

Peter Kortiš – Vladimír Olej – Karol Blunár \*

# AN EUGENETIC ALGORITHM FOR DELAY-CONSTRAINED MINIMUM-COST ROUTING OF MULTIPOINT CONNECTIONS

*A special kind of genetic algorithm - eugenic algorithm is presented for constructing minimum-cost tree with delay constraints. Telecommunication network is represented by an undirected graph, which uses three independent metrics: cost, delay and capacity. The efficiency of two different bit representations of individuals (adjacent matrix, list of vertices) is compared.*

## 1. Introduction

Recent advances in telecommunication and computer technology have led to a new variety of applications based on communication of multiple parties. This brings a new dimension into the existing telecommunication networks, routing of multipoint connections. Multipoint connections are used by conference services, which need to establish point-to-multipoint and multipoint-to-multipoint connections [6]. The main goal of routing algorithm is to find connection, which fulfils conditions of specific service, as are: free bandwidth, delay, delay variation, etc. The routing algorithm has to find such a connection, which fulfils all the given conditions, and the cost of this connection is as low as possible. The cost of connection depends on the length (geographical) of the connection, number of used network resources, utilization, delay of separate links in connection, bandwidth, etc. The routing algorithm prefers less utilized parts of the network, which results in efficient usage of network capacity and consequently the blocking probability will decrease. The crux of routing of multipoint connections is described in [9], where many deterministic algorithms are presented. Nowadays, algorithms for routing of multipoint connections could be based on genetic algorithm (GA) [3, 7, 8] or eugenic algorithm (EuA) [1, 4, 5].

## 2. Network model and problem definition

Having a telecommunication network represented by the graph

$$G = (V, E), \quad (1)$$

where the switching nodes of telecommunication network are represented by the set of vertices  $V$  and communication links are represented by the set of edges  $E$ . Consider three non-negative real functions associated with each edge (link)  $e$ :

$$C(e) \in \mathbb{R}^+ + \{0\}, D(e) \in \mathbb{R}^+ + \{0\}, F(e) \in \mathbb{R}^+ + \{0\}. \quad (2)$$

The function  $C(e)$  determines the cost of the link, which depends mainly on its utilization. The link delay  $D(e)$  is the delay of a data packet experiences on the corresponding link and the function  $F(e)$  determines free capacity of the link.

Given a set of nodes, which have to be connected:

$$U = \{u_1, u_2, u_3, \dots, u^m\}; (U \subseteq V), \quad (3)$$

This set represents the nodes in the network, which the subscribers of one multipoint connection are connected to. Every vertex of set  $U$  must be connected with every single vertex of set  $U$ . The problem of routing multipoint connection in the telecommunication network could be formulated as the problem of finding the subgraph  $G'$ , which is usually tree  $T$ . Consider the subgraph  $G'$  to be the tree  $T \subseteq G$ , where the set of vertices of the tree  $T$  includes all vertices from the set  $U$ . The desired tree must fulfil these two criteria:

- Free capacity of all edges must be equal or higher than the required value  $\Phi$ :  $\forall e \in T: F(e) \geq \Phi$ , (4)
- Delay between all couples of vertices from set  $U$  must be lower or equal to the maximum acceptable delay  $\Delta_{ij}$  defined for a selected couple of vertices:

$$\forall i, j \in 1 \dots m; i \neq j; P = \text{Path}(u_i, u_j) : D(P) \leq \Delta_{ij}, \quad (5)$$

where  $P$  is the path from vertex  $u_i$  to vertex  $u_j$ .

Cost  $C_T$  of the tree  $T$  is defined as follows:

$$C_T = 2 \sum_{e \in T} C(e) \quad (6)$$

The goal of the routing algorithm is to find the tree, which fulfils the criteria (4) and (5) and the cost of the tree is as low as possible.

If the cost of the tree is minimal, the tree is called Steiner tree and the vertices of the tree, which are not in the set  $U$ , are called Steiner vertices [2].

\* Peter Kortiš<sup>1</sup>, Vladimír Olej<sup>2</sup>, Karol Blunár<sup>1</sup>

<sup>1</sup>Department of Telecommunications, Faculty of Electrical Engineering, University of Žilina, Veľký Diel, 010 26 Žilina, Slovak Republic, E-mail: Peter.Kortis@fel.utc.sk

<sup>2</sup>Institute of System Engineering and Informatics, Faculty of Economics and Administration, University of Pardubice, Studentská 84, 532 10 Pardubice, Czech Republic E-mail: Vladimír.Olej@upce.cz

### 3. Eugenic algorithm

Eugenic algorithm [1, 4, 5] differs from GA [3] in many ways. The most significant difference is the method of creation of new individuals. Whereas GA creates new individuals randomly using crossover, mutation and inversion operators, the EuA creates a new individual using the analysis of selected individuals in population. Even though the analyses are deterministic, there are many random steps in the algorithm. Eugenic algorithm could be defined as follows [1, 4, 5]:

```

1. pop = random-population ( )          /* Randomly initialize the population pop */
   L = 0                                /* L counts the number of individuals created by EuA */
2. DO WHILE (L < Lmax)                  /* Creation of new individuals */
   G = {g1, g2, ..., gl}              /* G is the set of unset genes */
   rpop = pop                            /* rpop is the copy of population dedicated for restriction */
3. DO WHILE non-empty (G)               /* Choose unset gene and set its allele */
4. gg = most-significant-gene (rpop, G)
5. gg,a = set-allele (gg)              /* Set allele gg,a of the gene gg */
6. Remove gg from set G                 /* Remove unset gene gg from the set of unset genes */
7. E = epistasis-of-population (rpop, G) /* E is epistasis of restricted population rpop */
8. pr = probability-of-restriction (E)   /* Perform restriction, if epistasis E is high */
9. IF (random[0,1] < pr) THEN
10. rpop = restrict-population (rpop, gg,a)
    END WHILE
11. U = lowest-fitness-individual (pop)
12. popU = g                            /* Replace lowest-fitness individual with the new individual */
13. L = L + 1
    END WHILE

```

The EuA begins by generating initial random population. The algorithm creates a new individual  $g$ . The algorithm continues in creation of new individuals until the maximum number of individuals  $L_{max}$  has been created. Every new individual is created step by step.

The EuA constructs a new individual step-by-step (allele-by-allele) using the analysis of selected part of population ( $rpop$ ). In the initialization phase of the algorithm the set of unset genes is full. This set includes genes, where the alleles are still unknown and will be set in the next steps. Alleles of unset genes are not generated randomly, but according to their significance. The most significant genes are set as the first and the least significant as the last. The most significant gene and its allele is determined from the restricted population  $rpop$ . The allele of the most significant gene is selected randomly, but in accordance with its significance. When the allele of the gene is set, the gene is removed from the set of unset genes  $G$  and the restriction of population  $rpop$  could be performed. Whether the restriction is performed or not, depends on the amount of interdependence among a genotype's alleles called epistasis. The higher the epistasis, the higher is the probability that a population is restricted to only individuals containing selected allele of the most significant gene  $g_{g,a}$ . All individuals that do not have allele  $g_{g,a}$  are excluded from the population  $rpop$ . The primary goal of restriction is to leave in population such individuals who are similar to the individuals currently being created. This technique enables to estimate which genes are more significant

for creating individual and which allele is more suitable. If the gene significance analysis were performed over the entire population, there would be no regard for the relationship among separated genes. If one of the genes is set to the same allele in the entire population, then this gene is called gene without diversity. If such a gene appeared in population, its allele would never change. Because of this EuA has mechanism, which allows setting another allele for the gene without diversity. The probability of this change is equal to parameter creation rate CR. This procedure repeats until all genes of new individual  $g$  are set. If all alleles

of newly created individual are determined, the fitness of this individual is computed and the newly created individual will replace the lowest fitness individual in the population.

To find the most significant gene and allele and compute its epistasis, it's necessary to analyze fitness of separate alleles of individuals from restricted population  $rpop$ . The analysis is usually based on the computation of allele fitness average of unset genes. Calculated averages are normalized to probabilities of allele presence for each gene  $p(g_{g,a})$ . If the probability  $p(g_{g,a})$  is not one or zero and the absolute value of difference of probabilities  $p(g_{g,a})$  is maximal, the gene  $g_g \in G$  is the most significant gene. This could be expressed as follows:

$$g_g = \underset{g_k \in G: p(g_{k,0}) \neq 0 \wedge p(g_{k,1}) \neq 1}{\operatorname{argmax}} \{ |p(g_{k,0}) - p(g_{k,1})| \}. \quad (7)$$

The most significant allele of gene  $g_g$  is the allele with the highest probability of presence, i.e.:

$$g_{g,a} = \underset{g_{g,a}}{\operatorname{argmax}} \{ p(g_{g,0}), p(g_{g,1}) \}. \quad (8)$$

The amount of interdependence among a genotype's alleles (epistasis) could be calculated as follows:

$$E = 1 - |p(g_{j,0}) - p(g_{j,1})|, \quad (9)$$

where gene  $g_j \in G \setminus \{g_g\}$  is the second most significant gene after gene  $g_g$ .

If the allele probabilities are approximately equal (i.e. approximately equal to  $1/2$ ), the epistasis goes to one and if one of the probabilities goes to zero, the epistasis will equal zero (see expression 9).

The probability of restriction  $p_r$  is commensurable to epistasis and for the majority of applications it's sufficient, if probability is equal to epistasis  $p_r = E$ . The result of this fact is that the restriction of the population is more likely, when the relation among alleles (epistasis) of unset genes is low. The restriction is not performed if the number of individuals after restriction would be lower than minimum acceptable number of individuals  $MinN$ , otherwise the restriction could be performed with the probability  $p_r$ .

#### 4. Model design for routing of multipoint connections, simulation results and analysis.

Using the principles described in section 3, EuA for routing of multipoint connections was designed. The goal of the algorithm is to find minimum cost connection with regard to criteria (4) and (5). The proposed algorithm uses one of two chosen representations of individuals (connections): adjacent matrix and list of vertices.

The adjacent matrix is the square matrix with dimension equal to the number of nodes in the network (fig. 1). The matrix elements could only be set to one of two values (0 or 1), which determines, whether the connection between two particular nodes exists or not. If the matrix element on the  $i$ -th row and  $j$ -th column is 1, then the connection from node  $i$  to node  $j$  exists and vice versa.

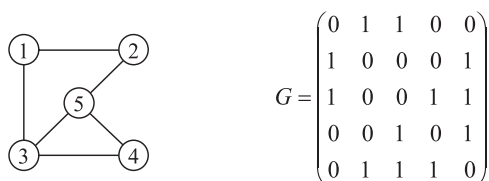


Fig. 1: The example of simple network and corresponding adjacent matrix

The list of vertices is the line vector with dimension equal to the number of nodes in the network. The vector element determines, if corresponding node must be presented in the connection or not. If the vector element is set to 1 the node is presented in connection, otherwise the corresponding node could be presented, depending on the state of the network. The tree is created using the minimum spanning tree (MST) algorithm, which respects only the delay between nodes.

The MST algorithm is described as follows [10]:

1. Given a graph  $G$ , construct a new graph  $G' = (U, E')$  where delay  $D(e')$ ;  $e' = (u_i, u_j)$  is the length of the shortest path from  $u_i$  to  $u_j$  in  $G$ .

2. Construct a minimum spanning tree  $T'$  in  $G'$ .
3. Convert the edges in  $T'$  to paths in  $G$  to form a solution  $T$ .

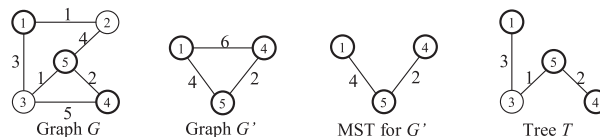


Fig. 2: Minimum spanning tree algorithm adjacent matrix

The list of vertices can represent only trees, unlike the adjacent matrix, which could represent subgraphs too.

The routing algorithm consists of two phases:

- pre-processing phase
- eugenic algorithm

The main goal of pre-processing phase is to reduce the graph  $G$  to the graph  $G_R$  with edges that represent links with sufficient free capacity. The reduced graph  $G_R$  includes only edges  $e$  with free capacity equal or higher than minimum required capacity  $\Phi$  i.e.  $F(e) \geq \Phi$ . After the reduction, the shortest paths among all vertices from set  $U$  are computed and compared with maximum acceptable delay given by parameter  $\Delta_{ij}$  (5), to ascertain if exists any suitable connection, which fulfils conditions (4) and (5). If such connection does not exist, the algorithm is interrupted; otherwise the eugenic algorithm is performed.

Given the network from fig. 3, which consists of 22 nodes interconnected with bidirectional links. The cost of every link is presented as the number beside the line. The delay of every link in the graph is 10 ms and free capacity is higher than minimum required value  $\Phi$ . Nodes 6, 14, 17, 19 and 21 drawn with thick circles must be connected.

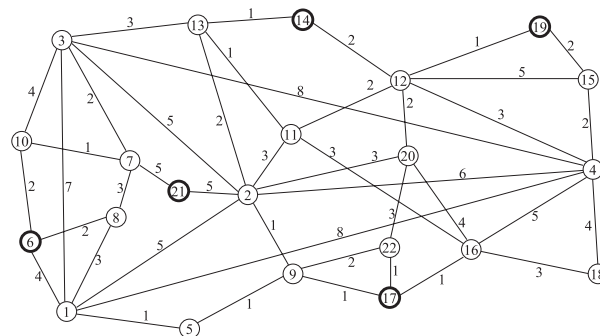


Fig. 3: The structure of the network

The proposed algorithm was tested on various network architectures, but this paper contains only results of the tests performed on the network from figure 3, because the tests on other networks give similar results. The tests were performed using both representations: adjacent matrix and list of vertices.

The cost of the resultant connection depends on these parameters as a number of generations  $L_{max}$ , size of population  $MaxN$ , and parameters  $MinN$  and  $CR$ . To find the effect of particular parameters to algorithm efficiency the following experiment was done. The average value of minimum, maximum and average cost of connection were calculated from 100 independent experiments for many combinations of the above mentioned parameters. Table 2 specifies the chosen values of tested parameters for both representations. The maximum acceptable delay  $\Delta_{ij}$  is set to 50 ms for all couples from the set  $U$ .

The sets of values for particular simulation parameters Tab. 2

$L_{max}$	{300; 1000}
$MaxN$	{10; 20; 50; 100}
$MinN$	{1; 0.1MaxN; 0.2MaxN; 0.3MaxN; 0.4MaxN; 0.5MaxN}
$CR$	{0; 0.001; 0.005; 0.01; 0.015; 0.02; 0.03; 0.05; 0.1; 0.2; 0.3}

Min. Cost	MinN				
	1	2	3	4	5
0	90	90	90	90	90
0.001	88.86	88.78	89.32	89	89.26
0.005	80.28	81.76	80.36	83.3	82.98
0.01	71.96	72.48	70.4	72.84	71.84
0.015	66.64	66.94	67.8	66.9	66.84
0.02	64.34	65.3	63.24	66.1	65.38
0.03	63.68	63.64	62.86	63.36	63.64
0.05	65.42	65.76	64.84	64.82	66.06
0.1	76.66	75.22	74.88	75.86	76.28
0.2	88.94	89.66	89.26	88.92	89.06
0.3	90	90	90	90	90

Note: figures 4, 5, 6 and 7 depict only the average value of minimum cost because the resultant connection is represented by the lowest cost individual in the population.

The experiments show that the representation by list of vertices gives significantly better results than the representation by an adjacent matrix because the best average cost of the connection represented by the adjacent matrix is 62.86 after 1000 generations, but for a list of vertices is 48 after only 300 generations (see fig. 4 and 6). This significant difference in performance is caused by the convergence speed, which is lower for the adjacent matrix than for the list of vertices representation because the minimum cost after 300 generations is approximately 72 when the adjacent matrix representation is used. (Note: The cost in 300th generation is not included in any table of this paper because it was obtained from another simulation.)

The cost of connection is independent on a minimal acceptable number of individuals after restriction  $MinN$ . Stochastic char-

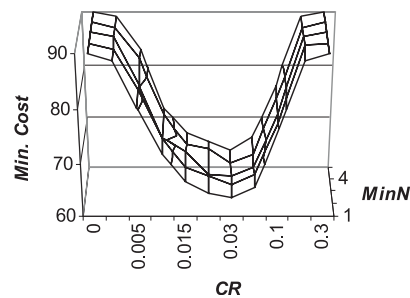


Fig. 4: Minimum cost (Adjacent matrix;  $MaxN = 10$ ;  $L_{max} = 1000$ )

Min. Cost	MaxN			
	10	20	50	100
0	90	90	90	90
0.001	88.86	88.84	87.68	85.9
0.005	80.28	76.42	73.68	84
0.01	71.96	65.88	73.34	83.06
0.015	66.64	64.52	75	81.92
0.02	64.34	65.3	74.36	81.1
0.03	63.68	66.82	76.52	79.58
0.05	65.42	69.6	80.72	84.72
0.1	76.66	83.6	87.52	88.98
0.2	88.94	89.78	89.92	89.94
0.3	90	89.96	90	90

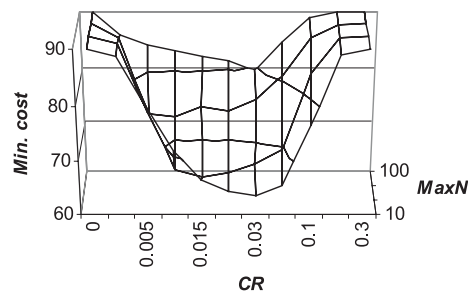


Fig. 5: Minimum cost (Adjacent matrix;  $MinN = 1$ ;  $L_{max} = 1000$ )

Min. Cost		MinN					
		1	2	4	6	8	10
CR	0	50.6	51.1	50.74	50.72	50.54	50.94
	0.001	50.02	49.86	49.74	50.06	49.92	50
	0.005	49.04	48.92	48.74	48.92	48.64	48.7
	0.01	48.44	48.28	48.32	48.48	48.26	48.18
	0.015	48.16	48.12	48.2	48.16	48.18	48.28
	0.02	48.1	48.12	48.14	48.24	48.1	48.14
	0.03	48.1	48.12	48.06	48.24	48.06	48.16
	0.05	48	48.06	48	48.06	48.08	48.06
	0.1	48.06	48.14	48.02	48.04	48.12	48.04
	0.2	48.98	48.9	48.66	48.8	48.86	48.84
0.3	50.1	49.7	49.96	49.9	49.78	49.78	

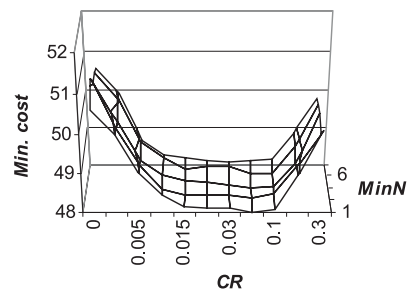


Fig. 6: Minimum cost (List of vertices;  $MaxN = 20$ ;  $L_{max} = 300$ )

Min. Cost		MaxN			
		10	20	50	100
CR	0	51.64	50.6	49.62	48.54
	0.001	51.28	50.02	49.18	48.6
	0.005	49.12	49.04	48.6	48.54
	0.01	48.7	48.44	48.46	48.4
	0.015	48.66	48.16	48.74	48.36
	0.02	48.58	48.1	48.44	48.28
	0.03	48.48	48.1	48.48	48.46
	0.05	48.2	48	48.42	48.38
	0.1	48.04	48.06	48.62	48.28
	0.2	48.22	48.98	48.42	48.3
0.3	49.46	50.1	49	48.48	

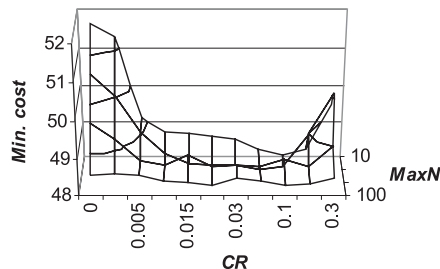


Fig. 7: Minimum cost (List of vertices;  $MinN = 1$ ;  $L_{max} = 300$ )

acter of the algorithm and low number (100) of performed experiments cause small differences among minimal costs for selected values of parameter  $MinN$ . The total number of individuals in the population ( $MaxN$ ) has significant effect on the cost of connection only when the adjacent matrix representation is used.

In the given adjacent matrix representation, increasing the number of individuals has negative impact on the speed of convergence to the optimal solution (see fig. 5). Despite the fact the parameter  $MaxN$  is significant, the parameter  $CR$  has bigger impact on the cost of the resultant individual. The optimal value for parameter  $CR$  is approximately 0.03.

If the list of vertices representation is used, the cost of the connection depends mainly on the parameter  $CR$ . The optimal value is approximately 0.05.

The difference in performance of both representations is illustrated in figures 8 and 9 that depict the final connections and appropriate parameters of simulations.

## 5. Conclusion

The performed experiments show that the representation using list of vertices gives better results than adjacent matrix representation; because the resultant connection does not contain any redundant links. The redundant links disappear when the number of generations increases, but it has negative impact on the execution time. Therefore the representation using adjacent matrix does not seem to be suitable. The mentioned kinds of representations lead to the solutions with a different number of degrees of freedom. The number of degrees of freedom for the adjacent matrix is equal to the number of links in the network  $|E|$ , which is higher than the number of degrees of freedom for the list of vertices representation, which is equal to  $|V| - |U|$ . A large number of degrees of freedom causes disproportionate increase of feasible search areas in search space and therefore the algorithm is more likely to stay in a local extreme.

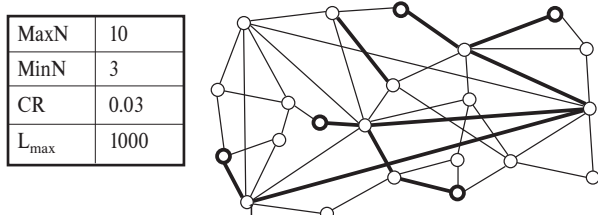


Fig. 8: Resultant connection for adjacent matrix representation (cost = 64)

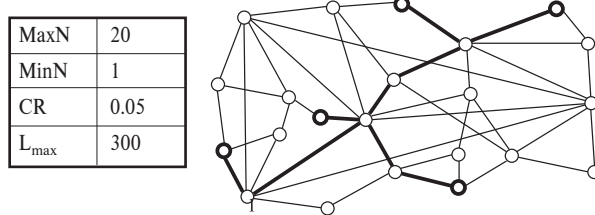


Fig. 9: Resultant connection for list of vertices representation (cost = 48)

## References

- [1] PRIOR, J. W.: *Eugenic Evolution for Combinatorial Optimization*. Report AI 98 - 268. Artificial Intelligence Laboratory. The University of Texas at Austin. 1998.
- [2] KAPSALIS, A., RAYWARD, SMITH, V. J., SMITH, G. D.: *Solving the Graphical Steiner Tree Problem Using Genetic Algorithms*. Journal of the Operational Research Society. Vol. 44, No.4, 1993, pp. 397-406.
- [3] OLEJ, V.: *Evolution Stochastic Optimization Algorithms: Genetic Algorithms and Evolution Strategies*. Electrical Engineering Journal, ISSN 1335-3632, Vol. 50, No. 9-10, 1999, pp. 266-272.
- [4] OLEJ, V.: *Prediction of Gross Domestic Product Development by Frontal Neural Networks with Learning Process on the Basis Genetic and Eugenic Algorithms*. Neural Network World, International Journal on Non-Standard Computing and Artificial Intelligence, editor: M. Novák, ISSN 1210-0552, Czech Republic, Vol. 12, No. 3, 2002, pp. 279-291.

- [5] OLEJ, V.: *Prediction of Gross Domestic Product Development on the Basis of Frontal Neural Networks*, Genetic and Eugenic Algorithms. 2nd Euro - International Symposium on Computational Intelligence, E - ISCI 2002, Intelligent Technologies - Theory and Applications, New Trend in Intelligent Technologies, IOS Press Ohmsha, ISSN 0922-6389, ISBN 1 58603 256 9 IOS Press, ISBN 4 274 90512 8 Ohmsha, Netherlands, 2002, pp. 309–314.
- [6] ITU, T.: *Recommendation F.700: Framework Recommendation for audiovisual/ multimedia services*. July 1996.
- [7] ESBENSEN, H.: *Computing Near - Optimal Solutions to the Steiner Problem in a Graph Using Genetic Algorithm* [online], 1995. <http://www.daimi.au.dk/PB/468/PB-468.ps.gz>.
- [8] VRBA, R.: *Možnosti tvorby viachodových spojení v sieťach B - ISDN pomocou evolučných stochastických optimalizačných algoritmov*. Dizertačná práca, Žilinská univerzita, Elektrotechnická fakulta, Žilina, 2000.
- [9] RAYWARD, SMITH, V. J., CLARE, A.: *On Finding Steiner Vertices*. Networks. Vol. 16. 1986. pp. 283–294.
- [10] TANAKA, Y., HUANG, P. C.: *Multiple Destination Routing Algorithms*. IEICE Transaction on Communications. Vol. E76-B. No. 5. 1993. s. 544–552.