2020 IEEE 16th International Conference on Control & Automation (ICCA)
Oct 9-11, 2020 (Virtual) Sapporo, Hokkaido, Japan

A Robust Genetic Algorithm to Solve Multi-Skill Resource Constrained Project Scheduling Problem with Transfer Time and Uncertainty Skills

Junqi Cai, Zhihong Peng, Shuxin Ding, Jingbo Sun

***Abstract*— Multi-skill resource-constrained project scheduling problem (MS-RCPSP) is one of the most investigated problems in operations research. Most researches ignore transfer time of resources between activities, which is regularly encountered in manufacturing and service industries. Traditional methods assume that the skill value of resource is fixed, but in practice, it changes with the influence of the environment. When using traditional approach, the optimizing procedure of the baseline project plan fails and leads to delays. To address this issue, we propose a robust model which employs a novel robust counterpart that is different from the previous literature. A new genetic algorithm using two new population initialization heuristic methods is proposed to find a robust schedule. Experiment shows the effectiveness of our proposed method in providing more robust schedules under resource skill uncertainty.**

I. INTRODUCTION

As one of the most important issues in decision making area, the resource-constrained project scheduling problem (RCPSP) aims to schedule a given set of nonpreemptive activities considering precedence relations and a finite set of renewable resources. Because RCPSP is NP-hard, the time cost is unacceptable when using exact algorithm to solve large scale problem [1] [2]. Researchers pay a lot attention to heuristic algorithms, which can obtain near-optimal solution in shorter time. Two categories of heuristic algorithm, constructive and meta-heuristic methods, are widely investigated. In constructive method, priority rules are widely used, such as latest start time (LST), longest processing time (LPT), latest finish time (LFT), shortest processing time (SPT), and most total successor (MTS). Researchers extent the traditional RCPSP to adapt to the the actual industry situation. Especially when it comes to schedule the resources like manpower or multi-purpose vehicles, each resource masters two or more kinds of skill. To deal with this problem, the multi-skill resource constrained project scheduling problem (MS-RCPSP) is proposed to make it be more close to the real-world situation [3]. A lot of meta-heuristic algorithms are proposed for RCPSP, such as genetic algorithm (GA) ([5]), particle swarm optimization (PSO) ([6] [7]), tabu search (TA) [8], simulated annealing (SA) [9], shuffled frog-leaping (SFLA) [10], harmony search (HS) [11], etc.

The standard RCPSP and MS-RCPSP assume that it doesn't take time for resource to transfer between different activities. In practice, sometimes the distance between the sites of activities are relatively long, moving resources between these sites may take a long time. In such cases, modeling and solving MS-RCPSP without considering transfer time could produce unachievable schedules. [12] develops heuristic framework for RCPSP considering the transfer time between activities. [13] develops several adaptations of the genetic algorithm (GA) for this problem. A flow-based tabu search algorithm is developed by [14]. Recently, an efficient genetic algorithm to deal with single mode RCPSP with transfer time is proposed by [5].

In most researches, MS-RCPSP is solved in a deterministic manner in which the parameters of the problem are assumed to be known with certainty and do not change while executing the project. However, in real-world environment, the project is often unable to stick to its given baseline schedule due to external uncontrollable events such as manpower unavailability, machine breakdowns, weather changes, and hence the scheduled completion time of the project is often delayed. Generally, there are two main strategies to deal with these uncertainties and to improve the robustness of a schedule. The first strategy is to insert time buffer into the scheduling plan to prevent the propagation of disruptions throughout the schedule as much as possible ([15] [16] [17]), this strategy is usually used to deal with the uncertainty related to activity duration. The second strategy focus on robust resource allocation, which is achieved by determining the sequence of resource transfers across the activities. As a consequence, precedence relations will be added to the original precedence network using the same resource requirement, and a robust objective function is formulated to evaluate the robustness of the schedule ([18] [19]). But optimizing the robust objective function is an indirect way to deal with the activity related uncertainties. This method can only evaluate the robustness of the scheduling plan as a whole, the activity in which cannot be treated differently. The relationship between optimization function and the robustness of executing a specific activity in the project is very vague. It is also impossible to estimate the probability of the satisfaction of the robustness constraints for an activity when a schedule is made.

However, when it comes to the project in food industry or service providing scenarios, the skill of resources are often

This work was supported by the Key Program National Natural Science Foundation of China (Grant No. U1613225).

Junqi Cai is a Ph.d student of the school of automation, Beijing Institute of Technology, Beijing 100081, China. bdcaijq@gmail.com

Zhihong Peng is a Professor of the school of automation, Beijing Institute of Technology, Beijing 100081, China. She is also a researcher in Peng Cheng Laboratory, Shen Zheng 518000, China. She is the corresponding author of this paper. peng@bit.edu.cn

Shuxin Ding is with the Signal and Communication Research Institute, China Academy of Railway Sciences Corporation Limited, Beijing 100081, China. dingshuxin@rails.cn

Jingbo Sun is a student of the school of automation, Beijing Institute of Technology, Beijing 100081, China. 534112966@qq.com

978-1-7281-9093-8/20/$31.00 ©2020 IEEE

subject to a high level of uncertainty for various reasons such as weather, traffic conditions and the working enthusiasm of human resources, the transfer time of resources must be considered. The MS-RCPSP with uncertainty of resource skills is more challenging than their deterministic version. In such cases, two new requirements are important to business. The first one is the full control on the degree of conservatism for every activity. The second one is a probabilistic guarantee must be made when small deviation exists in robust related constraints. To the best of the author's knowledge, there is no method which focus on solving the above problem in existing literature. To solve the problem, this paper investigate multi-skill resource constrained project scheduling problem with transfer time and skill uncertainty, a new problem formulation and its robust counterpart are proposed. Based on the robust counterpart, a novel genetic algorithm is presented. The experiment result shows effectiveness of our proposed method in providing more robust schedule under resource skill uncertainty.

II. MULTI-SKILL RESOURCE CONSTRAINED PROJECT SCHEDULING PROBLEM WITH TRANSFER TIME AND SKILL UNCERTAINTY

A. Problem Description

The problem can be described as follows: a set of activities $V = \{0, \ldots, i, \ldots, j, \ldots, n+1\}$ to be processed, 0 and n+1 represent dummy activities respectively. Each activity has a processing time p_j. Precedence constraints exist between activities. In the project, there are K resources $\{1, \ldots, k, \ldots, K\}$ and LN types of skills $\{1, \ldots, l, \ldots, LN\}$. Each resource masters one or several types of skill. Because the uncertainty in evaluating the skill professional level of a resource, in this paper, the skill professional level is represent by an interval. It takes some time for a resource to travel from one activity to another. Each activity requires one or several types of skill at a minimum professional level respectively. Each resource can only process one activity at a time. Each activity should be processed only once. MS-RCPSP with transfer time and skill uncertainty can be decomposed into two sub-problems: task scheduling problem and resource assigning problem. The objective of the problem is to schedule all activities to satisfy the precedence and resource constraints in such a way that the makespan of project is minimized. Figure 1 shows an example of MS-RCPSP with transfer time and uncertainty skill.

B. Deterministic Model

We make the following assumptions of the problem discussed in this paper: (1) Preemption is not allowed, that is, if an activity is being processed it must be processed to the end of that activity. (2) Each skill can be performed with different proficiency level corresponding to the ability that the resource masters. (3) Each resource can only contribute one skill it masters when the resource the resource is assigned to perform a certain activity. (4) An activity can start to be processed only if all the resource assigned to it have been transferred to the activity's location.

Using the notations in Table I, the deterministic problem formulation of multi-skill resource constrained project scheduling problem with transfer time can be formulated as follows:

$$\min \sum_{t=ES_{n+1}}^{LS_{n+1}} ts_{n+1,t} \tag{1}$$

s.t.

$$\sum_{t=ES_j}^{LS_j} ts_{jt} - \sum_{t=ES_i}^{LS_i} ts_{it} - p_i - \Delta_{ij} \cdot z_{ijk} \geq 0, \quad i \in V \backslash \{n+1\}, j \in V \backslash F_i, k \in K, \tag{2}$$

$$\sum_{t=ES_j}^{LS_j} x_{jkt} \leq 1, \ k \in R, j \in V_k, \tag{3}$$

$$\sum_{j \in V_k} \sum_{\tau=\max\{ES_j, t-p_j+1\}}^{\min\{LS_j, t\}} x_{jk\tau} \leq 1, \ k \in R_j, t \in T, \tag{4}$$

$$x_{jkt} \leq s_{jt}, \ j \in V \backslash \{0, n+1\}, k \in R_j, t \in \{ES_j, \ldots, LS_j\}, \tag{5}$$

$$x_{jkt} + 1 \geq s_{jt} + \sum_{l \in L^k \cap L_j} y_{jkl}, \quad j \in V \backslash \{0, n+1\}, k \in R, t \in \{ES_j, \ldots, LS_j\}, \tag{6}$$

$$\sum_{k \in R^l} y_{jkl} \cdot u_{kl} \geq r_{jl}, \ j \in V \backslash \{0, n+1\}, l \in L_j, \tag{7}$$

$$\sum_{t=ES_j}^{LS_j} x_{jkt} = \sum_{l \in L^k \cap L_j} y_{jlk}, \ j \in V \backslash \{0, n+1\}, k \in R_j, \tag{8}$$

$$\sum_{i \in V \backslash H_j} z_{ijk} \geq \sum_{e \in V \backslash F_j} z_{jek}, \ j \in V \backslash \{0\}, k \in R, \tag{9}$$

$$\sum_{j \in V} z_{ijk} \leq 1, \ i \in V, k \in K, \tag{10}$$

$$\sum_{t=ES_j}^{LS_j} x_{jkt} = \sum_{i \in V \backslash \{n+1\}} z_{ijk}, j \in V \backslash \{0, n+1\}, k \in R, \tag{11}$$

$$s_{jt} \in \{0,1\}, \ j \in V, t \in \{ES_j, \ldots, LS_j\}, \tag{12}$$

$$x_{jkt} \in \{0,1\}, \ j \in V_k, t \in \{ES_j, \ldots, LS_j\}, k \in R, \tag{13}$$

$$y_{jkl} \in \{0,1\}, \ j \in V_k, l \in L^k \cap L_j, k \in R. \tag{14}$$

C. Robust Model

Considering the uncertainty in evaluating the skill professional level of resources, the skill professional level is modeled as box uncertainty set. $u_{kl}, k \in R, l \in L$ denotes the level of skill l that resource k can provide. u_{kl} takes values according to a symmetric distribution with mean equal to $\overline{u}_{kl}$ from $[\overline{u}_{kl} - \hat{u}_{kl}, \overline{u}_{kl} + \hat{u}_{kl}]$. For each activity j, parameter Γ_j is introduced to adjust the robustness against the conservatism of the solution, up to $\lfloor \Gamma_j \rfloor$ of parameters u_{kl} are allowed to change, and for one specific u_{kl} changes by $\left(\Gamma_j - \lfloor \Gamma_j \rfloor\right) \hat{u}_{kl}$. Referring the robust modeling method proposed by [20], the robust counterpart can be formulated as follows:

min (1),
s.t. (2) to (6), (8) to (14),

$$\sum_{k \in R_j} u_{kl} y_{jkl} - z_{jl} \Gamma_j - \sum_{k \in R_j} w_{kl} \geq r_{jl}, \ j \in V, l \in L, \tag{15}$$

$$z_{jl} + w_{kl} \geq u_{kl}, \ j \in V \backslash \{0, n+1\}, k \in R, l \in L, \tag{16}$$

$$\overline{u}_{kl} - \hat{u}_{kl} \geq u_{kl} \leq \overline{u}_{kl} + \hat{u}_{kl}, \ k \in R, l \in L, \tag{17}$$

$$w_{kl} \geq 0, \ k \in R, l \in L, \tag{18}$$

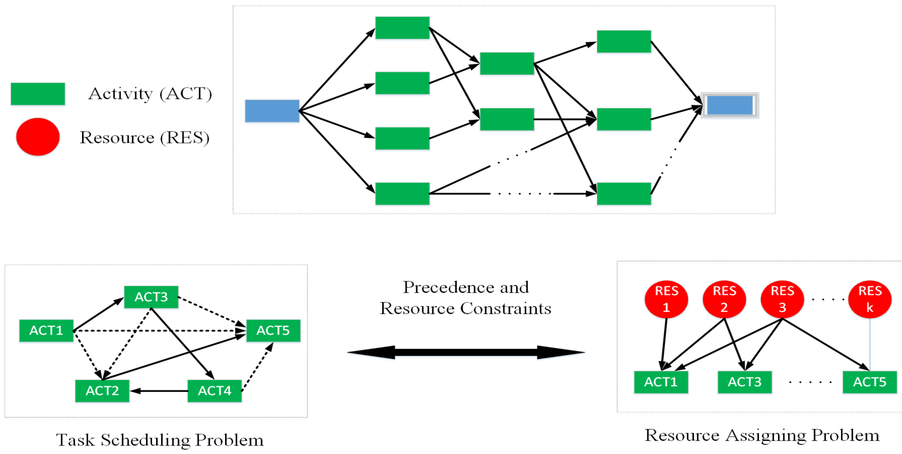

Fig. 1. Problem Illustration

TABLE I
THE NOTATIONS

Notations	
$V=\{0,\ldots,i,\ldots,j,\ldots,n+1\}$	set of activities, activity 0 and n+1 are dummy activities.
P_j	set of activities which are the direct predecessor of activity j.
F_j	set of activities which are the direct or indirect predecessor of activity j.
S_j	set of activities which are the direct successor of activity j.
H_j	set of activities which are the direct or indirect successor of activity j.
$R=\{1,\ldots,k,\ldots K\}$	set of resources.
$L=\{1,\ldots,l,\ldots LN\}$	set of skills.
L_j	set of skill required by activity j.
L^k	set of skill mastered by resource k.
V_k	set of activities requiring skills mastered by resource k.
R_j	set of resource that can contribute at least one skill required by activity j.
R^l	set of resources which can provide skill l.
r_{jl}	the levels of skill l required to process activity j.
u_{kl}	the units of skill l processed by resource k.
c_{jkl}	the cost of resource k to perform activity j with skill l.
p_j	the processing time of activity j.
f_j	the finish time of activity j.
sk_j	the slack time of activity j.
Δ_{ij}	the time needed to transfer resource from activity i to activity j.
$\mu(i,j)$	length of the longest path from activity i to activity j.
$T=\{0,\ldots,t,\ldots,UB\}$	set of discrete times in which activities might start, UB represent the upper bound.
$ES_j, LS_j,$	earliest and latest start time of activity j.
ST_j	the start time of activity j.
s_{jt}	decision variable to determine whether activity j to be processed at time t.
x_{jkt}	decision variable to determine whether resource k processes activity j at time t.
y_{jkl}	decision variable to determine whether resource k is assigned to activity j for performing skill l.
z_{ijk}	decision variable to determine whether resource k is transferred for i to j.

$$z_{jl} \geq 0,\ j \in V\backslash\{0,n+1\}, l \in L. \tag{19}$$

III. THE PROPOSED GENETIC ALGORITHM

In this section, we describe a new GA to solve resource-constrained project scheduling problem with transfer time and uncertainty skill. It starts with the modified regret-based biased sampling heuristic algorithm to generate activity list. Then the resource list generation algorithm is applied to make resource allocation. The makespan is selected as the fitness value of individuals. The proposed algorithm uses the two-point crossover proposed by [5]. Once the new population is obtained, a mutation operator is applied to each individual in the new population. The mutation operator work as follows: randomly select a position i from activity list and draw a random number xp_i from a uniform distribution between 0 and 1. Switch the activity in position i and $i+1$, if $i+1$ is not direct or indirect successor of i and $xp_i < \mu$, μ denotes the mutation probability. The pseudo-code of the proposed algorithm is presented in Algorithm 1.

A. Solution Representation

Our problem is an extension of RCPSP, considering the mull-skill nature and the heterogeneous of resources, so the activity-resource list is adopted in this paper as the encoding scheme. Specifically, a solution is encoded by an activity-resource list $\Lambda = [\pi, \tau]$, where $\pi = [\pi_1, \pi_2, \ldots, \pi_n]$ is the activity list which represents the priority activity sequence,

Algorithm 1: The proposed genetic algorithm

```
Generate G initial solutions using the modified regret-based biased sampling heuristic and resource list
    generation algorithm.
CG = 1 (Count of generation).
while CG ≤ L do
    Generate G offspring solutions using two-points crossover operators.
    Apply mutation operators to some of the offspring.
    Compute the fitness of each offspring.
    Add new G solutions to the previous G solutions.
    Remove solutions that do not satisfy robust constraints by solving Robust Constraints Satisfaction
        Problem(RCSP)
    Select the best G solutions as the next population.
    CG = CG + 1.
end
Return the best solution found.
```

$\tau = [\tau_1, \tau_2, \ldots, \tau_n]$ is the resource list which represents the resource allocation details for corresponding activity, τ_j represent the set of resources assigned to activity j.

B. Activity-list Generation

In this paper, the activity list is generated firstly using a modified regret-based sampling heuristic algorithm, as shown in Algorithm 2. Four different activity priority rules are involved: the latest finish time (LFT), the longest processing time (LPT), the latest start time (LST) and the Greatest Rank Positional Weight (GRPW). An iteration mode is selected randomly from {activity-mode, set-mode}.

Algorithm 2: Modified regret based sampling heuristic

```
Data: Activities, Precedence network
Result: Activity list
Initialize the activity list: π = {0}.
Initialize the eligible activity set E = {j ∈ V | P_j = 0}.
Randomly selected an activity rule from {LST, LPT, LFT, MTS }.
Randomly selected an iteration mode: {activity-mode, set-mode}.
if mode = activity-mode then
    while E ≠ ϕ do
        Compute priority value pv_j for j ∈ E
        For j ∈ E, calculate the regret value
          re_j = max_{i∈P_j, j∈E}{Δ_ij} − min_{i∈P_j}{Δ_ij} + max_{j∈E}{pv_j} − pv_j and the select
          probability ψ_j = re_j / Σ_{i∈E} re_i
        Select an activity j* ∈ E ψ_j using roulette selection method and pv_j.
        Add j* to the end of π.
        Update E = {j ∈ V | P_j ⊆ π and j ∉ π}.
    end
end
else if mode = set-mode then
    while E ≠ ϕ do
        Ecopy = E .
        while Ecopy ≠ ϕ do
            Compute priority value pv_j for j ∈ E
            For j ∈ E, calculate the regret value
              re_j = max_{i∈P_j, j∈E}{Δ_ij} − min_{i∈P_j}{Δ_ij} + max_{j∈E}{pv_j} − pv_j and the select
              probability ψ_j = re_j / Σ_{i∈E} re_i
            Select an activity j* ∈ E ψ_j using roulette selection method and pv_j.
            Add j* to the end of π.
            Remove j from Ecopy (Ecopy = Ecopy\j).
        end
        Update E = {j ∈ V | P_j ⊆ π and j ∉ π}.
    end
end
```

C. Resource-list Generation

After obtaining the activity list, a skill weight rule and a resource rule are designed to make resource allocation for the activities in activity-list. Skill weight rule defines the probability of the skill priority during the process of resource allocation. Similarly, the resource weight defines the sequence for resource to be allocated to a given activity. Resource rules are used to determine the order in which resources are allocated. The resource rule are defined as follows:

Resource-rule 1: Arrange the activities in an ascending order according to their transfer time to the target activity.

For an activity which has been assigned resources, the skill sequence (denoted by SE) describes the priority of skill type provided by resources. For a specific activity j waiting for the resource to be allocated, the relative relationship between its skill requirement and the total supply of the corresponding available resources determines the probability of the skill priority. The chosen probability of each skill is calculated as:

$$\kappa_l = \frac{r_{jl}}{\sum_{k \in AP_j} u_{kl}} \tag{20}$$

$$ps_l = \frac{\kappa_l}{\sum_{l^* \in L_j, l^* \notin SE} \kappa_{l^*}} \tag{21}$$

To make sure the robustness is satisfied for each activity after the original resource list has been generated, the Robust Constraints Satisfaction Problem (RCSP) is checked at each iteration. Assuming that j is specified in the precious algorithm according to the activity list, l is chosen form the skill sequence (SE), TA_{jl} represent the temporarily store pre-allocated resources which are to perform activity j with skill l. Based on robust model presented in Section II.B, the constraints of the RCSP at each iteration is defined as (15) to (19). Algorithm 3 shows the complete process to generate a robust resource list for a given activity list.

IV. Computational Result

To evaluate the proposed algorithm, we implemented it in C++ and tested it on a computer with Intel Core i5-6300 (2.3GHz) processor with 8GB RAM. There is no benchmark in the public literature or internet that is fully applicable to the problem investigated in this paper.

Hence, we modified the multi-skill resource constrained project scheduling problem instance generator and the parameters proposed by [21] to generate new instances which fit the problem investigated in this paper. The skill uncertainty is represent by a rate randomly selected from interval $[NCR, MCR]$. NCR represents the minimum uncertainty rate and MCR represents the maximum uncertainty rate. Three new instance sets are generated, each of them consists 300 instances,the parameter of the instance set is show in Table III. In each set, it is divided into n classes according to the different combination of "SF", "NC" and "RSS". We generated 20 instances for each combination of parameters.

Algorithm 3: Resource-list generation algorithm

```
Data: Activity-list(denoted by π), Resources, Activity processing time and skill requirement
Result: Resource list
t = 1.
Initialize the set of completed activities: CA = {0}.
Initialize the finish time of activities: f_j = 0, j ∈ V.
Initialize the locations of resources: LR_0 = R and LR_j = ϕ, j ∈ V\{0}.
Initialize the set of scheduled activities: SA = {0}.
Update the set of available activities: AA = {i ∈ V | P_i = CA}.
Arrange the activities in AA according to their sequence in Activity-list.
while |SA| < |V| do
    while AA ≠ ϕ do
        j = the first activity of AA .
        AP_j = ϕ (Set of resources that can provide skills to perform skill l)
        foreach i ∈ F_j do
            if f_i + Δ_ij ≤ t then
                AP_j = AP_j ∪ {k ∈ LR_i}.
            end
        end
        if Σ_{k∈AP_j} u_kl ≥ r_jl, l ∈ L_j then
            Arrange AP_j using Resource-rule.
            Generate the skill sequence SE using skill sequence probability.
            for l ∈ SE do
                v_jl = 0.
                TA_jl = ϕ (Temporarily store pre-allocated resources).
                while v_jl ≤ r_jl and AP_j ≠ ϕ do
                    k* = the first resource in AP_j
                    if u_k*l > 0 then
                        v_jl = v_jl + u_k*l , TA_jl = {k*} ∪ TA_jl , AP_j = AP_j\{k*}.
                    end
                    if v_jl ≥ r_jl then
                        Soving Robust Satisfaction Problem.
                    end
                end
            end
            if The robustness for each skill of j can be both santisfied then
                Change the position of resources {k ∈ TA_jl | ∀l ∈ L_j} to j
                f_j = t + p_j, SA = SA ∪ {j}.
            end
        end
        AA = AA\{j}.
    end
    t = t + 1
    Update CA = {∀j ∈ V | f_j < t}.
    Update AA = {∀j ∈ V − SA | P_j ⊂ CA}.
    if AA ≠ ϕ then
        Arrange the activities in AA according to the order in activiy-list.
    end
end
```

TABLE II

EXPERIMENT PARAMETERS

Parameter	Value
The number of execution on one instance	20
Population size	50
The numbers of individuals generated by initialization	50
Generation limitation	20
Crossover rate	0.8
Mutation rate	0.04

A comparative experiment is conducted between the propose genetic algorithm and a flow based tabu search [14], each instance is computed 20 times with different algorithm. The parameters of tabu search algorithm is set as Parameter Set 1 which is illustrated in [14]. Table II shows the parameter settings for the proposed algorithm. Since the optimal value of the makespan of an instance is very hard to obtain, we report the minimum and maximum deviations (Φ_{min}, Φ_{max}) from the critical path lower bound. Furthermore, we list the minimal and maximal improvements ($\Phi^0_{min}, \Phi^0_{max}$) from the initial solution. As well as the average computation

TABLE III

BASE PARAMETERS

Instance Name	set1	set2	set3
nAct	40	80	120
nStart	6	6	9
nFinish	7	7	10
MaxSucc	8	10	12
MaxPred	8	10	12
K	40	60	90
L	3	3	3
MSU	30	30	30
TS	{150,160,230}	{250, 360, 168}	{450, 820, 390}
maxSkill	3	3	3
NCR	0.1	0.1	0.2
MCR	0.4	0.3	0.5
Γ_j	$[10.5, 20.5]$	$[10.5, 30.5]$	$[15.5, 40.5]$

time (CT_{avg}). Due to the uncertainty in resource skill may cause insufficient resource allocation, we also calculated resource insufficient rate (RIR), which is short for the proportion of activity that do not have been allocated sufficient resources. The result is shown in Table IV. It can be seen that the proposed algorithm outperforms the compared algorithm in resource insufficient rate. The average RIR obtained by the proposed algorithm for 3 sets of instances is 5.1%, 9.5%, 6.4% respectively. There is significant decrease compared to the average RIR obtained by flow-based tabu search [14], which is 25.8%, 44.2%, 42.1%. The maller RIR means the more robustness of the schedule against uncertainty. The improvement rate shows both of the algorithms improve the initial solution to some extent. When it comes to the deviation from the critical path lover bound, the result shows the makespan of proposed algorithm is longer then flow-based tabu search [14]. However, considering that the planning result of [14] has much higher RIR, it will face more delay and uncertainty when executing this schedule in practice. The computation time of proposed algorithm is a little longer then the computation time of [14].

V. CONCLUSION

In this paper, we introduced a novel robust genetic algorithm for MS-RCPSP with transfer time and skill uncertainty. A new deterministic model and its robust counterpart are proposed. The problem is decomposed into task scheduling problem and resource assigning problem. A modified regret base sampling heuristic algorithm is proposed to generate high quality initial population for task scheduling problem, a new resource list generation algorithm is proposed to generate high quality initial population for resource assigning problem while considering the robust constraints. The computation result shows the proposed algorithm can significantly improve the robustness of the schedule under

TABLE IV

EXPERIMENT RESULT

Instance				The proposed GA						Flow-based Tabu search (2016)					
	SF	NC	RSS	Φ_{min}	Φ_{max}	Φ^0_{min}	Φ^0_{max}	CT_{avg}	RIR	Φ_{min}	Φ_{max}	Φ^0_{min}	Φ^0_{max}	CT_{avg}	RIR
set1	0.5	1.5	0.8, 0.6, 0.5	9.7%	11.2%	1.1%	1.5%	3.1	**5.3%**	8.4%	9.7%	1.2%	1.8%	2.9	**20.8%**
	0.5	1.5	0.5, 0.8, 0.6	9.6%	13.0%	1.1%	1.5%	3.5	**2.6%**	8.5%	11.3%	1.2%	1.7%	3.2	**34.2%**
	0.75	1.5	0.8, 0.6, 0.5	10.9%	12.8%	1.2%	1.6%	3.3	**4.7%**	9.4%	11.4%	1.4%	1.9%	2.7	**23.1%**
	0.75	1.5	0.5, 0.8, 0.6	10.7%	12.4%	1.3%	1.5%	3.3	**4.5%**	9.4%	10.9%	1.5%	1.8%	3.1	**22.5%**
	1	2.1	0.8, 0.6, 0.5	9.3%	11.2%	1.1%	1.8%	3.3	**6.8%**	8.4%	9.7%	1.3%	2.2%	3.0	**23.8%**
	1	2.1	0.5, 0.8, 0.6	9.7%	11.8%	1.7%	1.8%	3.2	**6.4%**	8.6%	10.4%	2.0%	2.2%	2.8	**30.2%**
average				10.0%	12.1%	1.3%	1.6%	3.3	5.1%	1.4%	1.7%	0.3%	0.4%	2.9	25.8%
set2	0.5	1.5	0.8, 0.6, 0.5	6.7%	8.8%	1.2%	2.3%	34.0	**11.1%**	6.1%	8.1%	1.5%	3.0%	24.7	**43.8%**
	0.5	1.5	0.5, 0.8, 0.6	7.9%	9.1%	1.7%	3.4%	35.6	**7.2%**	7.2%	8.4%	2.0%	4.2%	21.2	**41.7%**
	0.75	1.5	0.8, 0.6, 0.5	6.5%	8.6%	1.3%	2.6%	31.3	**11.1%**	6.1%	7.9%	1.8%	3.4%	21.6	**46.1%**
	0.75	1.5	0.5, 0.8, 0.6	7.0%	8.5%	2.8%	3.2%	27.8	**7.6%**	6.3%	7.6%	3.5%	4.0%	21.4	**37.5%**
	1	2.1	0.8, 0.6, 0.5	6.5%	8.7%	2.8%	3.1%	34.5	**9.9%**	5.9%	7.8%	3.4%	3.9%	23.9	**48.9%**
	1	2.1	0.5, 0.8, 0.6	7.1%	8.2%	3.2%	3.7%	29.4	**10.1%**	6.4%	7.4%	4.3%	4.9%	27.2	**47.1%**
average				7.0%	8.7%	2.2%	3.1%	32.1	9.5%	1.1%	1.2%	0.7%	0.8%	23.3	44.2%
set2	0.5	1.5	0.8, 0.6, 0.5	6.1%	7.8%	4.3%	5.8%	322.6	**5.0%**	5.6%	7.2%	5.6%	8.2%	178.1	**53.4%**
	0.5	1.5	0.5, 0.8, 0.6	5.7%	8.0%	3.1%	3.8%	205.3	**9.9%**	5.2%	7.4%	4.1%	5.3%	136.1	**42.3%**
	0.75	1.5	0.8, 0.6, 0.5	5.8%	7.5%	3.1%	4.9%	299.4	**5.2%**	5.3%	6.9%	4.1%	6.5%	135.4	**32.7%**
	0.75	1.5	0.5, 0.8, 0.6	6.7%	7.7%	4.4%	5.6%	315.6	**6.6%**	6.2%	7.0%	6.3%	7.8%	151.9	**27.4%**
	1	2.1	0.8, 0.6, 0.5	6.9%	7.9%	3.7%	4.9%	318.3	**7.8%**	6.4%	7.3%	4.6%	6.9%	214.8	**60.5%**
	1	2.1	0.5, 0.8, 0.6	6.2%	7.4%	5.0%	5.7%	330.0	**3.6%**	5.6%	6.9%	6.8%	8.3%	234.9	**36.6%**
average				6.2%	7.7%	3.9%	5.1%	298.5	6.4%	0.9%	1.2%	1.1%	1.4%	175.2	42.1%

resource skill uncertainty, while there is only a small increase in project's makespan and computation time.

REFERENCES

[1] Zhu, G., Bard, J.F. and Yu, G. A branch-and-cut procedure for the multimode resource-constrained project-scheduling problem. *INFORMS Journal on Computing*, 18(3), 2006, pp.377-390.

[2] Brucker, P., Knust, S., Schoo, A. and Thiele, O. A branch and bound algorithm for the resource-constrained project scheduling problem. *European journal of operational research*, 107(2), 1998, pp.272-288.

[3] Lin, J., Zhu, L. and Gao, K. A genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem. *Expert Systems with Applications*, 140, 2020, p.112915.

[4] Almeida, B.F., Correia, I. and Saldanha-da-Gama, F. Priority-based heuristics for the multi-skill resource constrained project scheduling problem. *Expert Systems with Applications*, 57, 2016, pp.91-103.

[5] Kadri, R.L. and Boctor, F.F. An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case. *European Journal of Operational Research*, 265(2), 2018, pp.454-462.

[6] Chen, R.M. Particle swarm optimization with justification and designed mechanisms for resource-constrained project scheduling problem. *Expert Systems with Applications*, 38(6), 2011, pp.7102-7111.

[7] Koulinas, G., Kotsikas, L. and Anagnostopoulos, K. A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem. *Information Sciences*, 277, 2014, pp.680-693.

[8] Pan, N.H., Lee, M.L. and Chen, K.Y., 2009. Improved Tabu Search algorithm application in RCPSP. In *Proceedings of the international multiconference of engineers and computer scientists*, Vol.1, 2009, p44..

[9] Bouleimen, K.L.E.I.N. and Lecocq, H.O.U.S.N.I. A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European journal of operational research*, 149(2), 2003, pp.268-281.

[10] Fang, C. and Wang, L., 2012. An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem. *Computers & Operations Research*, 39(5), 2012, pp.890-901.

[11] Giran, Omer, Rasim Temur, and Gebrail Bekdaş. ”Resource constrained project scheduling by harmony search algorithm.” *KSCE Journal of Civil Engineering*, 21.2, 2017, pp.479-487.

[12] Krüger, D. and Scholl, A. A heuristic solution framework for the resource constrained (multi-) project scheduling problem with sequence-dependent transfer times. *European Journal of Operational Research*, 197(2), 2009, pp.492-508.

[13] Cai, Z. and Li, X. A hybrid genetic algorithm for resource-constrained multi-project scheduling problem with resource transfer time. In 2012 IEEE International Conference on Automation Science and Engineering (CASE), 2012, pp.569-574.

[14] Poppenborg, J. and Knust, S., 2016. A flow-based tabu search algorithm for the RCPSP with transfer times. *Or Spectrum*, 38(2), 2016, pp.305-334.

[15] Van de Vonder, S., Demeulemeester, E. and Herroelen, W. Proactive heuristic procedures for robust project scheduling: An experimental analysis. *European Journal of Operational Research*, 189(3), 2008, pp.723-733.

[16] Hu, X., Cui, N. and Demeulemeester, E. Effective expediting to improve project due date and cost performance through buffer management. *International Journal of Production Research*, 53(5), 2015, pp.1460-1471.

[17] Zhang, J., Song, X. and Díaz, E. Critical chain project buffer sizing based on resource constraints. *International Journal of Production Research*, 55(3), 2017, pp.671-683.

[18] Leus, R. and Herroelen, W. Stability and resource allocation in project planning. *IIE transactions*, 36(7), 2004, pp.667-682.

[19] Liang, Y., Cui, N., Hu, X. and Demeulemeester, E. The integration of resource allocation and time buffering for bi-objective robust project scheduling. *International Journal of Production Research*, 2019, pp.1-16.

[20] Bertsimas, D. and Sim, M. The price of robustness.*Operations research*, 52(1), 2004, pp.35-53.

[21] Almeida, B.F., Correia, I. and Saldanha-da-Gama, F. An instance generator for the multi-skill resource-constrained project scheduling problem. Faculdade de Ciências da Universidade de Lisboa-Centro de Matemática, Aplicações Fundamentais e Investigação Operacional, 2015.