Journal of Industrial Engineering International *Islamic Azad University, Tehran South Branch*
September 2005, Vol. 1, No. 1, 1 - 9

A hybrid method to find cumulative distribution function of completion time of GERT networks

S. S. Hashemin*
Department of Industrial Engineering, Islamic Azad University, Science and Research Branch, Tehran, Iran

S. M. T. Fatemi Ghomi
Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran

Abstract
This paper proposes a hybrid method to find cumulative distribution function (CDF) of completion time of GERT-type networks (GTN) which have no loop and have only exclusive-or nodes. Proposed method is created by combining an analytical transformation with Gaussian quadrature formula. Also the combined crude Monte Carlo simulation and combined conditional Monte Carlo simulation are developed as alternative methods of solution procedure. Then, through a comparative study made for different solution procedures, the superiority of hybrid method is indicated. Computing time and accuracy are considered as fundamental factors for comparison purposes.

Keywords: GERT network; Completion time; Distribution function; Hybrid; Gaussian quadrature formula; Conditional simulation

1. Introduction

Graphical evaluation and review technique is a strong tool for the analysis of the industrial engineering problems. Taylor and Davis [10] have explored the use of GERT-type networks (GTN) analysis as a tool for planning and determining the expected time and cost of system implementation. Interrante and Biegel [4] have described a modified GERT network which has been developed for automatically acquiring temporal knowledge to be used in an intelligent simulation training system. Dowson [2] has introduced a dynamic sampling technique for the simulation of probabilistic and generalized activity networks. Zimmermann [13], using GERT network precedence constraints, has examined time complexity of single-and identical parallel-machine scheduling. In this study, the duration and precedence constraints of activities are assumed to be stochastic.

Acquisition of GTN completion time can provide useful data. Analytical methods for this matter have been introduced in [6, 7, 8, 11, 12]. Furthermore Whitehouse [11] has introduced simulation methods. Kurihara and Nishiuchi [5] have proposed efficient Monte Carlo simulation method to estimate GTN characteristics such as project time, cost, etc. Shibanov [9] has developed an algorithm to fulfill equivalent simplifying transformations of the structure of GTN.

This paper discusses GTN which have no loop and have exclusive-or nodes. The network has one start node and numerous end nodes. Activity durations are assumed to be continuous random variables or constants.

This paper also develops a hybrid (analytical-numerical) method created by combining an analytical transformation with integration through Gaussian quadrature formula. This procedure behavior is more exact in respective existing simulation methods and even when these methods combine with the proposed analytical transformation of this paper. In addition, the different solution procedures are capable to find the occurrence probability of end nodes of networks in every arbitrary given time.

The paper has the following structure:

Section 2 introduces notations. Section 3 transforms the above-mentioned GTN to a GERT network with parallel paths. Sections 4 and 5 combine crude Monte Carlo simulation and conditional Monte Carlo simulation with analytical transformation of section

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

3. Section 6 presents a hybrid method. Section 7 gives an example to illustrate the capabilities of hybrid method in comparison with combined crude Monte Carlo simulation and combined conditional Monte Carlo simulation. Finally section 8 is devoted to the concluding remarks and recommendations for future studies.

2. Notations

The following notations have been used in this paper:

N : Number of activities (arcs)
M : Number of end nodes
Q : Number of simulation runs
n_i : Number of paths which start from start node and terminate in i-th end node
S_{ij} : Activity set of j-th path which terminates in i-th end node
$\overline{P}_k$: Accomplishment probability of k-th activity, given that start node of this activity has occurred
P_{ij} : Occurrence probability of j-th path which terminates in i-th end node
$P_i(t)$: Occurrence probability of i-th end node in t
$\hat{P}_i(t)$: Estimation (approximation) of $P_i(t)$
P_i : Occurrence probability of i-th end node when $t \to +\infty$
t_k : Duration random variable of k-th activity
$t_k^{(q)}$: Duration of k-th activity in q-th simulation run
T_{ij} : Completion time of j-th path which terminates in i-th end node
$T_{ij}^{(q)}$: Completion time of j-th path which terminates in i-th end node in q-th simulation run
L_{ij} : Counter of $T_{ij}^{(q)}$ which smaller than or equal to t
$F_{t_r}(t)$: CDF of r-th activity duration
$F_{t_r}^{(q)}(t)$: $F_{t_r}(t)$ in q-th simulation run
$F_{ij}(t)$: CDF of j-th path which terminates in i-th end node
$F_{ij}^{(q)}(t)$: $F_{ij}(t)$ in q-th simulation run
$\hat{F}_{ij}(t)$: Estimation (approximation) of $F_{ij}(t)$
L'_{ij} : Saver of sum of $F_{ij}^{(q)}(t | t_k, k \in S_{ij}, k \neq r)$
$F_i(t)$: CDF of occurrence time of i-th end node, given that this node has occurred
$\hat{F}_i(t)$: Estimation (approximation) of $F_i(t)$
$f_{t_r}(t)$: Probability density function of r-th activity duration
$f_{t_r}^{(q)}(t)$: $f_{t_r}^{(q)}(t)$ in q-th simulation run

3. Transformation of GERT networks

In GTN which have no loop and have only exclusive-or nodes we can write:

$$F_i(t) = \frac{\sum_{j=1}^{n_i} P_{ij} F_{ij}(t)}{\sum_{j=1}^{n_i} P_{ij}} \qquad i = 1,2,\dots,M$$

When $t \to +\infty$, the limit of above equality would be:

$$\lim_{t \to +\infty} \frac{\sum_{j=1}^{n_i} P_{ij} F_{ij}(t)}{\sum_{j=1}^{n_i} P_{ij}} = \lim_{t \to +\infty} F_i(t) = 1$$

Based on $P_i = \sum_{j=1}^{n_i} P_{ij}$, we can write:

$$P_i = \lim_{t \to +\infty} \sum_{j=1}^{n_i} P_{ij} F_{ij}(t)$$

where $P_{ij} = \prod_{k \in S_{ij}} \overline{P}_k$.

For shorter time (when t is remarkably smaller than $+\infty$) we have:

$$P_i(t) = \sum_{j=1}^{n_i} P_{ij} F_{ij}(t)$$

and $\lim_{t \to +\infty} P_i(t) = P_i$ is evident. Consequently we can transform the above-mentioned GTN to the GERT networks with $\sum_{i=1}^{M} \sum_{j=1}^{n_i} 1$ parallel paths such that j-th path which terminates in i-th end node consists of activities which belong to S_{ij}. So if we can compute $F_{ij}(t)$, $F_i(t)$ can be defined. To compute $F_{ij}(t)$, we must define the CDF of sum of activity durations which belong to S_{ij} ($t = \sum_{k\in S_{ij}} t_k$). This task can be done by numerous methods. These methods have been introduced in the next sections.

4. Combined crude Monte Carlo simulation

If we generate the random value for each t_k, we can compute the completion time of all paths using:

$$T_{ij}^{(q)} = \sum_{k\in S_{ij}} t_k^{(q)} \qquad q = 1,2,...,\mathrm{Q}$$

An estimation of $F_{ij}(t)$ ($\hat{F}_{ij}(t)$) is obtained by dividing the number of $T_{ij}^{(q)}$ which is smaller than or equal to t over Q. Then $\hat{F}_i(t)$ and $\hat{P}_i(t)$ can be ascertained.

The following algorithm is proposed for the abovementioned computations:

Step 1: $q = 1$

Step 2: For j-th path which terminates in i-th end node set $L_{ij} = 0$.

Step 3: Generate random values $t_1^{(q)}, t_2^{(q)}, ..., t_N^{(q)}$, then compute $T_{ij}^{(q)}$ for $i = 1,2,...,M$, $j = 1,2,...,n_i$.

Step 4: If $T_{ij}^{(q)} \le t$, then $L_{ij} \leftarrow L_{ij} + 1$.

Step 5: Set $q \leftarrow q+1$. If $q \le Q$, go to step 3; otherwise go to step 6.

Step 6: First compute $\hat{F}_{ij}(t) = \frac{L_{ij}}{Q}$, then compute the following estimations:

$$\hat{F}_i(t) = \frac{\sum_{j=1}^{n_i} (\prod_{k\in S_{ij}} \overline{P}_k) \hat{F}_{ij}(t)}{\sum_{j=1}^{n_i} (\prod_{k\in S_{ij}} \overline{P}_k)} \qquad i = 1,2,...,M \qquad (1)$$

$$\hat{P}_i(t) = \sum_{j=1}^{n_i} (\prod_{k\in S_{ij}} \overline{P}_k) \hat{F}_{ij}(t) \qquad i = 1,2,...,M \qquad (2)$$

Step 7: Stop.

5. Combined conditional Monte Carlo simulation

Conditional Monte Carlo simulation has been proposed by Burt and Garman [1] for stochastic network analysis. Based on $T_{ij} = \sum_{k\in S_{ij}} t_k$ we can write:

$$P(T_{ij} \le t) = P(\sum_{k\in S_{ij}} t_k \le t)$$

If $r \in S_{ij}$, then

$$\mathrm{P}(\mathrm{T}_{ij} \le \mathrm{t}) = \mathrm{P}(\mathrm{t}_r \le \mathrm{t} - \sum_{\substack{k\in S_{ij} \\ k\ne r}} \mathrm{t}_k)$$

Using the conditional probability

$$\mathrm{P}(\mathrm{T}_{ij} \le \mathrm{t} \mid \mathrm{t}_k, k \in \mathrm{S}_{ij}, k \ne r) = \mathrm{P}(\mathrm{t}_r \le \mathrm{t} - \sum_{\substack{k\in S_{ij} \\ k\ne r}} \mathrm{t}_k \mid \mathrm{t}_k, k \in \mathrm{S}_{ij}, k \ne r)$$

Right hand side of the above equality represents the conditional CDF of r-th activity and left hand side of the above equality represents the conditional CDF of completion time of j-th path which terminates in i-th end node. So we can write:

$$\mathrm{F}_{ij}(\mathrm{t} \mid \mathrm{t}_k, k \in \mathrm{S}_{ij}, k \ne r) = \mathrm{F}_{t_r}(\mathrm{t} - \sum_{\substack{k\in S_{ij} \\ k\ne r}} \mathrm{t}_k \mid \mathrm{t}_k, k \in \mathrm{S}_{ij}, k \ne r)$$

The following algorithm is proposed for combined conditional Monte Carlo simulation implementation:

Step 1: $q = 1$

Step 2: For j-th path which terminates in i-th end node set $L'_{ij} = 0$.

Step 3: Generate random values $t_1^{(q)}, t_2^{(q)}, ..., t_N^{(q)}$

Step 4: Compute:

$$F_{ij}^{(q)}(t \mid t_k, k \in S_{ij}, k \ne r) = F_{t_r}^{(q)}(t - \sum_{\substack{k\in S_{ij} \\ k\ne r}} t_k^{(q)} \mid t_k, k \in S_{ij},$$

$k \ne r)$ for $i = 1,2,...M$, $j = 1,2,...,n_i$.

Then $L'_{ij} \leftarrow L'_{ij} + F_{ij}^{(q)}(t \mid t_k, k \in S_{ij}, k \ne r)$.

Step 5: Set $q \leftarrow q+1$. If $q \le Q$, go to step 3; otherwise go to step 6.

Step 6: Compute $\hat{F}_{ij}(t) = \frac{L'_{ij}}{Q}$. Also compute $\hat{F}_i(t)$ and $\hat{P}_i(t)$ using formulae (1) and (2).

Step 7: Stop.

6. A hybrid (analytical-numerical) method

In the previous section, it has been shown that:

$$F_{ij}(t \mid t_k, k \in S_{ij}, k \neq r) = F_{t_r}(t - \sum_{\substack{k \in S_{ij} \\ k \neq r}} t_k \mid t_k, k \in S_{ij}, k \neq r)$$

To avoid simulation error, we can compute $F_{ij}(t)$ exactly by the following relation.

$$F_{ij}(t) = \underset{\sum_{k \in S_{ij}} t_k \leq t}{\iint \dots \int} F_{t_r}(t - \sum_{\substack{k \in S_{ij} \\ k \neq r}} t_k) \prod_{\substack{k \in S_{ij} \\ k \neq r}} (f_{t_k}(t_k) dt_k)$$

Except for special cases, the above analytical computation is not so much easy job. So, now application of Gaussian quadrature formula that generalized for stochastic networks by Fatemi Ghomi and Hashemin [3] is being proposed. In utilization of proposed numerical method, the determination of integral's bounds is not an easy task. We have:

$$\begin{cases} F_{t_r}(t - \sum_{\substack{k \in S_{ij} \\ k \neq r}} t_k) \geq 0 & \text{if} \quad t - \sum_{\substack{k \in S_{ij} \\ k \neq r}} t_k \geq \min\{t_r\} \\ F_{t_r}(t - \sum_{\substack{k \in S_{ij} \\ k \neq r}} t_k) = 0 & \text{if} \quad t - \sum_{k \in S_{ij}} t_k < \min\{t_r\} \end{cases}$$

Then integration intervals for all integrals will be $[Min\{t_k\}, t]$ for $k \in S_{ij}, k \neq r$. So we can write

$$F_{ij}(t) = \int_{\min\{t_1\}}^{t} \dots \int_{\min\{t_k\}}^{t} \dots \int_{\min\{t_l\}}^{t} F_{t_r}(t - \sum_{\substack{k \in S_{ij} \\ k \neq r}} t_k) \prod_{\substack{k \in S_{ij} \\ k \neq r}} (f_{t_k}(t_k) dt_k)$$

Where $\{1, \dots, k, \dots, l\} = S_{ij} - \{r\}$. By computing $F_{ij}(t)$ for $i = 1, 2, \dots, M$, $j = 1, 2, \dots, n_i$, $F_i(t)$ and $P_i(t)$ can be approximated accurately using formulae (1) and (2).

7. Example

On a production line a part is manufactured at the beginning of the line. The manufacturing operation is assumed to take 4 hours. Before the finishing touches are put on the part, it is inspected, with 25% of parts failing the inspection and requiring rework.

The inspection time (including waiting for inspection) is assumed to be distributed according to the exponential distribution, with a mean of 1 hour. Reworking takes 3 hours, and 30% of the parts reworked fail the next inspection. The inspection of the reworked items is assumed to take $\frac{3}{4}$ hour. Parts which fail this inspection, are scrapped. If the part passes either of the above inspections, it is sent to the final finishing operation, whose time is distributed according to the exponential distribution with a mean of 1 hour in 40% of the time and $\frac{1}{2}$ hour in 60% of time. A final inspection, which takes 1 hour, rejects 5% of the parts; these are scrapped.

The manufacturer intends to know what are the probabilities of having non-defective and scrapped parts and the corresponding cumulative distribution function of the times will take for receipt of non-defective and scrapped parts. The GERT network for the above production line is illustrated in Fig. 1.

The example is designed in such a way that the exact analytical solution can be found. The aim has been the establishment of possibility for a straightforward comparison between the actual solution and the solution gained from the other methods. The transformed GERT network is shown in Fig. 2.

We should first compute P_{ij} for $i = 1, 2, \dots, M$, $j = 1, 2, \dots, n_i$.

$$P_{11} = \overline{P}_1 \overline{P}_2 \overline{P}_4 \overline{P}_6 = (1)(0.25)(1)(0.3) = 0.075$$

$$P_{12} = \overline{P}_1 \overline{P}_2 \overline{P}_4 \overline{P}_5 \overline{P}_7 \overline{P}_{10} = (1)(0.25)(1)(0.7)(0.4)(0.5) = 0.0035$$

$$P_{13} = \overline{P}_1 \overline{P}_3 \overline{P}_7 \overline{P}_{10} = (1)(0.75)(0.4)(0.05) = 0.015$$

$$P_{14} = \overline{P}_1 \overline{P}_2 \overline{P}_4 \overline{P}_5 \overline{P}_8 \overline{P}_{10} = (1)(0.25)(1)(0.7)(0.6)(0.05) = 0.00525$$

$$P_{15} = \overline{P}_1 \overline{P}_3 \overline{P}_8 \overline{P}_{10} = (1)(0.75)(0.6)(0.05) = 0.0225$$

$$P_{21} = \overline{P}_1 \overline{P}_2 \overline{P}_4 \overline{P}_5 \overline{P}_7 \overline{P}_9 = (1)(0.25)(1)(0.7)(0.4)(0.95) = 0.0665$$

$$P_{22} = \overline{P}_1 \overline{P}_2 \overline{P}_4 \overline{P}_5 \overline{P}_8 \overline{P}_9 = (1)(0.25)(1)(0.7)(0.6)(0.95) = 0.09975$$

$P_{23} = \overline{P}_1\overline{P}_3\overline{P}_7\overline{P}_9 = (1)(0.75)(0.4)(0.95) = 0.285$

$P_{24} = \overline{P}_1\overline{P}_3\overline{P}_8\overline{P}_9 = (1)(0.75)(0.6)(0.95) = 0.4275$

$F_{ij}(t)$ and then $F_i(t)$, $P_i(t)$ can be obtained exactly by analytical method. $\hat{F}_{ij}(t)$ and then $\hat{F}_i(t)$, $\hat{P}_i(t)$ can be obtained by:

I) Combined crude Monte Carlo simulation,
II) Combined conditional Mote Carlo simulation,
III) Hybrid method.

$\lim_{t\to+\infty} \sum_{i=1}^{M} P_i(t) = 1$ is evident. For smaller t $\sum_{i=1}^{M} P_i(t) < 1$, and the probability of none of end nodes to occur in t is $1-\sum_{i=1}^{M} P_i(t)$. In other word for each product probability of being on production line is $1-\sum_{i=1}^{M} P_i(t)$ in t. $\hat{F}_i(t)$ and $\hat{P}_i(t)$ are computed for 53 values of t through the above methods. Obtained results for some values of t are shown in tables 1 through 8.

Table 1. Analytical method results for end node 1.

t	$P_1(t)$	$F_1(t)$
5	0	0
6	0.01295409	0.1068378
7	.02573192	0.2122221
8	.04891807	0.403448
9	.08917144	0.7354345
10	0.1076029	0.8874465
11	.115723	.9544163
12	.1190537	.9818865
13	.1203853	.992868
14	.1209114	.9972074
15	.121118	.9989111
16	.1211987	.9995768
17	.1212301	.999836
18	.1212423	.9999366

Table 2. Analytical method results for end node 2.

t	$P_2(t)$	$F_2(t)$
5	0	0
6	0.2461276	0.2800883
7	.4889066	0.556366
8	.6142344	.6989865
9	0.6775267	0.7710118
10	.7696486	.8758448
11	.8289897	.9433738
12	.8573475	.9756444
13	.8697973	.989812
14	.8750677	.9958096
15	.8772534	.9982969
16	.8781474	.9993142
17	.8785092	.9997259
18	.8786544	.9998912

Table 3. Combined crude Monte Carlo simulation results for end node 1.

t	$\hat{P}_1(t)$	$\hat{F}_1(t)$
5	0	0
6	.012981	.1070598
7	.0257835	.2126474
8	.0486465	.4012083
9	0.08900435	.7340565
10	.1075707	.887181
11	.1158533	.9554911
12	.1192017	.9831064
13	.1203997	.9929872
14	.120904	.9971464
15	.1210846	.9986359
16	.1211615	.9992697
17	.1211952	.9995476
18	.1212184	.9997394

Table 4. Combined crude Monte Carlo simulation results for end node 2.

t	$\hat{P}_2(t)$	$\hat{F}_2(t)$
5	0	0
6	.246639	.2806703
7	.4898865	.5574811
8	.6119235	.6963568
9	.6779076	.7714454
10	.7687533	.8748259
11	.8295077	.9439632
12	.8575014	.9758195
13	.8691493	.9890746
14	.874741	.9954377
15	.8770324	.9980454
16	.8779225	.9990583
17	.8782778	.9994627
18	.8784346	.9996411

Table 5. Combined conditional Monte Carlo simulation results for end node 1.

t	$\hat{P}_1(t)$	$\hat{F}_1(t)$
5	0	0
6	.01292163	.1065701
7	.02567927	.2117878
8	.04889193	.4032324
9	.08919495	.7356284
10	.1075624	.8871123
11	.1157274	.9544529
12	.1190379	.9817562
13	.1203787	.9928139
14	.1209184	.9972649
15	.1211158	0.998893
16	.1211981	.9995716
17	.1212312	.9998449
18	.1212434	.9999459

Table 6. Combined conditional Monte Carlo simulation results for end node 2.

t	$\hat{P}_2(t)$	$\hat{F}_2(t)$
5	0	0
6	.2461815	.2801496
7	.4912224	.5590013
8	.6148416	.6996775
9	.6773212	.770778
10	.7696468	.8758427
11	.828837	.9432
12	.8575835	.9759129
13	.8697481	.9897559
14	.8751877	.9959461
15	.8772799	.998327
16	.8781955	.9993689
17	.8784835	.9996967
18	.8786638	.9999018

Table 7. Hybrid (analytical-numerical) method results for end node 1.

t	$\hat{P}_1(t)$	$\hat{F}_1(t)$
5	0	0
6	.01 295518	.1068469
7	.02573596	.2122553
8	.04892457	.4035016
9	.08917 915	.7354981
10	.1076112	.8875148
11	.1157315	.954487
12	.1190621	.9819555
13	.120393	.9929323
14	.1209184	.9972652
15	.1211241	.9989617
16	.121204	.9996207
17	.1212347	.9998735
18	.1212461	.9999681

Table 8. Hybrid (analytical-numerical) method results for end node 2.

t	$\hat{P}_2(t)$	$\hat{F}_2(t)$
5	0	0
6	.2461485	.2801121
7	.4889833	.5564532
8	.614358	.6991271
9	.6776732	.7711785
10	.7698062	.8760241
11	.8291526	.9435591
12	.8575065	.9758253
13	.8699453	.9899804
14	.8752006	.9959608
15	.8773703	.9984299
16	.8782485	.9994293
17	.8785957	.9998243
18	.878727	.9999738

8. Conclusions

a) Table 9 presents computing time and mean absolute error for different proposed methods.

b) Hybrid method presents more accurate solutions in shorter time. For more complex networks, dimension of integrals and computing time will increase. Simulation time will be longer for more complex networks too.

c) Combined conditional Monte Carlo simulation is more accurate than combined crude Monte Carlo simulation, but computing time of combined conditional Monte Carlo simulation is longer.

d) With increase in the value of t, the integration interval will be wider and consequently the integration error increases. With the division of integration intervals into the smaller intervals and the computation of integral for each small subintervals, the accuracy of solutions can be increased. In the proposed example, although this operation has not been performed, the solutions gained by Gaussian quadrature formula have been more accurate than the solutions of simulation methods.

e) The similar research can be done on the GERT networks with loop.

f) The similar research can be performed on the GERT networks with inclusive-or and AND nodes.

g) This paper assumed that the random variables of activities are of additive type. The similar research can be done under the assumption of multiplicative type for random variables of activities.

h) The creation of a power tool has been one of the aims of this research, so that it can help us in the next stages of research where the required resources to accomplish activities are limited.

Table 9. Computing time and mean absolute error of different proposed methods.

Comparison criteria / Methods	Computing time (Second)	Mean absolute error of computation for $P_i(t)$	Mean absolute error of computation for $F_i(t)$
Hybrid (analytical-numerical)	$\underset{\sim}{~}$0	0.0001174559	0.000174559
Combined crude Monte Carlo simulation	2.470703	0.000614849	0.001133693
Combined conditional Monte Carlo simulation	9.179688	0.0003926627	0.0005972685

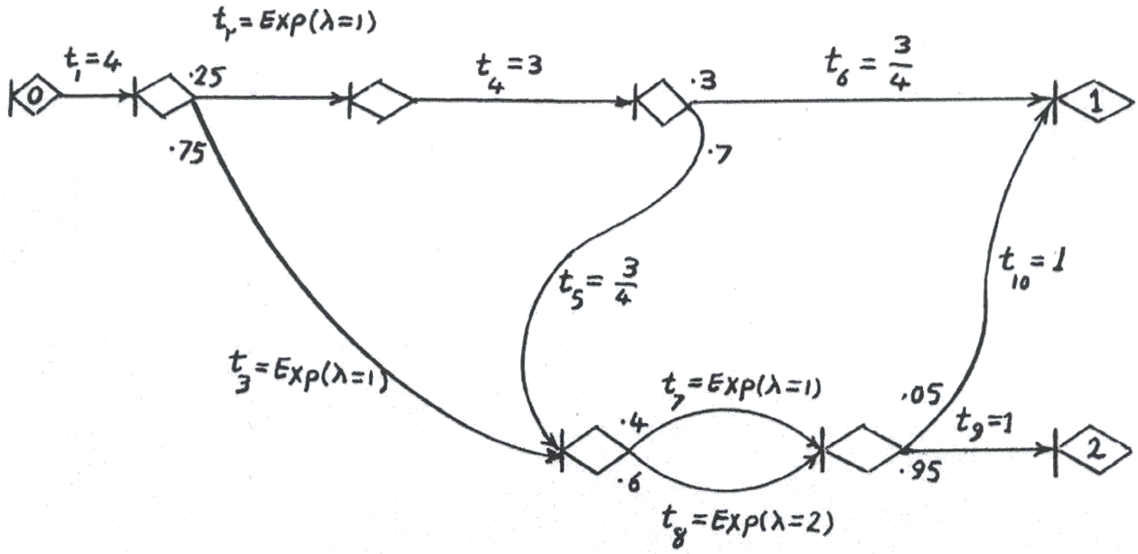

Figure 1. The GERT network of typical example.

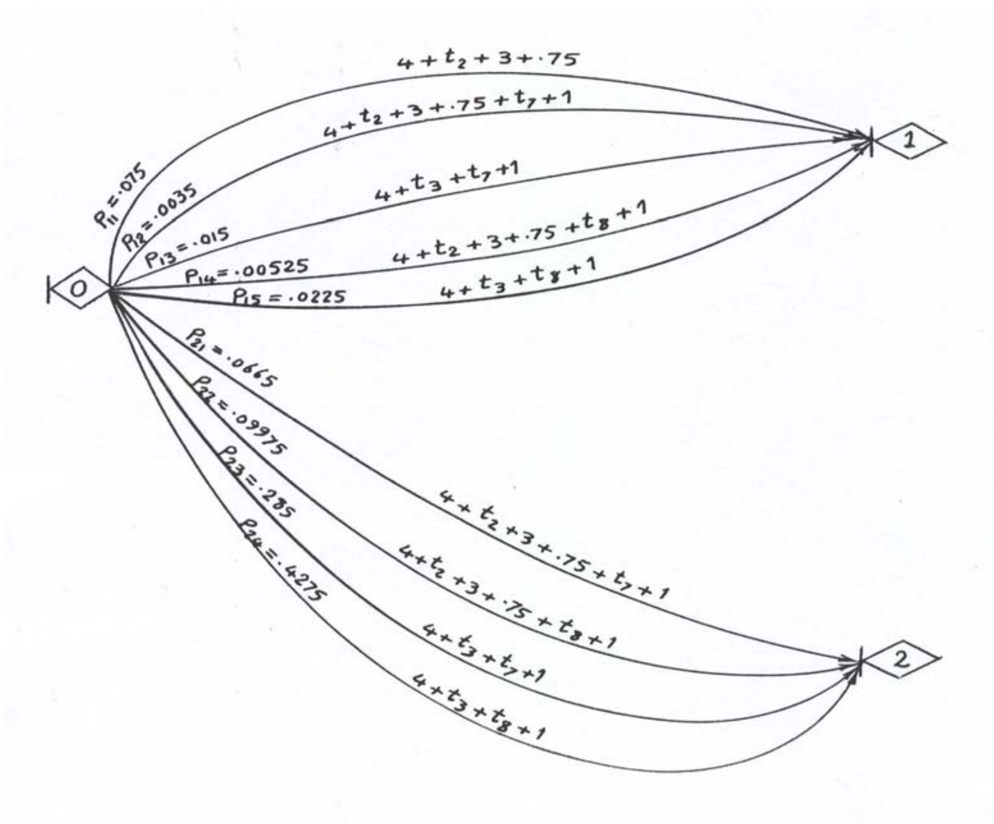

Figure 2. Transformed GERT network of typical example.

References

[1] Burt, J. M. and Garman, M. B., 1971, Conditional Monte Carlo: a simulation technique for stochastic network analysis. *Management Science*, 18, 207-217.

[2] Dawson, C. W., 1995, A dynamic sampling technique for the simulation of probabilistic and generalized activity networks. *Omega*, 23, 557-566.

[3] Fatemi Ghomi, S. M. T. and Hashemin, S. S., 1999, A new analytical algorithm and generation of Gaussion quadrature formula for stochastic network. *European Journal of Operational Research,* 114 , 610-625.

[4] Interrante, L. D. and Biegel, J. E., 1991, A modified GERT network for automatic acquisition of temporal knowledge. *Computers and Industrial Engineering*, 21, 79-83.

[5] Kurihara, K. and Nishiuchi, N., 2002, Efficient Monte Carlo simulation method of GERT-type network for project management. *Computers and Industrial Engineering*, 42, 521-531.

[6] Pritsker, A. A. B. and Happ, W. W., 1966, GERT: graphical evaluation and review technique, Part I, Fundamentals. *Journal of Industrial Engineering, 17*.

[7] Pritsker, A. A. B. and Whitehouse, G. E., 1966, GERT: graphical evaluation and review technique, Part II, Probabilistic and industrial engineering applications. *Journal of Industrial Engineering, 17*.

[8] Pritsker, A. A. B., 1966, GERT: graphical evaluation and review technique. *Rand Memorandum RM-4973-NASA.*

[9] Shibanov, A.P., 2003, Finding the distribution density of the time taken to fulfill the GERT network on the basis of equivalent simplifying transformation. *Automation and Remote Control*, 64, 279 - 287.

[10] Taylor, B. W. and Davis, K. R., 1978, Evaluating time/cost factors of implementation via GERT simulation. *Omega*, 6, 257-266.

[11] White house, G. A., 1973, *System Analysis and Design Using Network Techniques*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

[12] Whitehouse, G. E. and Pritsker, A. A. B. 1969, GERT-generating functions, conditional distributions, counters, renewal times and correlations. *AIIE Transaction.*

[13] Zimmermann, J., 1999, Time complexity of single-and identical parallel-machine scheduling with GERT network precedence constraints. *Mathematical Methods of Operations Research,* 49, 221-238.