Nima SIAMAKMANESH, MSc
E-mail: nsiamakmanesh@mail.kntu.ac.ir
Faculty of Industrial Engineering
K. N. Toosi University of Technology, Tehran, Iran
Associate Professor Amir Abbas NAJAFI, PhD (Corresponding Author)
E-mail: E-mail: aanajafi@kntu.ac.ir
Faculty of Industrial Engineering
K. N. Toosi University of Technology, Tehran, Iran
Lecturer Fatemeh AZIMI
E-mail: ff_azimi@yahoo.com
Department of Mathematics, Qazvin Branch
Islamic Azad University, Qazvin, Iran

A WORK-CONTENT BASED RESOURCE AVAILABILITY COST PROBLEM: MATHEMATICAL MODELING AND SOLVING PROCEDURE

Abstract. *Resource availability cost problem (RACP) is a project scheduling problem in which the resource availability levels are decision variables and the goal is to minimize the project resource costs. To make the RACP more realistic, the concept of work-content may be incorporated into the problem in order to lead to a reduction of costs. In this paper, the mathematical model for the problem under study is first proposed and a self-adaptive genetic algorithm (GA) is applied to tackle the problem. The self-adaptive approach allows the GA to utilize both serial and parallel schedule generation schemes simultaneously. Moreover, a local search operator is designed to improve the performance of the GA and its crossover and mutation operators. Results of several numerical instances show that the proposed GA performs relatively well. In addition, it is shown that incorporating the concept of the work-content into the RACP leads to reduce the project costs.*

Keywords*: Self-adaptive; Work-content; Genetic algorithm; Project scheduling.*

JEL Classification: C44, E40

1. Introduction

The resource availability cost problem is a form of project scheduling problem (PSP) in which the resource availability levels are decision variables and the goal is to schedule the activities in a way that minimizes the resource availability costs. Möhring (1984) introduced the RACP as a NP-hard problem and

DOI: 10.24818/18423264/56.1.22.05

applied an exact method to solve it. Later on Demeulemeester (1995) proposed another exact algorithm based on branch and bound, to solve the problem. Drexl & Kimms (2001) introduced the column generation and Lagrangian relaxation based techniques for obtaining the lower & upper bounds to tackle the RACP problem. Hsu & Kim (2005) proposed a priority rule heuristic to address the multi-mode version of the RACP. Yamashita et al. (2006) utilized a scatter search heuristic to solve the RACP problem which incorporates various advanced strategies such as path relinking based combination method. It should be noted that they also developed a multi-start heuristic for problems of medium and large sizes. Later on Yamashita et al. (2007) deepened their research by considering uncertainty about activity durations. This was done by defining a set of scenarios and robust optimization techniques to model the problem. In Yamashita et al. (2007) scatter search was used to solve the problem. The optimal solution for small instances and the output of a multi-start heuristic for medium and large instances were used to evaluate their heuristic. By means of the genetic algorithm (GA) Shadrokh & Kianfar (2007) solved a variation of the problem where tardiness of the project is permitted with penalty. Ranjbar et al. (2008) solved the RACP problem using path relinking and genetic algorithm. For more researches in this filed, we refer to Hartmann and Briskorn (2022), Khalili et al. (2013), Najafi et al. (2009), Shavandi et al. (2012) and Azimi and Fathallahi (2016) .

The standard project scheduling problems such as the resource constrained project scheduling problem (RCPSP) and the RACP consider the duration and the renewable resources demand levels of the activities to be constant during their execution. While the multi-mode versions of the RCPSP and the RACP consider more than only one combination of duration and resource demand for each activity, similar to the single-mode, in the multi-mode versions of both RCPSP and RACP problems, the resource demand is constant throughout each activity's execution. On the other hand, in reality, the resource allocated to each activity usually cannot remain constant throughout its execution and with respect to the project's condition and the availability levels of resources, the project manager may change the resource allocated to each day of an activity's execution. As a result in reality, activities' daily resource demand levels can be considered as decision variables. In order to solve this problem the concept of work-content can be utilized. In this technique neither the renewable resource demand nor the duration of activities are constant. Instead, any level of the renewable resource between the lower and upper bounds can be allocated to the activity in each period as long as the total resource allocated to the activity equals the activity's required work-content.

In the literature of project scheduling the concept of work-content has been identified by the following titles: work-content constraints, flexible resource profiles and flexible work profiles. The concept of work-content was first incorporated into the RCPSP by Kolisch et al. (2003). Fündeling and Trautmann (2010) utilized a priority rule method to tackle the RCPSP with work-content constraints. In their model a minimum time lag between consecutive changes in the demand of the work-content resource was required which they called the minimum block length. Further research in the field of RCPSP with flexible resource profiles (FRCPSP) is presented in Naber & Kolisch (2014). While there have been cases of utilizing the concept of work-content in the field of RCPSP as mentioned above, to the best of our knowledge the concept has not been incorporated into the RACP. In this paper, the concept of work-content is used to tackle the RACP for the first time and the model is presented. The characteristics of the GA are discussed and the computational results are studied.

This paper is organized as follows: The proposed work-content based RACP model is described in section 2. The basic scheme, chromosome structure, proposed self-adaptive approach, crossover, mutation and the local search operators alongside the pseudo-code of the proposed self-adaptive genetic algorithm are discussed in section 3. Section 4 is dedicated to comparing the work content and single mode approaches and the review of the computational results. Finally in section 5 the conclusion of this study is presented.

2. The proposed work-content based RACP model

In the work-content based RACP a single project consisting of $n+2$ activities is considered where the dummy activities 0 and $n+1$ denote the start and the finish of the project respectively. The precedence relations between all activities are Finish-Start type with a time lag of zero which is depicted by an activity-on-node network (AON). For each activity j a set of predecessors, $P(j)$ is assumed. In this model t denotes time and the project must finish by T, the project deadline. In this paper for every t, the set V_t represents the list of activities that are in progress.

There are $K+1$ renewable resources required in the project; one main resource which is also known as the work-content resource and K dependent resources. Each activity requires w_i units of the work-content resource throughout its execution. The work-content resource allocated to activity i during the interval $[t, t+1)$ is one of the main decision variables of the proposed model and is denoted by r_{it}. It should be noted that based on the nature of each activity, r_{it} is bounded by the lower, LB_{r_i}, and upper bounds, UB_{r_i}.The dependent renewable resource type k required by activity i is a function of r_{it} and is shown as $r_{ik}(r_{it})$. The cost for the availability of the work-content resource, R is C per unit and for the availability of the dependent resource k, R_k the cost is C_k per unit. Similarly to $r_{ik}(r_{it})$, the values for R and R_k are calculated using r_{it}.

The second main decision variable is x_{it} which is a binary variable and takes 1 if activity i is in progress during $[t, t+1)$ and 0 otherwise. Since there are bounds to the minimum and the maximum work-content resource an activity may use during its execution, there are also bounds to the duration of the activity, d_i, denoted by LB_{d_i} and UB_{d_i}. s_i and f_i are start and finish times for each activity which are calculated by means of x_{it}. It should be noted that in the proposed model preemption is not allowed. The problem can be modeled as follows.

$$Min \quad z = CR + \sum_{k=1}^{K} C_k R_k \tag{1}$$

Subject to:

$$\sum_{t=0}^{T} r_{it} = w_i \quad \forall i \tag{2}$$

$$\sum_{i \in V_t} r_{it} \le R \quad \forall t \tag{3}$$

$$\sum_{i \in V_t} r_{ik}(r_{it}) \le R_k \quad \forall k, \forall t \tag{4}$$

$$LB_{r_i}.x_{it} \le r_{it} \le UB_{r_i}.x_{it} \quad \forall i, \forall t \tag{5}$$

$$LB_{d_i} \le d_i \le UB_{d_i} \quad \forall i \tag{6}$$

$$d_i = \sum_{t=0}^{T} x_{it} \quad i \in \{1, \dots, n\} \tag{7}$$

$$f_i = \max_t (x_{it}.t) + 1 \quad i \in \{1, \dots, n\} \tag{8}$$

$$s_i = \min_t (t.x_{it} + (1 - x_{it}).M) \quad \forall i \tag{9}$$

$$s_i = f_i - d_i \quad \forall i \tag{10}$$

$$s_j \ge f_i \quad \forall i \in P(j) \tag{11}$$

$$\max(f_i) \le T \tag{12}$$

$$r_{it} \in Z^{\ge 0} \tag{13}$$

Equation (1) shows the availability cost incurred to the project for all resources. Constraint (2) stresses the fact that the total work-content resource allocated to each activity during its execution must be equal to the activity's work content. Constraints (3) and (4) prevent over allocation of work-content resource and dependent resources, respectively. Constraint (5) bounds the work-content resource allocated to each activity during its execution to the upper and lower bounds in each interval. The equation also limits the work-content resource allocated to each activity to 0 when it is not in progress. By means of Constraint (6) the duration of each activity is also restricted to the lower and upper bounds. The calculation of these bounds is shown by Equations (14) and (15) below. Constraint (7), (8) and (9) calculate the duration, the finish time and the start time of each activity. It should be noted that in the duration and finish times of dummy activities are not calculated by Constraints (7) and (8). The duration of dummy activities are set to zero and thus their finish times will be equal to their start times by means of Constraint (10) which makes the preemption of activities impossible. The precedence relations between activities is shown in Constraint (11). Constraint (12) guarantees that the project finishes before the deadline and Constraint (13) restricts the r_{it} to integer and non-negative values.

As mentioned previously, the values of r_{it} are bound by lower and upper bounds. If throughout an activity's execution the minimum possible amount of work-content resource is allocated to the activity in every interval, namely LB_{r_i}, the maximum possible duration of the activity is found, namely UB_{d_i}. Similarly using w_i and UB_{r_i}, the value of LB_{d_i} can be calculated. These calculations are shown by Equations (14) and (15).

$$UB_{d_i} = \left\lceil \frac{w_i}{LB_{r_i}} \right\rceil \text{ where } \lceil x \rceil \text{ is the smallest integer equal to or greater than } x \tag{14}$$

$$LB_{d_i} = \left\lceil \frac{w_i}{UB_{r_i}} \right\rceil \text{ where } \lceil x \rceil \text{ is the smallest integer equal to or greater than } x \tag{15}$$

3. Self-adaptive Genetic algorithm

In this section, different parts of the proposed Self-adaptive genetic algorithm are discussed. As shown by Möhring (1984) the RACP is a NP-hard problem which calls for the use of meta-heuristics. The review of the literature of the RACP in section 1 shows that the genetic algorithm is the most frequently technique used and as a result it is chosen for this study.

The two serial and parallel schedule generation schemes have different approaches towards scheduling the activities of a project which will lead to generation of different schedules. In order to let the proposed GA probe the solution space more thoroughly a self-adaptive approach is proposed.

3.1. Basic scheme

The primary population is created randomly. For each individual in the population one of the two serial and parallel schedule generation schemes is chosen with a probability of 50% and then the local search operator is used for the individual in order to reach a solution with lower resource demand in its neighborhood. Then the unfitness function i.e. the resource availability cost for each individual is calculated. For the next generations the self-adaptive approach is utilized for choosing the schedule generation scheme which is further discussed in sub-section 3.3. In each generation, the elite individuals are directly copied into the next generation. Then, based on the unfitness function values, pairs of individuals are selected from the population for crossover and mutation. For each pair, with a probability of P_{cr} the crossover operator is used. Whether the crossover operator is implemented on the pair or not, the outputs, which are either the parents or their crossover made children, are then processed by the mutation operator with a probability of P_{mu} and copied to the next generation. It should be noted that at any point during the crossover and mutation procedures, if a pair is not chosen to be processed by each operator, it is directly sent to the next step. To sum up the procedure, it can be said that there are only 4 types of outputs:

- Individuals that have undergone both crossover and mutation operators
- Individuals that have been only processed by crossover operator
- Individuals that have been only processed by mutation operator
- Individuals that have not been processed by neither of above operators

The resulting population is then processed by the local search and the cost corresponding to each individual is calculated. At the end of the construction of each generation, the individuals are sorted with respect to their unfitness function values and the new elite individuals are identified. This procedure is repeated until the termination condition is satisfied.

3.2. Chromosome structure

In the proposed algorithm each chromosome consists of 4 parts. The first part contains the activity list random values and is used to achieve a precedence feasible activity list. The second part is the duration profile. Part 3 contains the resource profile random values which is translated into a resource profile for the activities. The last part presents the resource availability levels. Each part is further discussed in the following. Since the start and finish dummy activities do not require any computation, for ease of computation they have not been considered in the proposed algorithm.

For the sake of simplicity the activity list random values matrix is going to be called AR and the corresponding precedence feasible activity list is going to be known as the AL. AR is a $1 \times n$ matrix of random numbers between 0 and 1 and AL is a $1 \times n$ matrix of zeroes. Each member of AL represents the corresponding activity. For example, the fifth member of the AL represents the priority level of activity 5. The process consists of *n* steps where in each step a set of candidates is defined. The candidates set consists of two groups of activities: The first group are the activities that do not have any predecessors and the second group are the activities that all of their predecessors have non-zero values in AL. It should be noted that activities with non-zero values in AL cannot be members of the candidates set. At each step the AR values of the candidates set is checked and the activity with the minimum value in AR is chosen to take the next priority level in AL which means its value changes from 0 to the step number. Here 1 represents the highest priority level and *n* represents the lowest. Since the technique filters the candidates set, the output is always precedence feasible.

The duration profile is a $1 \times n$ matrix of durations for activities where each member of the matrix represents the corresponding activity's duration. For example the third value in the duration profile matrix represents the duration of activity 3. The duration values of each member of the matrix always must be between the lower and upper bounds for the activity's duration and therefore the profile is always feasible regarding the duration bounds.

The resource profile random values, here known as the RP; is a matrix with *n* rows as for the number of activities and $\max_i \{UB_{d_i}\}$ columns as for the maximum possible duration of activities. With respect to the constraints regarding the work-content resource, the RP is translated into a resource profile matrix of the same size. The translation is performed in a manner that the sum of the values in each row of the resource profile matrix equals the work-content resource required by the corresponding activity. For example, the sum of the values of the resource profile's third row equals w_3. In the end, the resource profile for the dependent activities is calculated using the relation previously defined between each dependent resource and the work-content resource. In the RP, work-content resource profile and the dependent resources profiles, each column represents the number of the period since each activity's start. For example the value in the third row and the fifth column of the work-content resource profile, represents the work-content resource activity 3 requires on the fifth period since its start.

The resource availability matrix is matrix of 1 row and $K+1$ columns where the first member of the matrix represents the availability level of the work-content resource throughout the project and the rest of the members represent the dependent resources availability levels.

3.3. Serial, parallel and self-adaptive approaches

A schedule generation scheme (SGS) transforms an activity list into a schedule with respect to the resource availability constraints. In the literature of the RACP both schemes are utilized (Hsu & Kim, 2005; Kolish, 1996; Shadrokh and Kianfar, 2007; Shahsavar et al., 2011). Both schedule generation schemes use a priority list which is a precedence feasible list of activities that shows how urgent it is to schedule one activity before scheduling the rest. In the following the procedures of the serial and the parallel schedule generation schemes are briefly discussed and then the self-adaptive approach is presented.

3.3.1. Serial schedule generation scheme

The serial schedule generation scheme consists of *n* stages for a project of *n* activities. Each stage represents one activity. At each stage the activity with the highest priority level from the list of not yet scheduled activities is chosen and is scheduled as soon as possible in a precedence and resource feasible manner. The procedure is repeated until all activities have been scheduled. For further study on SSGS and its characteristics one can refer to Shadrokh & Kianfar (2007).

3.3.2. Parallel schedule generation scheme

In the parallel schedule generation scheme instead of activities, the main focus is on time. The PSGS consists of at most *n* stages for a project of *n* activities. Each stage represents a time when there is a set of yet unscheduled activities that are candidates for scheduling based on the precedence relations and start and finish times of their predecessors. First in each stage activities are sorted by their priority levels. Next with respect to the resource availability constraints, candidates are checked for scheduling. If scheduling an activity can violate the resource availability constraints, the algorithm moves to the next member of the set without scheduling the activity. For a comprehensive study on SSGS & PSGS one can refer to Kolisch (1996).

3.3.3. Proposed self-adaptive approach

As explained in Sub-section 3.1, primarily each one of the schedule generation schemes have an equal chance of being applied to each individual. Later on, in the beginning of creating each generation, these chances are modified with respect to the unfitness values of the previous generation. In order to do this, the population of the previous generation is divided into two groups: The group of individuals that SSGS was applied to them and the group of individuals that PSGS was applied to them. Then the sum of unfitness values for each group is calculated and the probability modification process gives a higher chance of being chosen for applying to the schedule generation scheme that has a lower total unfitness value. By doing this, while the search space is studied more thoroughly, the effectiveness of the proposed genetic algorithm is enhanced. It should be noted that in order to prevent the procedure from acquiring unnecessary excessive units of resources, both SSGS and PSGS reduce the primary resource availability levels to the maximum resource required by their output schedules at the end of their procedures.

3.4. Crossover operator

As explained in sub-section 3.2 the chromosome structure in the proposed self-adaptive GA consists of 4 parts. All of these parts are processed by the crossover operator in different ways as follows:

- The activity list random values matrix i.e. AR, is a matrix of random values between [0,1] and since the matrix is later on translated into a precedence feasible activity list, changing the values of the matrix under no circumstances will not create infeasible outputs. In the proposed method, a continuous uniform crossover is applied to the AR. The continuous uniform crossover takes two parents $p^f = \{p_1^f, p_2^f, \dots, p_n^f\}$, $p^m = \{p_1^m, p_2^m, \dots, p_n^m\}$ and also creates a matrix of random numbers between [0,1] of the same size of the AR and calculates the AR values for children: $ch_i^s = p_i^f \times \alpha_i + p_i^m \times (1-\alpha_i)$ and $ch_i^d = p_i^f \times (1-\alpha_i) + p_i^m \times \alpha_i$ for *i* where $ch^s = \{ch_1^s, ch_2^s, \dots, ch_n^s\}$ and $ch^d = \{ch_1^d, ch_2^d, \dots, ch_n^d\}$.
- Similarly to AR, a continuous uniform crossover is also applied to RP where the matrix of random values (β) has a size equal to the RP matrixes.
- The crossover procedure calculates the work-content resource availability level using the technique introduced by Shadrokh & Kianfar (2007). The dependent resources availability levels are calculated using their mathematical relation with the work-content resource. It should be noted that in the proposed method for the primary population the work-content resource lower bound for availability, $\underline{R}$,is the maximum possible work-content resource demand which is $\max_i\{UB_{r_i}\}$. For the next generations the lower and upper bounds are modified with regard to the population's work-content resource availability levels in order to achieve a higher chance of reaching solutions with lower unfitness values. The dependent resources availability

levels are at all times calculated using their relation with the work-content resource.

- As explained above the GA procedure decreases the resources availability levels in a way that sometimes the work-content resource availability level may be lower than the primary maximum possible work-content resource demand, UB_{r_i}. This means that the primary upper bounds for the work-content resource demand will sometimes lead to infeasible solutions. In order to prevent the procedure from reaching infeasible solutions regarding the resource demand, the upper bounds of work-content resource demand of activities are modified which leads to the modification of the duration bounds. The crossover procedure generates new duration values for the children with regard to the modified duration bounds.

3.5. Mutation operator

Similar to the crossover operator, all 4 parts of the chromosome are processed by the mutation operator. The 4 steps of the proposed mutation operator are as follows:

- The mutation operator first attends the AR. First an activity *i* is chosen randomly and then its value in the matrix, x_i, is replaced with $1-x_i$.
- Similarly to the AR value, the values of the RP matrix's row *i*, $y(i, j)$ are replaced with $1-y(i, j)$ for all *j*.
- Next, the resource availability matrix is processed. The mutation operator chooses one resource type r out of $K+1$ resources randomly and then replaces the value a_r with $a_r - 1$. At the end, the procedure adjusts the availability levels of the rest of the resource types in a way that feasibility of the output resource availability levels matrix is ensured.
- Similar to the technique utilized for duration values in the crossover operator, in the mutation operator the lower and upper bounds for duration are modified and then the duration of activity *i* is randomly chosen with respect to the lower and upper bounds for duration.

3.6. Local search operator

The proposed local search operator helps to reach better solutions by changing the work-content and dependent resource profiles. It should be noted that similarly to the crossover procedure, here only the work-content resource undergoes changes and then the dependent resources profiles are modified with regard to the work-content resource usage profile. This is done by identifying periods of the project that have the maximum resource demand of one or more resource types. The operator then changes the resource profile in a way that the excess resource demand is moved to other periods thus changing the maximum resource demand throughout the project by reducing the resource availability levels to lower new values. The procedure consists of *n* steps for a project of *n* real activities. At each step *i* the resource profile of activity *i* is checked and if necessary changed. In order to do this two sets are defined in each step:

- Set of candidates for decreasing the resource usage: candidates are the periods that the total resource demand for at least one of the resource types is equal to the resource availability levels and periods that the work-content resource demand of activity *i* is more than the lower bound, LB_{r_i} which will prevent the local search from creating infeasible resource profiles.
- Set of possible options for increasing the resource usage: once the candidates are identified and their excess resource usage is removed, a set of options is defined. The set consists of periods that adding the excessive resource demand will not cause new periods with maximum resource demand. It should be noted that the options set should not consist of periods that adding the excessive resource usage might violate the upper bound limit for activity *i*.

4. Computational results

In this section first using the example presented in section 3.6 it is shown that the concept of work-content helps to decrease the resource availability costs significantly. In order to do this, the example is translated into a series of single mode problems and the results are compared. Next, to validate the proposed meta-heuristic, a set of problems is generated, solved and compared to the global optimum solution acquired using the exact method on a computer with an i5 2.60 GHz processor and 4 GB RAM. Sub-section 4.1 is dedicated to showing the importance of the concept of work-content and its effect on the resource availability costs. Details regarding the evaluation of the proposed self-adaptive genetic algorithm including designing the problem set, solving the problems using the proposed self-adaptive generic algorithm and comparing its outputs with the global optimum solutions are presented in sub-section 4.2.

4.1. The importance of incorporating the work-content concept

In this section it is shown that the concept of work-content leads to a notable reduction of resource demand and consequently the total resource availability cost. To do this, an example is translated into 40 single-mode resource availability cost problems. To generate these problems, some modes of daily work-content resource usage and duration have been considered for each activity and a random combination of modes creates a problem.

After solving the problems, for 18 instances there could not be a feasible solution with the given deadline (45% of the instances). Also the results show that for all of the 22 remaining problems (55% of the instances), the single mode RACP could not reach the work-content based resource availability levels and their average resource availability cost was 59.45% percent higher than the work content based resource availability cost which is 252. This shows that while the conventional single-mode RACP cannot reach a feasible solution in almost half of the instances, the work-content approach can significantly reduce the availability cost and thus, it can reach more practical and desirable solutions.

4.2. Evaluation of the proposed self-adaptive genetic algorithm

In this subsection, the performance of the meta-heuristic is evaluated by solving 48 instances of problems. The problems are generated using the RanGen network generator introduced by Demeulemeester et al. (2003) and can be divided into 12 groups based on the number of real activities and the total number of resource types. It should be noted that the data required for each instance of problem that were not generated by RanGen, were generated randomly. There are 4 cases for the number of real activities: less than 10 activities, 10 activities, 15 activities and 20 activities. There are also 3 cases for the number of resources: 3, 4 and 5 resource types. In each case for the number of resources one of the resources is the work-content resource and considered to be the main resource. For each combination of number of activities and number of resource types, 4 problems are generated and solved.

In the following a comparison of the results obtained from the proposed meta-heuristic and the global optimum results acquired by solving the problems using the GAMS® software are presented. The run times are also presented in the tables for a better comparison. The results for 3, 4 and 5 resource type problems are shown in Tables 1, 2 and 3, relatively. In each table the average and the maximum values for deviation and run time and the percent of cases with less than 5% deviation are presented. Also the average values for parallel and serial schedule generation schemes probabilities for each combination of number of activities and number of resources are presented. The legend of these tables is as follows:

A: % deviation from the global optimum solution.

B: Percentage of lower than 5% deviation observations.

C: Run time in seconds

D: Average values PSGS and SSGS probabilities.

Table 1. Three resource type cases

No. of activities	No. of instances	A		B	C				D	
					GA		GAMS			
		Average	Max		Average	Max	Average	Max	Parallel	Serial
<10	4	0.30%	1.19%	100%	22.03	24.18	24.08	25.73	50%	50%
10	4	3.03%	5.52%	75%	42.51	49.00	3430.09	13379.65	54%	46%

15	4	2.15%	4.04%	100%	51.92	63.21	979.35	3523.59	49%	51%
20	4	2.22%	5.04%	75%	83.96	119.78	8751.53	18001.16	56%	44%
Total	**16**	**1.93%**	**5.52%**	**88%**	**50.10**	**119.78**	**3296.26**	**18001.16**	**52%**	**48%**

Table 2. Four resource type cases

No. of activities	No. of instances	A		B	C				D	
					GA		GAMS			
		Average	Max		Average	Max	Average	Max	Parallel	Serial
<10	4	0.00%	0.00%	100%	17.58	23.90	25.76	26.03	60%	40%
10	4	1.28%	4.87%	100%	31.69	37.90	44.47	43.55	49%	51%
15	4	3.39%	3.64%	100%	55.70	63.02	1013.34	3594.34	61%	39%
20	4	3.63%	5.08%	75%	75.48	91.71	827.83	3288.39	48%	52%
Total	**16**	**2.07%**	**5.08%**	**94%**	**45.11**	**91.71**	**477.85**	**3594.34**	**55%**	**45%**

Table 3. Five resource type cases

No. of activities	No. of instances	A		B	C				D	
					GA		GAMS			
		Average	Max		Average	Max	Average	Max	Parallel	Serial
<10	4	0.18%	0.72%	100%	22.63	29.79	27.58	39.39	46%	54%
10	4	2.74%	4.64%	100%	37.07	51.92	96.65	308.36	50%	50%
15	4	1.52%	2.74%	100%	54.44	64.84	2244.76	7261.56	54%	46%
20	4	3.08%	5.98%	75%	77.18	90.02	1543.07	2695.22	46%	54%
Total	**16**	**1.88%**	**5.98%**	**94%**	**47.83**	**90.02**	**978.02**	**7261.56**	**49%**	**51%**

As explained above, for each combination of number of activities and number of resources 4 examples are generated and in the tables 1, 2 & 3 above the average and maximum deviation and run-time values for each group of 4 examples are presented. At the end row of each table the data presented in the table are summed up. To do this, the average of average values for deviation and run time, less than 5% deviation occurrence and PSGS and SSGS probabilities are presented. Also, in order to better study the obtained results the maximum of maximum values for deviation and run time are calculated.

A general analysis of the above information shows that the maximum deviation ever achieved is 5.98 % and on average 92% of the time proposed procedure reaches solutions with less than 5% deviation from the global optimum solutions. The maximum run time of the proposed procedure is approximately 119 seconds while it is about 18000 seconds for the exact algorithm. Tables 4 and 5 below help to provide a better analysis regarding the performance of the proposed self-adaptive meta-heuristic.

Table 4. Performance details with respect to the number of activities

No. of activities	No. of instances	A	B	C (Average)		D (Average)	
				GA	GAMS	Parallel	Serial
<10	12	0.16%	100%	20.75	25.81	52%	48%
10	12	2.35%	92%	37.09	1190.40	51%	49%
15	12	2.36%	100%	54.02	1412.48	55%	45%
20	12	2.98%	75%	78.87	3707.48	50%	50%
Total	**48**	**1.96%**	**92%**	**47.68**	**1584.04**	**52%**	**48%**

Table 5. Performance details with respect to the number of resources

No. of activities	No. of instances	A	B	C (Average)		D (Average)	
				GA	GAMS	Parallel	Serial
3	16	1.93%	88%	50.10	3296.26	52%	48%
4	16	2.07%	94%	45.11	477.85	55%	45%
5	16	1.88%	94%	47.83	978.02	49%	51%
Total	**48**	**1.96%**	**92%**	**47.68**	**1584.04**	**52%**	**48%**

The information presented by Table 4 shows that the increase in number of activities leads to a slight increase in deviation. It also shows that the increase in the number of activities increases the computation time of both the proposed method and the exact algorithm. The data presented in Table 5 shows that the increase in number of resources will cause the procedure to lean to the SSGS instead of PSGS and it also shows that it slightly increases the computation time. As a result it can be concluded that while the number of activities tremendously increases the computation time of the meta-heuristic, the number of resources only mildly affects the computation time. Since the values for PSGS and SSGS are extremely close, it can be derived that while they have almost a similar performance and we cannot favor one procedure to another with data at hand, the PSGS slightly outperforms the SSGS. On average the procedure has a 1.96% percent deviation from the global optimum solution.

5. Conclusion

In this paper, the work-content based RACP was proposed and a self-adaptive GA was introduced to address the problem. While the problem is relatively close to reality in comparison with the literature reviewed; the problem has not been studied before. The proposed algorithm gives a higher degree of freedom for scheduling the activities and provides the chance of changing the resource profiles of activities and resource allocation. The meta-heuristics utilizes three operators for probing the solution space namely, crossover, mutation and the local search operator. The unique chromosome structure was thoroughly explained and the pseudo-code of the procedure was presented. The review of the computational results showed that the procedure can reach solutions with a little deviation from the global optimum solutions in a fast manner. The results also show that the incorporating the concept of work-content into the problem can lead to a notable reduction of resource demand and consequently their cost. Allowing the preemption of activities, solving the problem in a condition that lateness is permitted with a specific penalty, considering more than one main resource, solving the problem with multiple objectives are only few of many options for expanding the problem and further research.

REFRENCES

[1] **Azimi, F., Fathallahi, F. (2016)**, ***Fuzzy Multi Objective Project Scheduling under Inflationary Conditions***; *Economic Computation and Economic Cybernetics Studies and Research*, 50(3), 337-350; *ASE Publishing:*

[2] **Demeulemeester, E. (1995),** ***Minimizing Resource Availability Costs in Time-Limited Project Networks***. *Management Science*, 41(10), 1590-1598;

[3] **Demeulemeester, E., Vanhoucke, M., & Herroelen, W. (2003),** ***RanGen: A Random Network Generator for Activity-on-the-Node Networks.*** *Journal of Scheduling*, 6(1), 17-38;

[4] **Drexl, A. & Kimms, A. (2001),** ***Optimization Guided Lower and Upper Bounds for the Resource Investment Problem***. *Journal of the Operational Research Society*, 340-351;

[5] **Fündeling, C. U. & Trautmann, N. (2010),** ***A Priority-rule Method for Project Scheduling with Work-content Constraints.*** *European Journal of Operational Research*, 203(3), 568-574;

[6] **Hartmann, S., Briskorn, D. (2022), A*n Updated Survey of Variants and Extensions of the Resource-constrained Project Scheduling Problem*.** *European Journal of Operational Research*, 297(1), 1-14;
[7] **Hsu, C.-C. & Kim, D. S. (2005), *A New Heuristic for the Multi-mode Resource Investment Problem*.** *Journal of the Operational Research Society*, 56(4), 406-413;
[8] **Kolisch, R. (1996), *Serial and Parallel Resource-constrained Project Scheduling Methods Revisited: Theory and Computation*.** *European Journal of Operational Research*, 90(2), 320-333;
[9] **Kolisch, R., Meyer, K., Mohr, R., Schwindt, C. & Urmann, M. (2003), *Ablaufplanung fur die Leitstrukturoptimierung in der Pharmaforschung.*** *Zeitschrift fur Betriebswirtschaft*, 73(8), 825-848;
[10] **Möhring, R. H. (1984), *Minimizing Costs of Resource Requirements in Project Networks Subject to a Fixed Completion Time.*** *Operations Research*, 32(1), 89-120;
[11] **Naber, A. & Kolisch, R. (2014), *MIP Models for Resource-constrained Project Scheduling with Flexible Resource Profiles.*** *European Journal of Operational Research*, 239(2), 335-348;
[12] **Najafi, A.A., N. Esfandiari and F. Azimi, (2009), *Maximizing the Net Present Value of a Project under Inflationary Conditions;*** *Economic Computation and Economic Cybernetics Studies and Research*, 43(4), 199-212;
[13] **Ranjbar, M., Kianfar, F. & Shadrokh, S. (2008), *Solving the Resource Availability Cost Problem in Project Scheduling by Path Relinking and Genetic Algorithm.*** *Applied Mathematics and Computation*, 196(2), 879-888;
[14] **Shahsavar M., A.A Najafi and S.T.A. Niaki (2011), S*tatistical Design of Genetic Algorithms for Combinatorial Optimization Problems,*** *Mathematical Problems in Engineering*, 2011, 2-17;
[15] **Shavandi H., A.A. Najafi, A. Moslehirad (2012), *Fuzzy Project Scheduling with Discounted Cash Flows*;** *Economic Computation and Economic Cybernetics Studies and Research*, 46(1), 219-232;
[16] **Shadrokh, S. & Kianfar, F. (2007), *A Genetic Algorithm for Resource Investment Project Scheduling Problem, Tardiness Permitted with Penalty.*** *European Journal of Operational Research*, 181, 86-101;
[17] **Yamashita, D. S., Armentano, V. A. & Laguna, M. (2006), *Scatter Search for Project Scheduling with Resource Availability Cost.*** *European Journal of Operational Research*, 169(2), 623-637;
[18] **Yamashita, D. S., Armentano, V. A. & Laguna, M. (2007), *Robust Optimization Models for Project Scheduling with Resource Availability Cost.*** *Journal of Scheduling*, 10(1), 67-76.