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

AN EFFICIENT COMPLETE ENUMERATION METHOD FOR NETWORK DESIGN PROBLEMS AND ITS APPLICATIONS

Takeshi Koide
University of Marketing & Distribution Sciences

Shuichi Shinmori
Kagoshima University

Hiroaki Ishii
Osaka University

(Received October 31, 2001; Revised February 1, 2002)

Abstract In many network design problems, the best layout of components is searched considering construction cost and network reliability. Because of #P-completeness for computation of network reliability, most algorithms for the problems find approximate solutions. In this paper, we propose a complete enumeration method for the network design problems, which becomes a basis of exact algorithms. The detection of isomorphic networks appeared in the process of computation can reduce computational time compared with simple complete enumerations. We also discuss applications of the algorithm to other kinds of network design problems.

1. Introduction

In most network design problems, the best layout of components is searched considering construction cost and quality of services (QoS). There are many measurements on QoS such as reliability, throughput, transmission delay and packet loss probability. Since the possibility to communicate among terminals is quite significant, reliability, which estimates connectivity among terminals, is often adopted in network design problems [1, 5, 7, 11, 13, 12, 14, 15, 16, 23, 24]. Network reliability is generally the probability that target terminals can communicate each other under failures of components. There are famous network reliabilities such as all-terminal reliability, two-terminal reliability and k-terminal reliability, and it is $\#P$-complete to compute the network reliabilities exactly [6, 22]. In this paper, we focus on all-terminal reliability, that is the probability all nodes are connected by operational edges, but the contents of this paper are applicable to two-terminal reliability and k-terminal reliability. Many researchers have proposed algorithms for the network design problem to find an optimum network layout considering construction cost and all-terminal reliability. Most of the algorithms find approximate solutions due to the #P-completeness of computing all-terminal reliability. At first approximate solutions were searched by heuristic methods [1, 5, 12, 23, 24] and recently meta heuristics, such as genetic algorithms, tabu search and simulated annealing, have been applied even to more complex problems [3, 4, 7, 14, 15, 16, 17, 20, 21]. Jan *et al.* [11] made the first attempt to find the exact solution for the problem and Koide *et al.* [13] proposed an exact algorithm that can be applied to more general models than Jan's.

In this paper, we propose an algorithm as a complete enumeration method for the network design problems. A complete enumeration method needs enormous computational time in general but not only it becomes a basis of exact algorithms but also it is indispensable to estimate performances of other exact algorithms. Our algorithm can execute in shorter time than simple complete enumerations since it detects isomorphic networks in the process of computing and refers to computed results efficiently. Moreover, we discuss applications of the proposed algorithm to other kinds of network design problems, such as biobjective optimization, drawing state transition networks, network redesign problems and sensitivity analysis of network reliability.

In section 2, we define a network design problem and explain a simple complete enumeration method for the problem. We propose an improved complete enumeration in section 3. In section 4, numerical experiments show the efficiency of our algorithm. Section 5 discusses applications of the algorithm to other kinds of network design problems. Section 6 concludes this paper.

2. Preliminaries

2.1. Notations

- $G = (V, E)$: target network;
- V: set of n nodes;
- $E = \{e_1, e_2, \ldots, e_m\}$: set of m selectable edges;
- $E[G]$: set of edges in G;
- $G - e := (V, E - \{e\})$: the network obtained by deleting an edge $e \in E$ from G;
- $G - E' := (V, E - E')$: the network obtained by deleting all edges in $E' \subseteq E$ from G;
- G/e: the network obtained by contracting an edge $e \in E$ in G;
- G/E': the network obtained by contracting all edges in $E' \subseteq E$ in G;
- $G \cup e := (V, E \cup \{e\})$: the network obtained by adding an edge $e \notin E$ to G;
- $G_1 \subset G_2$: $E_1 \subset E_2$ where $G_1 = (V, E_1)$, $G_2 = (V, E_2)$;
- p_i: edge probability of edge $e_i \in E$, that is the probability e_i operates;
- c_i: construction cost for edge $e_i \in E$;
- x_i: integer variable on edge $e_i \in E$;
- $\boldsymbol{x} := [x_1, x_2, \ldots, x_m]$;
- $R(G)$: all-terminal reliability for G;
- $G_{\boldsymbol{x}}$: the network determined by $\boldsymbol{x}$;
- $E_{\boldsymbol{x}}$: the set of edges determined by $\boldsymbol{x}$;
- $t(G_{\boldsymbol{x}})$: the order of computations for $R(G_{\boldsymbol{x}})$ among all networks;

2.2. Network design problem and complete enumeration method

We consider a network design problem to find the optimal network layout considering total construction cost and all-terminal reliability. Let $G = (V, E)$ be target network which consists of a set of n nodes V and a set of m selectable edges $E = \{e_1, e_2, \ldots, e_m\}$ where $n \geq 2$ and $m \geq 1$. We assume that G is connected and has neither parallel edges nor self-loops. A network $G' = (V, E')$ where $E' \subseteq E$ is called *a subnetwork* of G. If we select $e_i \in E$ as a component of a subnetwork, we have to pay c_i as construction cost. Let x_i be a 0-1 variable on e_i if e_i is selected then $x_i = 1$, otherwise $x_i = 0$. Let $G_{\boldsymbol{x}} = (V, E_{\boldsymbol{x}})$ be the subnetwork which consists of V and the set of selected edges $E_{\boldsymbol{x}}$, i.e., $E_{\boldsymbol{x}} = \{e_i | x_i = 1, \ i = 1, \ldots, m\}$. Our aim is to find the optimal subnetwork $G_{\boldsymbol{x}}$ among all subnetworks. We call the network design problem NDP which is formulated in two kinds of forms as follows:

$$\text{NDP} \quad \left| \begin{array}{ll} \text{minimize} & \sum_{i=1}^{m} c_i \cdot x_i \\ \text{subject to} & R(G_{\boldsymbol{x}}) \geq R_0, \\ & x_i \in \{0,1\} \ i = 1, \ldots, m \end{array} \right. \quad \text{or} \quad \left| \begin{array}{ll} \text{maximize} & R(G_{\boldsymbol{x}}), \\ \text{subject to} & \sum_{i=1}^{m} c_i \cdot x_i \leq c_0 \\ & x_i \in \{0,1\} \ i = 1, \ldots, m \end{array} \right.$$

where R_0 and c_0 are reliability requirement and limit of cost, respectively.

$R(G_{\boldsymbol{x}})$ denotes all-terminal reliability of $G_{\boldsymbol{x}}$. We make some assumptions to define all-terminal reliability. Edges have two states, operational and failed, and the edge probability of $e_i \in E$, which is the probability e_i operates, is denoted by p_i. All edge probabilities are assumed to be statistically independent each other. Nodes always operate. Then, the probability that all nodes are connected by operational edges in G is called *all-terminal reliability* of G, denoted by $R(G)$. It is $\#P$-complete to compute the exact value of $R(G)$ and it is believed that there is no algorithm to compute $R(G)$ in polynomial time on m [6, 22].

We show a famous theorem about $R(G)$ that is called *the factoring theorem* [6, 22], which is frequently used to compute $R(G)$.

Theorem 1 *For a network $G = (V, E)$ and an edge $e_i \in E$, the following equation holds:*

$$R(G) = \begin{cases} R(G - e_i) & e_i : \textit{self-loop} \\ p_i R(G/e_i) & e_i : \textit{bridge} \\ (1 - p_i)R(G - e_i) + p_i R(G/e_i) & \textit{otherwise} \end{cases} \tag{1}$$

where $G - e_i$ and G/e_i denote the networks constructed by deleting and by contracting an edge e_i in G, respectively. A bridge is the edge whose removal makes networks disconnected. We can compute $R(G)$ by applying equation (1) to networks recursively until reliability can be computed easily, i.e., until networks have quite a few edges. Such a method to compute $R(G)$ is called *a factoring method* and the expansion of networks by equation (1) is called *a factoring.* The constructed networks by factorings are called *minors.*

A simple algorithm for a factoring method, named Procedure SFM, is shown as follows.

Procedure SFM(G)
1. **if** G is disconnected **then return** 0;
2. **if** $|E[G]| \leq 1$ **then return** $R(G)$;
3. extract the minimum index edge e_i from $E[G]$;
4. **if** e_i is self-loop **then return** SFM($G - e_i$);
5. **return** $(1 - p_i) \times$ SFM$(G - e_i) + p_i \times$ SFM(G/e_i);

end.

When G has no edge and more than one node, G is disconnected and $R(G) = 0$ is returned. Factorings are applied until networks have at most one edge (see step 2). Though we can extract any edge from E in step 3 in general, we assume to extract edges in increasing order of their indices in this paper. When e_i is a bridge in step 5, Procedure SFM($G - e_i$) obtains 0 since $G - e_i$ is disconnected, which satisfies equation (1). The time complexity of Procedure SFM(G) is $O((n + m)2^m)$ since step 1 needs $O(n + m)$.

Example 1 *In Figure 1, we show an execution of Procedure SFM(G) where $q_i = 1 - p_i$ ($i = 1, 2, 3$). Totally six minors are constructed, named G_1, G_2, G_{11}, G_{12}, G_{21} and G_{22}. Since G_{11} is disconnected, $R(G_{11}) = 0$. Since there is only one node in G_{22}, $R(G_{22}) = 1$. We can ascertain that G_{12} and G_{21} are isomorphic. $R(G)$ can be computed by the equation* (1).

Imai et al. [10] proposed an excellent method to compute reliability of a given network via BDDs (Binary Decision Diagrams) and it could be applied to large networks with about two hundred edges. We call their method *BDD method* appeared in Section 4.

From now on, we consider a complete enumeration method for NDP. A complete enumeration method is the algorithm to estimate all solutions and select an optimum solution. In this paper, we focus on the part to estimate all solutions. A complete enumeration method for NDP needs to compute both construction cost and reliability for all subnetworks. While

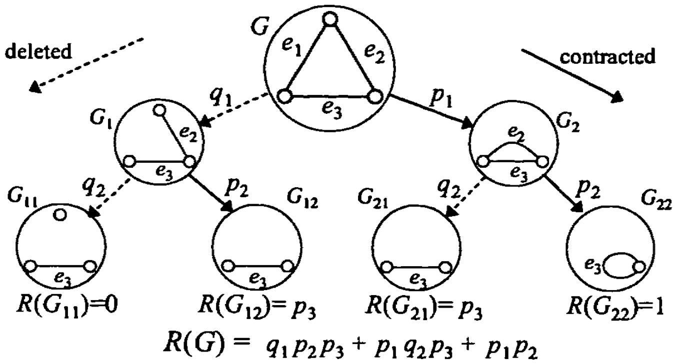


Figure 1: Execution of Procedure SFM

the construction cost of a given subnetwork $G_{\boldsymbol{x}}$ is computed in $O(E[G_{\boldsymbol{x}}])$, the computational time for reliability is expected to need exponential time on $|E[G_{\boldsymbol{x}}]|$ because of its $\#P$-completeness.

A simple algorithm as a complete enumeration method, named Procedure SCEM, is shown as follows. Procedure SCEM outputs both reliability and cost for all subnetworks $G_{\boldsymbol{x}}$ in lexicographical order on $\boldsymbol{x}$.

Procedure SCEM($G = (V, E)$)
1. let $\boldsymbol{x} := [x_1, \ldots, x_m]$ where $m = |E|$;
2. execute $sub\_scem(\boldsymbol{x}, 0, 1)$;

end.

Procedure $sub\_scem(\boldsymbol{x}, c, i)$
1. **if** $i > m$ **then**
2. $rel := R(G_{\boldsymbol{x}})$ by Procedure SFM($G_{\boldsymbol{x}}$);
3. **output** rel and c;
4. **else**
5. $x_i := 0$, execute $sub\_scem(\boldsymbol{x}, c, i+1)$;
6. $x_i := 1$, execute $sub\_scem(\boldsymbol{x}, c + c_i, i+1)$;
7. **end if**;

end.

Step 5 and 6 in Procedure *sub_scem* enumerate all subnetworks in lexicographical order. In the following, we estimate the time complexity of Procedure SCEM.

Theorem 2 *The time complexity of Procedure SCEM(G) is* $O((n+m)3^m)$.

Proof : Procedure SCEM constructs ${}_mC_k$ subnetworks with k edges where $k = 0, 1, \ldots, m$. The time complexity of Procedure SFM($G_{\boldsymbol{x}}$) is $O((n + |E[G_{\boldsymbol{x}}]|)2^{|E[G_{\boldsymbol{x}}]|})$. Hence, the total time complexity is $O((n+m)3^m)$ since

$$\sum_{k=0}^{m} {}_mC_k(n+k)2^k = n\sum_{k=0}^{m} {}_mC_k 2^k + 2\sum_{k=1}^{m} {}_mC_k k 2^{k-1} = n3^m + 2m3^{m-1} = 3^{m-1}(3n+2m).$$

□

3. Main Results

3.1. Existence of isomorphic networks

In factoring methods, two operations, deletions and contractions, construct minors. If a minor G_1' is constructed in computing reliability for a subnetwork G_1, G_1 is called *source subnetwork* of G_1'. In a minor G_1', an edge e_i has one of the following four states.

(a) e_i does not exist in the source subnetwork G_1
(b) e_i exists in the minor G'_1
(c) e_i existed in the source subnetwork G_1 and has deleted
(d) e_i existed in the source subnetwork G_1 and has contracted.

According to the edge states, we redefine the variable x_i for edge e_i $(i = 1, \ldots, m)$ as follows:

$$x_i = \begin{cases} 0 & e_i \text{ does not exist in the source subnetwork} \\ 1 & e_i \text{ exists in the minor} \\ 2 & e_i \text{ existed in the source subnetwork and has deleted} \\ 3 & e_i \text{ existed in the source subnetwork and has contracted.} \end{cases} \tag{2}$$

In the following, we show several properties about the new $\boldsymbol{x} = \{x_1, \ldots, x_m\}$.

Property 1 *For $G = (V, E)$, $\forall e_1 \in E$ and $\forall e_2 \in E$ $(e_1 \neq e_2)$. Then,*

- $(G - e_1) - e_2$ *is isomorphic to* $(G - e_2) - e_1$
- $(G/e_1)/e_2$ *is isomorphic to* $(G/e_2)/e_1$
- $(G - e_1)/e_2$ *is isomorphic to* $(G/e_2) - e_1$
- $(G/e_1) - e_2$ *is isomorphic to* $(G - e_2)/e_1$.

Proof: Contracting an edge is deleting an edge and unifying the terminals of the edge. Hence, structures of constructed networks are independent of the order among deletions and contractions. □

Property 1 shows that the structure of minors is independent of the order of operations. Hence, all minors are represented by the new $\boldsymbol{x}$ defined in equation (2). We redefine $G_{\boldsymbol{x}}$ as the network represented by the new $\boldsymbol{x}$. Note that the new $G_{\boldsymbol{x}}$ can still also represent subnetworks. We call $\boldsymbol{x}$ *the source vector* of network $G_{\boldsymbol{x}}$.

Example 2 *Figure 2 indicates the execution of Procedure* SCEM *for the network shown in Figure 1. Source vectors are denoted beneath networks. Parenthesized numbers represent the order of networks whose reliability Procedure* SCEM *has computed.*

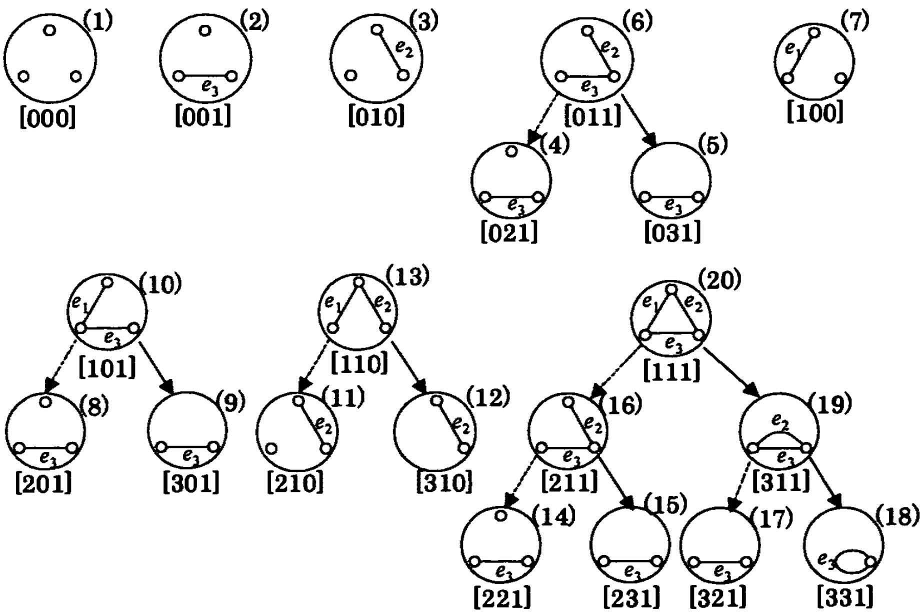


Figure 2: Procedure SCEM for the network shown in Figure 1

Property 2 *Let $\boldsymbol{x} = \{x_1, \ldots, x_m\}$ be a source vector and $x_i = 2$. Now we construct a new source vectors $\boldsymbol{y}$ as follows*

$$y_j = \begin{cases} 0 & j = i \\ x_j & j \neq i. \end{cases}$$

Then, the two networks $G_{\boldsymbol{x}}$ and $G_{\boldsymbol{y}}$ are isomorphic and $R(G_{\boldsymbol{x}}) = R(G_{\boldsymbol{y}})$.

Proof: Since Property 1 indicates that edge states in (a) and (c) are identical in considering structures of networks, then $G_{\boldsymbol{x}}$ and $G_{\boldsymbol{y}}$ are isomorphic. It holds that $R(G_{\boldsymbol{x}}) = R(G_{\boldsymbol{y}})$ since all edge probabilities are identical in the two networks. □

From now on, we call that $G_{\boldsymbol{x}}$ and $G_{\boldsymbol{y}}$ are *02-isomorphic* if the two networks satisfy Property 2. In Example 2, there are 02-isomorphic networks which are grouped as follows:

$$\begin{aligned} \mathcal{G}_1 &:= \{G_{[001]}, G_{[021]}, G_{[201]}, G_{[221]}\}, \quad \mathcal{G}_2 := \{G_{[010]}, G_{[210]}\}, \\ \mathcal{G}_3 &:= \{G_{[011]}, G_{[211]}\}, \quad \mathcal{G}_4 := \{G_{[031]}, G_{[231]}\}, \quad \mathcal{G}_5 := \{G_{[301]}, G_{[321]}\}. \end{aligned}$$

Note that the networks in $\mathcal{G}_4$ and $\mathcal{G}_5$ are isomorphic but not 02-isomorphic. Property 2 is not for all isomorphic networks.

In Example 2, when the factoring for $G_{[111]}$ constructs two minors $G_{[211]}$ and $G_{[311]}$, $R(G_{[011]})$ has been already computed. Since $G_{[211]}$ is 02-isomorphic to $G_{[011]}$, $R(G_{[211]})$ is equal to $R(G_{[011]})$. If $R(G_{[011]})$ has been memorized and can be referred to, we need not apply the factoring to $G_{[211]}$ in order to obtain $R(G_{[211]})$. This example illustrates that memorizing and referring to past computational results reduce some procedures. In the next subsection, we consider which networks to memorize and how to refer to past computational results efficiently in order to reduce total computational time.

3.2. How to refer to computed reliabilities of isomorphic networks

Let $t(G')$ be the order of a network G' whose reliability is computed in Procedure SCEM, which is represented as the parenthesized number in Figure 2. In the following, we have proved a lemma and theorems about $t(G')$.

Lemma 1 *Let G_1 and G_2 be source subnetworks with not less than two edges. Let G'_1 and G'_2 be minors of G_1 and G_2, respectively. If $G_1 \subset G_2$, the next inequality holds:*

$$t(G'_1) < t(G_1) < t(G'_2) < t(G_2). \tag{3}$$

Proof: After having computed the reliability for all the minors of a subnetwork, the reliability for the subnetwork is computed in Procedure SCEM. Then, it is proved that $t(G'_1) < t(G_1)$ and $t(G'_2) < t(G_2)$. Since the source vector of G_1 can be constructed by replacements from 1 to 0 on some variables in the source vector of G_2, the source vector of G_1 is ahead of that of G_2 in lexicographical order. Procedure SCEM computes reliability for all subnetworks in lexicographical order. Then, it is proved that $t(G_1) < t(G_2)$. It is proved $t(G_1) < t(G'_2)$ since Procedure SCEM starts to compute $R(G_2)$ after $R(G_1)$ has been computed. □

Lemma 2 *Let G_0 be a subnetwork (not a minor) with not less than one edge. If G' be a 02-isomorphic minor to G_0, then the following inequality holds:*

$$t(G_0) < t(G'). \tag{4}$$

Proof: Let G_s be the source subnetwork of G'. Since the minor G' is isomorphic to the subnetwork G_0, the number of nodes of G' is same as that of G_0, i.e., G' is constructed from G_s with some deletions of edges and without contractions. Then, $G' \subset G_s$ and $G_0 \subset G_s$. Lemma 1 proves $t(G_0) < t(G') < t(G_s)$. □

Lemma 3 *Let G_1 be a subnetwork with not less than two edges and G_1' be a minor of G_1 constructed by k contractions and finally one deletion of edges where $k = 0, 1, \ldots, E[G_1]-2$. Then, there is a subnetwork or a minor G_0 where G_0 is 02-isomorphic to G_1' and $t(G_0) < t(G_1')$.*

Proof: Assume $k > 0$. Let E_c be the set of contracted edges and e_i be the deleted edge where $|E_c| = k$ and $k < i$. Then, G_1' is represented by $(G_1/E_c) - e_i$. Let $\boldsymbol{y}$ be the source vector of $G_1' = (G_1/E_c) - e_i$. Then $y_j = 0$ or $y_j = 3$ for $j = 1, \ldots, i-1$ and $y_i = 2$. Here, we consider a subnetwork $G_1 - e_i$. Let $\boldsymbol{x}$ be the source vector of a minor $(G_1 - e_i)/E_c$ constructed from the subnetwork $G_1 - e_i$. Then $x_j = y_j$ for $j = 1, \ldots, i-1$ and $x_i = 0$, which proves that the minor $(G_1 - e_i)/E_c$ is 02-isomorphic to the minor $G_1' = (G_1/E_c) - e_i$ by Property 2. Since $G_1 - e_i \subset G_1$, $t((G_1 - e_i)/E_c) < t(G_1')$ is proved by Lemma 1. In case $k = 0$, $E_c = \phi$ and the network $(G_1 - e_i)/E_c$ is not a minor but a subnetwork. □

Lemma 2 and Lemma 3 describe that a minor (not a subnetwork) G_1' constructed by finally one edge deletion has a 02-isomorphic network (either a subnetwork or a minor) G_0 where $t(G_0) < t(G_1')$.

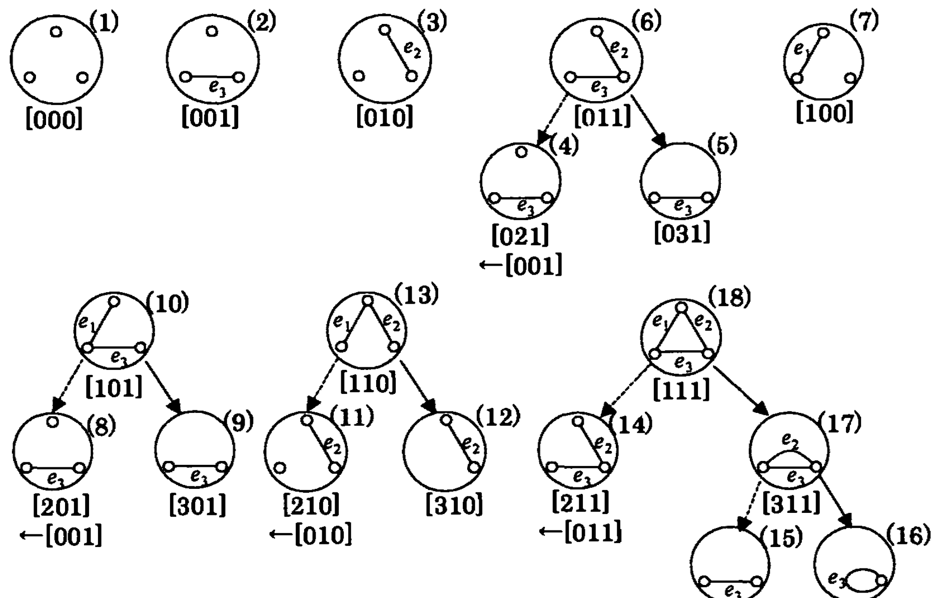


Figure 3: Referring to Computed Reliabilities of Isomorphic Networks in Figure 2

Example 3 *Figure 3 indicates how to refer to computed reliabilities of 02-isomorphic networks in Figure 2. The left arrows under source vectors denote the references to reliability of 02-isomorphic networks. All minors constructed by finally one edge deletion have 02-isomorphic networks whose reliability has been computed. In computing $R(G_{[211]})$, $R(G_{[011]})$ is referred to as $R(G_{[211]})$. Then, $G_{[221]}$ and $G_{[231]}$ do not have to be constructed.*

Figure 3 shows that all networks are not referred to their reliability. We have proved a theorem concerned with a judgment which networks should be memorized.

Lemma 4 *Let $G_{\boldsymbol{x}}$ be a network with at least one edge. Then, the number of times that $G_{\boldsymbol{x}}$ is referred to is equal to the number of elements x_i in $\boldsymbol{x}$ which satisfy the following three conditions:*

- $x_i = 0$
- $x_j = 0$ or 3 for $j = 1, \ldots, i-1$
- $x_j = 0$ or 1 for $j = i+1, \ldots, m$.

Proof: We consider the minor $(G_1 - e_i)/E_c$ of the subnetwork $G_1 - e_i$ in Lemma 3. $R((G_1 - e_i)/E_c)$ is referred to as $R((G_1/E_c) - e_i)$. Hence, if and only if $G_{\boldsymbol{x}}$ has l elements in $\boldsymbol{x}$ which satisfy the above conditions, $R(G_{\boldsymbol{x}})$ is referred to l times by 02-isomorphic networks. □

In Figure 3, $R(G_{[001]})$ is referred to twice, while $R(G_{[010]})$, $R(G_{[011]})$ and $R(G_{[301]})$ are once.

We propose the following procedures to reduce the number of memorized networks.

(1) Lemma 4 can compute the number of times that a network $G_{\boldsymbol{x}}$ is referred to. Hence, we can set free the memorized reliability if the reliability is no more referred to.

(2) If a network $G_{\boldsymbol{x}}$ is judged to be memorized by Lemma 4 but $R(G_{\boldsymbol{x}}) = 0$, $R(G_{\boldsymbol{x}})$ is not memorized. When $R(G_{\boldsymbol{x}})$ is searched but not found, it means $R(G_{\boldsymbol{x}}) = 0$.

(3) If a network $G_{\boldsymbol{x}}$ is judged to be memorized and $G_{\boldsymbol{x}}$ has at most k edges, $R(G_{\boldsymbol{x}})$ is not memorized where $k \in \{1, \ldots, m\}$. We apply Procedure SFM to compute reliability for networks with at most k edges.

When the value of k becomes bigger on procedure (3), Procedure SFM is executed to compute reliability for more networks and the computational time to compute reliabilities becomes greater. On the other hand, since the reliability for less networks are memorized, the size of memory space becomes smaller and the computational time to search reliabilities becomes also smaller. We estimate the correlation between the computational time of Procedure CEM$_k$ and the value of k on numerical experiments in Section 4.

3.3. Algorithm

We propose Procedure CEM$_k$ which compute reliabilities and costs for all subnetworks of a target network G where k is mentioned on procedure (3). Note that Procedure CEM$_m$ is identical to Procedure SCEM.

Procedure CEM$_k$($G = (V, E)$)
1. let $\boldsymbol{x} := [x_1, \ldots, x_m]$ where $m = |E|$, $\mathcal{M} := \phi$;
2. execute $sub\_cem_k(\boldsymbol{x}, 0, 1)$;

end.

Procedure $sub\_cem_k(\boldsymbol{x}, c, i)$
1. **if** $i > m$ **then**
2. $\quad rel := R(G_{\boldsymbol{x}})$ by Procedure FM$_k(\boldsymbol{x})$;
3. $\quad$ **output** rel and c;
4. **else**
5. $\quad x_i := 0$, execute $sub\_cem_k(\boldsymbol{x}, c, i+1)$;
6. $\quad x_i := 1$, execute $sub\_cem_k(\boldsymbol{x}, c + c_i, i+1)$;
7. **end if**;

end.

Procedure FM$_k(\boldsymbol{x})$
1. **if** $G_{\boldsymbol{x}}$ is disconnected **then return** 0;
2. **if** $|E[G_{\boldsymbol{x}}]| \leq k$ **then** $rel := R(G_{\boldsymbol{x}})$ by Procedure SFM($G_{\boldsymbol{x}}$);
3. **else**
4. $\quad$ extract the minimum index edge e_i from $E[G_{\boldsymbol{x}}]$;
5. $\quad$ **comment**: $x_i = 1$

```
6.   search (x', R(G_x')) from M where G_x' is 02-isomorphic to G_x - e_i;
7.   comment: x' = {x_1, ..., x_{i-1}, 0, x_{i+1}, ..., x_m}
8.   if R(G_x') is not found then R(G_x') := 0;
9.   if R(G_x') is no more searched then M := M - {(x', R(G_x'))};
10.  x_i = 3;
11.  rel := (1 - p_i) × R(G_x') + p_i × FM_k(x);
12.  x_i = 1;
13. end if;
14. if |E[G_x]| ≥ k and R(G_x) is referred to later then M := M ∪ {(x, rel)};
15. return rel;
end.
```

Procedure CEM_k is almost same as Procedure SCEM except for Procedure FM_k. We explain only Procedure FM_k. Step 2 and $|E[G_{\boldsymbol{x}}]| \geq k$ in step 14 are the procedures mentioned as procedure (3). In step 6, $R(G_{\boldsymbol{x}'})$ has already computed proved in Lemma 3. $\mathcal{M}$ is a memory for computed reliability. Procedure (1) and (2) are coded in step 9 and 8, respectively. Step 10 transforms $G_{\boldsymbol{x}}$ to $G_{\boldsymbol{x}}/e_i$. The judgment whether $R(G_{\boldsymbol{x}})$ is referred to later in step 14 is executed by Lemma 4.

In the following, we estimate the complexity of Procedure CEM_k.

Theorem 3 *The space complexity of Procedure $CEM_k(G)$ is $O((m-k)2^m)$.*

Proof: We estimate the total number of memorized networks, which leads to an upper bound of $|\mathcal{M}|$. Let $\mathcal{V}_m$ be the set of source vectors of networks which are referred to by 02-isomorphic networks and $N_m \equiv |\mathcal{V}_m|$. $\mathcal{V}_m$ is the set of source vectors of networks which have an element which satisfies the three conditions in Lemma 4. Moreover, the networks should have at least k edges (see step 14 in Procedure CEM_k). We proceed the proof by induction on $m \geq k$. If $m = k$, obviously $N_k = 0$. For the case $m = k+1$, assume $\boldsymbol{x} = [x_1, x_2, \ldots, x_m] \in \mathcal{V}_m$. Then, it holds that $x_1 = 0$ and $x_2 = \cdots = x_m = 1$. Hence, $N_{k+1} = 1$. Suppose $m > k+1$ and assume $\boldsymbol{x} = [x_1, x_2, \ldots, x_m] \in \mathcal{V}_m$. If $[x_2, \ldots, x_m] \in \mathcal{V}_{m-1}$, then $x_1 = 0$ or 3. Otherwise, $x_1 = 0,\ x_2 = 1,\ x_i = 0$ or 1 for $i = 3, \ldots, m$ and there are at least $k-1$ 1s in $\{x_3, \ldots, x_m\}$. Therefore, we obtain the next recurrence formula:

$$N_m = 2N_{m-1} + \sum_{i=k-1}^{m-2} {}_{m-2}C_i.$$

Let $\bar{N}_m$ be a upper bound of N_m. Then, we get a recurrence formula on $\bar{N}_m$:

$$\bar{N}_m = 2\bar{N}_{m-1} + 2^{m-2}$$

and initial conditions $\bar{N}_k = N_k = 0$ and $\bar{N}_{k+1} = N_{k+1} = 1$. By $k \geq 1$, we prove

$$\bar{N}_m = (m-k-1)2^{m-2} + 2^{m-k-1} < (m-k)2^{m-2}.$$

□

Theorem 4 *The number of networks whose reliability is computed in Procedure $CEM_k(G)$ are $O(2^m(m+2^k))$.*

Proof: Let $N_k(l)$ be the number of networks whose reliability is computed in Procedure $\mathrm{FM}_k(G')$ where G' is a subnetwork with l edges. We assume to apply factorings even to disconnected networks, i.e., assume to neglect the step 1 in Procedure FM_k in order to estimate $N_k(l)$ simply. Lemma 3 can draw Figure 4 that shows the process of computing

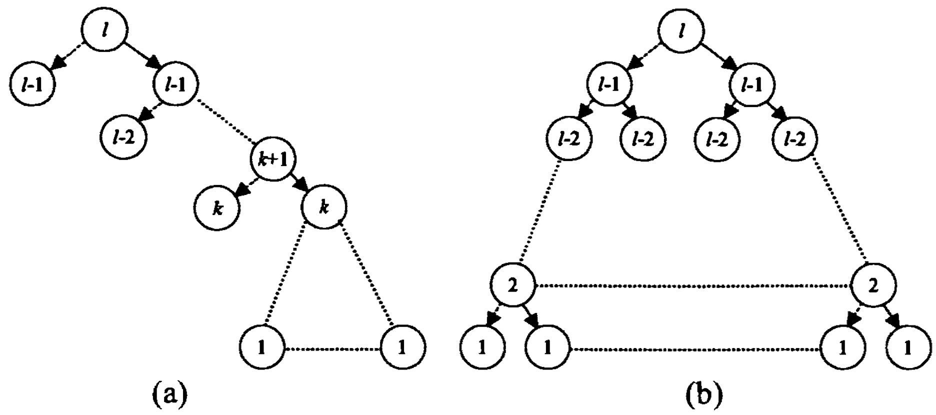


Figure 4: Process of Procedure FM_k: (a) $1 \leq k \leq l$ (b) $l \leq k \leq m$

reliability of a subnetwork with l edges by Procedure FM_k. In Figure 4, circles denote networks and the numbers in circles denote the number of edges in the networks. We can easily compute $N_k(l)$ by Figure 4:

$$N_k(l) = 2^{\min(l,k)} - 1 + 2\max(l-k, 0).$$

As a special case, we set $N_k(0) = 1$ for any k. Let N_k be the number of networks whose reliability is computed in Procedure $\mathrm{CEM}_k(G)$. Since $m \geq 1$ and $k \geq 1$, we get

$$\begin{aligned} N_k &= \sum_{l=0}^{m} {}_mC_l N_k(l) = 1 + \sum_{l=1}^{m} {}_mC_l\{2^{\min(l,k)} - 1 + 2\max(l-k,0)\} \\ &< 1 + \sum_{l=1}^{m} {}_mC_l\{2^k - 1 + 2(l-1)\} \\ &= 1 + (2^k - 3)(2^m - 1) + 2 \cdot m2^{m-1} \\ &< 2^m(m + 2^k). \end{aligned}$$

□

Theorem 5 *The time complexity of Procedure $CEM_k(G)$ is $O(2^m(n+m)(m+2^k))$.*
Proof: Since the size of $\mathcal{M}$ is $O((m-k)2^m)$, the time complexity of the procedure to search reliabilities in step 6 of Procedure FM_k is $O(m)$ by using binary search trees. Then, the most time-consuming procedure in Procedure FM_k except Procedure SFM is to check connectedness in step 1, which is $O(n+m)$. Theorem 4 proves that the time complexity of Procedure $\mathrm{CEM}_k(G)$ is $O(2^m(n+m)(m+2^k))$. □

4. Numerical Experiments

We have constructed a program in C language as Procedure CEM_k. Numerical experiments are executed on a Windows2000 PC, which has Pentium-III 1GHz CPU and 128MB memory. In this numerical experiments, target networks are generated so that their edge connectivities are as great as possible under given values of n and m.

4.1. Performance analysis of Procedure CEM_k on the valur of k

We apply Procedure CEM_k to a network with 8 nodes and 20 edges for considerable value of k in order to estimate the number of networks whose reliability is computed (named

Table 1: Estimation of Procedure CEM_k for a Network where $n = 8$ and $m = 20$

k	RELNET	CPU Time [s]	MEMNET (A)	MAX-M (B)	A/B [%]
1	14,121,458	28.0	3,820,508	467,780	12.2
2	14,121,458	27.3	3,457,010	467,745	13.5
3	14,384,077	26.8	3,092,877	467,598	15.1
4	15,369,533	26.9	2,723,311	467,118	17.2
5	17,837,166	28.4	2,352,174	465,585	19.8
6	23,148,859	32.5	1,980,520	461,122	23.3
7	33,796,666	41.8	1,608,808	448,229	27.9
8	54,369,219	60.7	1,237,317	414,412	33.5
9	92,462,890	96.4	880,494	350,980	39.9
10	157,771,458	158.8	566,473	263,970	46.6
11	256,886,397	254.6	322,410	172,243	53.4
12	385,027,446	379.8	159,005	95,591	60.1
13	522,195,278	515.2	66,512	44,301	66.6
14	640,952,992	633.8	23,033	16,784	72.9
15	722,165,855	716.3	6,406	5,056	78.9
16	764,797,002	760.3	1,372	1,165	84.9
17	781,309,652	777.5	212	192	90.6
18	785,745,047	782.1	21	20	95.2
19	786,481,698	782.9	1	1	100.0
20	786,538,624	783.0	0	0	—

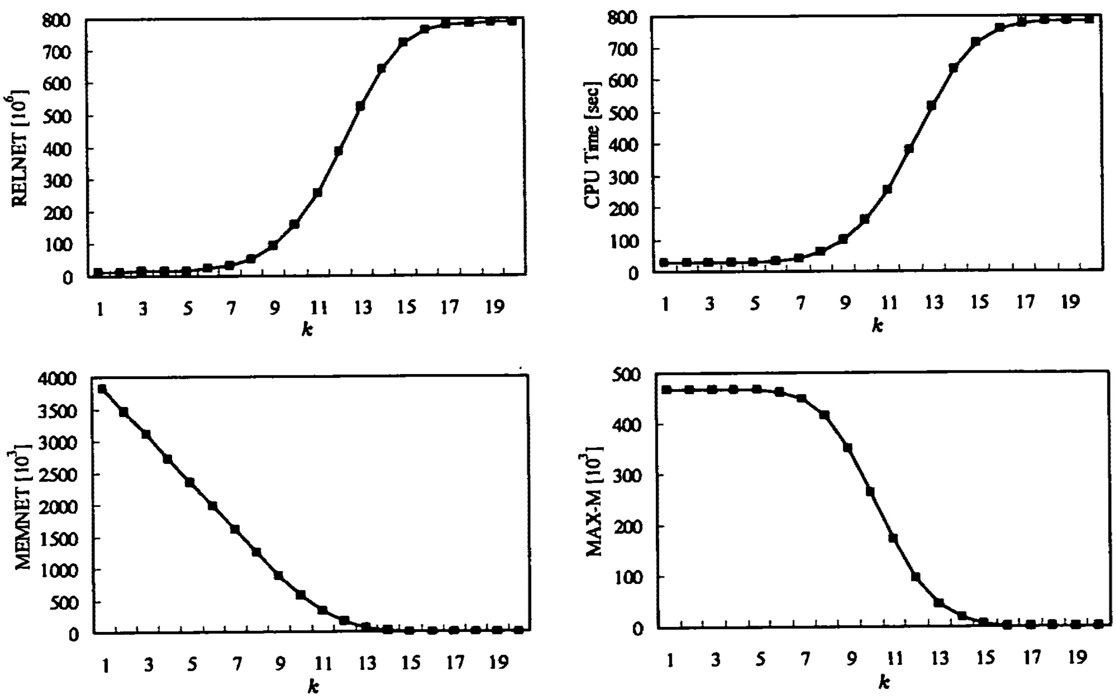


Figure 5: Estimation of Procedure CEM_k for a Network where $n = 8$ and $m = 20$

Table 2: Estimation of Procedure CEM_k

n	m	k^*	CEM_{k^*}/CEM_1 [%]	n	m	k^*	CEM_{k^*}/CEM_1 [%]
7	15	3	96.8	9	15	2	97.5
	16	3	96.9		16	2	98.8
	17	3	96.2		17	2	98.7
	18	3	96.2		18	2	98.4
	19	3	95.4		19	2	98.4
	20	4	94.5		20	3	97.5
	21	4	93.6		21	3	97.3
8	15	2	98.8	10	15	2	99.9
	16	3	97.5		16	3	99.2
	17	3	97.5		17	2	98.9
	18	3	96.9		18	2	99.0
	19	3	96.2		19	3	98.2
	20	3	95.7		20	3	97.6
	21	4	95.8		21	3	97.6

RELNET), CPU time, total number of memorized networks (MEMNET), the maximum size of $\mathcal{M}$ (MAX-M) and the rate between MEMNET and MAX-M. The result is shown in Table 1 and Figure 5.

In Table 1, Procedure CEM_{20} is identical to Procedure SCEM. The RELNET of Procedure CEM_1 is 1.80% of that of Procedure SCEM. In Figure 5, RELNET and CPU time varies on k quite similarly. The correlation coefficient between RELNET and CPU time is 0.99993. We are sure that it is quite efficient to refer to computed reliabilities in order to reduce RELNET which leads to reducing CPU time. Procedure CEM_3 executes in shortest time and its computational time is 95.7% of that by Procedure CEM_1.

Table 1 shows that MEMNET and MAX-M decrease on the value of k. Without the procedure (1) mentioned in subsection 3.2, MEMNET is equal to MAX-M. Though the average rate between MEMNET and MAX-M is 50.1%, the rate is quite small for smaller value of k. We conclude that the procedure (1) reduces memory space drastically.

Furthermore, we have researched CPU times for more networks where $n = 7, 8, 9, 10$ and $m = 15, \ldots, 21$ and ascertained similar results. k^* is defined as the value which makes CPU time shortest. Table 2 indicates the number of nodes in target networks, the number of edges, k^* and the rate of CPU time CEM_{k^*}/CEM_1. It is ascertained that the value of k^* is smaller for networks with more nodes and less edges, that is, for sparser networks. In this experiments, the average values of CEM_k/CEM_1 for $k = 2, 3, 4$ for all networks are computed, which are 98.2%, 97.7% and 100.0%, respectively.

4.2. Comparison with other algorithms

In order to ascertain the effectiveness of our algorithm, we have compared with other algorithms in numerical experiments. We construct a new procedure by replacing Procedure SFM with the procedure based on BDD method in Procedure SCEM. We call it Procedure BDD. We compare Procedure CEM_1 with SCEM and BDD on running time for many networks where $n = 7, 8, 9, 10$ and $m = 15, \ldots, 21$.

Table 3 shows the results and the efficiency of Procedure CEM_1. The more edges networks have, the more advantage Procedure CEM_1 takes. The average running time of Procedure CEM_1 is 8.6% of that of Procedure SCEM. Even compared with Procedure BDD, it

Table 3: Running Time to Compute Reliability for All Sub-networks

n	m	SCEM (A) [s]	BDD [s]	CEM_1 (B) [s]	A/B [%]	n	m	SCEM (A) [s]	BDD [s]	CEM_1 (B) [s]	A/B [%]
7	15	3.67	2.79	0.44	12.0	9	15	0.91	0.63	0.20	22.0
	16	10.90	6.94	1.05	9.6		16	3.54	2.17	0.56	15.8
	17	32.05	16.97	2.52	7.9		17	12.37	4.93	1.51	12.2
	18	95.60	44.22	5.88	6.2		18	45.86	15.68	3.94	8.6
	19	281.89	140.92	13.53	4.8		19	150.65	52.38	9.53	6.3
	20	827.95	370.48	30.82	3.7		20	500.83	147.37	22.90	4.6
	21	2362.71	894.80	70.51	3.0		21	1468.31	298.09	53.93	3.7
8	15	2.42	1.66	0.33	13.6	10	15	0.66	0.46	0.14	21.2
	16	8.79	6.43	0.87	9.9		16	2.29	1.41	0.39	17.0
	17	27.57	18.18	2.08	7.5		17	8.45	4.30	1.05	12.4
	18	85.10	51.58	4.98	5.9		18	32.67	17.16	2.78	8.5
	19	261.25	145.50	11.82	4.5		19	118.04	49.17	7.30	6.2
	20	784.50	354.17	27.95	3.6		20	433.91	159.84	18.81	4.3
	21	2282.61	959.22	63.60	2.8		21	1402.68	511.31	44.50	3.2

completes in 16.5% CPU time on average. Though Procedure CEM_1 naturally needs exponential computational time, the computational time increases more slowly than the other procedures. We conjecture that the reason why BDD takes more computational time is that the number of edges in the subnetworks is too small for the BDD method to demonstrate its effectiveness.

5. Application of Procedure CEM_k

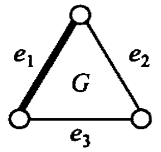
Since Procedure CEM_k is a complete enumeration method for a network design problem, some algorithms can be proposed by extending Procedure CEM_k. This section discusses applications of Procedure CEM_k to other kinds of network design problems.

5.1. Biobjective network design problem

Recently, meta heuristics approaches are applied to multiobjective network design problems with network reliability [14, 16, 20]. Here we aim to propose an exact algorithm for the following biobjective network design problem BNDP:

$$\text{BNDP} \quad \begin{array}{|ll} \text{minimize} & \sum_{i=1}^{m} c_i \cdot x_i \\ \text{maximize} & R(G_{\boldsymbol{x}}) \\ \text{subject to} & x_i \in \{0,1\} \ i = 1, \ldots, m. \end{array}$$

BNDP has two objective functions on construction cost and on network reliability. BNDP usually does not have solutions that optimize both objective functions. In multiobjective optimizations, pareto optimums are searched as the solutions in general. Since Procedure CEM_k estimates cost and reliability for all solutions, it can be extended to an algorithm for BNDP by estimating domination among solutions. We have estimated reliability and cost for all subnetworks of the network shown in Figure 6(a) and obtained five pareto optimums shown in Figure 6(b).



$p_1 = 0.9$, $p_2 = p_3 = 0.8$
$c_1 = 2$, $c_2 = c_3 = 1$

(a)

subnet	source vector	cost	Rel.	Pareto opt.
	[000]	0	0	O
	[001]	1	0	
	[010]	1	0	
	[011]	2	0.64	O
	[100]	2	0	·
	[101]	3	0.72	O
	[110]	3	0.72	O
	[111]	4	0.928	O

(b)

Figure 6: (a): Target Network (b):Pareto Optimums

5.2. Drawing state transition networks

Most algorithms for network design problems find one best solution. But network designers usually need not only the best network layout but also information about construction cost and reliability when they modify the best layout, i.e., when they adopt one more edge or they give up an adopted edge. We propose a diagram named *state transition network*, which is useful to such a demand and indicates transitions among subnetworks. The state transition network for the network in Figure 6(a) is shown in Figure 7 where the value in nodes denotes the number of subnetworks. An arc connected two nodes denotes that the two states represented by the two nodes can be transited to each other when one edge is added or deleted. Procedure CEM_k can be easily extended to a simple algorithm to draw a state transition network.

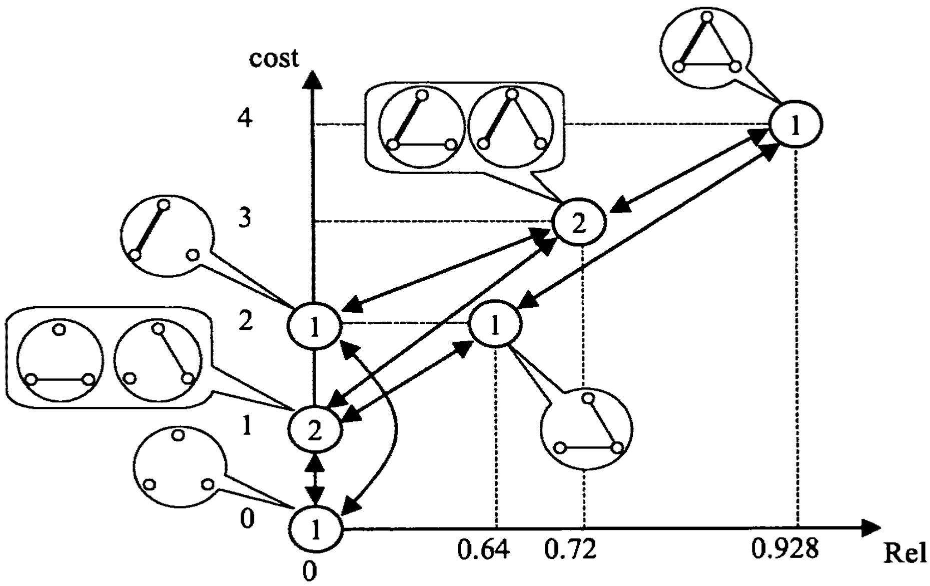


Figure 7: State Transition Network

Oyama and Morohoshi have proposed a quantitative method to measure another kind of network reliability [18]. The state transition network estimates variances for construction cost as well as network reliability.

5.3. Network redesign problem

In redesigning network layout, network designers modify existing networks to improve several criteria. It is also significant to solve the redesign problem efficiently by utilizing imformation obtained from solved problems, such as rescheduling in scheduling problems. Here, we propose a method to estimate reliability for modified subnetworks by using past computational results. We prove the following theorem on recomputing network reliability.

Theorem 6 *Let $G = (V, E)$ be a network and $E' \subseteq E$. Now, we construct a new network G' from G by changing edge probability of all edges $e \in E'$ from p_e to p'_e. Then, the following equation holds:*

$$R(G') = \sum_{F \subseteq E'} \alpha_F R(G - F) \tag{5}$$

where

$$\alpha_F = \prod_{e \in F}(1 - \alpha_e) \prod_{e \in E' - F} \alpha_e, \quad \alpha_e = \frac{p'_e}{p_e}.$$

Proof: We prove the theorem by reduction on $|E'|$. When $|E'| = 1$, let e be the edge whose edge probability is changed. Note that G/e and $G - e$ are isomorphic to G'/e and $G' - e$, respectively. By equation (1), we get

$$\begin{aligned} R(G) &= p_e R(G/e) + (1 - p_e) R(G - e), & (6) \\ R(G') &= p'_e R(G'/e) + (1 - p'_e) R(G' - e) & \\ &= p'_e R(G/e) + (1 - p'_e) R(G - e). & (7) \end{aligned}$$

Combine equation (6) and (7), we get

$$p_e R(G') - p'_e R(G) = p_e(1 - p'_e) R(G - e) - p'_e(1 - p_e) R(G - e).$$

Therefore, it holds

$$R(G') = \alpha_e R(G) + (1 - \alpha_e) R(G - e). \tag{8}$$

Assume equation (5) holds for $|E'| = k$. We construct a new network G'' by changing edge probability of edge $e \in E - E'$. Since equation (8) holds for G'', then we get

$$\begin{aligned} R(G'') &= \alpha_e R(G') + (1 - \alpha_e) R(G' - e) \\ &= \sum_{F \subseteq E'} \alpha_e \alpha_F R(G - F) + \sum_{F \subseteq E'} (1 - \alpha_e) \alpha_F R(G - e - F) \\ &= \sum_{F \subseteq E' \cup \{e\}} \alpha_F R(G - F). \end{aligned}$$

□

Theorem 6 indicates that when a network G' is constructed by changing edge probability for the edges in $E' \subseteq E$ on a network $G = (V, E)$, $R(G')$ can be computed by a linear combination of reliabilities for $2^{|E'|}$ subnetworks of G regardless of $|E|$. For example, we change p_2 to 0.9 in Figure 6(a). Let $G^{(1)}_{\boldsymbol{x}}$ be the new network obtained from $G_{\boldsymbol{x}}$. Then,

$$\begin{aligned} R(G^{(1)}_{[110]}) &= \frac{0.9}{0.8} R(G_{[110]}) + \left(1 - \frac{0.9}{0.8}\right) R(G_{[100]}) = 0.81 \\ R(G^{(1)}_{[111]}) &= \frac{0.9}{0.8} R(G_{[111]}) + \left(1 - \frac{0.9}{0.8}\right) R(G_{[101]}) = 0.954. \end{aligned}$$

Furthermore, we change p_3 to 0.9 and let $G_{\boldsymbol{x}}^{(2)}$ be the newest network obtained from $G_{\boldsymbol{x}}^{(1)}$. Then, $R(G_{[111]}^{(2)})$ is computed as follows:

$$\begin{aligned} & \left(\frac{0.9}{0.8}\right)^2 R(G_{[111]}) + \left(1-\frac{0.9}{0.8}\right)\frac{0.9}{0.8}\{R(G_{[101]})+R(G_{[110]})\} + \left(1-\frac{0.9}{0.8}\right)^2 R(G_{[100]}) \\ = \ & 1.1745 - 0.2025 + 0 = 0.972. \end{aligned}$$

5.4. Sensitivity analysis of network reliability

In this subsection, we estimate the variation of reliability by changing edge probabilities, which means edge importance for network reliability [2, 8, 9, 19]. It is also #P-complete to estimate the edge importance.

We have proved the following theorem which is another expression of equation (5).

Theorem 7 *Let $G=(V,E)$ be a network and $E' \subseteq E$. Now, we construct a new network G' from G by changing edge probability of all edges $e \in E'$ from p_e to $p_e+\delta_e$. Then, the following equation holds:*

$$R(G') = \sum_{F \subseteq E'} \beta_F \mathcal{R}(G,F) \tag{9}$$

where

$$\beta_F = \prod_{e \in F} \frac{\delta_e}{p_e}, \quad \mathcal{R}(G,F) = \sum_{F' \subseteq F} (-1)^{|F'|} R(G-F').$$

Proof: The theorem is proved to substitute p'_e with $p_e+\delta_e$ in equation (5). □

When $E'=\{e\}$, equation (9) can be transformed as

$$R(G') - R(G) = \frac{1}{p_e}\{R(G)-R(G-e)\}\delta_e \equiv s_e\delta_e. \tag{10}$$

Equation (10) indicates the sensitivity of δ_e to $R(G)$. For example, we compute the sensitivities for all edges in Figure 6(a).

$$\begin{aligned} s_{e_1} &= \frac{1}{0.9}(0.928-0.64) = 0.32, \\ s_{e_2} = s_{e_3} &= \frac{1}{0.8}(0.928-0.72) = 0.26. \end{aligned}$$

When we select an edge to increase its edge probability with a fixed degree, e_1 is the best choice to improve network reliability. If p_2 is changed from 0.8 to 0.9, the network reliability is changed from 0.928 to $0.928 + 0.26 \times 0.1 = 0.954$, which is identify to the result computed in the last subsection.

6. Conclusion

We have proposed an algorithm as a complete enumeration method for a network design problem considering reliability and cost. The algorithm can execute in much shorter time than a simple complete enumeration method and BDD method. Note that the proposed algorithm can be applied to similar problems concerned with two-terminal reliability and k-terminal reliability. We also describe applications of the algorithm to some kinds of network design problems. Though the proposed algorithm can be easily extended to algorithms for other problems since it computes reliability and cost for all subnetworks, simply extended algorithms need exponential memory space and much computational time. In the future, we would like to propose more efficient algorithms to need less memory space and computational time by utilizing the characteristics of the problems.

Acknowledgement

The authors grateful to Takeaki Uno for his stimulating discussions and valuable comments. We would like to thank two anonymous referees for their helpful comments and constructive suggestions.

References

[1] K. K. Aggarwal, Y. C. Chopra and J. S. Bajwa: Topological layout of links for optimizing the overall reliability in a computer communication system. *Microelectronics and Reliability*, **22** (1982) 347-351.

[2] M. J. Armstrong: Joint reliability-importance of components. *IEEE Transactions on Reliability*, **44** (1995) 408-412.

[3] M. M. Atiqullah and S. S. Rao: Reliability optimization of communication networks using simulated annealing. *Microelectronics and Reliability*, **33** (1993) 1303-1319.

[4] S. -T. Cheng: Topological optimization of a reliable communication network. *IEEE Transactions on Reliability*, **47** (1998) 225-232.

[5] Y. C. Chopra, B. S. Sohi, R. K. Tiwari and K. K. Aggarwal: Network topology for maximizing the terminal reliability in a computer communication network. *Microelectronics and Reliability*, **24** (1984) 911-913.

[6] C. J. Colbourn: *Combinatorics of Network Reliability* (Oxford University Press, New York, 1987).

[7] B. Dengiz, F. Altiparmak and A. E. Smith: Efficient optimization of all-terminal reliable networks, using an evolutionary approach. *IEEE Transactions on Reliability*, **46** (1997) 18-26.

[8] J. S. Hong and C. H. Lie: Joint reliability-importance of two edges in an undirected network. *IEEE Transactions on Reliability*, **42** (1993) 17-23.

[9] S. J. Hsu and M. C. Yuang: Efficient computation of marginal reliability-importance for reducible$^+$ Networks. *IEEE Transactions on Reliability*, **50** (2001) 98-106.

[10] H. Imai, K. Sekine and K. Imai: Computational investigations of all-terminal network reliability via BDDs. *IEICE Transactions on Fundamentals*, **E82-A(5)** (1999) 714-721.

[11] R. -H. Jan, F. -J. Hwang and S. -T. Chen: Topological optimization of a communication network subject to a reliability constraint. *IEEE Transactions on Reliability*, **42** (1993) 63-70.

[12] S. Kiu and D. F. McAllister: Reliability optimization of computer-communication networks. *IEEE Transactions on Reliability*, **37** (1988) 433-440.

[13] T. Koide, S. Shinmori and H. Ishii: Topological optimization with a network reliability constraint. *Discrete Applied Mathematics*, **115** (2001) 135-149.

[14] A. Kumar, R. M. Pathak and K. P. Gupta: Genetic-algorithm-based reliability optimization for computer network expansion. *IEEE Transactions on Reliability*, **44** (1995) 63-70.

[15] A. Kumar, R. M. Pathak, K. P. Gupta and H. R. Parsaei: A genetic algorithm for distributed system topology design. *Computers and Industrial Engineering*, **28** (1995) 659-670.

[16] B. Liu and K. Iwamura: Topological optimization models for communication network with multiple reliability goals. *Computers and Mathematics with Applications*, **39** (2000) 59-69.

[17] B. Ombuki, M. Nakamura, Z. Nakao and K. Onaga: Evolutionary computation for topological optimization of 3-connected computer networks. *Proceedings of IEEE SMC'99 Conference*, (1999) I659-664

[18] T. Oyama and H. Morohoshi: A quantitative method to measure the stable connectedness of the network structured system using shortest path counting method. *Abstracts of the 2001 Spring National Conference of ORSJ*, (2001) 60-61 (in Japanese).

[19] L. B. Page and J. E. Perry: Reliability polynomials and link importance in networks. *IEEE Transactions on Reliability*, **43** (1994) 51-58.

[20] S. Pierre and A. Elgibaoui: A tabu-search approach for designing computer-network topologies with unreliable components. *IEEE Transactions on Reliability*, **46** (1997) 350-359.

[21] F. -M. Shao and L. -C. Zhao: Topological optimization of computer network expansion with reliability constraint. *Computers and Mathematics with Applications*, **35** (1998) 17-26.

[22] D. R. Shier, *Network Reliability and Algebraic Structures* (Oxford University Press, New York, 1991).

[23] I. M. Soi and K. K. Aggarwal: Reliability indices for topological design of computer communication networks. *IEEE Transactions on Reliability*, **30** (1981) 438-443.

[24] A. N. Venetsanopoulos and I. Singh: Topological optimization of communication networks subject to reliability constraints. *Problems of Control and Information Theory*, **15** (1986) 63-78.

Takeshi Koide
Department of Hospital and Welfare Service
Faculty of Service Industries
University of Marketing and Distribution Sciences
3-1 Gakuen-Nishimachi, Nishi-ku,
Kobe, Hyogo 651-2188, Japan
E-mail: koide@umds.ac.jp