Time Optimization during Software Implementation for Timely Delivery Using Meta-Heuristic Algorithms

Omid Sojoodi Sheijani and Ali Izadi

***Abstract*—Scheduling of software development and implementation projects is one of the most important and challenging issues facing project managers in the highly competitive software industry. In the market, the final output of software is its price. The price of software is the foundation of the quality and application of that software. As a matter of course, innovation in the idea of software and, consequently, the time spent on designing it can determine its value and price. Therefore, proper scheduling, which is ultimately absolutely crucial for the release of a software application, plays a significant role in the success of that software. There are usually software tools for project management that, using algorithms and collecting data from the program and the conditions of its creation, can identify the best time of release for the software release. In this paper, we try to predict the time needed to make software using meta-heuristic algorithms and through rating tasks, schedule them in a way that the software can be operational in the least time. This paper is a graft between software engineering and meta-heuristic algorithms. The results of various researches have confirmed the superiority of the PSO algorithm to solve this problem compared to other algorithms. This algorithm results in answers with 9 repetitions and within a time span of 27.19 seconds.**

***Index Terms*—Software engineering, estimation of production time, meta-heuristic algorithms, scheduling.**

I. Introduction

Project management involves the use of knowledge, skills, tools and techniques to meet project needs. Also, issues such as project scheduling, planning, monitoring, control tasks and risk management are essential to the management of complex systems. Generally, the question of scheduling software projects involves finding and setting up an optimal calendar for software engineers to implement multiple tasks with minimal cost and minimum time. Scheduling problems can be classified considering various factors such as the amount of stages in the job process, the amount of machines present for each stage, different job processing requirements, setup time/cost requirements and the performance measure to be optimized [1]. The issue of the scheduling of the software projects is an NP-hard question with very complex, time-consuming and fatal combined optimization problems. Therefore, automatic scheduling of a software project with high-quality is very useful and absolutely essential for project managers [2]. In recent years, many researchers have turned their attention to 'scheduled production' for software projects using computer algorithms. The purpose of this paper is to minimize the time of the implementation of the software project through using meta-heuristic algorithms.

Knowing the right time to produce software can help in estimating the costs, and progressing in the publishing, releasing and organizing that software. Estimating the time of the production of an application allows the employer to choose the appropriate companies for software development and production, and, from among the companies that participate in the tenders, choose the company whose time and costs are compatible with the estimations of the employer. But scheduling, and announcing it, in software projects requires accurate and precise calculations. In fact, it seems that calculating the lines of software in its programming or the working hours of the programmers is enough to determine the release time and date. But in reality, there are other things involved too. History of software management is interwoven with the history of software development and production. Software was originally written for specific purposes or for setting up hardware. But with the introduction of the concept of the 'object-oriented programming' in 1960, programming based on this approach was welcomed by software development companies, and in the 1970s and 1980s, software development and production process experienced rapid growth and development [3]. There are different methods to control a project, depending on the features and attributes of its activities. In the past, several methods were proposed for predicting the completion time of the project, such as the critical path method, bar chart, project scheduling with repetitive activity, scheduling for limited resource projects, point-based and time-based linear scheduling, and etc. Bar Chart method was established by Henry Gantt in 1920. This method included horizontal bars, each bar indicating an activity and its length representing the time of that activity. In 1957, DuPont proposed the critical path method (CPM). This method was initially designed for chemical industry projects and has proved very effective for when activities have mutual connection and interrelation [4].

Scheduling point-based projects based on the balance line were also presented by the US Navy in 1950. This method was used to estimate and measure the time of the construction of units. In 1975, O'Brien offered a method for scheduling projects with repetitive activities. This method was called the general productivity method. In 1981, Johnson introduced 'linear scheduling' method [5].

Manuscript received June 9, 2019; revised August 7, 2019.

Omid Sojoodi Sheijani is with the Department of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran (e-mail: o_sojoodi@qiau.ac.ir).

Ali Izadi is with the Department of Computer Engineering, Maragheh Branch, Islamic Azad University, Maragheh, Iran (e-mail: izadi@iau-maragheh.ac.ir).

doi: 10.18178/ijmlc.2019.9.5.843

In Ref. [6] a multi-objective mathematical programming model is presented to consider quantitative and qualitative factors and risk to choose appropriate suppliers and allocate the optimal order quantity to them. The problem was solved by NSGA-II, an obtained Pareto solution set was ranked by the TOPSIS algorithm, and then the best possible solution was chosen. Also, in [7], [8] the allocation of a large number of jobs required to be scheduled on multiple and identical machines which run in parallel was proposed. They compared Tabu Search (TS) and Simulated Annealing (SA) performance of the eBPA.

To solve the problem of scheduling a project with limited resources, using innovative methods in researches such as [9] have been thoroughly investigated and scrutinized. Furthermore, super-innovative methods such as 'evolutionary algorithms' have been used in researches [10], [11], ants systems in research [12], the forbidden search method in research [13], [14] and simulated refrigeration in research [15], and have demonstrated good results for well-known benchmarks.

Search-Based Software Engineering seeks to reformulate software engineering problems as search-based optimization problems and applies a variety of metaheuristics based on local and global search to solve them (such as PSO, Genetic Algorithms and etc.). Though the term Search-Based Software Engineering (SBSE) was coined by Harman and Jones in 2001 to cover the application of computational search and optimization across the wide spectrum of software engineering activities [16].

II. Software Project Scheduling

The purpose of software engineering is to develop software systems based on the needs of the users in a scheduled way and with a determined budget. The name "software engineering" was first used in a conference in Germany and was initially accepted as a technique and then as a job definition. The purpose of software engineering is to create engineering principles for software development, and it was created to alleviate and/or overcome some issues such as late delivery, cost overruns, and failure to achieve the initial software goals [17].

Considering the fact that software becomes more complicated day by day, the importance of research to estimate software effort has naturally grown. Exact estimate of the effort and costs needed at the start of the software life cycle is crucial for software companies [18]. If estimating the effort needed for a project are more accurate and more reliable, to the same extent, directing, planning and completion of the project will be more successful [19].

Search based software engineering is one of the approaches to solves those problems using metaheuristics search techniques such as genetic algorithm, simulated annealing, tabu search, etc. [16].

The main purpose of scheduling a software project is to determine its start and end date. Typically, a start and end date is set for each task so that an overall timeframe is determined. The Project Manager cannot start working on scheduling until they have established a work break structure (WBS), list of tasks, effort required for each tasks, and list of available resources.

In project management, when all the deliverables are grouped into smaller deliverable components is called Work Breakdown Structure (WBS) [20]. Establishing a WBS is a part of the project scheduling exercise and this is one of the main ingredients for establishing requirements for funding and resources. Furthermore, during the life of the project, project managers are consistently working on schedule to ensure that project's deadline is achieved. Since this task is implemented at the very start of projects, sometimes it may be hard to determine start and end date due to many reasons such as availability resources, cuts in funding from project sponsors, and many more. Inappropriate scheduling of a project indicates that the project may not be completed within the given time frame and budget. Furthermore, it may also mean that third party resources (equipment, human resources, etc) that were allocated based on the initial start and end date of a task may not be available [21].

Estimating the effort required creating software and estimating the time-span of software projects is one of the most controversial issues in managing software projects. Many software projects have failed solely for the reason that they exhausted their time estimates. Therefore, it should be clearly stated that the realistic estimation of the implementation time of a software project is one of the most fundamental steps in the success of a software project [22].

The primary challenge of project management is to achieve all of the project goals and objectives while honoring the preconceived constraints. The primary constraints are scope time, quality and budget the secondary and more ambitious challenge is to optimize the allocation of necessary inputs and integrate them to meet pre-defined objectives [23].

Schedule is a vital project management attribute in a market driven economy where time to market is critical to success. It takes an important role in the project planning activity. It decides which tasks would be taken up when. Project managers frequently find themselves unable to compromise on schedule while being able to play better with the other two parameters. In order to schedule the project activities, a software project manager needs to do the following [24]:

1) Identify all the tasks to complete the project.
2) Break down large tasks into small ones and determine dependency among them.
3) Estimate effort for each task, and a resource list with availability for each resource (If these are not yet available, it may be possible to create something that looks like a schedule, but it will essentially be a work of fiction).
4) Determine the terminal dates i.e. starting and ending dates.
5) Determine the critical path.

When a project tends towards running out of its time, software engineers are asked (or expected) to work more hours so that the project proceeds according to the predicted path. The point here is that putting the team under pressure does not necessarily guarantee accomplishing the expected results [25].

Here the question of resource constraints enters the scene. In fact, before the work begins, it is necessary to anticipate such issues. Capability level and competence of the programming team is one of the critical factors in scheduling. The best thing to do here is to break or divide the project into smaller parts. The software engineer should be able to break his project into smaller parts and, through scheduling management of these activities, offer the best scheduling [25].

III. Critical Path Method for Scheduling Activities

A network is a graphic or visual diagram of the project that portrays project activities and the relationships between them. Each network or project must have a start event and an end event.

Critical Path method (CPM) is a step-by-step project management technique for process planning that defines critical and non-critical tasks with the goal of preventing time-frame problems. CPM is a heuristic method that is based on the past experience of the project planner for problem solving.

There are several methods for network analysis, one of the most important of which is the CPM, which was developed in 1960, is one of the most important and practical tools in project planning and control. This method involves determining the critical path, critical activities and critical events in the project network, while taking into consideration the earliest possible completion time of the project. In project management and control, when the timing of activities is definite and determined, using classical techniques, such as the critical path, is a good tool for project planning and control [26].

The critical path in the network is the path that has the largest expected set of times. It should be noted that with any delay in the implementation of activities in the critical path, completion of the project according to the initial scheduling will be delayed. In the critical path method, with full use and attention to the duration of time, communications, dependencies and sequence of activities, the earliest and the latest start and end times of each activity are determined accurately through the all-at-once calculation of all activities. The main focus of this method is about calculating the floatation times and the rate of flexibility at the time of the implementation of the activities [4].

In Ref. [27], presented software to determine the probability of finishing the project on deadline. Later, in [28] proposed a new technique based on CPM networks to solve the RCPSP[1] with stochastic activities duration and activities insertion; and subsequently, in [29] presented a procedure that combines the Stochastic Critical Path method with the Envelope Method.

In CPM, simple calculations are used to create the project scheduling program and to assess the criticality of activities through concepts such as floatation and the critical path and an absolute focus on time. CPM gained popularity and became widespread because of its exceeding simplifying of the problem and the low volume of the basic information required. However, many critics have criticized CPM for not taking into account the resources in scheduling calculations. The major drawbacks of CPM in scheduling Linear Repetitive Projects are [30]-[32]: (1) Assuming the rate of resource efficiency rate as constant and stable, which results in its ineffective and inappropriate use. (2) Scheduling based on definite times and not taking into consideration the uncertainties governing the project. (3) Inability to eliminate work interruptions of resources (because in this method the activities are scheduled based on the earliest start time.) (4) Large volume of network calculations in Linear Repetitive Projects with a large number of units and the complexity of the pre-requisite relationships between the activities of successive units of the project. In this paper, we have used the meta-heuristic algorithm to overcome the weaknesses and drawbacks of CPM.

In using CPM, the general approach is that after dividing the project into smaller activities and tasks, we need to find out the interrelations and dependencies between these activities and then look for a way to schedule these activities. If, as in Fig. 1, we consider each activity as a point, these points will be related and connected to each other. For instance, activity D depends on activity A.

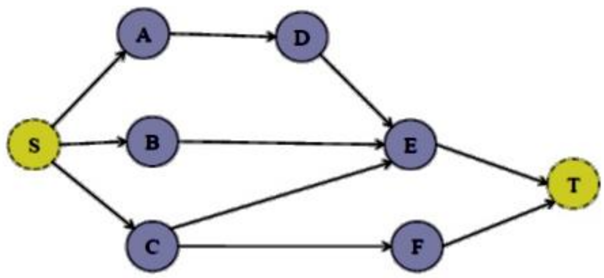


Fig. 1. Magnetization as a function of applied field.

Each of these activities requires a specific time to be performed. They need to be scheduled in a way that the pre-requisite for each activity is done before it starts. The next point is the question of available resources. It is necessary to calculate how much of the work each part of the team can do. In the critical path method, a hierarchy of tasks is presented, by doing which, the final time of the project will be completed in the best possible way [3].

IV. Research Method

In this paper, we created a table of activities by using a series of data obtained from the www.mpsplib.com website, which professionally carries out the task of estimating the project time. The critical method proposes several paths. Using the PSO algorithm, these paths have been investigated and the optimal route has been selected. The activities are as follows. This project consists of 30 small activities that are interconnected (Table I).

These relationships mean that activity 6, for example, is related to activity 2. Or activity 14 is associated with the two activities 9 and 12, meaning after doing these two activities, activity 14 can start work. Each of these activities has a specific time to be carried out. The number of resources for activities is 4. It can be assumed that we have four programming teams to do this, and each of these teams has a

[1] Resource-Constrained Project Scheduling Problem

limited ability to do the work.

Several methods are proposed for carrying out activities using the critical path method. The number of these methods is very high, but using the PSO algorithm we find the best scheduling in these methods. In Table II, we will show the duration of each activity. These times are an hourly basis.

TABLE I: THIRTY ACTIVITIES

Activity	Previous	Activity	Previous	Activity	Previous	Activity	Previous
1	-	**9**	4	**17**	13, 14	**25**	10, 15, 20
2	1	**10**	4	**18**	13	**26**	11
3	1	**11**	2	**19**	8	**27**	7, 8
4	1	**12**	8	**20**	5, 11, 18	**28**	21, 27
5	4	**13**	3	**21**	16	**29**	19
6	2	**14**	9, 12	**22**	16, 17, 18	**30**	6, 24, 25
7	3	**15**	2	**23**	20, 22	**31**	26, 28
8	3	**16**	10	**24**	19, 23	**32**	29, 30, 31

Each of our four programming teams can work the following maximum hours: 12, 4, 13, and 12.

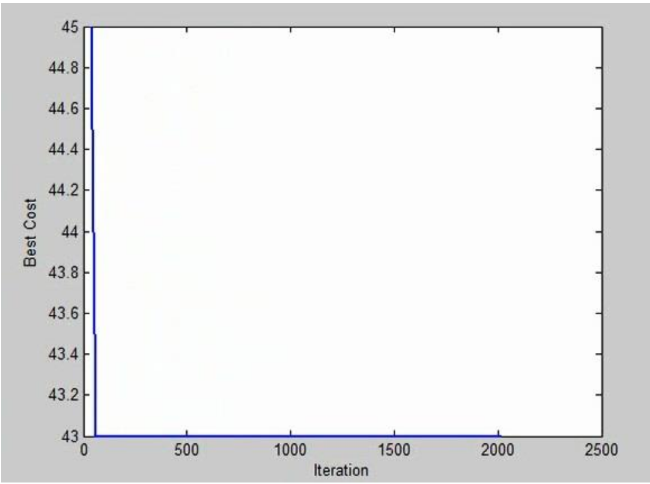


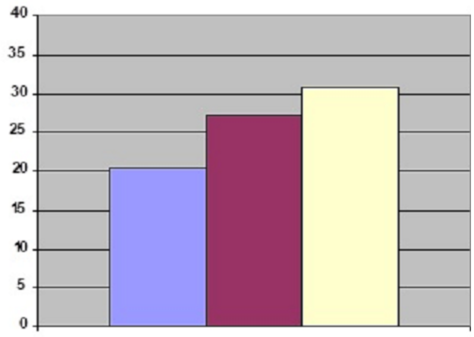
Fig. 2. Repetition diagram and the best cost.

The hypothesis is that the programming team can only focus on a single given task in one day. In this case, different modes for scheduling the tasks emerge.

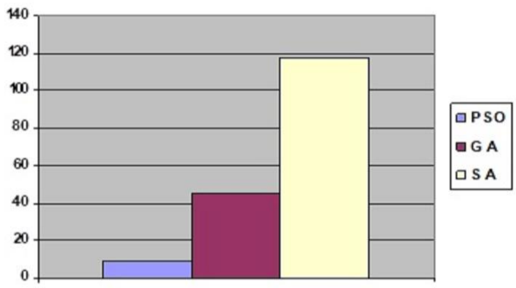
V. RESULTS AND COMPARING THE ALGORITHMS

Using the PSO algorithm, we were able to obtain the optimal values as 43 days. The PSO algorithm achieves this in a very short time. The result is shown in Fig. 2.

Compared to the other algorithms, the PSO (Particle Swarm Optimization) algorithm reaches the optimal amount faster and in shorter time. This optimal amount for the problem is obtained with fewer repetitions compared to the other algorithms. This algorithm is more efficient than other algorithms in solving this problem, due to its high convergence speed, flexibility, and not getting entangled in local optimal values. In the following, Fig. 3 compares the number of repetitions to reach the answer by several algorithms, the best of which is the PSO algorithm.



(a) Comparing runtime



(b) Comparing the number of repetitions

Fig. 3. (a): Comparing run-time, (b): Comparing the number of repetitions.

TABLE II: DURATION OF EACH ACTIVITY

32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
0	2	2	7	3	8	7	3	3	2	7	2	7	3	5	6	10	9	3	6	2	9	7	2	9	5	8	3	6	4	8	0

VI. CONCLUSION

Today, the issue of scheduling plays a key role in all areas of science. Scheduling, in each discipline, has its own principles. Scheduling method is more likely to be of great importance in construction projects. For example, in civil engineering activities, such as roadway construction, the issue of repetitive activities is also considered during the implementation of the project. The difference between scheduling, creating a software and a mass building project of a structure or a road is evident in a few aspects. In the first aspect, it should be emphasized that in programming it is not possible to calculate time fully and totally calculated, for the programming team cannot be expected to perform a precise and given the volume of tasks each day. At the same time, the problems encountered during programming should be taken into consideration. But in civil engineering projects, the work time of the human workforce can be clearly defined in advance, and there is even the capacity to replace human resources. However, in programming, each team should follow its own part and role from the beginning of the work, and the replacement of the human workforce will be difficult. In fact, if we intend to replace the human workforce, we may have to redo programming for that team. Repetitive activities are also of great significance in civil engineering projects, and it raises the issue of scheduling to a whole new level. For example, in a mass building project of a structure, the actions that must be done to build each unit can be quite repetitive, but in the field of programming, a need for innovation and innovativeness in each section is highly expected.

In software design projects, estimating project time is of paramount importance. So far, different scheduling methods have been proposed for this task. In this paper, we tried to offer the optimal scheduling for designing software through combining one of the proposed methods with meta-heuristic algorithms. The results listed in the previous section were obtained using a device with 8 GHz 2.4 core and the Matlab software.

These results indicate the superiority of the PSO algorithm to solve this problem compared to the other algorithms. This algorithm reaches response with 9 repetitions and in the timespan of 27.19 seconds. The problem of finding the best time in the critical path method is one of the complex problems that was optimized in this paper using the PSO algorithm. By using the PSO algorithm, the critical path method will have the ability to compete with other methods for estimating project time. In the future, we can use other algorithms to optimize this problem and compare the results.

REFERENCES

[1] A. Allahverdi, "The third comprehensive survey on scheduling problems with setup times/costs," *European Journal of Operational Research*, vol. 246, pp. 345-378, October 2015.

[2] V. Jagtap and P. Joglekar, "Study of project scheduling and resource allocation using ant colony optimization," in *Proc. the 13th International Conference on Computer Science and Information Technology (ICCIT 2015)*, December 2015, vol. 2.

[3] G. D. Oberlender, *Project Management for Engineering and Construction*, 2nd ed., McGraw-Hill Company, New York, 2000.

[4] S. Lowsley and C. Linnett, *About Time: Delay Analysis in* Construction, RICS Business Services Limited, UK, 2006.

[5] T. Hegazy and W. Menesi, "Critical Path Segments (CPS) scheduling technique," *Journal of Construction Engineering and Management, ASCE*, vol. 136, pp. 1078-1085, October 2010.

[6] A. Nazeri and B. M. Khakzar, "Implementation of meta-heuristic algorithms for supplier selection and evaluation and multi product order allocation," *Journal UMP Social Sciences and Technology Management*, vol. 3, issue 3, pp. 566-577, 2015.

[7] A. O. Adewumi and S. Chetty, "Investigating the enhanced Best Performance algorithm for annual crop planning problem based on economic factors," *PloS One*, vol. 12, no. 8, August 2017.

[8] S. Chetty and O. Adewumi, "A study on the enhanced best performance algorithm for the just-in-time scheduling problem," *Discrete Dynamics in Nature and Society*, vol. 12, January 2015.

[9] R. Kolisch andS. Hartmann, "Experimental investigation of heuristics for resource-constrained project scheduling: An update," *European Journal of Operational Research*, vol. 174, pp. 23-37, October 2006.

[10] J. Alcaraz and C. Maroto, "A robust genetic algorithm for resource allocation in project scheduling," *Annals of Operations Research*, vol. 102, pp. 1-4, February 2001.

[11] Y. Kochetov and A. Stolyar, "Evolutionary local search with variable neighborhood for the resource constrained project scheduling problem," in *Proc. the 3rd International Workshop on Computer Science and Information Technologies*, Russia, April 2003.

[12] D. Merkle, M. Middendorf, and H. Schmeck, "Ant colony optimization for resource-constrained project scheduling," *IEEE Transaction on Evolutionary Computation*, vol. 6, pp. 333-346, November 2002.

[13] K. Nonobe and T. Ibaraki, "Formulation and Tabu search algorithm for the resource constrained project scheduling problem (RCPSP)," *Essays and Surveys in Metaheuristics*, Boston, Kluwer, pp. 557-588, January 2002.

[14] B. Vahdani, S. T. A. Niaki,, and S. Aslanzade, "Production-inventory-routing coordination with capacity and time window constraints for perishable products: Heuristic and meta-heuristic algorithms," *Journal of Cleaner Production*, vol. 161, pp. 598-618, September 2017.

[15] K. Bouleimen and H. Lecocq, "A new efficient simulated annealing algorithm for the resource constrained project scheduling problem and its multiple mode version," *European Journal of Operational Research*, vol. 149, pp. 268-281, September 2003.

[16] M. Harman and B. F. Jones, "Search-based Software Engineering," *Information and Software Technology*, vol. 43, pp. 833-839, 2001.

[17] R. Pressman, *Software Engineering*, 5th ed, MC Graw-Hill, 2001.

[18] X. Hu, M. Yang, J. Cheng, and L. Chen, "A new product development cost estimation method based on the optimal weight combination," in *Proc. Fifth International Conference on Natural Computation (ICNC '09)*, Tianjin, 2009, vol. 2, pp. 329-333.

[19] A. L. I. Oliveira, "Estimation of software project effort with support vector regression," *Neurocomputing*, vol. 69, pp. 1749–1753, August 2006.

[20] W. Herroelen, "Project scheduling — theory and practice," *Production and Operations Management*, vol. 14, pp. 413-432, December 2005.

[21] J. Lewis, "Large limits to software estimation," *ACM Software Engineering Notes*, vol. 26, pp. 54-59, July 2001.

[22] B. Boehm and R. Fairly, "Software Estimation perspective," *IEEE Software*, vol. 17, pp.22-26, November 2000.

[23] J. Smith and B. Kenneth, *Engineering Quality Software*, Elsevier Science Publishers LTD, 1992.

[24] R. Mall, *Fundamentals of Software Engineering*, 4th ed, PHI Learning Private Limited, New Delhi, 2008.

[25] F. Hemmstra, "Software cost estimation," *Information and Software Technology*, vol. 34, pp. 626-704, 1992.

[26] L. J. Krajewski and L. P. Ritzman, *Operations Management: Process and Value Chains*, 7th Edition, Prentice-Hall, New Jersey, 2005.

[27] D.-E. Lee, "Probability of project completion using stochastic project scheduling simulation", *Journal of Construction Engineering and Management*, vol. 131, pp. 310-318, March 2005.

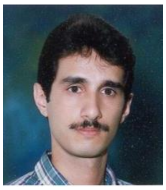
[28] S. Archer, R. L. Armacost, and J. Pet-Armacost, "Effectiveness of Resource buffers for the stochastic task insertion problem," *The Journal of Management and Engineering Integration*, vol. 2, pp. 14-21, Winter 2009.

[29] N. Kokkaew and N. Chiara, "Modeling completion risk using stochastic critical path-envelope method: A BOT highway project application," *Construction Management and Economics*, vol. 28, pp. 1239-1254, December 2010.

[30] P. G. Ipsilandis, "Multi objective linear programming model for scheduling linear repetitive projects," *Journal of Construction Engineering and Management*, vol. 133, pp. 417–424, June 2007.

[31] L. D. Long and A. Ohsato, "A genetic algorithm-based method for scheduling repetitive construction projects," *Journal of Automation in Construction, ASCE 9*, vol. 18, pp 499-511, July 2009.

[32] R. I. Carr and W. L. Meyer, "Planning construction of repetitive building units," *Journal of the Construction Division, ASCE*, vol. 100, pp. 403-412, September 1974.



Omid Sojoodi Sheijani is an assistant professor in artificial intelligence at Islamic Azad University, Qazvin Branch. He received his Ph.D in artificial intelligence from The UPM Malaysia University. His research interests include data mining, machine learning, big data analysis and robotics.

Ali Izadi is an instructor in software engineering at Islamic Azad University, Maragheh Branch. He received his MSc insoftware engineering from The Islamic Azad University, Qazvin Branch and is currently a Ph.D student at Islamic Azad University, Qazvin Branch. His research interests include software engineering, search based software engineering.