Evaluation of Multi-objective Networks based on Topological Analysis

Nastumi Takahashi†
Department of Industrial and Systems Engineering
Aoyama Gakuin University, Kanagawa, Japan
Tel: (+81) 42-759-6303, Email: takahashi@ise.aoyama.ac.jp

Tomoaki Akiba
Department of Management Information Science
Chiba Institute of Technology, Chiba, Japan
Tel: (+81) 47-478-0342, Email: tomo@akiaki.net

Hisashi Yamamoto, Xiao Xiao
Department of Management Systems Engineering
Tokyo Metropolitan University, Tokyo, Japan
Tel: (+81) 42-585-{8674, 8462}, Email: {yamamoto, xiaoxiao}@tmu.ac.jp

Koji Shingyochi
Department of Career Planning and Information Studies
Jumonji University, Saitama, Japan
Tel: (+81) 48-477-0557, Email: shin@jumonji-u.ac.jp

Abstract. There are many network systems in the world, for example, Internet, electricity network and traffic network. In this study, we consider network design problem with multi-criteria, costs, reliability, processing time and so on. In the general, it is a rare case that a solution (network) makes all of criteria optimal simultaneously, as there exists a trade-off between criteria. Therefore we consider an algorithm for obtaining Pareto solutions (networks). Even if the existing algorithms are effective in calculating one criterion for a network, such algorithms become inefficiency in obtaining Pareto solutions. Especially, it takes much time to obtain Pareto solutions when the number of nodes and edges is large. So, we propose the algorithm for Pareto solutions. Our proposed algorithm selects a part of networks and obtains criteria for their networks only. We researched properties of the solutions, which were near Pareto solutions. By using these properties, we find out the search space which includes many solutions near Pareto solutions. Proposed algorithm is constructed for obtaining Pareto solutions, which obtains criteria for a part of networks (not all networks). The algorithm is evaluated by numerical experiment and is compared with the existing algorithm.

Keywords: Network Design Problem, Multi-objective Optimization, Pareto Solutions

1. INTRODUCTION

There are many network systems in the world, for example, Internet, electricity network and traffic network, etc. These systems forms basis of society, so stabilizing operation of systems and designing systems to operate effectively are important problems. The evaluation of network systems considers various measures, for example, distances, flow, construction or transfer cost and time. Network evaluation have been widely studied. Reliability in particular is classified into several types according to the number of target nodes that must be connected. Many papers have been published with regards to reliability(Lin, 2002; Boesch, 2009).

Making a decision reasonably needs to consider plural measures when we evaluate networks. In this study, we consider all-terminal reliability(Colbourn, 1987) and construction cost for network as important measures for evaluation. All-terminal reliability is one of network reliabilities. However, there exists a trade-off relationship between reliability and construction cost. It is a rare case that a solution (network system) has maximized all-terminal reliability and minimized construction cost simultaneously. Therefore, this problem is multi-objective optimization problem of multi-objective network. In other words, the obtained optimal solution is Pareto solutions.

As one solution for this problem, Akiba et al.(2007) improved Koide's algorithm(2002) which calculated all-terminal reliabilities of all sub-networks, and applied improved algorithm to two-objective network. However, this algorithm also needs to calculate all-terminal reliabilities of all sub-networks. So, their algorithm has such problems that it takes more computing time and larger memory area, as the number of nodes and edges in network become larger. However, only a part of networks becomes Pareto solutions. That is, not all networks are necessarily needed to construct Pareto solutions. For an efficient algorithm, we researched properties that Pareto solutions were likely to satisfy. Based on these network properties, we reduce the number of networks whose reliability and cost must be calculated, and propose a new algorithm for obtaining the proper subset of Pareto solutions efficiently.

2. DEFINITION OF PROBLEM

2.1 Notations

For describing our problem, we define some notations as follows.

n : The number of nodes

m : The number of edges of complete graph with n nodes, that is $m = \dfrac{n(n-1)}{2}$

e_i : The *i*-th edge

E : The set of undirected edges $E = (e_1, e_2, \cdots, e_m)$, where m is the number of edges

x_i : Binary variable where $x_i = 1$, if e_i is included in network, and $x_i = 0$ if not

$\mathbf{x}$: $\mathbf{x} = (x_1, x_2, \cdots, x_m)$ is *m*-dimensional vector whose x_i is binary variable. Note that network is specified by $\mathbf{x}$, so $\mathbf{x}$ denotes the network

X : The set of network $\mathbf{x}$

p_i : Reliability of edge e_i $(i = 1, 2, \cdots, m)$

c_i : Construction cost of e_i $(i = 1, 2, \cdots, m)$

$R(\mathbf{x})$: All-terminal reliability of network $\mathbf{x}$

$C(\mathbf{x})$: Total construction cost of network $\mathbf{x}$

2.2 Definition of Problem

In this study, we consider network design problem with all-terminal reliability and construction cost. For any network $\mathbf{x}$ with n nodes and undirected edges, we suppose the following throughout this paper.

a) p_i and c_i of each e_i are known values.

b) In network $\mathbf{x}$, n nodes are always connected and these all nodes do not fail.

The problem considered in this paper can be expressed as follows.

$R(\mathbf{x}) \rightarrow \max$

$C(\mathbf{x}) \rightarrow \min$

$s.t. \quad \mathbf{x} \in X$

Where $C(\mathbf{x})$ is defined by following function.

$$C(\mathbf{x}) = \sum_{i=1}^{m} c_i x_i \quad (m = \frac{n(n-1)}{2})$$

The way of calculating $C(\mathbf{x})$ is mentioned in section 2.3.

We aim to obtain networks which satisfy above functions, and this problem is defined as network design problem with all-terminal reliability and construction cost. It is a rare case that a network makes all-terminal reliability and construction cost optimal simultaneously. Obtained optimal solution is Pareto solutions.

Next, we define the Pareto solutions considered in this study. For network $\mathbf{x}, \mathbf{x}' \in X$, network $\mathbf{x}$ becomes a strong Pareto solution, when there are no other networks $\mathbf{x}'$ which satisfies one of following 3 conditions among all of X.

$R(\mathbf{x}) < R(\mathbf{x}')$ and $C(\mathbf{x}) > C(\mathbf{x}')$,

$R(\mathbf{x}) = R(\mathbf{x}')$ and $C(\mathbf{x}) > C(\mathbf{x}')$,

$R(\mathbf{x}) < R(\mathbf{x}')$ and $C(\mathbf{x}) = C(\mathbf{x}')$.

Our study obtains strong Pareto solutions. The other networks are inferior solutions.

2.3. Calculation of All-terminal Reliability

Similar to existing algorithm for calculating all-terminal reliability $R(\mathbf{x})$, we introduce recursive algorithm by Koide (2002). Outline of this algorithm is shown below.

This algorithm reduces the number of nodes or edges by two operations. One is deletion, which means removing edge e_i from the network. The other one is contraction, which means removing edge e_i from the network after unifying both nodes connected by edge e_i. Let $\mathbf{x} - x_i$ and $\mathbf{x} / x_i$ be the network $\mathbf{x}$ with edge e_i deleted and the network $\mathbf{x}$ with edge e_i contracted, respectively. Then, all-terminal reliability $R(\mathbf{x})$ can be expressed as the following formula. For $i = 1, 2, \cdots, m$,

$$R(\mathbf{x}) = (1 - p_i) \times R(\mathbf{x} - x_i) + p_i \times R(\mathbf{x} / x_i) \qquad (1)$$

$R(\mathbf{x})$ is calculated by transforming network until the number of edges becomes one. In this paper, networks, whose all nodes are connected with less than $n(n-1)/2$ edges, are called sub-networks. Sub-networks appear in process of calculating all-terminal reliability from spanning tree through complete graph.

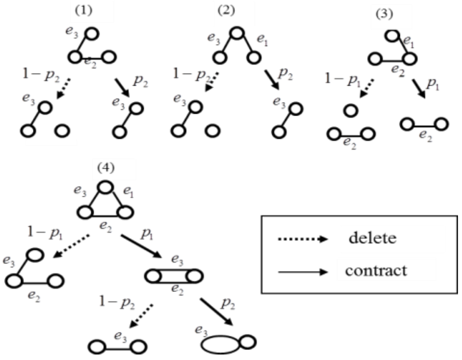

Figure 1: Calculation process for all-terminal reliability.

As an example, network with 3 nodes is constructed. All-terminal reliability is calculated while sub-networks are constructed by the number of edges as shown in Figure 1. All-terminal reliability of sub-network (1), which has $n-1$ edges, is derived from formula (1).

$$R(\mathbf{x})=(1-p_2)*0+p_2*p_3$$

For sub-networks (2) and (3) with $n-1$ edges, all-terminal reliabilities are calculated similarly. This process continues for all networks with edges more than $n-1$. Moreover, Koide (2002) proposed the efficient algorithm that calculated reliabilities were memorized and referred in recursive process. He called it "factmem method". For network (4) in Figure 1, sub-network deleted edge e_2 is the same as sub-network (1) whose all-terminal reliability has been calculated. Let $R(\mathbf{x_1})$ be all-terminal reliability of network (1). All-terminal reliability $R(\mathbf{x})$ of network (4) calculated as follows.

$$R(\mathbf{x})=(1-p_1)*R(\mathbf{x_1})+p_1*\{(1-p_2)*p_3+p_2*1\}$$

Akiba et al.(2007) proposed stopping calculation for obvious network about reliability in factmem method. This advanced algorithm is called "improved factmem method". As regards construction cost $C(\mathbf{x})$, they summed up edge cost c_i, and extended to two-objective network design problem with reliability. However, their algorithm must calculate reliabilities and costs for all sub-networks in calculation process of all-terminal reliability. This process makes computing time increase, as the number of nodes becomes large.

3. OUTLINE OF PROPOSED ALGORITHM

For obtaining Pareto solutions efficiently, considered networks are restricted to close sub-networks to Pareto front in all-terminal reliability and construction cost. This section describes outline of our proposed algorithm. For explanation, the following notations are defined.

For $k=n-1,n,\cdots,m$,

$\mathbf{x}_k$: The sub-network with n nodes connected by k edges, $\mathbf{x}_k=(x_1,x_2,\cdots,x_m)$ and $\sum_{i=1}^{m}x_i=k$

X_k : The set of sub-network $\mathbf{x}_k$

$\mathbf{px}_k$: The sub-network which satisfies definition of Pareto solutions in X_k, $\mathbf{px}_k=(x_1,x_2,\cdots,x_m)$

PX_k : The set of sub-network $\mathbf{px}_k$

Figure 2 shows outline of our proposed algorithm. Proposed algorithm reduces the number of sub-networks whose all-terminal reliability and construction cost must be calculated.

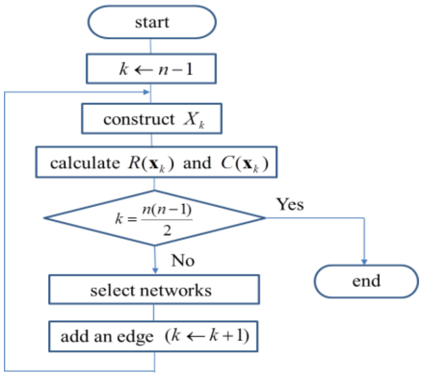

Figure 2: The outline of proposed algorithm.

We construct all sub-networks $\mathbf{x}_k$ of set X_k, where $k=n-1$. Among X_k, a part of sub-networks is selected as the candidate sub-networks for constructing sub-network set X_{k+1}. By adding edge e_i to selected networks $\mathbf{x}_k=(x_1,\cdots,0_i,\cdots x_m)$ $(\in X_k)$ from X_k, elements of X_{k+1} are constructed. For X_{k+1}, a part of X_{k+1} are also selected, and edge e_i is added. This process continues until $k=n(n-1)/2$.

This study consider how to select networks $\mathbf{x}_k \in X_k$ from X_k and add edges e_i to $\mathbf{x}_k$. Following section 4.1 and section 4.2 explain the area for selected networks and criteria for added edge, respectively.

4. RESEARCH IN NETWORK PROPERTIES

Pareto solutions draw the curve in objective space. This curve is named Pareto front. Proposed algorithm reduces search space by properties of sub-networks which compose Pareto front. We researched effective properties in reducing sub-networks which must be calculated. First, these properties are mentioned.

4.1 Networks Composing Pareto Front

Proposed algorithm construct X_{k+1} based on X_k. This study experimented with Pareto solutions $\mathbf{px}_k(\in PX_k)$ and inverse Pareto solutions $^{-}\mathbf{x}_k(\in X_k)$. This experiment researched how to change all-terminal reliability and construct cost when edges are added to each network.

For $^{-}\mathbf{x}, \mathbf{x} \in X$, suppose network $^{-}\mathbf{x}$ is inverse Pareto solution, when there are no other networks $\mathbf{x}$ which satisfies one of following 3 conditions among all of X.

$R(^{-}\mathbf{x}) > R(\mathbf{x})$ and $C(^{-}\mathbf{x}) < C(\mathbf{x})$

$R(^{-}\mathbf{x}) = R(\mathbf{x})$ and $C(^{-}\mathbf{x}) < C(\mathbf{x})$

$R(^{-}\mathbf{x}) > R(\mathbf{x})$ and $C(^{-}\mathbf{x}) = C(\mathbf{x})$

As the example, network SX was generated, which had 5 nodes and 10 edges. Each p_i and c_i of SX was given by random number uniformly between $0.50 \le p_i \le 0.99$ and $50 \le c_i \le 99$, respectively. $\mathbf{px}_4$ and $^{-}\mathbf{x}_4$, which were derived from SX, were added edge. From constructed networks by $\mathbf{px}_4$ and $^{-}\mathbf{x}_4$, we found following three tendencies.

(a) $\mathbf{px}_4$ constructed all $\mathbf{px}_5$ among PX_5, but $^{-}\mathbf{x}_4$ couldn't construct $\mathbf{px}_5$ at all.

(b) For each $\mathbf{px}_4 = (x_1, x_2, \cdots, x_m)(\in PX_4)$, $p_i / c_i (= f_i)$ is calculated, where edge e_i with p_i and c_i satisfies $x_i = 0$. Edges e_i, that $x_i = 0$, were divided into superior edges with high f_i and inferior edges with low f_i. Each edge were added to $\mathbf{px}_4$. When superior edges with high f_i were added to $\mathbf{px}_4$, all $\mathbf{px}_5$ among PX_5 were constructed.

(c) We analyzed contained edges in PX_k with a few k. As the result, edge e_i, which were contained frequently in PX_k, were likely to compose Pareto solutions with more than k edges.

As an example, Figure 3 shows two cases that two kinds of edge are added to $\mathbf{px}_4$ which is based on SX. The arrow A means adding the superior edge. The arrow B means adding the inferior edge.

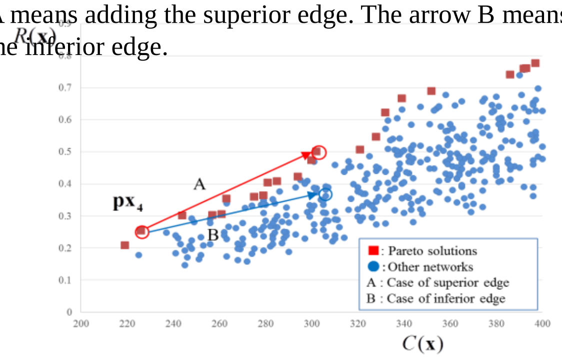

Figure 3: The influence of efficiency of edges for networks.

4.2 Areas for Considered networks

4.2.1 Ranking of Pareto Solutions

From tendency (a) shown in section 4.1, this study derive a following property.

Property 1 : $\mathbf{x}_{k+1}$ constructed by $\mathbf{px}_k$ is likely to be Pareto solutions $\mathbf{px}_{k+1}(\in PX_{k+1})$.

For applying Property 1 to our algorithm, this study applies following ranking procedure.

Definition 1 (Ranking of Pareto solutions). For network set X, let $pareto(X)$ be subset of X which satisfies definition of strong Pareto solutions. For X_k, rank $r(r = 1, 2, \cdots, R)$ of Pareto solutions is defined as following equation.

$$PX_{r,k} \equiv pareto(X_k \setminus \bigcup_{s=1}^{r-1} PX_{s,k}) \qquad (2)$$

In above definition, when rank $r = 1$, $PX_{r,k}$ corresponds to Pareto solutions PX_k of X_k. Left figure in Figure 4-1 illustrates $PX_{1,k}$ which are assigned rank 1. Light figure in Figure 4-1 shows $PX_{2,k}$ which are assigned rank 2. $PX_{2,k}$ denotes set of sub-network such as satisfies definition of Pareto solutions where $PX_{1,k}$ is removed. This process continues until rank R is assigned. By ranking sub-networks, proposed algorithm selects candidate sub-networks.

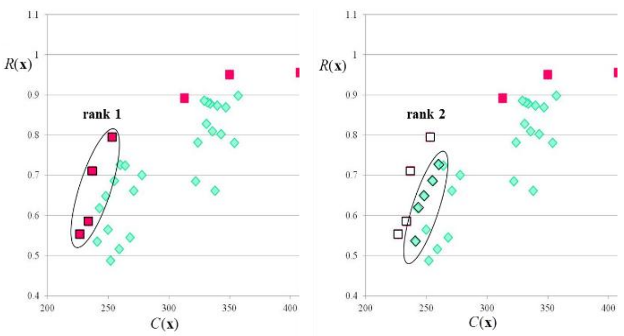

Figure 4-1: Ranking process of Pareto solutions.

4.2.2 Slope of Pareto Solutions

If more ranks of Pareto solutions are considered, we can obtain many Pareto solutions. However equation (2) suggests that processes increase for obtaining many ranks. This increase seems to give loads for calculation, so we propose the other area for considered networks. When Pareto solutions are plotted on two-dimensional space, whose axes are $R(\mathbf{x})$ and $C(\mathbf{x})$, Pareto front is drawn like an exponential curve as shown by ■ in Figure 4-2.

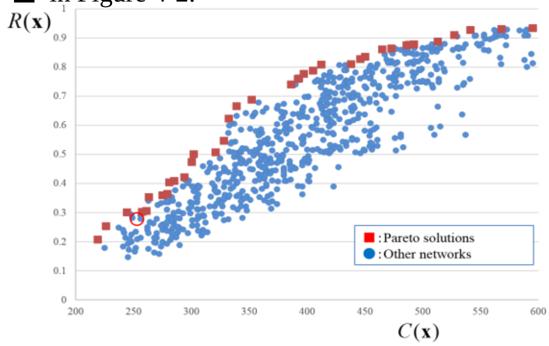

Figure 4-2: The shape of Pareto solutions.

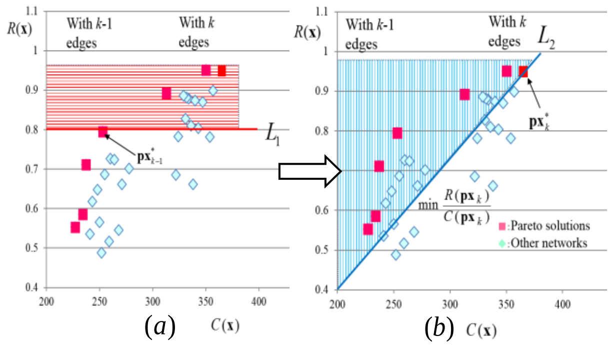

Figure 4-3: Two lines by two properties.

Second area for considered networks is derived from this shape. Search space is restricted to near space of Pareto front by the slope of certain linear function. This study assume following two properties which Figure 4-3 illustrates.

Property 2-1. For $\mathbf{px}^*_{k-1} = \arg\max_{\mathbf{px}_{k-1} \in PX_{k-1}} R(\mathbf{px}_{k-1}) (\in PX_{k-1})$, $\mathbf{x}_k$ which satisfies $R(\mathbf{px}^*_{k-1}) < R(\mathbf{x}_k)$ can construct $\mathbf{px}_{k+1} (\in PX_{k+1})$ by adding an edge.

Property 2-1 means selecting higher network in reliability than $\mathbf{px}^*_{k-1}$ of Figure 4-3(a). This area is upper space of line L_1 drawn on Figure 4-3(a).

Property 2-2. For $\mathbf{px}''_k = \arg\min_{\mathbf{px}_k \in PX_k} \frac{R(\mathbf{px}_k)}{C(\mathbf{px}_k)} (\in PX_k)$, $\mathbf{x}_k$ which satisfies $\frac{R(\mathbf{px}''_k)}{C(\mathbf{px}''_k)} < \frac{R(\mathbf{x}_k)}{C(\mathbf{x}_k)}$ can construct $\mathbf{px}_{k+1} (\in PX_{k+1})$ by adding an edge.

Property 2-2 means selecting networks with large increment of reliability per cost. This area is upper space of line L_2 drawn on Figure 4-3(b).

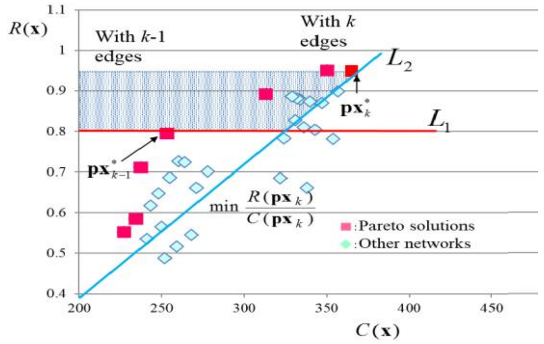

Figure 4-4: The image of restricted space by two straight lines.

Figure 4-4 shows the area satisfying Property 2-1 and Property 2-2. The area of network selection is surrounded by two lines L_1 and L_2.

4.3 Criteria for Adding Edges

4.3.1 Edge Efficiency

Next, we consider adding edge to selected networks $\mathbf{x}_k$. Tendency (b) of section 4.1 reveals Pareto solutions depend on reliability p_i and cost c_i of each edge which are added to networks. Therefore this paper defines following index.

Definition 2 (Edge efficiency) : For e_i, $f_i (= p_i / c_i)$ denotes efficiency of edge.

Fig. 3 shows when edge, which is added to network, takes high f_i, this network is more likely to approximate Pareto front. So, this paper considers edge with high to be efficient edge, and proposes adding efficient edges.

4.3.2 Edge Validity

Tendency (c) shown in section 4.1 appears when edges form networks. Whether a network becomes Pareto solution depends on the way of edge link in addition to the value of reliability and cost of edge. Therefore, for obtained Pareto solutions $\forall \mathbf{px}_k (\in PX_k)$, we sum up x_i of each edge e_i and define the following index.

Definition 3 (Edge validity): For Pareto solutions PX_k, edge validity $vl_k(i)$ is defined as following equation.

$$vl_k(i) \equiv \sum_{\{\mathbf{px}_k \in PX_k\}} x_i$$

4.3.3 Network Topology

As the third edge criteria, we focus on characteristic of network topology. Depending on the way of edge connection, all-terminal reliability takes different values. When all nodes of network are included in cycle structure, its all-terminal reliability is known to be higher than other topologies' all-terminal reliabilities(Jan, 1993). Comparing networks $\mathbf{x}_{k,1}$ and $\mathbf{x}_{k,2}$ in Figure 5, all-terminal reliability of network $\mathbf{x}_{k,1}$ is superior to that of network $\mathbf{x}_{k,2}$. Therefore, characteristic of network cycle is useful in obtaining dominant solutions.

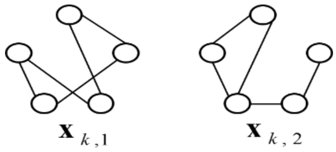

Figure 5: The relation of Network Topology and Reliability.

5. PROPOSED ALGORITHM

Applying the above-mentioned criteria of edge and areas of considered networks to basic algorithm, this study constructs algorithms. Table 1 shows algorithms which are different in combination of criteria of edge and areas of considered networks. Details about each algorithm explained in Section 5.1 and after. About the way of combining area of selected networks, if rank of Pareto solutions is used together with slope of Pareto solutions, search space narrows too much and increase of search processes applies load to calculation. So, one of areas for selected networks is included in algorithms. As edge criteria, considering cycle topology, one of edge efficiency and validity is included.

Table 1: Combination of ideas in evaluated algorithms

	Areas of networks		Criteria of edge	
	Rank	Slope	Efficiency	Validity
AR_E	used		used	
AR_V	used			used
AS_E		used	used	
AS_V		used		used

5.1 Criteria for Adding Edges

In the following explanation, we describe AR_E and AR_V of Table 1, which apply ranking of Pareto solution. In adding edges, AR_E and AR_V use edge efficiency f_i and edge validity $vl_v(i)$, respectively.

For $\mathbf{x}=(x_1,\cdots,0_i,\cdots,x_m)$, let $\mathbf{x}+(1)_i$ be the sub-network with $(x_1,\cdots,1_i,\cdots x_m)$. ec and R are given as parameter. ec denotes the number of added edge to one selected network $\mathbf{x}_k$, and R denotes the number of ranks in which networks are selected. ec means constructing $\{\mathbf{x}_k+(1)_{i(1)},\mathbf{x}_k+(1)_{i(2)},\cdots,\mathbf{x}_k+(1)_{i(ec)}\}\in X_{k+1}$.

Algorithm AR_E

STEP 1：Set the considered number of rank R and the added number of edge ec. $GP_{r,k}\leftarrow\varphi(r=1,2,\cdots,R)$ and $k\leftarrow n$-1.

STEP 2：If $k=n-1$, construct the set of networks X_k. Else go to **STEP 3**.

STEP 3：For $^{\forall}\mathbf{x}_k\in X_k$, calculate $R(\mathbf{x})$ by improvement factmem method and give $C(\mathbf{x})$ the sum of edge cost.

STEP 4：If $k=\frac{1}{2}n(n-1)$, obtain $pareto(\bigcup_{i=n-1}^{k}X_i)$ and end of algorithm. Else, for $r=1,2,\cdots,R$, $PX_{r,k}\leftarrow pareto(X_k\setminus\bigcup_{s=1}^{r-1}PX_{s,k})$.

STEP 5

STEP 5-1：If $k=n-1$, for $i=1,2,\cdots,m$, calculate $f_i=p_i/c_i$. Set $l\leftarrow ec$.

STEP 5-2 : select network $\mathbf{px}_k\,(\in\bigcup_{r=1}^{R}PX_{r,k})$.

STEP 5-3：Select $e_i\,(\in E\setminus E')$ which satisfies $\max f_i$ and $x_i=0$. Construct $\mathbf{px}_k+(1)_i$. Set $X_{k+1}\leftarrow X_{k+1}\bigcup\{\mathbf{px}_k+(1)_i\}$, $E'\leftarrow E'\bigcup\{e_i\}$ and $l\leftarrow l-1$.

STEP 5-4：If $l=0$, go to STEP 5-5. Else go to STEP 5-3.

STEP 5-5：If select all $\mathbf{px}_k\,(\in\bigcup_{r=1}^{R}PX_{r,k})$ in STEP 5-2 , go to STEP 5-6. Else $l\leftarrow ec$ and go to STEP 5-2.

STEP 5-6： If $k=n-1$, construct sub-networks CX with cycle topology independently from f_i, and $X_{k+1}\leftarrow X_{k+1}\bigcup CX$.

STEP 5-7：$k\leftarrow k+1$ and go to **STEP 2**.

Next, let v be the number of edges for calculating edge validity. When the number of edges $k=v$, validity $vl_v(i)$ is generated by PX_k. If $k\le v$, edge efficiency is used. Algorithm AR_v applies edge validity $vl_v(i)$ when the number of edge $k>v$.

Algorithm AR_v

This algorithm substitutes following **STEP 5** for **STEP 5** of algorithm AR_E.

STEP 5 :

STEP 5-1：If $k=n-1$, For $i=1,2,\cdots,m$, $vl_v(i)\leftarrow f_i$. Else if $k=v+1$, calculate $vl_v(i)=\sum_{\{\forall\mathbf{px}_{k-1}\in PX_{k-1}\}}x_i$ and $vl_v(i)\leftarrow vl_v(i)+f_i$.

STEP 5-2：Select $\mathbf{px}_k=(V,E')(\in\bigcup_{r=1}^{R}PX_{r,k})$. Set $l\leftarrow ec$.

STEP 5-3：Select $e_i\,(\in E\setminus E')$ which satisfies $\max vl_v(i)$, and $x_i = 0$. Construct $\mathbf{px}_k + (1)_i$. Set $X_{k+1} \leftarrow X_{k+1} \cup \{\mathbf{px}_k + (1)_i\}$, $E' \leftarrow E' \cup \{e_i\}$ and $l \leftarrow l - 1$.

STEP 5-4：If $l = 0$, go to STEP 5-5. Else go to STEP 5-3.

STEP 5-5：If select all $\mathbf{px}_k \ (\in \bigcup_{r=1}^{R} PX_{r,k})$ in STEP 5-2, go to STEP 5-6. Else $l \leftarrow ec$ and go to STEP 5-2.

5.2 Slope of Pareto Solutions

This section describes algorithms which apply slope of Pareto solutions as the range for considered networks.

Algorithm AS$_E$ and AS$_V$

These algorithms substitute following **STEP 4** for **STEP 4** of AR$_E$ and AV$_E$, respectively. In following **STEP 4**, $PX_{2,k}$ means the set of selected networks by using slope of Pareto solutions.

STEP 4:

STEP 4-1 : If $k = \frac{1}{2}n(n-1)$, calculate $pareto(\bigcup_{i=n-1}^{k} X_i)$ and end algorithm. Else $PX_{1,k} \leftarrow pareto(X_k)$ and go to STEP 4-2.

STEP 4-2 : For $\mathbf{px}_{k-1}^{*} = \arg\max_{\mathbf{px}_{k-1} \in PX_{k-1}} R(\mathbf{px}_{k-1})$, if $^{\forall}\mathbf{x}_k \in X_k$ satisfies $R(\mathbf{px}_{k-1}^{*}) < R(\mathbf{x}_k)$, $PX_{2,k} \leftarrow PX_{2,k} \cup \{\mathbf{x}_k\}$.

STEP 4-3 : For $\mathbf{px}_k^{''} = \arg\min_{\mathbf{px}_k \in PX_k} \frac{R(\mathbf{px}_k)}{C(\mathbf{px}_k)}$, if $^{\forall}\mathbf{x}_k \in PX_{2,k}$ satisfies $\frac{R(\mathbf{px}_k'')}{C(\mathbf{px}_k'')} > \frac{R(\mathbf{x}_k)}{C(\mathbf{x}_k)}$, $PX_{2,k} \leftarrow PX_{2,k} \setminus \{\mathbf{x}_k\}$.

STEP 4-4：If select all $\mathbf{x}_k \in (X_k \setminus PX_{1,k})$ in STEP 4-2, go to **STEP 5**. Else go to STEP 4-2.

6. NUMERICAL EXPERIMENT

Reduction of considered networks influences obtaining Pareto solutions. This process tends to bring us the proper subset of Pareto solutions. Moreover, if the number of ranks R and the added number of edges ec increase, $|X_k|$ also increases. These increases influence the accuracy of search and computing time. Therefore we conducted numerical experiments to compare influence of parameters.

Experiments were executed using a PC with CPU Intel Core i5-3330 (3.0 GHz), Memory 8.0GB and Microsoft Windows 8.1. Used program was written in C language and compiled by Visual Studio 2010. We experimented by using a network with 6 nodes and 15 edges. Cost c_i and reliability p_i of each edge were prepared in a following pattern. p_i was given by random number uniformly between 0.50 and 0.99. c_i was set for $c_i = 100 * p_i + \alpha_i$, where α was given by random number uniformly between 5 and 10.

6.1 Obtained Rate

We compared with each algorithm in obtained rate and computing time. Obtained rate means the percentage of obtained Pareto solutions by our algorithm to real Pareto solutions.

Figure 6 shows comparison result of AR$_V$ and AS$_V$ in obtained rate and computing time. AR$_V$ and AS$_V$ is based on edge validity $vl_7(i)$. $vl_7(i)$ means edge validity is generated by Pareto solutions with 7 edges PX_7. In Figure. 6, the added number of edge ec varies from 1 to 10. AR(R)$_V$ $(R = 1,2,3)$ denotes the number of considered ranks is R. For example, AR(2)$_V$ means the number of considered ranks is 2 and edges are added by edge validity $vl_7(i)$. AS$_V$ denotes algorithms with slope of Pareto solutions. As shown in Figure 6, with obtained rate increasing, the more computing time is taken. Furthermore, within our experiment, it is inefficient that we make the number of ranks increase as increment of computing time is large. Therefore, ASv is effective. There are similar trend when we used other edge validity $vl_v(i)$ with $v \neq 7$.

Next Figure 7 shows comparison result with edge validity and edge efficiency as criteria for added edges. The value of edge validity is $vl_7(i)$ which is generated by PX_7. Vertical axis of Figure 7 means the differences between algorithm AR$_V$ and AS$_V$, and between algorithm AR$_E$ and AS$_E$ in obtained rate. Horizontal axis means the number of added edges ec. As the result, algorithm AR$_V$ and AS$_V$ increase obtained rates. For $vl_v(i)$, this tendency especially appeared in case that $7 \le v \le 9$.

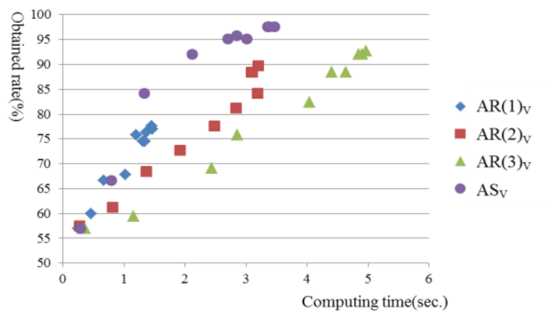

Figure 6: The comparison of AR$_V$ and AS$_V$ in obtained rate.

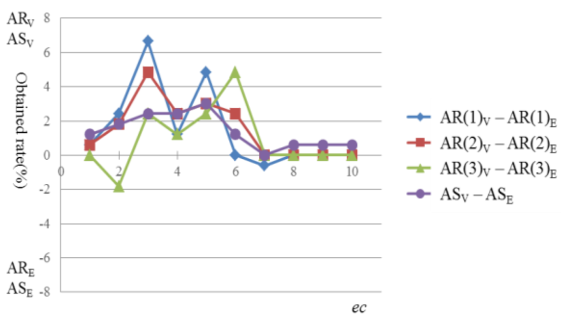

Figure 7: The difference between AR and AS in obtained rate.

6.2 Error Rate

Next we evaluate error rate. Our algorithms output some non-Pareto solutions as optimum solutions. Error rate means the percentage of these non-Pareto solutions by our algorithm to real Pareto solutions.

Figure 8 and Figure 9 compare error rates when the number of added edges varied from 1 to 10 in each algorithm. In Figure 8, comparing algorithm AR_V and AS_V, AS_V tends to decrease error rate. Vertical axis of Figure 9 means the differences between algorithm AR_V and AS_V, and algorithm AR_E and AS_E in error rate. Figure 9 suggests when AR_V increases the number of considered ranks, edge validity was ineffective in error rate. This tendency was also shown in case of $vl_v(i)$ other than $vl_7(i)$.

As the result of evaluation by obtained rate and error rate, algorithm AS_V could obtain the proper subset of Pareto solutions more accurately and efficiently among algorithms which we constructed.

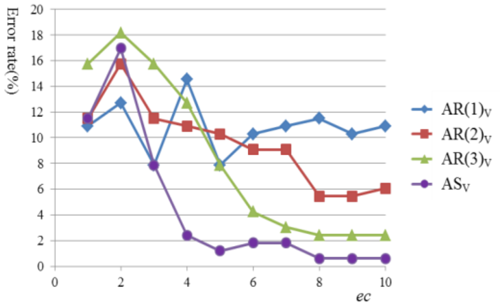

Figure 8: The comparison of ARV and ASV in error rate.

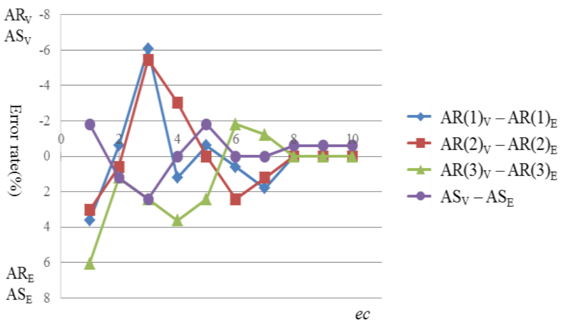

Figure 9: The difference between AR and AS in error rate.

7. CONCLUSION

In our study, we focused on two-objective network design problem with all-terminal reliability and construction cost. Our study introduced criteria for added edges and areas of considered networks. Combining these properties, we proposed the efficient algorithm which restricted the number of calculated networks. Comparing the criteria for added edges, numerical experiment suggested edge validity was superior to edge efficiency in obtained rate. In error rate, superior criterion varied by each algorithm. As area of considered networks, computing time and error rate suggest slope of Pareto solutions was effective. In our numerical experiments, algorithm AS_V could obtain the proper subset of Pareto solutions more accurately and efficiently. Our proposed algorithms can be extended easily to search for weak Pareto solutions. As the future works, we need to adapt these network.

REFERENCES

Lin,Y.K. (2002) Study om the system capacity for a multicommodity stochastic-flow network with node failure. *Reliability Engineering and System Safety*, **78**, 57-62.

Boesch, F.T., Satyanarayana, A., and Suffel, C. L. (2009) A Survey of Some Network Reliability Analysis and Synthesis Results. *Networks*, **54**(2), 678-682.

Colbourn, C. (1987) The Combinatorics of Network Reliability, Oxford University Press, NY.

Akiba,T., Asukagawa,Y., Yamamoto,H., and Nagatsuka, H. (2007) An efficient algorithm for Pareto-solutions of a net-work with the all-terminal reliability and construction cost. *Proc. of International Conference on Mathematical Methods in Reliability 2007*, CD-ROM.

Koide, T., Shinmori, S. and Ishii, H. (2002) An Efficient Complete Enumeration Method for Network Design Problems and Its Applications. *Journal of the Operations Research Society of Japan*, **45**(3), 299-316.

Jan, R. H. (1993) Design of Reliability Network. *Computers & Operations Research*, **20**, 25-34.