Journal of the Operations Research
Society of Japan
2007, Vol. 50, No. 2, 73-81

STUDY ON LONGER AND SHORTER BOUNDARY DURATION VECTORS WITH ARBITRARY DURATION AND COST VALUES

Yi-Kuei Lin
National Taiwan University of Science and Technology

(Received April 15, 2005; Revised December 26, 2006)

Abstract We use the project network in AOA form (arrows denote the activities and nodes denote the status of the project) to represent a large-scale project. The activity duration and the activity cost are both random variables which take arbitrary integer values with the arbitrary probability distribution. Under the project time (the deadline to complete the project) constraint and the budget constraint, this paper studies how to schedule all activity durations of the project. Two algorithms are proposed to generate all upper and lower boundary vectors for the project, respectively. All feasible activity durations of the project are among such upper and lower boundary vectors.

Keywords: Project planning, activity duration, activity cost, upper and lower boundary vectors, minimal path

1. Introduction

PERT (program evaluation and review technique) and CPM (critical path method) are the most prominent procedure to manage a large-scale project [4]. The project can be 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 a status of the project, and in AON form, arrows denote the relationships between activities. Traditionally, assuming the probability distribution of the activity duration (the time needed to complete the activity) is a beta distribution in advance, three estimates (most likely estimate, most optimistic estimate and most pessimistic estimate) of activity duration are used [4, 5].

In general, the activity cost increases as the shortening of the activity duration. And the activity duration increases if the activity cost decreases. Lin [10] proposed algorithms to schedule all activity durations for the case that each activity duration takes possible values: a, $a+1$, $a+2$, $\ldots$, b and its corresponding cost takes values: c, $c-1$, $c-2$, $\ldots$, $c-(b-a)$. In other words, the activity duration as well as the activity cost takes consecutive integers. This paper extend the consecutive integers case to the general case that the activity duration and the activity cost are both integer random variables which take arbitrary values with the arbitrary probability distribution. Given the project time (the time required to complete the project) and budget (total activity cost) constraints of the project, we try to schedule all activity durations. Two algorithms are proposed to enumerate all upper boundary vectors and lower boundary vectors for such constraints, respectively. Component i of such a vector represents the duration of activity i. All feasible activity durations can be obtained from the upper and lower boundary vectors.

1.1. Nomenclature and notation

Minimal path:	a minimal path is a path from the source node (start status) to the sink node (end status) in which no proper subset is a path.
n	number of activities of the project
a_i	activity (arrow) i, $i = 1, 2, \ldots, n$
x_i	the (current) duration of activity i, i.e., the time needed to complete activity i. x_i is a non-negative integer.
u_i	number of possible values of x_i
x_{ij}	jth possible value of $x_i, j = 1, 2, \ldots, u_i$
X	$(x_1, x_2, \ldots, x_n)$: duration vector
Z	$(z_1, z_2, \ldots, z_n)$: pseudo duration vector, $z_i \geq 0$ is an integer for $i = 1, 2, \ldots, n$
c_i	the (current) cost of activity i
c_{ij}	jth possible value of c_i, $j = 1, 2, \ldots, u_i$
C	$(c_1, c_2, \ldots, c_n)$: cost vector
D	$(d_1, d_2, \ldots, d_n)$: pseudo cost vector, $d_i \geq 0$ is an integer for $i = 1, 2, \ldots, n$
m	number of minimal paths
P_j	jth minimal path, $j = 1, 2, \ldots, m$
$\mathrm{T}(X)$	project time under X (the least time needed to complete the project, i.e., $\mathrm{T}(X) = \max\limits_{1 \leq j \leq m} \sum\limits_{a_i \in P_j} x_i$)
$\mathrm{B}(X)$	total cost of the project under X
T	required project time (the deadline to complete the project)
B	budget of the project
e_i	$(x_1, x_2, \ldots, x_n)$ with $x_i = 1$ and 0 others
$X \leq Y$	$(x_1, x_2, \ldots, x_n) \leq (y_1, y_2, \ldots, y_n) : x_i \leq y_i$ for $i = 1, 2, \ldots, n$
$X < Y$	$(x_1, x_2, \ldots, x_n) < (y_1, y_2, \ldots, y_n) : X \leq Y$ and $x_i < y_i$ for at least one i

1.2. Assumptions

1. Activity duration x_i takes possible integer values: $x_{i1} < x_{i2} < \ldots < x_{iu_i}$ with arbitrary probability distribution, and its corresponding cost c_i takes values: $c_{i1} > c_{i2} > \ldots > c_{iu_i}$.
2. Different activity durations are statistically independent.

2. Model Building

In this paper, we use AOA form to represent the project network in which the dummy activity is set to be duration zero with probability 1. The project manager is required to complete the project within deadline T and within the budget B. In order to satisfy the project time constraint, the activity duration should be shortened, and that will cause the increase of the activity cost. In order to satisfy the budget constraint, the activity cost should be reduced, and that will increase the activity duration. The purpose of this paper is to find all feasible duration vectors X without contradicting the deadline constraint and the budget constraint. That is, we generate all duration vectors 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). However, the number of such X will increase very quickly if the project is large. It is thus not a wise way to enumerate all such X. Hence, we propose an idea of upper boundary vectors and lower boundary vectors for (T, B). A duration vector X is defined to be an upper boundary vector for (T, B) if X satisfies (T, B) and $\mathrm{T}(Y) > T$ for each duration vector Y such that $Y > X$. Similarly, X is defined to be a lower boundary vector for (T, B) if X satisfies (T, B) and $\mathrm{B}(Y) > B$ for each duration vector Y such that $Y < X$. In order to generate all upper and lower boundary vectors, we introduce pseudo duration vector $Z = (z_1, z_2, \ldots, z_n)$ and pseudo cost vector $D = (d_1, d_2, \ldots, d_n)$, respectively. The pseudo duration vector is defined to be the n-tuple vector which satisfies the time T exactly. In particular, the component z_i in a pseudo duration vector is an integer and is not necessary to be any value from $\{x_{i1}, x_{i2}, \ldots, x_{iu_i}\}$. Similarly, the component d_i in a pseudo cost vector is an integer and is not necessary to be any value from $\{c_{i1}, c_{i2}, \ldots, c_{iu_i}\}$. The following lemma plays the crucial role in generating all upper boundary vectors for (T, B).

Lemma 1 *For each upper boundary vector X for (T, B), there exists at least one pseudo duration vector Z with $Z \geq X$ such that $\mathrm{T}(Z) = T$ and $\mathrm{T}(Z') > T$ for each pseudo duration vector $Z' > Z$.*

Proof: Note that each activity on the critical path has slack time 0.

(i). For the case that $\mathrm{T}(X) = T$, if all activities have slack time 0, then Z is taken as X. Otherwise, choose an arrow a_p which is not on any critical path, and set $Z_1 = X+$ (slack time of $a_p) \cdot e_p$. Then $\mathrm{T}(Z_1) = T$ and a_p is on a critical path. Calculate the slack time of each activity under Z_1. Repeat the same procedure for other arrows that are not on the critical paths. However, this procedure will stop in finite steps, i.e., there exists an integer k such that $Z_k > Z_{k-1} > \ldots > Z_1$ with $\mathrm{T}(Z_k) = T$ and each minimal path under Z_k is a critical path. Hence, $Z = (z_1, z_2, \ldots, z_n)$ may be taken as Z_k and thus $\sum_{a_i \in P_j} z_i = T$ for each P_j.

(ii). If $\mathrm{T}(X) < T$, without loss of generality, we assume $\mathrm{T}(X) = T - 1$. Then there exist a minimal path P_j such that $\sum_{a_i \in P_j} x_i = T - 1$. Choose an arrow $a_q \in P_j$, set $Z^* = X + e_q$. Thus, $\mathrm{T}(Z^*) = T$. Treat Z^* as the role X in proof (i) and repeat the proof (i). The proof is completed. □

Hence, we can first generate all pseudo duration vectors Z such that $\mathrm{T}(Z) = T$ and $\mathrm{T}(Z') > T$ for each pseudo duration vector $Z' > Z$. For each Z, the proof of Lemma 1 indicates that $\sum_{a_i \in P_j} z_i = T$ for each P_j. Then for each Z, find the largest duration vector X such that $X \leq Z$. Each X is thus a candidate of upper boundary vector for (T, B). Assuming all minimal paths are derived in advance, the algorithm to generate all upper boundary vectors for (T, B) is proposed as follows.

Algorithm I: Generate all upper boundary vectors for (T, B)

Step 1. Find all pseudo duration vectors $Z = (z_1, z_2, \ldots, z_n)$ satisfying constraints (1) and (2)

$$\sum_{a_i \in P_j} z_i = T \quad \text{for } j = 1, 2, \ldots, m \tag{1}$$

$$z_i \geq x_{i1} \quad \text{for } i = 1, 2, \ldots, n \tag{2}$$

Step 2. For each Z, find the largest duration vector $X = (x_1, x_2, \ldots, x_n)$ such that $X \leq Z$ as follows: for each z_i, find x_{ip} such that $x_{ip} \leq z_i < x_{i(p+1)}$ and set $x_i = x_{ip}$.

Step 3. Check each X whether its total cost $\mathrm{B}(X)$ exceeds the budget B. If yes, delete X. Suppose the remainder is $\{X_1, X_2, \ldots, X_w\}$.

Step 4. Remove those non-maximal ones from $\{X_1, X_2, \ldots, X_w\}$ to obtain the set of upper boundary vectors for (T, B)s.

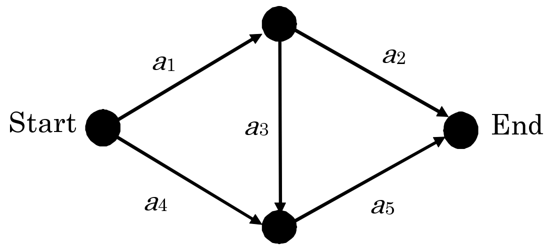

Figure 1: A simple project network

Similarly, Lemma 2 plays the crucial role in generating all lower boundary vectors for (T, B).

Lemma 2 *If X is a lower boundary vector for (T, B) and its corresponding cost vector is $C = (c_1, c_2, \ldots, c_n)$, then there exists at least one pseudo cost vector $D = (d_1, d_2, \ldots, d_n)$ with $D \geq C$ such that $d_1 + d_2 + \ldots + d_n = B$.*

Proof: (i). If $c_1 + c_2 + \ldots + c_n = B$, then D is taken as C. (ii). Without loss of generality, we assume $c_1 + c_2 + \ldots + c_n = B - 1$. Choose an a_p with $c_p < c_{p1}$, set $D = C + e_p$. Then $d_1 + d_2 + \ldots + d_n = B$. □

The algorithm to generate all lower boundary vectors for (T, B) is proposed as follows.

Algorithm II: Generate all lower boundary vectors for $(\boldsymbol{T}, \boldsymbol{B})$

Step 1. Find all pseudo cost vectors $D = (d_1, d_2, \ldots, d_n)$ satisfying constraints (3) and (4).

$$d_1 + d_2 + \ldots + d_n = B \tag{3}$$

$$d_i \in \{c_{i1}, c_{i1} - 1, c_{i1} - 2, \ldots, c_{iu_i}\} \text{ for } i = 1, 2, ..., n \tag{4}$$

Step 2. For each $D = (d_1, d_2, \ldots, d_n)$, find the largest cost vector $C = (c_1, c_2, \ldots, c_n)$ such that $C \leq D$ as follows: for each d_i, find c_{ip} such that $c_{i(p-1)} > d_i \geq c_{ip}$ and set $c_i = c_{ip}$.

Step 3. Transform each C to the corresponding $X = (x_1, x_2, \ldots, x_n)$ and check whether $\mathrm{T}(X) > T$. If yes, delete X. Suppose the remainder is $\{X_1, X_2, \ldots, X_v\}$.

Step 4. Remove those non-minimal ones from $\{X_1, X_2, \ldots, X_v\}$. The remainder is the set of lower boundary vectors for (T, B).

3. A Numerical Example

We use a simple project network to illustrate the proposed algorithms. 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 10 weeks and within the budget US$ 2800. Table 1 shows the data of activity duration and activity cost. It is known that there are three minimal paths; $P_1 = \{a_1, a_2\}, P_2 = \{a_1, a_3, a_5\}$ and $P_3 = \{a_4, a_5\}$. All upper and lower boundary vectors for $(10, 28)$ are obtained by the following procedure.

Algorithm I

Step 1. Find all feasible solutions $Z = (z_1, z_2, z_3, z_4, z_5)$ of constraints (5) and (6).

$$\begin{cases} z_1 + z_2 = 10 \\ z_1 + z_3 + z_5 = 10 \\ z_4 + z_5 = 10 \end{cases} \tag{5}$$

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

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

Table 2: The results of Algorithm I for the example

Pseudo duration vector Z (Step 1)	Duration vector X (Step 2)	B(X) (Step 3)	Is an upper boundary vector for (10,28)? (Step 4)
(2,8,1,3,7)	(2,5,1,2,7)	30*	NO (B(X) $>$ 28)
(2,8,2,4,6)	(2,5,1,4,5)	30*	NO (B(X) $>$ 28)
(2,8,3,5,5)	(2,5,3,4,5)	28 (X_1)	YES ($\overline{X_1}$)
(2,8,4,6,4)	(2,5,4,6,3)	26 (X_2)	NO ($X_2 \leq X_5$)
(2,8,5,7,3)	(2,5,4,6,3)	26 (X_3)	NO ($X_3 \leq X_5$)
(3,7,1,4,6)	(2,5,1,4,5)	30*	NO (B(X) $>$ 28)
(3,7,2,5,5)	(2,5,1,4,5)	30*	NO (B(X) $>$ 28)
(3,7,3,6,4)	(2,5,3,6,3)	28 (X_4)	NO ($X_4 \leq X_5$)
(3,7,4,7,3)	(2,5,4,6,3)	26 (X_5)	YES ($\overline{X_2}$)
(4,6,1,5,5)	(4,5,1,4,5)	28 (X_6)	YES ($\overline{X_3}$)
(4,6,2,6,4)	(4,5,1,6,3)	28 (X_7)	NO ($X_7 \leq X_8$)
(4,6,3,7,3)	(4,5,3,6,3)	26 (X_8)	YES ($\overline{X_4}$)
(5,5,1,6,4)	(4,5,1,6,3)	28 (X_9)	NO ($X_9 \leq X_8$)
(5,5,2,7,3)	(4,5,1,6,3)	28 (X_{10})	NO ($X_{10} \leq X_8$)
(6,4,1,7,3)	(6,4,1,6,3)	28 (X_{11})	YES ($\overline{X_5}$)

*B(X) $>$ 28

$$z_1 \geq 2, z_2 \geq 3, z_3 \geq 1, z_4 \geq 2, z_5 \geq 3 \tag{6}$$

Step 2 to Step 4: The results are listed in Table 2. We obtain 5 upper boundary vectors for (10,28): $\overline{X_1} = (2,5,3,4,5), \overline{X_2} = (2,5,4,6,3), \overline{X_3} = (4,5,1,4,5)$, $\overline{X_4} = (4,5,3,6,3)$ and $\overline{X_5} = (6,4,1,6,3)$.

Algorithm II

Step 1. Find all $D = (d_1, d_2, d_3, d_4, d_5)$ satisfying (7) and (8).

$$d_1 + d_2 + d_3 + d_4 + d_5 = 28 \tag{7}$$

$$4 \leq d_1 \leq 8, 3 \leq d_2 \leq 7, 4 \leq d_3 \leq 8, 3 \leq d_4 \leq 7, 4 \leq d_5 \leq 8 \tag{8}$$

Step 2 to Step 4: We obtain 9 lower boundary vectors for (T, B): $\underline{X_1} = (6,4,1,6,3)$, $\underline{X_2} = (4,5,3,4,3)$, $\underline{X_3} = (4,5,1,6,3)$, $\underline{X_4} = (4,5,1,4,5)$, $\underline{X_5} = (4,4,3,6,3)$, $\underline{X_6} = (2,5,4,4,3)$, $\underline{X_7} = (2,5,3,6,3)$, $\underline{X_8} = (2,5,3,4,5)$ and $\underline{X_9}$= (2,4,4,6,3).

Figure 2 shows the relationships among all upper and lower boundary vectors for $(10, 28)$. If the duration vector X is scheduled to be $(3,5,3,6,3)$, then there exists an upper boundary vector $\overline{X_4} = (4,5,3,6,3)$ and a lower boundary vector $\underline{X_7} = (2,5,3,6,3)$ such that $\overline{X_4} > X > \underline{X_7}$.

4. Computational Complexity Analysis

4.1. Algorithm I

According to constraints (1) and (2), the value z_i will be dominated by $T \geq z_i \geq x_{i1}$ for $i = 1, 2, \ldots, n$. The number of feasible solutions of constraint (1) and (2) is bounded by $\Psi \equiv \prod_{i=1}^{n} (T - x_{i1} + 1)$. In the worst case, it takes $O(n)$ time to transform each Z into the largest duration vector and $O(n\Psi)$ time to transform all Z. Each solution of step 2 needs $O(n)$ time to be tested for the satisfaction of the cost constraint. Thus, it takes at most $O(n\Psi)$ time to execute step 3. In step 4, it takes at most $O(n\Psi)$ time to compare each candidate with others, and thus $O(n\Psi^2)$ time for all candidates. Therefore, the computational time complexity of Algorithm I in the worst case is $O(n\Psi^2) = O(n\Psi) + O(n\Psi) + O(n\Psi^2)$.

4.2. Algorithm II

The number of feasible solutions of constraint (4) is $\Phi = \prod_{i=1}^{n} (c_{i1} - c_{iu_i} + 1)$. Hence, the number of solutions of constraints (3) and (4) is bounded by Φ. Similarly, the number of Xs transformed in step 3 is bounded by Φ. It takes $O(n)$ time to transform each solution in step 1 into the largest cost vector. Hence, step 2 needs $O(n\Phi)$ time in the worst case. It takes $O(n)$ time to transform each solution in step 2 into the corresponding X, and $O(mn)$ time to test $\mathrm{T}(X) \leq T$. Thus, it takes at most $O(mn\Phi)$ to execute step 3. In step 4, it takes at most $O(n\Phi)$ time to compare each candidate with others, and thus $O(n\Phi^2)$ time for all candidates. Therefore, the computational time complexity of Algorithm II in the worst case is $O(n\Phi(m + \Phi)) = O(n\Phi) + O(mn\Phi) + O(n\Phi^2)$.

5. Discussion and Conclusion

This article discusses the problem that the project should be completed within the project time and budget, in which activity duration is a random variable which takes arbitrary integer values with an arbitrary probability distribution. We propose an idea of upper and

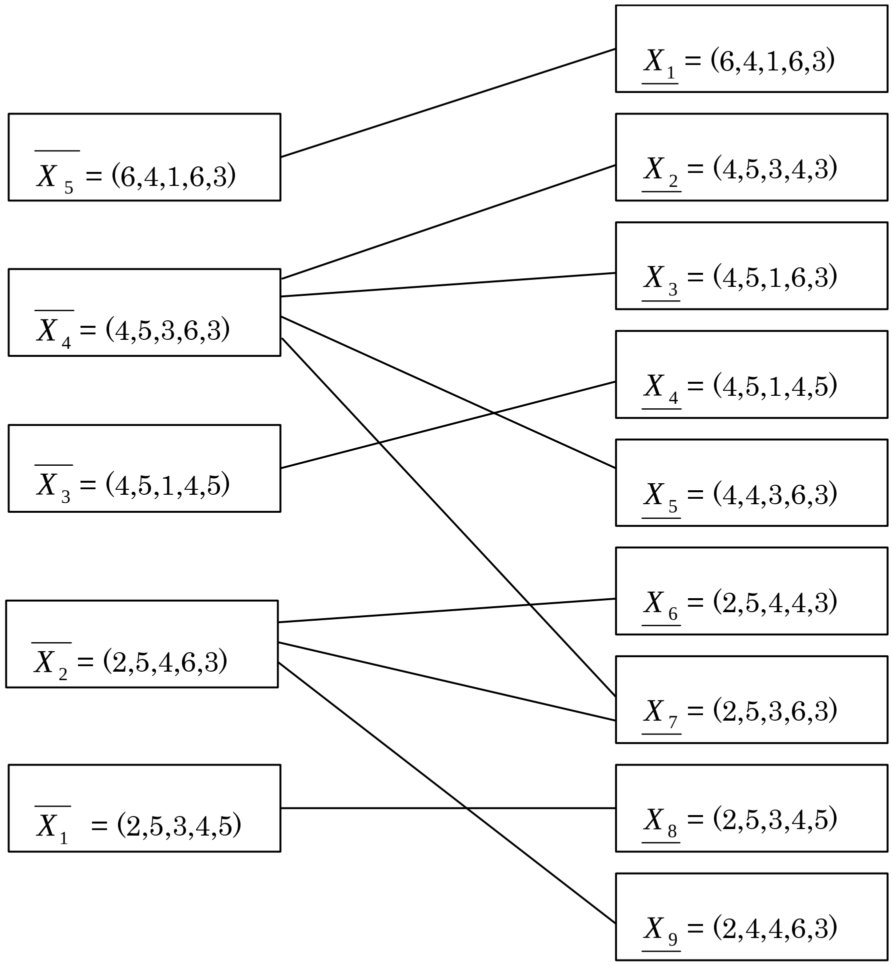

Figure 2: Relationships among upper boundary vectors $\overline{X_1}$ to $\overline{X_5}$ for $(10, 28)$ and lower boundary vectors $\underline{X_1}$ to $\underline{X_9}$ for $(10, 28)$

lower boundary vectors. All feasible activity durations are among such upper and lower boundary vectors. Two algorithms in terms of minimal paths are proposed to generate all upper and lower boundary vectors, respectively. Minimal paths can be efficiently derived from those algorithms discussed in [2, 6, 12, 13]. We use the pseudo duration vector and pseudo cost vector to generate all upper and lower boundary vectors for (T, B) in algorithms I and II, respectively.

In order to solve constraints (1) and (2) in Algorithm I, we can use the implicit enumeration method (e.g., branch-and-bound or backtracking) which is always denoted by a search tree composed of nodes and arcs. Choose any variable as the starting variable and treat all equations in (1) and inequalities in (2) as constraints of the search tree. Repeat this procedure to find all integer solutions of constraints (1) and (2). That is a similar way to solve constraints (3) and (4) in Algorithm II.

It seems that constraint (2) can be modified to be

$$x_{i1} \leq z_i \leq x_{iu_i} \quad \text{for } i = 1, 2, \ldots, n. \tag{9}$$

In fact, it can not be guaranteed that the pseudo duration vectors Z stated in Lemma 1 satisfy $z_i \leq x_{iu_i}$ for $i = 1, 2, \ldots, n$. And thus we will lose some upper boundary vectors X for (T, B) if the constraint (2) is replaced by (9). In the numerical example, each solution Z contradicts to constraint (9). That is why the constraint (2) limits x_i to only $x_i \geq x_{i1}$.

If the probability distribution of activity duration is given, then the probability to complete the project within a project time and the budget can be easily computed by using state-space decomposition methods discussed in [1, 3, 7-9, 11, 14] in terms of upper and lower boundary vectors. Such a probability can be treated as a performance index in project management. Future research can study the case that the activity duration is a discrete random variable and takes real-number values.

Acknowledgements

This work was supported in part by the National Science Council, Taiwan, Republic of China, under Grant No. NSC 94-2213-E-238-001.

References

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

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

[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 J.P. Clements: *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] 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.

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

[8] 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.

[9] Y.K. Lin: Reliability evaluation for a multi-commodity flow model under budget constraint. *Journal of the Chinese Institute of Engineers*, **25** (2002), 109–116.

[10] Y.K. Lin: Find all longer and shorter boundary duration vectors under project time and budget constraints. *Journal of the Operations Research Society of Japan*, **45** (2002), 260–267.

[11] Y.K. Lin: Reliability of a stochastic-flow network with unreliable branches & nodes under budget constraints. *IEEE Transactions on Reliability*, **53-5** (2004), 381–387.

[12] Y. Shen: A new simple algorithm fir enumerating all minimal paths and cuts of a graph. *Microelectronics and Reliability*, **35-6** (1995), 973–976.

[13] W.C. Yeh: A simple algorithm to search for all d-MPs with unreliable nodes. *Reliability Engineering and System Safety*, **73** (2001), 49–54.

[14] W.C. Yeh: A simple approach to search for all d-MCs of a limited-flow network. *Reliability Engineering and System Safety*, **71** (2001), 15–19.

Yi-Kuei Lin
Department of Industrial Management
National Taiwan University of Science and Technology
Taipei, Taiwan 106, R.O.C.
E-mail: `yklin@im.ntust.edu.tw`