International Journal of Industrial Engineering & Production Research (2017)

December 2017, Volume 28, Number 4
pp. 429- 439

DOI: 10.22068/ijiepr.28.4.429 http://IJIEPR.iust.ac.ir/

A New Heuristic Algorithm for The Preemptive and Non-Preemptive Multi-Mode RCPSPs

Parham Azimi [*] **& Naeim Azouji**

Parham Azimi, *Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University*
Naeim Azouji, *Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University*

KEYWORDS
Optimization via Simulation,
Multi-mode Resource Constraint Project Scheduling Problem,
Genetic Algorithm.

ABSTRACT
In this paper, a novel modeling and solving method have been developed to address the so-called resource-constrained project scheduling problem (RCPSP) where project tasks have multiple modes, and also the preemption of activities is allowed. To solve this NP-hard problem, a new general optimization via simulation (OvS) approach has been developed which is the main contribution of the current research. In this approach, the mathematical model of the main problem is relaxed and solved; the optimum solutions were then used in the corresponding simulation model to produce several random feasible solutions to the main problem. Finally, the most promising solutions were selected as the initial population of a genetic Algorithm (GA). To test the efficiency of the problem, several test problems were solved by the proposed approach; according to the results, the proposed concept has a good performance to solve such a complex combinatorial problem. Also, the concept could be easily applied to other similar combinatorics.

1. Introduction

A project is an individual process with a set of activities to produce a product or present some special services. Project managers have some limitations, such as resources, cost and time, to achieving their goals; therefore, the scheduling process is very important to make a balance among the constraints for them. Despite the existence of several uncertain events in real-world applications, the majority of previous researchers have focused on resource-constrained scheduling problems (RSPSPs) with deterministic parameters. Although we have resource constraints in a scheduling problem, allocating the resources to the activities is one of the most significant duties of a project manager (Rasekh and Brumbelow, 2015). In the traditional RCPCP, each activity has a single execution. Multi-mode Resource-constrained project scheduling problem (MRCPSP) is a more general version than the RCPSP in which each activity can be implemented in several modes (Ghamginzadeh et al., 2014). Each mode needs

[*] **Corresponding author:** ***Parham Azimi***
Email: ***p.azimi@yahoo.com***
Received 13 February 2017; revised 9 October 2017; accepted 15 November 2017

its own duration and resource consumption in order to be implemented. In the MRCPSP, the objective is to decide when an activity begins and how it is performed so that the goal of the project is optimized. On the other side, the project may have some renewable and nonrenewable resources. In real-world cases, where lots of non-deterministic events happen, the simulation technique is one the most promising tools to model the problem and check the results after making the necessary changes. However, the simulation technique could be integrated with meta-heuristic algorithms in the optimization process of combinatorics. This property helps us to find a set of parameters optimizing the efficiency of system. This approach is called optimization via simulating (OvS) technique (Hong and Nelson, 2009). Rasekh and Brumbelow (2015) studied a dynamic simulation-based optimization model for adaptive management of urban water distribution system contamination threat. In this research, the OvS technique is applied to the scheduling problem using the Genetic algorithm (GA) and the simulation technique. Simulation allows us to carry out a lot of offline analyses on system performance. The basic goal of a project scheduling problem is minimizing the makespan (project completion time) considering precedence constraints among activities and also available resources. At the beginning of 1960s, scheduling problems was discussed for allocating resources to the activities to create a balance between the total cost and the makespan. RCPSP includes activities that should be planned considering precedence constraints and resources to reduce the completion time of the project. Brucker et al. (1999) introduced the RCPSP including non-preemption activities. However, Bianco et al. (1999), Brucker and Knust (2001), Debels and Vanhoucke (2008), Demeulemeester and Herroelen (1996), and Nudtasomboon and Randhawa (1997) considered preemption in their models. It means that we could start and stop the execution of an activity after each unit time. The preemption in activities could be shown by β in α│β│γ. Another aspect of a RCPSP is the resource constraints. In a simple RCPSP, we have only a renewable resource which will be available in each period of time completely. In a multi-mode RCPSP, we have two different kinds of renewable and non-renewable resources most of the time. This version was developed by Slowinski (1981) and Weglarz (1981). A delay in an activity j(L_j)is a difference between its finishing time and its due time (dj). Kolisch (2000), Vianaand De Sousa (2000) considered weighted-tardiness objective functions. Nudtasomboon and Randhawa (1997) represented minimum–maximum tardiness and also minimum weighted-tardiness. Vianaand De Sousa (2000) considered weighted-tardiness objective functions. Nudtasomboon and Randhawa (1997) represented minimum–maximum tardiness and also minimum weighted-tardiness. Slowinski (1981) was the first to study the multi-objective RCPSP. He presented a linear programming model for the multi-objective, multi-mode RCPSP with the consideration of resource constraints. He discussed the usability of goal programming and fuzzy linear programming to solve this problem. Objective functions used in this research include project completion time, net present value, total resource consumption, total number of the delayed activities, and weight of consumed resources. Al-fawzan and Haourai (2005) considered MRCPSP with limited resources and proposed a two-objective Tabu search algorithm to minimize the makespan and maximize the robustness. Vianaandde Sousa (2000) proposed multi-objective annealing simulation and Tabu search algorithms to minimize: the makespan, weight lateness of activities, and violation of resource constraints. Abbasi et al. (2006) studied RCPSP with renewable resource constraints with two objective functions: Makespan and robustness. They proposed a simulation annealing algorithm along with the weighted summation method to deal with the two-objective problem. Abdelaziz et al. (2007) considered MRCPSP with renewable limitations and suggested a multi-objective ant colony algorithm to find non-dominant solutions. The objectives considered in this article include makespan, project costs, and probability of the project success. Ballestin and Blanco (2011) presented an algorithm based on the concept of non-dominant solutions. They also proposed special rules to help solving the problem. Nabipoor-Afruzi et al. (2013) considered a multi-mode resource-constrained discrete time–cost tradeoff problem and solved it with an adjusted fuzzy dominance genetic algorithm. Aboutalebi et al. (2012) proposed NSGA-II and MOPSO algorithms to solve this problem, and according to some defined indices, they showed that NSGA-II is more efficient than MOPSO is. Kazemi and Tavakoli-Moghadam (2011) studied the multi-objective RCPSP considering the maximization of the net present value and minimization of the project makespan in terms of the renewable resource constraints, although there are usually non-renewable resource constraints to execute activities in the real-world applications. For example, in construction projects, non-renewable resources are very important in the project scheduling such as cement, plaster, ironware, etc. Hence, adding this constraint results in a more realistic model. On the other hand, because this problem is a multi-mode problem and in each mode, a certain level of non-renewable resources is needed to perform each activity, defining non-renewable resources in the problem model seems to be essential. Thus, to get closer to the reality, we consider a multi-objective RCPSP problem (MRCPSP) considering non-renewable and renewable resource constraints to minimize the project makespan. In addition, preemption of activities is allowed in the model, so the problem is named as P-MRCPSP. Concerning the P-MRCPSP, there are a few studies in the literature. Najafi and Majlesi (2014) investigated the MRCPSP with just renewable resources, and the preemption of activities was allowed. They developed a GA algorithm to solve the problem. Recently, Yongyi et al. (2015) developed a hybrid particle swarm optimization procedure to solve the preemptive resource-constrained project scheduling problem in which a maximum of one interruption per activity is allowed. In this research, a new multi-objective meta-heuristic algorithm is developed for the P-MRCPSP, including nonrenewable and renewable resources, based on an OvS approach to finding the near-optimum solutions as the main contribution of the paper. The paper is organized as follows:

In Section 2, the mathematical model of the problem is presented. In the next section, the framework of the OvS algorithm, including the parameter definitions, is proposed. In Section 4, the experiment results and discussions are provided. Finally, conclusions and recommendations for the future researches are mentioned in Section 5.

2. Problem Modeling

The project is represented as an activity-on-the-node network G=(N,A), where N is the set of activities and A is the set of pairs of activities in which a finish-start precedence relationship exists in each pair. A set of activities, numbered from 1 to |N| with a dummy start node 0 and a dummy end node |N|+ 1. In the P-MRCPSP, activities are allowed to be preempted at any time and restarted later on at no additional cost.

2-1. Mathematical model

The parameters in the P-MRCPSP can be conceptually formulated as follows:

R^{ρ} : Set of renewable resource.

R^{v} : Set of nonrenewable resource.

m_i: Each activity i ∈ N is performed in a mode m_i, which is chosen out of a set of $|M_i|$ different execution modes M_i= {1, ... , $|M_i|$}.

d_{im_i}: The duration of activity i, when executed in mode m_i .

$r^{p}_{im_ik}$: Each mode m_i requires $r^{p}_{im_ik}$from the kth renewable resource units (k ∈R^{ρ}).

$r^{v}_{im_ik}$: Each mode m_i requires $r^{v}_{im_ik}$from the kth nonrenewable resource units (k ∈R^{v}).

a^{p}_{k} : Constant availability of each renewable resource (k∈R^{ρ}) throughout the project horizon.

a^{v}_{l} : Total available of nonrenewable resource (l∈R^{v}) .

$S_{i.v}$: Starting time for the v^{th} part of each activity ($v \in \{1.\ d_{im_i} - 1\}$).

$\mathrm{X_{im_i}}$: Equals to 1, if activity i is executed in mode m_i, otherwise 0.

$S\ (t)$: Denotes the set of activities in progress in period [t-1,t];t ∈ {1, ... ,$\mathrm{S_{n+1.0}}$}

M: a sufficiently big enough positive number.

According to Ghamginzadeh et al. (2014), the P-MRCPSP can be formulated as follows:

Problem 1

$$\mathrm{Min.\,S_{n+1,\,0}} \tag{1}$$

s. t .

$$(S_{i.\,d_{im_i}-1}+1)\left(1-M\mathrm{X_{im_i}}\right) \le s_{j.\,0} \qquad \forall (i.\ j) \in A \tag{2}$$

$$(S_{i.\,v-1}+1)\left(1-M\mathrm{X}_{\mathrm{im_i}}\right) \le S_{i.v} \qquad \forall i \in N.\ \forall v \in \left\{1.\ d_{im_i}-1\right\} \tag{3}$$

$$\sum_{i\in S(t)} r^{p}_{im_ik}\left(\sum_{\forall m_i\in M_i}\mathrm{X}_{\mathrm{im_i}}\right) \le a^{p}_{k}\forall k\in R^{\rho} \qquad \forall m_i \in M_i \tag{4}$$

$$\sum_{i=1}^{|N|} r^{v}_{im_ik}\left(\sum_{\forall m_i\in M_i}\mathrm{X}_{\mathrm{im_i}}\right) \le a^{v}_{l}\forall l\in R^{v} \qquad \forall m_i \in M_i \tag{5}$$

$$S_{0.\,0}=0 \tag{6}$$

$$\textstyle\sum_{\forall m_i\in M_i}\mathrm{X}_{\mathrm{im_i}} = 1, \qquad \forall i\in N \tag{7}$$

$$S_{i.v}\in int^{+}\,.X_{im_i}\in\{0.1\}.\forall i\in N\text{،}\forall v\in\{0\text{،}d_{im_i}-1 \tag{8}$$

The objective function (1) minimizes the total makespan of the project. In constraint set (2), the earliest start time of an activity j cannot be smaller than the finish time for the last unit of duration of its predecessor i. Constraint set (3) guarantees that the starting time for every time instance of an activity has to be at least one time-unit larger than the start time for the previous unit of duration. Constraints (4) and (5) deal with the renewable and nonrenewable resource limitations, respectively. Constraint (6) forces the project to start at time instance zero. Constraint set (7) ensures that each activity is just executed in one of its available modes. Constraint (8) ensures that the activity start times assume a nonnegative integer.

2-2. The proposed algorithm

Problem 1 is a NP-Hard problem (Ghamginzadeh et al., 2014). To handle its complexity, a relaxation technique is used in this research. If the integer constraint set (8) is relaxed (using equations (9) and (10)), the resulting problem (Problem 2) is very similar to the original problem in terms of the objective functions and nearly all constraints. Problem 2 is a linear model with continuous variables which could be solved easily. The optimum solution to Problem 2 is a lower bound for the problem, because the integer constraints have been removed from Problem 1. The only issue about Problem 2 is its non-integer solutions.

$$0 \le \mathrm{X}_{\mathrm{im_i}} \le 1 \tag{9}$$

$$\mathrm{S}_{\mathrm{i.v}} \ge 0 \tag{10}$$

In Problem 2, according to constraint set (7), the summation of $\mathrm{X}_{\mathrm{im_i}}$ equals to 1 and these decision variables are also positive; therefore, they could be interpreted as a probability distribution function for each activity to be executed in each mode. The non-integer value of $\mathrm{X}_{\mathrm{im_i}}$ is used as the probability of executing activity i in mode $\mathrm{m_i}$. These continuous values will be used in the simulation software as the distribution functions of executing each activity in its available modes. In this way, the simulation will be more intelligent in producing random feasible solutions for Problem 1. This is a very smart innovation to strength the quality and speed of the proposed approach in the simulation model to make the randomly generated solutions near to unknown optimums.

Consider an example project with 8 non-dummy activities, each with 2 modes. For each mode, 1 renewable resource and 1 nonrenewable resource are presented in this paper to show our procedure. The availability of the renewable (nonrenewable) resource is 7 units. The activity-on-the-node network is shown in Fig. 1. In Table 1, duration d_{im_i} and resource requirements ($r^{p}_{im_ik}$and $r^{v}_{im_ik}$) for mode m_i of activity i are shown. Figure 2 depicts the flow chart of the proposed approach to solving the P-MRCPSP. At first, the problem is modeled similar to Problem 1. Then, its integer constraints are relaxed (according to eq. (10)) to formulate Problem 2. Problem 2 which is similar to Problem 1 (except its continuous variables instead of integer ones in Problem 1) is inserted in the GAMS software and solved. For the provided example, the optimum solutions are provided in Table 2. Then, Problem 1 is modeled in the simulation software including all constraints such that a feasible solution to Problem 1 is produced at each replication (Figure 3). Also, the optimum value of X_{im_i} is inserted in the simulation software for each activity. At each replication, the simulation software uses a mode for each activity according to its probability. For example, according to Table 2, in 53% of replications, the simulation software uses mode 1 and in 47% of cases uses mode 2 for activity 1. Now, the simulation model is replicated several times to produce a lot of feasible solutions for Problem 1. Since the execution times of activities are not so long, the simulation replication is too short. For example, producing 500 feasible solutions of the example just takes 1.1 seconds on the PC basis. Some randomly promising feasible solutions are taken from the simulation experiments and inserted in a GA algorithm to continue the optimization process. In a GA, starting from a promising initial population ensures the early termination of the algorithm and helps it to achieve high-quality solutions.

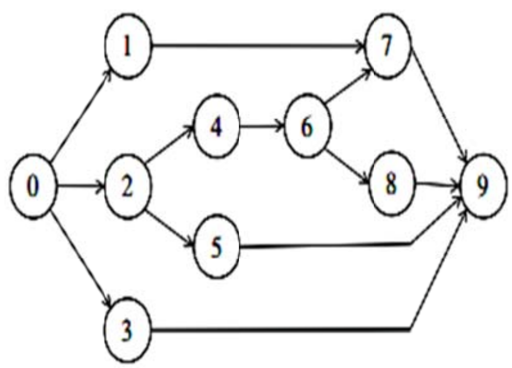

Fig. 1. Project network

Tab. 1. Project information

Act i	Mode m_i	d_{im_i}	r_{ij}^{ρ}
0	1	0	0
1	1	4	3
	2	5	2
2	1	1	3
	2	2	2
3	1	1	2
	2	2	1
4	1	2	5
	2	3	4
5	1	2	4
	2	5	3
6	1	1	1
	2	3	1
7	1	1	3
	2	3	2
8	1	2	3
	2	2	3
9	1	0	0

Tab. 2. GAMS result

Activity	m_1	m_2
1	0.53	0.47
2	0.95	0.05
3	0.51	0.49
4	0.51	0.49
5	0.84	0.16
6	0.49	0.51
7	0.78	0.22
8	0.11	0.89

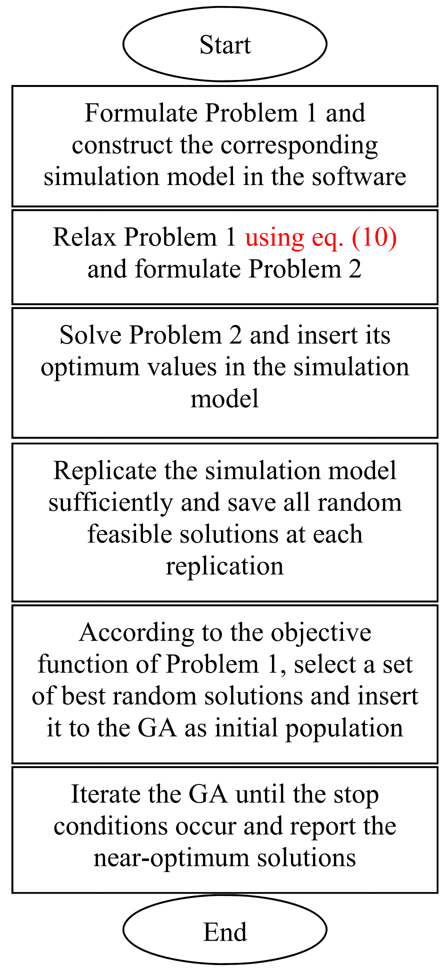

Fig. 2. Flowchart of the proposed algorithm

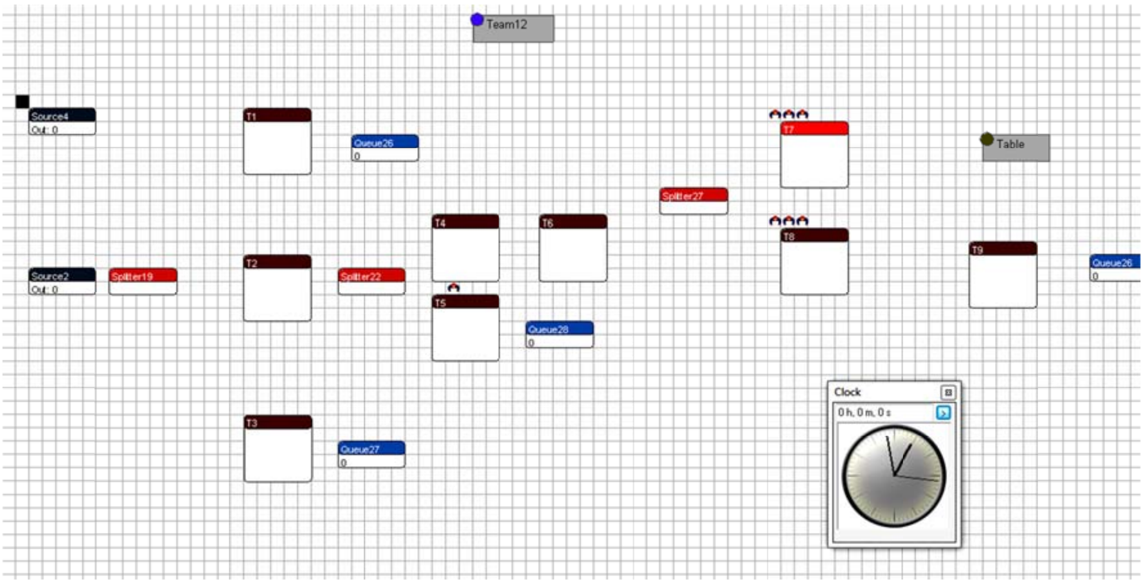

Fig. 3. The simulation model of the provided example

To simulate the models, ED 8.0 was used as the simulation software and all constraints in Problem 1 were coded in this software using its coding language named "4DScript". So, at each iteration of a simulation model, a random feasible solution to Problem 1 is generated, and its information is saved in a table (Figure 3). The validation of the simulation model is easy, because its task is just to produce feasible solutions, and one may easily check all solutions to Problem 1 constraints to see their validity.

2-3.Genetic algorithm

Genetic Algorithm (GA) is a nature-inspired optimization method. It is an iteration-based algorithm, and its basic principles are derived from genetics science. It was invented by mimicking some of the processes observed in the natural development. GA was invented by John Holland in 1967 (Nabipoor-Afroozi et al., 2013) as a well-known stochastic optimization method. In fact, genetic algorithms utilize the Darwin's principle of natural selection to find the optimal formula for prediction or pattern matching. The main application of GA is in computer. However, the GA methods are also applicable to industrial engineering, production planning, production management, IT management, and industrial management. In the proposed genetic algorithm, the chromosome consists of several parts including the earliest activity start time (e_s), the earliest activity finish time (e_f), the latest activity start time (l_s), the latest activity finish time (l_f) matrices, the matrix representing the duration for each activity (d), the matrix representing the executive mode for each activity (m), and the matrices representing the renewable and non-renewable resources. These all have a (1×N) dimension. The other part of the chromosome is the matrix, where the start and finish times of the activities are recorded. It has a dimension of (N-1) ×2.

GA starts with the forward and backward procedures. The initial population of the GA is obtained by combining the results obtained from the forward and backward procedures with the best results obtained from the simulation replications. The GA imposes resource constraints using penalty makespan (violated cmax). Total V will be defined in this part which reached by summation of V_1, V_2,V_3, and V_4. V_1, V_2 are for the first and second kinds of renewable resources; V_3, V_4 are for first and second kinds of irremovable resources. Actually, resource constraints will be covered in this part. The mutation operator in the GA is a random one. Roulette wheel has been used for selecting parents. In a crossover, all members of the parent's matrix will be changed into 0-1 values. Afterwards, a one-point crossover has been used for making a new pattern of 0 & 1 becoming a number on the 10^{th} basis and making y_1, y_2 as matrices of children.

$$\textit{Patern1chrom} = \begin{bmatrix} a_{11} & a_{12} \\ \vdots & \vdots \\ a_{n-1,1} & a_{n-1,2} \end{bmatrix} [\,0\;0 \vdots 1\;0\,] \Longrightarrow [\,0\;0\;0\;1\,]$$

$$\textit{Patern1chrom} = \begin{bmatrix} b_{11} & b_{12} \\ \vdots & \vdots \\ b_{n-1,1} & b_{n-1,2} \end{bmatrix} [\,0\;1 \vdots 0\;1\,] \Longrightarrow [\,0\;1\;1\;0\,]$$

$$y_1 = \begin{bmatrix} c_{11} & c_{12} \\ \vdots & \vdots \\ c_{n-1,1} & c_{n-1,2} \end{bmatrix} y_2 = \begin{bmatrix} d_{11} & d_{12} \\ \vdots & \vdots \\ d_{n-1,1} & d_{n-1,2} \end{bmatrix}$$

Fig. 4. An example for crossover

Using forward and backward methods, e_s and l_f will be calculated and compared to the offspring's values. It will be adjusted in 3 different ways:

1. The number in the first column is smaller than corresponding e_s.
2. The number in the second child column is greater than corresponding l_f.
3. The distance between the two elements of the columns is not less than the duration of the activity.

2-4. Tuning the parameters

In this section, first of all, the parameters of GA have been adjusted in order to improve the solution quality and the computational speed using Taguchi's design (3^k). Initial population, number of iterations, mutation, and crossover rates are our parameters chosen for parameters adjusting process. In Table 3, the levels of each parameter are shown.

Tab. 3. Parameters values

	Level 1	**Level 2**	**Level3**
Npop	50	150	250
Max it	70	100	200
Pc	0.3	0.6	0.8
Pm	0.2	0.4	0.7

The tuning process was carried out on the previous example which was taken from the **PSPLIB** digital library. Using the **Minitab** software and considering chart 3 in Taguchi's drawing, 9 different designs were experimented by the combination of parameters, 9 algorithms were executed for each mode, and the results were recorded. The optimal values are presented in Table 4. Figures 5 and 6 show the sample chromosome structure and related information.

Tab. 4. Parameters adjusting results

Parameters	Value
Npop	150
Max it	200
Pc	0.87
Pm	0.07

```
>> pop(1).Sol.Chrom

ans =

     1     7
     1     1
     8     9
     2     4
     2     9
     5     7
     8     8
     8     9
     0     0
```

Fig. 5. A sample chromosome

```
ans =

    Chrom: [9x2 double]
        d: [4 1 2 3 5 3 1 2 0]
      mod: [1 1 2 2 2 2 1 2 1]
       r1: [3 3 1 4 3 1 3 3 0]
       r2: [3 4 1 3 2 3 3 3 0]
       es: [1 1 1 2 2 5 8 8 10]
       ef: [4 1 2 4 6 7 8 9 10]
       Ls: [5 1 8 2 5 5 9 8 10]
       Lf: [8 1 9 4 9 7 9 9 10]
```

Fig. 6. Information of each activity in the genetic algorithm

In Figure 6, the first line shows the duration of each activity, the second line shows the processing mode of each activity, third line shows the usage rate of the first type sources, the fourth line shows the usage rate of the second type sources, the fifth line shows the soonest beginning time, the sixth line shows the soonest termination time, the seventh line shows the latest beginning time, and the eighth line shows the latest termination time of each activity.

3. Experiment Results

The algorithms was coded in MATLAB 7.14.0.739 software. The program was run on a PC with Core i7, 2.67GHz CPU, 4 GB RAM under Windows 8 platform. In this section, 30 samples were solved by the proposed genetic algorithm including 10 samples with 14 activities, 10 samples with 18 activities, and 10 samples with 30 activities. All samples were taken from **PSPLIB** digital library. After solving the examples, CPU time and makespan were recorded, and the results are shown briefly in Table 5.

Tab. 5. Computational results

Examples	Num. of activity	Num. of modes	Renewable resource	Nonrenewable resource	Makespan	CPU time (s)
1	14	3	2	2	18	208.094
2	14	3	2	2	37	206.146
3	14	3	2	2	37	209.139
4	14	3	2	2	18	203.989
5	14	3	2	2	18	201.626
6	14	3	2	2	32	211.828
7	14	3	2	2	16	199.625
8	14	3	2	2	21	206.307
9	14	3	2	2	23	203.435
10	14	3	2	2	19	201.419
11	18	3	2	2	15	260.399
12	18	3	2	2	38	276.499
13	18	3	2	2	25	275.019
14	18	3	2	2	36	283.358
15	18	3	2	2	39	278.776
16	18	3	2	2	31	282.844
17	18	3	2	2	28	273.976
18	18	3	2	2	34	282.395
19	18	3	2	2	30	272.964
20	18	3	2	2	22	280.133
21	30	3	2	2	57	481.639
22	30	3	2	2	39	481.321
23	30	3	2	2	45	470.277
24	30	3	2	2	39	461.958
25	30	3	2	2	27	455.147
26	30	3	2	2	44	469.242
27	30	3	2	2	40	470.459
28	30	3	2	2	37	485.936
29	30	3	2	2	44	491.581
30	30	3	2	2	62	485.17

In order to test the efficiency of the proposed algorithm, the results were compared to the ones obtained from a normal genetic algorithm, without using the simulation results. For this reason, all 30 samples were solved using the normal GA, and the results are listed in Table 6. The first column in Table 6 shows the number of samples in PSPLIB. The second column shows the result of using the proposed algorithm for the problem makespan. The third column shows the results using GA, alone. The last column shows the percentage difference between the two algorithms using the following formula:

$$\frac{\text{The } prop.alg.makespan - The \text{ GA makespan}}{The \text{ GA makespan}} * 100 \qquad (11)$$

Tab. 6. Comparison result

No.	Makespan using Prop. Alg.	Makespan using GA	The % difference
1	17	18	-5.56
2	36	37	-2.70
3	36	37	-2.70
4	17	17	0.00
5	18	19	-5.26
6	30	31	-3.23
7	15	15	0.00
8	19	21	-9.52
9	20	20	0.00
10	18	18	0.00
11	14	12	16.67
12	31	31	0.00
13	23	25	-8.00
14	31	31	0.00
15	35	37	-5.41
16	29	30	-3.33
17	26	29	-10.34
18	34	33	3.03
19	29	29	0.00
20	20	21	-4.76
21	47	48	-2.08
22	35	37	-5.41
23	39	40	-2.50
24	33	35	-5.71
25	24	25	-4.00
26	34	35	-2.86
27	40	41	-2.44

28	32	33	-3.03
29	39	40	-2.50
30	58	60	-3.33

According to Table 6, nearly in all samples, the proposed algorithm behaved better than the normal GA, in terms of the solution quality. On average, the proposed algorithm solutions are 2.50% better than the GA solutions. Concerning the computational speed, modeling the problem in the simulation software and also the simulation replications are low time consuming processes. On the other side, the high-quality solutions obtained from the simulation replications caused the algorithm to be terminated very soon. On average, both algorithms have the same computational times. However, the solution quality of the proposed method is not significant; the proposed model is very efficient in the real-world applications where the duration times of the tasks are probabilistic. The proposed method uses the simulation technique which is the best tool to model stochastic problems.

4. Conclusions

In this research, the scheduling of a multi-mode project was studied considering the resources restrictions, interruption feasibility, and the restarting of activities. The problem goal was to minimize the project makespan, while considering both prerequisite limitations and resource limitations. The main contribution of the research is the introduction of a novel modeling approach where the simulation model uses the optimum values of the relaxed problem. Such a simulation is a very smart move to produce several random feasible solutions in a very short time. Then, the best promising results of the simulation replications were used in a GA algorithm to make the algorithm stop faster than a generic GA. A novel approach in OvS was constituted through integration of the simulation technique and GA. The approach was tested on several RCPSPs and preemptions are allowed while most previous studies just considered non-preemptive RCPSPs. As the results show, the solution quality of the proposed algorithm is better while the computational speeds are the same. For future researches, we recommend to assume stochastic duration times for activities instead of deterministic values to make the problem closer to the real-world applications.

References

[1] Abdelaziz, F.B., Krichen, S., Dridi, S.,"A multi-objective resource-constrained project scheduling problem", K.Mellouli (Ed.): ECSQARU, LNAI 4724, 2007, pp. 719–730.

[2] Aboutalebi, R.S., Najafi, A.A., Ghorashi, B., "Solving multi-mode resource-constrained project scheduling problem using two multi objective evolutionary algorithms", African Journal of Business Management, Vol. 6, No. 11, 2012, pp. 4057-4065.

[3] Abbasi, B., Shadrokh, S., Arkat, J., "Bi-objective resource constrained project scheduling with robustness and makespan criteria", Appl. Math. Comput. Vol. 180, 2006, pp. 146-152.

[4] Afshar-Nadjafi B. and Majlesi M., "A solution procedure for preemptive multi-mode project scheduling problem with mode changeability to resumption", Applied Computing and Informatics ،sci. Iran. Vol. 20, No. 3, 2014, pp. 16-25.

[5] Al-Fawzan, M., Haouari, M., "A bi-objective model for robust resource constrained project scheduling", International Journal of Production Economics Vol. 96, 2005, pp. 175–187.

[6] Ballestin, F., Blanco, R., "Theoretical and practical fundamentals for multi-objective optimization in resource-constrained project scheduling problems", Computers & Operations Research, Vol. 38, 2011, pp. 51–62.

[7] Bianco, L., Dell'Olmo, P., Speranza, M.G., "Heuristics for multimode scheduling problems with dedicated resources", European Journal of Operational Research Vol. 107, 1998, pp. 260–271.

[8] Brucker, P., Drexl, A., Möhring, R., Neumann, K., Pesch, E., "Resource-constrained project scheduling: notation, classification, models, and methods",

European Journal of Operational Research Vol. 112, 1999, pp. 3–41.

[9] Brucker, P., Knust, S., "Resource-constrained project scheduling and timetabling", Lecture Notes in Computer Science Vol. 2079, 2001, pp. 277–293.

[10] Debels, D., Vanhoucke, M., "The impact of various activity assumptions on the lead time and resource utilization of resource-constrained projects",Computers and Industrial Engineering Vol. 54, 2008, pp. 140–154.

[11] Demeulemeester, E.L., Herroelen W.S., "An efficient optimal solution procedure for the preemptive resource-constrained project scheduling problem", European Journal of Operational Research Vol. 90, No. 2, 1996, pp. 334–348.

[12] Kazemi, F.S., and Tavakkoli-Moghaddam, R., "Solving a multi-objective multi-mode resource-constrained project scheduling problem with particle swarm optimization", International Journal of Academic Research, Vol. 3, No. 1, 2011, pp. 103-110.

[13] Kolisch, R., "Integrated scheduling, assembly area- and part-assignment for large-scale, make-to-order assemblies", International Journal of Production Economics Vol. 64, Nos. 1–3, 2000, pp. 127–141.

[14] Nabipoor-Afruzi, E., Roghanian, E., Najafi, A.A., Mazinani, M., "A multi-mode resource-constrained discrete time – cost tradeoff problem solving using an adjusted fuzzy dominance genetic algorithm", Scientia Iranica E, Vol. 20, No. 3, 2013, pp. 931-944.

[15] Nudtasomboon, N. , Randhawa, S. U., " Resource-constrained project scheduling with renewable and non-renewable resources and time-resource tradeoffs", Computers and Industrial Engineering Vol. 32, No. 1, 1997, pp. 227–242.

[16] Słowin´ski R.," Multi objective network scheduling with efficient use of renewable and nonrenewable resources", European Journal of Operational Research Vol. 7, 1981, pp. 265–273.

[17] Viana, J., de Sousa, P., "Using metaheuristics in multi objective resource constrained project scheduling", European Journal of Operational Research Vol. 120, No. 2, 2000, pp. 359–374.

[18] Weglarz, J.," On certain models of resource allocation problems", Cybernetics, Vol. 9, 1981, pp. 61–66 .

[19] YongyiShou, Ying Li, and Changtao Lai, "Hybrid particle swarm optimization for preemptive resource-constrained project scheduling",Neurocomputing, Vol. 148, 2015, pp. 122-128.

[20] Rasekh, A, Brumbelow, K, "A dynamic simulation–optimization model for adaptive management of urban water distribution system contamination threats", Applied Soft Computing,Vol. 32, 2015, pp. 59-71.

[21] Hong, L.J., Nelson, B.L., "A brief introduction to optimization via simulation", Proceedings of the 2009 Winter Simulation Conference (WSC), Austin, USA, DOI: 10.1109/WSC.2009.5429321, 75-85, (2009).

[22] Ghamginzadeh, A., Najafi, A.A., Azimi, P., "Solving a multi-objective resource-constrained project scheduling problem using Cuckoo optimization algorithm", Scientia Iranica E, Vol. 21, No. 6, 2014, pp. 2419-2428.

Follow This Article at The Following Site

Azimi P, Azouji N. An Optimization via Simulation approach for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problems. IJIEPR. 2017; 28 (4) :429-439

URL: http://ijiepr.iust.ac.ir/article-1-715-en.html