Presenting a super-heuristic genetic algorithm for investment in project resource

Journal of Industrial Strategic Management
Year, 2023,
Volume 8 (Issue 1),
Pages: 1 - 10.

Noushin Hafezizadeh[a], *Mohammad Mahdi Movahedi*[b*]

a Ph.D. student, Department of Industrial Management, Firuzkoh Branch, Islamic Azad University, Firuzkoh, Iran.
b Associate Professor, Department of Industrial Management, Firuzkoh Branch, Islamic Azad University, Firuzkoh, Iran.

Article history:
Received: 29/05/2021
Received in revised: 08/06/2022
Accepted: 27/06/2022

Abstract
Developing a suitable plan and optimal use of available facilities are considered important factors in today's competitive world. The aim of this research is to provide an innovative genetic algorithm for the problem of investment in project resources. In terms of the purpose, this research is an applied and, in terms of data collection, it is of a mathematical analytical type. According to the positive experiences of using genetic algorithm to solve the problems of the specification in limited resources, this research aims to create two genetic algorithms for a type of allocation problem called investment problem in resources. Genetic algorithm designed was tested on the problems investigated by Mohring representing that the above problems are not complicated enough, because genetic algorithm has obtained optimal solution for the problems rapidly. So, more problems were generated by Progen software through more tests, and, in general, more than 15,000 problems tested by genetic algorithm. Then, by making changes in the above algorithm and using Akpan method and modifying this method, genetic algorithm has been improved. The method developed has also been compared with the previous method during the tests. After setting the parameters on 20 activity problems, the tests were conducted on 10 and 14 activity problems. It represented that new algorithm works more efficiently on these problems. On 30 activity problems in Dergzel and Kims, new and previous genetic algorithms were compared by using multivariate variance analysis and Duncan's test indicating a significant improvement in the answers.

Keywords: Project Scheduling, Genetic Algorithm, Resource Limitation.

1. Introduction

In general, there can be two types of limitations in planning problems. The first one is the limitation of precedence and the other is the limitation of resources, each of which, in turn, can have different forms. In many of the researches conducted and the models created, for the sake of simplicity, one of these two types of limitations has been removed or simplified. In allocating limited resources to project control, the structure of the problem is such that both types of limitations should be considered, and this issue causes this type of planning problem becomes more difficult than other planning problems. (Young et al, 2017) The general state of resource allocation in project control is a number of activities in which the limitation of priority and delay must be observed and the activities can be done in different ways, each method of doing an activity requires its own time and the necessary amount of its own resources. There can be different hypotheses on the sources or limitations of precedence and delay. The problem is to find a plan for allocating the resources to the activities, so that the limitations of resources and activities are met and the function of the intended goal is optimized. (Ziarti et al., 2011).

According to the results obtained by Belzwick, it can be easily shown that the problem of resource allocation in project control is NP-hard. Therefore, for big problems (here maybe even up to 100 activities and 2-3 resource constraints), optimizing algorithms are not able to solve the problem in a reasonable time, and even the methods giving only one acceptable answer are considered useful. For this reason, in recent years, the scope of using innovative sensory methods (heuristic) in solving this type of

* Corresponding author Email address: mmovahedi@gmail.com

problems is increasing. In the last 30 years, a wide range of heuristic methods have been created and tested (Chang, 2014). Most of these methods are based on prioritizing the activities according to different prioritization rules. Due to the complex structure of the problem and the fact that some of the problems project planning are included in the form of complex problems, in recent years, the desire of researchers to use meta-heuristic algorithm has increased. (Mendes et al., 2009)

The resources can be divided into four categories - renewable resources; the resources are considered renewable that, regardless of the length of the project in each of the time periods, are constant and their only limitation is the amount of their use during the period. For example, the source of workers employed in a construction project can be considered as a renewable resource, because if we consider the periods as days, the number of hours available for the source of employed workers is constant in each time period. (Chen et al, 2020). The amount of these resources is measured in terms of resource units in the period. Non-renewable resources; non-renewable resources are the amount of which we have are limited for the entire duration of the project implementation (Chen, 2010). In other word, the amount of these resources will be reduced by using and not renewed in next period or periods. But, in any period of time, as long as we do not exceed the maximum available amount, it can be used by any amount. For example, the total budget considered for the implementation of the project is limited and it can all be consumed in during a period, or certain raw materials are available for the project, but when spending, it is not renewed for the next period or periods. Double limited resources; Resources that are not only limited to be used in each period, but their amount is limited for the entire project. In some cases, the budget can be one of these limitations, so that the total budget is limited for the project, and, at the same time, its daily consumption is also limited. Another example can be the right in computer network, which is limited to all purchased hours, and at the same time, limited by the number of the hours used per day. (Khosh jahan et al., 2013) Semi-renewable resources; In this case, the intended source is limited for use in a number of courses. For example, the workers in a workshop have a limit on the total number of working hours per week, so that, for example, their maximum working hours should not exceed 40 hours per week, and at the same time, this number of working hours can be planned for each worker per week, and the above 40 hours will be renewed every week. Butcher also represented that renewable and non-renewable resources are special cases of this type of resource (Gordo, 2010)

According to the positive experiences of using the cuckoo algorithm in solving the problems of limited resource allocation, in this research, we will create a cuckoo algorithm for a type of resource allocation called the problem of investment in resources and the results will be compared with the genetic algorithm. (Zabihi et al, 2019)

2. Research literature

In 2011, Ziarati et al. studied the use of bee algorithm for RCPSP problems. Three methods have been developed based on the well-known Bee Algorithm, Artificial Bee Colony and Bee Swarm Optimization. The proposed algorithm has repeatedly solved RCPSP problem by exploiting the intelligent behavior of honey bees. Each algorithm has three main stages: initial, update, end. In the initial stage, a set of schedules are randomly generated as the initial population of the algorithm. Then, the initial population is iteratively improved to reach the access condition. The update stage forms the body of the algorithm. Each algorithm uses different types of bees to provide a suitable level of search throughout the desired space while keeping good answers for exploitation. Three new local search methods cooperate in the proposed method to achieve more efficiency. Also, an efficient constrained routing method is described to re-solving impossible solutions. The performance of the proposed algorithm has been compared with a set of art algorithms cases. The simulated results have shown that the bee algorithm is an efficient way to solve RCPSP problems and produces competitive results with other algorithms in this paper. In 2010, Montoya and Torres et al. have used genetic algorithm to solve the problems of project timing by resources limitations. Compared to the usual genetic algorithm, this paper proposes an alternative representation of chromosomes taking advantage of a model of Multi-array object-oriented using programming features of most common languages in designing a decision support system (DSS). This approach has confirmed a set of standard problems and tested on the Internet freely. Numerical results confirm the effectiveness of the presented algorithm and show that this procedure is calculated in less time than the most of past results. In 2009, Mendes et al. presented a genetic algorithm for scheduling resource-constrained project. The proposed chromosome for the problem is based on random keys. This schedule is made by using heuristic priority rule, where the priority of the activities is defined by a genetic algorithm. This heuristic method produces parameterized active schedules. This approach is tested on a set of standard written problems and compared with other approaches. Numerical results have confirmed the effectiveness of the presented algorithm. In 20133, Khoshjahan et al. have addressed RCPSP problem considering the delivery date of each activity. The goal is to minimize the net present value (NPV) of late and early penalty costs. The problem is firstly modeled and then two meta-heuristic algorithms GA and SA are used to solve this NP-hard problem. Finally, the comprehensive calculation results are explained, implemented on a set of examples, and the results are analyzed and discussed. In 2011, Mobini et al. have considered the minimization of the length of a project as a goal. Due to the complexity of this problem, an algorithm called artificial immune algorithm (AIA) inspired by the immune system of the spine has been chosen to solve the problem. In order to confirm the application of this algorithm in terms of the quality of the answer, it has been tested with the answer of different problems in different researches. Computational results have shown that the presented algorithm has comparable results with other benchmark algorithms. In 2009, Damak et al. have paid attention to RCPSP problem with several execution modes for each activity and minimizing the duration of the project. To solve this problem, the differential evolution cycle (DE) algorithm is proposed. It is also focused on the performance of this algorithm to solve the problem through a small amount of time for each activity. Finally, the results are presented through a computational study. The answers were obtained in six categories from the set of problems and compared with the answers of other algorithms representing that this algorithm is better. (Snauwaertet al, 2021).

The problem of investment in resources:
According to Zimmerman's division, one of the cases of leveling the resource is the well-known as investment problem in resources, in which the level of resources is determined at the beginning of the planning horizon and its amount remains constant until the end. On the other hand, a time frame is considered for the completion of the project before which the project must be finished. In this problem, the goal is to determine the level of resources so that the cost of resources is minimized and the project ends before the allowed time. (Kazemipour et al. 2012)

The problem of resources investment in general (MRIP/max):
In MRIP/max problem, the following assumptions are true:
1.Activities cannot be broken.
2.All parameters of the model are definite and there is no randomness in any parameter.
3.The connection between two activities can be with maximum delay or minimum delay.
4.For renewable resources, the amount of resource levels is constant during the project.
5.The duration of the project is predetermined and we call it T.
Another case of the condition 5 is that the project can be completed later than the specified period, but with paying a fine.
Let's assume that the cost of using each unit of renewable resource … during the time ... and the cost of using each unit of non-renewable resource is …, in this case, by considering other, we will have the following general model:

$$\text{Min} \sum_{k\in K^{\rho}} C_k^{\rho} R_k^{\rho} + \sum_{k\in K^{\upsilon}} C_k^{\upsilon} R_k^{\upsilon}$$

$$S_O = 0$$

If J has a minimum tardiness to i:

$$S_j - S_i \geq \sum_{m_i\in M_i} \sum_{m_j\in M_j} x_{im_i} . x_{jm_j} T_{im_i jm_j}^{\min}$$

If J has a maximum tardiness to i:

$$S_j - S_i \leq \sum_{m_i\in M_i} \sum_{m_j\in M_j} x_{im_i} . x_{jm_j} T_{im_i jm_j}^{\max}$$

$$S_{n+1} \leq T$$

$$\sum_{m_j\in M_j} x_{jm_j} = 1$$

$$r_k^{\rho}(S, x, R^{\rho}, R^{\upsilon}, t) \leq R_k^{\rho} \qquad k \in K^{\rho},$$

$$t = 1,2,\ldots,T$$

$$r_k^{\upsilon}(x) \leq R_k^{\upsilon} \qquad k \in K^{\upsilon}$$

$$S \geq 0$$

$$R^{\rho} \geq 0$$

$$R^{\upsilon} \geq 0$$

$$x_{jm_j} \in \{0,1\} \quad j \in \mathrm{V},\ m_j \in M_j$$

There has been no solution to this problem so far [Drexl A., Kimms A. (2001)].

The problem of investing in resources in simplified mode (RIP) when the delay in completing the project has possible penalty (RIPT):
In the simplified case, each activity has only one execution method and is considered as only renewable resources. On the other hand, there are no time limits. This simplified problem was first defined by. He also obtained an algorithm based on the theory of networks to get optimal solution for the problem and tested his algorithm on 16 problems defined.
We remove some assumptions from the MRIP/max problem and obtain the following defined problem:
1. All resources are renewable. That is, there are no non-renewable resources.
2. Each activity has only one execution method
3. There is only a minimum delay in the relation between priority and delay, and its value is equal to the duration of the execution of the activity (of course, its duration can actually be any number, and the above assumption is not a limiting assumption about the time. That is, if the activity *i* is a prerequisite for the activity *j*, then:
4. Delay is allowed, but the fee must be paid for each unit of delay.

We also consider the following definitions:
…: the latest project completion time that will occur if the resources are at their minimum possible.

$T_{\max}$: The latest time of completion of the project will occur if the resources are in their minimum possible.

C_d : Cost of each tardiness unit of delivery time t.

K: A ρ -member set of the renewable resources k ρ } {1,…,

D_i : The duration of implementation of the activity i

C_k : Cost of each unit of kth renewable resource $k \in K$

R_k : Level of kth renewable resource $k \in K$

P_i : A set of pre-requirement activities i

S_i : Start time of ith activity

r_{ik} : Some of the renewable resource K that is used by activity i per time unit, $k \in K$

y_{it} : It takes the value of 1 if activity i starts at time t, and otherwise it takes the value of zero. $t \in \{0,...,T_{\max}\}$

We have $S_i = \sum_{t=0}^{T_{\max}} t \times y_{it}$ and the following model:

$$Min\{\sum_{k=1}^{\rho} C_k R_k + C_d \times Max\{0, S_{n+1} - T\}\} \quad (1)$$

$$\sum_{t=0}^{T_{\max}} t \times y_{it} \geq D_j + \sum_{t=0}^{T_{\max}} t \times y_{jt} \quad , \quad j \in P_i \quad , \quad i = 1,\ldots,n+1 \quad (2)$$

$$\sum_{i=1}^{n} \sum_{u=t-D_i+1}^{t} r_{ik} \times y_{iu} \leq R_k \; , \; t = 0,\ldots,T_{\max,} \quad k \in K = \{1,\ldots,\rho\} \quad (3)$$

$$\sum_{t=0}^{T_{\max}} y_{it} = 1 \quad , \quad i = 1,\ldots,n+1 \quad (4)$$

$$y_{01} = 1 \quad (5)$$

$$y_{it} \in \{0,1\}, \quad i = 1,\ldots,n+1, \; t = 0,\ldots,T_{\max} \quad (6)$$

$$R_k \geq 0, \quad k \in K = \{1,\ldots,\rho\} \quad (7)$$

The objective function (1) minimizes the total resource cost and delay penalty. The constraint (2) considers the precedence and delay constraints of two activities. The constraint (3) controls the limit of resource usage. The constraint (4) guarantees the start of the activity desired at one time only. The set of constraints (6) and (7) determine the range of changes of the variables. We call the above model RIPT.

In the case where there is no time delay, we can refer to Akpan's algorithm [Akpan, E. O. P (1997)]. Akpan's algorithm plans the activities based on the least-latest completion time method. The working principle of this method is that it, firstly, plans the activities in the earliest time and considers the level of resources at maximum in this case. Then, by defining an efficiency criterion for the resources, it tries to reduce a resource that, firstly, decreases the most amount of the cost and secondly, creates the least increase in the cost during the time of the project. It reduces the resource selected by one unit and schedules the activities based on the method of the latest completion time. By the method of the latest completion time, the latest finish time or the latest start time will be obtained based on the maximum time of project completion in each activity. Then, all activities planned before are considered and among them one selected for planning based on a criterion. Here, Eligible Activity Set and Parallel Schedule Generation Scheme can be used to create a program (SGS). Then again, the list of activities whose prerequisites are met is considered and an activity is selected for planning, the algorithm continues till, all activities are planned. The algorithm ends when, by reducing any of the other resources, the project cannot be finished in the desired time.

In fact, this method solves a simple restricted resource specification problem (RCPSP) in each round of the algorithm, and for this purpose, it uses the least-latest finish time method. Yang, Tray & Sum [Yang, Tray & Sum (1995)] showed that this method is suitable among precedence rules. But it is clear that even when the algorithm ends; it may be possible to reduce costs by changing the schedule of activities.

Asmand (1377), As the thesis of his master's course, tested Akpan algorithm with different criteria for selecting a resource that should be reduced and compared different criteria with each other.

Another scheme for chromosomes in genetic algorithm for RIP problem:

Akpan algorithm is presented to solve RIP problems. At first, the above algorithm is modified so that, it can be used to solve RIPT problems, and then a new structure is designed for chromosomes, and Akpan method modified is used in the new genetic algorithm. The results obtained represent that the efficiency of the new algorithm is better than the previous algorithm.

General model:

Scheme of chromosomes:

Each person from the community has a chromosome that, unlike the previous model, has only a list of activities. In this case, a chromosome is defined as follows:

j_u^I: Uth element of the ith person order part of the community, which is the number one of the project activities, which is in the Uth location of the chromosome order part.

Ith person I will be as follows:

$$\mathrm{I} = (j_1^I,..., j_n^I)$$

For each chromosome (person), the order of activities from left to right $(j_1^I,..., j_n^I)$ is considered so that eligible activity set are placed in left side of that activity.

That is, $\{j_0^I,..., j_{n+1}^I\} = \{0,\ldots,n+1\}$ and

$P_{j_a^I} \subseteq \{j_0^I,..., j_{a-1}^I\}$ for a = 1,…, n+1.

In fact, any chromosome (Genotype) can produce only one phonotype.

Using the order of activities in the chromosome to determine the level of resource, Akpan idea [Akpan (1997)] has been used, and (AA), and its algorithm has been changed so that there is a possibility of tardiness for phonotype. We call this algorithm the modified Akpan algorithm and we denote it by MAA.

In AA, the activities are firstly scheduled in their earliest time regardless of resource constraints. Then the resource with the least mean consumption is selected and a unit is reduced and a resource constrained allocation problem is solved. If the solution does not exceed the delivery time, the above operation will be repeated and otherwise the unit reduced from the desired resource will be added again and the resource with the next least mean consumption will be selected and the above operation continues until any units can be subtracted from any source

To create a chromosome (list of activities) at the determination j_a^I phase when ,…, j_{a-1}^I, j_0^I are clear, consider the EAS= $\{u \mid P_u \subset \{j_0^I,...,j_{a-1}^I\}\}$, and suppose n(EAS) is the number of EAS activities. Then use the following two steps [Hartmann (1998)].

j_0^I a=1, N = N(EAS), EAS = {U | $P_u = \{0\}$ },

0= primary step

Main Step: If n= 0, stop; otherwise select one of the activities inside the EAS and name it I, then put:

$j_a^I = i$,

EAS = {u| $u \notin \{j_0^I,\dots,j_a^I\}$, $P_u \subset \{j_0^I,..., j_a^I\}$},

N = N(EAS),

a=a+1

And repeat the main step.

Determining the level of resources:

We use MAA to determine the resource level. MAA is similar to AA, except that first any time each unit is reduced from the desired resource, if the finish time of the project exceeds delivery time of the project, again all resources candidate to decrease. Secondly, the mean consumption is calculated base on the finish time of the project if the project finish time is larger than the delivery time.

We call the total consumption of the resource K as TU_k and obtain it as follows:

$$TU_k = \sum_{\forall i} r_{ik} * D_i$$

Consider the set of all resources RES and take the following steps:

For chromosome I, schedule any activity in the earliest time with the order in the desired chromosome.

Consider the level of all resources for chromosome I equal to the minimum amount required for phonotype of the earliest implementation time of the activities. UF_{cu}, get the amount of mismatch of chromosome I.

Reduce a unit the resource level that has the least value.

$$\left(\frac{TU_k}{S_{n+1} * R_k}\right)$$

That is, if:

$$\underset{k \in RES}{\text{Min}} \, (TU_k \,/\, (S_{n+1} \times R_k)) = TU_v \,/\, (S_{n+1} \times R_v)$$

Then reduce a unit of resource capacity. We'll have:

$$R_v = R_v - 1$$

Depending on the level of resources and using the SSS method, get activities starting activities. If $\bar{S}_1,..., \bar{S}_{n+1}$ is the start time of the activities, UF_{new}, calculate the mismatch of chromosome I to this level of resources.

If $UF_{cu} < UF_{new}$, remove the resource v from the RES and consider::

$$R_v = R_v + 1$$

If RES ≠ Ø, stop, otherwise go step 3. If so, $UF_{new} \le UF_{cu}$ consider $UF_{cu} = UF_{new}$ and go to step 6.

If $S_{n+1} < \bar{S}_{n+1}$, consider RES set as the sets of all resources and go to step 3.

Table1: Combination of different parameters

P_{mu}	P_{cr}	POP	Av. Dev. %	Max. Dev.%	RIPT Optimal %	RIP Optimal %
0/1	0/5	10	0/28	2/9	49	92/2
0/1	0/5	20	0/30	3/1	48/3	88/2
0/1	1	10	0/24	2/5	54/3	95/5
0/1	1	20	0/28	3/0	51/1	90/1
0/2	0/5	10	0/29	2/9	49/2	89/3
0/2	0/5	20	0/30	3/0	48/5	89
0/2	1	10	0/25	2/6	49/2	91/1
0/2	1	20	0/28	2/8	47/8	90/2

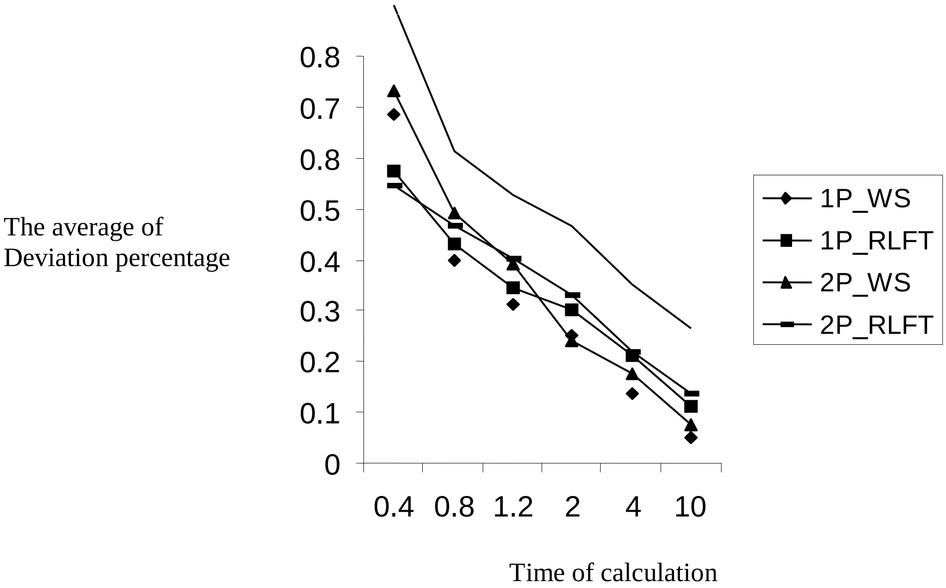

Figure1: The average value of the deviation from the optimal solution for RIPT

Here, too, several chromosomes may result in a solution in the phonotype (ie a few genotypes give a phonotype) and the relationship of the chromosomes and the phonotype from them is n to 1.

Testing the genetic algorithm and comparing the results:

Consider the same problem of 20 activities. By considering the delivery time as the coefficients of earliest project finish time, ignoring resource limitations and considering 6 different modes for the above coefficient…, the number of problems under investigation is 6*20= 120.

For 20- activity problems, in order to set the parameters, we have considered the values of the parameters of GA algorithm as follows:

$$، \{05/0 ،1/0 ،2/0\} \in P_{se}$$

$$\{1/0 ،2/0 \} . P_{cr} \in \{5/0 ،1\} و P_{mu} \in$$

A complete factorial design is considered for all values of the parameters, such that We will have 2*2*3*2*120= 2160 problems to be solved by genetic algorithm. To set GA parameters including P, SSS method has been used and WS weighting method to select the activities from within EAS. The average and maximum deviation from the optimal solution and the percentage of problems led to optimal solution have also been calculated. Some results for running the program in 2 seconds are shown in Table 1. For the statistical analysis of deviation percentage from the optimal solution, we need to know the type of deviation distribution function. Although this random value does not necessarily have a normal distribution, but, because we consider the average deviation of 120 problems, we can confidently assume that this value has a normal distribution based on the central limit theorem. The chi-square hypothesis test with the first type error 0.05 also accepted the hypothesis that the data have a normal distribution. In the statistical analysis of GA parameters, multi-factor variance analysis method was used. Also, by using Duncan's test, we get the following classification: for POP…, for *P*; and for *P*; so that … is the indicator of being better and … much being better. Because, for *P*, the value of 0.1 is better than other values in all cases. In Table 1, there is no column assigned to *P*. This point is important that, as the genetic algorithm of Chapter 7, here also, if the above problems are considered as RIP, the percentage of problems reaching to optimal value will be 86.2 to 92.2. The last column of Table 1 shows the percentage of problems that reach the optimal solution when RIP problem is considered. To test the intersection effect and selection method from EAS, we have considered two intersection methods 1P and2P and two selection methods WS and RLFT and use POP = 10 obtained from the above test results. An experiment with a complete factorial design and 2*2*120= 480 problems is considered. The average value of the deviation from the optimal solution is considered in the times of 0.4, 0.8, 2, 1, 2, 4, 10 seconds and its results are shown in Figure 8. The analysis of the variance represents a significant difference between intersection methods and also between selection methods.

According to the Figure 1, RLFT method works better for time periods less than 1.2 seconds, and WS is more efficient for the times more than 1.2 seconds. This result was expected, because WS tries to give a uniform probability distribution for the selection of all members of the society, and naturally, in the early generations, it is possible to have chromosomes with high mismatches as well as chromosomes with low mismatches, while in RLFT method, from the beginning, we are looking for the chromosomes with few mismatches.

But over time, when WS is given enough time, it will focus more on the right chromosomes, and because WS finds the right chromosomes in the entire space of the answer, the performance will be better than RLF, because RLFT focuses more.

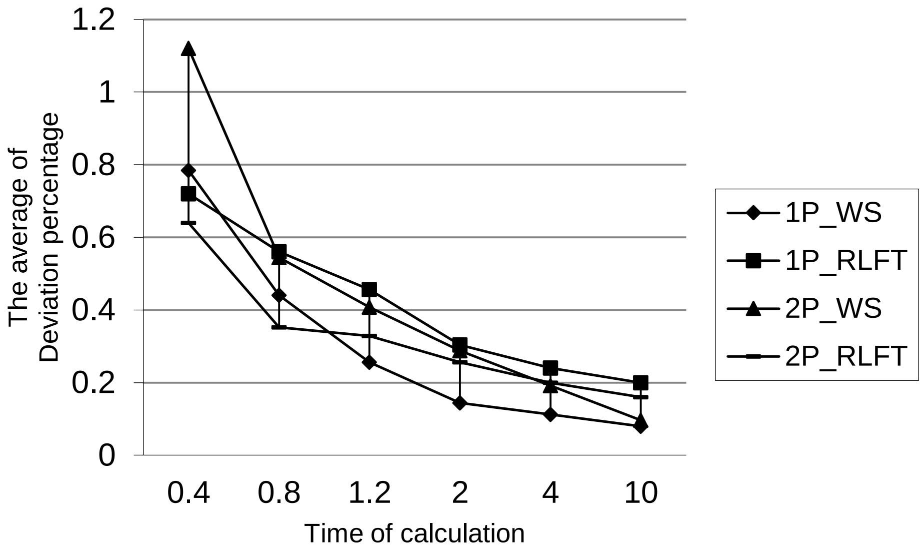

Figure 2: Average value of deviation from the optimal solution for RIP

It locates itself on the appropriate chromosomes according to LFT criteria, which are not necessarily the best answers. On the other hand, Duncan's test indicates that:
1P << 2P or two-point intersection works better than one point.

The above problems are also tested for the case where the tardiness penalty is large. In this case, we actually have a RIP issue. The results can be seen in Figure 9. Variance analysis does not show a significant difference between the intersection methods after 0.8 seconds. But, there is a significant difference between the selection methods. Here too, the results are similar to the RIPT problem.

For the problems 10 and 14, an activity has been proposed that 20 of them with 10 activities and 20 other with 14 activities. All of them have 4 sources. New genetic algorithm, according to the parameters set for 20 activity problems and considering *T*= , … were tested. As a result, 2*20*6=240 problems were generated and examined both as RIP and RIPT (480 problems in total). These problems have been tested with the new genetic algorithm with the following parameters:

…

The tables 14 and 15, respectively, represent that the test results for RIPT and RIP, if the test duration is 0.4 seconds. These tables also show the results of tests on 20 activity problems with 0.4 seconds. By comparing the tables 1 with 2 and 3, it can be seen that the efficiency of the new genetic algorithm is higher.

Since Mohring's problems are very simple, testing the new genetic algorithm on them has been dismissed. Also, by considering the parameters as POP=10 … and WS for 30 activity problems, the test was also done on these problems. The choice of WS method is because the tests are done in 2 seconds for each problem and according to the above results, we know that WS works better than RLFT in the case of more than 1.2 second.

The results of the tests on the problems with 30 activities are given in Table 4. To compare the new genetic algorithm with the previous algorithm, we come to the conclusion that the new method performs significantly better than the previous method by using variance analysis of several variables. Figure 2 shows the result of Duncan's test (Statgraph software output).

3. Conclusion

In this research, 2 genetic algorithms were developed for the problem of investment in resources in the case where the delay in project delivery is allowed but with a cost (called RIPT problem) and the tests represented that these methods have good efficiency. Previous studies have only considered the case where delays in project delivery are not allowed (RIP),and therefore, the problem under consideration is more general. After preliminary studies and the examination of resource specification problems in project control, a special case of investment problem in resources is addressed through project control. The initial form of this problem was firstly investigated in 1984 by Mohring12.
Designed genetic algorithm was tested on the problems investigated by Mohring and indicated that the above problems are not complicated enough, because the genetic algorithm obtained the optimal solution for these problems in a very short time. So, more problems were generated by Progen 13 software for more tests, and in general, more than 15,000 problems were tested by genetic algorithm. Then, by making changes in the above algorithm, using Akpan method and modifying this method, genetic algorithm has been improved. Improved method has also been compared with the previous method during the tests.

According to the first method, each chromosome has two parts. We called the first part as order part or list of activities and the second part as resource level or list of resources. SSS and PSS schedules were used to convert a chromosome to a program. It was also proved that RIPT problem has at least one active optimal program.

Table 2: Results of GA on RIPT for the problems with 10, 14 and 20 activities in 0.4 seconds

Number of activities	Mean deviation percentage	Maximum deviation percentage	The percentage of optimal solutions
10	0/04	0/25	50/50
14	0/05	0/29	43/03
20	0/55	3/95	38/21

Table 3: Results of GA on RIP for problems with 10, 14 and 20 activities in 0.4 seconds

Number of activities	Mean deviation percentage	Maximum deviation percentage	The percentage of optimal solutions
10	0/00	0/00	100/00
14	0/04	3/98	97/80
20	0/63	7/20	69/70

Table 4: The average of upper limit improvement percentage for 30 activity problems

n = 30		$\theta = 1$	$\theta = 1.1$	$\theta = 2.1$	$\theta = 3.1$	$\theta = 4.1$	$\theta = 5.1$
NC = 1.5	RF = 0.25	34.60	40.56	42.46	43.36	43.86	44.12
	RF = 0.5	32.25	41.54	46.60	50.36	53.56	55.84
	RF = .075	36.27	42.95	47.43	51.23	54.27	56.95
	RF = 1	38.20	43.68	48.30	52.00	55.13	57.50
NC = 1.8	RF = 0.25	29.68	36.99	40.04	41.11	41.45	41.60
	RF = 0.5	30.26	39.00	44.17	48.56	51.08	53.56
	RF = .075	33.53	39.79	44.51	48.09	51.07	53.78
	RF = 1	29.69	37.36	42.05	46.40	49.83	52.52
NC = 2.1	RF = 0.25	24.79	34.70	38.30	38.92	39.18	39.40
	RF = 0.5	27.88	36.66	42.49	46.75	49.79	52.52
	RF = .075	26.50	35.15	40.41	44.56	48.14	50.90
	RF = 1	29.80	37.50	42.71	46.63	49.49	52.16

```
Multiple Range Tests for Per. Imp. by GA

--------------------------------------------------------------------------------
Method: ۹۵,۰ percent LSD
GA              Count      LS Mean            Homogeneous Groups
--------------------------------------------------------------------------------
Old GA          ۷۲         ۴۰,۳۴۴۲            X
New GA          ۷۲         ۴۸,۴۶۹۶             X
--------------------------------------------------------------------------------
Contrast                                  Difference           +/-  Limits
--------------------------------------------------------------------------------
Old GA - New GA                           *-۸,۱۲۵۴۲            ۰,۸۹۶۱۹۵
--------------------------------------------------------------------------------
* denotes a statistically significant difference.
```

Figure2: Duncan's test for comparing two genetic algorithms

Since, SSS scheme generates active schedules, the use of SSS method can lead to the optimal solution. Four methods were used to select the activities within EAS, among which WS method was designed so that, when searching the solution space, the probability of chromosomes appearing is more uniform. In addition to Hartmann's intersection methods, another two-point intersection method was developed and proved to produce acceptable solutions. After designing a mutation using a local search algorithm, the tests were done on the above algorithm. In order to conduct the tests, firstly, 20 activity problems were created by ProGen software, their optimal solutions were obtained and the parameters of the genetic algorithm were set by them. Then, other tests were performed on the problems of 10 and 14 activities, all produced by ProGen, and the optimal solution of all of them was also obtained by using EM method. By the parameters set on the problems of 20 activities, good efficiency of the algorithm was evident. Testing Mohring's problems showed that the above problems are not complicated enough, because genetic algorithm obtained the optimal solution for these problems in a very short time, even in some cases of the first generation. Other tests were conducted on 30 activity problems by Dragzel and Kims and the results represented that the above genetic algorithm performs significantly better than column generation and Lagrange methods.

Then, by changing the genetic algorithm chromosome under study, a new genetic algorithm was presented, in which the chromosomes consist only of the list of activities. To determine the resource capacity as well, Akpan algorithm with the changes made on it, was used for RIPT problem. After adjusting the parameters on 20-activity problems, the tests were performed on 10 and 14 activity problems, indicating that the new algorithm works more efficiently on these problems. About 30 activity problems of Dergzel and Kims new and previous genetic algorithm were compared by using the analysis of Multivariable variance and Duncan's test, resulting that there was a significant improvement in the answers.

The topic of project planning is very broad and many researchers are currently engaged in this field. Due to the complexity of the problem, there is still a long way to reach comprehensive and practical methods. On the other hand, the importance of this issue in practice and many potential savings that can be created are great driving force to research in this field. The goal, eventually, is to get a comprehensive project planning softwar.

References

Chen, R., Liang, C., Gu, D., & Zhao, H. (2020). A competence-time-quality scheduling model of multi-skilled staff for it project portfolio. Computers & Industrial Engineering, 139, 106183.

Chen, R., et al., ‘Using novel particle swarm optimization scheme to solve resource-constrained scheduling problem in PSPLIB’, *Expert systems with applications*, (2010), Vol.37, pp. 1899-1910.

Cheng, J., Fowler, J., Kempf, K., Mason, S., ‘Multi-mode resource-constrained project scheduling Problems with non-preemptive activity splitting’, *Computer & Operations Research*, (2014), Vol. 53, pp. 275-287.

Cordeau, J., Laporte, G., Pasin, F. and Ropke, S., ‘Scheduling technicians and tasks in a telecommunications company’, *Journal of Scheduling*, (2010), Vol. 13, pp. 393–409.

Damak N., Jarboui B., Siarry P., Loukil T., 2009, Differential evolution for solving multi-mode resource-constrained project scheduling problems, Computers &OperationsResearch36(2009)2653—2659.

Kazemipoor, H., Tavakkoli-Moghaddam, R., Shahnazari-Shahrezaei, P.,‘Solving a mixed-integer linear programming model for a multi-skilled project scheduling problem by simulated annealing’, *Management Science Letters*, (2012), Vol. 2, pp. 681–688.

Khoshjahan Yazdan, Najafi Amir Abbas, Afshar-Nadjafi Behrouz, 2013, Resource constrained project scheduling problem with discounted earliness–tardiness penalties: Mathematical modeling and solving procedure, Computers & Industrial Engineering 66 (2013) 293–300.

Mendes J.J.M., Gonçalves J.F., Resende M.G.C., 2009, A random key based genetic algorithm for the resource constrained project scheduling problem, Computers & Operations Research 36 (2009) 92 – 109.

Mobini Mahdi, Mobini Zahra, Rabbani Masoud, 2011, An Artificial Immune Algorithm for the project scheduling problem under resource constraints, Applied Soft Computing 11 (2011) 1975–1982.

Montoya-Torres Jairo R., Gutierrez-Franco Edgar, Pirachican-Mayorga Carolina, 2010, Project scheduling with limited resources using a genetic algorithm, International Journal of Project Management 28 (2010) 619–628.

Snauwaert, J., & Vanhoucke, M. (2021) A new algorithm for resource-constrained project scheduling with breadth and depth of skills. European Journal of Operational Research, 292 (1), 43–59.

Young, K. D., Feydy, T., & Schutt, A. (2017) Constraint programming applied to themulti-skill project scheduling problem. In International conference on principles and practice of constraint programming (pp. 308–317), Springer.

Zabihi, S., Kahag, M. R., Maghsoudlou, H., & Afshar-Nadjafi, B. (2019) Multi-objective teaching-learning-based meta-heuristic algorithms to solve multi-skilled project scheduling problem. Computers & Industrial Engineering, 136, 195–211.

Ziarati Koorush, Akbari Reza, Zeighami Vahid, 2011, on the performance of bee algorithms for resource-constrained project scheduling problem, Applied Soft Computing 11 (2011) 3720–3733.

Zhu, L., Lin, J., Li, Y.-Y., & Wang, Z. J. (2021)4. A decomposition-based multi-objective genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem, In Knowledge-based systems. (p. 107099).