2, \dots, k, \dots, K\})$ , and the availability of each type of resource is  $|R_k|$ ; that is,  $R_k = \{r_{k1}, r_{k2}, r_{k3}, \dots, r_{ki}, \dots\}$ . Moreover, the project needs  $s_K$  types of skills, and the set is  $S_k (S_k = \{s_1, s_2, \dots, s_k, \dots, s_K\})$ . The skill(s) mastered by each resource is predefined, and we assume that all resources in the set of  $R_k$  master skill  $s_k$ , which we call the initial skill of resource  $r_{ki}$ . Whether there is mastery of other skills or not is randomly generated, and it is indicated by  $x_{r_{ki}, s_k}$ , where  $x_{r_{ki}, s_k} = 1$  indicates that resource  $r_{ki}$  has mastery of skill  $s_k$ . When  $\sum_{s_k \in S_k} x_{r_{ki}, s_k} > 1$ , resource  $r_{ki}$  masters more than one skill. We describe the uncertainty in resource availability by the Markov stochastic process, and we set the state of the resource as  $Z_t (Z_t = \{z_t^{r_{11}}, z_t^{r_{12}}, \dots, z_t^{r_{1i}}, \dots, z_t^{r_{ki}}, \dots | t = 1, 2, \dots, T\})$ , where  $T$  is the deadline of project and  $z_t^{r_{ki}}$  represents whether resource  $r_{ki}$  is available or not at time  $t$ . If it is available,  $z_t^{r_{ki}} = 1$ ; otherwise, it is 0. We assume that the project starts at time zero; thus,  $Z_0$  denotes the initial available/unavailable status of all resources, and we set  $z_0^{r_{ki}} = 1$  for each resource.

The variables and parameters are shown in Table 1.

Table 1. Variables and parameters.

Variables and Parameters	
$x_{j,t}$	Binary variable, if activity $j$ is executed in period $[t, t + 1)$ or not.
$x_{r_{ki}, j, s_k}$	Binary variable, if resource $r_{ki}$ with skill $s_k$ or not.
$x_{r_{ki}, j, s_k, t}$	Binary variable, if resource $r_{ki}$ with skill $s_k$ in period $[t, t + 1)$ or not.
$R_{s_k, t}$	The additional amount of the resource with skill $s_k$ at period $t$ .
$V = \{1, 2, \dots, j, \dots, n + 1\}$	The set of activities.
$SUCC_j$	The immediate successor set of activity $j$ .
$Z_t = \{z_t^{r_{11}}, z_t^{r_{12}}, \dots, z_t^{r_{1i}}, \dots, z_t^{r_{ki}}, \dots   t = 1, 2, \dots, T\}$	The set of resources state.
$z_t^{r_{ki}}$	Binary parameters, if the resource $r_{ki}$ is available at time $t$ or not.
$S_j$	The start time of activity $j$ .
$d_j$	The duration of activity $j$ .

**Table 1.** Cont.

Variables and Parameters	
$R = \{1, 2, \dots, k, \dots, K\}$	The set of $K$ types of renewable resources.
$R_k = \{r_{k1}, r_{k2}, r_{k3}, \dots, r_{ki}, \dots\}$	The set of $R_k$ .
$ R_k $	The availability of $R_k$ .
$S_k = \{s_1, s_2, \dots, s_k, \dots, s_K\}$	The set of skills.
$x_{r_{ki}, s_k}$	Binary parameters, if resource $r_{ki}$ has mastery of the skill $s_k$ or not.
$r_{j, s_k}$	The requirements of skill $s_k$ for executing activity $j$ .
$p_{r_{ki}}$	The unavailability probability of resource $r_{ki}$
$c_{s_k}$	The cost of the resource with skill $s_k$ .
$T$	The deadline of the project.
$P_{r_{ki}}$	The state transition matrix of $r_{ki}$ .
$u_t^{r_{ki}}$	The state probability vector of resource $r_{ki}$ at period $t$ .
$e$	The degree of infeasibility.
$M$	A sufficiently large penalty coefficient.

General RCPSP’s goal is to study how to schedule activities under the constraint of resources and precedence relations in order to minimize the makespan of the project. While in the MSRCPS, as resources are multi-skilled, it is necessary to decide not only activity scheduling but also the skill assignment of resources. To ensure that activities remain uninterrupted and to avoid delays in the project, we allow other idle resources with the same skill to replace the unavailable resources when the skill requirements of activities cannot be met. If this still does not work, additional resources with the same skill would be considered (purchasing, renting, or overtime). Adopting additional resources is an easy and popular method to increase flexibility [26]. According to some studies, it is a proper assumption where there is no difference in performance between temporary and permanent resources [27–29]. Given this situation, a problem arises as to which idle resource will be selected and how to assign them, how many additional resources are needed to meet skill requirements. To solve this, we set the goal to minimize additional costs. Thus, the MSRCPS with uncertainties in resource availability studied in this paper can be described as follows: Under the constraints of precedence relations, resource availability, and project deadline, determine activity scheduling and dynamic skill assignment of multi-skilled resources to minimize additional costs of the project. The research assumptions of this paper is as follows:

- (1) There are many types of multi-skilled resources needed by the project. Each resource can possess one or more skills, and the initial skill of the resource has priority in scheduling. Skill levels are homogeneous among resources.
- (2) The availability or unavailability of different resources is independent of each other.
- (3) The idle resources are allowed to switch skills to replace unavailable resources.
- (4) The unavailability probability of resource  $r_{ki}$  is  $p_{ki}$ , and the state transition matrix of  $r_{ki}$  can be denoted as follows:

$$P_{ki} = \begin{pmatrix} 1 & 0 \\ p_{ki} & 1 - p_{ki} \end{pmatrix} \tag{1}$$

$p_{ki}$  cannot be changed in per unit time, and the state  $z_t^{r_{ki}}$  is subject to the Bernoulli distribution of  $p_{ki}$ . If  $u_t^{ki}$  represents the state probability vector of resource  $r_{ki}$  at period  $t$ , then  $u_{t+1}^{ki} = u_t^{ki} P_{ki}$ , and the initial state probability vectors of all resources are (0,1).

- (5) The initial resource cost of the project is fixed; in other words, it will not decrease or increase because of the unavailability of resources. Thus, the objective function does not consider the initial sunk cost of the resource and only considers additional costs.

Mathematically, the MSRCPSPP with uncertainty in resource availability can be conceptually formulated as follows.

$$\text{Min} \sum_{k=1}^K c_{s_k} R_{s_k,t} + eM \tag{2}$$

$$S_j - S_g \geq d_j \quad (j \in V, g \in \text{SUCC}_j) \tag{3}$$

$$S_{n+1} \leq T \tag{4}$$

$$\sum_{t \in [S_j, S_j + d_j]} x_{j,t} = d_j \quad (j \in V) \tag{5}$$

$$\sum_{t \in (0, S_j) \cup (S_j + d_j, S_{n+1})} x_{j,t} = 0 \quad (j \in V) \tag{6}$$

$$\sum_{t \in T} t(x_{j,t} - x_{j,t-1}) = S_j \quad (j \in V) \tag{7}$$

$$\sum_{s_k \in S_k} \sum_{j \in V} x_{r_{ki},j,s_k,t} \leq 1 \quad (k \in K, t \in T) \tag{8}$$

$$\sum_{j \in V} r_{j,s_k} \leq \sum_{r_{ki} \in R_k} x_{r_{ki},s_k} z_t^{r_{ki}} x_{r_{ki},j,s_k,t} + R_{s_k,t} \quad (k \in K, s_k \in S_k, t \in T) \tag{9}$$

$$u_{t+1}^{r_{ki}} = u_t^{r_{ki}} P_{r_{ki}} \quad (r_{ki} \in R_k, k \in K, t \in T) \tag{10}$$

$$x_{j,t}, x_{r_{ki},j,s_k}, x_{r_{ki},j,s_k,t} \in \{0, 1\} \tag{11}$$

$$R_{s_k,t} \geq 0 \tag{12}$$

Objective Function (2) minimizes the cost of additional resources, and a penalty value of  $eM$  is considered if scheduling is infeasible. Constraint (3) ensures that the precedence relations among activities need to be satisfied. Constraint (4) ensures that the makespan of the project should not exceed the deadline. Equations (5) and (6) ensure that the activity cannot be interrupted. Equation (7) is a representation of the activity’s start time. Constraint (8) ensures that every resource can be assigned only in one activity at any time. Constraint (9) guarantees that the activity’s skills need to be met. Equation (10) is the state probability vector of each resource. Equations (11) and (12) describe the domain of decision variables.

#### 4. The GA-PR Algorithm

As the MSRCPCP is NP-Hard [30], the possibility of solving the problem optimally using exact solution procedures is limited by the size of instances. However, real instances of project scheduling problems are considerably large. Therefore, having efficient heuristics for finding good quality solutions is of great relevance, especially when considering the dynamic skill assignment of multi-skilled resources with uncertainties in resource availability.

The proposed model not only optimizes activity scheduling (the start time of each activity) but also resource scheduling (dynamic skill assignment of multi-skilled resources). To solve the new model, we designed a modified genetic algorithm combined with priority rules, called GA-PR. Based on the characteristics of the model, the algorithm is divided into two layers. The outer layer comprises activity scheduling optimization according to genetic algorithm, and the inner layer comprises resource scheduling optimizations according to priority rules.

#### 4.1. The Outer Algorithm-GA

##### Step 1. Initialization of activity scheduling

In the outer layer, activity scheduling is the decision maker. Assuming that the project consists of  $n + 1$  activities, there are  $n + 1$  genes on each chromosome, representing the start time ( $ST$ ) of each activity. The earliest start time ( $ES$ ) of the dummy activity 0 is 1, and the project deadline is set as the latest start time ( $LS$ ) of the dummy activity  $n + 1$ . According to the forward and backward iteration algorithms in the critical path theory (CPM), the start time interval of activity  $j$  is  $[ES_j, LS_j]$ . The initial  $ST_j$  is a random value among  $[ES_j, LS_j]$ .

##### Step 2. Calculation of objective function

The objective is to minimize the additional resource cost. First, if the activity scheduling is subjected to precedence constraints, insert it into the inner algorithm. If not, the fitness value is a relatively large penalty value  $eM$ , where  $M$  is a sufficiently large penalty coefficient, and  $e$  reflects the degree of infeasibility—the degree of violating constraints. All individuals in the population are listed in a descending order of fitness value, and the individual with the minimum fitness value is set as the optimal individual.

##### Step 3. Selection, crossover, and mutation

The binary tournament method is used to select parent individuals from the population. Subsequently, crossover and mutation are carried out to generate the new population. Then the optimal individual is updated. This step is iterated until the maximum number of iterations has been reached, and then the final optimal individual is output. (Parameters, such as crossover probability and mutation probability, are determined after many tests.)

#### 4.2. The Inner Algorithm-PR

Step 1. Identify the unavailability probability  $p_{ki}$  ( $k = 1, 2, \dots, K$ ,  $i = 1, 2, \dots$ ) of each resource and generate the resource state matrix  $Z_t$  based on the Markov process.

##### Step 2. Internal resources ranking

Assume that the project requires  $s_K$  types of skill, and there are  $K$  types of resource. For each type of resource, we rank the internal resource's scheduling order according to their skill number in ascending order, and the internal resource's scheduling order is  $A_k$ . Here, we define the first resource-scheduling priority rule.

**Priority Rule 1: Within each type of resource, the one with the lowest skill number is scheduled preferentially because the one with more skills can replace unavailable resources.**

##### Step 3. External skills ranking

The set of activities that are executed at moment  $t$  is  $O_t$ , generated based on activity scheduling, which is the outer layer's solution. Calculate the total demand of each type of skill at moment  $D_{st}$ , and calculate the total initial supply of each type of skill at moment  $S_{st}$  according to the resources' initial skills and their available state. The gap between  $D_{st}$  and  $S_{st}$  is defined as skill-demand tension  $L_{st}$ . The smaller the gap, the smaller the demand tension of  $s$ . We rank these skills according to the demand tension in ascending order, and the external skill order at moment  $t$  is  $W_t$ .

When the scheduling of the skill with the smallest demand tension is completed, the unscheduled resources with this skill can convert its skill to the next skill that needs to be scheduled and so on. Thus, the demand tension of the next skill can be alleviated. Therefore, here, we define the second skill scheduling priority rule.

**Priority Rule 2: Among different skills, the one with the smaller demand tension is scheduled preferentially.**

#### Step 4. Feasibility analysis of activity scheduling with resource constraints

There are three cases for resource scheduling with constraints.

**Case 1:** The skill demand tensions are all negative, indicating that the initial skill provided by resources can satisfy the demand under the given activity scheduling (see Step 5).

**Case 2:** The skill demand tensions are all positive, indicating that the initial skills provided by resources cannot satisfy the demand under the given activity scheduling. This means that every skill needs to be complemented by additional resources (see Step 6).

**Case 3:** In other cases, it indicates that the initial skills provided by the resources cannot satisfy the demand under the given activity scheduling, but multi-skilled resources may satisfy skill needs through skill switching (see Step 6).

#### Step 5. Resource scheduling without multi-skill

First, the skill with the smallest demand tension is scheduled. The scheduling order of resources with initial skill  $W_{1t}$  is  $A_{W_{1t}}$ . For each resource, it is necessary to determine whether it is available. If available, it is removed from  $A_{W_{1t}}$  and added to the resource profile of the task in  $O_t$ . If it is unavailable, it is removed directly from  $A_{W_{1t}}$ .

#### Step 6. Resource scheduling with multi-skill

The cost of skill is used as the basis of the skill scheduling order; the skill with high cost is scheduled first to satisfy its demand as far as possible and to minimize the cost of additional resources as much as possible. Here, we define the third skill scheduling priority rule.

**Priority rule 3: When the skill's demand tension is positive at a certain moment, the skill with high costs has priority.**

For other cases, based on step 5, after completing the scheduling of skill  $W_{1t}$ , as its demand tension is the smallest, if  $A_{W_{1t}}$  is a non-empty set, then merge  $A_{W_{1t}}$  into  $A_{W_{2t}}$  to schedule skill  $W_{2t}$  and so on. If it is still unable to satisfy the skills demand, to ensure that activity scheduling is feasible, additional resources should be considered, and the project's cost will increase.

#### Step 7. Inner iteration

Steps 4–6 are iterated before resource scheduling is completed. Then, output the final fitness, activity scheduling, and resource scheduling.

The flow chart of GA-PR algorithm is shown in Figure 1.



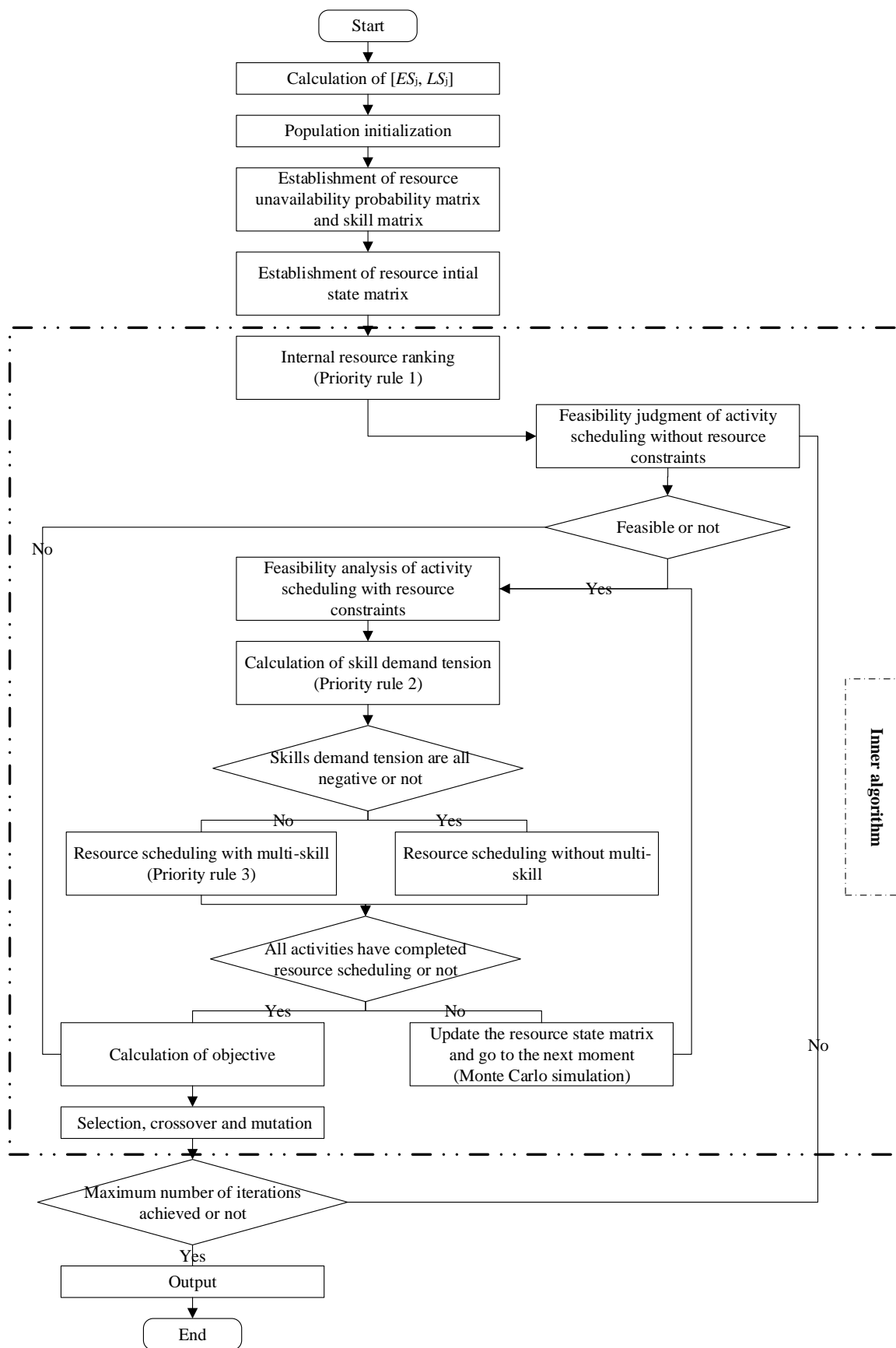


Figure 1. Flowchart of the GA-PR.

### 5. A Numerical Example

A project with 12 activities is defined to illustrate the new model and the new algorithm. The duration of each activity is known, and precedence relations are shown in Figure 2 as finish-to-start relations. Assume that the project needs three types of renewable resources:  $R_1, R_2,$  and  $R_3,$  i.e.,  $K = 3,$  and the total number of each type of resources was considered be 10; then, the set of each type of resources can be described as follows:  $R_1 = \{r_{11}, r_{12}, r_{13}, r_{14}, r_{15}, r_{16}, r_{17}, r_{18}, r_{19}, r_{110}\}; R_2 = \{r_{21}, r_{22}, r_{23}, r_{24}, r_{25}, r_{26}, r_{27}, r_{28}, r_{29}, r_{210}\}; R_3 = \{r_{31}, r_{32}, r_{33}, r_{34}, r_{35}, r_{36}, r_{37}, r_{38}, r_{39}, r_{310}\}.$  All resources in  $R_1$  master initial skill  $s_1,$  all resources in  $R_2$  master initial skill  $s_2,$  all resources in  $R_3$  master initial skill  $s_3,$  and whether these resources master other skill(s) is generated randomly. The resource state matrix is generated by a Markov process (see Table A2 in Appendix A for details). Each activity demands different skills. Table 2 shows the number of skills required for performing activities and other information.

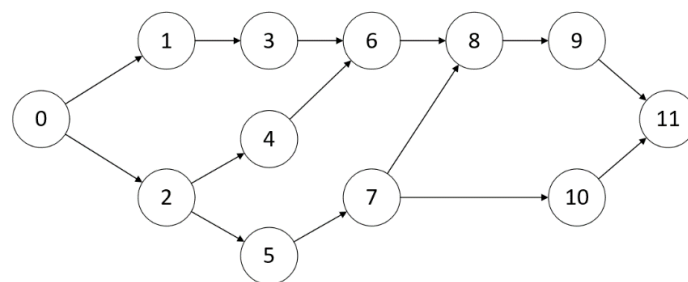


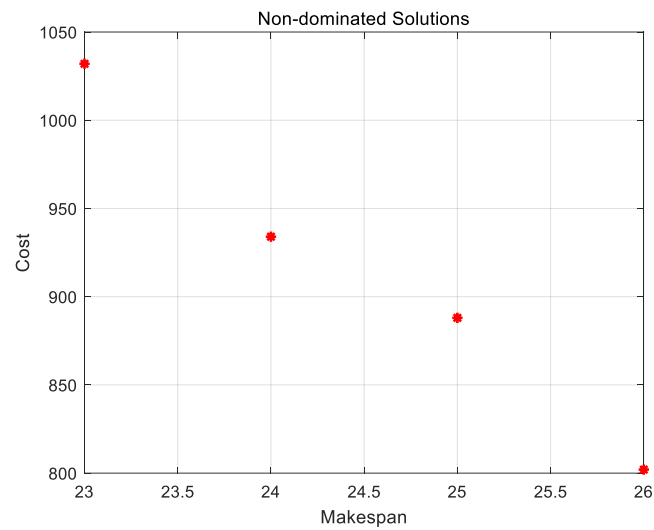
Figure 2. Project Network Example.

Table 2. Project information.

Activists	Successors	$d_i$	$s_1$	$s_2$	$s_3$
0	1, 2	0	0	0	0
1	3	2	5	6	6
2	4, 5	3	7	5	4
3	6	6	6	4	4
4	6	5	4	6	3
5	7	3	7	7	6
6	8	6	5	6	4
7	8, 10	4	8	5	5
8	9	5	3	4	3
9	11	4	6	5	4
10	11	2	6	3	6
11	-	0	0	0	0
	$c_{s_k}$		10	8	12

#### 5.1. The Effectiveness of Dynamic Scheduling

According to the critical path method, the shortest makespan of the project is 23. Considering the unavailability of resources, we assume that the deadline of the project is 26 (any number greater than 23 is allowed) (that is,  $T = 26$ ). According to the GA-PR algorithm, the optimized scheduling scheme can be obtained, as shown in Figure 3 and Table 3. Taking scheme 1 as an example, the detailed project scheduling is shown in Figure 4 and the detailed resource scheduling is shown in Table 4.

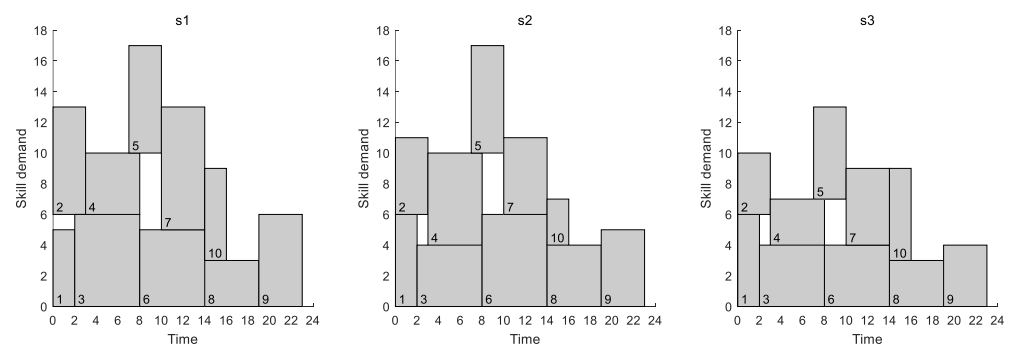


**Figure 3.** The schedule solution. Note: The red dots indicate the makespan and additional cost of 4 scheduling schemes respectively.

**Table 3.** Optimization Schedules.

Scheme	Makespan	Objective Value (Costs of Temporary Resources)	Schedule
1	23	1032	[0, 1, 1, 3, 4, 8, 9, 11, 15, 20, 15, 23]
2	24	934	[0, 1, 2, 3, 5, 9, 10, 12, 16, 21, 18, 24]
3	25	888	[0, 3, 1, 5, 4, 9, 11, 12, 17, 22, 21, 25]
4	26	802	[0, 3, 1, 5, 7, 4, 12, 12, 18, 23, 21, 26]

According to Figure 3, it can be found that with the extension of the project deadline, the cost of the additional resources becomes increasingly smaller. In other words, the buffer period plays a role in alleviating resource unavailability. Resource scheduling in Table 3 shows that multiple resources have performed more than one skill. This gives scheduling more flexibility. Therefore, the dynamic scheduling considering multi-skilled resources can effectively alleviate the impact of uncertainty in resource availability on a project’s makespan and cost.



**Figure 4.** The scheduling of scheme 1.

**Table 4.** Multi-skilled resource scheduling of scheme 1.

<i>j</i>	<i>t</i>	<i>s</i> <sub>1</sub>	<i>s</i> <sub>2</sub>	<i>s</i> <sub>3</sub>
1	1	<i>r</i> <sub>11</sub> ; <i>r</i> <sub>12</sub> ; <i>r</i> <sub>14</sub> ; <i>r</i> <sub>15</sub> ; <i>r</i> <sub>110</sub>	<i>r</i> <sub>21</sub> ; <i>r</i> <sub>22</sub> ; <i>r</i> <sub>23</sub> ; <i>r</i> <sub>24</sub> ; <i>r</i> <sub>27</sub> ; <i>r</i> <sub>29</sub>	<i>r</i> <sub>32</sub> ; <i>r</i> <sub>33</sub> ; <i>r</i> <sub>34</sub> ; <i>r</i> <sub>36</sub> ; <i>r</i> <sub>37</sub> ; <i>r</i> <sub>39</sub>
	2	<i>r</i> <sub>14</sub> ; <i>r</i> <sub>17</sub> ; <i>r</i> <sub>110</sub> ; 2 temporary	<i>r</i> <sub>22</sub> ; <i>r</i> <sub>29</sub> ; 4 temporary	<i>r</i> <sub>32</sub> ; <i>r</i> <sub>35</sub> ; <i>r</i> <sub>37</sub> ; <i>r</i> <sub>38</sub> ; <i>r</i> <sub>13</sub> ; <i>r</i> <sub>210</sub>
2	1	<i>r</i> <sub>13</sub> ; <i>r</i> <sub>16</sub> ; <i>r</i> <sub>17</sub> ; <i>r</i> <sub>18</sub> ; <i>r</i> <sub>19</sub> ; 2 temporary	<i>r</i> <sub>25</sub> ; <i>r</i> <sub>26</sub> ; <i>r</i> <sub>28</sub> ; <i>r</i> <sub>210</sub> ; 1 temporary	<i>r</i> <sub>31</sub> ; <i>r</i> <sub>35</sub> ; <i>r</i> <sub>38</sub> ; <i>r</i> <sub>310</sub>
	2	<i>r</i> <sub>16</sub> ; <i>r</i> <sub>18</sub> ; <i>r</i> <sub>19</sub> ; <i>r</i> <sub>24</sub> ; <i>r</i> <sub>25</sub> ; <i>r</i> <sub>28</sub> ; 1 temporary	5 temporary	<i>r</i> <sub>33</sub> ; <i>r</i> <sub>36</sub> ; <i>r</i> <sub>310</sub> ; <i>r</i> <sub>11</sub>
	3	<i>r</i> <sub>12</sub> ; <i>r</i> <sub>14</sub> ; <i>r</i> <sub>17</sub> ; <i>r</i> <sub>110</sub> ; 3 temporary	<i>r</i> <sub>22</sub> ; <i>r</i> <sub>23</sub> ; <i>r</i> <sub>29</sub> ; <i>r</i> <sub>210</sub> ; 1 temporary	<i>r</i> <sub>13</sub> ; <i>r</i> <sub>34</sub> ; <i>r</i> <sub>37</sub> ; <i>r</i> <sub>38</sub>
3	3	<i>r</i> <sub>11</sub> ; <i>r</i> <sub>15</sub> ; <i>r</i> <sub>18</sub> ; <i>r</i> <sub>19</sub> ; <i>r</i> <sub>25</sub> ; <i>r</i> <sub>28</sub>	<i>r</i> <sub>27</sub> ; 3 temporary	<i>r</i> <sub>31</sub> ; <i>r</i> <sub>33</sub> ; <i>r</i> <sub>36</sub> ; <i>r</i> <sub>39</sub>
	4	<i>r</i> <sub>11</sub> ; <i>r</i> <sub>12</sub> ; <i>r</i> <sub>14</sub> ; <i>r</i> <sub>15</sub> ; <i>r</i> <sub>17</sub> ; <i>r</i> <sub>110</sub>	<i>r</i> <sub>21</sub> ; <i>r</i> <sub>22</sub> ; <i>r</i> <sub>23</sub> ; <i>r</i> <sub>29</sub>	<i>r</i> <sub>32</sub> ; <i>r</i> <sub>34</sub> ; <i>r</i> <sub>36</sub> ; <i>r</i> <sub>39</sub>
	5	<i>r</i> <sub>11</sub> ; <i>r</i> <sub>12</sub> ; <i>r</i> <sub>15</sub> ; <i>r</i> <sub>17</sub> ; <i>r</i> <sub>18</sub> ; <i>r</i> <sub>110</sub>	<i>r</i> <sub>21</sub> ; <i>r</i> <sub>22</sub> ; <i>r</i> <sub>27</sub> ; <i>r</i> <sub>29</sub>	<i>r</i> <sub>34</sub> ; <i>r</i> <sub>36</sub> ; <i>r</i> <sub>37</sub> ; <i>r</i> <sub>39</sub>
	6	<i>r</i> <sub>12</sub> ; <i>r</i> <sub>14</sub> ; <i>r</i> <sub>17</sub> ; <i>r</i> <sub>110</sub> ; <i>r</i> <sub>26</sub> ; 1 temporary	<i>r</i> <sub>22</sub> ; <i>r</i> <sub>29</sub> ; <i>r</i> <sub>210</sub> ; 1 temporary	<i>r</i> <sub>23</sub> ; <i>r</i> <sub>32</sub> ; <i>r</i> <sub>37</sub> ; <i>r</i> <sub>38</sub>
	7	<i>r</i> <sub>12</sub> ; <i>r</i> <sub>13</sub> ; <i>r</i> <sub>14</sub> ; <i>r</i> <sub>15</sub> ; <i>r</i> <sub>18</sub> ; <i>r</i> <sub>19</sub>	<i>r</i> <sub>21</sub> ; <i>r</i> <sub>22</sub> ; <i>r</i> <sub>27</sub> ; <i>r</i> <sub>29</sub>	<i>r</i> <sub>32</sub> ; <i>r</i> <sub>36</sub> ; <i>r</i> <sub>37</sub> ; <i>r</i> <sub>39</sub>
	8	<i>r</i> <sub>13</sub> ; <i>r</i> <sub>16</sub> ; <i>r</i> <sub>110</sub> ; <i>r</i> <sub>28</sub> ; 2 temporary	<i>r</i> <sub>23</sub> ; <i>r</i> <sub>27</sub> ; 2 temporary	<i>r</i> <sub>34</sub> ; <i>r</i> <sub>35</sub> ; <i>r</i> <sub>39</sub> ; <i>r</i> <sub>310</sub>
4	4	<i>r</i> <sub>13</sub> ; <i>r</i> <sub>18</sub> ; <i>r</i> <sub>19</sub> ; <i>r</i> <sub>38</sub>	<i>r</i> <sub>25</sub> ; <i>r</i> <sub>26</sub> ; <i>r</i> <sub>28</sub> ; <i>r</i> <sub>210</sub> ; <i>r</i> <sub>310</sub> ; 1 temporary	<i>r</i> <sub>31</sub> ; <i>r</i> <sub>33</sub> ; <i>r</i> <sub>35</sub>
	5	<i>r</i> <sub>13</sub> ; <i>r</i> <sub>19</sub> ; <i>r</i> <sub>310</sub> ; 1 temporary	<i>r</i> <sub>23</sub> ; <i>r</i> <sub>24</sub> ; <i>r</i> <sub>26</sub> ; <i>r</i> <sub>28</sub> ; <i>r</i> <sub>210</sub> ; <i>r</i> <sub>38</sub>	<i>r</i> <sub>32</sub> ; <i>r</i> <sub>33</sub> ; <i>r</i> <sub>35</sub>
	6	<i>r</i> <sub>16</sub> ; <i>r</i> <sub>18</sub> ; <i>r</i> <sub>19</sub> ; <i>r</i> <sub>21</sub>	<i>r</i> <sub>25</sub> ; <i>r</i> <sub>27</sub> ; <i>r</i> <sub>28</sub> ; 3 temporary	<i>r</i> <sub>31</sub> ; <i>r</i> <sub>33</sub> ; <i>r</i> <sub>36</sub>
	7	4 temporary	<i>r</i> <sub>23</sub> ; <i>r</i> <sub>25</sub> ; <i>r</i> <sub>26</sub> ; <i>r</i> <sub>28</sub> ; <i>r</i> <sub>38</sub> ; <i>r</i> <sub>310</sub>	<i>r</i> <sub>31</sub> ; <i>r</i> <sub>33</sub> ; <i>r</i> <sub>35</sub>
5	8	<i>r</i> <sub>14</sub> ; <i>r</i> <sub>17</sub> ; <i>r</i> <sub>19</sub> ; <i>r</i> <sub>24</sub>	<i>r</i> <sub>29</sub> ; <i>r</i> <sub>210</sub> ; 4 temporary	<i>r</i> <sub>11</sub> ; <i>r</i> <sub>31</sub> ; <i>r</i> <sub>36</sub>
	8	<i>r</i> <sub>15</sub> ; <i>r</i> <sub>26</sub> ; 5 temporary	<i>r</i> <sub>22</sub> ; 6 temporary	<i>r</i> <sub>18</sub> ; <i>r</i> <sub>25</sub> ; <i>r</i> <sub>33</sub> ; <i>r</i> <sub>38</sub> ; 2 temporary
	9	<i>r</i> <sub>12</sub> ; <i>r</i> <sub>14</sub> ; <i>r</i> <sub>17</sub> ; <i>r</i> <sub>110</sub> ; <i>r</i> <sub>26</sub> ; 2 temporary	<i>r</i> <sub>22</sub> ; <i>r</i> <sub>29</sub> ; <i>r</i> <sub>210</sub> ; 4 temporary	<i>r</i> <sub>13</sub> ; <i>r</i> <sub>23</sub> ; <i>r</i> <sub>32</sub> ; <i>r</i> <sub>35</sub> ; <i>r</i> <sub>37</sub> ; <i>r</i> <sub>38</sub>
	10	<i>r</i> <sub>14</sub> ; <i>r</i> <sub>17</sub> ; <i>r</i> <sub>110</sub> ; <i>r</i> <sub>26</sub> ; 3 temporary	<i>r</i> <sub>22</sub> ; <i>r</i> <sub>29</sub> ; 5 temporary	<i>r</i> <sub>13</sub> ; <i>r</i> <sub>23</sub> ; <i>r</i> <sub>37</sub> ; <i>r</i> <sub>38</sub> ; 2 temporary
	9	<i>r</i> <sub>11</sub> ; <i>r</i> <sub>15</sub> ; <i>r</i> <sub>18</sub> ; <i>r</i> <sub>19</sub> ; <i>r</i> <sub>310</sub>	<i>r</i> <sub>21</sub> ; <i>r</i> <sub>25</sub> ; <i>r</i> <sub>27</sub> ; 3 temporary	<i>r</i> <sub>31</sub> ; <i>r</i> <sub>33</sub> ; <i>r</i> <sub>36</sub> ; <i>r</i> <sub>39</sub>
6	10	<i>r</i> <sub>15</sub> ; <i>r</i> <sub>18</sub> ; <i>r</i> <sub>19</sub> ; <i>r</i> <sub>24</sub> ; <i>r</i> <sub>28</sub>	<i>r</i> <sub>25</sub> ; <i>r</i> <sub>27</sub> ; 4 temporary	<i>r</i> <sub>11</sub> ; <i>r</i> <sub>33</sub> ; <i>r</i> <sub>36</sub> ; <i>r</i> <sub>310</sub>
	11	<i>r</i> <sub>11</sub> ; <i>r</i> <sub>12</sub> ; <i>r</i> <sub>15</sub> ; <i>r</i> <sub>15</sub> ; <i>r</i> <sub>17</sub>	<i>r</i> <sub>21</sub> ; <i>r</i> <sub>22</sub> ; <i>r</i> <sub>23</sub> ; <i>r</i> <sub>24</sub> ; <i>r</i> <sub>27</sub> ; <i>r</i> <sub>29</sub>	<i>r</i> <sub>34</sub> ; <i>r</i> <sub>36</sub> ; <i>r</i> <sub>37</sub> ; <i>r</i> <sub>39</sub>
	12	<i>r</i> <sub>12</sub> ; <i>r</i> <sub>14</sub> ; <i>r</i> <sub>17</sub> ; <i>r</i> <sub>110</sub> ; <i>r</i> <sub>26</sub>	<i>r</i> <sub>210</sub> ; 5 temporary	<i>r</i> <sub>37</sub> ; <i>r</i> <sub>38</sub> ; <i>r</i> <sub>23</sub> ; <i>r</i> <sub>13</sub>
	13	<i>r</i> <sub>11</sub> ; <i>r</i> <sub>12</sub> ; <i>r</i> <sub>14</sub> ; <i>r</i> <sub>16</sub> ; <i>r</i> <sub>17</sub>	<i>r</i> <sub>22</sub> ; <i>r</i> <sub>23</sub> ; <i>r</i> <sub>28</sub> ; <i>r</i> <sub>29</sub> ; 2 temporary	<i>r</i> <sub>32</sub> ; <i>r</i> <sub>34</sub> ; <i>r</i> <sub>36</sub> ; <i>r</i> <sub>37</sub>
	14	<i>r</i> <sub>12</sub> ; <i>r</i> <sub>14</sub> ; <i>r</i> <sub>15</sub> ; <i>r</i> <sub>17</sub> ; <i>r</i> <sub>110</sub>	<i>r</i> <sub>21</sub> ; <i>r</i> <sub>22</sub> ; <i>r</i> <sub>23</sub> ; <i>r</i> <sub>24</sub> ; <i>r</i> <sub>27</sub> ; <i>r</i> <sub>29</sub>	<i>r</i> <sub>34</sub> ; <i>r</i> <sub>36</sub> ; <i>r</i> <sub>37</sub> ; <i>r</i> <sub>39</sub>
7	11	<i>r</i> <sub>13</sub> ; <i>r</i> <sub>18</sub> ; <i>r</i> <sub>19</sub> ; 5 temporary	<i>r</i> <sub>25</sub> ; 4 temporary	<i>r</i> <sub>31</sub> ; <i>r</i> <sub>32</sub> ; <i>r</i> <sub>35</sub> ; <i>r</i> <sub>38</sub> ; <i>r</i> <sub>310</sub>
	12	<i>r</i> <sub>11</sub> ; <i>r</i> <sub>15</sub> ; <i>r</i> <sub>16</sub> ; <i>r</i> <sub>18</sub> ; <i>r</i> <sub>19</sub> ; <i>r</i> <sub>21</sub> ; <i>r</i> <sub>24</sub> ; <i>r</i> <sub>25</sub>	<i>r</i> <sub>27</sub> ; 4 temporary	<i>r</i> <sub>31</sub> ; <i>r</i> <sub>33</sub> ; <i>r</i> <sub>36</sub> ; <i>r</i> <sub>39</sub> ; <i>r</i> <sub>310</sub>
	13	<i>r</i> <sub>13</sub> ; <i>r</i> <sub>18</sub> ; <i>r</i> <sub>19</sub> ; 5 temporary	5 temporary	<i>r</i> <sub>31</sub> ; <i>r</i> <sub>33</sub> ; <i>r</i> <sub>35</sub> ; <i>r</i> <sub>38</sub> ; <i>r</i> <sub>310</sub>
	14	<i>r</i> <sub>13</sub> ; <i>r</i> <sub>18</sub> ; <i>r</i> <sub>19</sub> ; 5 temporary	<i>r</i> <sub>25</sub> ; <i>r</i> <sub>210</sub> ; <i>r</i> <sub>310</sub> ; 2 temporary	<i>r</i> <sub>31</sub> ; <i>r</i> <sub>32</sub> ; <i>r</i> <sub>33</sub> ; <i>r</i> <sub>35</sub> ; <i>r</i> <sub>38</sub>
8	15	<i>r</i> <sub>11</sub> ; <i>r</i> <sub>14</sub> ; <i>r</i> <sub>110</sub>	<i>r</i> <sub>22</sub> ; <i>r</i> <sub>23</sub> ; <i>r</i> <sub>27</sub> ; <i>r</i> <sub>29</sub>	<i>r</i> <sub>36</sub> ; <i>r</i> <sub>37</sub> ; <i>r</i> <sub>310</sub>
	16	<i>r</i> <sub>12</sub> ; <i>r</i> <sub>15</sub> ; <i>r</i> <sub>110</sub>	<i>r</i> <sub>22</sub> ; <i>r</i> <sub>26</sub> ; <i>r</i> <sub>27</sub> ; <i>r</i> <sub>29</sub>	<i>r</i> <sub>34</sub> ; <i>r</i> <sub>36</sub> ; <i>r</i> <sub>37</sub>
	17	<i>r</i> <sub>14</sub> ; <i>r</i> <sub>15</sub> ; <i>r</i> <sub>110</sub>	<i>r</i> <sub>22</sub> ; <i>r</i> <sub>23</sub> ; <i>r</i> <sub>24</sub> ; <i>r</i> <sub>29</sub>	<i>r</i> <sub>34</sub> ; <i>r</i> <sub>36</sub> ; <i>r</i> <sub>37</sub>
	18	<i>r</i> <sub>14</sub> ; <i>r</i> <sub>15</sub> ; <i>r</i> <sub>110</sub>	<i>r</i> <sub>21</sub> ; <i>r</i> <sub>22</sub> ; <i>r</i> <sub>27</sub> ; <i>r</i> <sub>29</sub>	<i>r</i> <sub>34</sub> ; <i>r</i> <sub>36</sub> ; <i>r</i> <sub>37</sub>
	19	<i>r</i> <sub>14</sub> ; <i>r</i> <sub>15</sub> ; <i>r</i> <sub>110</sub>	<i>r</i> <sub>21</sub> ; <i>r</i> <sub>22</sub> ; <i>r</i> <sub>27</sub> ; <i>r</i> <sub>29</sub>	<i>r</i> <sub>36</sub> ; <i>r</i> <sub>37</sub> ; <i>r</i> <sub>310</sub>
9	20	<i>r</i> <sub>13</sub> ; <i>r</i> <sub>15</sub> ; <i>r</i> <sub>17</sub> ; <i>r</i> <sub>18</sub> ; <i>r</i> <sub>19</sub> ; <i>r</i> <sub>25</sub>	<i>r</i> <sub>21</sub> ; <i>r</i> <sub>22</sub> ; <i>r</i> <sub>23</sub> ; <i>r</i> <sub>24</sub> ; <i>r</i> <sub>27</sub>	<i>r</i> <sub>32</sub> ; <i>r</i> <sub>34</sub> ; <i>r</i> <sub>36</sub> ; <i>r</i> <sub>37</sub>
	21	<i>r</i> <sub>11</sub> ; <i>r</i> <sub>12</sub> ; <i>r</i> <sub>14</sub> ; <i>r</i> <sub>15</sub> ; <i>r</i> <sub>16</sub> ; <i>r</i> <sub>17</sub>	<i>r</i> <sub>21</sub> ; <i>r</i> <sub>22</sub> ; <i>r</i> <sub>23</sub> ; <i>r</i> <sub>24</sub> ; <i>r</i> <sub>29</sub>	<i>r</i> <sub>31</sub> ; <i>r</i> <sub>32</sub> ; <i>r</i> <sub>33</sub> ; <i>r</i> <sub>37</sub>
	22	<i>r</i> <sub>11</sub> ; <i>r</i> <sub>12</sub> ; <i>r</i> <sub>14</sub> ; <i>r</i> <sub>15</sub> ; <i>r</i> <sub>16</sub> ; <i>r</i> <sub>110</sub>	<i>r</i> <sub>21</sub> ; <i>r</i> <sub>22</sub> ; <i>r</i> <sub>23</sub> ; <i>r</i> <sub>24</sub> ; <i>r</i> <sub>29</sub>	<i>r</i> <sub>34</sub> ; <i>r</i> <sub>36</sub> ; <i>r</i> <sub>37</sub> ; <i>r</i> <sub>39</sub>
	23	<i>r</i> <sub>11</sub> ; <i>r</i> <sub>12</sub> ; <i>r</i> <sub>15</sub> ; <i>r</i> <sub>16</sub> ; <i>r</i> <sub>17</sub> ; <i>r</i> <sub>110</sub>	<i>r</i> <sub>21</sub> ; <i>r</i> <sub>22</sub> ; <i>r</i> <sub>23</sub> ; <i>r</i> <sub>27</sub> ; <i>r</i> <sub>29</sub>	<i>r</i> <sub>34</sub> ; <i>r</i> <sub>36</sub> ; <i>r</i> <sub>37</sub> ; <i>r</i> <sub>39</sub>
10	15	<i>r</i> <sub>12</sub> ; <i>r</i> <sub>13</sub> ; <i>r</i> <sub>16</sub> ; <i>r</i> <sub>17</sub> ; <i>r</i> <sub>18</sub> ; <i>r</i> <sub>19</sub>	<i>r</i> <sub>26</sub> ; <i>r</i> <sub>28</sub> ; <i>r</i> <sub>210</sub>	<i>r</i> <sub>25</sub> ; <i>r</i> <sub>31</sub> ; <i>r</i> <sub>33</sub> ; <i>r</i> <sub>38</sub> ; <i>r</i> <sub>310</sub> ; 1 temporary
	16	<i>r</i> <sub>13</sub> ; <i>r</i> <sub>16</sub> ; <i>r</i> <sub>17</sub> ; <i>r</i> <sub>18</sub> ; <i>r</i> <sub>19</sub> ; 1 temporary	<i>r</i> <sub>25</sub> ; <i>r</i> <sub>210</sub> ; <i>r</i> <sub>310</sub>	<i>r</i> <sub>31</sub> ; <i>r</i> <sub>32</sub> ; <i>r</i> <sub>33</sub> ; <i>r</i> <sub>35</sub> ; <i>r</i> <sub>38</sub> ; <i>r</i> <sub>39</sub>

### 5.2. Comparison of Three Scheduling Methods

As shown in Section 4, we know that scheduling multi-skilled resources is mainly determined by the priority rules in the inner algorithm, which we call dynamic scheduling. To further analyze the effectiveness of these priority rules, we set up two groups of experiments to compare with the dynamic scheduling method proposed in this paper—random scheduling and static scheduling. Random scheduling means that the priority rules designed in Section 4.2 and the skill scheduling order and resource scheduling order are random. Static scheduling means that once a resource is assigned a skill, it cannot be changed. The pseudo-codes of dynamic, random and static scheduling are shown in Appendix B, and the results relative to three different scheduling methods are shown in Figure 5a–c, respectively. The figures indicate that the objective of dynamic scheduling is the best: There is minimum additional cost, followed by random scheduling and static scheduling. This is because of the design of scheduling priority rules. This proves the effectiveness of the proposed dynamic scheduling method. The operation times of the three scheduling methods are 19.44 s, 16.39 s, and 0.57 s, respectively.

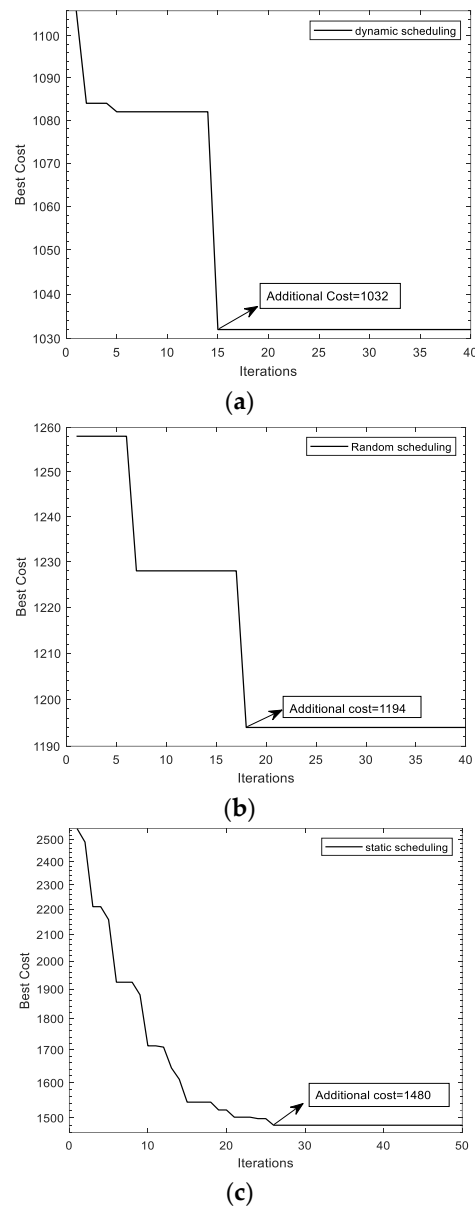


Figure 5. (a) Dynamic scheduling. (b) Random scheduling. (c) Static scheduling.

### 5.3. The Effect of the Buffer Period

We can know that a project’s deadline affects the objective value from the results of Figure 3. Therefore, we define parameter  $C_{deadline}$  to represent the buffer coefficient, which is used to reflect the margin of buffer duration interval— $C_{deadline} = T/DueDate$ , and the *DueDate* refers to the makspan of the critical path of the project. The larger  $C_{deadline}$  is, the larger the buffer period. The objective values of three scheduling methods with different parameters are shown in Table 5.

**Table 5.** The effect of the buffer period.

Parameter	Dynamic	Random	Static	
	Best Cost			
$C_{deadline}$	1.1	952	1154	1076
	1.3	538	802	556
	1.5	364	476	402
	1.7	250	406	270

Table 5 shows that with the increase in buffer period, the objective values decrease for three different scheduling programs. This is because when the activities cannot be started due to the unavailable resources, the larger the buffer period, the more likely that activity can be allowed to delay, and fewer additional resources will be added. Thus, the cost will be reduced.

## 6. Experiment Analysis

In this section, a computational experiment is designed to assess the performance of the heuristic algorithm proposed in this paper. The algorithm was coded in Matlab R2018b and ran in the environment of Microsoft Windows 10 (CPU 1.68 GHZ, RAM 8 GB).

### 6.1. Test Data

Considering that the resources are multi-skilled and their availabilities are uncertain, the following changes are made to the dataset from PSPLIB (<http://www.om-db.wi.tum.de/psplib/>, accessed on 17 June 2020):

- (1) The number of skill types required by the project corresponds to that of resource types required by the project in the original PSPLIB. Each type of resource has a corresponding initial skill. Assuming that the original PSPLIB J30 needs four types of resources ( $R_1, R_2, R_3,$  and  $R_4$ ), the project in this paper needs four types of skills in which all resources in  $R_1$  have initial skill  $s_1$ , those in  $R_2$  have initial skill  $s_1$ , and so on. Except for the initial skills, whether every resource has other skill(s) is randomly generated.
- (2) The skill requirements of activities correspond to the resource requirements in PSPLIB—assuming that the resource requirements of  $R_1, R_2, R_3,$  and  $R_4$  for activity 1 in a case of PSPLIB is 4, 5, 7, and 8, respectively, then the skill requirements of  $s_1, s_2, s_3,$  and  $s_4$  for activity 1 is 4, 5, 7, and 8 in this paper.

Network complexity (*NC*) reflects the precedence relations of activities. Resource Strength (*RS*) reflects the intensity of resources, where the larger the value is, the more resources are supplied. Resource Factor (*RF*) reflects the activity’s skill requirement, where the larger the value is, the more skills are needed. In addition, we defined Modified Resource Strength (*MRS*) based on the Resource Strength (*RS*), which reflects the skill strength mastered by resources. The formula of *MRS* is shown in Equation (13). The larger the value, the more the skills are mastered by resources. The Rate of Resource Unavailability (*RRU*) is introduced to reflect the unavailability of resources. The larger the value, the greater the probability that the resource is unavailable.

$$MRS = \frac{(\sum_i \sum_{k=1}^K X_{k i s_k}) / (\sum_i \sum_{k=1}^K r_{ki})}{K} \tag{13}$$

As the problem studied in this paper adds uncertainty in resource availability and dynamic skill assignment to the classical MSRCPS, this improves scheduling flexibility while increasing the difficulty for solutions. As such, solution times increase exponentially. Therefore, this paper only selected the J30-dataset from the PSPLIB. The parameters of dataset are shown in Table 6 below.

**Table 6.** Parameters and values.

Parameters	Values				
NC	1.5	1.8	2.1	—	—
RS	0.2	0.5	0.7	1	—
RF	0.25	0.5	0.75	1	—
MRS	0.25	0.625	1	—	—
RRU	0.1	0.2	0.3	0.4	0.5

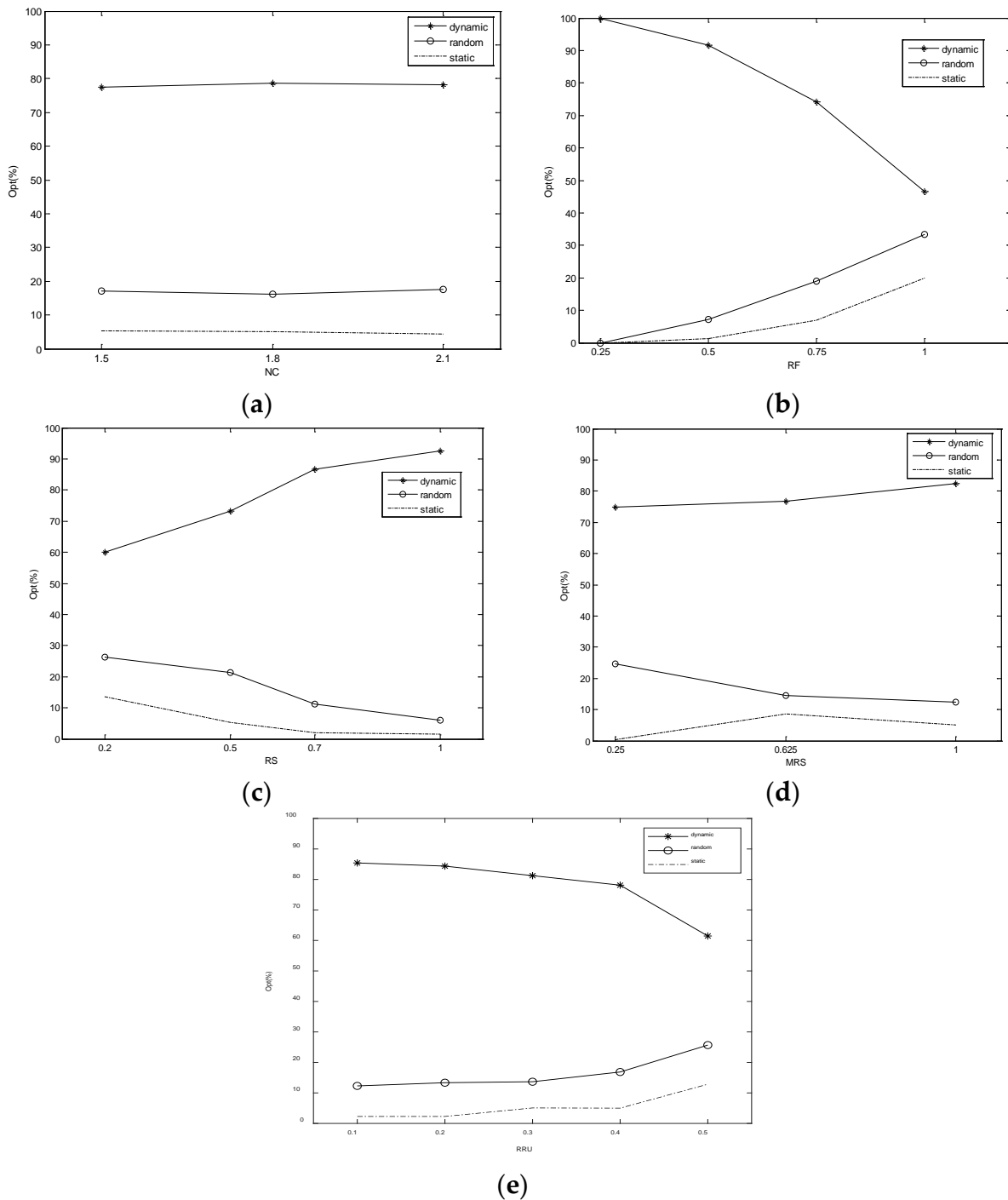
### 6.2. Computational Results

In this section, dynamic, random, and static scheduling are compared based on the same dataset. The comparison results under different parameters are shown in Figure 6a–e, where *Opt*(%) is the proportion of the optimal solution. The optimal solution refers to the minimum cost of the three scheduling methods (dynamic, stochastic, and static). The proportion of the optimal solution refers to that of the number of optimal solutions in all 480 instances.

As Figure 6a–e indicate, regardless of how parameters change, the performance of dynamic scheduling is always superior than random and static scheduling. Moreover, compared to static scheduling, random scheduling is superior. This is because static scheduling limits the resource’s skill switching; thus, more additional resources need to be supplied to reach skill requirements, leading to an increase in costs. This shows that multi-skilled scheduling can effectively alleviate the disturbance caused by resource uncertainty, and the design of priority rules in dynamic scheduling leads to improved optimization results, which further shows the effectiveness of the proposed algorithm.

As shown in Figure 6a, network complexity (*NC*) has no obvious impact on the results of the three scheduling schemes. This is because the three types of scheduling mainly optimize results from the perspective of resources and skills. Therefore, regardless of how *NC* changes, the results are relatively stable. With the increase in Resource factor (*RF*), the proportion of optimal solution of dynamic scheduling gradually decreases, as shown in Figure 6b. With limited resources, when the activity skill requirements increase, project scheduling is more easily affected by resources. With the increase in skill requirements, the flexibility of multi-skilled resources is limited. As such, more additional resources are needed, which leads to an increase in costs. Therefore, compared with random and static scheduling, the optimal proportion of the target value decreases. As observed in Figure 6c, with the increase in *RS*, the proportion of the optimal solution of dynamic scheduling gradually increases. This is because the greater *RS* is, the more resources are likely to meet the skill requirements of activities, and the replacement of idle resources is easier. Thus, there is less necessity for additional resources, and costs will be less. In other words, the increase in *RS* weakens the impact of uncertainty in resource availability. Similarly, *MRS* reflects the strength of skills. The greater the *MRS*, the more skills the resources can master. There is more flexibility in resources scheduling, and it is more likely to produce improved solutions compared with static scheduling. Consequently, costs are lower, as shown in Figure 6d. As *RRU* reflects the state of resources, the higher the value, the greater the probability of resource unavailability. At this time, the role of multi-skilled resources will be weakened, and there are fewer resources that can meet skill requirements. Therefore, with

the increase in  $RRU$ , more additional resources are needed, and the optimal proportion of dynamic scheduling will also decrease, as shown in Figure 6e.



**Figure 6.** (a) Comparison results of  $NC$ . (b) Comparison results of  $RF$ . (c) Comparison results of  $RS$ . (d) Comparison results of  $MRS$ . (e) Comparison results of  $RRU$ .

Therefore, as a project manager, to reduce the additional cost of the project in an uncertain environment, it is necessary to improve the multi-skilled level of resources, avoid the unavailability of resources, and pay attention to the order in which skills and resources are scheduled, which have certain guiding significance for maximizing project benefits.



## 7. Conclusions

This paper studies the MSRCPS with uncertainty in resource availability, and dynamic skill assignment and additional resource replenishment are considered simultaneously when resources are not available. Although extensive research has been conducted for the MSRCPS, little research focused on uncertainty in resource availability. In an uncertain environment, it is easy to encounter resource shortage or conflict. Idle resources can replace unavailable resources to complete activities through skill switching; thus, dynamic skill assignment has become a method for alleviating resource conflict. If it still does not work, additional resources are considered to ensure that activities are not interrupted. To solve this complex problem, a new model is built and a nested GA-PR dynamic scheduling algorithm is proposed. Finally, an example and numerical experiments are used to verify the performance of the algorithm. Simultaneously, the performance of the algorithm is illustrated by comparing the solution efficiency of static and random scheduling, and the influences of different parameters on the algorithm scheduling are analyzed.

Numerical experiments show that, although the running time of the proposed dynamic scheduling is not optimal, its solution is always superior compared to the other two scheduling methods. When uncertainties in resource availability are considered, the resource has more skills and more resources are supplied, and the dynamic scheduling method has improved performance; on the other hand, the higher the probability of resource unavailability and the more skills are required, the worse the dynamic scheduling method performs. Moreover, by comparing the performance of dynamic scheduling and random scheduling, we can find that the scheduling order has a significant impact on the results and the three new priority rules contribute to the optimization of costs. By comparing the performance of dynamic scheduling and static scheduling, we can find that the skill switching of multi-skill resources also plays an important role.

Our research findings can also provide project managers with some guidance when scheduling projects in an uncertain environment. First, at the start-up stage of the project, managers should select as many multi-skilled resources as possible when establishing the project team. Second, at the project planning stage, it is important to decide which skill should be scheduled first and which resource should be scheduled first. Moreover, dynamic skill assignment and additional resource replenishment are great methods for alleviating resource shortages. Third, during the project-execution period, managers can take some incentives to encourage single-skilled person to learn from multi-skilled persons to master more skills.

However, it should be noted that this paper has still some limitations. First, this paper assumes that the skill level of each resource is homogeneous. For future research, the heterogeneity of skill level can be considered. Second, when the splitting of activity is allowed, resource conflicts can also be solved by interrupting activities so as to reduce the additional cost of resources. In this case, scheduling will be a more interesting and difficult problem. Finally, in this paper, the costs of resource skill switching are not taken into account, which will render MSRCPS a trade-off problem between additional resource costs and skill-switching costs.

**Author Contributions:** All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by G.L., M.W. and X.L. The first draft of the manuscript was written by M.W., and all authors commented on previous versions of the manuscript. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data of PSPLIB can be downloaded from "[http://www.omdb.wi-tum.de/psplib/getdata\\_sm.html](http://www.omdb.wi-tum.de/psplib/getdata_sm.html)" (accessed on 17 June 2015).

**Acknowledgments:** We are grateful to anonymous reviewers for their thorough reviews and valuable comments.

**Conflicts of Interest:** The authors declare no conflict of interest.

**Appendix A**

**Table A1.** The skill matrix mastered by resources.

Resource \ Skill	$s_1$	$s_2$	$s_3$	Resource \ Skill	$s_1$	$s_2$	$s_3$	Resource \ Skill	$s_1$	$s_2$	$s_3$
$r_{11}$	1	0	1	$r_{21}$	1	1	0	$r_{31}$	1	1	1
$r_{12}$	1	1	0	$r_{22}$	0	1	0	$r_{32}$	1	0	1
$r_{13}$	1	1	1	$r_{23}$	0	1	1	$r_{33}$	1	0	1
$r_{14}$	1	0	0	$r_{24}$	1	1	0	$r_{34}$	0	0	1
$r_{15}$	1	0	0	$r_{25}$	1	1	1	$r_{35}$	1	0	1
$r_{16}$	1	0	1	$r_{26}$	1	1	0	$r_{36}$	0	0	1
$r_{17}$	1	1	0	$r_{27}$	0	1	0	$r_{37}$	0	0	1
$r_{18}$	1	0	1	$r_{28}$	1	1	0	$r_{38}$	1	1	1
$r_{19}$	1	1	1	$r_{29}$	0	1	0	$r_{39}$	0	0	1
$r_{110}$	1	0	0	$r_{210}$	0	1	1	$r_{310}$	1	1	1

**Table A2.** The resource state matrix.

Resource	$t$																						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
$r_{11}$	1	1	1	1	1	0	0	1	1	1	1	1	1	0	1	0	0	1	1	0	1	1	1
$r_{12}$	1	0	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1
$r_{13}$	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
$r_{14}$	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	0	1	1	1	0	1	1	0
$r_{15}$	1	0	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1
$r_{16}$	1	1	0	0	0	1	0	1	0	0	1	1	1	0	1	1	1	0	1	0	1	1	1
$r_{17}$	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
$r_{18}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$r_{19}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$r_{110}$	1	1	1	1	1	1	0	1	1	1	0	1	0	1	1	1	1	1	0	0	0	1	1
$r_{21}$	1	0	0	1	1	1	1	0	1	0	1	1	0	1	0	0	0	1	1	1	1	1	1
$r_{22}$	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
$r_{23}$	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
$r_{24}$	1	1	0	0	1	0	0	1	0	1	1	1	0	1	0	0	1	0	0	1	1	1	1
$r_{25}$	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
$r_{26}$	1	0	0	1	1	1	1	1	1	1	0	1	0	0	1	1	1	1	1	0	1	1	0
$r_{27}$	1	0	1	0	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0	0	1
$r_{28}$	1	1	1	1	1	1	1	1	0	1	0	0	1	0	1	0	0	0	1	0	0	1	1

Table A2. Cont.

Resource	t																						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
r <sub>29</sub>	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1
r <sub>210</sub>	1	1	1	1	1	1	0	1	1	0	0	1	0	1	1	0	1	1	1	1	1	1	1
r <sub>31</sub>	1	0	1	1	0	1	1	1	1	0	1	1	1	1	1	0	0	1	0	1	0	0	0
r <sub>32</sub>	1	1	0	1	1	1	1	0	1	0	1	0	1	1	0	1	0	0	1	1	1	0	1
r <sub>33</sub>	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
r <sub>34</sub>	1	0	1	1	1	0	0	1	0	0	1	0	1	1	0	1	1	0	0	1	0	1	1
r <sub>35</sub>	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	1	1	1	0	0	0	0
r <sub>36</sub>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
r <sub>37</sub>	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
r <sub>38</sub>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0
r <sub>39</sub>	1	0	1	1	1	0	1	1	1	0	1	1	0	1	1	1	1	0	1	0	0	1	1
r <sub>310</sub>	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1

Appendix B

Algorithm for Dynamic Scheduling	Algorithm for Random Scheduling	Algorithm for Static Scheduling
<p><b>Begin</b>                      Input <math>d_j, r_{js_k}, R, S, p_{ki}, Z_{t=1}</math>  <b>While</b> it &lt; MaxIt  <b>for</b> <math>j = 1 : n</math>                          Calculate <math>[ES_j, LS_j]</math>  <b>end</b>                      Initialize <math>ST_j</math>                      Resource ranking (Priority rule 1)  <b>If</b> <math>ST_j + d_j \leq ST_g, g \in SUCC_j, \forall j</math>                          <b>Then</b>                              <b>for</b> <math>t = 1 : T</math>                                  Calculate <math>L_{st}</math> (Priority rule 2)                                  Update <math>O_t, W_t</math>                                  <b>Case 1</b> <math>L_{st} \leq 0, \forall K</math>                                      Resource scheduling                                  <b>Case 2</b> <math>L_{st} &gt; 0, \forall K</math> (Priority rule 3)                                      Resource scheduling                                  <b>Case 3</b> others (Priority rule 3)                                      Resource scheduling                                  Update <math>Z_{t+1}</math>                              <b>end</b>                              Calculate <math>R_s</math> and Output Result                          <b>Else</b>                              Output Result                              Update <math>ST_j</math> (Select, Crossover, Mutate)  <b>End</b></p>	<p><b>Begin</b>                      Input <math>d_j, r_{js_k}, R, S, p_{ki}, Z_{t=1}</math>  <b>While</b> it &lt; MaxIt  <b>for</b> <math>j = 1 : n</math>                          Calculate <math>[ES_j, LS_j]</math>  <b>end</b>                      Initialize <math>ST_j</math>  <b>If</b> <math>ST_j + d_j \leq ST_g, g \in SUCC_j, \forall j</math>                          <b>Then</b>                              <b>for</b> <math>t = 1 : T</math>                                  Update <math>O_t</math>                                  Resource scheduling (No order)                                  Update <math>Z_{t+1}</math>                                  <b>end</b>                                  Calculate <math>R_s</math> and Output Result                          <b>Else</b>                              Output Result                              Update <math>ST_j</math> (Select, Crossover, Mutate)  <b>End</b></p>	<p><b>Begin</b>                      Input <math>d_j, r_{js_k}, R, S, p_{ki}, Z_{t=1}</math>  <b>While</b> it &lt; MaxIt  <b>for</b> <math>j = 1 : n</math>                          Calculate <math>[ES_j, LS_j]</math>  <b>end</b>                      Initialize <math>ST_j, SK_i</math>  <b>If</b> <math>ST_j + d_j \leq ST_g, g \in SUCC_j, \forall j</math>                          <b>Then</b>                              Calculate <math>R_s</math>                          <b>Else</b>                              Output Result                              Update <math>ST_j, SK_i</math> (Select, Crossover, Mutate)  <b>End</b></p>

References

1. Afshar-Nadjafi, B. Multi-skilling in scheduling problems: A review on models, methods and applications. *Comput. Ind. Eng.* **2021**, *151*, 107004.
2. Bellenguez, O.; Néron, E. A Branch-and-Bound method for solving Multi-Skill Project Scheduling Problem. *RAIRO-Oper. Res.* **2007**, *41*, 155–170. [[CrossRef](#)]

3. Benavides, A.J.; Ritt, M.; Miralles, C. Flow shop scheduling with heterogeneous workers. *Eur. J. Oper. Res.* **2014**, *237*, 713–720. [[CrossRef](#)]
4. Heimerl, C.; Kolisch, R. Scheduling and staffing multiple projects with a multi-skilled workforce. *OR Spectr.* **2010**, *32*, 343–368. [[CrossRef](#)]
5. Isah, M.A.; Kim, B.S. Integrating schedule risk analysis with multi-skilled resource scheduling to improve resource-constrained project scheduling problems. *Appl. Sci.* **2021**, *11*, 650. [[CrossRef](#)]
6. Chen, R.; Liang, C.Y.; Gu, D.X.; Joseph, Y.-T.L. A multi-objective model for multi-project scheduling and multi-skilled staff assignment for IT product development considering competency evolution. *Int. J. Prod. Res.* **2017**, *55*, 6207–6234. [[CrossRef](#)]
7. Polo-Mejía, O.; Artigues, C.; Lopez, P.; Basini, V. Mixed-integer/linear and constraint programming approaches for activity scheduling in a nuclear research facility. *Int. J. Prod. Res.* **2020**, *58*, 7149–7166. [[CrossRef](#)]
8. Filip, D. Applying to the mathematical methods to optimize the launching process in manufacturing. *Acta Tech. Napoc. Ser.-Appl. Math. Mech. Eng.* **2018**, *61*, 585–592.
9. Filip, D. Modern methods and tools to improve the production processes from small series and unique production. *Acta Tech. Napoc. Ser.-Appl. Math. Mech. Eng.* **2018**, *61*, 575–584.
10. Li, H.; Womer, K. Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm. *J. Sched.* **2009**, *12*, 281–298. [[CrossRef](#)]
11. Correia, I.; Saldanha-da-Gama, F. The impact of fixed and variable costs in a multi-skill project scheduling problem: An empirical study. *Comput. Ind. Eng.* **2014**, *72*, 230–238. [[CrossRef](#)]
12. Firat, M.; Hurkens, C. An improved MIP-based approach for a multi-skill workforce scheduling problem. *J. Sched.* **2012**, *15*, 363–380. [[CrossRef](#)]
13. Snauwaert, J.; Vanhoucke, M. A new algorithm for resource-constrained project scheduling with breadth and depth of skills. *Eur. J. Oper. Res.* **2021**, *292*, 43–59. [[CrossRef](#)]
14. Snauwaert, J.; Vanhoucke, M. Mathematical formulations for project scheduling problems with categorical and hierarchical skills. *Comput. Ind. Eng.* **2022**, *169*, 108147. [[CrossRef](#)]
15. Buddhakulsomsiri, J.; Kim, D.S. Priority rule-based heuristic for multi-mode resource-constrained project scheduling problems with resource vacations and activity splitting. *Eur. J. Oper. Res.* **2007**, *178*, 374–390. [[CrossRef](#)]
16. Goldratt, E. *Critical Chain*; North River Press: Great Barrington, MA, USA, 1997.
17. Herroelen, W.; Demeulemeester, E.; De Reyck, B. A classification scheme for project scheduling. In *International Series in Operations Research and Management Science; Project Scheduling: Recent Models, Algorithms and Applications*; Weglarz, J., Ed.; Kluwer Academic: Boston, MA, USA, 1998.
18. Lambrechts, O.; Demeulemeester, E.; Herroelen, W. Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities. *J. Sched.* **2008**, *11*, 121–136. [[CrossRef](#)]
19. Lambrechts, O.; Demeulemeester, E.; Herroelen, W. Time slack-based techniques for robust project scheduling subject to resource uncertainty. *Ann. Oper. Res.* **2011**, *186*, 443–464.
20. Ahmadpour, S.; Ghezavati, V. Modeling and solving multi-skilled resource-constrained project scheduling problem with calendars in fuzzy condition. *J. Ind. Eng. Int.* **2019**, *15*, 179–197. [[CrossRef](#)]
21. Azizi, N.; Liang, M. An integrated approach to worker assignment, workforce flexibility acquisition, and task rotation. *J. Oper. Res. Soc.* **2013**, *64*, 260–275. [[CrossRef](#)]
22. Wongwai, N.; Malaikrisanachalee, S. Augmented heuristic algorithm for multi-skilled resource scheduling. *Autom. Construct.* **2011**, *20*, 429–445.
23. Kher, H.V.; Malhotra, M.K.; Philipoom, P.R.; Fry, T.D. Modelling simultaneous worker learning and forgetting in dual resource constrained systems. *Eur. J. Oper. Res.* **1999**, *115*, 158–172. [[CrossRef](#)]
24. Yue, H.; Slomp, J.; Molleman, E.; Vanderzee, D.J. Worker flexibility in a parallel dual resource constrained job shop. *Int. J. Prod. Res.* **2008**, *46*, 451–467. [[CrossRef](#)]
25. Gans, N.; Zhou, Y. Managing Learning and Turnover in Employee Staffing. *Oper. Res.* **2002**, *50*, 991–1006. [[CrossRef](#)]
26. De Bruecker, P.; Ven den Bergh, J.; Beliën, J.; Demeulemeester, E. Workforce planning incorporating skills: State of the art. *Eur. J. Oper. Res.* **2015**, *243*, 1–16. [[CrossRef](#)]
27. Bard, J.F.; Purnomo, H.W. Preference scheduling for nurses using column generation. *Eur. J. Oper. Res.* **2005**, *164*, 510–534. [[CrossRef](#)]
28. Lagodimos, A.G.; Leopoulos, V. Greedy heuristic algorithms for manpower shift planning. *Int. J. Prod. Econ.* **2000**, *68*, 95–106. [[CrossRef](#)]
29. Lagodimos, A.G.; Mihiotis, A.N. Overtime vs. regular shift planning decisions in packing shops. *Int. J. Prod. Econ.* **2006**, *101*, 246–258. [[CrossRef](#)]
30. Correia, I.; Lourenço, L.L.; Saldanha-da-Gama, F. Project scheduling with flexible resources: Formulation and inequalities. *OR Spectr.* **2012**, *34*, 635–663. [[CrossRef](#)]