AUT Journal of Modeling and Simulation

AUT J. Model. Simul., 51(1) (2019) 15-32
DOI: 10.22060/miscj.2019.15033.5121

A Hybridized Metaheuristic Algorithm to Solve the Robust Resource Constrained Multi-Project Scheduling Problem

E. Nabipoor Afruzi[1], A. Aghaie[2*]

[1]PhD, Faculty of Industrial Engineering, K.N. Toosi University of Technology, Tehran, Iran
[2] Professor, Faculty of Industrial Engineering, K.N. Toosi University of Technology, Tehran, Iran

ABSTRACT: In this paper, the multi-project scheduling problem is studied. The duration of the activities is subjected to the considerable uncertainty and the robust optimization approach is considered to deal with the uncertainty. The maximum total tardiness of the projects is defined as the objective function which should be minimized. In order to allocate the constrained resources to the multi-projects, two models are proposed. In the first model, the projects are scheduled separately while in the second model, the multi-project approach is applied and the resource sharing policy is used. It is demonstrated that how the tardiness of the projects will be decreased when the multi-project approach is applied. Also, the Adaptive Bee Genetic Algorithm (ABGA) is designed as a hybrid metaheuristic algorithm and proposed in this paper to solve the first stage model of the Robust Resource Constrained Multi-Project Scheduling Problem (RRCMPSP). The results of ABGA is compared with the results of scenario-relaxation algorithm as an exact algorithm for the small size problems. Also, the performance of ABGA is studied compared to the Genetic Algorithm (GA) and Artificial Bee Colony (ABC) as two basic algorithms for the large size problems. The results show the effectiveness of the proposed algorithm in solving the RRCMPSP.

Review History:

Received: 25 September 2018
Revised: 30 March 2019
Accepted: 24 April 2019
Available Online: 1 June 2019

Keywords:

Resource Constrained Multi-Project Scheduling Problem
Robust Optimization
Maximum Total Tardiness
ABGA

1- Introduction & Literature Review

Resource-Constrained Project Scheduling Problem (*RCPSP*) is one of the most well-known problems studied over the past decade. Today, the single project management rarely occurs and the companies usually manage more than one project simultaneously titled “multi project management”. Lova and Tormos [1] found that among the 202 Spanish companies, 84% of them executed multiple projects in parallel. The literature of project management problem is heavily biased towards the single project environment while the studies related to the multi-project problem is little [2]. Aritua et al. [3] mentioned that managing of multi-project is not simply an aggregate of single project efforts. Zheng et al. [4] explained that the main reason of not much fruits on the topic of multi-project scheduling, comes from its high complexity, which is affected by many factors, such as the huge solution space, the intensely contending for resources, various and conflicting objectives, the inter-project dependence and priority, the high level of uncertainty and so on. Many researchers have studied recently the multi-project problem to overcome this identified gap [5-9]. In this paper, the multi-project problem is studied while the renewable resource constraint is considered during the project scheduling problem.

The decision about resources is one of the most important aspects of the multi project scheduling problem. The characteristics of resource usage by individual project in the multi project environment is described as resource management policy [10]. Among the different existing policy [5, 7, 8], in this paper, the most common one; resource sharing policy determines how to allocate the common resources among projects. In addition this policy is compared with the situation in which the projects are scheduled separately with their proprietary resources.

In contrary to many studies assuming the parameters of the model deterministic, in a rapidly changing environment, the project activities are subject to uncertainty. When the data takes values different from the nominal ones, several constraints may be violated, and the optimal solution found using the nominal data may be no longer optimal or even feasible [11]. In fact, due to employees’ absenteeism, delays in materials supply, bad weather conditions and many other uncontrollable factors, some project activities may last longer than expected, threatening the operational viability of the planned schedule [12]. The duration of the activities has a considerable uncertainty in this study.

There are fundamental approaches for scheduling projects under uncertainty; namely, reactive scheduling, stochastic scheduling, scheduling under fuzziness, proactive (robust) scheduling, and sensitivity analysis [13]. The nature and characteristic of the under study problem dictates which approach is appropriate to deal with the uncertainty.

Many studies apply stochastic approach to consider uncertainty in project scheduling problem [14-16]. Also, the fuzzy approach is applied in many studies like [17-20].

*Corresponding author’s email: aaghaie@kntu.ac.ir

The serious challenging point is that determining fuzzy membership function or fitting distribution function for the activities duration is not easy and accurate. Therefore, from a practical point of view, the application of these two approaches are seriously limited.

Robust optimization is totally compatible with the nature of project scheduling problem which is applied in this paper. The main advantages of robust optimization compared to stochastic programming or fuzzy approach are that no assumptions are needed for probability distribution or membership function of the uncertain data [22]. There are only four studies applying a robust optimization approach for the *RCPSP* with uncertain duration in the single project problem. Artigues et al. [21] proposed models for project scheduling when there is considerable uncertainty in the activity durations. They developed and implemented a scenario-relaxation algorithm and a scenario-relaxation-based heuristic. Bruni et al. [12] proposed an adaptive robust optimization model to derive the resource allocation decisions that minimizes the worst-case makespan, under general polyhedral uncertainty sets. Chakrabortty et al. [23] studied the *RCPSP* in which the activity durations are random variables with different probability distribution functions. They proposed robust optimization-based approach which produces good solutions under any input data scenario. Bruni et al. [24] in another study introduced two exact decomposition approaches to tackle the solution of the robust resource-constrained project scheduling problem under budgeted uncertainty polytope in which the resource constrained project scheduling problem is studied assuming that activity durations are subject to interval uncertainty.

In this paper, the robust optimization approach is applied for multi-project scheduling problem. Also, two situations are compared with each other. First, when there are some separate projects to be scheduled while they have constrained proprietary resources. Second, when the projects are in the shape of multi-project problem and they are not scheduled separately; in other words, the resource sharing policy links the projects to each other. It is shown how the multi-project approach can reduce the tardiness of the projects when there is a considerable uncertainty in the duration of the activities. Both of these situations are represented in a two-stage model in which, the objective function is to minimize the maximum total tardiness of the projects. In this study, the robust optimization of the multi-project scheduling problem is studied which has totally different characteristics for scheduling in comparison to the single project scheduling problem. Indeed, the calculation method related to the scheduling of the multi-projects are more sophisticated than the calculation for the single project. The projects use the resource pool under the Resource Sharing policy and all of the projects have a predefined due date. The objective is to minimize the maximum total tardiness of the all projects. The *ABGA* as a hybridized metaheuristic algorithm is designed and proposed for the first time in this paper to solve the large size problem of *RRCMPSP*.

The structure of the paper is as follows: Problem description is in Section 2. The proposed models for both the single project and the multi-project cases are presented in Section 3. The exact approach is explained in Section 4. Also in Section 4, the numerical example is presented to demonstrate the comparison between the single project and multi-project approaches. The *ABGA* is designed and explained in detail in Section 5. In Section 6, the computational result obtained by *ABGA* is reported and compared by scenario relaxation algorithm for the small size problems. Also, the comparison of *ABGA* results with *ABC* and *GA* are explained. Finally, Section 7 shows the conclusion and further research.

2- Problem Description

Consider resource constrained projects: $G = 1, 2, \dots, q$. In the first model, these projects are scheduled separately. In the second model, they are scheduled in the form of multi-project problem by resource sharing policy in which there are common resources in the resource pool. The structure of the projects is activity-on-node (*AON*) network $Graph = (V, E)$. Each project has *n* real activities. The precedence relationship between activities is shown by $E \subset V \times V$. In each project, the (dummy) activities of *0* and *n+1* represent the start and end of the project, respectively with zero durations and zero resource requirement. The possible values for duration of activity $i \in V$ of project *g* is shown by a set $P_{ig} \subset R_+$. Therefore, in the set of $P_{ig} = \{p_{ig1}, p_{ig2}, p_{ig3}, \dots, p_{ig|P_i|}\}$, the minimum and maximum durations for activity *i* of project *g* are $P_{ig}^{\min} \equiv \min_{P_{ig}} P_{igc}$ and $P_{ig}^{\max} \equiv \max_{P_{ig}} P_{igc}$ respectively. The set of renewable resources is shown by *R*. The required resource type *k* for execution of activity $i \in V$ of project *g* is $b_{igk} \in N$. If $F \subset V$ contains any subsets of activities which there is no precedence relationship between them and at least for one type of resource $\sum_{i \in F} b_{ik} > b_k$, then the set *F* is one "Forbidden Set" and its activities cannot execute in parallel because of resource conflict. The "Minimal Forbidden Set" is a forbidden set which its subset cannot be a forbidden set. Any resource conflicts can be removed by adding extra precedence relationships to the primary precedence graph for postponing some activities in such a way that the makespan can be determined by applying an early start policy (*ES*-Policy) on an extended graph. According to Balas [25] the set *X*, containing pairs of activities that leads to one feasible *ES*-policy, can be called a sufficient selection. After adding the extra precedence relationship *X* to the primary precedence graph *E*, the resource constraints can be ignored according to the precedence relationship in the *EUX* and the makespan can be obtained by calculating the critical path problem on the extended graph $Graph'(V, (E \cup X))$.

For demonstrating the resource flows between the activities, the transshipment networks are applied [21] which can be called as (resource) flow network. A flow $f(i, j, k) \equiv f_{ijk} \in \mathbb{N}$ represents the number of resource type *k* transferring from the end of activity *i* to the start of activity *j*.

For each project, a due date DD_g is determined by the project manager which is defined as a deadline for finishing each of the projects. The aim is to minimize the deviation of each project makespan from its due date. In the first situation, this minimization occurs for each project separately while the projects use their dedicated resources. In the second case, this minimization happens when the projects form a multi-project problem with the shared common resources. The question is how to allocate the resources between the activities in such a way that under the condition of uncertain activity durations, the maximum tardiness for all projects will be minimized.

3- Proposed Models

In this section, the proposed mathematical model for both approaches; the single project scheduling and multi-project scheduling is presented in the form of a bi - stage model.

3-1- Robust Resource Constrained Single Project Scheduling Problem

3-1-1- Notations

In this part, the variables, parameters and indices which are applied in the model are introduced.

- **Indices**

V The set of activity nodes

R The set of renewable resources

E The set of precedence relations between activities

P The set of scenarios belonging to activities duration

- **Parameters**

DD The due date of project

P_i^h The duration of activity i under scenario h

b_{ik} The required resource type k for performing activity i

b_k The capacity of resource type k

$P_i^{\min}$ The minimum scenario value for duration of activity i

$P_i^{\max}$ The maximum scenario value for duration of activity i

- **Variables**

Z The objective function of problem

Ta The tardiness of project

S_i^h The start time of activity i under scenario h

x_{ij} The decision variable with value one when activity i is the predecessor of activity j. otherwise it takes the value zero.

f_{ijk} The number of resource units of type k that are transferred from the end of activity i to the start of activity j.

a_i The decision variable with value one if the duration of activity i takes the maximum value and it takes the value zero if the duration of activity i takes the minimum value.

$\varphi_{ij}^{\max}$ $\varphi_{ij}^{\min}$ The minimum and maximum flow transferred from

3-1-2- The First Stage Model of *RRCPSP*

In the equations (1) - (15) the first stage model is presented for the single project scheduling approach.

$$Z = \min Ta \tag{1}$$

$$s.t.$$

$$Ta \geq S_{n+1}^h - DD \quad , h = 1,\ldots,|P| \tag{2}$$

$$S_j^h \geq S_i^h + P_i^h - M(1 - x_{ij}) \quad , \forall (\mathrm{i},\mathrm{j}) \in V \times V,\ i \neq j,\ h = 1,\ldots,|P| \tag{3}$$

$$x_{ij} + x_{ji} \leq 1 \quad , \forall (\mathrm{i},\mathrm{j}) \in V \times V,\ i < j \tag{4}$$

$$x_{iu} \geq x_{ij} + x_{ju} - 1 \quad , \forall (\mathrm{i},\mathrm{j},\mathrm{u}) \in V \times V \times V,\ i \neq j \neq u \tag{5}$$

$$\sum_{\substack{\mathrm{i} \in V, \\ i \neq 0}} f_{0ik} = b_k \quad , \forall k \in R \tag{6}$$

$$\sum_{\substack{j \in V, \\ j \neq n+1}} f_{jn+1k} = b_k \quad , \forall k \in R \tag{7}$$

$$\sum_{\substack{j \in V, j \neq i \\ j \neq n+1}} f_{jik} = b_{ik} \quad , \forall i \in V \setminus \{0, n+1\},\ \forall k \in R \tag{8}$$

$$\sum_{\substack{j \in V, j \neq 0 \\ j \neq i}} f_{ijk} = b_{ik} \quad , \forall i \in V \setminus \{0, n+1\},\ \forall k \in R \tag{9}$$

$$f_{ijk} \leq \min\{b_{ik}, b_{jk}\}.x_{ij} \quad , \forall (i,j) \in V \times V,\ \forall k \in R \quad , i,j \neq 0, n+1 \tag{10}$$

$$x_{ij} = 1 \quad , \forall (i,j) \in E \tag{11}$$

$$Ta \geq 0 \tag{12}$$

$$S_\mathrm{i}^h \geq 0 \quad , \forall i \in V,\ h = 1,\ldots,|P| \tag{13}$$

$$f_{ijk} \geq 0 \quad , \forall (i,j) \in V \times V,\ \forall k \in R \tag{14}$$

$$x_{ij} \in \{0,1\} \quad , \forall (i,j) \in V \times V \tag{15}$$

The objective function (1) displays the minimization of the project tardiness (Ta). The calculation of the project tardiness is in constraint (2) resulted by the difference of the project makespan (S_{n+1}) and the project due date (DD) for each scenario (h). The precedence relationship between the activities is presented in constraint (3) where M is a big number, P_i^h shows the duration of activity i based on scenario h, S_i^h is the start time of activity i according to scenario h and x_{ij} is the binary variable with value 1 if activity i *is the predecessor of activity j*. Eq. (4) prevents cycle in the graph. If activity i is the predecessor of activity j and activity j is the predecessor of activity u then activity i would be the predecessor of activity u. This rule is demonstrated in constraint (5). The sum of resource flows type k sending from activity *0* to other activities (f_{0ik}) is equal to the available capacity of resource type k (b_k). Also, the sum of resource flows type k sending from the activities to the dummy activity *n+1* (f_{jn+1k}) is equal to the available capacity of resource type k. These rules are shown in Eqs. (6) and (7) respectively. As represented in Eq. (8), the sum of incoming resource flows type k sending from other activities to activity i (f_{jik}) is equal to the required resource type k for activity i (b_{ik}). In the same way, Eq. (9) shows that the sum of output resource flows type k sending from activity i to other activities (f_{ijk}) is equal to the required resource type k for activity i. Constraint (10) ensures that the resource flow type k transferring from activity i to the activity j is utmost equal to the minimum value of $\{b_{ik}, b_{jk}\}$. In addition, this equation prevents resource transferring between two activities that there is no precedence relationship between them. Eq. (11) demonstrates that the binary variable x is equal to 1 for the two activities with precedence relationship between them. In constraint (12), it is mentioned that the project tardiness cannot be negative, so it ensures that $Ta = \max\{0\ ,\ S_{n+1}^{h} - DD\}$ is true for all scenarios. Nonnegative decision variables related to the start time of activities and the resource flow between activities are introduced in constraints (13) and (14) respectively. Finally, the binary variable x is described in Eq. (15).

The objective function of the first stage model is minimization of the project tardiness for the existing scenarios. In addition, the optimized structure of *EUX* is obtained with respect to the existing scenarios. In other words, the best structure is resulted for resource allocation with respect to the precedence relationships and resource requirements in such a way that the project tardiness become minimum. This structure is needed as an input data for the second stage model.

3-1-3- The Second Stage Model of *RRCPSP*

$$Z_2 = \max Ta \qquad (16)$$

$$s.t.$$

$$Ta \le [\sum_{(i,j)\in EUX} (p_i^{\min}.\varphi_{ij}^{\min} + p_i^{\max}.\varphi_{ij}^{\max})] - DD \qquad (17)$$

$$\sum_{(i,j)\in EUX} (\varphi_{ij}^{\min} + \varphi_{ij}^{\max}) = 1 \qquad , for\ i = 0 \qquad (18)$$

$$\sum_{(i,j)\in EUX} (\varphi_{ij}^{\min} + \varphi_{ij}^{\max}) = 1 \qquad j = n+1 \qquad (19)$$

$$\sum_{(i,j)\in EUX} \varphi_{ij}^{\min} + \varphi_{ij}^{\max} = \sum_{(j,i)\in EUX} \varphi_{ji}^{\min} + \varphi_{ji}^{\max} \qquad , \forall i \in V \setminus \{0, n+1\} \qquad (20)$$

$$\sum_{(i,j)\in EUX} \varphi_{ij}^{\max} \le a_i \qquad , \forall i \in V \setminus \{0, n+1\} \qquad (21)$$

$$\sum_{(i,j)\in EUX} \varphi_{ij}^{\min} \le 1 - a_i \qquad , \forall i \in V \setminus \{0, n+1\} \qquad (22)$$

$$Ta \ge 0 \qquad (23)$$

$$\varphi_{ij}^{\min} \ge 0 \qquad , \forall (i,j) \in EUX \qquad (24)$$

$$\varphi_{ij}^{\max} \ge 0 \qquad , \forall (i,j) \in EUX \qquad (25)$$

$$a_i \in \{0,1\} \qquad , \forall i \in V \qquad (26)$$

$$a_i = 0 \qquad , for\ i = 0, n+1 \qquad (27)$$

Maximization of the project tardiness is described in Eq. (16). In Eq. (17), the way of calculating the project tardiness is explained. In Eq. (17), for obtaining the finish time of project, the longest path method is applied. In fact, the structure of *EUX* is considered as an overall structure included the primary precedence relationships between activities (*E*) and the extra precedence relationships caused by resource constraint (*X*). In order to obtain the longest path, the formula $\sum_{(i,j)\in EUX} (p_i.\varphi_{ij})$ can be used where p_i is the duration of activity i and φ_{ij} is the transferring flow from activity i to activity j. Because of the multiplication of p_i and φ_{ij}, this formula becomes nonlinear. As mentioned before, the only values of $p_i^{\min}$ and $p_i^{\max}$ are needed for each activity, so, the binary variable a_i can be used and the nonlinear formula can be converted to $\sum_{(i,j)\in EUX} (p_i^{\min}.\varphi_{ij}^{\min} + p_i^{\max}.\varphi_{ij}^{\max})$ which is linear. For more information about obtaining the longest path and how to linearize the mentioned formula please refer to Artigues et al. [21]. Eq. (18) implies that the sum of flows sending from activity *0* to the overall structure should be equal to 1. Also Eq. (19) ensures that the sum of flows sending from overall structure to activity *n+1* should equal to 1. The flow conservation law is demonstrated in Eq. (20) in which for each kind of flows, the sum of flows entering to the activity *i* should be equal to the sum of flows exiting from the activity *i*. Constraints (21) and (22) are the constraints created for linearizing the longest path formula. The constraint (23) to constraint (25) represent the nonnegative variables of project tardiness and the flows related to the longest path respectively. The binary variables *a* is introduced in Eq. (26). Equation (27) shows that for activity *0* (with zero duration), the binary variable *a* takes the value 0.

3-2- Robust Resource Constrained Multi Project Scheduling Problem

3-2-1- The First Stage Model of *RRCMPSP*

$$\min TTa^* = \sum_{g=1}^{G} Ta_g \qquad (28)$$

$$s.t.$$

$$Ta_g \ge S_{n+1,g}^{h} - DD_g \qquad , \forall g \in G, h = 1,\dots,|P| \qquad (29)$$

$$S_{j,g'}^{h} \ge S_{i,g}^{h} + P_{i,g}^{h} - M(1 - x_{igjg'}) \qquad , \forall (i,j) \in V \times V,\ \forall g, g' \in G \times G\ ,\ i \ne j\ or\ g \ne g',\ h = 1,\dots,|P| \qquad (30)$$

$$\sum_{g'} \sum_{\substack{i\in V,\\ i\ne 0}} \sum_{g} f_{0gig'k} = b_k \qquad , \forall k \in R \qquad (31)$$

$$\sum_{g} \sum_{\substack{j\in V,\\ j\ne n+1}} \sum_{g'} f_{jgn+1g'k} = b_k \qquad , \forall k \in R \qquad (32)$$

$$\sum_{g'\in G} \sum_{\substack{j\in V, j\ne n+1\\ (j\ne i\ or\ g\ne g')}} f_{jg'igk} = b_{igk} \qquad , \forall i \in V \setminus \{0, n+1\},\ \forall k \in R,\ \forall g \in G \qquad (33)$$

$$\sum_{g'\in G} \sum_{\substack{j\in V, j\ne 0\\ (j\ne i\ or\ g\ne g')}} f_{igjg'k} = b_{igk} \qquad , \forall i \in V \setminus \{0, n+1\},\ \forall k \in R,\ \forall g \in G \qquad (34)$$

$$f_{igjg'k} \le \min\{b_{igk}, b_{jg'k}\}.\ x_{igjg'} \qquad , \forall (i,j) \in V \times V,\ \forall (g,g') \in G \times G,\ \forall k \in R,\ i, j \ne 0, n+ \qquad (35)$$

$$x_{igjg'} = 1 \qquad , \forall (i,g,j,g') \in E \qquad (36)$$

$$S_{0g} = 0 \qquad , \forall g \in G \qquad (37)$$

$$Ta_g \geq 0 \qquad ,\forall g \in G \tag{38}$$

$$S_{i,g}^h \geq 0 \qquad ,\forall i \in V,\ \forall g \in G,\ h = 1,...,|P| \tag{39}$$

$$f_{igjg'k} \geq 0 \qquad ,\forall (i,j) \in V \times V,\ \forall (g,g') \in G \times G,\ \forall k \in R \tag{40}$$

$$x_{igjg'} \in \{0,1\} \qquad ,\forall (i,j) \in V \times V\ ,\forall g,g' \in G \times G \tag{41}$$

In the equations (28) - (41) the first stage model is presented for the multi-project scheduling approach.

In fact, the above equations are transformed from the single project scheduling to the multi-project scheduling approach. As explained in part 2, G is the set of projects and g, g' and g'' are the index of projects. The notation of g and g' are imported to all variables and parameters in order to show the projects. For example, $x_{igjg'}$ is the binary variable with value 1 if activity i of project g is the predecessor of activity j of project g'.

3-2-2 The Second Stage Model of *RRCMPSP*

The second stage model is presented for the multi-project case in equations (42) - (59).

$$\max TTa^* = \sum_{g=1}^{G} Ta_g \tag{42}$$

$s.t.$

$$Ta_g \leq (LP_g - DD_g) \qquad ,\forall g \in G \tag{43}$$

$$LP_{g''} = \sum_{(i,g,j,g') \in EUX} (p_{ig}^{\min} . \varphi_{g''igjg'}^{\min} + p_{ig}^{\max} . \varphi_{g''igjg'}^{\max}) \qquad ,\forall g'' \in G \tag{44}$$

$$\sum_{(i,g,j,g') \in EUX} (\varphi_{g''igjg'}^{\min} + \varphi_{g''igjg'}^{\max}) = 1 \qquad , for\ i = 0\ ,\ \forall g'' \in G \tag{45}$$

$$\sum_{(i,g,j,g') \in EUX} (\varphi_{g''igjg'}^{\min} + \varphi_{g''igjg'}^{\max}) = 1 \qquad ,\forall g'' \in G,\ j = n+1\ ,g' = g'' \tag{46}$$

$$\varphi_{g''igjg'}^{\min} = 0 \qquad ,\forall i \in V\ ,\forall g, g'' \in G \times G,\ j = n+1\ ,g' \neq g'' \tag{47}$$

$$\varphi_{g''igjg'}^{\max} = 0 \qquad ,\forall i \in V\ ,\forall g, g'' \in G \times G,\ j = n+1\ ,g' \neq g'' \tag{48}$$

$$\sum_{(i,g,j,g') \in EUX} \varphi_{g''igjg'}^{\min} + \varphi_{g''igjg'}^{\max} = \sum_{(j,g',i,g) \in EUX} \varphi_{g''jg'ig}^{\min} + \varphi_{g''jg'ig}^{\max} \qquad ,\forall i \in V \setminus \{0, n+1\},\ \forall g \in G,\ \forall g'' \in G \tag{49}$$

$$\sum_{(i,g,j,g') \in EUX} \varphi_{g''igjg'}^{\max} \leq a_{ig} \qquad ,\forall i \in V \setminus \{0, n+1\},\ \forall g \in G\ ,\forall g'' \in G \tag{50}$$

$$\sum_{(i,g,j,g') \in EUX} \varphi_{g''igjg'}^{\min} \leq 1 - a_{ig} \qquad ,\forall i \in V \setminus \{0, n+1\},\ \forall g \in G,\ \forall g'' \in G \tag{51}$$

$$Ta_g \geq 0 \qquad ,\forall g \in G \tag{52}$$

$$\varphi_{g''igjg'}^{\min} \geq 0 \qquad ,\forall (i,g,j,g') \in EUX,\ \forall g'' \in G \tag{53}$$

$$\varphi_{g''igjg'}^{\max} \geq 0 \qquad ,\forall (i,g,j,g') \in EUX,\ \forall g'' \in G \tag{54}$$

$$x_{igjg'} = 1 \qquad ,\forall (i,g,j,g') \in E \tag{55}$$

$$x_{igjg'} \in \{0,1\} \qquad ,\forall (i,j) \in V \times V\ ,\forall g,g' \in G \times G \tag{56}$$

$$a_{ig} \in \{0,1\} \qquad ,\forall i \in V,\ \forall g \in G \tag{57}$$

$$S_{0g} = 0 \qquad ,\forall g \in G \tag{58}$$

$$a_{0g} = a_{n+1g} = 0 \qquad ,\forall g \in G \tag{59}$$

In the above formulation, $\varphi_{g''igjg'}$ is the flow type g'' transferred from activity i of project g to activity j of project g' in order to calculate the longest path of project g''. The longest path of each project in the multi project problem, is presented in Eq. (44). As mentioned before, the resource sharing policy is applied in this paper which links the projects by extra precedence relationships. So, the projects are related to each other in the *EUX* structure. In the structure of multi-project problem studied in this paper, for calculating the longest path, we should send a flow per project from the 0 activities to other activities in the overall structure. It means, the number of flows calculating the longest paths in the *EUX* structure is equal to the number of projects in the multi-project problem. The first index in the flow variable $\varphi_{g''igjg'}$, g'' shows for which project we want to calculate the longest path.

4- The Scenario Relaxation Algorithm

In this approach, the scenarios are gradually added to the problem structure in the sequential iterations. First, one scenario for activity durations is considered and the model of the first stage is solved. The aim is to obtain the structure $E \cup X$ for which the difference between the makespan of projects and their due dates is minimized. In other words, the search is for optimized $E \cup X$ according to the existing scenario, for which the total tardiness is minimized. In the next step, the second stage model results the worst scenario for the obtained structure of first stage in such a way that the objective (total tardiness of projects) will be maximized. Then the mentioned scenario should be added to the scenario set of first stage model and this algorithm continues until the objective functions of both stages become equal. The steps of the applied approach are as follows (*iter* is the counter of the algorithm iterations).

1) The set $\hat{P}_1$ containing only one scenario p^1 for the duration of all activities of the projects is considered. Also $iter = 1$, $LB = 0$ and $UB = +\infty$ are assumed.

2) **The first stage model is solved in order to obtain** $LB = TTa^*(\hat{P}_{iter})$. Also, the corresponding *ES-policy;* X_{iter} is resulted.

3) **The second stage model is solved and the maximum** $TTa^{\max}(X_{iter})$ for the X_{iter} is obtained. The corresponding worst scenario; p^{iter+1} is resulted. In addition, the $UB = TTa^{\max}(X_{iter})$ is considered.

4) **When** $LB = UB$ then stop the algorithm. If $LB \neq UB$, then $iter = iter + 1$ and $\hat{P}_{iter} = \hat{P}_{iter-1} \cup \{p^{iter}\}$ is considered and the algorithm should continue from step 2.

For more information about scenario relaxation algorithm, how to implement it in the single project problem and how to implement it in the multi-project problem, please refer to the references [26], [21] and [33] respectively.

4-1- Numerical Example of scenario relaxation algorithm: Scheduling of *RRCPSP*- Single Project

Consider 3 projects with six activities (the start activities and end activities are dummy) as shown in Fig. 1.

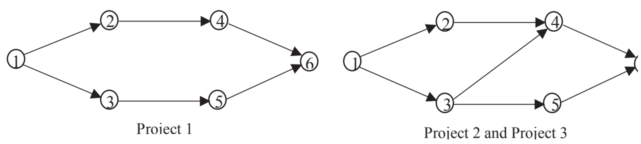

Fig.1. The network structure of three projects

There is only one renewable resource ($b_1 = 7$). The required resource for performing each activity, the possible durations of activities, the determined due date of projects are all represented in Table 1.

Table 1. The required data of three projects

Project 1 ($DD_1 = 7$)			Project 2 ($DD_2 = 9$)			Project 3 ($DD_3 = 7$)		
i	b_{i1}	p_i	i	b_{i1}	p_i	i	b_{i1}	p_i
1	0	{0}	1	0	{0}	1	0	{0}
2	5	{3,4,5}	2	3	{6,7,8}	2	4	{3,5}
3	4	{4,6}	3	2	{2,3}	3	5	{5,6,7,8}
4	3	{7,8,10}	4	4	{5,7}	4	2	{2,3,4}
5	4	{3,4}	5	3	{3,4}	5	3	{4,5}
6	0	{0}	6	0	{0}	6	0	{0}

Project 1) In the single project manner, these projects are scheduled separately. After the execution of the first stage and the second stage models, the first project has terminated after 2 iterations while the optimized objective function is 12. It means that if the project 1 execute in such a way that the resource transfers based on the optimized flows (Fig. 2), then

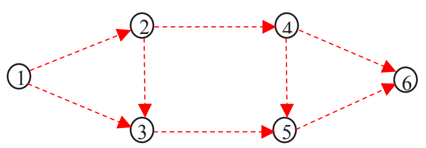

Fig.2. The optimized resource flow of project 1

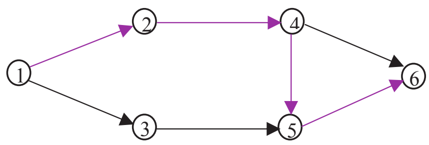

Fig.3. The longest path of project 1

the tardiness of this project will be minimum.

In addition, in the obtained optimized structure (*EUX), the maximum tardiness will be 12 based on the worst scenario happening. The critical path of project 1 is demonstrated in Fig. 3.*

Project 2) For the second project, after 2 iterations of implementing the first stage and the second stage models, the algorithm is terminated. The optimized objective function is 13 and the final resource flows is depicted in Fig. 4. Also, the critical path of project 2 is presented in Fig. 5.

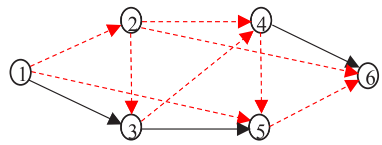

Fig.4. The optimized resource flow of project 2

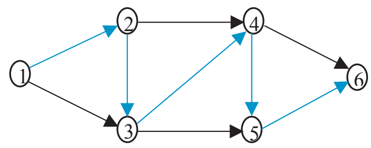

Fig.5. The longest path of project 2

Project 3) In the same way, after 2 iterations, the algorithm terminates by getting the optimal value of the objective function; 15. The final resource flow and the critical path of the third project are shown in Fig. 6 and Fig. 7 respectively.

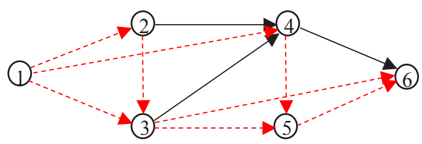

Fig.6. The optimized resource flow of project 3

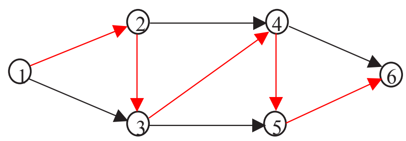

Fig.7. The longest path of project 3

4-2- Scheduling of RRCMPSP- Multi Project

In this part, the three projects described in part 4.1.1 are scheduled as a multi-project problem based on the resource

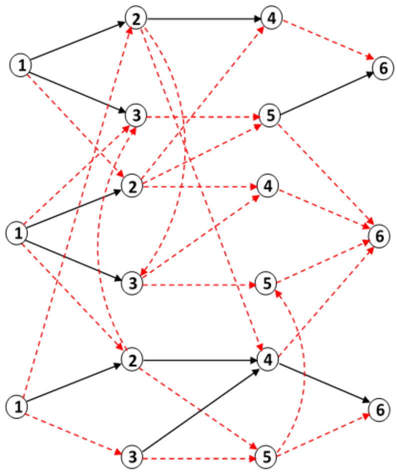

Fig. 8. The optimized resource flow for the multi-project problem

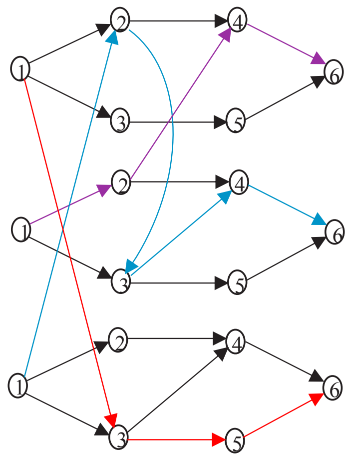

Fig. 9. The longest path of the projects in multi-project structure

sharing policy. In the multi-project problem, the projects can share their common resources from the resource pool. So, the *EUX* structure would be obtained as a whole for all of the projects in the multi-project problem. In this example, after 8 iterations, the algorithm is terminated. The summation of the projects tardiness is 22 as a total tardiness of the multi-project problem. The resource flows for these projects and the critical paths of each project are depicted in Fig. 8 and Fig. 9, respectively.

This simple numerical example shows the advantages of the multi-project scheduling approach in comparison to the scheduling of the projects separately. It is observed that under the uncertainty, when the multi-project approach is applied, the total tardiness of the projects is reduced significantly: (12+13+15)-22= 18. So, the resource sharing policy applied in the multi-project scheduling approach can influence greatly on reducing the finish time of the projects which is an important achievement in project scheduling problems.

5 -Hybridizing Scenario Relaxation Algorithm with *ABGA*

In the first stage model ,the *RCMPSP* is solved under different scenario. The *RCPSP* as a generalization of the classical job shop scheduling problem belongs to the class of NP-hard optimization problems [28]. The *RCMPSP* as a generalization of the *RCPSP* is also *NP*-hard [29]. So, as shown in Table 2, the first stage model has the main role in consuming run time. Therefore, it is necessary to develop metaheuristic algorithms in order to obtain results for the large size problems.

To solve large size problems, we hybridize the scenario relaxation algorithm (as an exact algorithm) with *ABGA*. In this case, the first stage model should be solved by *ABGA* and the stop criteria is $UB - LB \prec \varepsilon$ where ε is a positive number, small enough in comparison to the values of LB and UB.

The *ABGA* itself is a hybrid metaheuristic algorithm designed for the first time in this paper which benefits from the characteristics of *GA* (Genetic Algorithm) and *ABC* (Artificial Bees Colony) algorithms. The performance of this algorithm is compared with its basic algorithms; *GA* [30] and *ABC* [31].

Before the step by step description of the proposed algorithm, the encoding and decoding of a solution should be devised. Also, the explanation of the schedule generation algorithm which assigns a figure of merit to each encoded solution as a fitness function is required.

5-1- Encoding and Decoding of a Solution

Each solution can be shown as a $1 \times N$ matrix where N is the total number of activities belonging to all projects. The order of activities in this matrix demonstrates the priority of selecting in the scheduling process.

2	6	3	1	4	7	5

Fig. 10. The Structure of the Solution Matrix

Fig. 10 shows the encoding of a solution belonging to the multi-project with total 7 activities. In this example, activity 2 is the first activity in the list of activities order. In other words, activity 2 is the first activity which should be checked in the schedule generation algorithm.

5-2- Schedule Generation Algorithm

Each solution is evaluated by the value of its fitness based on the objective function presented in Section 3-2. Before the explanation of the schedule generation algorithm, it seems necessary to introduce the notations used in this algorithm.

5-2-1- Notation of the Schedule Generation Algorithm

PAS Parallel activities, i.e., activities that are not the predecessor or successor of each other either directly or indirectly

E'	The set containing any pair of activities which have precedence relationship with each other directly or indirectly (including transitive relations)
$n_{activity}$	The total number of activities
SM	Solution matrix
US	The set of activities which have received their required resources of any type, but they have not been scheduled yet
NSS	The set containing the activities that their predecessors are not in *US* and so, these activities cannot be chosen as j^*
CT_k	The set containing the activities that completely transferred their resource type *k* to other activities and do not have any resource type *k*
CO_k	The set containing the activities that completely obtained their required resource type *k*
n_{US}	Number of activities in *US*
n_{CO_k}	Number of activities in CO_k
f'_{iqk}	The sum of resources type *k* which have been transferred from other activities to activity *i* of project *q*, shown by the formula $f'_{iqk} = \sum_{j}\sum_{q'} f_{jq'iqk}$
f''_{iqk}	The sum of resources type *k* which have been transferred from activity *i* of project *q* to other activities, shown by the formula: $f''_{iqk} = \sum_{j}\sum_{q'} f_{iqjq'k}$
nscenario	The number of scenarios for the duration of activities
m	The counter of activities
h	The index of scenario for the activities duration
SA	Scheduled activities
RS	The activities which are ready to schedule (based on the precedence relationship) but they have not been scheduled yet
Predecessor(j)	The predecessor of activity j
$P(j^*)$	Duration of activity j^*
$n\ pre(j^*)$	Number of the predecessor of activity j^*
np_{j^*}	The counter for the predecessor of activity j^*
Finish(i)	Finish time of activity i

5-2-2- The flowchart of the Schedule Generation Algorithm

The schedule generation algorithm has two phases. In the first phase, the resources are transferred from activity 1 through the network. Each activity obtains its required resources and after performance, it releases them to other activities. By the end of this phase, all the activities will have obtained their required resources and will have joined US (Box 36). In the second phase, the activities are scheduled (Boxes 37 to 56). The flowchart of this algorithm is depicted in Fig. 11 schematically.

5-2-3- Explanation of the Schedule Generation Flowchart Phase 1 (Resource Transfer)

First, we choose one random generated chromosome as a solution matrix (Box 4). The first activity (activity one) has a role of project starter. This activity should be added to the sets US and CO_k for any kind of resources (Box 7). It transfers all kind of resources to the network. These resources are gathered finally by the end activity as the sink of the flow network. *k=1* (any resource type can be chosen) is set to start resource transferring process (Box 8). The first activity in the solution matrix (from the beginning of the chromosome) that is in *US* and CO_k sets but does not exist in *CTk* should be selected as u^* (Box 9). The u^* is supposed to transfer resources to other activity called j^*. In order to choose j^*, we select the first activity from the beginning of the chromosome that does not exist in *COk* and *NSS* (Box 10). In the resource transferring process (Box 24), the quantity of transferred resource; $f_{u^*qj^*q'k}$, is the minimum value of what remains for u^* to transfer (the amount of resource that u^* had at first minus the amount that it has transferred up to now) and what j^* needs at the moment (the amount j^* needed at first minus what it has received up to current time). This process continues until all of the activities join *US*.

Phase 2 (Activities Scheduling)

The algorithm continues from Box 37 where the number of scenarios is read. The activity 1 is scheduled at the beginning of the scheduling process. So, we add it to *SA* (Box 39). One of the activities which have not been scheduled yet but their predecessors were scheduled (*RS*) should be chosen randomly (Box 41). The start time of this activity is the maximum value

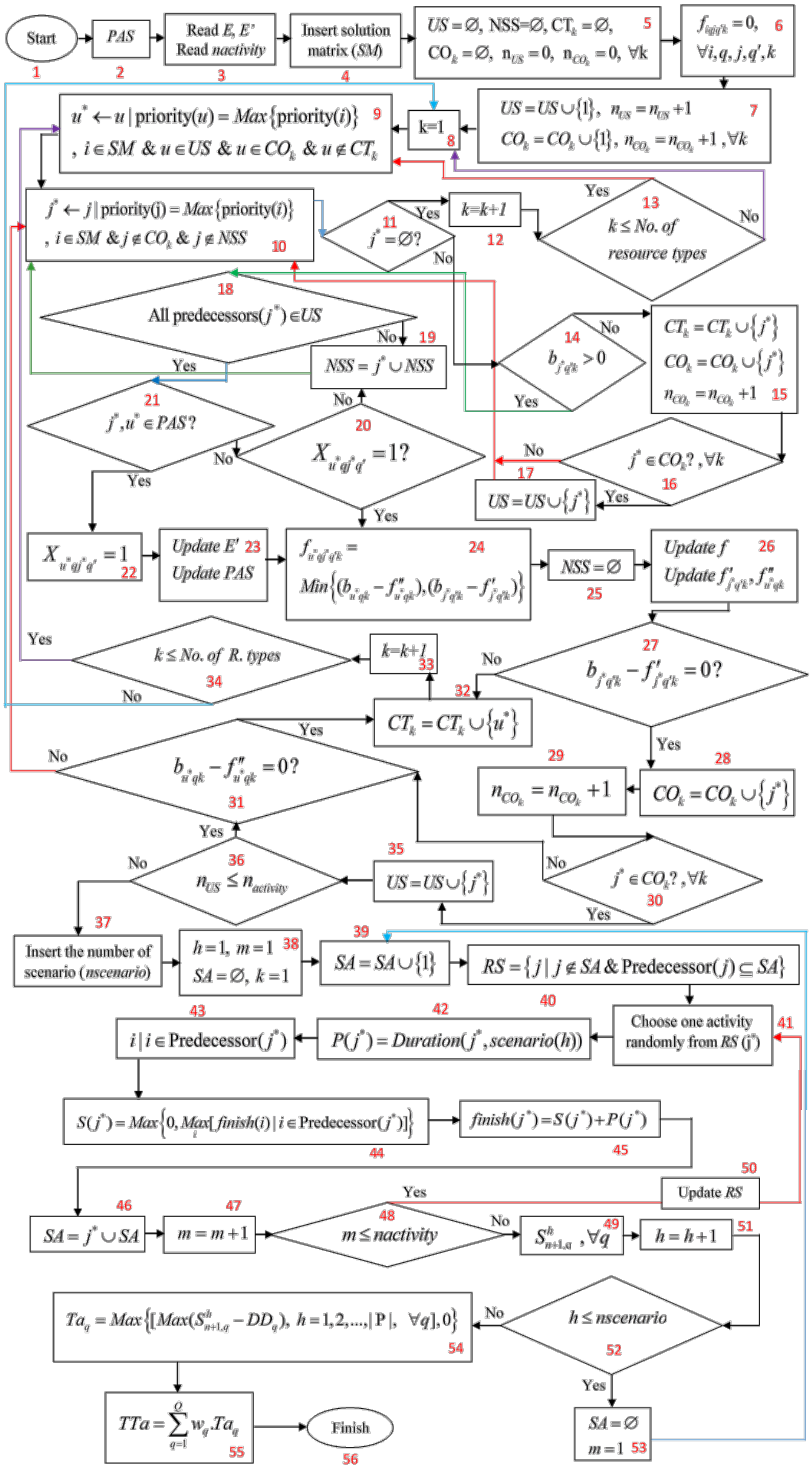

Fig. 11. Schedule Generation Schema for RRCMPSP

among the finish times of its predecessors (Box 44).

By adding the duration of this activity to its start time, the finish time can be obtained (Box 45). This process should repeat until all of the activities join the *SA* (Box 48). Then, the finish time of each project (*q*) can be easily obtained by the value of $S_{n+1,q}^{h}$ (Box 49). When the finish times of all projects are calculated for all scenarios, the tardiness of each project will be obtained by the formula in Box 54. Box 55 shows the total tardiness of projects as the problem objective function and the algorithm ends at this step.

5-3- ABGA

An approach called *ABGA* is presented to solve combinatorial optimization problems. This approach incorporates the artificial bee's colony into the genetic algorithm to improve the performance of the genetic algorithm. This approach can be viewed as a genetic algorithm with *ABC* operator as a neighboring generation and selection operators. This hybridization helps the *GA* to search solutions around the better members and so, it increases the diversity of the solutions and prevents the *GA* from getting stuck in the local optimum point.

5-3-1- Initialization

The proposed algorithm starts with introducing the parameters (Box 2) as depicted in Fig. 14. In this step, the initial number of sites (*No. site*), the initial number of recruit bees (*No. rec. bees*) and the number of population (*npop*) should be set. These values are determined as *No. site*=1, *No. rec. bees*=30 and $npop_0$=100 for the first generation.

The solutions are produced randomly with size *npop0* as pop1. The fitness function is then evaluated in order to check whether the algorithm should be terminated or not (Boxes 4 and 5). If the stop criterion is not satisfied, the crossover and mutation probabilities should be tuned (Box 6) as explained in part 5-3-2.

5-3-2- Adaptive Parameter Tuning of *Pc* and *Pm*

The rate at which solutions are subjected to crossover is shown by crossover probability *Pc*. The higher the value of *Pc*, the quicker the new solutions introduced into the population. Also, as a secondary operator, large values of *Pm*, transform the *GA* into a purely random search algorithm, while some mutation is required to prevent the premature convergence of the *GA* to suboptimal solutions [34].

In this study, the adaptive parameter tuning method presented in Srinivas and Patnaik [34] is applied which calculates the values of *Pc* and *Pm* by the expressions (60) and (61) respectively.

$$p_c = k_1(f_{\max} - f')/(f_{\max} - \bar{f}), \qquad k_1 \le 1 \tag{60}$$

$$p_m = k_2(f_{\max} - f)/(f_{\max} - \bar{f}), \qquad k_2 \le 1 \tag{61}$$

In order to preserve 'good' solutions, lower values of P_c and P_m should be set for high fitness solutions and higher values of P_c and P_m for low fitness solutions. While the high fitness solutions aid in the convergence of the *GA*, the low fitness solutions prevent the *GA* from getting stuck at a local optimum

The value of P_m should depend not only on $f_{\max} - \bar{f}$, but also on the fitness value *f* of the solution. Similarly, *Pc* should depend on the fitness values of both the parent solutions. The closer *f* is to *fmax*, the smaller *Pm* should be, *i.e*., *Pm* should vary directly as $f_{\max} - f$. Similarly, *Pc* should vary directly as $f_{\max} - f'$, where f' is the larger of the fitness values of the solutions to be crossed.

5-3-3- Crossover and Mutation Operations

The crossover operation is done with probability *Pc* calculated by the formula (60). A roulette wheel selection is applied to choose a number of solutions from the population for the crossover operation. Then, the one point, the two-point, and the uniform operators are executed as different crossover operators (Fig. 12) and create population 2 (pop2).

In addition, a number of solutions are selected randomly from the population for the mutation operation. The probability of mutation operation is *Pm* calculated by the formula (61). The swap, insertion and reversion operators (Fig. 13) are performed as different mutation operators and form population 3 (pop3).

5-3-4- Elitism Selection of *GA*

At this stage, the Boxes 7 and 8 of Fig. 14 are accomplished and this algorithm continues from Box 9 where the pop1, pop2, and pop3 are merged together. After fitness function evaluation (Box 10), the elitism selection is done based on the population size (*npop*) (Box 11). As described in part 5-3-1, the population size was set on 100 at the first stage; *npop0*=100. In this algorithm, the population size is tuned in each iteration adaptively which will be explained in part 5-3-8. So, in Box 11, the phrase "considering by *npopt*" means that if *npopt*>*npop1*+ *npop2*+ *npop3*, then, some random solutions should be produced and added to the population until the number of the population receives to *npopt*.

5-3-5- *ABC* Operators

Our approach creates a number of different search paths by changing the location of parallel activities (the activities with no precedence relationship) in the best chromosomes of the population trying to find better solutions simultaneously. Each of these new chromosomes is known as a neighbor.

In each iteration, the best solutions of *GA* are selected with size *No. site* and considered as the sites of *ABC* algorithm. Then, for each site, the neighbors (recruit bees) are produced with size *No. Rec. Bee* (Boxes 12 to 14). As mentioned in Box 12, the *No. site* and the *No. Rec. Bee* are tuned adaptively, describing in part 5-3-6.

5-3-6- Adaptive Parameter Tuning of *No. site* and *No. rec. Bees*

The formulation (62) is proposed to tune the parameters of *No. site* and *No. rec. Bees* adaptively.

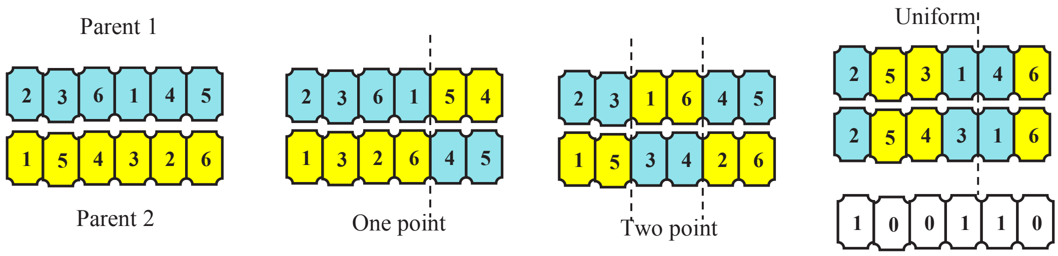

Fig. 12. Crossover Operators

$$X'_{iter} = \frac{Var_{iter} - Var_{iter-1}}{Var_{\max} - Var_{\min}} \tag{62}$$

Where Var_{iter} and Var_{iter-1} are population variance in the current and previous iterations, respectively. $Var_{\max}$ and $Var_{\min}$ are the maximum and minimum variance of the population. The formula (63) is presented in which the number of sites (*No. site*) can be determined based on the variance of the population.

$$No.\ site_{iter} = Max\{round[(1-X'_{iter}) \\ .No.\ site_{iter-1} + No.\ site_{iter-1}],1\} \tag{63}$$

The same way, formula (64) is represented for adaptive tuning of the *No. rec. bees* parameter.

$$No.\ rec\ bees_{iter} = Max\{round \\ [(1-X'_{iter}).No.\ rec\ bees_{iter-1} \\ +No.\ rec\ bees_{iter-1}],1\} \tag{64}$$

So, we check the diversity (variance) of the population in each iteration. If the variance decreases, the factor X'_{iter} will be negative and this increases the number of *No. site* and *No. rec. Bees* and vice versa. In this way, the *ABC* operator prevents the algorithm to get stuck in the local optimum point.

5-3-7- Selection the best solution in each site of *ABC*

The sites and the recruit bees should all be evaluated according to their fitness values (Box 15). Then the best of bees for each site should be selected (Box 16). If the best recruit bee is better than its site (Box 17), the site should be replaced with the fittest bee (Box 19). Otherwise, the site should be kept and the bee should be eliminated (Box 18).

5-3-8- Adaptive Parameter Tuning of $npop_t$

At this stage, one iteration of the whole *ABGA* has been accomplished and so, the population size can be adjusted for the next generation according to the population in the current generation (Box 20).

The method proposed by Eiben et al. [35] for adjustment of the population size (*npop*) is applied in this research in order to adaptive parameter tuning of *npop* based on the formula (65).

$$X = increase\ Factor \\ .(\max\ Eval\ Num - curr\ Eval\ Num). \\ \frac{\max Fitness_{new} - \max Fitness_{old}}{init\ Max\ Fitness} \tag{65}$$

Where *increase Factor* is an external parameter from the interval (0,1). *max Eval Num* and *Curr Eval Num* denote the given maximum number of fitness evaluations and the current evaluation number. *max Fitnessnew*, *max Fitnessold*, and *init Max Fitness* are the best fitness values in the current generation, the best fitness values in the preceding generation and the best fitness value in the initial population.

If the improvement of the algorithm occurs in less than or equal to *m* iterations, then the size of the population will increase by the factor *X* introduced in formula (65). If no improvement occurs for *k* iterations, then again the population increases by *X* factor. If the improvement happens for more than *m* iterations, the population should decrease by a little percentage; e.g. 1-5%. For more information about this method, please refer to [35].

Based on the above explanations, the proposed *ABGA* approach consists of a two-stage cycle. The first stage relates to the search by the *GA* and the second stage is the evolution by the *ABC* operators.

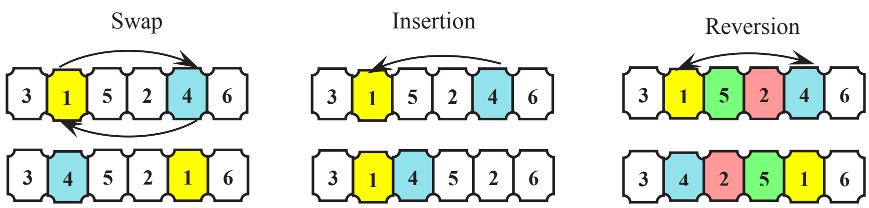

Fig. 13. Mutation Operators

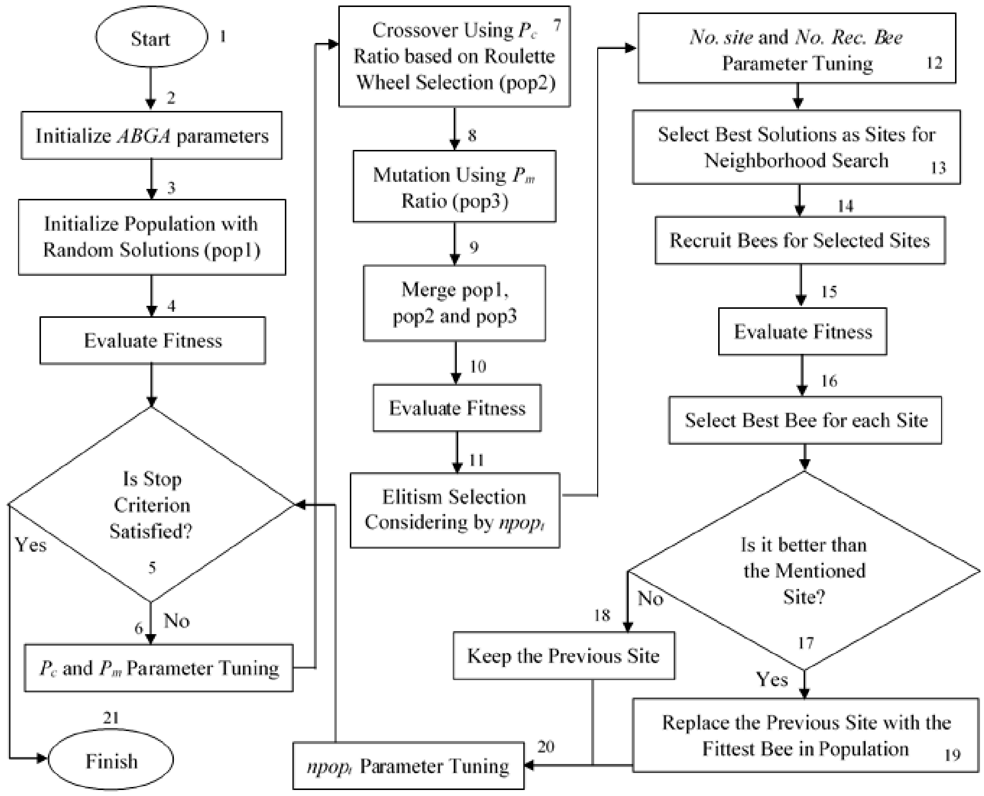

Fig. 14. ABGA Algorithm

6- The Computational Experiments of Scenario Relaxation/ *ABGA* Algorithm

The algorithm *ABGA is proposed to solve the first stage model of RRCMPSP. All the experiments are* coded in MATLAB 2017b and run on a personal computer with an Intel Core i7-4702MQ CPU @ 2.20 GHz and 8 GB RAM under Windows 10 operating system. Three groups of test problems; *i.e. n=10, 20 and 30 are solved by both the proposed metaheuristic algorithm and scenario relaxation algorithm in order to validate the ABGA for the small size problems.* In order to implement the scenario relaxation algorithm, the proposed models are coded in GAMS v24.1.2 and solved with the CPLEX solver. The experiments are run on a personal computer with an Intel(R) Core(TM) i5-6400 CPU @ 2.70 GHz 2.70 GHz and 16 GB RAM under Windows 10 operation system. Also, for the large size problems, the *ABGA is implemented for n=40 and compared with its basic algorithms; GA and ABC in order to demonstrate*

Table 2. Global Optimum Values and the Relative Gaps for ABGA (n=10)

Problem No.	Parameters			Global Optimum	Relative Gap
	OS	RF	RC		
A1	0.4	0.25	0.3	7.4	$1.3*10^{-2}$
A2	0.4	0.25	0.6	8.1	0
A3	0.4	0.5	0.3	10	$1.8*10^{-2}$
A4	0.4	0.5	0.6	12.3	0
A5	0.4	0.75	0.3	11	0
A6	0.4	0.75	0.6	6.5	0
A7	0.7	0.25	0.3	7.7	$2*10^{-3}$
A8	0.7	0.25	0.6	9.1	0
A9	0.7	0.5	0.3	6	0
A10	0.7	0.5	0.6	10.6	$1.9*10^{-2}$
A11	0.7	0.75	0.3	8.2	0
A12	0.7	0.75	0.6	3.7	$1.8*10^{-2}$

its effectiveness. For this purpose, each experiment is run 10 times. Then, the average value of objective function is calculated and reported. Worth to mention that there are three projects in each multi project problem where *n is the number of activities in each project.*

All the test problems are generated by the software RanGen [32] with different structural characteristics; *i.e.* order strength (*OS*), resource factor (*RF*) and resource constrainedness (*RC*) which effect on the complexity of the test problems. For more information about these factor please refer to [27]. Additional data are considered for adapting the test problems to *RRCMPSP*.

6-1- Results for small size *RRCMPSP* (*n*=10, 20 and 30)

First, the samples of size *n*=10 are considered and solved by *ABGA*. Table 2 shows the exact results of the scenario relaxation algorithm and the relative gaps of the results obtained by *ABGA*. Then the same results for *n*=20 and 30 are calculated and presented in Tables 3 and 4 respectively.

As observed in Tables 2 to 4, the *ABGA* presents good solutions with negligible relative gaps in comparison to the global solutions. In order to demonstrate the performance of the proposed algorithm in comparison to the other algorithms, some samples with different sizes are chosen randomly in order to depict their results.

As shown in Fig. 15, the proposed algorithm; *ABGA* performs better than *GA* and *ABC* in finding a better solution for the small size test problems.

6-2- Results for large size *RRCMPSP* (*n*=40)

Table 5 shows the means and standard deviations obtained for the cost of the test problems with *n*=40. Also, the means and standard deviations of the *NFE*s are reported. In addition, based on the results of the algorithm, the rank of algorithms is presented.

As shown in Table 5, the *ABGA* performs better in comparison to the other algorithms. The convergence characteristics graphs for some randomly selected problems with *n*=40 are illustrated in Fig. 16.

As shown in Fig. 16, the proposed algorithm; *ABGA*

Table 3. Global Optimum Values and the Relative Gaps for ABGA (n=20)

Problem No.	Parameters			Global Optimum	Relative Gap
	OS	RF	RC		
B1	0.4	0.25	0.3	11.3	$1.8*10^{-2}$
B2	0.4	0.25	0.6	4.5	$3*10^{-1}$
B3	0.4	0.5	0.3	7.7	$4*10^{-2}$
B4	0.4	0.5	0.6	10	$2.1*10^{-3}$
B5	0.4	0.75	0.3	15.5	0
B6	0.4	0.75	0.6	16	$2*10^{-2}$
B7	0.7	0.25	0.3	14.3	0
B8	0.7	0.25	0.6	14.5	$1.5*10^{-1}$
B9	0.7	0.5	0.3	8	$3*10^{-2}$
B10	0.7	0.5	0.6	17.2	$2*10^{-2}$
B11	0.7	0.75	0.3	11.1	0
B12	0.7	0.75	0.6	16.8	$3.1*10^{-2}$

Table 4. Global Optimum Values and the Relative Gaps for ABGA (n=30)

Problem No.	Parameters			Global Optimum	Relative Gap
	OS	RF	RC		
C1	0.4	0.25	0.3	12.4	$3.4*10^{-2}$
C2	0.4	0.25	0.6	7.6	$1.5*10^{-2}$
C3	0.4	0.5	0.3	11	$2*10^{-2}$
C4	0.4	0.5	0.6	15.2	$8*10^{-2}$
C5	0.4	0.75	0.3	8.5	$3*10^{-2}$
C6	0.4	0.75	0.6	9	$2.1*10^{-1}$
C7	0.7	0.25	0.3	12.2	0
C8	0.7	0.25	0.6	7.5	$1.4*10^{-3}$
C9	0.7	0.5	0.3	12.1	$3*10^{-2}$
C10	0.7	0.5	0.6	21.7	$1.9*10^{-2}$
C11	0.7	0.75	0.3	13	$1.3*10^{-2}$
C12	0.7	0.75	0.6	14.8	$2.3*10^{-1}$

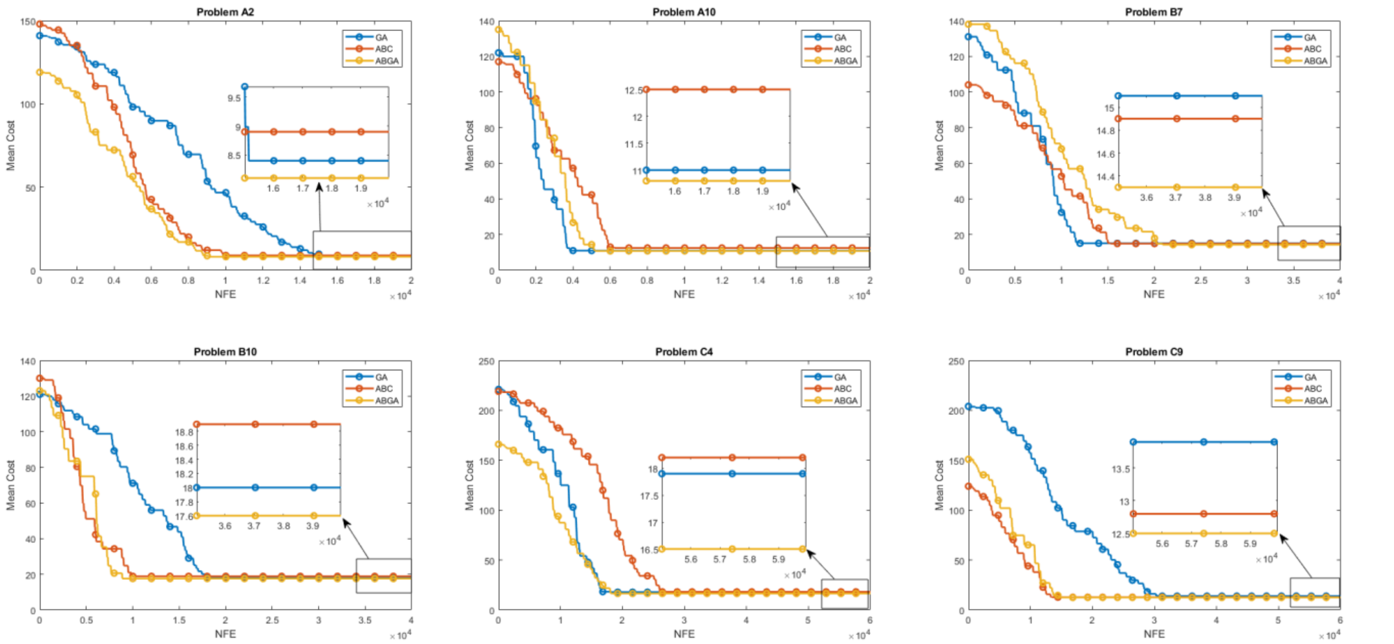

Fig. 15. Convergence of the Algorithms for Random Selected Test Problems

performs better than *GA* and *ABC* in finding a better solution for the large size test problems.

7- Conclusion and Further Research

Two models were introduced for robust resource-constrained multi-project scheduling problem (*RRCMPSP) in this paper. In the first model, the maximum tardiness of the projects was minimized while the projects were scheduled separately as single project scheduling problems. The second model considered the projects as a multi-project problem while the common resources were shared among the projects. The durations of the activities were uncertain and expressed by discrete values.* The advantages of the multi-project approach in scheduling the projects under uncertainty in comparison with the scheduling of the projects separately was presented by a numerical example. The *ABGA* algorithm was designed and presented in this paper as a hybrid metaheuristic algorithm for the large size *RRCMPSP. The performance of the proposed algorithm was validated according to the results of exact algorithm for the small size RRCMPSP. So, the scenario-relaxation algorithm was implemented for the proposed models. In addition, for the large size RRCMPSP,*

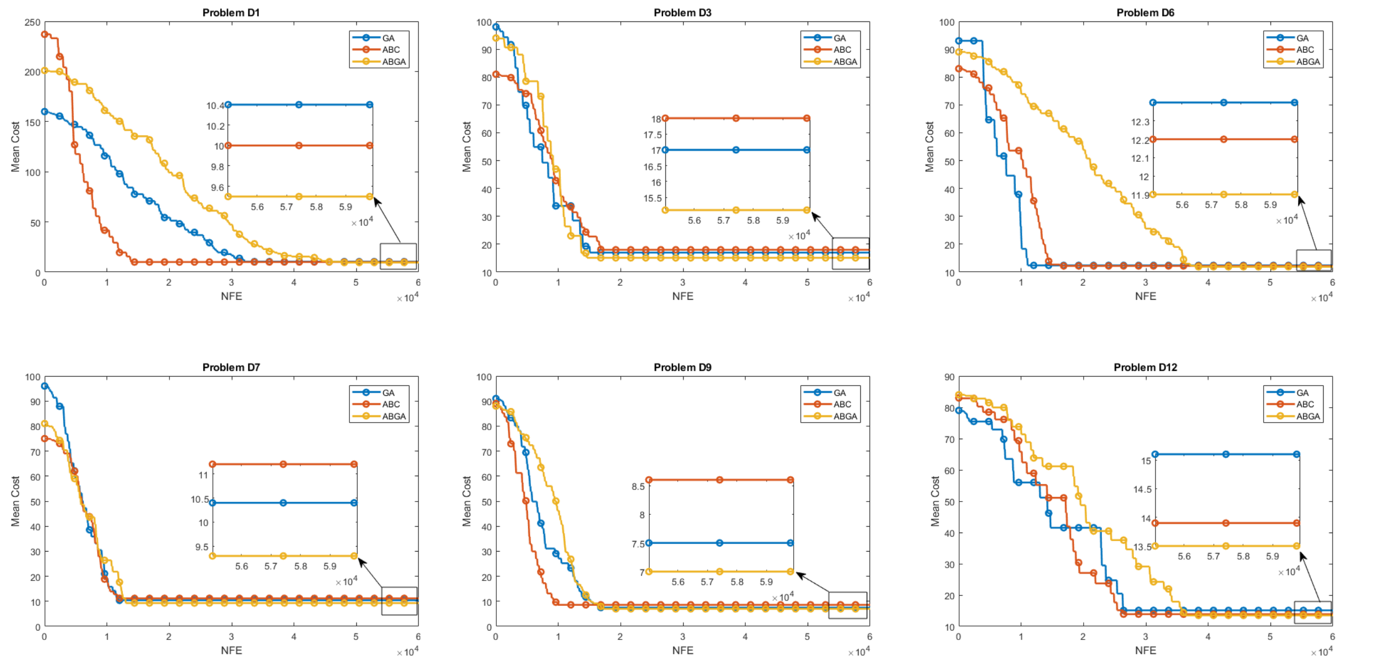

Fig. 16. Convergence of the Algorithms for Test Problems D1, D3, D6, D7, D9, D12

Table 5. Computational Results of RRCMPSPTT for n=40

Functions	Index	GA	ABC	ABGA
D1	Mean Cost ± std. Dev.	10.4±0.1	10±0.15	9.5±0.05
	Mean NFE ± std. Dev.	31000±1200	13320±980	43800±800
	Rank	3	2	1
D2	Mean Cost ± std. Dev.	13.5±0.5	13.1±0.04	12.8±0.1
	Mean NFE ± std. Dev.	17600±420	14200±670	28500± 1250
	Rank	3	2	1
D3	Mean Cost ± std. Dev.	17 ± 0.2	18 ± 0.1	15.1 ± 0
	Mean NFE ± std. Dev.	14530 ± 650	16290 ± 1050	14250 ± 450
	Rank	2	3	1
D4	Mean Cost ± std. Dev.	21.2 ± 0.17	21.8 ± 0.25	20.9 ± 0.3
	Mean NFE ± std. Dev.	23800 ± 1450	11300 ± 580	26400 ± 1100
	Rank	2	3	1
D5	Mean Cost ± std. Dev.	14.8 ± 0.05	14.2 ± 0.1	14.4 ± 0.2
	Mean NFE ± std. Dev.	13700 ± 880	31500 ± 1440	25230 ± 2100
	Rank	3	1	2
D6	Mean Cost ± std. Dev.	12.4 ± 0.3	12.2 ± 0.2	11.9 ± 0.1
	Mean NFE ± std. Dev.	10490 ± 670	15280 ± 1100	37650 ± 330
	Rank	3	2	1
D7	Mean Cost ± std. Dev.	10.4 ± 0.05	11.2 ± 0.1	9.3 ± 0
	Mean NFE ± std. Dev.	11200 ± 680	10760 ± 1240	11340 ± 1800
	Rank	2	3	1
D8	Mean Cost ± std. Dev.	15.9 ± 0.1	15.7 ± 0.2	15.3 ± 0
	Mean NFE ± std. Dev.	18570 ± 430	22640 ± 890	34200 ± 1750
	Rank	3	2	1
D9	Mean Cost ± std. Dev.	7.5 ± 0.02	8.6 ± 0.06	7 ± 0
	Mean NFE ± std. Dev.	15840 ± 660	9710 ± 310	15060 ± 2100
	Rank	2	3	1
D10	Mean Cost ± std. Dev.	20.8 ± 0.1	20.6 ± 0.07	19.7 ± 0.05
	Mean NFE ± std. Dev.	15690 ± 1320	33700 ± 1600	28900 ± 750
	Rank	3	2	1
D11	Mean Cost ± std. Dev.	11.1 ± 0	11.4 ± 0.24	10.5 ± 0.16
	Mean NFE ± std. Dev.	18950 ± 210	14800 ± 1100	13550 ± 530
	Rank	2	3	1
D12	Mean Cost ± std. Dev.	15.1 ± 0.1	13.9 ± 0.04	13.5 ± 0
	Mean NFE ± std. Dev.	25620 ± 840	25600 ± 920	34980 ± 1200
	Rank	3	2	1

the computational results of ABGA were compared by GA and ABC algorithms and showed that ABGA performed better than these algorithms.

Other constraints can be added as an extension of the proposed model like the multi-mode activities, nonrenewable resources, multi-skill resources, etc. Developing other algorithms in order to solve large size RRCMPSP is another extension. Finally considering the cost of projects as a second objective function and having a robust time-cost tradeoff model would be of interest.

References

1. Lova, A., and Tormos, P., "Analysis of scheduling schemes and heuristic rules performance in resource-constrained multiproject scheduling", Annals of Operations Research, Vol. 102, pp. 263–286, 2001.
2. Blismass, N., Sher, WD., Thorpe, A., and Baldwin, AN., "Factors influencing delivery within construction clients' multi-project environments", Engineering, Construction and Architectural Management, Vol. 11, No. 2, pp. 113–125, 2004a.
3. Aritua, B., Smith, N.J., and Bower, D., "Construction client multi-projects – A complex adaptive systems perspective", International Journal of Project Management, Vol. 27, pp. 72–79 (2009).
4. Zheng, Z., Shumin, L., Ze, G., and Yueni, Z., "Resource constraint multi project scheduling with priority and uncertain activity durations", International Journal of Computational intelligence systems, Vol. 6, No. 3, pp. 530-547, 2013.
5. Singh, A., "Resource Constrained Multi-Project Scheduling with Priority Rules & Analytic Hierarchy Process", Procedia Engineering, Vol. 69, pp. 725 – 734, 2014.
6. Wei-xin, W., Xu, W., Xian-long, G., and Lei, D., "Multi-objective optimization model for multi-project scheduling

on critical chain", Advances in Engineering Software, Vol. 68, pp. 33–39, 2014.

7. Wauters, T., Kinable, J., Smet, P., Vancroonenburg, W., Berghe, G., and Verstichel, J., "The Multi-Mode Resource-Constrained Multi-Project Scheduling Problem", Journal of Scheduling, The MISTA 2013 challenge, 2013.

8. Besikci, U., Bilge, U., and Ulusoy, G., "Multi-mode resource constrained multi-project scheduling and resource portfolio problem", European Journal of Operational Research, Vol. 240, pp. 22–31, 2015.

9. Zheng, Z., Shumin, L., Ze, G., Yueni, Z., and Zhang, X., "A critical chains-based distributed multi-project scheduling approach", Neurocomputing, Vol. 143, pp. 282–293, 2014.

10. Klerides, E., and Hadjiconstantinou, E., "A decomposition-based stochastic programming approach for the project scheduling problem under time/cost trade-off settings and uncertain durations", Computers and Operations Research, Vol. 37, No. 12, pp. 2131–2140, 2010.

11. Bertsimas, D., and Sim, M., "The price of robustness", Operations Research, Vol. 52, No. 1, pp. 35-53, 2004.

12. Bruni, M.E., Puglia Pugliese, L., Di Beraldi, P., and Guerriero, F., "An adjustable robust optimization model for the resource-constrained project scheduling problem with uncertain activity durations", Omega, Vol. 71, pp. 66-84, 2017.

13. Herroelen, W., and Leus, R., "Project scheduling under uncertainty: Survey and research potentials", European Journal of Operational Research, Vol. 165, pp. 289–306, 2005.

14. Klerides, E. and Hadjiconstantinou, E. "A decomposition-based stochastic programming approach for the project scheduling problem under time/cost trade-off settings and uncertain durations", Computers and Operations Research, 37(12), pp. 2131–2140 (2010).

15. Chiang, A.J., and Jeang, A., "Stochastic project management via computer simulation and optimisation method", International Journal of Systems Science: Operations & Logistics, Vol. 2, No. 4, pp. 211–230, 2015.

16. Xiong, J., Chen, Y. W., Yang, K. W., Zhao, Q. S., and Xing, L. N., "A hybrid multi objective genetic algorithm for robust resource constrained project scheduling with stochastic durations", Mathematical Problems in Engineering, 24 pages, 2012.

17. Duc Long, L., and Ohsato, A., "Fuzzy critical chain method for project scheduling under resource constraints and uncertainty", International Journal of Project Management, Vol. 26, pp. 688–698, 2008.

18. Knyazeva, M., Bozhenyuk, B., and Rozenberg, I., "Resource-constrained project scheduling approach under fuzzy conditions", Procedia Computer Science, Vol. 77, pp. 56-64, 2015.

19. Dixit, V., Srivastava, R.K., and Chaudhuri, A., "Procurement scheduling for complex projects with fuzzy activity durations and lead times", Computers & Industrial Engineering, Vol. 76, pp. 401–414, 2014.

20. Masmoudi, M., and Haït, http://www.sciencedirect.com/science/article/pii/S095219761200200X A., "Project scheduling under uncertainty using fuzzy modelling and solving techniques", Engineering Applications of Artificial Intelligence, Vol. 26, No. 1, pp. 135–149, 2013.

21. Artigues, C., Leus, R., and Nobibon, F. T., "Robust optimization for resource-constrained project scheduling with uncertain activity durations", Flexible Services and Manufacturing Journal, Vol. 25, pp. 175–205, 2013.

22. Yamashita, D.S., Armentano, V. A., and Laguna, M., "Robust optimization models for project scheduling with resource availability cost", Journal of scheduling, Vol. 10, pp. 67–76, 2007.

23. Chakrabortty, R.K., Sarker, R.A., and Essam, D.L., "Resource constrained project scheduling with uncertain activity durations", Computers & Industrial Engineering, http://dx.doi.org/10.1016/j.cie.2016.12.040, 2017.

24. Bruni, M. E. Di Puglia Pugliese, L. Beraldi, P., and Guerriero, F. A., "computational study of exact approaches for the adjustable robust resource constrained project scheduling problem". Computers & Operations Research, Vol. 99, pp. 178-190, 2008.

25.Balas, E., "Project scheduling with resource constraints", In E.M.L. Beale, (Ed.), Applications of Mathematical Programming Techniques, pp. 187-200, The English Universities Press Ltd, 1971.

26. Assavapokee, T., Realff, M.J., Ammons, J.C. and Hong, I-H. "Scenario relaxation algorithm for finite scenario-based min–max regret and min–max relative regret robust optimization", Computers & Operations Research, Vol. 35, pp. 2093 – 2102, 2008.

27. Patterson, J. H., "Project Scheduling: The Effects of Problem Structure on Heuristic Performance", Naval Research Logistic, Vol. 23, No. 1, pp. 95-123, 1976.

28. Blazewicz, J. Lenstra, J. K., and Rinnooy Kan, A. H. G., "Scheduling subject to resource constraints: classification and complexity". Discrete Applied Mathematics, Vol. 5, No. 1, pp. 11 – 24, 1983.

29. Goncalves, J. F., Mendes, J. J. M., and Resende, M. G. C., "A genetic algorithm for the resource constrained multi-project scheduling problem". European Journal of operational Research, 189, pp. 1171-1190, 2008.

30. Kim-Fung, M., Tang, K., and Kwong, S. "Genetic algorithms: concepts and applications [in engineering design". IEEE transactions on Industrial Electronics, Vol. 43, No. 5, pp. 519-534, 1996.

31. Dervis, K., "An idea based on honey bee swarm for numerical optimization". Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, Vol. 200, 2005.

32. Demeulemeester, E., Vanhoucke, M. and Herroelen, W. "A random network generator for activity-on-the-node networks". Journal of Scheduling, 6, pp. 13-34, 2003.

33. Nabipoor Afruzi, E. Aghaie, A., and Najafi, A. A., "Robust Optimization for the Resource Constrained Multi-Project Scheduling Problem with Uncertain Activity Durations". Scientia Iranica, Articles in Press, Accepted Manuscript, Available Online

from 05 August 2018.

34. Srinivas, M., and Patnaik, L. M. "Adaptive Probabilities of Crossover Genetic in Mutation and Algorithms". IEEE Transactions on Systems, Man and Cybernetics, Vol. 24, No. 4, pp. 656-667, 1994.

35. Eiben, A. E. Marchiori, E., and Valko, V. A., "Evolutionary Algorithms with on-the-fly Population Size Adjustment". In X. Yao et al., editors, Parallel Problem Solving from Nature PPSN VIII, LNCS 3242, pp. 41–50, 2004.

HOW TO CITE THIS ARTICLE

E. Nabipoor Afruzi[1], A. Aghaie, A Hybridized Metaheuristic Algorithm to Solve the Robust Resource Constrained Multi-Project Scheduling Problem, AUT J. Model. Simul., 51(1) (2019): 15 - 32.

DOI: 10.22060/miscj.2019.15033.5121