

# A Spanning Tree-Based Genetic Algorithm for Bicriteria Topological Network Design

Mitsuo GEN, Kenichi IDA & Jongryul KIM  
Department of Industrial and Information Systems Engineering  
Graduate School of Engineering  
Ashikaga Institute of Technology  
Ashikaga, 326-8558, Japan  
e-mail: {gen ida kjr}@genlab.ashitech.ac.jp

**Abstract**— The optimization problems for the communication network have been taken attentions by many related researchers, such as network designers, network analysts, network administrators. And the use of computer communication networks has been rapidly increasing in order to share expensive hardware/software resources and provide access to main system from distant locations. These network problems have many applications in telecommunications, computer networking, and the related domains in electric, gas, and sewer networks. Especially, in the computer networking, LANs are commonly used as the communication infrastructure that meets the demands of the users in local environment. These networks typically consist of several LAN segments connected together via bridges. The use of these transparent bridges requires *loop-free* paths between LAN segments. Therefore, only spanning tree topologies can be used as active LAN configurations.

Recently, genetic algorithms have got a great advancement in related research fields, such as network optimization problem, combinatorial optimization, multiobjective optimization, and so on. Also genetic algorithm have received a great deal of attention about its ability as optimization techniques for many real-world problem.

In this paper, we attempt to the topology design problem in LAN with bicriteria which is minimizing the cost and average message delay using spanning tree-based genetic algorithm, because the interconnection between the network service centers must yield spanning tree configurations. Finally, we get some experiments in order to certify the quality of the networks designs obtained by using spanning tree-based genetic algorithm.

**Keywords**— Network Design, Message Delay, Spanning Tree-Based Genetic Algorithm, Local Area Network(LAN)

## I. INTRODUCTION

The optimization problems for the communication network have been taken attentions by many related researchers, such as network designers, network analysts, network administrators according to the scale of communication network. The use of computer communication networks has also been rapidly increasing in order to share expensive hardware/software resources and provide access to main system from distant locations. These network problems have many applications in telecommunications, computer networking, and the related domains in electric, gas, and sewer networks. Especially, in the computer networking, LANs are commonly used as the communication infrastructure that meets the demands of the users in local environment. These computer networks typically consist of several LAN segments connected together via bridges. The use of these transparent bridges requires *loop-free* paths between LAN segments[10]. Therefore, only spanning tree topologies can be used as active LANs

configurations. Also, as designing the topology of these computer network systems, one of an important step is to find the best layout of components to optimize the performance criteria, such as cost, message delay, traffic, reliability, and so on. The performance criteria of these systems are important and are largely determined by network topology. We can also address that the network topology design problem includes two main issues: clustering and routing. The clustering problem consists of the two issues, *i.e.*, how many segments(clusters) should the LAN be divided into and how to allocate the users(stations) to the LAN segment(cluster). The routing problem is defined as the determination of segments interconnection spanning tree topology.

Genetic algorithm(GA) has received a great deal of attention regarding their potential as optimization techniques for network optimization problems and is often used to solve many real-world problems[1][2]. And there have been many investigations of GA for multicriteria problems. GA approach has been reported in many applications with multiobjective, such as the transportation problems[9], the minimum spanning tree problems[8], the production process planning problems[7], and so on. The determination of the optimal LAN topology is a very complicated combinatorial optimization problem and can be classified as *NP-hard* optimization problem. Also, when the topology design problem has two objectives, the topology design of LAN is more difficult. Therefore, a heuristic algorithm that is based on a genetic algorithm gets attentions. There were several papers proposed with the methods to solve the network topology design problem using genetic algorithm. Kumar *et al*[14][15] developed a genetic algorithm considering diameter, distance, and reliability to design and expand computer networks. Walters and Smith[16] proposed an evolutionary algorithm for the optimal layout design of networks with tree structure. Deeter and Smith[17] presented a genetic algorithm-based approach to design of networks considering all-terminal reliability with alternative link options, allowing edges to be chosen from different components with different costs and reliabilities. Dengiz, Altiparmak, and Smith[12][13] focused on large backbone communication network design considering the all-terminal network reliability and used a GA, but appreciably customizes it to the all-terminal design problem to give an effective and efficient optimization methodology. Elbaum and Sidi[10] used genetic algorithms based on Huffman tree to solve the topological design of LANs.

In this paper, we attempt to the network topology design problem with bicriteria which is minimizing the cost and the message delay using spanning tree-based genetic algorithm. Because of the feature that only spanning tree topology can be used as active LAN configurations, we employ the spanning tree-based genetic algorithm. We also employ the Prüfer number encoding to represent the spanning tree, which is capable of equally and uniquely representing all possible trees[8]. Using Prüfer number encoding, the memory only requires  $n+m-2$  on the problem with  $n$  service centers and  $m$  users. The objectives of this paper is to minimize the cost needed to design the networks system and also to minimize the average networks message delay.

## II. THE BICRITERIA LAN TOPOLOGY DESIGN

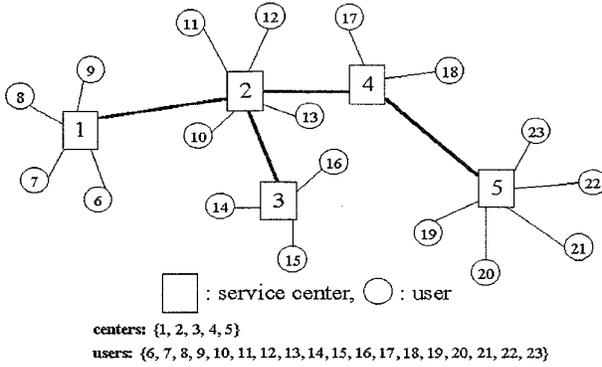


Fig. 1. The sample LAN architecture

### A. Formulation of the problem

We define the following notations in order to formulate the bicriteria LAN topology design considering the message delay and connected cost as the criteria:

#### Notation

|                |  |
|----------------|--|
| $n$            | the number of service center   |
| $m$            | the number of user to be clustered   |
| $U = [u_{ij}]$ | $m \times m$ user traffic matrix where $u_{ij}$ represents the traffic from user $i$ to user $j$   |
| $T = [t_{ij}]$ | $n \times n$ service center traffic matrix where $t_{ij}$ represents the traffic forwarded from users in center $i$ to users in center $j$ |
| $X_1$          | $n \times n$ service center topology matrix where $x_{1ij} = 1$ , if the centers $i$ and $j$ are connected, $x_{1ij} = 0$ , otherwise      |
| $X_2$          | $n \times m$ clustering matrix where $x_{2ij} = 1$ , if user $j$ belongs to center $i$ , $x_{2ij} = 0$ , otherwise                         |
| $X$            | $n \times (n+m)$ spanning tree matrix ( $[X_1 \ X_2]$ )  |

$$a_{ij}^k(\mathbf{X}) = \begin{cases} 1, & \text{if traffic from center } i \text{ to center } j \\ & \text{through center } k \text{ exists} \\ 0, & \text{otherwise} \end{cases}$$

$$b_{ij}^{(k,l)}(\mathbf{X}) = \begin{cases} 1, & \text{if traffic from center } i \text{ to center } j \\ & \text{passes through existing link} \\ & \text{connecting centers } k \text{ and } l \text{ exists} \\ 0, & \text{otherwise} \end{cases}$$

$C_k$  the traffic capacity of center  $k$

$\beta_{ij}$  the delay per bit due to the link between centers  $i$  and  $j$

$g_i$  the maximum number which is capable of connecting to center  $i$

$w_{1ij}$  the weight of the link between centers  $i$  and  $j$

$w_{2ij}$  the weight of the link between center  $i$  and user  $j$ .

And M/M/1 model[4] is used in this paper to describe a single cluster(LAN segment) behavior. Then we can formulate the bicriteria LAN topology design problem as the following nonlinear 0-1 programming model:

$$\min \quad c_1(\mathbf{X}) =$$

$$\frac{1}{\Gamma} \left[ \sum_i^n \frac{l_i(\mathbf{X})}{C_i - l_i(\mathbf{X})} + \sum_{i=1}^n \sum_{j=1}^n \beta_{ij} \cdot f_{ij}(\mathbf{X}) \right] \quad (1)$$

$$\min \quad c_2(\mathbf{X}) =$$

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^n w_{1ij} \cdot x_{1ij} + \sum_{i=1}^n \sum_{j=1}^m w_{2ij} \cdot x_{2ij} \quad (2)$$

$$\text{s. t.} \quad \sum_{j=1}^m x_{2ij} < g_i, \quad i = 1, 2, \dots, n \quad (3)$$

$$\sum_{i=1}^n x_{2ij} = 1, \quad j = 1, 2, \dots, m \quad (4)$$

$$l_i(\mathbf{X}) < C_i, \quad i = 1, 2, \dots, m \quad (5)$$

where the total offered traffic  $\Gamma$  is represented as follows:

$$\Gamma = \sum_{i=1}^n \sum_{j=1}^n t_{ij}.$$

The total traffic at center  $k$ ,  $l_k(\mathbf{X})$ , is represented as follows:

$$l_k(\mathbf{X}) = \sum_{i=1}^n \sum_{j=1}^n t_{ij} \cdot a_{ij}^k(\mathbf{X}), \quad k = 1, \dots, n.$$

The total traffic through link  $(k, l)$ ,  $f_{kl}(\mathbf{X})$ , is represented as follows:

$$f_{kl}(\mathbf{X}) = \sum_{i=1}^n \sum_{j=1}^n t_{ij} \cdot b_{ij}^{(k,l)}(\mathbf{X}),$$

$$k = 1, \dots, n \quad l = 1, \dots, n.$$

We can also represent the service center traffic matrix as follows:  $T = X_2^T U X_2$ .

## III. SPANNING TREE-BASED GENETIC ALGORITHM

### A. Representations and Initialization

The genetic representation is a kind of data structure which represents the candidate solutions of the problem

in coding space. Usually different problems have different data structures or genetic representations. Here we employ the spanning tree representation using Prüfer number in order to represent active LAN configurations.

One of the classical theorems in graphical enumeration is Cayley's theorem that there are  $k^{(k-2)}$  distinct labeled trees on a complete graph with  $k$  nodes. Prüfer provided a constructive proof of Cayley's theorem by establishing an one-to-one correspondence between such trees and the set of all strings of  $k - 2$  digits[8]. This means that we can use only  $k - 2$  digits permutation to uniquely represent a tree where each digit is an integer between 1 and  $k$  inclusive. This permutation is usually known as the *Prüfer number*. For any tree there are always at least two leaf nodes[18]. Based on this observation we can easily construct an encoding as follows:

**Procedure: Encoding**

- Step 1:* Let node  $i$  be the smallest labeled leaf node in a labeled tree  $T$ .
- Step 2:* Let  $j$  be the first digit in the encoding as the node  $j$  incident to node  $i$  is uniquely determined. Here we build the encoding by appending digits to the right, and thus the encoding is built and read from left to right.
- Step 3:* Remove node  $i$  and the link from  $i$  to  $j$ ; thus we have a tree with  $k - 1$  nodes.
- Step 4:* Repeat the above steps until one link is left. We produce a Prüfer number or an encoding with  $k - 2$  digits between 1 and  $k$  inclusive.

It is also possible to generate a unique tree from a Prüfer number via the following procedure:

**Procedure: Decoding**

- Step 1:* Let  $P$  be the original Prüfer number and let  $\bar{P}$  be the set of all nodes not included in  $P$ , which designates as eligible nodes for consideration in building a tree.
- Step 2:* Let  $i$  be the eligible node with the smallest label. Let  $j$  be the leftmost digit of  $P$ . Add the edge from  $i$  to  $j$  into the tree. Remove  $i$  from  $\bar{P}$  and  $j$  from  $P$ . If  $j$  does not occur anywhere in  $P$ , put it into  $\bar{P}$ . Repeat the process until no digits are left in  $P$ .
- Step 3:* If no digits remain in  $P$ , there are exactly two nodes,  $r$  and  $s$ , still eligible for consideration. Add link from  $r$  to  $s$  into the tree and form a tree with  $k - 1$  links.

An example is given to illustrate this kind of encoding. The Prüfer number(1 1 2 2) correspondings to a spanning tree on a five node complete graph represented in Fig. 2. The construction of the Prüfer number is described as follows: Locate the leaf node having the smallest label. In this case, it is node 3. Since node 1 is adjacent to node 3 in the tree, assign 1 to the first digit in the Prüfer number, and then remove node 3 and the link (3, 1). Repeat the process on the subtree until link (2, 6) is left and the Prüfer number of this tree with three digits is finally produced.

On the contrary, for the Prüfer number  $P = (1\ 1\ 2\ 2)$ , the nodes 3, 4, 5, and 6 are eligible and  $\bar{P} = \{3, 4, 5, 6\}$ . Node 3 is the eligible node with the smallest label. node 1 is the leftmost digit in  $P$ . Add link (3, 1) to the tree, re-

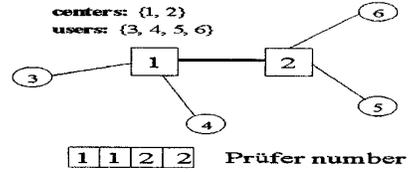


Fig. 2. A tree and its Prüfer number

move node 3 from  $\bar{P}$  for further consideration, and remove the leftmost digit 1 of  $P$  leaving  $P = (1\ 2\ 2)$ . Node 4 is now the eligible element with the smallest label and the second node 1 is the leftmost digit in remaining  $P$ . Then add link (4, 1) to the tree, remove node 4 from  $\bar{P}$  for further consideration, and remove the digit 1 from  $P$  leaving  $P = (2\ 2)$ . Because node 1 is now no longer in the remaining  $P$ , it becomes eligible and is put into  $\bar{P} = \{1, 5, 6\}$ . Then, node 1 is the eligible node with the smallest label, add link (1, 2) to the tree, remove the leftmost digit 2 of  $P$ , and remove node 1 from  $\bar{P}$ . Now  $P = (2)$  and only nodes 5 and 6 are eligible. Add edge(5, 2) to the tree, remove the last digit of  $P$ , and designate node 5 is not eligible and remove it from  $\bar{P}$ . Because node 2 is now no longer in the remaining  $P$  and is added to  $\bar{P}$ .  $P$  is now empty and only nodes 2 and 6 are eligible. Thus add link (2, 6) to the tree and stop. The tree Fig. 2 is formed.

Prüfer number encoding is not only capable of equally and uniquely representing all possible spanning tree, but also explicitly contains the information of node degree that any node with degree  $d$  will appear exactly  $d - 1$  times in the encoding, *i.e.*, when a node appear  $d$  times in Prüfer number, the node exactly have  $d + 1$  connections with other node.

Prüfer number is more suitable for encoding a spanning tree, especially in some research fields, such as transportation problems, minimum spanning problems, and so on. Also, the verification for the excellence of Prüfer number is addressed by Zhou and Gen[8].

From Fig. 2, we can see that all digits of the Prüfer number are the figures of the centers displayed by box. So, the initialization of a chromosome(*i.e.*, a Prüfer number) is performed from that randomly generated  $n + m - 2$  digits in range  $[1, n]$ . Therefore in a chromosome, the center is represented but the user does not appear in a chromosome and only is used as the elements of the set  $\bar{P}$ . Here suppose the centers are represented by the number from 1 to  $n$  and the users are represented by the number from  $n + 1$  to  $n + m$ .

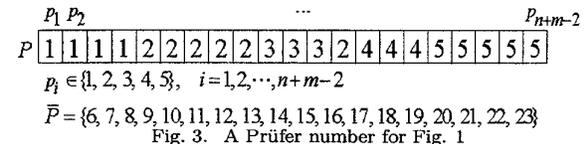


Fig. 3. A Prüfer number for Fig. 1

The representation chromosome for LAN topology shown in Fig. 1 can be represented by Fig. 3 based on encoding procedure.

## B. Evaluation

The single objective LAN topology problem can easily be manipulated by calculating the fitness value of each chromosome according to the objective function. But with bicriteria LAN topology design problem, we can only calculate each objective value and can not simply evaluate its fitness value when the objective functions conflict with each other in practice. In other words, we can not obtain the absolute optimal solution but can only get the Pareto optimal solutions. Here, the Pareto optimal solutions concept is defined as follows:

**Definition 1:** Given a set of feasible solution  $S$ , a solution  $(\mathbf{X}' \in S)$  is said a Pareto optimal solution (or nondominated solution) for the bicriteria LAN topology design problem if and only if there does not exist another solution  $(\mathbf{X} \in S)$  satisfying the following conditions:

$$\begin{aligned} c_i(\mathbf{X}) &\leq c_i(\mathbf{X}'), \text{ for all } i \in \{1, 2\} \\ c_j(\mathbf{X}) &< c_j(\mathbf{X}'), \text{ at least one } j \in \{1, 2\} \end{aligned}$$

In other words, a feasible solution  $(\mathbf{X}')$  is called Pareto optimal if there no exist no feasible solution  $\mathbf{X}$  which would decrease some objective function without causing a simultaneous increase in at least one other objective function.

As to the fitness function for survival, we employ the weighted sums method to construct the fitness function. Then we use the following evaluation for combining the bicriteria objective functions into one overall fitness function and evaluating each chromosome.

*Step 1:* Convert chromosomes represented by Prüfer number to spanning matrix  $\mathbf{X}$ .

*Step 2:* Calculate the objective values  $(c_i(\mathbf{X}), i = 1, 2)$ .

*Step 3:* Choose the solution points which contain the minimum  $c_i^{\min}$  (or the maximum  $c_i^{\max}$ ) corresponding to each objective function value, and then compare with the stored solution points at the previous generation and select the best points to save again.

$$c_i^{\min(t)} = \min_k \{c_i^{\min(t-1)}, c_i^{(t)}(\mathbf{X}_k)\}$$

$$c_i^{\max(t)} = \max_k \{c_i^{\max(t-1)}, c_i^{(t)}(\mathbf{X}_k)\}$$

$$i = 1, 2 \quad k = 1, 2, \dots, chr\_size$$

where  $c_i^{\min(t)}$ ,  $c_i^{\max(t)}$  is the minimum value and the maximum value of  $i$ -th objective function at generation  $t$  respectively,  $c_i^{(t)}$  is the  $i$ -th objective function value of  $k$ -th chromosome at generation  $t$ , and  $chr\_size$  is equal to the  $pop\_size$  plus the offsprings generated after genetic operations.

*Step 4:* Solve the following equation to get weights for the fitness function:

$$\alpha_i = c_i^{\max(t)} - c_i^{\min(t)}, \quad i = 1, 2$$

$$\lambda_i = \frac{\alpha_i}{\sum_{i=1}^2 \alpha_i}, \quad i = 1, 2$$

*Step 5:* Calculate the fitness function value for each chromosome as follows:

$$eval(\mathbf{X}_k) = \sum_{i=1}^2 \lambda_i c_i(\mathbf{X}_k), \quad k = 1, 2, \dots, chr\_size$$

*Step 6:* Convert the evaluation value to fitness as following equation:

$$F(\mathbf{X}_k) = \frac{1}{eval(\mathbf{X}_k)}, \quad k = 1, 2, \dots, chr\_size$$

where  $F(\mathbf{X}_k)$  means a fitness value for the  $k$ -th chromosome.

As to the bicriteria LAN topology design problem, the mechanism of evaluation in the proposed genetic algorithm can be briefly explained as follows: the adaptive objective evaluation hyperplane divides the search space into two parts. One contains Pareto optimal solutions and the ideal point which is unreachable, the other only contains dominated solutions. During the evolutionary process and under the pressure of selection operation, the adaptive objective evaluation hyperplane is updated repeatedly and moved toward the ideal point to enforce all Pareto solutions get close to the ideal point as much as possible. Finally all Pareto solutions get to its Pareto frontier.

## C. Selection

In GA, the selection plays a very important role. When we regard the genetic operation as the exploration for the search in solution space, the selection can be thought as the exploitation for the GA to guide the evolutionary process. The selection used here is the method combined with the *roulette wheel* and *elitist* approach, in order to enforce the GA proposed to freely search solution space. The roulette wheel selection which is one of the fitness-proportional method is used to randomly reproduce new generation and the elitist method is employed to preserve the best chromosome for the next generation and overcome the stochastic errors of sampling. Then we can express our selection method used here as follows:

*Step 1.* Calculate a cumulative probability  $a_p$  for each chromosome  $\mathbf{X}_p$ , ( $p = 1, 2, \dots, chr\_size$ ).

*Step 2.* Generate a random real number  $r$  in  $[0, 1]$ .

*Step 3.* If  $r \leq a_1$ , then select the first chromosome  $\mathbf{X}_1$ ; otherwise select the  $p$ -th chromosome  $\mathbf{X}_p$  ( $2 \leq p \leq chr\_size$ ) such that  $a_{p-1} < r \leq a_p$ .

*Step 4.* Repeat Steps 2 and 3  $pop\_size$  times and obtain  $pop\_size$  copies of chromosomes.

*Step 5.* If the best chromosome is not selected in the next generation, replace one randomly from new population by the best one.

Using this selection process, we can keep the best chromosome from the current generation to the next generation.

## D. Crossover

We employed the multi-point crossover (or called uniform crossover). This type of crossover is accomplished by selecting two parent solutions and randomly taking a component from one parent to form the corresponding component of the offspring (see Fig. 4).

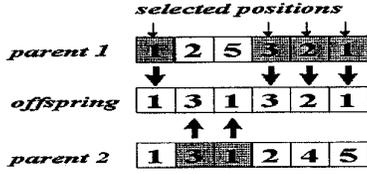


Fig. 4. Illustration of the multi-point crossover operator

### E. Mutation

We used here swap mutation which simply select two positions at random and swap their contents(see Fig. 5).

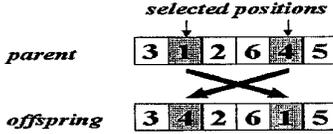


Fig. 5. Illustration of the swap mutation operator

### F. Modification of the Chromosomes

Because of the existence of the maximum number which is capable of connecting on each center, the chromosomes generated randomly in the initial population and the offspring produced by crossover may be illegal in the sense of violating the maximum number of connection for each center. There are two strategies to deal with this violation: the penalizing strategy and the repairing strategy. For some combinatorial optimization problems, it is really difficult to give out an adequate penalizing factor for those illegal chromosomes when the illegality is not easily measured quantitatively. In this case the penalizing strategy has the disadvantage that it may never generate legal solutions and the search process may be stranded in illegal local optima[11]. Because it is relatively easy to repair an illegal chromosome for the spanning tree-based genetic algorithm for LAN topology problems, we adopt the repairing strategy to modify the connection number for center in an illegal chromosome.

Let  $\bar{G}$  be the set of centers whose the maximum number of connection has not been checked and modified in a chromosome. If a center  $i$  violates the constraint with the maximum number  $g_i$  of connection for center  $i$ , this means that the number of this center in the chromosome is more than  $g_i - 1$ . Then decrease the number of the center by checking the extra center and randomly replace it with another center from  $\bar{G}$ .

Obviously it is easy to go on with this modification method although there are different connection constraints on different centers.

## IV. NUMERICAL EXAMPLES

The spanning tree-based genetic algorithm proposed in this paper for solving bicriteria LAN topology design problem is implemented in C language and run on a Pentium 133 PC. The performance of this approach is tested

with the following problem: we employ the problem with 6 service centers( $n = 6$ ), 30 users( $m = 30$ ),  $g_i = 10$ ,  $C_i = 300$ , and  $\beta_{ij} = 0.1$ . And it can be seen that the user matrix  $U$  is as follows[10]:

$$U = \begin{pmatrix} U_3 & U_1 & 0 & U_1 & 0 & U_1 \\ U_1 & U_3 & U_1 & 0 & U_1 & 0 \\ 0 & U_1 & U_3 & U_1 & 0 & U_1 \\ U_1 & 0 & U_1 & U_3 & U_1 & 0 \\ 0 & U_1 & 0 & U_1 & U_3 & U_1 \\ U_1 & 0 & U_1 & 0 & U_1 & U_3 \end{pmatrix}$$

where 0 is the zero matrix and

$$U_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}; U_3 = \begin{pmatrix} 0 & 3 & 3 & 3 & 3 \\ 3 & 0 & 3 & 3 & 3 \\ 3 & 3 & 0 & 3 & 3 \\ 3 & 3 & 3 & 0 & 3 \\ 3 & 3 & 3 & 3 & 0 \end{pmatrix}$$

And the connection costs, *i.e.*,  $w_{1ij} \in [100, 250]$ ,  $w_{2ij} \in [1, 100]$ , are randomly generated.

The parameters for genetic algorithm are set as follows:  $pop\_size = 10$ ,  $max\_gen = 500$ ,  $p_C = 0.4$ ,  $p_M = 0.1$  and experimented by 50 times. The Pareto optimal solutions can be found at most time which are illustrated in Fig. 6. The obtained solutions is formed in Pareto frontier. From the result, we can see the proposed algorithm has a good performance on the bicriteria LAN topology design problem.

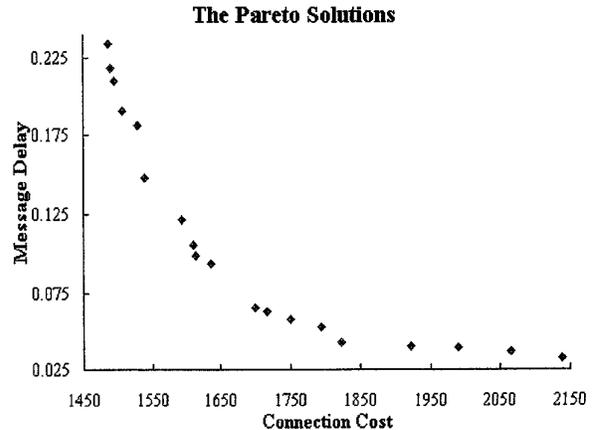


Fig. 6. The Pareto Solutions for Example

We used the TOPSIS method proposed by Yoon and Hwang[19], to determine the best compromise solution among Pareto solutions. TOPSIS stands for technique for order preference by similarity to ideal solution, which is based upon the concept that the chosen alternative should have the shortest distance from the positive ideal solution and the farthest from the negative ideal solution. The detail information about TOPSIS can be referred in references [2] and [19]. So, in Fig. 7, we represent the best compromise solution for the Pareto solutions of Fig.6, using TOPSIS method. And the chromosome of the best compromise solution in Fig. 7 is

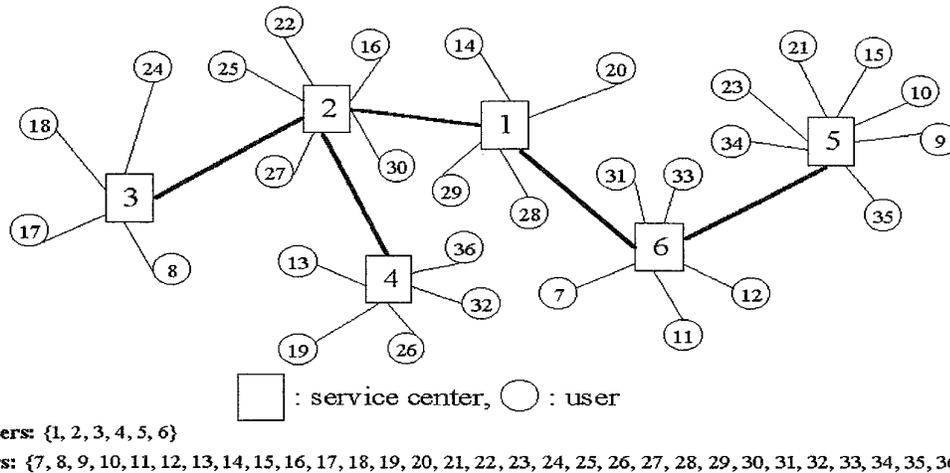


Fig. 7. The Best Compromise Solution for Pareto Solutions

[6355664152334152532242112646556124] with 1614 connection cost and 0.09935 message delay.

## V. CONCLUSION

In this paper, we proposed a evolutionary algorithm by employing spanning tree-based genetic algorithm to solve bicriteria LAN topology design problems. We also developed the method to search for the Pareto solutions. And our proposed spanning tree-based method is able to find the LAN configurations, *i.e.*, partition of the users into service centers and routing between the service centers, with minimizing the connection cost and the average message delay. Finally, we try on some experiments to demonstrate the effectiveness of our proposed method. Also, the registered Pareto optimal solutions technique was used to give out the compromise solution and the TOPSIS method is used to determine the best compromise solution among Pareto solutions. The numerical analysis shows that the set of Pareto solutions approximately in nondominated hyperplane of the criteria space.

## ACKNOWLEDGEMENT

This research was supported by the International Scientific Research Program, the Grant-in-Aid for Scientific Research(No. 07045032:1995.4-1998.3) between Ashikaga Institute of Technology, Japan and Institute of Policy and Management, Chinese Academy of Sciences by the Ministry of Education, Science and Culture, the Japanese Government.

## REFERENCES

- [1] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, 2nd ed., Springer-Verlag, New York, 1994.
- [2] Gen, M. and R. Cheng, *Genetic Algorithms and Engineering Design*, John Wiley & Sons, New York, 1997.
- [3] Cormen, T. H., C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, The MIT Press, 1990.
- [4] Bertsekas, D. and R. Gallager, *Data Networks*, 2nd ed., Prentice-Hall, New Jersey, 1992.
- [5] Tanenbaum, A. S., *Computer Networks*, 3rd ed., Prentice-Hall, New Jersey, 1996.

- [6] Gen, M., R. Cheng, and D. Wang, Genetic algorithms for solving shortest path problems, *Proceedings of IEEE International Conference on Evolutionary Computation*, pp.401-406, 1997.
- [7] Zhou, G., and M. Gen, Evolutionary computation on multicriteria production process planning problem, *Proceedings of IEEE International Conference on Evolutionary Computation*, pp.419-424, 1997.
- [8] Zhou, G., and M. Gen, Approach to degree-constrained minimum spanning tree problem using genetic algorithm, *Engineering Design & Automation*, Vol.3, No.2, pp.157-165, 1997.
- [9] Gen, M. and Y. Z. Li, Spanning tree-based genetic algorithm for bicriteria transportation problem, *Proceedings of Japan-China Joint International Workshops on Information Systems*, pp.123-134, 1997.
- [10] Elbaum, R. and M. Sidi, Topological design of local-area networks using genetic algorithms, *IEEE/ACM Transactions on Networking*, Vol.4, No.5, 1996.
- [11] Orvosh, D. and L. Davis, Using a genetic algorithms to optimize problems with feasibility constraints, *Proceedings of IEEE International Conference on Evolutionary Computation*, IEEE Press, Orlando, Florida, pp.548-552, 1994.
- [12] Dengiz, B., F. Altiparmak, and A. E. Smith, "A Genetic Algorithm Approach to Optimal Topology Design of All Terminal Networks", *Intelligent Engineering Systems Through Artificial Neural Network*, Vol.5, pp.405-410, 1995.
- [13] Dengiz, B., F. Altiparmak, and A. E. Smith, "Efficient Optimization of All-terminal Reliable Networks Using Evolutionary Approach", *IEEE Transaction on Reliability*, Vol.46, No.1, pp.18-26, 1997.
- [14] Kumar, A., R., M. Pathak, Y. P. Gupta, and H. R. Parsaei, "A Genetic Algorithm for Distributed System Topology Design", *Computers and Industrial Engineering*, Vol.28, No. 3, pp.659-670, 1995.
- [15] Kumar, A., R., M. Pathak, and Y. P. Gupta, "Genetic-Algorithm-Based Reliability Optimization for Computer Network Expansion", *IEEE Transactions on Reliability*, Vol.44, No.1, pp.63-72, 1995.
- [16] Walters, G. A. and D. K. Smith, "Evolutionary Design Algorithm for Optimal Layout of Tree Networks", *Engineering Optimization*, Vol.24, pp.261-281, 1995.
- [17] Deeter, D. L. and A. E. Smith, "Heuristic Optimization of Network Design considering All-Terminal Reliability", *Proceedings of Annual Reliability and Maintainability Symposium*, pp.194-199, 1997.
- [18] Skiena, S., *Implementing Discrete Mathematics Combinatorics and Graph Theory with Mathematica*, Addison-Wesley, Reading, MA, 1990.
- [19] Hwang, C. and K. Yoon, *Multiple Attribute Decision Making: Methods and Applications*, Springer-Verlag, Berlin, 1981.