Journal of the Operations Research
Society of Japan
Vol. 45, No. 3, September 2002

FIND ALL LONGER AND SHORTER BOUNDARY DURATION VECTORS UNDER PROJECT TIME AND BUDGET CONSTRAINTS

Yi-Kuei Lin
Van Nung Institute of Technology

(Received June 15, 2001; Revised November 29, 2001)

Abstract A large-scale project can be modeled as a project network in AOA form (arrows denote the activities and nodes denote the events). We study the case that activity durations are integer random variables. Given the project time (the deadline required to complete the project) constraint and budget constraint, we use some techniques in network analysis to develop two algorithms in order to generate all longer boundary duration vectors and shorter boundary duration vectors, respectively. Each feasible duration vector is between such longer and shorter boundary vectors. Whenever accidents happening in the project duration, the project manager can update the activity durations according to the longer and shorter boundary duration vectors without contradicting project time and budget constraints.

1. Introduction and Problem Description

In project management, PERT (program evaluation and review technique) and CPM (critical path method) are the most prominent procedure to manage a large-scale project. In general, the activity will cost more if it is required to shorten the activity duration (the time needed to complete the activity). For convenience, the activity duration is called duration throughout this paper. The project is modeled as a project network (a graph with nodes and arrows) to portray the interrelationships among the activities of a project, which can be represented in AOA (activity on arrow) form or AON (activity on node) form. In AOA form, each node denotes an event of the project, and in AON form, arrows denote the relationships between activities. A project network is called a stochastic project network throughout this paper if each duration is a random variable. Traditionally, assuming that each duration is a random variable with beta distribution in advance, three duration estimates (most likely estimate, most optimistic estimate and most pessimistic estimate) are used [4, 5, 7]. The probability that the project is completed within a given time can be approximated.

This paper is mainly to consider the case that each duration is an integer random variable with arbitrary probability distribution. We use AOA form to represent the project network in which the dummy activity is set to be duration 0 with probability 1. Given the project time (the deadline to complete the project) constraint and the budget (total cost) constraint, this paper tries to find all longer boundary duration vectors and shorter boundary duration vectors. However, in the process of executing the project, whenever some accidents happening, the project manager should update the durations without contradicting project time and budget constraints. Hence, the project manager can adjust the durations according to the longer and shorter boundary duration vectors. We use the properties of minimal paths discussed in network analysis and operations research to solve this problem. A minimal path is an ordered sequence of arrows from the source node (start event) to the sink node (end event) that has no cycle. Note that a minimal path is different from the so-called minimum path. The latter is a path with minimum cost. Two algorithms are proposed to enumerate all longer boundary duration vectors and shorter boundary duration vectors under such constraints, respectively. Such vectors represent the duration of each activity. Each feasible duration vector must be between the longer and shorter boundary duration vectors.

1.1. Notation

n	number of activities of the project
a_i	activity (arrow) i, $i = 1, 2, \ldots, n$
X	$(x_1, x_2, \ldots, x_n)$: duration vector where x_i denotes the (current) duration of activity i (the time needed to complete activity i)
C	$(c_1, c_2, \ldots, c_n)$: cost vector where c_i denotes the (current) cost of activity i
u_i; l_i	maximum duration of activity i; minimum duration of activity i. Thus, $l_i \leq x_i \leq u_i$
w_i	maximum cost of activity i
m	number of minimal paths
P_j	minimal path j, $j = 1, 2, \ldots, m$
$\mathbf{M}(P_j)$	maximum duration of P_j, i.e., $\mathrm{M}(P_j) = \sum_{a_i \in P_j} u_i$
$\mathbf{T}(X)$	project time under X (the earliest time to complete the project under X)
$\mathbf{B}(X)$	total cost under X
T	required project time (the deadline to complete the project)
B	budget of the project
e_i	$(d_1, d_2, \ldots, d_n)$ with $d_i = 1$ and $d_j = 0$ for $j \neq i$, $i = 1, 2, \ldots, n$
$X \leq Y$	$(x_1, x_2, \ldots, x_n) \leq (y_1, y_2, \ldots, y_n)$: $x_i \leq y_i$ for each $i = 1, 2, \ldots, n$
$X < Y$	$(x_1, x_2, \ldots, x_n) < (y_1, y_2, \ldots, y_n)$: $X \leq Y$ and $x_i < y_i$ for at least one i

1.2. Assumptions

1. Duration x_i takes possible values: $l_i, l_i + 1, l_i + 2, \ldots, u_i$ and its corresponding cost c_i takes values: $w_i, w_i - 1, w_i - 2, \ldots, w_i - (u_i - l_i)$.
2. Different durations are statistically independent.

2. Stochastic Project Network

The project manager is required to complete the project both within project time T and within the budget B. In order to satisfy the project time constraint, each duration should be shortened possibly. In order to satisfy the budget constraint, each activity cost should be reduced possibly. The project manager is required to schedule the feasible duration vector X such that $\mathrm{T}(X) \leq T$ and $\mathrm{B}(X) \leq B$. For convenience, such an X is called to satisfy (T, B). We propose a concept of longer boundary duration vectors and shorter boundary duration vectors for (T, B). The duration vector X is called a longer boundary duration vector for (T, B) if $\mathrm{T}(X) = T$, $\mathrm{B}(X) \leq B$ and $\mathrm{T}(Y) > T$ for each duration vector Y such that $Y > X$. Similarly, X is called a shorter boundary duration vector for (T, B) if $\mathrm{B}(X) = B$, $\mathrm{T}(X) \leq T$ and $\mathrm{B}(Y) > B$ for each duration vector Y such that $Y < X$. The following Lemmas show that all duration vectors satisfying (T, B) are between the longer and shorter boundary duration vectors.

Lemma 1. If X satisfies (T, B), then there exists a longer boundary duration vector ξ such that $X \leq \xi$.

Proof: It is known that $\mathrm{T}(X) \leq T$. If $\mathrm{T}(X + e_p) > T$ for each arrow a_p with $x_p < u_p$, then ξ is taken as X. Otherwise, there exists an arrow a_k with $x_k < u_k$ such that $\mathrm{T}(X + e_k) \leq T$

and $\mathrm{B}(X+e_k) < B$. Let $X_1 = X + e_k$. If $\mathrm{T}(X_1 + e_p) > T$ for each arrow a_p with $x_p < u_p$, then ξ is taken as X_1. Otherwise, the same procedure can be repeated for X_1 in finite steps, i.e., there exists an integer r such that $X_r > X_{r-1} > \ldots > X_1$ with $\mathrm{T}(X_r) = T$ and $\mathrm{T}(X_r + e_p) > T$ for each arrow a_p with $x_p < u_p$. The proof is concluded by letting $\xi = X_r$.

Lemma 2. If X satisfies (T, B), then there exists a shorter boundary duration vector ψ such that $\psi \leq X$.

Proof: It is known that $\mathrm{B}(X) \leq B$. If $\mathrm{B}(X - e_p) > B$ for each arrow a_p with $x_p > l_p$, then X is taken as ψ. Otherwise, there exists an arrow a_k with $x_k > l_k$ such that $\mathrm{B}(X - e_k) \leq B$ and $\mathrm{T}(X - e_k) < T$. Let $X_1 = X - e_k$. If $\mathrm{B}(X_1 - e_p) > B$ for each arrow a_p with $x_p > l_p$, then X_1 is taken as ψ. Otherwise, the same procedure can be repeated for X_1 in finite steps, i.e., there exists an integer r such that $X_r < X_{r-1} < \ldots < X_1$ with $\mathrm{B}(X_r) = B$ and $\mathrm{B}(X_r - e_p) > B$ for each arrow a_p with $x_p > l_p$. The proof is concluded by letting $\psi = X_r$.

3. Proposed Algorithms

As those approaches proposed in [9–11], we suppose that all minimal paths have been pre-computed. Minimal paths can be efficiently derived from those algorithms discussed in [1, 8, 13]. Two algorithms in terms of minimal paths to generate all longer and shorter boundary duration vectors for (T, B) are proposed in the following, respectively.

3.1. Algorithm I: Generate all longer boundary duration vectors for (T, B)

Step 1. Compute $\mathrm{M}(P_j)$ and let $\lambda_j = \begin{cases} T \text{ if } M(P_j) \geq T \\ \mathrm{M}(P_j) \text{ if } \mathrm{M}(P_j) < T \end{cases}$ for $j = 1, 2, \ldots, m$.

Step 2. Find all integer solutions $X = (x_1, x_2, \ldots, x_n)$ satisfying constraints (3.1) and (3.2)

$$\sum_{a_i \in P_j} x_i = \lambda_j \text{ for } j = 1, 2, \ldots, m, \tag{3.1}$$

$$l_i \leq x_i \leq u_i fori = 1, 2, \ldots, n. \tag{3.2}$$

Step 3. Check each X of step 2 whether its total cost $\mathrm{B}(X)$ exceeds the budget B. If yes, delete X. Otherwise, X is a longer boundary duration vector for (T, B).

Constraint (3.2) means that X is a duration vector, and constraint (3.1) implies that $\mathrm{T}(X) = T$ and $\mathrm{T}(X + e_p) > T$ for any a_p with $x_p < u_p$. Hence, steps 2 and 3 generate all longer boundary duration vectors for (T, B). In order to solve step 2, we can apply the implicit enumeration methods (e.g., branch-and-bound [5, 6] or backtracking [6]) which are always denoted by a search tree composed nodes and arrows. Choose any variable as the starting variable, treat all constraints as bounding functions of the search tree. Repeat this procedure to find all integer solutions of constraints (3.1) and (3.2).

3.2. Algorithm II: Generate all shorter boundary duration vectors for (T, B)

Step 1. Find all integer solutions $C = (c_1, c_2, \ldots, c_n)$ satisfying (3.3) and (3.4).

$$c_1 + c_2 + \ldots + c_n = B \tag{3.3}$$

$$w_i - (u_i - l_i) \leq c_i \leq w_i \text{ for } i = 1, 2, \ldots, n \tag{3.4}$$

Step 2. Transform each C in step 1 to the corresponding $X = (x_1, x_2, \ldots, x_n)$.

Step 3. Check each X whether $\mathrm{T}(X) \leq T$. If yes, X is a shorter boundary duration vector for (T, B).

Steps 1 and 2 generate all duration vectors X such that $\mathrm{B}(X) = B$, and step 3 saves those Xs satisfying $\mathrm{T}(X) \leq T$. Hence, algorithm II generates all shorter boundary duration vectors for (T, B). In step 3, we can easily use the minimal paths to check the condition $\mathrm{T}(X) \leq T$ by calculating $\mathrm{T}(X) = \max_{1 \leq j \leq m} \{ \sum_{a_i \in P_j} x_i \}$. Similar to Algorithm I, the implicit enumeration method can be applied to solve step 1.

4. A Numerical Example

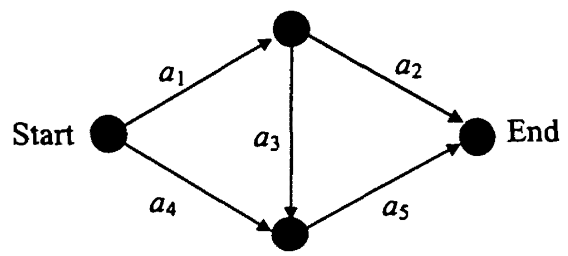

Figure 1: A project network.

A project composed of five activities is represented as a project network with five arrows and four nodes in Figure 1. The project manager is required to complete the project within 7 weeks and within the budget US$ 3400. The data of activity duration and activity cost are listed in table 1. It is known that $T = 7$, $B = 34$, $P_1 = \{a_1, a_2\}$, $P_2 = \{a_1, a_3, a_5\}$, $P_3 = \{a_4, a_5\}$, $(l_1, u_1) = (2, 4)$, $(l_2, u_2) = (3, 5)$, $(l_3, u_3) = (1, 3)$, $(l_4, u_4) = (2, 4)$ and $(l_5, u_5) = (3, 5)$. All longer and shorter boundary duration vectors for (7,34) are obtained by the following procedure.

Table 1: Data of activity duration and activity cost of Figure 1.

Activity	Duration (weeks)	Cost (US$ 100)	Activity	Duration (weeks)	Cost (US$ 100)
a_1	2	8	a_4	2	6
	3	7		3	5
	4	6		4	4
a_2	3	8	a_5	3	8
	4	7		4	7
	5	6		5	6
a_3	1	7			
	2	6			
	3	5			

Algorithm I

Step 1. $\mathrm{M}(P_1) = u_1 + u_2 = 4 + 5 = 9$, $\mathrm{M}(P_2) = u_1 + u_3 + u_5 = 4 + 3 + 5 = 12$ and $\mathrm{M}(P_3) = u_4 + u_5 = 4 + 5 = 9$. Hence, $\lambda_1 = \lambda_2 = \lambda_3 = T = 7$.

Step 2. Find integer solutions $X = (x_1, x_2, x_3, x_4, x_5)$ of constraints (4.1) and (4.2).

$$\begin{cases} x_1 + x_2 = 7 \\ x_1 + x_3 + x_5 = 7 \\ x_4 + x_5 = 7 \end{cases} \tag{4.1}$$

$$2 \leq x_1 \leq 4, 3 \leq x_2 \leq 5, 1 \leq x_3 \leq 3, 2 \leq x_4 \leq 4, 3 \leq x_5 \leq 5 \tag{4.2}$$

The solutions are $\overline{X_1} = (2, 5, 1, 3, 4)$, $\overline{X_2} = (2, 5, 2, 4, 3)$ and $\overline{X_3} = (3, 4, 1, 4, 3)$.

Step 3. The corresponding cost vector of X_1 is $(8, 6, 7, 5, 7)$. Thus $\mathrm{B}(\overline{X_1}) = 8+6+7+5+7 = 33$. Similarly, $\mathrm{B}(\overline{X_2}) = 8+6+6+4+8 = 32$ and $\mathrm{B}(\overline{X_3}) = 7+7+7+4+8 = 33$. Hence $\overline{X_1}, \overline{X_2}$ and $\overline{X_3}$ are all longer boundary duration vectors for $(7, 34)$.

Algorithm II

Step 1. Find integer solutions $C = (c_1, c_2, c_3, c_4, c_5)$ satisfying constraints (4.3) and (4.4).

$$c_1 + c_2 + c_3 + c_4 + c_5 = 34 \tag{4.3}$$

$$6 \leq c_1 \leq 8, 6 \leq c_2 \leq 8, 5 \leq c_3 \leq 7, 4 \leq c_4 \leq 6, 6 \leq c_5 \leq 8 \tag{4.4}$$

18 $(c_1, c_2, c_3, c_4, c_5)$'s are obtained (see Table 2).

Steps 2 and 3. Table 2 shows the final result.

Table 2: Result of algorithm II.

$(c_1, c_2, c_3, c_4, c_5)$ (Step 1)	Corresponding $X = (x_1, x_2, x_3, x_4, x_5)$ (Step 2)	T(X) (Step 3)	$(c_1, c_2, c_3, c_4, c_5)$ (Step 1)	Corresponding $X = (x_1, x_2, x_3, x_4, x_5)$ (Step 2)	T(X) (Step 3)
(8,7,7,4,8)	(2,4,1,4,3) = $\underline{X_1}$	7	(7,8,6,5,8)	(3,3,2,3,3)	8*
(8,6,7,5,8)	(2,5,1,3,3) = $\underline{X_2}$	7	(7,7,6,6,8)	(3,4,2,2,3)	8*
(7,8,7,4,8)	(3,3,1,4,3) = $\underline{X_3}$	7	(6,8,6,6,8)	(4,3,2,2,3)	9*
(7,7,7,5,8)	(3,4,1,3,3) = $\underline{X_4}$	7	(8,8,6,4,8)	(2,3,1,4,4)	8*
(7,6,7,6,8)	(3,5,1,2,3)	8*	(8,7,6,5,8)	(2,4,1,3,4) = $\underline{X_8}$	7
(6,8,7,5,8)	(4,3,1,3,3)	8*	(8,6,6,6,8)	(2,5,1,2,4) = $\underline{X_9}$	7
(8,8,6,4,8)	(2,3,2,4,3) = $\underline{X_5}$	7	(7,8,6,5,8)	(3,3,1,3,4)	8*
(8,7,6,5,8)	(2,4,2,3,3) = $\underline{X_6}$	7	(7,7,6,6,8)	(3,4,1,2,4)	8*
(8,6,6,6,8)	(2,5,2,2,3) = $\underline{X_7}$	7	(6,8,6,6,8)	(4,3,1,2,4)	9*

* T(X) > 7

In sum, we obtain three longer boundary duration vectors for (7,34) and nine shorter boundary duration vectors for (7,34): $\underline{X_1} = (2, 4, 1, 4, 3)$, $\underline{X_2} = (2, 5, 1, 3, 3)$, $\underline{X_3} = (3, 3, 1, 4, 3)$, $\underline{X_4} = (3, 4, 1, 3, 3)$, $\underline{X_5} = (2, 3, 2, 4, 3)$, $\underline{X_6} = (2, 4, 2, 3, 3)$, $\underline{X_7} = (2, 5, 2, 2, 3)$, $\underline{X_8} = (2, 4, 1, 3, 4)$ and $\underline{X_9} = (2, 5, 1, 2, 4)$. Figure 2 shows the relationships between longer and shorter boundary duration vectors for $(7, 34)$. If the duration vector X is scheduled to be $(2, 5, 2, 3, 3)$, then there exists a longer boundary duration vector $\overline{X_2} = (2, 5, 2, 4, 3)$ and a shorter boundary duration vector $\underline{X_6} = (2, 4, 2, 3, 3)$ such that $\overline{X_2} > X > \underline{X_6}$. Hence, $X = (2, 5, 2, 3, 3)$ is a feasible duration vector under the project time constraint and budget constraint.

5. Computational Time Complexity Analysis

5.1. Algorithm I

The number of feasible solutions of constraint (3.2) is $\Phi \equiv \prod_{i=1}^{n} (u_i - l_i + 1)$. Hence, the number of solutions of constraints (3.1) and (3.2) is bounded by Φ. In the worst case, step 1 needs $O(n)$ time for each minimal path and $O(m \cdot n)$ time for all minimal paths. Each solution of constraint (3.2) needs $O(n)$ time to test whether it satisfies $\sum_{a_i \in P_j} x_i = \lambda_j$ for each minimal path and $O(m \cdot n)$ time for all minimal paths in the worst case. Hence, it takes $O(m \cdot n \cdot \Phi)$ time to execute step 2 in the worst case. Each solution in step 2 needs

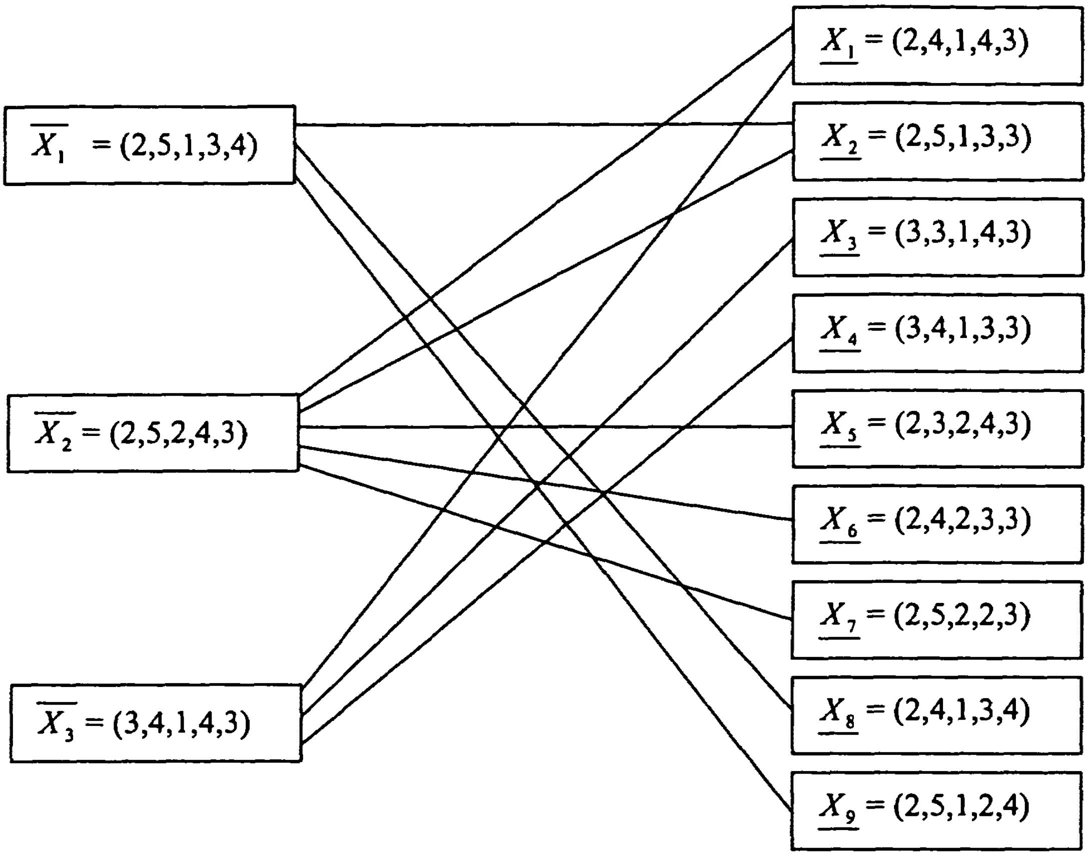

Figure 2: Relationships between longer boundary duration vectors $\overline{X_1}$ to $\overline{X_3}$ for $(7, 34)$ and shorter boundary duration vectors $\underline{X_1}$ to $\underline{X_9}$ for $(7, 34)$

$O(n)$ time to further test the budget constraint. Hence, step 3 needs $O(n \cdot \Phi)$ time in the worst case. Therefore, the computational time complexity of Algorithm I in the worst case is $O(m \cdot n \cdot \Phi) = O(m \cdot n) + O(m \cdot n \cdot \Phi) + O(n \cdot \Phi)$.

5.2. Algorithm II

The number of feasible solutions of constraint (3.4) is $\Phi = \prod_{i=1}^{n} (u_i - l_i + 1)$. Hence, the number of solutions of constraints (3.3) and (3.4) is bounded by Φ. Similarly, the number of Xs transformed in step 2 is bounded by Φ. Each solution of constraint (3.4) needs $O(n)$ time to test constraint (3.3). Hence, it takes $O(n \cdot \Phi)$ time to execute step 1 in the worst case. Each solution in step 1 needs $O(n)$ time to transform to the corresponding X. Hence, step 2 needs $O(n \cdot \Phi)$ time in the worst case. In the worst case, it further needs $O(m \cdot n)$ time to test $\mathrm{T}(X) \leq T$ for each X and $O(m \cdot n \cdot \Phi)$ time for all Xs. Thus, the computational time complexity of Algorithm II in the worst case is $O(m \cdot n \cdot \Phi) = O(n \cdot \Phi) + O(n \cdot \Phi) + O(m \cdot n \cdot \Phi)$.

6. Discussion and Further Research

This article generate all longer and shorter boundary duration vectors under the project time constraint and budget constraint, in which each activity duration x_i takes value from $\{l_i, l_i + 1, l_i + 2, \ldots, u_i\}$, and the corresponding cost c_i takes value from $\{w_i, w_i - 1, w_i - 2, \ldots, w_i - (u_i - l_i)\}$. We use a stochastic project network to represent the project, and use the properties of minimal paths to develop two algorithms in order to generate all longer and shorter boundary duration vectors, respectively. If the probability distribution of each duration is given, then the probability that the project is completed within a project time and within the budget can be computed by using the state-space decomposition discussed in [2, 3, 9–12] in terms of longer and shorter boundary duration vectors. Such a probability is a performance index of project management. In particular, the probability distribution of activity duration is not limited to any existing distribution but arbitrary. Furthermore, we can apply the sensitivity analysis for performance index to evaluate the most important activity.

Future research can develop algorithms to generate all longer and shorter boundary duration vectors for the general case that the activity duration x_i takes integer values $l_{i1} < l_{i2} < \ldots < l_{ir_i}$ (l_{ij} is an integer number for $j = 1, 2, \ldots, r_i$) and its corresponding cost c_i takes integer values $w_{i1} > w_{i2} > \ldots > w_{ir_i}$, where r_i is the number of possible values of x_i. Furthermore, calculate or approximate the probability that the project is completed under project time constraint and budget constraint. Another interesting topic is to study the optimal duration vector, keeping both $\mathrm{B}(X)$ and $\mathrm{T}(X)$ as small as possible.

References

[1] A.G. AM: A heuristic technique for generating minimal path and cutsets of a general network. *Computers and Industrial Engineering*, **36** (1999) 45–55.

[2] C. Alexopoulos: A note on state-space decomposition methods for analyzing stochastic flow networks. *IEEE Transactions on Reliability*, **44** (1995) 354–357.

[3] P. Doulliez and J. Jamoulle: Transportation networks with random arc capacities. *RAIRO, Recherche Operationnelle Operations Research*, **3** (1972) 45–60.

[4] J. Gido and Clements, J.P: *Successful project management* (South-western college publishing, Ohio, 1999).

[5] F. S. Hiller and G. J. Liberman: *Introduction to operations research* (McGraw-Hill, NY, 2001).

[6] E. Horowitz and S. Sahni: *Fundamentals of computer algorithms* (Computer Science Press, Potomac, MD, 1978).

[7] H. Kerzner: *Project management* (Van Nostrand America, NY, 1992).

[8] K. Kobayashi and H. Yamamoto: A New Algorithm in Enumerating all Minimal Paths in a Sparse Network. *Reliability Engineering and System Safety*, **65** (1999) 11–15.

[9] J. S. Lin: Reliability evaluation of capacitated-flow networks with budget constraints. *IIE Transactions*, **30** (1998) 1175–1180.

[10] Y. K. Lin and J. Yuan: A new algorithm to generate d-minimal paths in a multi-state flow network with non-integer arc capacities. *International Journal of Reliability, Quality and Safety Engineering*, **5** (1998) 269–285.

[11] Y. K. Lin: A Simple Algorithm for Reliability Evaluation of a Stochastic-Flow Network with Node Failure. *Computers and Operations Research*, **28**-13 (2001) 1277–1285.

[12] Y. K. Lin: On Reliability Evaluation of a Stochastic-Flow Network in Terms of Minimal Cuts. *Journal of Chinese Institute of Industrial Engineers*, **18**-3 (2001) 49–54.

[13] Y. Shen: A New Simple Algorithm fir Enumerating all Minimal Paths and Cuts of a Graph. *Microelectronics and Reliability*, **35**-6 (1995) 973–976.

Acknowledgement

This work was supported in part by the National Science Council, Taiwan, Republic of China, under Grant No. Nsc89-2213-E-238-008.

Yi-Kuei Lin
Department of Information Management
Chung-Li city, Tao-Yuan, Taiwan 320, R.O.C.
E-mail: yklin@cc.vit.edu.tw