Project Portfolio Selection with the Maximization of Net Present Value

Mostafa Nikkhahnasab[a,*], Amir Abbas Najafi[b]

[a] MSc, Faculty of Industrial Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

[b] Assistant Professor, Faculty of Industrial Engineering, K.N. Toosi University of Technology, Tehran, Iran

Received 12 August, 2011; Revised 13 March, 2012; Accepted 27 June, 2012

Abstract

The aim of this study is to present an efficient solving method for the project portfolio selection problem. The objective is to maximize the net present value (NPV) of the project portfolio. The problem is first modeled mathematically. Then, two metaheuristics, the genetic algorithm and simulated annealing, are applied to solve this NP-hard problem. Finally, a comprehensive computational experiment is performed on a set of instances. The results of the computational experiment show that the genetic algorithm performs better than the simulated annealing algorithm.
Keywords: Projects selection; Project scheduling; Discounted cash flows; Resource constraint.

1. Introduction

The continual extension of knowledge and the need for resources that nowadays are valuable and limited are issues which should to be included in decision making more carefully than the past. Project selection and scheduling is amongst the problems that find portfolio of projects beneficial and satisfactory for the effective use of resources. In many organizations, a set of projects is offered and the decision maker selects a portfolio of them. The selection can include one or more projects or even all the projects. In a competitive environment, the importance of having an efficient portfolio, as the success of a company or organization, is considered for using resources in the most appropriate way, because they invested in the selected project portfolio. This is more critical in research and development and information technology projects.

Early studies used dynamic or integer linear programming techniques to support decision making about project portfolio selection (Asher, 1962; Beged-Dov, 1965; Hess 1962). Later, some studies extended these models and took account of factors that occur in practice, thus they had a more realistic perspective. For example, Badri et al. (2001) used a goal programming model for information system project selection, and Gabriel et al. (2006) prepared a multi objective integer optimization model with distributions of costs probability. More recently, new studies were done on project selection and scheduling. Carazo et al. (2010) introduced a comprehensive model for the portfolio of several objectives, and shou et al. (2010) presented the auction algorithm for project portfolio selection and scheduling to maximize the NPV. Rafiee and Kianfar (2011) also offered a scenario tree approach for multi-period project selection problems using the real option valuation method. Last but not least, Gutjahr et al. (2011) proposed the multi-objective decision analysis for competence-oriented project portfolio selection.

The project selection methods are quite systematic. Projects are evaluated one by one and then are ranked based on their values according to a set of predetermined criteria (Henriksen and Traynor,1999; Linton et al., 2002; Meade and Presley, 2002). In processes involving decision-making about investing in R&D projects, using the net present value (NPV) is the accepted criterion (Nakamura and Tasuji, 2004). Doerner et al. (2006) showed that project selection problem is a non-deterministic polynomial-time hard (NP-hard) problem, which has been on the rise in recent years and could be solved by metaheuristics. Examples of metaheuristics are evolutionary algorithms (Medaglia et al., 2007) and ant colony algorithms (Doerner et al., 2006).

A review of the project portfolio selection studies revealed that the issue of project scheduling was not taken into consideration in a majority of them (Coffin and Taylor, 1996). Yet, a few studies such as Archer and Ghasemzadeh (1999) and Carazo et al. (2010) included scheduling into their models, but their work was done on only a single project. From the perspective of project selection and portfolio scheduling, if we consider only selection aspects and eliminate scheduling from

*Corresponding author E-mail: mostafanic@yahoo.com

equations, we cannot make efficient use of resources. Therefore, to tackle the problem of simultaneous selection and scheduling, projects need to be prepared with more flexibility with respect to the use of resources and facilities. Recent studies conducted by Shou et al. (2010) and Gutjahr et al. (2008), therefore, considered new models for project portfolio selection and scheduling. In this study, we have considered selection and scheduling of multi-projects simultaneously, with the resources being dependent on the projects' scheduling.

Multiple projects scheduling has interactions with the quality of project portfolio decision. Chen and Askin (2009) proposed an implicit numerical algorithm for a single-project resource constrained scheduling problem (RCPSP) and stated if all activities of the all selected projects are considered simultaneously, it can be claimed that the algorithm is consistent with reality, because the problem of scheduling a portfolio is in fact a multi-project scheduling problem. Furthermore, considering the scheduling of multi projects separately neglects the interdependencies and synergy among the multiple projects and prevents effective resource allocation (Kurtulus and Davis, 1982). As mentioned before, project portfolio selection is an NP-hard problem (Demeulemeester and Herroelen, 2002). If we solve these problems with exact methods, we achieve illogical solution times and then heuristics are appropriate for these problems such as a resource constrained multi project scheduling problem (RCMPSP). Heuristics can easily integrate themselves with project portfolio selection problems and also have fast performance.

The rest of the paper is organized as follows. In section two, the problem is defined and modeled. Two metaheuristics are then proposed in section three to solve the model. Section four is devoted to tune the parameters of the metaheuristics. The computational experiment and the analysis of the results come in section five. Finally, conclusions are given in section six.

2. The Description of the Mathematical Model

A set of N candidate projects is given. Each project has a network of activities and precedence relations. Project i consist of ni activities and activity ni is dummy and represents the project completion. There is also a set of K resources and each of the resources has a certain capacity. This is regardless of the dependencies between the projects, but the resources can be shared between the projects. Candidate projects are considered based on their net present value and finally a subset of them is selected. It is assumed that cash inflow is achievable only when a project is finished. A goal is to find a project portfolio such that the net present value of the portfolio maximizes. The scheduling of the project will impact the project selection; therefore, we encounter a project portfolio selection and scheduling problem.
The notations used are as follows:

N: the number of candidate projects.
n_i: the number of activities in project i (i=1,2,…,N).
T: the upper bound of portfolio completion time.
K: the number of required resources.
d_{ij}: the duration of activity j of project i.
P_{ij}: the set of immediate predecessors for activity j of project i.
R_k: the availability of the k^{th} resource type in each period (k=1,2,…,K).
r_{ijk}: the per unit time usage of resource k required for activity j of project i.
$NPV_i(t)$: the NPV of project i if the project ends in period t.
To formulate the problem, let us define two sets of decision variables as follows:
Y_i: a binary variable which equals to one if project i is selected in the portfolio and zero otherwise.
X_{ijt}: a binary variable which equals to one if activity j of project i is completed in period t and zero otherwise.
Now, the problem can be formulated in this way:

$$Maximize \quad Z=\sum_{i=1}^{N}\sum_{t=0}^{T}\mathrm{NPV_i(t).X_{in_i}}_{t} \qquad (1)$$

subject to:

$$\sum_{i=1}^{N}\sum_{j=1}^{n_j}\sum_{l=t}^{t+d_{ij}-1} r_{ijk}X_{ijl} \le R_k ; \qquad (2)$$
$$t=0,1,...,T\ , \quad k=1,2,...,K$$

$$\sum_{t=0}^{T} tX_{iht} \le \sum_{t=0}^{T} tX_{ijt}+d_{ij} \ ; \qquad (3)$$
$$i=1,2,...,N\ ,\ j=1,2,...,n_i\ , \quad \forall h\in P_{ij}$$

$$\sum_{t=0}^{T} X_{ijt}=Y_i ; \qquad (4)$$
$$i=1,2,...,N\ ,\ j=1,2,...,n_i$$

$$X_{ijt}=\{0,1\} ; \qquad (5)$$
$$i=1,2,...,N\ ,\ j=1,2,...,n_i\ , \quad t=0,1,...,T$$

$$Y_i=\{0,1\} ; \qquad i=1,2,...,N \qquad (6)$$

The objective function (1) maximizes the net present value of the project portfolio. Constraint (2) enforces the resource constraints over time. Inequality (3) ensures the precedence relations between the activities. Equation (4) states that every activity must be completed only once if its project is selected. Finally, constraints (5) and (6) denote the domain of the variables.

3. Metaheuristic Algorithms

The two metaheuristic algorithms selected, i.e. simulated annealing and genetic algorithm, are briefly described in this section.

3.1. Simulated annealing

Simulated annealing (SA) is a well-known local search metaheuristic. It has been used to solve many scheduling problems in studies like Sadeghieh (2006), Low (2005) and Mika et al. (2005). The main idea behind the proposed SA algorithm is to have a number of iterations where in every iteration of the algorithm, there is a single random pair exchange in the sequence. We evaluate the value of new exchanges and the value of sequences before new exchanges. If the exchange improves the objective function, then it accepts the exchange and the new sequences are preserved. But, if the objective function is not improved, then it is only allowed to accept the exchange with some small probability p. As the number of iterations increases, the probability p for which the algorithm is allowed to accept an exchange that did not improve the objective function is reduced exponentially. This reduction in the probability is usually expressed as a function of the start temperature (T1) that is reduced by a cooling factor to reach a final temperature. This technique of reducing the probability of accepting non-improving exchanges has proven to be very useful in escaping local optimums during the course of the search for the global optimum.
The following is the algorithmic description of the SA:

```
Begin the algorithm
Set T1 to the initial temperature value
Create a random sequence of projects
Compute the value of the objective function
Let this value be the best solution
While T1 is greater than the final temperature value
  Repeat n (the number of activities) times
  Set L1 to the value of the best objective function
  Create a new sequence of projects
      For this sequence
        Pick two random positions j and k
        Do exchange with j and k
        Compute the objective function after the exchange
     End for
  Set L2 to the value of the objective function after the
swapping
    If L2 is greater than or equal to L1, then
        Accept the swap
        Replace this sequence with the related sequence of
L1 as the best solution
        Keep this value of objective function as the best
value
    Else
    If L2 is less than L1, then
         Generate a random number between (0, 1)
         If the random number is less than p, then
            Accept the swap
            Replace this sequence with the L1 related
sequence as the best solution
            Keep this value of objective function as the best
value
        End if
   End if
  where
```

d is equal to $\left|\frac{L2-L1}{L1}\right|$

p is equal to $e^{\frac{d}{T1}}$

```
  End Repeat
 Set T1 to T1*cooling factor value
End While
End algorithm
```

3.2. Genetic Algorithm

Genetic algorithm (GA) is based on the mechanisms of biological evolution and natural genetics. In general, GAs work as follows: a population of individuals is initialized, where each individual represents a potential solution to the problem. The quality of each solution is evaluated using a fitness function. A selection process is applied during each iteration of a GA in order to form a new population. New solutions are created by combining two existing solutions (crossover) and by applying a unary neighborhood operator with a small probability (mutation). This procedure is repeated until convergence is reached. The best solution found is expected to be a near-optimum solution.

3.2.1. Crossover

In this research, we use two approaches for crossover operator. The first approach is crossover between projects and the other approach is crossover into projects. In addition, a single-point crossover is used for both.

In the first crossover, after determining the random number and point of crossover, before the crossover point all numbers of each parent chromosome will be transferred directly to the corresponding section in the offspring chromosomes. After the crossover point, if the number of under checking gene does not exist in the former genes of the offspring's chromosome, with observing the precedence relations, that number is transmitted to the corresponding gene in the offspring chromosomes. Otherwise, the first normal number that does not exist in the offspring chromosomes is devoted to the under studying gene. The chromosomes will be justified. Figure 1 depicts an example of the first crossover operation.

Chromosome 1	1	3	8	2	5	7	6	9	4	11	10	12	13
Chromosome 2	1	2	8	6	7	9	10	11	3	5	4	12	13
Offspring 1	1	3	8	2	5	7	6	9	4	10	11	12	13
Offspring 2	1	2	8	6	7	9	10	11	3	4	5	15	13

Fig. 1. An example of the crossover operation

In the second crossover, the crossover point, since we want to combine a project between the two chromosomes, we have a number of candidate projects to be determined. For example, consider the example of Figure1. Suppose the number "2" is considered as the crossover point. So we want to combine the second project between the two chromosomes. In the example, the first and third projects will be transferred directly to the offspring's chromosomes. By transferring the chromosomes' numbers from a parent to the offspring's chromosome, we are advancing from the beginning of a project that is selected respectively. If the number that is considered in other positions of offspring's chromosome does not exist, that number will be transferred but if it exists, the smallest number that can be found as precedence relations will be transferred to the offspring chromosome. Finally, checking the sequence of numbers in the chromosome is necessary.

3.2.2. Mutation

This operator modifies a chromosome by a small random change to generate a new individual. The main objective of mutation is to add some diversity by introducing more genetic material into the population in order to avoid being trapped in a local optimum. In this research, we randomly select two genes of the chromosome and move their values in order to generate a feasible chromosome.

The steps of the GA are as follows:

Begin the algorithm
Initialize a population of solutions randomly
Repeat the following steps for Generation number times
 Compute the value of objective function for each of the chromosomes
 Select chromosomes of the new population by using the ROLETTE WHEEL method
 Repeat the following steps, population size times
  Generate a random number between 0 and 1 (r1)
   If r1< crossover probability
   First crossover
    Randomly choose two different compatible individuals (parents) to crossover
    Select compatible segments in the two parents
    Do the first crossover
    Save the new sequences
   Second crossover
   Choose randomly one point between the projects of each sequence
    Do the crossover
End if
Generate a random number between 0 and 1 (r2)
  If r2< mutation probability
   Randomly choose a segment to mutation
   Do the mutation
End if
Copy the chromosome in the new generation
End repeat
Store the best solution from the population as the final solution
End the algorithm

4. Tuning the Parameters of the Metaheuristics

In this study, the Taguchi method is employed to tune the parameters of the metaheuristics. In the Taguchi method, orthogonal arrays are used to study a large number of decision variables with a small number of experiments. Taguchi creates a transformation of the repetition data to another value called the measure of variation. The transformation is the signal-to-noise (S/N) ratio which explains why this type of parameter design is called robust design. In the Taguchi approach, the objective functions are categorized into the three groups of "the smaller-the better," "the larger-the-better," and "the nominal- value-is-expected", with each having a particular formula for the (S/N) ratio. The most common performance measure including the mean flow time used in the literature to compare all the algorithms is the average of the relative deviation index ($\overline{RDI}$) that is defined by:

$$\overline{RDI}=\frac{A\,lg_{sol}-Min_{sol}}{Max_{sol}-Min_{sol}}*100 \qquad (7)$$

Where the A lg_{sol} is the average of the solutions obtained by a given algorithm, the Max_{sol} is the average of the best solutions obtained among all algorithms and the

Min_{sol} is the average of the worst solutions obtained in each iteration. For each configuration of the algorithm, the response $\overline{RDI}$ is considered in the Taguchi orthogonal design. Since the aim of this paper is to minimize the $\overline{RDI}$, "the smaller-the better" type is considered for the S/N.

$$S/N= -10*\log \left[\frac{1}{n}\sum_{i=1}^{n} y_i^2\right] \quad (8)$$

where the y_i refers to the value of a response.

As follows, the parameters along with their levels are first introduced. Then the proper scheme of the Taguchi method is selected. Next, the results are analyzed through the analysis of variance (ANOVA). Finally, the best combination of the parameters is selected for the tuned SA and GA. There are four parameters (factors) that may affect the performances of the proposed algorithms. The factors and their levels are presented in Tables 1 and 2.

Table 1
Factors and their levels for the SA

Factors	Levels	
	1	2
Initial temperature	500	1000
Final temperature	100	300
Cooling factor	0.95	0.98
Iteration (Repeat)	n	2n

Table 2
Factors and their levels for the GA

Factors	Levels	
	1	2
Population size	50	100
Crossover probability	0.6	0.8
Mutation probability	0.2	0.3
Generation number	2n	4n

where, 'n' is the number of activities.

The selected orthogonal array L8 is shown in Tables 3 and 4, where the control factors are assigned to the columns of the matrix and the corresponding integers indicate the levels of the factors.

Table3
The orthogonal array L8 for the parameters of the SA

Trial	Initial temperature	Final temperature	Cooling factor	Iteration (Repeat)
1	1	1	1	1
2	1	1	1	2
3	1	2	2	1
4	1	2	2	2
5	2	1	2	1
6	2	1	2	2
7	2	2	1	1
8	2	2	1	2

Table 4
The orthogonal array L8 for the parameters of the GA

Trial	Population size	Crossover probability	Mutation probability	Generation number
1	1	1	1	1
2	1	1	1	2
3	1	2	2	1
4	1	2	2	2
5	2	1	2	1
6	2	1	2	2
7	2	2	1	1
8	2	2	1	2

The experiments are carried out for a set of problems. Each trial is experimented with regard to five instances to yield more reliable information (with each instance being tackled five times.) Hence, there are 25 results for each trial to perform the statistical analyses. The results are analyzed by the response $\overline{RDI}$ for which the S/N is obtained. To explore the relative significance of individual factors in terms of their main effects on the response, the analysis of variance (ANOVA) is conducted.

Table 5
The results of an ANOVA for the S/N ratio for the SA

Factors	DF	SS	MS	F
Initial temperature	1	64031.174	64031.174	5.425
Final temperature	1	64030.107	64030.107	5.424
Cooling factor	1	64035.46	64035.46	5.425
Iteration (Repeat)	1	63870.95	63870.95	5.411
Error	3	35408.38		
Total	7	291376.071		

Table 6
The results of an ANOVA for the average of the relative deviation index for the SA

Factors	DF	SS	MS	F
Initial temperature	1	3.766	3.766	17.22
Final temperature	1	3.77	3.77	17.24
Cooling factor	1	3.748	3.748	17.14
Iteration (Repeat)	1	3.746	3.746	17.13
Error	3	0.656		
Total	7	15.686		

Based on the results in Tables 5 and 6, all factors have significant effects on the $\overline{RDI}$ and the S/N at 95% confidence level. The optimum values of the SA parameters are obtained by considering both the S/N ratio and the $\overline{RDI}$. The results are as follows:

Initial temperature=1000, Final temperature=300, Cooling factor=0.95 and Iteration (Repeat) =n.

Table 7
The results of an ANOVA for the S/N ratio for the GA

Factors	DF	SS	MS	F
Population size	1	49902.72	49902.72	20.28
Crossover probability	1	49852.18	49852.18	20.26
Mutation probability	1	49905.88	49905.88	20.29
Generation number	1	49902.72	49902.72	20.28
Error	3	7378.632		
Total	7	206942.132		

Table 8
The results of an ANOVA for the average of the relative deviation index for the GA

Factors	DF	SS	MS	F
Population size	1	3.65	3.65	23.81
Crossover probability	1	3.81	3.81	24.85
Mutation probability	1	3.81	3.81	24.85
Generation number	1	3.81	3.81	24.85
Error	3	0.4599		
Total	7	15.5399		

In addition, based on the results in Tables 7 and 8, the optimum values of the GA parameters are obtained and the results are as follows:

Population size=100, Crossover probability=0.6, Mutation probability=0.3 and Generation number=2n.

5. Computational Results

In this section, we present the results of a computational experiment on the implementations of the two algorithms for the problem. In order to compare the proposed algorithms, using the ProGen software a problem set was generated that included problems with 5 projects with activities in different sizes (5, 10, 30 and 90) and 1, 2 and 3 resources. In addition, some new data were required for our problems according to its mathematical model which were produced randomly. The proposed algorithms were coded in the R2009a software and the experiments were conducted on a corei5 CPU with 4 GB RAM. We compared the performance of the metaheuristics using three measures: average percentage error (Error), standard deviation (Std), and the percentage of times that the best solutions are obtained (NOB). The percentage error is defined as the absolute deviation from the best solution known. There were 12 combinations for different values of the number of resources and the number of activities. Twenty five replicates were generated for each combination. The results of the computational experiment are summarized in Table 9.

Table 9
Computational results

No. of Resources	No. of Activities	SA			GA		
		NOB	Error	Std	NOB	Error	Std
1	5	100	6.33	0	100	0	0
	10	20	2.61	13.35	80	0.037	0.72
	30	40	0.97	1.48	60	0.046	1.13
	90	40	25.48	0.4888	60	0.015	0.174
2	5	40	1.01	4.58	100	0	0
	10	60	5.01	8.38	80	0.205	3.108
	30	80	0.103	2.84	100	0	0
	90	70	45.71	4.8	80	0.091	1.464
3	5	80	0.43	7.28	100	0	0
	10	40	10.05	9.64	60	0.039	0.595
	30	40	8.6	0.888	40	0.06	0.718
	90	50	40.86	5.22	60	0.15	2.71

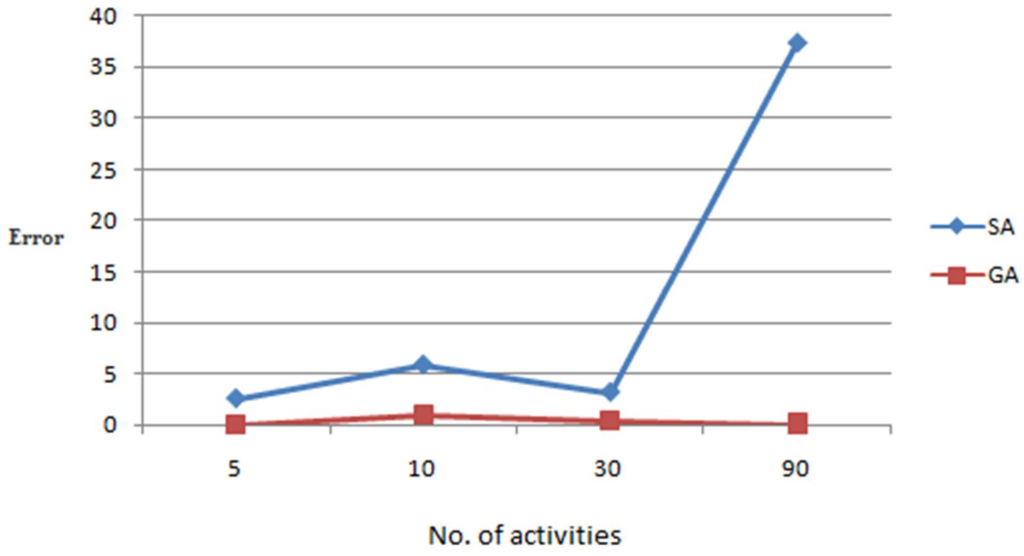

Fig. 2. The overall error of different values of the activities

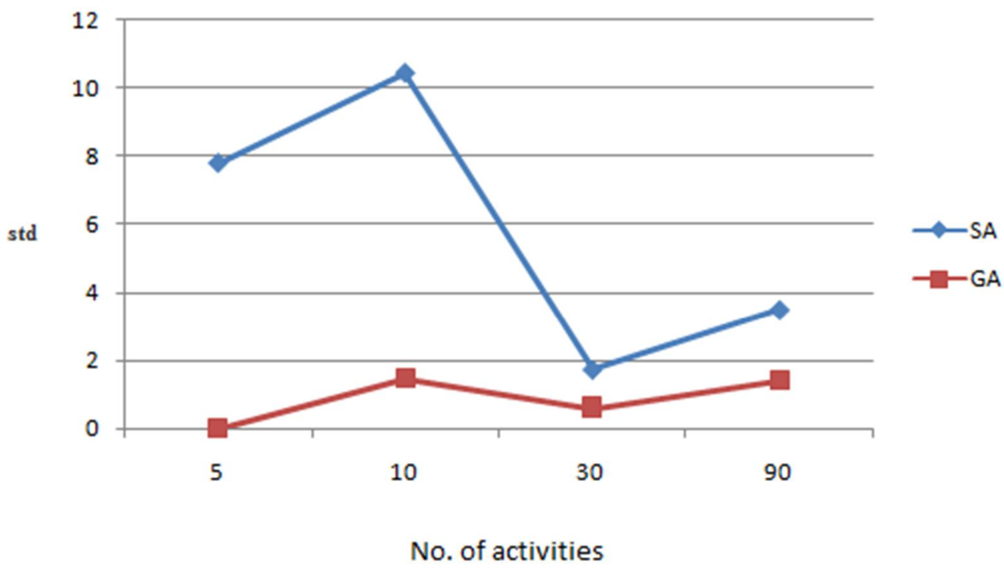

Fig. 3. The overall standard deviation of different values of the activities

In addition, Figures 2 and 3 depict the computational results which reveal that the GA performs better than the SA.

These results were statistically tested by using a t-test, too. The following hypothesis testing was conducted for the test problems.

- **Null hypothesis:** The average error of the GA equals the average error of the SA.
- **Alternative hypothesis:** The average error of the GA is smaller than the average error of the SA.

The null hypothesis was rejected at a 99% significance level. This implies that the average error of the GA is statistically smaller than that of the SA.

6. Conclusions

In this study, we considered the project portfolio selection problem to maximize the net present value of projects portfolio. Two metaheuristic algorithms including simulated annealing (SA) and genetic algorithm (GA) were proposed. To improve the efficiency of the proposed GA and SA, their parameters were fine tuned using the Taguchi method. Then the metaheuristics were compared on the basis of a computational experiment performed on a set of instances. The results of the computational experiment showed that the GA performs better than the SA. One possible direction of research would be to address the problem with respect to two objective functions such as maximizing the NPV and minimizing the completion time of project portfolio.

7. References

[1] Archer, N.P.and Ghasemzadeh, F. (1999). An integrated framework for project portfolio selection. International Journal of Project Management. 17. 207-216.

[2] Asher, D.T. (1962). A linear programming model for the allocation of R and D efforts.IEEE Trans. Eng. Manag. 9, 154–157.

[3] Badri, M.A., Davis, D., Davis, D.(2001). A comprehensive 0-1 goal programming model for project selection. Int. J. Proj.Manag.,19(4):243-252. [doi:10.1016/S0263-786399)00078-2]

[4] Beged-Dov, A.G. (1965). Optimal assignment of R&D projects in a large company using an integer programming model, IEEE Trans. Eng. Manag. 12, 138–142.

[5] Carazo, A.F., Gomez, T., Molina, J., Hernandez-Diaz, A.G., Guerero, F.M., Caballero, R., (2010). Solving a comprehensive model for multiobjective project portfolio selection.Comput.Oper. Res., 37(4):630-639. [doi:10.1016/j. cor.2009.06.012]

[6] Chen, J., Askin, R.G., (2009). Project selection, scheduling and resource allocation with time dependent returns. Eur. J. Oper. Res., 193(1):23-34. [doi:10.1016/j.ejor.2007.10.040]

[7] Coffin, M.A., Taylor, B.W., (1996). Multiple criteria R&D project selection and scheduling using fuzzy logic. Comput.Operat. Res., 23(3):207-220. [doi:10.1016/0305-0548(96)81768-0]

[8] Demeulemeester, E., Herroelen, W., (2002). Project Scheduling:a Research Handbook. Kluwer Academic Publishers, New York, 203 206.

[9] Doerner, K.F., Gutjahr, W.J., .Hartl, R.F., Strauss, C. and Stummer, C. (2006). Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection. European Journal of Operational Research, 171, 830-841.

[10] Gabriel, S.A., Kumar, S., Ordonez, J., Nasserian, A., (2006). A multiobjective optimization model for project selection with probabilistic considerations.Soc.-Econ. Plan. Sci., 40(4):297-313. [doi:10.1016/j.seps.2005.02.002].

[11] Gutjahr, W.J, Katzensteiner, S. Reiter, P. Stummer, C. and Denk, M. (2008). Competence-driven project portfolio selection, scheduling and staff assignment. Central European Journal of Operational Research, 16, 281-306.

[12] Henriksen, A.D., Traynor, A.J.(1999). A practical R&D project-selection scoring tool. IEEE Trans. Eng. Manag.,46(2):158-170. [doi:10.1109/17.759144].

[13] Hess, S.W. (1962). A dynamic programming approach to R and D budgeting and project selection, IRE Trans. Eng. Manag. 9, 170–179.

[14] Kurtulus, I., Davis, E.W., (1982). Multi-project scheduling: categorization of heuristic rule performance. Manag. Sci., 28(2):161-172. [doi:10.1287/mnsc.28.2.161]

[15] Linton, J.D., Walsh, S.T., Morabito, J., (2002). Analysis, ranking and selection of R&D projects in a portfolio. R&D Manag., 32(2):139-148. [doi:10.1111/1467-9310.00246]

[16] Low, C. (2005). Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines. Comput Oper Res 32(8):2013–2025

[17] Meade, L.M., Presley, A., (2002). R&D project selection using the analytic network process. IEEE Trans. Eng. Manag., 49(1):59-66. [doi:10.1109/17.985748]

[18] Medaglia, A.L., Graves, S.B., Ringuest, J.L., (2007). A multiobjective evolutionary approach for linearly constrained project selection under uncertainty. Eur. J. Oper. Res., 179(3):869-894. [doi:10.1016/j.ejor.2005.03.068]

[19] Mika, M., Waligóra, G., Weglarz, J. (2005). Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models. Eur J Oper Res 164(3):639–668

[20] Nakamura, Y., Tsuji, M., (2004). Study on the allocation of R&D investment: method of evaluating and selecting R&D projects for investment. Int. J. Innov. Technol. Manag., 1(1):55-74. [doi:10.1142/S0219877004000027]

[21] Sadegheih A (2006) Scheduling problem using genetic algorithm, simulated annealing and the effects of parameter values on GA performance. Appl Math Model 30(2):147–154

[22] Rafiee, M. and Kianfar, F. (2011). A scenario tree approach to multi-period project selection problem using real-option valuation method.The International Journal of Advanced Manufacturing Technology, 56, 411.