Research Article

A New Approach to Determine the Critical Path in Stochastic Activity Network

Weng-Ming Chu,¹ Koan-Yuh Chang,² Chien-Yu Lu,³ Chang-Hung Hsu,⁴
Chien-Hung Liu,⁵ and Yung-Chia Hsiao⁶

¹ Department of Electrical Engineering, National Chin-Yi University of Technology, Taichung 41170, Taiwan

² Department of Electronic Engineering, Chienkuo Technology University, Changhua 500, Taiwan

³ Department of Industrial Education and Technology, National Changhua University of Education, Changhua 500, Taiwan

⁴ Department of Power Mechanical Engineering, Army Academy, Zhongli 32093, Taiwan

⁵ Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei 10608, Taiwan

⁶ Department of Electro-Optical and Energy Engineering, MingDao University, Changhua 52345, Taiwan

Correspondence should be addressed to Koan-Yuh Chang; kychang@cc.ctu.edu.tw

Received 26 February 2014; Accepted 13 April 2014; Published 14 May 2014

Academic Editor: Her-Terng Yau

Copyright © 2014 Weng-Ming Chu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The determination of the critical path (CP) in stochastic networks is difficult. It is partly due to the randomness of path durations and partly due to the probability issue of the selection of the critical path in the network. What we are confronted with is not only the complexity among random variables but also the problem of path dependence of the network. Besides, we found that CP is not necessarily the longest (or shortest) path in the network, which was a conventional assumption in use. The Program Evaluation and Review Technique (PERT) and Critical Path Index (CPI) approaches are not able to deal with this problem efficiently. In this study, we give a new definition on the CP in stochastic network and propose a modified label-correcting tracing algorithm (M-LCTA) to solve it. Based on the numerical results, compared with Monte Carlo simulation (MCS), the proposed approach can accurately determine the CP in stochastic networks.

1. Introduction

The CP does not only help managers in making decision for optimal resource allocation but also help in doing network sensitivity analysis. Although the conventional critical path method (CPM) can be used to solve CP very easily [1], CPM does not work properly in stochastic networks due to the reason that the durations of activities and paths are all random variables. The values of random variables are in the form of probability, which lets the selection of CP become very difficult to solve (refer to [2, 3]). The well-known and popular PERT model did not solve it as well according to [4]. In other cases, some papers use the method of CPI, where [5] is the early one involved, [6, 7] used “crude MCS” and “conditional MCS” methods to simulate CPI, and [8–11] approximated CPI through complex mathematical analysis. No matter what the results of the above approaches are,

the CP is not any more the path that has the most important priority to accept the rare allocated resources in the network. This will be explained later in this study. Besides, none of the above papers can efficiently afford big-size network which is an important issue in practical applications.

For approximating the critical path duration, we find that [12, 13] have just only aimed at the mean of duration for the shortest path (SP) instead of procuring the values of probability distribution. Moreover, [14–16] have applied the way of simplifying the stochastic network into a deterministic network before solving the SP. In other cases, the MCS was suggested and [17] used the discrete Markov chain to deal with SP.

Most of the above studies are either too simple or gotten with the heavy calculations. Further, [18] has indicated that the CP will be more complex by taking path dependency into consideration. In this study, we have also found that the CP is

not necessarily the longest (or shortest) path in the stochastic network, as what the project managers suppose in ordinary. In this study, the CP is redefined as follows.

Definition 1. The CP should have dominated the completion time of the network.

Definition 2. The duration perturbation of activities included in CP would be more sensitive to the change of completion time than that of other activities.

A new approach of the modified M-LCTA with the revised stochastic dominance operation (RSDO) is developed in this study, where the detailed descriptions will be given in the following sections. The rest of this paper is organized as follows. Section 2 discusses how difficult it is to solve the CP in stochastic networks. Section 3 provides detailed explanations for M-LCTA and RSDO. Section 4 demonstrates the proposed algorithm via two examples. Section 5 gives further numerical experiments of various large sized network instances to illustrate the feasibility of the proposed approach for CP. Finally, the concluding remarks are presented in Section 6.

2. The Difficulty of Solving Critical Path in Stochastic Network

The CP of stochastic network is not as instinctive and easy as using CPM in deterministic network for the sake of five reasons listed as follows.

- For solving CP, it has to approximate all completion paths' durations in the network.
- In general, the selection of the CP depends on the sorting of all the network completion paths. The sorting has to be executed through the max or min operations among the probability distribution functions (PDF) of path durations, instead of comparing the paths' mean value of duration (such as PERT model).
- There is a bias between the results of PERT model and max or min operations among random variables. In this study, the bias is named as the longest path bias (LPB) or the shortest path bias (SPB) in respect to the max or min operations. They have to be considered during the network analysis.
- The path dependence caused by the shared subpath in the network generates a bias, too. It is rarely noticed and no solutions have been found yet in the present paper.
- As mentioned in the above section, each completion path could have the probability to be CP in stochastic network. Therefore, it needs to have a criterion for comparing the probability values of all completion paths and to select one of them as CP.

According to the above reasons, there are more detailed explanations presented in the following subsections.

FIGURE 1: The network with four nodes.

2.1. The Approximation of Path Duration. In the network, the realization of path duration time is conducted through series of convolution and max operations (or min operations) over the nodes in the network. By taking node (j) , that is, γ_j as an example, the calculations are represented as follows:

$$\begin{aligned}
 Y_i^j &= \gamma_i \oplus X_{ij}, \quad i \in B_j, \\
 \Pr(Y_i^j) &= \int_{-\infty}^{\infty} g_i(t - \tau) f_{ij}(\tau) d\tau, \\
 \gamma_j &= \max \{Y_1^j, Y_2^j, \dots, Y_i^j\}, \\
 \Pr(\gamma_j \geq x) &= \Pr(Y_1^j \geq x, Y_2^j \geq x, \dots, Y_i^j \geq x) \\
 &= \Pr(Y_1^j \geq x) \cdot \Pr(Y_2^j \geq x) \cdots \Pr(Y_i^j \geq x), \\
 q &= |B_j|,
 \end{aligned} \tag{1}$$

where the symbol $\oplus$ denotes the convolution operation and $\Pr(\cdot)$ denotes the probability value of the internal function. Y_i^j is the i th incident path of node (j) , B_j is the precedence nodes set of node (j) , and $g_i(\cdot)$ and $f_{ij}(\cdot)$ denote the PDF of γ_j and X_{ij} , respectively. It is well known that the estimation of the PDF for the completion time of a project is a nonpolynomial hard problem (please, see [3] for further reference).

Over the past double decades, Dodin's research [20] was a rare one which can afford a feasible algorithm, denoted as Dodin algorithm (DA), to solve network completion time and has been applied to big-size network. DA used the discretization technique to "discretize" the PDF of each node's duration in the network first; it then executed the convolution and max operations consecutively on nodes so as to delete the LPB problem. Finally DA approximates the PDF of the network completion time. By taking Figure 1 as an example, the iterative steps of the DA are as follows:

$$\begin{aligned}
 \gamma_2 &= X_{12}, \\
 Y_2^3 &= \gamma_2 \oplus X_{23}, \\
 \gamma_3 &= \max \{X_{13}, Y_2^3\}, \\
 Y_2^4 &= \gamma_2 \oplus X_{24} \text{ (1-2-4)}, \\
 Y_3^4 &= \gamma_3 \oplus X_{34} \text{ (1-3-4)}, \\
 \gamma_4 &= \max \{X_{14}, Y_2^4, Y_3^4\} \text{ (1-2-3-4)}.
 \end{aligned} \tag{2}$$

Though DA provides an elegantly simple and intuitively appealing procedure for dealing with the realization times of paths in stochastic network, there are still two shortages. First, there exists neither an exact data structure nor implementation scheme for the computer programming. Second, an approximated error may arise from assuming independence between subpaths in the activity network; such a bias will be getting larger when increasing the network size [19].

2.2. The Sorting among Network Paths. In the stochastic network, all the completion paths could have their probabilities to be selected as the CP. Therefore, one has to offer a criterion to judge and select the highest probability of completion paths to be the CP from all the paths. Dodin algorithm [20] has suggested the stochastic dominance operation (SDO) for two random variables' comparison. By taking variables Y_1^j and Y_2^j as examples, if Y_1^j dominates Y_2^j (represented as $Y_1^j > Y_2^j$), it means that Y_1^j has a greater possibility than Y_2^j to be the longest path or the shortest path in the network. Their equation relations are represented as follows:

$$\Pr(Y_1^j > Y_2^j) \geq \Pr(Y_2^j > Y_1^j). \quad (3)$$

The SDO just operates for only two variables, because of the problem of the variables transitivity. It means that if we have $Y_1^j > Y_2^j$ and $Y_2^j > Y_3^j$, it does not necessarily imply $Y_1^j > Y_3^j$. In this study, transitivity problem will be solved by the proposed modified SDO in the following section.

2.3. The Longest Path Bias and Shortest Path Bias. Let $Y_1^j \sim Y_m^j$ represent totally m incident paths of node (j) and γ_j the output duration of node (j) . According to the PERT model, γ_j is secured as γ_j^{PERT} and calculated by picking the largest mean value of paths $Y_1^j \sim Y_m^j$, where $E[\gamma_j^{\text{PERT}}] = \max(E[Y_1^j], E[Y_2^j], \dots, E[Y_m^j])$. But in mathematical realization, the real mean value of γ_j should be calculated as $E[\gamma_j^{\text{Real}}] = E[\max(Y_1^j, Y_2^j, \dots, Y_m^j)]$, where the max operation is executed among all incident paths $Y_1^j \sim Y_m^j$, $E[\gamma_j^{\text{Real}}]$ is usually greater than $E[\gamma_j^{\text{PERT}}]$ and their deviation is named as the longest path bias (LPB).

2.4. The Bias of Path Dependency among Subpaths. The paths in a stochastic network are not necessarily independent, since they usually share activities with each other. This may also cause significant bias in approximation of network completion time. However, most studies did not consider such factor in their approaches. In Figure 2(a), the paths $\langle 1, 2, 3, 4, 5, 7 \rangle$ and $\langle 1, 2, 3, 4, 6, 7 \rangle$ are assumed to be independent with each other; then we have $\gamma_4 = X_{12} \oplus X_{23} \oplus X_{34}$, $\gamma_5 = \gamma_4 \oplus X_{45}$, $\gamma_6 = \gamma_4 \oplus X_{46}$, $Y_1^7 = \gamma_5 \oplus X_{57}$, $Y_2^7 = \gamma_6 \oplus X_{67}$, and $\gamma_7 = \max\{Y_1^7, Y_2^7\}$. With the above numerical calculations, the expected duration of γ_7 is solved as $E[\gamma_7] = 17.69$ which is different from the result $E[\gamma_7] = 15.89$ simulated out of MCS within 20,000 samples (the result of MCS could be taken as the nearly real solution). The above deviation is caused by the

FIGURE 2: A simple network example.

FIGURE 3: A simple network example.

shared subpath $\langle 1, 2, 3, 4 \rangle$ between the paths $\langle 1, 2, 3, 4, 5, 7 \rangle$ and $\langle 1, 2, 3, 4, 6, 7 \rangle$. We named it as the shared activity bias (SAB) in this study.

For further comprehension of the causes of SAB, from Figure 2(b), $\langle 1, 2, 3, 4 \rangle$ and $\langle 1', 2', 3', 4' \rangle$ are shared paths, where $X_{12} = X_{1'2'}$, $X_{23} = X_{2'3'}$, and $X_{34} = X_{3'4'}$. From the view of DA, subpaths $\langle 1, 2, 3, 4 \rangle$ and $\langle 1', 2', 3', 4' \rangle$ are treated as two individual subpaths in the network. The max operation executed on node (7) will cause extra LPB from subpath $\langle 1', 2', 3', 4' \rangle$, since the duration variance of the shared path $\langle 1-2-3-4 \rangle$ is double-counted by subpath $\langle 1', 2', 3', 4' \rangle$ in the max operation. It brings more to the overlapping domain of variable distribution between subpaths $\langle 1, 2, 3, 4 \rangle$ and $\langle 1', 2', 3', 4' \rangle$.

2.5. A New Criterion for CP Judgment. The PERT model and the method of CPI do not properly determine the CP in stochastic network. Taking Figure 3 as an example, there are three completion paths represented as $\langle 1, 2, 5, 7 \rangle$, $\langle 1, 3, 6, 7 \rangle$, and $\langle 1, 4, 6, 7 \rangle$. The durations of activities are assumed as exponential distribution where their expected values are shown in the graph of Figure 3.

By using the PERT model, $\langle 1, 2, 5, 7 \rangle$ is taken as the CP since its expected path duration is 12 which is greater than those of the other paths. Accordingly, the expected completion time of network is calculated as 12, too. On the other hand, the authors use MCS within 20000 samples to simulate the CPI of all paths that indicate the possibilities of how the path can become the critical path of the network. The results

are as follows: $\langle 1, 2, 5, 7 \rangle$ CPI = 0.426, $\langle 1, 3, 6, 7 \rangle$ CPI = 0.286, and $\langle 1, 4, 6, 7 \rangle$ CPI = 0.288. From the results of MCS, the path $\langle 1, 2, 5, 7 \rangle$ is still to be taken as the CP due to its highest CPI value. But, the completion time of MCS is simulated as $E[\gamma_7] = 16.62$, which is different from the result of PERT. For further comprehension of above deviation, the authors check the expected output of node (6) simulated as $E[\gamma_6] = 9.86$ and $E[Y_1^7] = E[\gamma_6] + E[X_{67}] = 12.86$, which is greater than the expected output value of path $\langle 1, 2, 5, 7 \rangle$. Obviously, the above result has shown that the network completion time is not dominated by path $\langle 1, 2, 5, 7 \rangle$ but by a two-pair path, $\langle 1, 3, 6, 7 \rangle$ and $\langle 1, 4, 6, 7 \rangle$, which is named as incidence path clusters (IPC) in this study, even though the path $\langle 1, 2, 5, 7 \rangle$ has the highest CPI value.

From above experiment results, it seems that we should need a new criterion to judge CP in stochastic network. The CP does not only have to do with the variables sorting, LPB (or SPB) and SAB in stochastic network but also need to develop an implementation algorithm so as to efficiently execute all the max and convolution operations of nodes in the network.

3. The Proposed Algorithm for Critical Path in a Stochastic Network

A new proposed algorithm named as the modified M-LCTA is developed for CP in stochastic network. The label-correcting tracing algorithm (LCTA) is originally developed for approximating the completion time of stochastic network [19]. In this study, it is added with a RSDO for executing the paths sorting, so as to solve the CP.

LCTA can efficiently get rid of the LPB, SPB, and SAB that are mentioned in the last section during the approximation, something that no other approaches can deal with properly. There are more detailed explanations and many practical experiments for comparing various methods with LCTA in [19]. Due to space limitations, the comparisons are abridged in this paper.

3.1. The Expanded-Tree Structure and Node Data Structure. The M-LCTA has to first transfer the physical network into a tree structure named as expanded-tree structure (ETS) according to the connected relation between the nodes in the network. Each node in the network has a unique index number i where node (1) represents the start node and node (n) represents the terminal node (there are a total n number of nodes in the network). Figure 4 is an example of the ETS.

M-LCTA has to visit each node in sequence and commit calculations; the generated results during the processes have to be saved in a specified data structure in the nodes of ETS. For the objective of this research, the structure of node (i) includes the contents as follows:

i : index number of node (i);

B_i : set of precedent nodes of node (i);

output _{i} : the realization time of output of node (i);

FIGURE 4: The tree structure transformation.

finish_flag _{i} : visited flag of node (i) where it is initially set as 0; when node (i) has been visited, it will be set as 1;

node (i).flag: visited-flag set of incident paths of node (i) where all are initially set as 0; when the k th incident path of node (i) has finished its time realization, the corresponding k th position of node (i).flag will be set as 1;

node (i).Inpath: for each incidence path of node (i), it includes a sequence of nodes from node (1) to node (i); the realization time of incidence paths will be saved in node (i).Inpath.

3.2. The Two Tracing Procedures in M-LCTA. When applying the M-LCTA, the authors employ two tracing procedures, namely, Upward_Tracing (last_node, current_node) and Downward_Tracing (current_node), where the current_node is the node currently visited in the tracing process and the last_node is the precedent node visited in the current process. These two procedures are listed as shown in Algorithm 1.

Initially, the M-LCTA executes Downward_Tracing(n) and starts tracing from the terminal node (n). Then, it goes downward until it reaches node (1). It, then, traces upward and starts node-visiting and calculations. The labels of visited node are put into stack where data access follows the rule of First In/First Out (FIFO). Labels recording and reading are denoted as *push_stack()* and *pull_stack()* in the procedures, respectively. The stack is used for recording the path-track of visited nodes by which the tracing procedures can judge and find their ways to the nodes they have not visited yet.

M-LCTA can sequentially visit each node in the network. According to the post-order tracing algorithm (POTA) rule, “before visiting the node (i), it has to confirm that all the child nodes of node (i) have been visited. The visits of child nodes should be in sequence following the left-to-right direction.”

Using Figure 1 as an example, we first change the network into a tree-structure and the sequence for visiting the nodes is shown in Figure 5. The tracing procedures start from terminal node (4), and, through recursive executions of downward and upward procedures, the visiting sequence of M-LCTA is shown in blue arrow line. Finally, it stops and backs to terminal node (4).


```

Downward_Tracing (current_node)
i = current_node;
while ( $\gamma_N$  has not been calculated) do
  if (finish_flagi = 1)
    Shared_flag = 1;
    k = current_node;
    i = pull_stack();
    perform Upward_Tracing(k, i);
  Else
    if (i = 1?)
       $\gamma_1$  = 0;
      finish_flagi = 1;
      k = i;
      i = pull_stack();
      perform Upward_Tracing(k, i);
    Else
      push_stack(i);
      i = k;
    End
  End
End

Upward_Tracing (last_node, current_node )
k = last_node;
i = current_node;
while ( $\gamma_N$  not been calculated) do
  if (Has node(k) been visited?)
     $Y_k^i = \begin{bmatrix} \mu(\gamma_k) \\ 1 \end{bmatrix} \oplus X_{ki}$ ; // an constant mean  $\mu(\gamma_k)$ 
    Shared_flag = 0;
  Else
     $Y_k^i = \gamma_k \oplus X_{ki}$ ;
    Shared_flag = 0;
  End
  if (finish_flagi = 1)
     $W^i = \text{RSDO}(i) = \{Y_k^i \mid \forall k \in B_i\}$ ;
     $\gamma_i = \min(Y_k^i)$  or  $\gamma_i = \max(Y_k^i)$ ;
    Shared_flag = 0;
    k = i;
    i = pull_stack();
  Else
    push_stack(i);
    perform Downward_Tracing(i);
  End
End

```

ALGORITHM 1

3.3. The Function of Eliminating Shared Activity Bias. *Shared_flag* is a Boolean variable in both tracing procedures for identifying shared activities between paths in the network. During the implementation of the M-LCTA, the Upward_Tracing procedure first checks the value of the *Shared_flag* for the current node. If the current node has been previously visited and its realization time has been obtained, the *Shared_flag* will be set as 1 by the Downward_Tracing procedure. Using the values of the *Shared_flag*, procedures can find all of the activities shared by the subpaths in the network. Once the shared activities have been found, the SAB can be eliminated by applying a heuristic to “correct”

FIGURE 5: The tracing sequence.

the computations in the max operation; the PDF of the redundant shared activities is replaced with the value of their mean (which is a fixed constant) in the Upward_Tracing procedure. For more details of the SAB, one could refer to the experiments in [19].

3.4. The Function of Finding the Critical Path. The operation of RSDO in the algorithms is responsible for sorting the sequence order of paths. It is executed in M-LCTA during the node visiting. The processes are described below.

In the Upward_Tracing procedure, if all the child nodes of the current node *node* (*i*) have been visited, *node* (*i*) will execute the RSDO and max (or min operation) operations among its incidence paths as follows:

- $W^i = \text{RSDO}(i) = \{Y_k^i \mid \forall k \in B_i\}$ of *node* (*i*) is to rearrange the child-node position of *node* (*i*) according to the results of $\text{RSDO}(i)$. The highest value $\text{RSDO}(i)$ of child node will be put on the extreme right side of child-node position.
- $\gamma_i = \min(Y_k^i)$ or $\gamma_i = \max(Y_k^i)$, for all $k \in B_i$, is executed for the shortest path or the longest path, respectively. These calculations can release SPB or LPB in the approximation. The result γ_i is saved in the output_{*i*} of *node* (*i*).
- Otherwise, the tracing procedures will execute $Y_k^i = \gamma_k \oplus X_{ki}$ to accumulate the *k*th incidence path of *node* (*i*) and save Y_k^i in the *node* (*i*). Inpath_{*k*} of *node* (*i*). Then it goes back in the Downward_Tracing procedure.

The executions of max (or min) operation and convolution in M-LCTA apply techniques of discretization. M-LCTA first needs to replace the continuous PDF of X_{ij} with its discretized version using the approach of Chebychev sampling points [21]. The discretized PDF is represented by an ordered vector as:

$$\begin{pmatrix} v_1^{ij} & v_2^{ij} & \cdots & v_m^{ij} \\ p_1^{ij} & p_2^{ij} & \cdots & p_m^{ij} \end{pmatrix}, \quad (4)$$

where v_k^{ij} and p_k^{ij} are the values of the *k*th sample point of X_{ij} and its corresponding probability, respectively, and *m* is the

total number of sample points. Equations (1) executed in the algorithm are all calculated in the form of discretization (see details in [19]).

Through the iterative executions of the Downward Tracing and Upward Tracing procedures, the M-LCTA will finally visit all the nodes and stop at the terminal node node (n) . Meanwhile, all incidence paths of nodes have rearranged their position sequence of child node according to the sorting operation of RSDO in the procedures, so that the CP can be found.

3.5. The Revised Stochastic Dominance Operator (RSDO).

RSDO is different from Dodin's SDO in solving the transitivity problem [15]. The RSDO determines the sorting order of random variables through matrix operations. First, it has to construct an $n \times n$ pairwise comparison matrix (PCM), represented as P for n stochastic variables Z ($Z = (z_1, z_2, \dots, z_n)$) and shown in

$$P = \begin{bmatrix} 1 & a_{12} & \cdots & a_{1n} \\ a_{21} & 1 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \frac{w_1}{w_1} & \frac{w_1}{w_2} & \cdots & \frac{w_1}{w_n} \\ \frac{w_2}{w_1} & \frac{w_2}{w_2} & \cdots & \frac{w_2}{w_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{w_n}{w_1} & \frac{w_n}{w_2} & \cdots & \frac{w_n}{w_n} \end{bmatrix}, \quad (5)$$

where $a_{ij} = z_i/z_j = w_i/w_j$ and $W = (w_1, w_2, \dots, w_n)$ is the ideal-sorting ratio within Z variables. The sorting order of Z variables could be judged by the values of W . If all variables fulfill the characteristics of transitivity, each pair of variables should follow the equation below:

$$\begin{aligned} a_{ij} \cdot a_{jk} &= \left(\frac{w_i}{w_j} \right) \cdot \left(\frac{w_j}{w_k} \right) \\ &= \left(\frac{w_i}{w_k} \right) \\ &= a_{ik}, \quad i, j, k \in [1, \dots, n]. \end{aligned} \quad (6)$$

In the case of (5), the matrix P is called the consistency matrix and the W can be directly calculated as follows:

$$\begin{aligned} \because \sum_{j=1}^n a_{ij} w_j \frac{1}{w_j} &= n, \quad a_{ij} \cdot \frac{w_j}{w_i} = 1, \\ \sum_{j=1}^n a_{ij} w_j &= n w_i, \quad i = 1, \dots, n, \\ PW &= nW, \quad W = (w_1, \dots, w_n)^T. \end{aligned} \quad (7)$$

The n and $(w_1, w_2, \dots, w_n)$ could be solved as the eigenvalue and eigenvector of P , respectively. In other cases, if any one of the elements in P does not fulfill the condition of (5), W cannot be taken as the solution. The eigenvalue of P would be changed to n' and the eigenvector $W' = (w'_1, w'_2, \dots, w'_n)$ is used as the approximation of W .

By referring to (3) and (5), the element a_{ij} of the PCM could be calculated between variables z_i and z_j , $i, j \in [1, n]$, where

$$\begin{aligned} a_{ij} &= \Pr(z_i > z_j), \\ a_{ji} &= \Pr(z_j > z_i) = 1 - \Pr(z_i > z_j). \end{aligned} \quad (8)$$

Therefore, PCM can be represented as P' ,

$$\begin{aligned} P' &= \begin{bmatrix} 1 & a_{12} & \cdots & a_{1n} \\ (1 - a_{12}) & 1 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ (1 - a_{1n}) & (1 - a_{2n}) & \cdots & 1 \end{bmatrix} \\ &\neq \begin{bmatrix} \frac{w_1}{w_1} & \frac{w_1}{w_2} & \cdots & \frac{w_1}{w_n} \\ \frac{w_2}{w_1} & \frac{w_2}{w_2} & \cdots & \frac{w_2}{w_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{w_n}{w_1} & \frac{w_n}{w_2} & \cdots & \frac{w_n}{w_n} \end{bmatrix}, \\ &0 < a_{ij} < 1. \end{aligned} \quad (9)$$

From (9), the lower triangle of P' does not fulfill the pattern of (6). It leads W not to be represented as the sorting ratio of Z . Therefore, P' needs to be regulated before calculating its eigenvalue and eigenvector. The elements of P' are regulated as below:

$$\frac{w'_i}{w'_j}, \quad \text{where } w'_i = a_{ij}, \quad w'_j = a_{ji} = (1 - a_{ij}). \quad (10)$$

The regulation will not ruin the sorting ratio of Z , since it just changes the complementary relations (refer to (8)) into the proportional relations of variables. After the regulation, P' is changed to P'' and is represented as follows:

$$P'' = \begin{bmatrix} 1 & \frac{w'_1}{w'_2} & \cdots & \frac{w'_1}{w'_n} \\ \frac{w'_2}{w'_1} & 1 & \cdots & \frac{w'_2}{w'_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{w'_n}{w'_1} & \frac{w'_n}{w'_2} & \cdots & 1 \end{bmatrix}. \quad (11)$$

The eigenvalue of P'' is calculated as the sorting sequence ratio of incidence paths.

By taking Figure 6 as an example, node (i) executes RSDO on four incidence paths $Y_1^i, Y_2^i, Y_3^i, Y_4^i$.

According to (8), we calculate stochastic dominate probability value (SDPV) for each incidence path $Y_1^i, Y_2^i, Y_3^i, Y_4^i$, where we have

TABLE 1: The comparison of running time between the proposed method (M-LCTA) and MCS.

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15 (100 nodes)
M-LCTA	0.003	0.006	0.01	0.012	0.015	0.018	0.019	0.021	0.031	0.028	0.034	0.079	0.099	0.132	0.31
MCS	0.8	1.5	2.3	3.3	4.4	4.5	5.2	6.4	6.7	9.2	10.7	17.7	44.6	199.8	1840.6

FIGURE 6: Four incident paths of node (i).

TABLE 2

FIGURE 7: The 9 nodes of demonstrative example.

The shortest path: $\langle 1-3 \rangle$ The longest path: $\langle 1-2-3 \rangle$

Index	1	2	3
Mean	5.0	10.0	14.3

TABLE 3

The shortest path: $\langle 1-2-4 \rangle$ The longest path: $\langle 1-2-3-4 \rangle$

Index	1	2	3	4	5
Mean	50.0	33.3	25.0	41.7	40.0

$$\begin{aligned}
 a_{12} &= \Pr(Y_1^i > Y_2^i) = 0.2, & a_{21} &= 1 - a_{12} = 0.8, \\
 a_{13} &= \Pr(Y_1^i > Y_3^i) = 0.15, & a_{31} &= 1 - a_{13} = 0.85, \\
 a_{14} &= \Pr(Y_1^i > Y_4^i) = 0.123, & a_{41} &= 1 - a_{14} = 0.877, \\
 a_{23} &= \Pr(Y_2^i > Y_3^i) = 0.248, & a_{32} &= 1 - a_{23} = 0.752, \\
 a_{24} &= \Pr(Y_2^i > Y_4^i) = 0.2, & a_{42} &= 1 - a_{24} = 0.8, \\
 a_{34} &= \Pr(Y_3^i > Y_4^i) = 0.333, & a_{43} &= 1 - a_{34} = 0.667.
 \end{aligned} \tag{12}$$

By referring to (10), we have P'' as

$$P'' = \begin{bmatrix} 1 & 0.25 & 0.17 & 0.14 \\ 4 & 1 & 0.33 & 0.25 \\ 6 & 3 & 1 & 0.5 \\ 7 & 4 & 2 & 1 \end{bmatrix}. \tag{13}$$

The eigenvalue of P'' is calculated as $n'' = 4.102$ which is close to $n = 4$, the eigenvector is calculated as the $W' = (0.617, 0.224, 0.097, 0.062)$ which is the sorting ratio of incidence paths. According to the W' and sorting rule of M-LCTA, the Y_1^i should be put on the extreme right side of child-node position, where we have a sequence order of $(Y_4^i, Y_3^i, Y_2^i, Y_1^i)$.

To represent the M-LCTA procedure scheme conveniently, we define the above operation as $W' = \text{RSDO}(i)$ to show the execution of RSDO on the node (i) and produce the sorting result W' from its incidence paths.

4. Demonstrative Examples

By taking the network of Figure 7 as the demonstrative example, there are a total of 9 nodes and all the activity durations are assumed to be exponential distributions where their duration means are shown in the figure.

There are three completion paths included: $\langle 1-2-3-9 \rangle$, $\langle 1-4-5-6-8-9 \rangle$, and $\langle 1-4-5-7-8-9 \rangle$ where the path $\langle 1-2-3-9 \rangle$ is taken as the shortest path from the viewpoint of the PERT model. In another method, MCS within 20,000 samples is used to simulate the same network (MCS result could be taken as the nearly real solution) and the result is different from that of PERT. This shows obviously that PERT cannot solve CP correctly in stochastic network.

Let us apply our method to the same example. It has to transfer the network into a tree structure first, as shown in Figure 8. M-LCTA starts the Downward Tracing (9) at node (9). Downward Tracing and Upward Tracing procedures are executed iteratively with each other. Finally, it calculates the expected completion time as $\gamma_9 = 7.12$ from node (9), which is very close to the simulated result of the MCS.

During the executions, $W^8 = \text{RSDO}(8) = \{Y_k^8 \mid \forall k \in B_8\}$ and $\min(Y_1^8, Y_2^8)$ are performed at node (8). We find that the two incidence paths have to swap their child-node position because of the sorting result of the RSDO. The modified M-LCTA finally stops at node (9) and executes $W^9 = \text{RSDO}(9) = \{Y_k^9 \mid \forall k \in B_9\}$ and $\min(Y_1^9, Y_2^9)$. The result of the RSDO (9)

FIGURE 8: The tree structure of Figure 3.

TABLE 4

The shortest path: $\langle 1-3-5 \rangle$ The longest path: $\langle 1-2-3-4-5 \rangle$

Index	1	2	3	4	5	6	7
Mean	12.5	16.7	11.1	20.0	14.3	33.3	25.0

TABLE 5

The shortest path: $\langle 1-2-4-6 \rangle$ The longest path: $\langle 1-6 \rangle$

Index	1	2	3	4	5	6	7
Mean	25.0	100.0	14.3	28.6	20.0	16.7	22.2
Index	8	9					
Mean	16.7	25.6					

shows that the incidence paths need not to change their child-node position at node (9). After the repositioning of all incident paths in the network, Figure 8 gives the final result. The path $\langle 1-4-5-7-8-9 \rangle$ is selected as the critical path, which lays on the extreme right side of the tree.

Finding the longest path is the same as above, except that the min operator is replaced by the max operator at node (8) and node (9).

TABLE 6

The shortest path: $\langle 1-4-6 \rangle$ The longest path: $\langle 1-2-3-4-5-6 \rangle$

Index	1	2	3	4	5	6	7
Mean	10.0	11.1	2.5	5.0	3.3	12.5	2.5
Index	8	9	10	11			
Mean	5.0	10.0	3.3	3.3			

TABLE 7

The shortest path: $\langle 1-2-5-7 \rangle$ The longest path: $\langle 1-2-3-6-7 \rangle$

Index	1	2	3	4	5	6	7
Mean	10.0	8.3	20.0	12.5	5.0	25.0	33.3
Index	8	9	10	11			
Mean	25.0	41.7	6.7	6.3			

From the above experiments, it is noticed that the completion time of network is actually dominated by one of IPC of terminal node, instead of the result of PERT model or the highest CPI. The domination on the network completion time should be the key characteristic for CP, so as to let CP be the right target for project managers to allocate the resources.

5. Numerical Experiments

This section will verify the efficiency and accuracy of our proposed approach on various sizes of stochastic networks, especially the larger ones. The first part of network instances is generated from [22], where there are a total of 14 instance networks and the activity number ranges from 3 to 76. All activities durations are also assumed to have exponential distribution. Their parameters, information, and experimental results are shown in the appendix.

FIGURE 9: The 100-node network instance of [19].

TABLE 8

The shortest path: $\langle 1-2-7-8 \rangle$ The longest path: $\langle 1-2-4-6-8 \rangle$

Index	1	2	3	4	5	6	7
Mean	10.0	11.1	12.5	10.0	11.1	12.5	10.0
Index	8	9	10	11			
Mean	12.5	10.0	11.1	10.0			

TABLE 9

The shortest path: $\langle 1-2-7 \rangle$ The longest path: $\langle 1-3-4-5-7 \rangle$

Index	1	2	3	4	5	6	7
Mean	5.0	4.0	6.3	5.0	10.0	6.3	2.0
Index	8	9	10	11	12	13	14
Mean	4.0	5.0	12.5	11.1	10.0	8.0	10.0

The second part is a network of 100 nodes and 135 activities generated by [19]. It was designed by following the rules of [23], where the network complexity factor was set to 2.0. For each node, the incidence path number was randomly determined from a range of [1, 2] and the PDF was determined randomly from normal, exponential, or uniform distribution. The correspondent parameters of PDF were also randomly selected within the range of [2, 16]. Figure 9 shows its detailed structure and experiment results as follows.

The shortest path:

$\langle 1-3-5-10-15-23-33-55-68-78-86-92-97-99-100 \rangle$.

The longest path:

$\langle 1-2-4-8-13-20-30-42-54-63-76-84-90-96-98-100 \rangle$.

Since the simulated results of MCS are close to the real solution (please, refer to [7, 20]), the authors employ the MCS within 20,000 samples as the benchmark and simulate

TABLE 10

The shortest path: <1-4-7-10>

The longest path: <1-2-5-9-10>

Index	1	2	3	4	5	6	7
Mean	50.0	33.3	25.0	40.0	28.6	22.2	20.0
Index	8	9	10	11	12	13	14
Mean	16.7	33.3	50.0	66.7	50.0	40.0	33.3

TABLE 11

The shortest path: <1-2-4-7-10>

The longest path: <1-2-4-5-7-8-10>

Index	1	2	3	4	5	6	7
Mean	6.0	10.0	5.0	10.0	4.0	5.0	10.0
Index	8	9	10	11	12	13	14
Mean	3.0	3.0	4.0	2.0	6.0	7.0	2.0
Index	15	16	17				
Mean	8.0	6.0	9.1				

all the network instances twice: one for the longest path and the other for the shortest path. The MCS simulations focus on checking the accuracy of RSDO results and the output duration of each node in the networks. After completing all experiments, we have excellent performances that the experiment networks are all matching the results of MCS. Beyond that, the efficiency of proposed method is also extraordinarily superior to that of MCS (see Table 1). According to the network of 100 nodes, MCS requires 1840.6 seconds to complete the simulation, while our method takes only 0.31 seconds.

TABLE 12

The shortest path: <1-4-7-13-14>

The longest path: <1-3-5-9-11-12-14>

Index	1	2	3	4	5	6	7
Mean	16.7	25.0	10.0	14.3	12.5	25.0	12.5
Index	8	9	10	11	12	13	14
Mean	5.0	14.3	20.0	12.5	14.3	11.1	11.1
Index	15	16	17	18			
Mean	20.0	11.1	25.0	16.7			

TABLE 13

The shortest path: <1-2-5-13-17>

The longest path: <1-2-5-7-11-16-17>

Index	1	2	3	4	5	6	7
Mean	40.0	50.0	45.0	54.9	30.0	35.0	40.0
Index	8	9	10	11	12	13	14
Mean	25.0	30.0	40.0	50.0	26.0	28.0	20.0
Index	15	16	17	18	19	20	21
Mean	25.0	30.0	20.0	18.0	36.0	45.0	15.0
Index	22	23	24				
Mean	59.9	25.0	46.1				

6. Conclusions

Since activity duration and path duration are both random variables, the methods employed for deterministic networks cannot be used in this study. In addition, the popular PERT model and CPI method do not work either, since they ignore the effects of the LPB, SPB, and SAB which have to be considered in the stochastic networks. That is why we modify the LCTA of [19] by adding the revised stochastic dominance

TABLE 14

The shortest path: <1-2-3-16-20>

The longest path: <1-2-3-5-8-12-15-18-20>

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean	16.0	30.0	16.0	20.0	10.0	18.0	15.0	11.0	12.0	40.0	20.0	16.0	10.0	8.0	1.4
Index	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Mean	21.0	16.0	20.0	19.0	17.0	15.0	18.0	21.0	16.0	20.0	18.0	16.0	12.0	18.0	15.0
Index	16	31	32	33	34	35	36	37	38						
Mean	30.0	24.0	11.0	11.0	22.0	24.0	30.0	15.0	8.0						

operation (RSDO) in this research and employ it to solve CP in stochastic networks. In the experiments of this study, we have various sizes of instance networks from [19, 22], where the activity numbers ranged from 3 to 135. According to the experimental results, they all comply with the simulation results of the MCS within 20,000 samples. Beyond that, the efficiency of the proposed method is extraordinarily great. As shown in Table 1, it is almost 780 times faster than that of MCS in average. Therefore, according to the overall performance, we confirmed that the proposed algorithm in this study is an excellent tool for solving CP in stochastic networks.

In addition, this study has also proved and pointed out that the CP generated from traditional methods (i.e., PERT or CPI methods) is not necessarily the key path in the network. The CP should follow the spirit of Definitions 1 and 2, which is “the critical path of stochastic network should be highest influence of dominating the network completion time.” With the above assertion, it could help the project managers to keep off allocating the precious resources into null and void activities in the network and that would also be the main contribution in this study.

Appendix

Network 1: see Table 2.

Network 2: see Table 3.

Network 3: see Table 4.

Network 4: see Table 5.

Network 5: see Table 6.

Network 6: see Table 7.

Network 7: see Table 8.

Network 8: see Table 9.

Network 9: see Table 10.

Network 10: see Table 11.

Network 11: see Table 12.

Network 12: see Table 13.

Network 13: see Table 14.

Network 14: see Table 15.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

TABLE 15

The shortest path: <1-3-5-12-20-27-30-35-36>
The longest path: <1-4-6-13-16-22-28-34-36>

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean	16.7	12.5	14.3	2.0	5.0	10.0	11.1	1.7	3.3	14.3	12.5	2.9	3.8	2.3	11.8
Index	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Mean	40.0	1.8	13.3	10.5	6.7	4.0	6.7	2.9	2.5	2.7	2.1	11.8	12.5	6.7	4.0
Index	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
Mean	4.0	2.9	2.5	14.3	1.7	6.7	4.0	6.7	15.4	13.3	6.7	4.5	2.9	1.9	14.3
Index	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
Mean	2.9	8.3	14.3	7.7	4.0	2.9	2.2	4.8	11.9	5.0	2.2	1.8	11.1	8.3	5.6

Acknowledgment

This research was supported in part by the National Science Council of Taiwan, under the Grant no. NSC 102-2221-E-270-005.

References

[1] S. E. Elmaghraby, *Activity Networks: Project Planning and Control by Network Models*, John Wiley & Sons, New York, NY, USA, 1977.

[2] V. G. Adlakha and V. G. Kulkarni, "A classified bibliography of research on stochastic PERT networks: 1966–1987," *Information Systems and Operational Research*, vol. 27, no. 3, pp. 272–296, 1989.

[3] J. N. Hagstrom, "Computational complexity of PERT problems," *Networks*, vol. 18, no. 2, pp. 139–147, 1988.

[4] C. A. Gallagher, "Note on PERT assumption," *Management Science*, vol. 33, pp. 1360–1362, 1987.

[5] J. J. Martin, "Distribution of the time through a directed, acyclic network," *Operations Research*, vol. 13, pp. 46–66, 1965.

[6] R. M. van Slyke, "Monte Carlo methods and the PERT problem," *Operations Research*, vol. 11, pp. 839–861, 1963.

[7] C. E. Sigal, A. A. B. Pritsker, and J. J. Solberg, "The use of cutsets in Monte Carlo analysis of stochastic networks," *Mathematics and Computers in Simulation*, vol. 21, no. 4, pp. 376–384, 1979.

[8] B. Dodin, "Determining the K most critical paths in PERT networks," *Operations Research*, vol. 32, no. 4, pp. 859–877, 1984.

[9] B. M. Dodin and S. E. Elmaghraby, "Approximating the criticality indices of the activities in PERT networks," *Management Science*, vol. 31, no. 2, pp. 207–223, 1985.

- [10] S. E. Elmaghraby, "On criticality and sensitivity in activity networks," *European Journal of Operational Research*, vol. 127, no. 2, pp. 220–238, 2000.
- [11] S. M. T. Fatemi Ghomi and E. Teimouri, "Path critical index and activity critical index in PERT networks," *European Journal of Operational Research*, vol. 141, no. 1, pp. 147–152, 2002.
- [12] J. Kamburowski, "An upper bound on the expected completion time of PERT networks," *European Journal of Operational Research*, vol. 21, no. 2, pp. 206–212, 1985.
- [13] J. Kamburowski, "Normally distributed activity durations in pert networks," *Journal of the Operational Research Society*, vol. 36, no. 11, pp. 1051–1057, 1985.
- [14] R. P. Loui, "Optimal paths in graphs with stochastic or multidimensional weights," *Communications of the Association for Computing Machinery*, vol. 26, no. 9, pp. 670–676, 1983.
- [15] P. B. Mirchandani and H. Soroush, "Optimal paths in probabilistic networks: a case with temporary preferences," *Computers & Operations Research*, vol. 12, no. 4, pp. 365–381, 1985.
- [16] I. Murthy and S. Sarkar, "A relaxation-based pruning technique for a class of stochastic shortest path problems," *Transportation Science*, vol. 30, no. 3, pp. 220–236, 1996.
- [17] G. A. Corea and V. G. Kulkarni, "Shortest paths in stochastic networks with arc lengths having discrete distributions," *Networks*, vol. 23, no. 3, pp. 175–183, 1993.
- [18] G. H. Polychronopoulos and J. N. Tsitsiklis, "Stochastic shortest path problems with recourse," *Networks*, vol. 27, no. 2, pp. 133–143, 1996.
- [19] M.-J. Yao and W.-M. Chu, "A new approximation algorithm for obtaining the probability distribution function for project completion time," *Computers & Mathematics with Applications*, vol. 54, no. 2, pp. 282–295, 2007.
- [20] B. Dodin, "Approximating the distribution functions in stochastic networks," *Computers and Operations Research*, vol. 12, no. 3, pp. 251–264, 1985.
- [21] M. K. Agrawal and S. E. Elmaghraby, "On computing the distribution function of the sum of independent random variables," *Computers and Operations Research*, vol. 28, pp. 473–483, 2001.
- [22] A. P. Tereso, M. M. T. Araújo, and S. E. Elmaghraby, "Adaptive resource allocation in multimodal activity networks," *International Journal of Production Economics*, vol. 92, no. 1, pp. 1–10, 2004.
- [23] R. Kolisch and A. Sprecher, "PSPLIB—a project scheduling problem library," *European Journal of Operational Research*, vol. 96, no. 1, pp. 205–216, 1996.