Evaluating project’s completion time with Q-learning

Mohammad Abdolshah
Semnan Branch, Islamic Azad University, Semnan, Iran

ABSTRACT

Nowadays project management is a key component in introductory operations management. The educators and the researchers in these areas advocate representing a project as a network and applying the solution approaches for network models to them to assist project managers to monitor their completion. In this paper, we evaluated project’s completion time utilizing the Q-learning algorithm. So the new algorithm is proposed to solve completion time project. Then, we run this algorithm in example network in Matlab. The results showed that the algorithm has achieved the best time to complete the project.

Keywords
completion time, project, Q-learning

1. Introduction

Nowadays project management has been an important issue for industrial organizations because it should maximize resource utilization and minimize cost and time for organizations. Although Time management play more significant role in project management compares to scheduling control, resource management and cost management. But, Research has shown that many projects are not completed on time. In large projects, only approximately 10%_15% of them finishes on time [8, 14]. In fact, major subject of this point is not only their planned schedule, but their budget is more important. The network techniques used to tackle project analysis are Critical Path Method (CPM) and Project Evaluation and Review Technique (PERT) [15]. With CPM a deterministic assessment for activity time was used [16, 17], while with PERT random time assessments were employed [18, 19, 20].

Introductory discussions has begun Expression the basic concepts of activities, durations, and precedence relationships, followed by the development of network representations of a project; earliest and latest start and completion times; slack; and critical path. The problem of determining the completion time of a project has been extensively dealt with in management science/operations research/industrial engineering. The educators and the researchers in these areas advocate representing a project as a network and applying the solution approaches for network models to them to assist project managers to monitor their completion. The purpose of this paper is to Evaluate project’s completion time utilizing the Q-learning algorithm.

2. Related work

One of the most important problems in project management is obtain the total completion time. The main approaches used are the Program Evaluation and Review Technique (PERT) and the Critical Path Method (CPM).Martin in 1965 calculates the completion time using polynomials by approximating task duration density functions. Wolf in 1985 represented assessed using a sound approach to calculate the completion time for practical and managerial purposes, what matters is the criticality of each activity within a PERT network. Many researchers have focused on this issue using different methods. A survey of recent developments can be found in Table 1.

Table 1- recent developments

ID	Researchers	Year	Subject	Result
1	Maria Elena Bruni, Francesca Guerriero, Erika Pinto	2008	Evaluating project completion time in project networks with discrete random activity durations	Deterministic models for project scheduling suffer from the fact that they assume complete information and neglect random influences that occur during project execution. This phenomenon occurs even in the absence of resource constraints and has been the subject of extensive research in the scientific community. This paper: a method for obtaining relevant information about the project make span for scheduling models, with dependent random processing time available in the form of scenarios [1].
2	AlirezaAlagheband, Mohammad Ali Soukhakian	2012	An efficient algorithm for calculating the exact overall time distribution function of a project with uncertain task durations	Calculating the probability distribution function (pdf) for project completion time is caused by structural and statistical dependence between activities. This paper: a method for taking into accounts the structural dependence between activities and provides a generalized algorithm to evaluate the exact PDF for project completion time [2].
3	Peng-Jen Lai, Hsien-ChungWu	2011	Evaluate the fuzzy completion times in the fuzzy flow shop scheduling problems using the virus-evolutionary genetic algorithms	- A computational procedure is proposed to obtain the approximated membership function of fuzzy completion time for each schedule. We plan to minimize the fuzzy make span and total weighted fuzzy completion time. Evolutionary genetic algorithms to search for the best schedules. - MATLAB[3].
4	Cs. Király, M. Garetto,M. Meo,M. AjmoneMarsan, R. Lo Cigno	2005	2. Analytical computation of completion time distributions of short-lived TCP connections 3.	A new technique for the analytical evaluation of distributions (and quintiles) of the completion time of short-lived TCP connections Open multiclass queuing network (OMQN) models of the TCP protocol and computes a discrete approximation, with arbitrary accuracy. - Computationally efficient, its asymptotic complexity is independent of the network topology, of the number of concurrent flows, and of other network parameters [4].
5	Roger Buehler, Johanna Peetz, Dale Griffin	2010	4. Finishing on time: When do predictions influence completion times? 5.	We examined whether and when task completion predictions influence actual completion times. As hypothesized, the prediction manipulation influenced completion times under certain conditions defined by the nature of the target task. Manipulated predictions affected completion times of closed tasks, defined as tasks carried out within a single, continuous session but not of open tasks, defined as tasks requiring multiple work sessions. This implies that task completion predictions help to initiate action, but their impact diminishes over the course of extensive, multi-stage projects [5].
6	Yousry H. Abdelkader	2004	Evaluating project completion times when activity times are Weibull distributed 7.	In this paper, a development of the moments method based on Weibull distribution of activity time is presented. The method provides an accurate estimate of the project completion time compared with other d.f.'sestimates[6].
7	KishanMehrotra, John Chai, Sharma Pillutla	1996	. A study of approximating the moments of the job completion time in PERT networks 9.	We propose an approximation to determine the TCJ which *explicitly* recognizes this dependency. Dependency in networks arises due to commonality of activities among various paths in the network. We develop an approximation which is simple to use and makes use of readily available tables. The activities on the critical paths are divided into an independent portion and a dependent portion. The dependent portion comprises activities common to various critical paths. Order statistics are used in computing the time for the dependent portion of the critical path. - Closer to the simulation results. [7].

3. Q-learning

Machine learning is a growing field with increasing importance. It is utilized when normal algorithms are too difficult and complex to develop and also because it has the ability to recognize complex pattern that otherwise might be overlooked. Machine learning methods can also be used when the task at hand is in a changing environment where adaptability is important in order to overcome new obstacles that might arise [9].

Reinforcement learning is a subset of the machine learning techniques. These techniques learn through trial and error in search of an optimal solution. Q-learning is a simple reinforcement learning algorithm. The main advantage of this algorithm is that it is simple and thus easy to implement; the actual algorithm consists of only one line of code. Chris Watkins developed Q-learning back in 1989 when he combined several previous fields into what today is known as reinforcement learning [10, 11]. An agent is a Supervisor in the project implementation, either a computer or an interface towards humans. To understand Q-learning it is crucial to understand how a problem is represented. It is possible to divide almost all problems into several situations, so called states the state of the environment is the current Specified Activity.

An action is a transition from one state to another, which is what a Supervisor does or can do in specific state. An action generates feedback from the environment. The feedback can be either positive or negative. The feedback is used in the Q-learning algorithm for estimating how good an action is. The term reward is used for positive feedback and negative feedback is called punishment. In every state the algorithm visits it checks the best possible action it can take. It does this by first checking the Q-value of every state it has the possibility to get to in one step. It then takes the maximum of these future values and incorporates them into the current Q-value. When feedback is given, the only value that is updated is the Q-value corresponding to the state action pair that gave the feedback in question. The algorithm for updating Q-values is shown in equation 1, where st and at corresponds to the state and action at a given time [10, 11].

The fact that only one value is updated when feedback is given, gives Q-learning an interesting property. It takes a while for the feedback to propagate backwards through the matrix. The next time a certain state-action pair is evaluated, the value will be updated with regards to the states it can lead to directly. What this means is that in order for feedback to propagate backwards to the beginning, the same path need to be taken the same number of times the paths are long, each new iteration on the path propagating the effects backwards one step. If the algorithm always took the best path it knew of it is very likely it will end up in a local maximum where it takes the same path all the time. Even though the path it takes is the best it knows of that does not mean there is no better path. To counter this it is necessary to let the algorithm sometimes take a new path and not the best one, in order for it to hopefully stumble upon a better solution.

If there is one, and only one, stable solution, that is which action is the best in each state, the Q-learning algorithm will converge towards that solution. Because of the back-propagation property of Q-learning, this however requires a large amount of visits to every possible state action pair [13].

There are two parameters to the algorithm, the learning rate "a" and the discount factor "g". These both affect the behavior of the algorithm in different ways. The learning rate decides how much future actions should be taken into regard. A learning rate of zero would make the agent not learn anything new and a learning rate of one would mean that only the most recent information is considered. The discount factor determines how much future feedback is taken into account by the algorithm. If the discount factor is zero, no regard is taken to what happens in the future. Conversely, when the discount factor approaches one a long term high reward will be prioritized.

A problem with the Q-learning algorithm is how exploration of new paths is done. There are a couple of ways to execute this task. The simplest is sometimes to simply use a random action and hope it provides a better solution. Another way is to have a phase of exploration in the beginning of the agent's lifetime where all possible state-action pairs are explored.

The agent used in the implementation uses an exploration phase. This means that after the exploration

Phase, no further exploration is done. This could pose a problem since an agent trained against one opponent might not learn the most efficient behavior against another opponent because there is no real exploration being done. We have chosen to ignore this risk, trusting that the Q-learning algorithm will cope with it. During the exploration phase each state-action pair is tested until it has a decent Q-value. Because of back-propagation this means that a state-action pair will be tested multiple times. This ensures that a reliable result is achieved.

$$Q(s_t,a_t) \leftarrow \underbrace{Q(s_t,a_t)}_{\text{old Q-value}} + \underbrace{\alpha}_{\text{learning rate}} \times \Big[\overbrace{\underbrace{r_{t+1}}_{\text{feedback}} + \underbrace{\gamma}_{\text{discount factor}} \underbrace{\max_a Q(s_{t+1},a)}_{\text{max future Q-value}}}^{\text{expected discounted feedback}} - \overbrace{Q(s_t,a_t)}^{\text{old Q-value}} \Big] \quad (1)$$

4. Evaluate project's completion time using Q-learning

In Algorithm, each activity is considered a state. We begin from first state as first activity, by Breadth first search algorithm; agent can go one activity to another activity that between activities to another activity should be edge. R-matrix is adjacent matrix graph of the project. Matrix R contains the state that is connected by a path straight to the other state. In this algorithm, the matrix R is the reward. The matrix R is a direct route between the time-consuming states is element of matrix, otherwise, "-in" is valued.

Parameter gamma is the ratio of the learning rate, which determines the learning of Q-learning algorithm. Gamma can be considered from 0.01to0.100. If learning rate is higher, value of Q-table is increasing. In small projects can be considered 1 to give infinitive answer. But in large projects are generally considered to bathe bigger the amount of time it takes to get an answer.

Episode parameter specifies the number of repetitions and since this parameter is one of the parameters in the algorithm complexity is essential to be careful in the choice of its value. Q-Learning project's completion time algorithm pseudo-code is given in figure 1:

1. Create a project using the R-matrix graph
2. Set the gamma value
3.Initialise Q with zero
4.Initialise Q1 with zero
5.For (each episode) do
Initialize s
Determine the value of y using ran perm function
Determine the state using y(1)
Find R(state) Greater equality of zero
q (state,x1)<- R(state,x1)+gamma*qMax(x1)
State <- x1
Until s is terminal

6. Time Project <- q (1, j)

Figure 1- Q-Learning project's completion time algorithm pseudo-code

5. Numerical Example

In this section, a hypothetical project problem is presented to demonstrate the computational process of Q-learning proposed above. Suppose there is a project network, as Figure 2, with the set of node $N = \{1, 2, 3, 4, 5, 6, 7, 8\}$, the activity time for each activity as shown in figure. All of the durations are in hours.

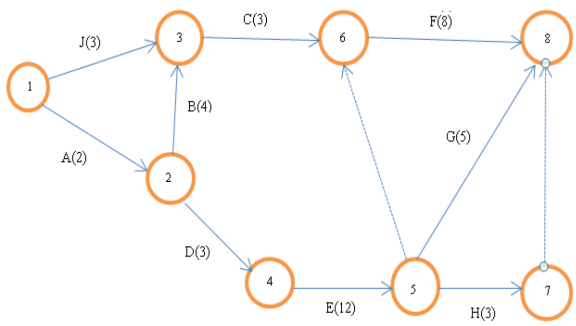

Figure 2- project network

After running the algorithm on the project network in Figure 2, Q-matrix is obtained. In fact, output of Q-algorithm is called Q-matrix so that ith as the answer. The matrix Q is obtained which is given below.

q =

0	25	14	0	0	0	0	0
0	0	15	23	0	0	0	0
0	0	0	0	0	11	0	0
0	0	0	0	20	0	0	0
0	0	0	0	0	8	3	0
0	0	0	0	0	0	0	8
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

In Table 2 are gave the project time and the number of iterations of the Q-algorithm. Time project comes from the maximum amount of first row of the Q-matrix. Number of repetitions is the number of loop iterations Q-algorithm.

Table 2- TP and episode

TP	25
episode	1069

The matrix Q is given in graph form in Figure3. As it can be seen that the maximum value of25as the time during every activity of the project after the project has been obtained.

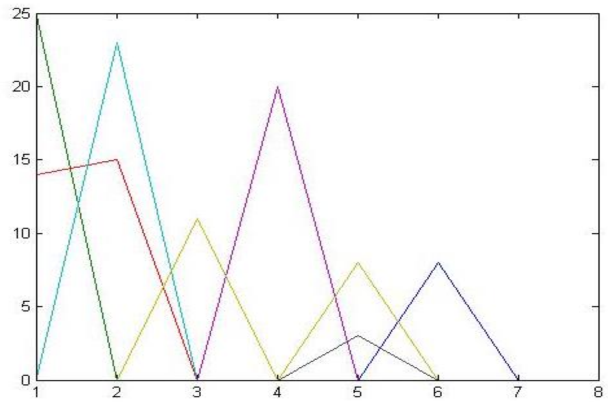

Figure 3 –plot (q)

6. Conclusion

We evaluated project's completion time utilizing the Q-learning algorithm. So, the new algorithm is proposed to solve completion time project. In Algorithm, each activity is considered a state. We begin from first state as first activity, by Breadth first search algorithm; agent can go one activity to another activity that between activities to another activity should be edge. R-matrix is adjacent matrix graph of the project. Matrix R contains the state that is connected by a path straight to the other state. In this algorithm, the matrix R is the reward. The matrix R is a direct route between the time-consuming state is element of matrix, Otherwise, "-inf" is valued. Parameter gamma is the ratio of the learning rate, which determines the learning of Q-learning algorithm. Gamma can be considered from 0.01to0.100. If learning rate is higher, value of Q-table is increasing. In small projects can be considered 1 to give a definitive answer. But in large projects are generally considered to bathe bigger the amount of time it takes to get an answer. Episode parameter specifies the number of repetitions and since this parameter is one of the parameters in the algorithm complexity is essential to be careful in the choice of its value. Then, we run this algorithm in example network in Matlab. Time complexity of the algorithmic linear and is defined by number of episodes. Complexity Space simply is project the network input and Q-matrix in the algorithm that is $2 * n^2$. The results show that the algorithm has achieved the best time to complete the project and ineffective.

7. Future work

For future work, we have implemented a Q-Learning algorithm for fuzzy number as time project, of course because fuzzy numbers are Triangular or trapezoidal, the structure of the input matrix and state matrix and Q-matrix should make a difference that is described in this article. Cell array structure for fuzzy number scan is used. Cell array is a multi dimensional array, each element of which is an array. Fuzzy numbers that are1 x4 or1 x 3 matrixes; each of them will be element of the array.

References

1. Maria Elena Bruni,FrancescaGuerriero,Erika Pinto(2008), Evaluating project completion time in project networks with discrete random activity durations ,Computers & Operations Research ,Volume 36, Issue 10, October 2009, Pages 2716–2722.
2. AlirezaAlagheband,Mohammad Ali Soukhakian(2012), An efficient algorithm for calculating the exact overall time distribution function of a project with uncertaintask durations, Indian Journal of Science and Technology, Vol. 5 No. 9 (Sep. 2012) ISSN: 0974-6846.
3. Peng-Jen Lai, Hsien-Chung Wu(2011), Evaluate the fuzzy completion times in the fuzzy flow shop scheduling problems using the virus-evolutionary genetic algorithms, Applied Soft ComputingVolume 11, Issue 8, December 2011, Pages 4540–4550.
4. Cs. Király, M. Garetto,M. Meo, M. AjmoneMarsan, R. Lo Cigno(2005), Analytical computation of completion time distributions of short-lived TCP connections,Performance EvaluationVolume 59, Issues 2–3, February 2005, Pages 179–197.
5. Roger Buehler, Johanna Peetz, Dale Griffin(2010), Finishing on time: When do predictions influence completion times?, Organizational Behavior and Human Decision ProcessesVolume 111, Issue 1, January 2010, Pages 23–32.
6. Yousry H. Abdelkader(2004), Evaluating project completion times when activity times are Weibull distributed, European Journal of Operational Research, Volume 157, Issue 3, 16 September 2004, Pages 704–715.
7. KishanMehrotra, John Chai, Sharma Pillutla(1996), A study of approximating the moments of the job completion time in PERT networks, Journal of Operations Management, Volume 14, Issue 3, September 1996, Pages 277–289.
8. J. Coppendale, Manage risk in product and process development and avoid unpleasant surprises, Eng. Manage. J. 5 (1) (1995) 35–38.
9. Mitchell, Tom M. 2006 (July). The Discipline of Machine Learning. http://www.cs.cmu.edu/ tom/pubs/MachineLearning.pdf.
10. Sutton, Richard S., &Barto, Andrew G. 1998. Reinforcement Learning: An introduction. MITPRess, Cambridge, MA.
11. Watkins, Christopher John Cornish Hellaby. 1989. Learning from Delayed Rewards. Ph.D.thesis, Cambridge.
12. Hassoun, Mohamad H. 1995. Fundamentals of Artificial Neural Networks.MIT Press.
13. Watkins, Christopher John Cornish Hellaby, & Dayan, Peter. 1992. Technical Note: Q-learning.Machine Learning.
14. P. Chatzoglou, L. Macaulay, A review of existing models for project planning and estimation and the need for a new approach, Int. J. Proj. Manage. 14 (1996) 173–183.
15. Taylor, B. W., Introduction to Management Science, New York, Prentice-Hall, Inc., 1996.
16. Chen, Y. L., Rinks, D. and Tang, K., Critical path in an activity network with time constraints,European Journal of Operational Research, Vol.100, pp.122-133, 1997.
17. Singh, G. and Zinder, Y., Worst-case performance of critical path type algorithms, International Transactions in Operational Research, Vol.7, pp.383-399, 2000.
18. Levner, E. and Kats, V., A parametric critical path problem and an application for cyclic scheduling, Discrete Applied mathematics, Vol.87, pp.149-158, 1998.
19. Lootsma, F. A., Stochastic and fuzzy PERT, European Journal of Operational Reasearch, Vol.43, pp.174-183, 1989.
20. Soroush, H., Risk taking in Stochastic PERT networks, European Journal of Operational Research, Vol.67, pp.221-241, 1993.