

A Monte Carlo Method for Estimating the Extended All-terminal Reliability

Shiang-Ming Huang¹, Quincy Wu² and Shi-Chun Tsai¹

¹*Department of Computer Science, National Chiao Tung University*
 {smhuang,sctsai}@csie.nctu.edu.tw

²*Graduate Institute of Communication Engineering, National Chi Nan University*
 solomon@ipv6.club.tw

Abstract

Designing a network with optimal deployment cost and maximum reliability considerations is a hard problem, especially when the all-terminal reliability is required. For efficiently finding out an acceptable solution, Genetic Algorithms (GAs) have been widely applied to solve this problem. In these GAs, the reliability values could be calculated in their objective functions. In year 2002, an extended network reliability model was proposed which considers the connection important level between each pair of nodes. This paper proposes an approximation algorithm based on Monte Carlo simulation for the new network reliability model. This approximation algorithm can be integrated into GAs to solve the optimal cost reliable network design problem under the extended model.

1. Introduction

Reliability and cost are two important considerations for designing communications networks, such as backbone telecommunications networks, wide area networks and data communications networks located in industrial facilities [1]. Based on the reliability and cost considerations, the design goal of a communications network is either to increase network reliability or to decrease deployment cost. However, increasing reliability needs to add redundancy to the network while optimizing cost needs to remove any extra redundancy from the network. It is not easy to find out a saddle point that compromises both maximum reliability and optimal cost considerations at the same time.

To find out an acceptable solution that meets both the reliability and cost considerations, the above network design problem can be transformed into a simpler problem that is to either *maximize the reliability given a maximum cost constraint*, or to

minimize the cost given a minimum reliability constraint. Both these two problems are NP-hard [2], which means these problems are unlikely to be solved within polynomial time, and all the existing algorithms require huge computational effort that will increase exponentially as the network size increases.

Several genetic algorithms (GAs) [2,3,4,5] have been proposed for minimizing the topology deployment cost under a given reliability constraint. In these GAs, a communications network is modeled as an undirected probability graph $g = G(N, L, p)$ where N is a set of nodes (e.g., computer sites), L is a set of links (e.g., communication connections), and p is the connection (link) failure probability ($0 \leq p \leq 1$). The GAs encode each of the possible network topologies into a bit-string as a 'gene', and then perform GA operations on these genes for cost optimization.

A typical execution flow of the GAs is illustrated in Fig. 1 where the population pool (Fig. 1 ①) stores many encoded genes. From the population pool, several genes are selected as parents (Fig. 1 ②) for generating the offspring (Fig. 1 ③). The offspring are also genes that are recombined (Fig. 1 ④) and mutated (Fig. 1 ⑤) from the parent genes. These offspring return to the population pool after a survival selection (Fig. 1 ⑥). The above process (Fig. 1 ①, ②, ③ and ④) repeats for a predetermined number of rounds, or until the optimization requirement is reached, depending on which one happens earlier. Both the cost and the reliability could be calculated in the GA objective function that is a scoring system that calculates evolutionary fitness of a gene for the parent selection and survival selection functions (Fig. 1 ① and ⑥).

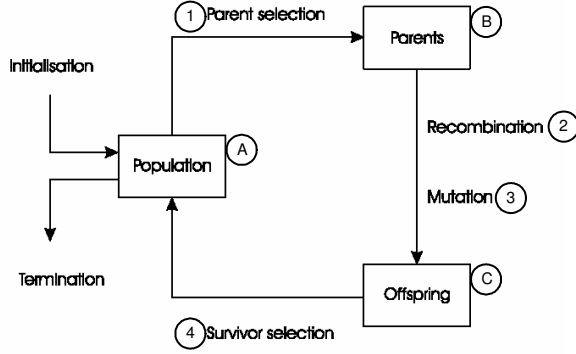


Figure 1: A typical execution flow of GAs.

The network design problem is especially difficult when considering *all-terminal* network reliability that is defined as the probability that every pair of nodes can communicate with each other. The all-terminal reliability is difficult because it is also NP-hard [6]. To efficiently estimate the all-terminal network reliability, cut set count [7] and Monte Carlo simulation [8] are widely used in the GAs. These approximation methods ensure the all-terminal reliability to be calculated in polynomial time. Otherwise, the fitness calculation for each gene would be an NP-hard problem and thus the overall calculation of the GAs would not be practical.

In year 2002, Feng et al. proposed an extended network reliability model [9], which extended the original model [1] with a weighting on each pair of nodes. It is more meaningful than the basic model under link failure. Unfortunately, according to our best understanding, no approximation algorithm has been proposed for it. In this paper, we propose an approximation algorithm based on Monte Carlo simulation for the extended network reliability model.

This paper is organized as follows. Section 2 describes the basic network reliability model (the calculation is NP-hard), and introduces two approximation methods for reliability estimation. Also, the extended network reliability model (the calculation is harder than the basic model) is described in this section. In Section 3, we propose an approximation algorithm for the extended reliability model and then prove its correctness. Section 4 demonstrates the proposed algorithm by a numerical example. Finally, Section 5 concludes this paper.

2. Previous works

A network is a set of facilities, i.e., nodes and links, where a failure probability p_i is associated with each facility i . In this network model, the nodes are assumed to be perfectly reliable and the links have independent

failure probabilities. If a network has only three nodes a, b, c and two links (a, b) and (b, c) , then the probability that both the links (a, b) and (b, c) are working is $(1 - p_{(a, b)})(1 - p_{(b, c)})$, which is also defined as the reliability of that network.

Given a minimum network reliability constraint Rel_0 , the cost optimization problem for the GAs is:

$$\text{Minimizing the total cost } C(g) = \sum_{i=1}^{|N|-1} \sum_{j=i+1}^{|N|} c_{ij} x_{ij}$$

Subject to: $Rel(g) \geq Rel_0$. (1)

Where c_{ij} is the cost of the (i, j) link, $x_{ij} \in \{0, 1\}$ indicates whether the link (i, j) is working or not, and $Rel(g)$ is the reliability of the network g ($0 \leq Rel(g) \leq 1$, $0 \leq Rel_0 \leq 1$).

Based on the basic network reliability measure, the all-terminal reliability $Rel_{all}(g)$ of a network $g = G(N, L, p)$ is:

$$Rel_{all}(g) = \sum_{g_s \in \Omega} Rel(g_s) \quad (2)$$

Where $g_s = G(N, L_s, p)$ is a sub-network of g and $L_s \subseteq L$. Ω is the set of g 's sub-networks that every pair of nodes can communicate with each other (e.g., Fig. 2 (B) and (C)). $Rel(g_s)$ is the reliability of the sub-network g_s . Each g_s corresponds to a link failure scenario of g , and the reliability of g_s is:

$$Rel(g_s) = \prod_{i \in \text{working}} (1 - p_i) \prod_{i \notin \text{working}} p_i \quad (3)$$

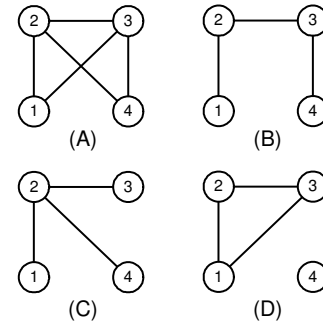


Figure 2: Network configurations. (A): g ; (B) and (C): $g_s \in \Omega$; (D) $g_s \notin \Omega$.

The $|L|$ -links-network g has $2^{|L|}$ failure scenarios. It means to obtain $Rel_{all}(g)$, there are at most $2^{|L|}$ failure scenarios to be examined. The time for calculating $Rel_{all}(g)$ grows exponentially as the number of links in g increases. To approximate $Rel_{all}(g)$ efficiently, *cut set count* and *Monte Carlo simulation* are two widely used methods in the GAs.

2.1 Cut set count

The cut set count method applies *Set Theory* to estimate the all-terminal reliability. The basic definitions in this method are: 1) F_i is the event that all links connected to node i failed, and 2) \bar{F}_i is F_i 's complement. Apparently, F_i is a cut set of the network $g = G(N, L, p)$, and the network g fails (which means some pair of nodes in this network are unable to communicate with each other) under the scenarios containing the event F_i . Since $F_1, F_2, F_3 \dots F_{|M|}$ are collectively a subset of all the possible network failure scenarios that fail the network g , the network failure probability, i.e., $1 - Rel_{all}(g)$, could be written as:

$$\begin{aligned} 1 - Rel_{all}(g) &\geq \Pr[F_1 \cup F_2 \cup \dots \cup F_{|M|}] \\ &= \Pr[F_1] + \Pr[F_2 \cap \bar{F}_1] + \Pr[F_3 \cap \bar{F}_2 \cap \bar{F}_1] \\ &\quad + \dots + \Pr[F_{|M|} \cap \bar{F}_{1 \dots |M|-1} \cap \dots \cap \bar{F}_1] \\ &= \Pr[F_1] + \Pr[F_2] \Pr[\bar{F}_1 | F_2] \\ &\quad + \Pr[F_3] \Pr[\bar{F}_2 \cap \bar{F}_1 | F_3] \\ &\quad + \dots + \Pr[F_{|M|}] \Pr[\bar{F}_{1 \dots |M|-1} \cap \dots \cap \bar{F}_1 | F_{|M|}] \quad (4) \end{aligned}$$

From derivation in [7], inequality (4) could be rewritten as:

$$\begin{aligned} Rel_{all}(g) &\leq 1 - \Pr[F_1] - \Pr[F_2] \Pr[\bar{F}_1 | F_2] \\ &\quad - \Pr[F_3] \Pr[\bar{F}_2 | F_3] \Pr[\bar{F}_1 | F_3] - \dots - \\ &\quad - \Pr[F_{|N|}] \Pr[\bar{F}_{1 \dots |N|-1} | F_{|N|}] \dots \Pr[\bar{F}_1 | F_{|N|}] \quad (5) \end{aligned}$$

The cut set count method could be calculated in polynomial time, which is suitable to be used in the GAs. However, sometimes the estimation derived by this method is not accurate enough for practical usage [8].

2.2. Monte Carlo simulation

For larger networks, Monte Carlo simulation is used to approximate the network reliability [10,11]. For the network $g = G(N, L, p)$, Monte Carlo simulation randomly generates failure scenarios $g_s = G(N, L, p)$. The random generation process is repeated many times to create multiple failure scenarios. Whenever a failure scenario g_s is randomly generated, this g_s is examined to see if every pair of nodes can communicate with each other. After this process is repeated for several iterations, the average of all feasible solutions will give an approximation to the exact all-terminal reliability.

With T iterations, the Monte Carlo simulation algorithm for measuring the all-terminal reliability of the network g is shown in Algorithm 1.

Input: A network $g = G(N, L, p)$.

Output: an approximation of $Rel_{all}(g)$.

1. Initialization: $X \leftarrow 0$.
2. For iteration $k = 1$ to T , do:
 - (a) Generate a random failure scenario g_s of g : for each link i in g , sample U from $Uniform(0,1)$; if $(U > p_i)$, $x_i=1$; else $x_i=0$.
 - (b) Use depth first search to check if every pair of nodes in g_s can communicate with each other: if true, $X \leftarrow X + 1$.
3. Return X/T .

Algorithm 1: Monte Carlo simulation.

As shown in [8], this method can approximate the all-terminal reliability more accurately than the cut set count method.

2.3. Extended reliability model

The all-terminal reliability model in Equation (2) is the summation of the probabilities that every pair of nodes remain able to communicate with each other under different network failure scenarios. According to this model, different network configurations, e.g., Fig. 2 (B) and (C) may have the same all-terminal reliability [9]. This means that the all-nodes-connected probabilities of the two networks are the same.

However, the two network configurations having the same all-nodes-connected probability does not imply that all their network failure scenarios have the same impact on them, e.g., the link failure on link (2, 3) for the network configuration in Fig. 2 (B) causes four pairs of nodes losing their connectivity, while the same link failure only disconnects three pairs of nodes in Fig. 2 (C). Because the all-terminal reliability is the summation of probabilities, it only represents the probability that every pair of nodes can communicate with each other in that network configuration. In real-life networks, different pairs of nodes have different importance to the whole network, and each pair of nodes shall have a connection importance level (CIL) [9]. If the connectivity of a node pair is more important in the network, its CIL would be higher than all other pairs of nodes in the network.

With a CIL defined on each pair of nodes, the extended all-terminal reliability of the network g is:

$$Rel'_{all}(g) = \sum_{g_s \in \Omega} Rel(g_s) + \sum_{g_s \notin \Omega} w(g_s) * Rel(g_s) \quad (6)$$

The $Rel'_{all}(g)$ includes two parts. The first part is the fully connectivity probability (every pair of nodes can communicate with each other), which is the same as the basic model in (2). The second part is related to partial connectivity probability (not every pair of nodes can communicate with each other). $g_s = G(N, L_s, p)$ is a failure scenario of g . Ω is the set of sub-networks that every pair of nodes can communicate with each other, and $Rel(g_s)$ is the reliability of g_s . As defined in [9], $w(g_s)$ is the weighting function as follows.

$$w(g_s) = \left(\frac{\sum(\text{CIL of the node pairs remain connected in } g_s)}{\sum(\text{CIL of all node pairs in } g)} \right)^n, \quad \text{for } n \geq 0. \quad (7)$$

The extended model in (6) extends the basic model in (2), and its calculation is thus more complex than that of (2).

3. Proposed approximation algorithm

To solve the all-terminal reliability network design (optimal cost) problem using GAs and adopt the extended model in (6) to calculate the reliability constraint, apparently an approximation of the extended reliability measure should be used. If we could not find an approximation algorithm to calculate the reliability constraint in polynomial time, measuring the fitness of each gene would require exponential time, and therefore makes the GAs impractical at all.

As described in Section 2, the cut set count method and Monte Carlo simulation are useful for estimating the reliability of the basic model. Based on the Monte Carlo simulation for the basic reliability model, we propose an algorithm (see Algorithm 2) to approximate the extended reliability model.

In this algorithm, a network failure scenario g_s is randomly generated according to the link failure probabilities. With T iterations, $(1/T)*W$ is an estimation for $Rel'_{all}(g)$, W is the summation of all the $w(g_s)$ values generated in the T iterations. U is a random number sampled uniformly from (0, 1) for each link i in g . Let indicator x_i be a random variable indicating whether the i -th link of g is working or not. That is

$$x_i = \begin{cases} 1 & \text{if } U > p_i \text{ (link working),} \\ 0 & \text{otherwise (link failure).} \end{cases}$$

Together, the variables $x_1, x_2, \dots, x_{|L|}$ generate a failure scenario g_s of g , $g_s = G(N, L_s, p)$ and $L_s \subseteq L$.

Input: a network $g = G(N, L, p)$.

Output: an approximation of $R'_{all}(g)$.

1. Initialization: $W \leftarrow 0$.
2. For iteration $k = 1$ to T, do:
 - (a) Generate a random failure scenario g_s of g : for each link i in g , sample U from $Uniform(0,1)$; if $(U > p_i)$, $x_i=1$; else $x_i=0$.
 - (b) Use depth first search to check if every pair of nodes in g_s can communicate with each other and compute $w(g_s)$; $W \leftarrow W + w(g_s)$.
3. Return $(1/T)*W$.

Algorithm 2: Estimation algorithm for the extended all-terminal reliability model.

We use these definitions to prove the correctness of Algorithm 2.

Lemma 1: The probability that a g_s to be chosen in one iteration is $Rel(g_s)$.

Proof: $\Pr[\text{a } g_s \text{ is chosen}]$
 $= \Pr[\text{the variables } x_1, x_2, \dots, x_{|L|} \text{ match the link configuration in } g_s]$
 $= \prod_{i \in \text{working}} (1-p_i) \prod_{i \notin \text{working}} p_i$
 $= Rel(g_s). \quad \blacksquare$

For a chosen g_s , let Y_i be an indicator random variable with

$$Y_i = \begin{cases} 1 & \text{if the } g_s \text{ is chosen in the } i\text{-th iteration,} \\ 0 & \text{otherwise.} \end{cases}$$

Because we use an independent and identical method for choosing g_s in the T iterations, Y_1, Y_2, \dots, Y_T are independent identically distributed (i.i.d.) random variables. By **Lemma 1**, we obtain that for a chosen g_s

$$\Pr[Y_i = 1] = Rel(g_s). \quad (8)$$

Lemma 2: Among T iterations, if N_{g_s} of them choose a g_s , then $Rel(g_s) \approx N_{g_s}/T$ for a sufficiently large T.

Proof: Let $Y = (1/T)*(Y_1 + Y_2 + \dots + Y_T)$ be sample mean of Y_1, Y_2, \dots, Y_T , then $Y = N_{g_s}/T$ and

$$E[Y] = (1/T)*T*E[Y_i] = \Pr[Y_i = 1] = Rel(g_s). \quad \text{(by Equation (8))} \quad (9)$$

Because $Y_i = Y_i^2$, so

$$E[Y_i^2] = E[Y_i] = Rel(g_s), \quad (10)$$

and

$$\begin{aligned} \text{Var}[Y_i] &= E[Y_i^2] - E^2[Y_i] \\ &= Rel(g_s) - Rel^2(g_s). \end{aligned} \quad (11)$$

By differentiation on (11), we obtain that $\text{Var}[Y_i]$ has a maximum 0.25 when $Rel(g_s)=0.5$. Applying Chebyshev inequality, we obtain $\Pr[|Y - E[Y]| \geq \varepsilon] \leq \text{Var}[Y]/\varepsilon^2$ which is

$$\Pr[|N_{g_s}/T - Rel(g_s)| \geq \varepsilon] \leq (\text{Var}[Y_i]/T)/\varepsilon^2 \leq 0.25/T\varepsilon^2 \quad (12)$$

Therefore, by choosing a sufficiently large number of samples T , the probability that we make an inaccurate estimation (which differs from $Rel(g_s)$ by more than ε) will be as close to zero as we wish¹. In other words, it is a high probability that our estimation is accurate. ■

Corollary 3: Suppose among T iterations, if N_{g_s} of them have chosen a g_s , then $w(g_s)*Rel(g_s) \approx w(g_s)*(N_{g_s}/T)$ for a sufficiently large T .

Proof: Proved by **Lemma 2**. ■

Lemma 4: $w(g_s) = 1$ if every pair of nodes in g_s can communicate with each other.

Proof: Proved by the definition of $w(g_s)$ in (7). ■

Theorem 5 (Correctness of Algorithm 2): Let N_{g_s} be the total occurrences of chosen g_s among T iterations and g_{sk} be the k -th chosen g_s in the T iterations,

$$Rel'_{all}(g) \approx (1/T) * \sum_{k=1}^T w(g_{sk}) \quad \text{for a}$$

sufficiently large T .

Proof: From (6) and the definition of $w(g_s)$, the reliability of the extended model is

$$\begin{aligned} Rel'_{all}(g) &= \sum_{g_s \in \Omega} 1 * Rel(g_s) + \sum_{g_s \notin \Omega} w(g_s) * Rel(g_s) \\ &= \sum_{g_s \in \Omega} w(g_s) * Rel(g_s) + \sum_{g_s \notin \Omega} w(g_s) * Rel(g_s) \\ &\quad \text{(by Lemma 4)} \end{aligned}$$

¹ Note that with the number of samples increases, the required computation effort also increases. To reduce the complexity while retaining the estimated accuracy is an important issue for future study.

$$\begin{aligned} &= \sum_{g_s} w(g_s) * Rel(g_s) \\ &\approx \sum_{g_s} w(g_s) * (N_{g_s}/T) \quad \text{(by Corollary 3)} \\ &= (1/T) * \sum_{g_s} w(g_s) * N_{g_s} \\ &= (1/T) * \{w(g_{s1}) + w(g_{s2}) + \dots + w(g_{sT})\} \\ &= (1/T) * \sum_{k=1}^T w(g_{sk}). \quad \blacksquare \end{aligned}$$

The accuracy of Algorithm 2 increases as the number of iterations increases. It gives an excellent approximation as long as ε is small and we choose T which is large enough to make $0.25/T\varepsilon^2$ sufficiently small (as we has shown earlier in (12)). In other words, if we want to have $1-2\delta$ confidence that the estimated reliability falls in the range of exact reliability $\pm \varepsilon$, we must set $0.25/T\varepsilon^2 \leq \delta$, i.e., set $T \geq 0.25/(\delta*\varepsilon^2)$.

4. Numerical results

In this section, we utilize the network configuration in Fig. 2 (A) as an example to demonstrate Algorithm 2. More complex scenarios are under study at the time this paper is written. We set the link failure probabilities and the CIL of the links in Fig. 2 (A) with the values in Tab. 1 (A) and (B). With the parameter n in $w(g_s)$ to be 1 (see Eq.(7))², the exact all-terminal reliability of the network configuration is 0.98032.

Table 1: Properties of the links.

(A) Link failure probabilities.

Links	2	3	4
1	0.1	0.2	1
2		0.2	0.1
3			0.1

(B) CIL.

Links	2	3	4
1	2	2	1
2		2	1
3			2

The estimated results of the network configuration are illustrated in Fig. 3 and Fig. 4. Fig. 3 illustrates the relation between the number of the iterations (x-axis) and the estimated reliability (y-axis). Fig. 4 illustrates the relation between the number of iterations (x-axis)

² Note that the cases for n unequal to 1 are exactly the same as the $n=1$ case, so they are not elaborated here.

and the difference between the exact reliability and the estimated reliability.

In Fig. 3, the estimated reliability approaches 0.98032 as the number of iteration increases. When more than 118 iterations are performed, the difference between the exact reliability and the estimated reliability is always smaller than 0.005 (see Fig. 4).

Fig. 4 shows that the estimated reliability converges to the exact reliability with a sufficiently large T value. Although for some small network configuration, we could directly examine the $2^{|L|}$ different g_s to calculate the exact all-terminal reliability, it is almost infeasible to do that for more complex network configurations. For large networks, approximation is required to estimate the all-terminal reliability in order to get the computation done quickly.

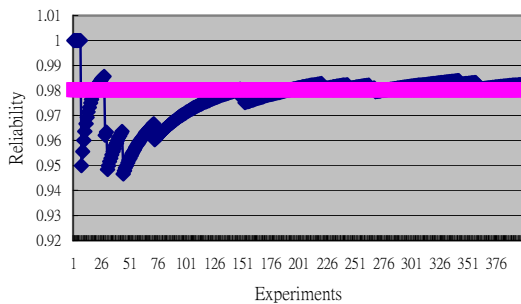


Figure 3: Estimated all-terminal reliability.

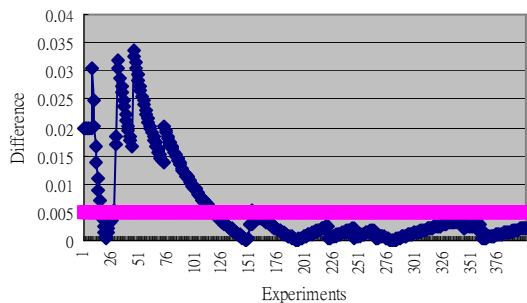


Figure 4: Difference between the exact reliability and the estimated reliability.

5. Conclusions

This paper described the optimal cost reliable network design problem which is NP-hard. To solve this complex problem, several GAs were proposed to minimize network cost given a minimum required network reliability constraint. In these GAs, the all-terminal network reliability needs to be calculated very frequently. However, the exact calculation of the all-terminal network reliability is NP-hard, so

approximation is required. For the basic network reliability model, the approximation of all-terminal network reliability was calculated using cut set count and Monte Carlo simulation. Nevertheless, for the extended network reliability model, no approximation algorithm was developed.

We proposed an approximation algorithm based on Monte Carlo simulation for that extended network reliability model. By adopting our network reliability approximation algorithm, the GAs will be capable of solving the optimal cost reliable network design problem under the extended network reliability model.

We did not mention the computational intensive of the proposed algorithm in this paper. This is an important issue for future study.

6. References

- [1] A. Kershenbaum, *Telecommunications Network Design Algorithms*, McGraw-Hill, New York, NY, 1993.
- [2] B. Dengiz, F. Altıparmak and A.E. Smith, "Local search genetic algorithm for optimal design of reliable networks", *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 3, Sep. 1997, pp. 179-188.
- [3] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Kluwer Academic Publishers, Boston, MA, 1989.
- [4] Y. S. Kavian and M. Naderi, "An Evolutionary Approach to Design WDM Telecommunication Survivable Networks" in *Proc. IEEE International Conference on Engineering of Intelligent Systems*, Sep. 2006, pp. 1-6.
- [5] S. Zhang, J. Chen, J. Jia and X.-R. Bao, "Searching Algorithm of the Optimal Sensor Set in Wireless Sensor Network", in *Proc. IEEE International Conference on Wireless Communications, Networking and Mobile Computing*, Sep. 2006, pp. 1-4.
- [6] M.R. Garey and D.S. Johnson, *Computers and Intractability*. Freeman, New York, NY, 1979.
- [7] A. Konak and A.E. Smith, "An improved general upper bound for all-terminal network reliability", in revision at *IIE Transactions*.
- [8] D.L. Deeter and A.E. Smith, "Economic design of reliable networks", *IIE Transactions*, vol. 30, 1998, pp. 1161-1174.
- [9] M. Feng, H.L. Vu and M. Zukerman, "A new reliability measure for telecommunication networks", *IEEE Communications Letters*, vol. 6, no. 9, Sep. 2002, pp. 400-402.
- [10] K.-P. Hui, *Network Reliability Estimation*, Ph. D. Thesis, University of Adelaide, South Australia, 2005.
- [11] M.O. Ball, T.L. Magnanti, C.L. Monma and G.L. Nemhauser (eds.), *Network Models: Handbooks in Operations Research and Management Science*, vol. 7, North-Holland, Amsterdam, 1995.