Indian Journal of Science and Technology Vol. 5 No. 9 (Sep. 2012) ISSN: 0974- 6846

An efficient algorithm for calculating the exact overall time distribution function of a project with uncertain task durations

Alireza Alagheband[1,*] and Mohammad Ali Soukhakian[2]
[1]*Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran*
[2]*Department of Management, Shiraz University, Shiraz, Iran*
[1]alaghebandalireza@yahoo.com[*], [2]mha_soukhakian@yahoo.com

Abstract

For stochastic PERT networks the main difficulty in calculating the probability distribution function (pdf) for project completion time is caused by structural and statistical dependence between activities. This paper presents a method for taking into account the structural dependence between activities and provides a generalized algorithm to evaluate the exact PDF for project completion time. The proposed procedure provides an exact pdf for project completion time when the duration times of activities are discrete. It can be applied for PERT networks with statistical dependence as well as structural dependent relationship between activities and can be applied for PERT networks with discrete or continues distribution. It has been tested using simple activity network models with different structural dependence between activities

Keywords: PERT, pdf, Project completion time.

Introduction

One of the most important problems in the analysis of PERT networks is the determination of the distribution function for project completion times. When the duration times of the activities of a project are random variables, the completion time of the project is also a random variable, with a distribution function that is a complex function of the distribution functions for each activity.

For networks with a special structure, the distribution function for project completion time can be obtained by reducing the network to a single, equivalent activity starting at an initial node (1) and ending at a terminal node (N). Assuming structural independence of the durations of the network activities, the reduction is possible through repeated application of two well-known operations: convolution and greatest. Convolution and greatest operations both involve the combining of probability distributions.

If the PERT network satisfies the conditions necessary for the direct use of convolution and greatest operations, then the network is termed reducible; otherwise, it is termed irreducible. If the network is reducible to a single equivalent activity (1, N), then it is termed as completely reducible. If the network is completely reducible, the analytical form of the distribution function of the project completion time can be determined. However, irreducibility of the network prevents such analytical determination.

In conventional PERT network models it is assumed that different paths are structurally independent. This is not true for irreducible networks, because in irreducible networks at least two paths share one or more common activities, like "Wheatstone bridge".

This paper presents a method for taking into account dependence between activities in PERT networks and provides a generalized algorithm to evaluate the project completion time. To accomplish this, we consider different activity network models with different structural dependence between distributions, and we employ a calculation procedure which retains a memory of structural links for later use.

The methodologies adopted previously can be classified into three main categories as follows:

1. Exact methods: Martin (1965) presented a method to compute the distribution function of the project duration time. The algorithm is described that reduces a series-parallel network to a single arc whose density function is that of the time through the original network. Also the pdf 's of the duration of activity are nominally distributed. Soukhakian (1988) presents a method for taking into account dependence between activities and provides a generalized algorithm to evaluate the project completion time and criticality index of each activity and path using a Controlled Interval and Memory (CIM) approach proposed by Chapman and Cooper (1983). Fisher *et al.* (1985) approach applied to only the cases in which the durations of the activities are independent, and exponentially or general-gamma distributed. Kulkarni and Adlakha (1986) presented simple and computationally stable algorithms to compute the distribution and moments of project completion time, the probability that a given path is critical, and other related performance measures, and developed analytical procedures for Markov PERT networks with independent and exponentially distributed activity durations. Schmidt and Grossmann (2000) present a new technique for computing the exact overall duration of a project, when task durations have independent distributions. Also a semi-analytical procedure is proposed to compute the cumulative distribution function (cdf) directly by integrating a linear transformation of the pdf of the task durations. Since the analytical calculations in the exact approaches are too time consuming, (because of heavy computational load), it was impossible for project managers to use them for large-size stochastic activity network (SAN).

Indian Journal of Science and Technology Vol. 5 No. 9 (Sep. 2012) ISSN: 0974- 6846

2. Approximation and Simulation methods: Van Slyke (1963) first developed the concept of using crude Monte Carlo methods to analyze the distribution function of the project completion time, and then defined a "critically" index for each activity. Dodin (1985) proposed an analytical procedure to approximate the distribution functions in stochastic networks. The procedure is efficient in the sense of its accuracy and its computational requirements. Pontrandolfo (2000) provided an approximate estimate of the project duration by deriving the equations that relates the duration of the project and those of every possible PERT-path. Yao and chu (2007) improved the techniques of discretization and present an algorithm to approximated pdf for the completion time of large size projects.

3. Bounding methods. Elmaghraby (1967) provided lower bounds for the true expected project completion time. Shogan (2006) developed a new method for obtaining probability distributions that bound the exact probability distribution of the project duration from above and below. The expected values of these bounding distributions, then furnish lower and upper bounds on the expected project duration. Azaron and Fatemi Ghomi (2008) apply the stochastic dynamic programming to obtain a lower bound for the mean project completion time in a PERT network, where the activity durations are exponentially distributed random variables (Azaron & Fatemi Ghomi, 2008).

Proposed procedure

In this paper, we describe the proposed procedure and present the algorithm of determining the probability distribution function of project completion time for PERT networks when duration times of activities are discretely distributed and structural dependence between activities is considered.

The proposed procedure is mainly based on Garman's (1972) approach. However, instead of fixing the duration times of chosen activities sample values, we conditionalize the chosen activities by fixing the random variables at their realization times. Consider this simple example:

Explaining main steps by simple example

Before explaining the proposed procedure, let's review this simple example to illustrate the main steps:

Fig. 1. A sample network

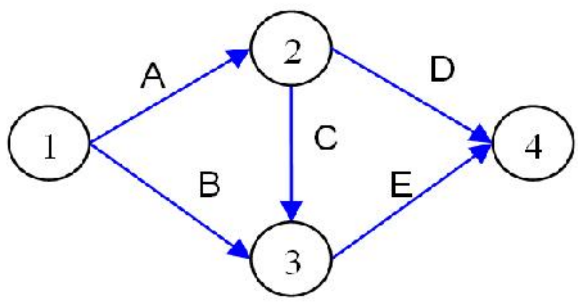


Fig. 2. Equivalent networks by fixing first realization time of A

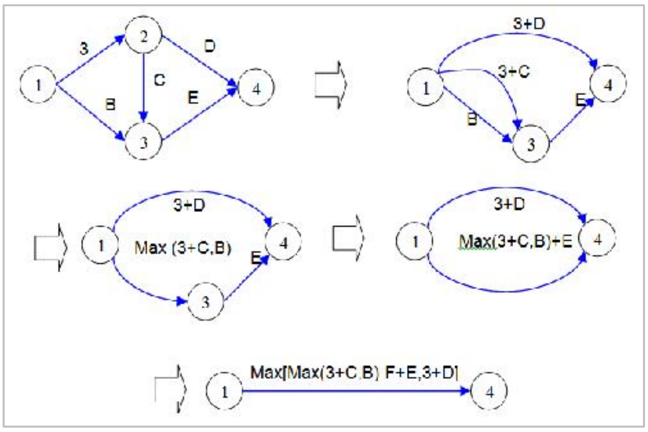
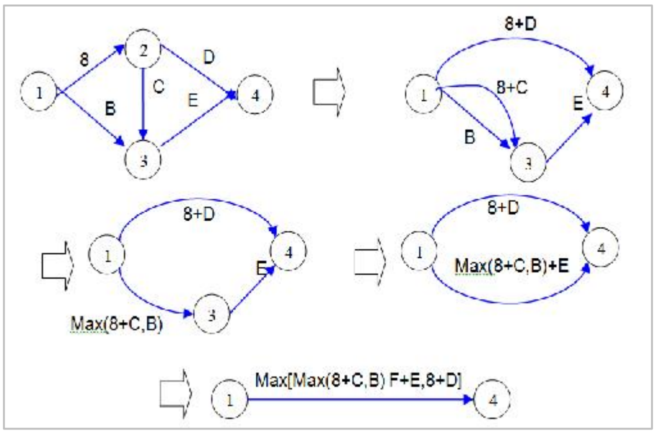


Fig. 3. Equivalent networks by fixing second realization time of A



By fixing on the first realization time of A, 3, we conditionalize this activity and change the network of Fig. 1 to that of Fig. 2 and all path durations are independent (Table 1).

Table 1. Probability of realization times of each activity

Activity	Duration 1	Probability	Duration 2	Probability
A	3	0.8	8	0.2
B	6	0.6	9	0.4
C	4	0.3	6	0.7
D	4	0.9	5	0.1
E	1	0.5	2	0.5

The last row of Table 2 shows probability of project completion times by fixing first realization time of A (3). It is clear that the probability of occurrence of this table is equal to probability of occurrence of fixed duration of activities conditioned. In this sample probability of occurrence of this table is 0.8. By fixing on the second realization time of A, (8) change the network of Fig. 1 to that of Fig. 3 and all path durations are independent.

The last row of Table 3 shows probability of project completion times by fixing second realization time of A, (8). The probability of occurrence of this table is 0.2. Now we should decondition the last row of Table 2 and the last

Table 2. Probability of realization times of equivalent activities

Activity	Duration1	Probability1	Duration2	Probability2	Duration3	Probability3	Duration4	Probability4
8+D	7	0.9	8	0.1	-	-	-	-
8+C	7	0.3	9	0.7	-	-	-	-
Max(8+C,B)	7	0.18	9	0.82	-	-	-	-
Max(8+C,B)+E	8	0.09	9	0.09	10	0.41	11	0.41
Max[Max (8+C,B) F+E,8+D]	8	0.09	9	0.09	10	0.41	11	0.41

row of Table 3, by multiplying the resulting distribution function to the probability of realization of each table (0.8 for Table 2 and 0.2 for Table 3) and finally, simple addition of the probabilities for each realization time, gives the unconditional pdf of project completion time as shown in Table 4.

Table 3. Probability of realization times of equivalent activities

Activity	Duration1	Probability1	Duration2	Probability2	Duration3	Probability3	Duration4	Probability4
8+D	12	0.9	13	0.1	-	-	-	-
8+C	12	0.3	14	0.7	-	-	-	-
Max(8+C,B)	12	0.3	14	0.7	-	-	-	-
Max(8+C,B)+E	13	0.15	14	0.15	15	0.35	16	0.35
Max[Max(8+ C,B) F+E,8+D]	13	0.15	14	0.15	15	0.35	16	0.35

An innovative method to reduce iterations

In above sample, we selected activity A for conditioning and this activity had two realization times, so we had to solve network twice. Assuming a network that has 5 activities that should be conditionalized and assume that on average each conditionalized activity has 5 realization time, in forward pass the project needs to be solved 5*5*5*5*5=55=3125 times. Imagine that the network structure is such that while using backward pass, only 3 activities need to be conditionalized, so, the project needs to be solved 5*5*5=53=125 times, which needs 4% of the computation effort required in forward pass. The fact is that both “the number of activities should be conditionalized” and “the number of realization times of each activity” can vary in forward and backward pass, while pdf of project completion time is the same in two ways.

Determination of the best direction of network before solving help us to do less computation effort, so, we proposed procedure in two phases, first select the best direction then solve the network by using selected direction.

Phase I: Selecting the best direction to solve network

In phase 1, the network is solved only 2 times by using first realization times of all conditionalized activities, first in forward pass and then in backward pass. We

Fig. 4. First sample network

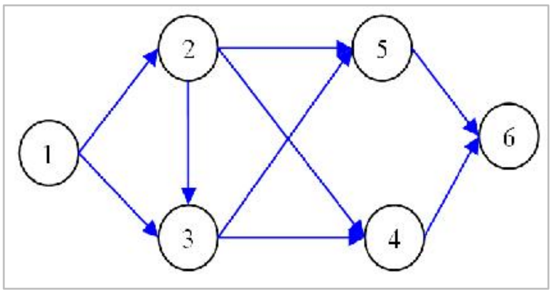


Fig. 5. Equivalent of network in fig. 4 (Conditioning activity 1->2)

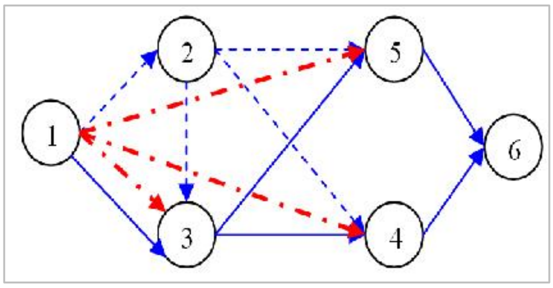


Fig. 6. Equivalent of network in fig. 5. after reducing

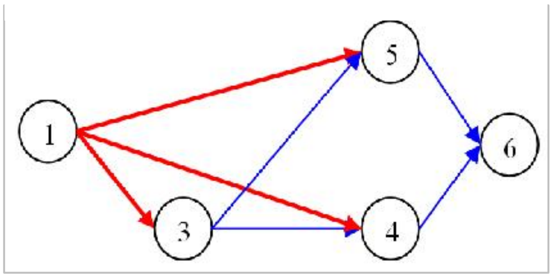


Fig. 7. Equivalent of network in fig. 6. (Conditioning new activity 1->3)

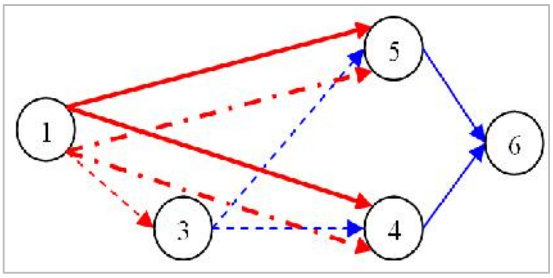


Fig. 8. Equivalent of network in figure 7 after reducing

Fig. 9. Comparing Dodin algorithm versus exact cumulative distribution of project completion time (proposed algorithm) (first sample network)

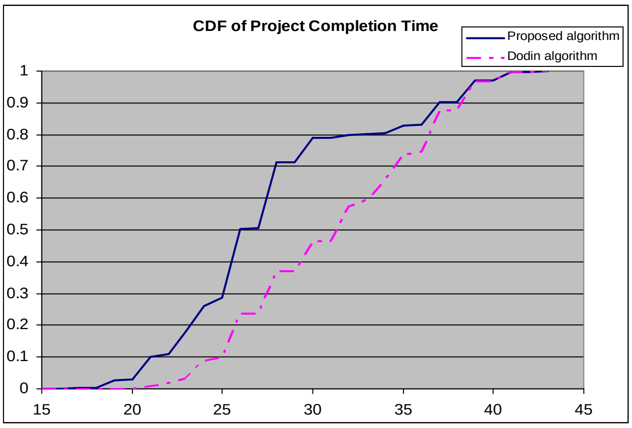


Fig. 10. Second sample network

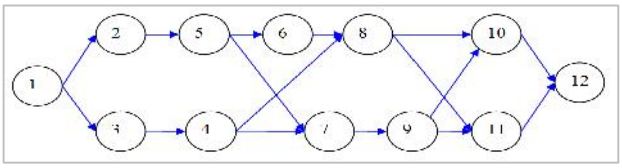


Fig. 11. Comparing Dodin algorithm versus exact cumulative distribution of project completion time (proposed algorithm) (second sample network)

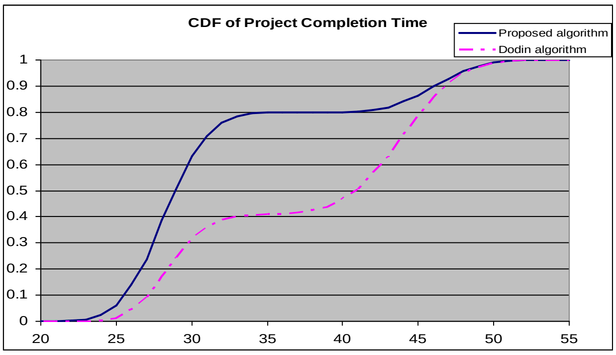


Fig. 12. Third sample network

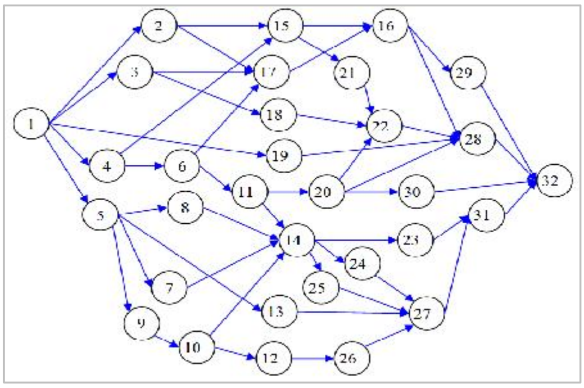
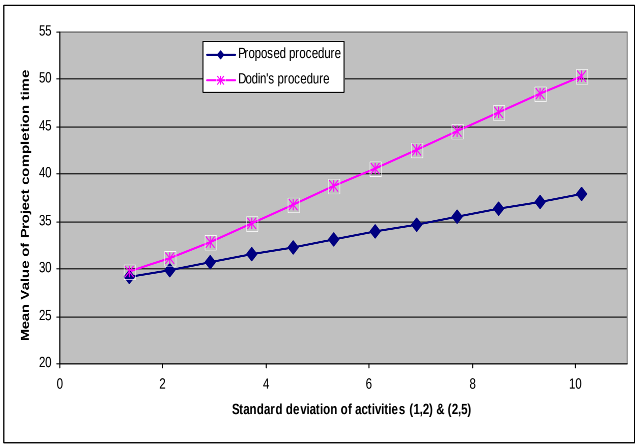


Fig. 13. Relationship between mean value of project completion time and standard deviation of common activities (1,2) & (2,5). (second sample network)



calculate two indices KF for forward pass and KB for backward pass. These two indices give us an approximated value of number of times that project should be solved. Based on these two indices, we decide that which direction needs less iteration to solve network.

Phase II: Determining the pdf of project completion time

In this phase network solved by using all realization times of all conditionalized activity:

1. Reduce the network to its irreducible form using convolution and greatest operations. (Algorithm of reduce network is developed by Dodin (1985).
2. If the network is reduced to an equivalent activity starting in node 1 and ending in node *N*, stop. The pdf of the duration time of the final activity is equal to *F(t).*
3. If the network is not completely reducible, calculate the indegree and outdegree of every node $i \neq N \quad (I(i), O(i))$. Then choose one activity '*a*' such that '*a*' has more than one successor while each of its successor has only '*a*' as a predecessor.
4. Conditionalize by setting the chosen activity '*a*' at its k[th] realization time, T_a^k; this is done by deleting '*a*', adding T_a^k to the successor of '*a*', and maintaining the implied precedence of activities in the conditionalized network.
5. Decondition the pdf of the final activity of step 4.
6. Determine the pdf of project completion time, mean and variance.

Table 4. pdf of project completion times

S.no	Duration	Probability
1	8	0.072
2	9	0.072
3	10	0.328
4	11	0.328
5	13	0.03
6	14	0.03
7	15	0.07
8	16	0.07

Indian Journal of Science and Technology Vol. 5 No. 9 (Sep. 2012) ISSN: 0974- 6846

Table 5. Probability of realization times of each activity of network Fig. 4.

Start Node	End Node	Duration1	Probablity1	Duration2	Probablity2	Duration3	Probablity3
1	2	2	0.2	7	0.6	18	0.2
1	3	4	0.2	6	0.6	8	0.2
2	3	5	0.2	7	0.6	9	0.2
2	4	6	0.2	8	0.6	10	0.2
2	5	7	0.2	9	0.6	11	0.2
3	4	4	0.2	6	0.6	8	0.2
3	5	5	0.2	7	0.6	9	0.2
4	6	4	0.2	6	0.6	8	0.2
5	6	3	0.2	5	0.6	7	0.2

Table 6. Exact pdf (and cdf) compare with the results of Dodin algorithm

Project Completion Time	Exact		Dodin algorithm	
	Probability	Cumulative	Probability	Cumulative
15	0.00003	0.00003	0.00000	0.00000
16	0.00003	0.00006	0.00000	0.00000
17	0.00224	0.00230	0.00002	0.00002
18	0.00103	0.00333	0.00001	0.00002
19	0.02299	0.02632	0.00084	0.00086
20	0.00401	0.03033	0.00029	0.00115
21	0.06884	0.09917	0.00805	0.00920
22	0.01147	0.11065	0.00763	0.01683
23	0.07004	0.18068	0.01627	0.03310
24	0.07989	0.26058	0.05497	0.08807
25	0.02691	0.28749	0.00947	0.09754
26	0.21600	0.50349	0.13780	0.23534
27	0.00314	0.50662	0.00124	0.23658
28	0.20736	0.71398	0.13308	0.36966
30	0.07661	0.79059	0.09404	0.46371
31	0.00004	0.79063	0.00209	0.46580
32	0.00943	0.80006	0.10695	0.57275
33	0.00275	0.80281	0.02171	0.59445
34	0.00051	0.80333	0.06100	0.65545
35	0.02642	0.82975	0.07992	0.73537
36	0.00046	0.83021	0.01113	0.74650
37	0.07200	0.90221	0.12960	0.87610
39	0.06912	0.97133	0.09216	0.96826
41	0.02554	0.99686	0.02855	0.99680
43	0.00314	1.00000	0.00320	1.00000
Mean	28.17		31.63	

Proposed algorithm

Phase I

1. ***Reduce*** network to its irreducible form using convolution and greatest operations.[3]
2. ***If*** new network (after reducing), contains only 1 activity ***then**: the pdf* of this activity is the *pdf* of project completion time; ***Stop***
3. Let *KF* = 1; *i*=1
4. ***Scan*** for selecting activity to be conditioned and name selected activity A(*i*)
5. Let k(*i*), the number of realization times of A(*i*)
6. ***Condition*** selected activity, and calculate new *pdf* of new activities by using first realization time of selected activity.
7. Let *KF* = *KF* * k(*i*) ; *i*= *i* +1
8. ***Reduce*** network
9. ***If*** new network (after reducing), contains more than 1 activity ***then**: go to 4*
10. ***Reverse*** reduced network resulted in line 1
11. Let *KB* = 1; *i*=1
12. ***Scan*** for selecting activity to be conditioned and name selected activity A(*i*)
13. Let k(*i*), the number of realization times of A(*i*)
14. ***Condition*** selected activity, and calculate new *pdf* of new activities by using first realization time of selected activity.
15. Let *KB* = *KB* * k(*i*) ; *i*= *i* +1
16. ***Reduce*** network
17. ***If*** new network (after reducing), contains more than 1 activity ***then**: go to 12*
18. ***If KB*** ≥ ***KF then:*** use main network to apply main algorithm
19. ***else:*** use reversed network to apply main algorithm

Phase II

1. Let *i* =1
2. Let current network, G(*i*) and do necessary bookkeeping
3. ***Scan*** for selecting activity to be conditioned and name selected activity A(*i*)
4. Let k(*i*), the number of realization times of A(*i*)
5. Let J(*i*)=1
6. Let p = Probability of occurrences J(*i*)th realization time of A(*i*)
7. If (*i*=1) ***then**:* let p(*i*) = p
   a. ***else:*** let p(*i*) = p * p(*i*-1)
8. ***Condition*** selected activity A(*i*) in network G(*i*),

Table 7. Probability of realization times of each activity of network Fig. 10.

Start Node	End Node	Duration1	Probablity 1	Duration2	Probablity 2	Duration3	Probablity 3
1	2	1	0.2	2	0.6	20	0.2
1	3	4	0.2	6	0.6	8	0.2
2	5	5	0.2	7	0.6	9	0.2
3	4	1	0.2	2	0.6	3	0.2
4	7	2	0.2	4	0.6	6	0.2
4	8	4	0.2	6	0.6	8	0.2
5	6	1	0.2	2	0.6	3	0.2
5	7	4	0.2	6	0.6	8	0.2
6	8	3	0.2	5	0.6	7	0.2
7	9	1	0.2	2	0.6	3	0.2
8	10	4	0.2	6	0.6	8	0.2
8	11	5	0.2	7	0.6	9	0.2
9	10	1	0.2	2	0.6	3	0.2
9	11	2	0.2	4	0.6	6	0.2
10	12	4	0.2	6	0.6	8	0.2
11	12	1	0.2	3	0.6	6	0.2

and calculate new *pdf* of new activities by using J(*i*)th realization time of A(*i*)

9. *Reduce* network G(*i*)
10. Let *i* = *i* + 1
11. *If* new network (after reducing), contains more than 1 activity *then*: *go to 2*
12. Multiply all probabilities of realizations time of the only remaining activity by p(*i*-1) and keep them as a part of project completion time *pdf*.
13. Let *i* = *i* - 1
14. Let J(*i*)= J(*i*) + 1
15. *If* J(*i*) ≤ k(*i*) *then*: *go to 6*
16. *If i* > 1 *then*: *go to 13*
17. Concatenate all parts of *pdf* resulted in line 12. Resulted pdf is project completion time pdf.

Numerical experiments

In this section three samples are presented to verify the efficiency of the algorithm. This algorithm was coded by *matlab Ver. 2010a*.

Solving a sample network to show numeric results

Consider this network (Table 5, Fig.4):

First of all, in phase 1, two indices must be calculated (KF=12, KB=27). Based on these indices, forward pass is used to solve the network.

This network is irreducible, according to algorithm; activity “1->2” is selected for conditioning. Network is solved as follow. Note that first iteration is explained in detail and other iterations are similar.

Conditioning: Activity Selected for conditioning: 1->2 (Fig. 5)

Distribution functions of activities of network in Fig. 6 are as follow:

1->3	7(0.16)	8(0.04)	9(0.6)	11(0.2)
1->5	9(0.2)	11(0.6)	13(0.2)	0(0.0)
1->4	8(0.2)	10(0.6)	12(0.2)	0(0.0)
3->4	4(0.2)	6(0.6)	8(0.2)	0(0.0)
3->5	5(0.2)	7(0.6)	9(0.2)	0(0.0)
4->6	4(0.2)	6(0.6)	8(0.2)	0(0.0)
5->6	3(0.2)	5(0.6)	7(0.2)	0(0.0)

Conditioning: Activity Selected for conditioning: 1->3 (Fig. 7)

The distribution function of activity (1,6) shown in Fig. 8 is as follow, and probability of occurrence of this, is 0.032: (Fig. 8)

1->6: 15(0.001024) 16(0.576) 17(0.063936) 19(0.428544)
18(0.012864) 21(0.403200) 20(0.011456) 23(0.078400)

The above result is obtained by "conditioning first duration of activity 1->2, then, *Reducing* network and conditioning first duration of activity 1->3 (in new network)". Now we repeat steps by other realization times of these two selected activities. Finally we decondition the results of all iterations, so we calculate distribution function of project completion time which shown in Table 6. The result of Dodin algorithm also is given in this table. The results also are shown in Fig. 9.

A sample network that shows the significant variation of Dodin algorithm from exact pdf (calculated by proposed algorithm)

Consider this network(Fig. 10, Table 7)*:* As shown in Fig. 11 the results of Dodin algorithm are significantly different from exact pdf:

In order to evaluate effect of standard deviations of common activities on the mean value of project completion time we gradually increase the standard deviation of activities (1,2) and (2,5). Diagram shown in Fig. 13 illustrate effects of standard deviation of common activities (1,2) & (2,5) on the mean value of project completion time obtained using (a) Dodin's procedure and (b) Proposed procedure.

This diagram shows that mean completion time obtained using Dodin's procedure is biased optimistically. As shown in diagram, when variance of duration’s time is low, the error of Dodin's algorithm is negligible but by increasing the standard deviation of only two activities from 1 to 10, the error increases from 2% to 33%. The reason is that, in Dodin's procedure it is assumed that all paths are structurally independent. In fact independence assumption among paths in Dodin's procedure is one of the sources of error in determination of project completion times in irreducible networks and this is an increasing function of the standard deviation of common activities and number of activities which emanate from the nodes in which these common activities terminate.

A sample network that shows effect of phase I to increase the efficiency of algorithm

The network shown in Fig. 12 contains 50 activities with random durations: In forward pass, 11 activities must be convolute and KF=708588 and in backward pass, 8 activities must be convolute and KB=68040. The sample network shows that by calculating KF and KB by solving only 2 times, can reduce about 90% of the iterations and calculation time will also be reduced, significantly. In this network, the results of Dodin algorithm significantly are different from exact pdf obtained by proposed procedure. By increasing standard deviation of only two common activities (27,31) and (28,32) in backward pass, the mean completion time obtained using Dodin's procedure with standard deviation 10 has about 2.5% error whereas with standard deviation 50 the resulting error increase to 28% in comparison with exact mean completion time obtained using proposed procedure.

If we increase the variance of more activities, the discrepancy of Dodin's results with the exact pdf obtained by our proposed procedure will be increased. On the other hand this example shows that, since the proposed procedure can calculate exact pdf of this network with many common activities in approximately 3 min using a PC with Core i3-2100 CPU and 2GB RAM, so this algorithm can be used for solving larger networks with less common activities in reasonable time.

Conclusion and future research

In this article we have presented an efficient procedure to determine the probability distribution function of project completion time for PERT network with discrete statistically independent distribution. This algorithm possesses the following advantages and can be expanded to meet the following results:

1. Provide an exact pdf for project completion time in pert network with discrete distribution because the project managers are very interested to have the pdf of the project completion time so as to have complete insight into the randomness of the realization of the project.

2. It can be applied for PERT networks with statistical dependence as well as structural dependence relationship between activities.

3. It can be applied for PERT networks with discrete or continues distribution. To evaluate the performance of the proposed algorithm, we randomly generate large sample networks. Then we compared the proposed algorithm with Dodin's algorithm (DA) in precision aspect. We concluded that in some cases the proposed algorithm significantly outperforms the DA in precision aspect. This means that the Dodin algorithm in some cases lead to a pessimistically biased estimates of the occurrence time of events.

References

1. Azaron SM and Fatemi Ghomi T (2008) Lower bound for the mean project completion time in dynamic PERT networks. *Eur. J. Operational Res.* 186, 120-127.
2. Chapman CB and Cooper DF (1983) Risk engineering: basic controlled interval and memory models. *J. Operational Res. Soc.* 34(1), 51-60.
3. Dodin (1985) Approximating the distribution functions in stochastic networks. *Compu. & Operations Res.* 12, 251-264.
4. Elmaghraby SE (1967) On the expected duration of PERT type networks. *Manage. Sci.* 13, 299-306.
5. Fisher DL, Saisi D and Goldestein WM (1985) Stochastic PERT networks: OP diagrams, critical paths, and the project completion time. *Comput. & Operation Res.* 12, 471-482.
6. Garman MB (1972) More on conditioned sampling in the simulation of stochastic network. *Manage. Sci.* 19, 90-95.
7. Kulkarni VG and VG Adlakha (1986) Markov and Markov-regenerative PERT networks. *Operations Res.* 34, 769-781.
8. Martin JJ (1965) Distribution of the time through a directed, acyclic network. Operation Res. 13, 46-66.
9. Pontrandolfo P (2000) Project duration in stochastic networks by the PERT-path technique. *Int. J. Project Manage.* 18, 215-222.
10. Schmidt CW and Grossmann IE (2000) The exact overall time distribution of a project with uncertain task durations. *Eur. J. Operational Res.* 126, 614-636.
11. Shogan AW (2006) Bounding distributions for a stochastic pert network. *Networks.* 7, 359-381.
12. Soukhakian MA (1988) A generalized algorithm to evaluate project completion time and criticality indices for PERT networks. *Ph.D. Thesis Univ. Southampton*, England.
13. Van Slyke RM (1963) Monte carlo methods and the pert problems. *Operations Res.* 11, 839-860.
14. Yao MJ and Chu WM (2007) A new approximation algorithm for obtaining the probability distribution function for project completion time. *Compu. & Maths with Appl. Archive.* 54, 282-295.