

Optimizing for a Resource-constrained Multi-project Scheduling Problem with Planned Resource Unavailability

Jinwen Tian*, Xingye Dong and Sheng Han

Beijing Key Lab of Traffic Data Analysis and Mining, School of Computer and IT, Beijing Jiaotong University, Beijing 100044, China

*Corresponding author

Abstract—Based on a real producing scenario, we established a resource-constrained multi-project scheduling problem with planned resource unavailability model (RCMPSP-PRU) to minimize the makespan. Different from the traditional resource-constrained multi-project scheduling model, RCMPSP-PRU introduces some new concepts such as site, movable resource and unmovable resource, and accompanied with planned resource unavailability. In order to solve RCMPSP-PRU, we firstly proposed a heuristic algorithm called ISHPR based on serial generation scheme and priority rules, then two improved algorithms named ISG-PS and ISG-PSTS was designed respectively, in which the genetic algorithm, particle swarm optimization and tabu search are incorporated. The GA and PSO algorithms were used to enhance the selection of better site for each job, and the TS was used to exploit better solutions when a resource unavailability occurred. The experimental results based on a real world instance show that ISG-PSTS has the best performance, which illustrates the effectiveness of this work. In addition, the method combined with various intelligent algorithms to solve scheduling problems can inspire later research.

Keywords—multi-project scheduling; resource-constrained; planned resource unavailability; genetic algorithm; particle swarm optimization; tabu search

I. INTRODUCTION

Resource-constrained project scheduling problem (RCPSP) is to allocate the required resources and arrange the start and finish time for each task in the project in the case of satisfying a series of constraints, including the resource constraints, so as to achieve specific goals, such as the minimum makespan, the minimum cost and the maximum income. In the actual production process of enterprises, resource-constrained multi-project scheduling problems (RCMPSP) are widespread, and the most of them are with multiple resources constraints [1]. In addition, for actual scheduling problems, resources are not always be available because of many factors such as equipment failure and maintenance. (This is called planned resource unavailability in this paper.) Therefore, it is necessary and important to consider the planned resource unavailability in RCMPSP. However, there are lack of researches on multi-project scheduling problems considering planned resource unavailability in the existing literatures. O. Lambrechts et al. [2] analyze the impact of unexpected resource breakdowns on activity durations, and they propose an approach for inserting explicit idle time into the project schedule in order to protect it

as well as possible from disruptions caused by resource unavailability. H. Mogaadi and B.F. Chaar [3] focus on the resource-constrained project scheduling problem with uncertain activity duration, and they propose an adaptive robust genetic approach with a sophisticated initial population and a Forward-Backward Improvement heuristic. J. Poppenborg and S.A. Knust [4] propose a tabu search algorithm for the resource-constrained project scheduling problem with transfer times. However, these researches fail to solve the resource-constrained multi-project scheduling problem with planned resource unavailability. Therefore, we propose a resource-constrained multi-project scheduling problem with planned resource unavailability model (RCMPSP-PRU) which is to solve the possible planned resource unavailability and aimed at minimizing the makespan of multi-project.

The rest of this paper is organized as follows. In Section 2, the RCMPSP-PRU model is described in detail. In Section 3, we introduces three algorithms for solving the RCMPSP-PRU model: First, the improved serial heuristic algorithm based on the priority rules (ISHPR) is proposed; secondly, the improved serial algorithm based on the genetic-particle swarm (ISG-PS) is proposed to optimize ISHPR; Finally, an improved serial algorithm combining tabu search and genetic-particle swarm (ISG-PSTS) is proposed to further optimize ISG-PS. In Section 4, the experimental results and analysis are given. Finally, the paper is concluded in Section 5.

II. THE MODEL OF RCMPSP-PRU

In this section, we first describe the actual scheduling problems researched in this paper; then, we describe the mathematical model of the RCMPSP-PRU.

A. Problem Description

In RCMPSP-PRU, n jobs consist a set of jobs which is denoted as $N = \{1, 2, \dots, n\}$. The jobs are independent of each other, but share the same resource pool. Each of the job needs to complete M activities, and the set of activities for each job i ($i \in N$) is $A_i = \{0, 1, \dots, m\}$. In A_i the first and last activities are virtual activities that do not need to consume resources and time, and they only represent the beginning or the end of the activities. The activity j ($j \in A_i$) of job i is (i, j) . The finish time ft_{ij} of each activity is equal to the sum of its start time st_{ij} and its duration p_{ij} . All of the activities are carried out during the time $[0, T^*]$. The set of activities executed at time t is denoted as K_t .

Activities and precedence constraints of multi-project are represented by a finish-to-start activity-on-node (AON) network. Figure 1 is an example of an AON network (not the AON of RCMPSP-PRU). The AON nets of all jobs are the same. Activity cannot be started before all its predecessors have been finished, and once the activity is started, it must not be interrupted. In addition to precedence constraints, there are dependent constraints and mutually exclusive constraints between activities. The start time of activity (i, j) must be later than the finish time of its dependent activity and must use the resources that its dependent activity uses; activity (i, p) and activity (i, j) are mutually exclusive, indicating that these two activities cannot be executed simultaneously. We denote $Pre_{ij}(i \in N, j \in A_i)$ as set of all predecessor activities of (i, j) , $Dep_{ij}(i \in N, j \in A_i)$ as set of all activities that have dependent relationships with (i, j) , $Exc_{ij}(i \in N, j \in A_i)$ as set of all activities that have mutually exclusive relationships with (i, j) .

The job can execute activities only when it is placed on a specific type of site. There are a total of w different types of sites, and the set of sites type is $S = \{S_1, \dots, S_w\}$. Each type of site contains a set of sites $S_w = \{1, 2, \dots, n\}$. Different types of sites support different types and quantities activities. When the job can't continue to execute the activities at a site, it needs to be transferred to one of the rest of the site with speed sv_i to execute the remaining activities. The distance between site p and site q is denoted as d_{pq} . The time when the job i leaves the station p denoted as LT_{ip} and the time to reach station q denoted as AT_{iq} should satisfy the restrictions of sv_i and d_{pq} .

Each of the activities needs a variety of specific types of movable and unmovable resources to carry out. There are many different types of movable resources. Each type of movable resource contains multiple specific resources. The unmovable resources are similar to it. The set of movable resources (unmovable resources) is $M = \{1, \dots, r\}$ ($UM = \{1, \dots, r\}$). Each type of movable resource (unmovable resource) contains a set of resources $M_i = \{1, 2, \dots, n\}, i \in M$ ($UM_i = \{1, 2, \dots, n\}, i \in UM$). The movable resources can move to any site to provide services. The unmovable resources are distributed around the sites. The amount of r type movable resources (unmovable resources) required for (i, j) is denoted as R_{ijr}^M ($R_{ijr}^M < |M_r|$) (R_{ijr}^{UM} ($R_{ijr}^{UM} < |UM_r|$)). Moreover, CR_{ij}^{UM} (CR_{ij}^M) represents the set of unmovable (movable) resource used constantly by activity (i, j) during an operation phrase. Unmovable resources are not always available due to planned resource unavailability. The set of break periods for unmovable resources r ($r \in UM$) is denoted as $PRU_r = \{(fs_1, fe_1), \dots, (fs_l, fe_l)\}$. Inversely, the set of working periods for unmovable resources r ($r \in UM$) is denoted as $UPRU_r = \{[s_1, e_1], \dots, [s_l, e_l]\}$. The duration of an activity must be included in one same working period of $UPRU_r$.

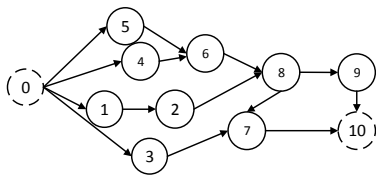


FIGURE 1. AN EXAMPLE OF AON NETWORK

B. Mathematical Model

Based on the above problem description, the mathematical model of RCMPSP-PRU is established in this section. The aim of the RCMPSP-PRU is to find a feasible multi-project scheduling plan $S = (S_1, S_2, \dots, S_N)$ (where $S_i = ((st_{i0}, ft_{i0}), (st_{i1}, ft_{i1}) \dots (st_{im}, ft_{im}))$) and minimize the makespan. In each schedule S_i , the constraint relations between the activities, sites, and resources mentioned above are satisfied. The mathematical model is as follows:

$$F(S) = \min\{ \max ft_{im}, i \in N, m \in A_i \} \quad (1)$$

s.t.

$$ft_{ij} = st_{ij} + p_{ij}, \forall i \in N, \forall j \in A_i \quad (2)$$

$$st_{ij} \geq ft_{ib}, \forall i \in N, \forall j \in A_i, \forall (i, b) \in Pre_{ij} \quad (3)$$

$$st_{ij} \geq ft_{id}, R_{ijr}^{UM} = R_{idr}^{UM} \vee R_{ijr}^M = R_{idr}^M, \forall i \in N, \forall j \in A_i, \forall (i, d) \in Dep_{ij}, r \in UM \vee r \in M \quad (4)$$

$$st_{ie} \geq ft_{ij} \vee ft_{ie} \leq st_{ij}, \forall i \in N, \forall j \in A_i, \forall (i, e) \in Exc_{ij} \quad (5)$$

$$st_{ij}, ft_{ij} \in [0, T'], \forall i \in N, \forall j \in A_i \quad (6)$$

$$e_l - s_l \geq p_{ij} \wedge st_{ij} \geq s_l, \forall i \in N, \forall j \in A_i, \exists (s_l, e_l) \in UPRU_r, \forall r \in CR_{ij}^{UM} \quad (7)$$

$$\sum_{(i,j) \in K_t} R_{ijr}^{UM} \leq |UM_r|, i \in N, j \in A_i, \forall r \in UM, t \in [0, T'] \quad (8)$$

$$\sum_{(i,j) \in K_t} R_{ijr}^M \leq |M_r|, i \in N, j \in A_i, \forall r \in M, t \in [0, T'] \quad (9)$$

$$AT_{ip} \geq LT_{iq} + d_{pq}/sv_i, \forall i \in N, \{p, q\} \in S_w, w \in S \quad (10)$$

The objective (1) is to minimize the makespan of multi-project. Constraint (2) represents the relationship among the start time, finish time, and duration of activities. Constraint (3) represents the precedence constraints among the activities. Constraint (4) and (5) represent mutual exclusion constraints and dependent constraints among activities, respectively. Constraint (6) represents the time range of the whole project. Constraint (7) represents the planned resource unavailability constraints of unmovable resources. Constraint (8) and (9) represent the amount constraints of unmovable resources and movable resources, respectively. Constraint (10) represents the constraint of jobs when they transfer from one site to another.

III. SOLUTION PROCEDURES

Different from the traditional resource-constrained multi-project scheduling model, RCMPSP-PRU involves many new objects such as site, movable resource and unmovable resource, and it is with planned resource unavailability. RCMPSP-PRU is more complex than the traditional RCPSp, an efficient and flexible solution is really needed. Inspired by the existing researches on RCMPSP [11, 12], we propose three algorithms to solve the model RCMPSP-PRU. First, the improved serial heuristic algorithm based on the priority rules (ISHPR) is

proposed. Then the selection process of sites in ISHPR is optimized, and the improved serial algorithm based on the genetic-particle swarm (ISG-PS) is proposed. Finally, based on ISG-PS and combining the tabu search algorithm, the processing method of planned resource unavailability is optimized. In this way, an improved serial algorithm combining tabu search and genetic-particle swarm (ISG-PSTS) is proposed. This paper will use pseudo-code to describe the three algorithms, and Table 1 shows some symbols used in algorithm description.

TABLE I. THE MAIN SYMBOLS IN THE ALGORITHM

Symbol	Meaning
TJ	$TJ = \{i i \in N\}$: the set of jobs who participate in the current project in a scheduling
TJO	$TJO = \{i i \in TJ\}$: the set of transfer order of jobs
D	The scheduled set, consists of all the activities whose start time, finish time and needed resources have been assigned
UD	The unscheduled set, consists of all the activities that have not been scheduled and all of their predecessor activities belong to D
(i^*, j^*)	Basic activity determined by basic activities rules(will be shown below)
$\{(i, j)^*\}$	Parallel activities set, determined by parallel activities rules(will be shown below)

A. Improved Serial Heuristic Algorithm Based on Priority Rules (ISHPR)

The schedule generation scheme (SGS) proposed by Kelley [5] is the core of most heuristic algorithms for solving project scheduling problems. Parallel multi-project schedule generation scheme (PMPSGS) and serial multi-project schedule generation scheme (SMPSGS) are two types of SGS [6], and SMPSGS is widely used because of its larger search space [7]. The process of SMPSGS is shown in Table 2. In RCMPSP-PRU, each job has its own set of activities. So in the case of job and resource constraints, many activities can be processed at the same time to improve the efficiency of SMPSGS. Thus, according to characteristics of RCMPSP-PRU, we combine the idea of improved SMPSGS and design the priority rules to propose the algorithm ISHPR. The priority rules of this paper are as follows:

Site Rules: (1) The maximum number of activities supported by the site (2) The earliest available time for the site (3) Maximum amount of unmovable resources can be obtained.

Resource Rules: (1) The minimum failure rate RA_r ($RA_r = \sum_i (f e_i - f s_i) / |P_e - P_s|$, P_e and P_s respectively represents the estimated end time and start time that resources r is used under current scheduling state) (2) The earliest available time (3) Unmovable resource priority, that is, an activity can be provided service either by movable resource or unmovable resource, then we select the unmovable resources.

Transfer Order Rules: (1) The job that remains more activities undone is firstly transferred (2) The time available for the target site is earlier.

Basic Activity Rules: (1) The earliest start time (2) The earliest end time.

Parallel Activities Rules: (1) The candidate activities and the basic activity belong to the same job (2) The finish time of the candidate activities is earlier than the finish time of the basic activity. (Need to satisfy the above two at the same time.)

The process of ISHPR is shown in Table 3.

TABLE II. THE MAIN PROCESS OF SMPSGS

1.	For each job i , initialize $st_{i1} = ft_{i1} = 0$; $D = \{(i, 1) i \in N\}$;
2.	Calculate the UD based on the current scheduling environment;
3.	while (UD is not empty)
4.	Select (i^*, j^*) form UD by priority rules;
5.	Assign necessary resources, start time and finish time for (i^*, j^*) ;
6.	Update resource usage;
7.	Update D , $D = D \cup \{(i^*, j^*)\}$; $UD = UD - \{(i^*, j^*)\}$;
8.	end while
9.	Output result, end.

TABLE III. THE MAIN PROCESS OF ISHPR

1.	Initialize solution $S^* = \text{null}$;
2.	Compute the UD based on the current scheduling environment;
3.	while (UD is not empty)
4.	Compute the UD ;
5.	if (UD is empty)
6.	Break ;
7.	end if
8.	if (TJ is not empty)
9.	Compute TJO by Transfer Order Rules;
10.	for $i = 1$ to $ TJO $
11.	Select the best site for each job by Site Rules and transfer it to the site selected;
12.	end for
13.	end if
14.	Pre-schedule activities in UD by Resource Rules;
15.	Select (i^*, j^*) form UD by Basic Activity Rules;
16.	Select $\{(i, j)^*\}$ form UD by Parallel Activities Rules;
17.	Schedule all operations in $\{(i, j)^*\}$ and update the usage information of resources, station, and job;
18.	update $D = D \cup \{(i, j)^*\}$; $UD = UD - \{(i^*, j^*)\}$;
19.	end while
20.	Output S^* , end.

B. Improved Serial Algorithm Based on the Genetic-Particle Swarm (ISG-PS)

In RCMPSP-PRU, there are multiple jobs in different states that need to be scheduled at the same time. Meanwhile, the resources required by each job are also in different states and have planned resource unavailability. It is found that when multiple jobs need to be transferred at the same time, the transfer order of jobs will have a great impact on the scheduling results. (The job that is first transferred has the priority of the selection of site.) In ISHPR, we compute the transfer order of jobs by the Priority Rules designed by experience, but this method does not always lead to a better transfer order. Moreover, when the number of job is large, there are many kinds of possibility of transfer order. Therefore, the meta-heuristic algorithm is considered to optimize the transfer order of jobs. Genetic algorithm and particle swarm optimization are two extensive used meta-heuristic algorithms:

1) Genetic algorithm is widely used because of its strong global search capability, but its local search capability is poor. A single genetic algorithm does not work well, which always is time-consuming and inefficient in later evolutionary stage [8].

2) Particle swarm optimization has strong local search capability, and is simple to implement and fast convergent. However, its fast convergence speed also leads to a rapid decline in population diversity, resulting in poor global search ability and premature convergence [8].

In order to quickly find the optimal transfer order of job through a finite number of iterations, we are inspired by the idea of ensemble learning in machine learning [9] and combine the advantages of genetic algorithms and particle swarm optimization to propose the algorithm ISG-PS. The main idea of the ISG-PS is: first we use genetic algorithm to find a better set of transfer order globally; then, the better set of transfer order derived from the genetic algorithm is used as the initial of the particle swarm optimization. The Particle swarm algorithm to further search for global optimal solutions using its strong local search capability. Next, we introduce the algorithm ISG-PS in detail.

$$V'_i = V_i + T(P_i - X_i) + U(P_g - X_i) \quad (11)$$

$$X'_i = X_i + V_i \quad (12)$$

$$fit(p) = 1 / FS(ISHPR, p) \quad (13)$$

TABLE IV. THE MAIN PROCESS OF ISG-PS

1.	Initialize the optimal solution $S^* = null$, the optimal transfer order $P_g = null$;
2.	Initialize parameters: set the number of iterations max_iter , iteration number of genetic algorithm iterG , population size M , population crossover probability α , population mutation probability β ;
3.	Initialize the population P : randomly generate M sequences of job transfer order to form the initial population P ;
4.	For $p \in P$ DO $fit(p)$
5.	For $iter = 1$ to $iterG$ (We use (13) when it comes to calculating fitness values of p in step 7 to step 10.)
6.	{
7.	Selection: selection operation using the method of roulette and elite for P ;
8.	Crossover: crossover operation with partial matching crossover and probability α for P ;
9.	Variation: mutation operation with probability β for P ;
10.	}
11.	}
12.	end for
13.	$P' = P$, P' as the initial particle swarm;
14.	Set the internal exchange order of each particle in P' randomly as its initial velocity;
15.	for $iter = iterG+1$ to max_iter
16.	for $p \in P'$ DO
17.	IF $fit(p) > fit(P_p)$ then $P_p = p$
18.	end for
19.	for $p \in P'$ DO
20.	if $fit(P_p) > fit(P_g)$ then $P_g = P_p$
21.	end for
22.	Adjust velocity and state of particles according to (10) and (11);
23.	end for
24.	Use the solution architecture of ISHPR to obtain S^* . (P_g is used as the transfer order of job rather than compute by the Transfer Order Rules.)
25.	Output S^* , end.

In ISG-PS, the ID of each job is taken as a gene of the chromosome, and a permutation sequence of all the job numbers constitutes a chromosome (also can be used as a particle). Equations (11) and (12) are the velocity update formula and the state update formula of particle [10], where: T and U ($T, U \in [0,1]$) are random numbers; X_i and V_i represent the state and speed of particle i respectively; P_i and P_g represent the best position experienced by each particle and the group respectively. Equation (13) is the fitness function, where p is a

particle or an individual, and p also represents a transfer order of jobs generated in the process of ISG-PS rather than computed by the Transfer Order Rules mentioned above. $FS(ISHPR, p)$ represents the objective function value computed by (1) in ISHPR (The transfer order is replaced by p in ISHPR). The process of ISG-PS is shown in Table 4 (the steps 5~11 are the process of genetic algorithm, and the steps 13~23 are the process of particle swarm optimization).

C. Improved Serial Algorithm Combining Tabu Search and Genetic-Particle Swarm (ISG-PSTS)

In RCMPSP-PRU, unmovable resources provide service for sites. Because of the planned resource unavailability of some unmovable resources, the service capability of the sites are weakened. Therefore, when the job needs to select a site to transfer, different selection strategies make different scheduling results. In ISG-PS, the genetic-particle swarm optimization was used to determine a better transfer order of job. Then the Site Rules are used to select the optimal site for each job. However, this method of selecting sites does not take into account the weakness of the support capability of sites due to the planned resource unavailability. If the site selected by the Site Rules is badly constrained by the resource unavailability, then the scheduling result obtained by ISG-PS is bad. Therefore, we have made improvements to ISG-PS for its method of solving the planned resource unavailability.

Tabu search is a widely used intelligent algorithm, which has the feature of avoiding roundabout searching [11] by introducing a flexible storage structure and corresponding tabu criteria. Combining with the characteristics of planned resource unavailability in RCMPSP-PRU and the characteristics of tabu search, we design a tabu object ($\{(i, j), r', count, isflag\}$) to assist each job to quickly find the optimal site. Based on ISG-PS, by introducing the tabu search to select the optimal site, we propose a two-stage algorithm ISG-PSTS: the first stage is to select the optimal transfer order of job by ISG-PS; the second stage is to improve the site selection process of ISHPR by tabu search (The improved algorithm is named as ISHPR-TS).

$$fit_{ts}(p) = 1 / TSFT(ISHPR-TS, p) \quad (14)$$

Equation (14) is the fitness function of ISG-PSTS, where p is a transfer order of jobs, $TSFT(ISHPR-TS, p)$ represents the function value computed by objective (1), $TSFT$ represents the scheduling process of ISHPR-TS. Table 6 is the process of the algorithm ISG-PSTS.

(1) The main process of ISHPR-TS is shown in Table 5 (the second stage of ISG-PSTS). $TS_q (q \in S_w, S_w \in S)$ is the tabu list of site q and each site has a tabu list to store its tabu objects; $\{(i, j), r, count, isflag\}$ is a tabu object, and its meaning is: When $\{(i, j), r, count, isflag\}$ exists in TS_q and the value of $isflag$ is true, the unmovable resource r has planned unavailability, which lead that there is no ability to execute (i, j) on S_q . When the job selects site to execute activities, it is necessary to judge the capability determined by unmovable resources of all sites. The unmovable resource that in the tabu list is not selected, and then the tabu list guide the job gradually finds the optimal site that is the least affected by the planned

unavailability of unmovable resources. The main ideas of ISHPR-TS as follows:

1) First, a parallel set of activities $\{(i, j)^*\}$ (step 6 in Table 5) is scheduled without considering the planned resource unavailability.

2) Secondly, reschedule each activities of $\{(i, j)^*\}$ in consideration of the planned resource unavailability (step 10 in Table 5).

3) For (i, j) ($(i, j) \in \{(i, j)^*\}$), if the start time obtained by 2) is later than the start time in the case of step 1), it indicates that the unmovable resources used by (i, j) weakened the capacity of the current site due to the planned unavailability. This moment, the $\{(i, j), r, count, isflag\}$ in the tabu list of current site are updated, and then it is used to guide the jobs to select the sites that are least affected by the planned resource unavailability in the cyclic search (step 11~19 in Table 5).

TABLE V. THE MAIN PROCESS OF ISHPR-TS

1.	Initialize the job transfer order p , current optimal solution S^* , search termination marker $flag = true$;
2.	while ($flag$)
3.	Compute the UD based on the current scheduling environment;
4.	Update $flag = false$;
5.	while (UD is not empty)
6.	Execute the 4~16 steps in ISHPR(p is used as the transfer order of job rather than compute by the Transfer Order Rules) to compute $\{(i, j)^*\}$;
7.	Set $\{(i, j)^{\#}\} = \{(i, j)^*\}$;
8.	while ($\{(i, j)^{\#}\}$)
9.	Get the start time st_{ij} , the site S_q , the required unmovable resource r' of (i, j) ;
10.	Recalculate st'_{ij} of (i, j) under the premise of considering the planned unavailability of unmovable resource
11.	if ($st'_{ij} > st_{ij}$);
12.	if ($\{(i, j), r, count, isflag\} \in TS_t, r \in CR_{ij}^{UM}$)
13.	$\{(i, j), r, count, isflag\} = \{(i, j), r, count + +, isflag\}$;
14.	else
15.	Set: $isflag = true, count = count + 1$ in $\{(i, j), r, count, isflag\}$ And update $TS_q = TS_q \cup \{(i, j), r, count, isflag\}$;
16.	end if
17.	Update $st_{ij} = st'_{ij}$, and update the finish time of (i, j) ;
18.	$flag = isflag$;
19.	end while
20.	Update $\{(i, j)^{\#}\} = \{(i, j)^{\#}\} - (i, j)$;
21.	end while
22.	Schedule all activities in $\{(i, j)^{\#}\}$ and update resources, station, and job usage;
23.	Update $D = D \cup \{(i, j)^{\#}\}$;
24.	end while
25.	Set S' as the best solution of all candidate solutions in step 2-24;
26.	if (S' is better than S^*)
27.	Update $S^* = S'$;
28.	end if
29.	Output S^* , end.

(2) The overall process of ISG-PSTS is shown in Table 6, which combines tabu search and genetic-particle swarm optimization to optimize the solution process.

TABLE VI. THE SECOND STAGE OF ISG-PSTS

1.	Generate an initial solution S by ISG-PS, set the current optimal solution $S^* = S$;
2.	Use the solution architecture of ISG-PS to obtain S^* . (the fitness function $fit(p)$ is replaced by $fit_{ts}(p)$)
3.	Output S^* , end.

IV. EXPERIMENT

In order to verify the performance of the proposed algorithms, we use two representative cases provided by a company to conduct experiments. There are 30 activities, including 14 pairs of mutually exclusive activities and 5 pairs of dependent activities. The detailed description of the cases are shown in Table 7. In each case, the planned resource unavailability of the unmovable resources is randomly generated, and we select nine instances to carry out the experiment.

The algorithms are programmed in C++ and run on a LENOVO PC (Intel(R) Core(TM) i5 – 4300U CPU @ 2.50GHz, memory: 8GB). In this paper, the results of *SMPSGS* are compared with the results of the ISHPR, ISG-PS, and ISG-PST respectively, and the performance of the algorithm is measured by *RPD*. *RPD* represents the average relative percentage, which is calculated as: $RPD = (\text{initial solution} - \text{optimal solution}) / \text{initial solution}$ (initial solution refers to solution obtained by *SMPSGS*, and optimal solution refers to the solution obtained by ISHPR, ISG-PS or ISG-PSTS). The experimental results are shown in Table 8 and Table 9.

TABLE VII. DETAILS OF THE CASES

The environment element	Case 1	Case 2
Site type	12	12
Site	60	60
Unmovable resource type	15	15
Unmovable resource	135	135
Move resource type	12	12
Move resource	40	40
Job	9	12
Activities (in per job)	30	30
Pairs of mutually exclusive activities	14	14
Pairs of dependent activities	5	5

TABLE VIII. EXPERIMENTAL RESULTS I

Instance	Case 1						
	<i>SMPSGS</i> (s)	ISHPR (s)	RPD (%)	ISG-PS (s)	RPD (%)	ISG-PSTS (s)	RPD (%)
1	5433	5163	4.97	4157	23.49	3519	35.23
2	6740	6421	4.73	5093	24.44	4483	33.49
3	6364	6117	3.88	4974	21.84	4281	32.73
4	5668	5446	3.92	4613	18.61	3819	32.62
5	6778	6492	4.22	5401	20.32	4647	31.44
6	5947	5669	4.67	4785	19.54	4079	31.41
7	6816	6548	3.39	5388	20.95	4757	30.21
8	6412	6184	3.56	5221	18.57	4392	31.50
9	6333	6027	4.83	5112	19.28	4295	32.18

TABLE IX. EXPERIMENTAL RESULTS 2

Instance	Case 2						
	SMPSGS S (s)	ISHPR (s)	RPD (%)	ISG-PS (s)	RPD (%)	ISG-PSTS (s)	RPD (%)
1	7594	7243	4.62	5906	22.23	5118	32.60
2	7768	7419	4.49	6187	20.35	5457	29.75
3	7402	7097	4.12	5946	19.67	5106	31.02
4	8164	7912	3.09	6554	19.72	5912	27.58
5	7809	7507	3.87	6245	20.03	5622	28.01
6	8104	7819	3.52	6619	18.32	5849	27.83
7	7398	7103	3.99	6109	17.42	5124	30.74
8	8002	7752	3.12	6682	16.50	5907	26.18
9	8014	7734	3.49	6421	19.88	5993	25.22

The experimental results show that compared with the results of SMPSGS, the results of ISHPR, ISG-PS, and ISG-PSTS have improved to varying degrees. The ISHPR is only slightly improved (less than 5%), which indicates that heuristic algorithm based on priority rules and SMPSGS is far from the ideal solution. The results of ISG-PS has obvious improvement (about 20%). It shows that the use of genetic-particle swarm hybrid algorithm to improve the transfer order of the job is a correct and effective optimization direction. It also indicates that the transfer order of the job has a great influence on the scheduling results. The algorithm ISG-PSTS has the best optimization effect (about 30%), which indicates that the planned resource unavailability affects the scheduling results to a great extent, and the optimal solution of RCMPSP-PRU can be obtained by optimizing from the aspects of job transfer order and planned resource unavailability.

V. CONCLUSION

In order to solve an actual resource-constrained project scheduling problem, the model RCMPSP-PRU is established aiming at minimizing the makespan. Different from the traditional RCMPSP model, RCMPSP-PRU introduces some new concepts such as site, job, activity, movable resource and unmovable resource, and it is with planned resource unavailability. In order to solve the model, three algorithms are proposed in this paper. First, based on SMPSGS and priority rules, ISHPR is proposed. Secondly, the selection process of sites in ISHPR is optimized, we propose the ISG-PS. Finally, ISG-PSTS is proposed, which is based on ISG-PS and combining the tabu search algorithm to optimize the processing method of planned resource unavailability. Experimental results show that ISG-PSTS achieves the best results, and it can correctly and efficiently solve the RCMPSP-PRU model. In addition, this paper combines various intelligent algorithms to solve the scheduling problem, which can provide new ideas for other similar scheduling problems.

REFERENCES

- [1] Payne J H. Management of multiple simultaneous projects: a state-of-the-art review[J]. *International Journal of Project Management*, 1995, 13(3):163-168.
- [2] Lambrechts O, Demeulemeester E, Herroelen W. Time slack-based techniques for robust project scheduling subject to resource uncertainty[J]. *Annals of Operations Research*, 2011, 186(1):443-464.
- [3] Mogaadi H, Chaar B F. Scenario-Based Evolutionary Approach for Robust RCMPSP[C]. *Proceedings of the Second International Afro-*

- European Conference for Industrial Advancement AECIA 2015, 2016, 44-45.
- [4] Poppenborg J, Knust S. A flow-based tabu search algorithm for the RCMPSP with transfer times[J]. *OR Spectrum*, 2016, 38(2):305-334.
- [5] Kelley J E. The critical-path method: Resources planning and scheduling[J]. *J.f.muth G.l.thompson Industrial Scheduling*, 1963, volume 37(11):108-111.
- [6] Rainer Kolisch. Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation[J]. *European Journal of Operational Research*, 1996, 90(2):320-333.
- [7] Arno Sprecher, Rainer Kolisch, Andreas Drexl. Semi-active, active, and non-delay schedules for the resource-constrained project scheduling problem[J]. *European Journal of Operational Research*, 1995, 80(1):94-102.
- [8] Min J, Hua-Xiang L U. A multi-subgroup hierarchical hybrid of genetic algorithm and particle swarm optimization[J]. *Control Theory & Applications*, 2013, 30(10):1231-1238.
- [9] Y. Liu, X. Yao. Ensemble learning via negative correlation[J]. *Neural Netw*, 1999, 12(10):1399-1404.
- [10] Shia X H, Lianga Y C, Leeb H P, et al. Particle swarm optimization-based algorithms for TSP and generalized TSP[J]. *Information Processing Letters*, 2007, 103(5):169-176.
- [11] Bomze I M, Budinich M, Pardalos P M, et al. *Handbook of Combinatorial Optimization*[J]. Kluwer Academic Publishers Boston Ma, 2013:viii,394.
- [12] Browning T R, Yassine A A. Resource-constrained multi-project scheduling: Priority rule performance revisited[J]. *International Journal of Production Economics*, 2010, 126(2):212-228.
- [13] Krüger D. A heuristic solution framework for the resource constrained (multi-)project scheduling problem with sequence-dependent transfer times[J]. *European Journal of Operational Research*, 2009, 197(2):492-508.