

Weighted Sum-based Genetic Algorithm for Bicriteria Network Design Problem

Mitsuo Gen* and Lin Lin**

* Graduate School of Information, Production & Systems, Waseda University

email: gen@waseda.jp

** Graduate School of Information, Production & Systems, Waseda University

email: lin@ruri.waseda.jp

Abstract - This paper proposes a new Multiobjective Genetic Algorithm (moGA) approach for Bicriteria Network Design Problem (BNP). The objectives are to maximize flow and minimize cost. The proposed method adopts priority-based encoding method to represent a path in the network. Different from other encoding methods, such as path oriented encoding method, priority-based encoding method can be applied for different network design problems, i.e., Shortest Path Problem (SPP), Maximum Flow Problem (MXF), Minimum Cost Flow Problem (MCF), etc. In the proposed method, while weighted-sum approach is employed to evaluate solutions found in the search process, nondominated sorting technique is used to obtain Pareto optimal solutions. Numerical analysis shows the efficiency and effectiveness of the moGA approach on the BNP.

I. INTRODUCTION

Network design problems are fundamental issue in the several fields, including applied mathematics, computer science, engineering, management, and operations research. Networks provide a useful way to modeling real world problems and are extensively used in many different types of systems: communications, hydraulic, mechanical, electronic and logistics.

Shortest path problem (SPP), *maximum flow problem* (MXF) and *minimum cost flow problem* (MCF) are also well known network design problems. While in SPP, a path is determined between two specified nodes of a network that has minimum length, or the maximum reliability or takes least time to traverse, MXF finds a solution that sends the maximum amount of flow from a source node to a sink node. MCF is the most fundamental of all network design problems. In this problem, the purpose is to determine a least cost shipment of a commodity through a network in order to satisfy demands at certain nodes from available supplies at other nodes (Ahuja, 1993). These problems have been well studied and many efficient polynomial-time algorithms have been developed by Dijkstra (1959), Dantzig (1960), Ford and Fulkerson (1956), Elias *et al.* (1956), Ford and Fulkerson (1962) and Zadeh (1973).

In the real world, there are usually such cases that one has to consider simultaneously multicriteria in network design problems. The problems may arise when designing a communication system, logistic systems and highways. For

example, in a logistic system, besides the cost of shipment, other factor such as maximum flow can be considered, or in communication system, the construction cost and the delay cost can be taken consideration with together. The multicriteria network design problem is not simply an extension from single objective to two objectives. In generally, we can not get the optimal solution of the problem because these objectives usually conflict with each other in practice. The real solutions to the problem are a set of Pareto optimal solutions (Chankong and Haimes, 1983) [19], but the calculation of it is a difficult task because it is an NP-hard problem and no previous work in this area has been reported in literature.

Recently, Genetic Algorithm (GA) has received one of great deal of attention regarding their potential as optimization techniques for network design problems and is often used to solve many real world problems, including the effective approaches on the multiobjective optimization problems (Fonseca and Fleming, 1993; Schaffer, 1984). In this paper, Bicriteria Network Design problem (BNP) with maximum flow and minimum cost has been considered and a new genetic algorithm approach is proposed. The proposed method adopts priority-based encoding method to represent a path in the network. Different from other encoding methods, such as path oriented encoding method, priority-based encoding method can be applied for different network design problems, such as shortest path problem, maximum flow problem, minimum cost flow problem, etc. In the proposed method, while weighted-sum approach is employed to evaluate solutions found in the search process, nondominated sorting technique is used to find Pareto optimal solutions.

The paper is organized as follows: In Section 2, bicriteria network design problem is defined. While our GA approach is discussed in Section 3, computational results are given in Section 4. Section 5 includes conclusion.

II. BICRITERIA NETWORK DESIGN PROBLEM

The bicriteria shortest path problem is one of Bicriteria Network Design Problems (BNP), which of finding a diameter-constrained shortest path from a specified source node s to another specified sink node t . This problem, termed the multi-objective shortest path problem (MOSP) in the literature, is NP-complete and Warburton [20] presented the first fully polynomial approximation scheme (FPAS) for it. Hassin [21] provided a strongly polynomial FPAS for the problem which improved the running time of Warburton [20].

Cheng and Gen [3] presented a compromise approach-based genetic algorithm for the problem.

The general classes of BNP with minimum two objectives (under different cost functions) are defined and extended to the more multi-criteria network design problems. Ravi *et al.* [22] presented an approximation algorithm for finding good broadcast networks. Ganley *et al.* [23] consider a more general problem with more than two objective functions. Marathe *et al.* [24] consider three different criteria of network and presented the first polynomial-time approximation algorithms for a large class of BNP.

In this paper, we study the complexity case of BNP. The two objectives we consider are: (1) maximum flow, and (2) minimum cost. Network design problems where even one flow measure be maximized, are often NP-hard. For solving the BNP with maximum flow and minimum cost, the efficient set of paths may be very large, possibly exponential in size. Thus the computational effort required to solve it can increase exponentially with the problem size in the worst case.

Let $G=(N,A)$ be a directed network defined by a set N of n nodes and a set A of m directed arcs. Each arc $(i,j) \in A$ has an associated cost c_{ij} that denotes the cost per unit flow on that arc. We assume that the flow cost varies linearly with the amount of flow. We also associate with each arc $(i,j) \in A$ has a capacity u_{ij} that denotes the maximum amount that can flow on the arc and a lower bound 0 that denotes the minimum amount that must flow on the arc. The decision variables in BNP are the maximum possible flow z_1 with minimum cost z_2 from a specified source node s to another specified sink node t . And x_{ij} represents the flow on an arc $(i,j) \in A$. BNP is a multiobjective optimization model formulated as follows:

$$\begin{aligned} \max \quad & z_1 = f \\ \min \quad & z_2 = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s. t.} \quad & \sum_{j=1}^m x_{ij} - \sum_{k=1}^m x_{ki} = \begin{cases} f & (i = 1) \\ 0 & (i = 2, 3, \dots, n-1) \\ -f & (i = n) \end{cases} \\ & 0 \leq x_{ij} \leq u_{ij}, \quad \forall (i, j) \in A \\ & f \geq 0 \end{aligned}$$

III. NEW APPROACH OF MULTIOBJECTIVE GA

In this section, we present the proposed approach of multiobjective GA, which uses priority-based encoding method (in Section3-A), show the paths growth algorithm (in Section3-B) that can find various paths by one chromosome for solving BNP. We present a new crossover operator (in Section3-D). Also combines adaptive weights approach (AWA) (in Section3-C).

A. Genetic Representation

How to encode a path for a graph is a critical step. Special difficulty arises because (1) a path contains variable number of nodes, and (2) a random sequence of edges usually does

not correspond to a path. To overcome such difficulties, Cheng and Gen adopted an indirect approach: encode some guiding information for constructing a path in a chromosome, but not a path itself (Cheng and Gen, 1994) [25]. As mentioned earlier, the encoding method, called proposed priority-based encoding, was introduced. In this method, the position of a gene was used to represent a node and the value of the gene was used to represent the priority of the node for constructing a path among candidates. The path corresponding to a given chromosome is generated by sequential node appending algorithm with beginning from the source node 1 and terminating at the sink node n . At each step, there are usually several nodes available for consideration, only the node with the highest priority is added into path. Consider the network shown in Fig. 1 and a priority-based encoding shown in Fig. 2. Suppose we want to find a path from node 1 to node 10.

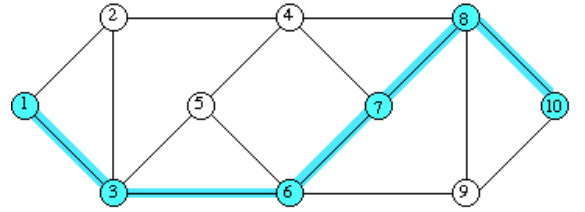


Figure 1: A simple network with 10 nodes and 16 edges

position:	node ID	1	2	3	4	5	6	7	8	9	10
value:	priority	7	3	4	6	2	5	8	10	1	9

Figure 2: An example of priority-based encoding

At the beginning, we try to find a node for the position next to node 1. Nodes 2 and 3 are eligible for the position, which can be easily fixed according to adjacent relation among nodes. The priorities of them are 3 and 4, respectively. The node 3 has the highest priority and is put into the path. The possible nodes next to node 3 are nodes 2, 5 and 6. Because node 6 has the largest priority value, it is put into the path. Then we form the set of nodes available for next position and select the one with the highest priority among them. Repeat these steps until we obtain a complete path (1, 3, 6, 7, 8, 10).

B. Population Initialization

In general, there are two ways to generate the initial population, heuristic initialization and random initialization. Although the mean fitness of the heuristic initialization is already high so that it may help the GAs to find solutions faster. Unfortunately, in most large scale problems, for example network design problems, it may just explore a small part of the solution space and never find global optimal solutions in the worst case because of the lack of diversity in the population [5]. Therefore, random initialization is adopted in this paper.

algorithm 1: priority-based encoding

input: number of nodes n

output: chromosome v_k

step 0: Initial chromosome $v_k(j) \leftarrow j$, $j = 1, 2, \dots, n$;

Initial index $i = 1$.

step 2: **If** index $i > \lceil n/2 \rceil$, **goto** step 4;

otherwise, continue.

step 3: Assign a random number less n to j and l ;

If $j \neq l$, **swap** the gene $v_k(j)$ and $v_k(l)$.

step 4: **Output** the chromosome v_k .

C. Decoding Method

To describe this decoding method, we first define a one-path growth algorithm that decodes one path base on the generated chromosome with given network; and then present a overall-path growth algorithm that obtain overall possible path for the given chromosome.

1) One-path growth algorithm: The path is generated by One-path growth algorithm that is given in algorithm 2; with beginning from the specified node 1 and terminating at the specified node n . At each step, there are usually several nodes available for consideration. We add the one with the highest priority into path.

algorithm 2: One-path growth

input: number of nodes m , chromosome v_k ,

the set of nodes S_i with all nodes adjacent to node i .

output: path P_k

step 0: the source node $i \leftarrow 1$, $P_k \leftarrow \emptyset$

step 1: **if** $S_i = \emptyset$, **goto** step 3; **otherwise, continue.**

step 2: select l from S_i with the highest priority, and go back to step 1.

if $v_k(l) \neq 0$ **then**

$v_k(l) = 0$;

$P_k \leftarrow P_k \cup \{x_{il}\}$;

$i \leftarrow l$;

else $S_i \leftarrow S_i \setminus \{l\}$

step 3: **output** the complete path P_k .

$$P_k = \{x_{1l_1}, x_{l_1l_2}, x_{l_2l_3}, \dots, x_{l_{n-1}l_n}\}$$

2) Overall-path growth algorithm: For a given path, we can calculate its flow f_k and the cost c_k . By removing the used capacity from u_{ij} of each arc, we have a new network with the new flow capacity \tilde{u}_{ij} . With the one-path growth algorithm, we can obtain the second path. By repeating this algorithm we can obtain the maximum flow for the given chromosome till no new network can be defined in this way.

algorithm 3: Overall-paths growth

input: network data (V, A, C, U) , chromosome v_k , the set of nodes S_i with all nodes adjacent to node i

output: number of paths L_k , the flow f_i^k and the cost c_i^k of each path, $i \in L_k$

step 0: number of paths $l \leftarrow 0$

step 1: **If** $S_i = \emptyset$, go to step 7; **otherwise**, $l \leftarrow l + 1$, **continue.**

step 2: The implementation of path P_l^k growth is based on

algorithm 2.

$$S_i \leftarrow S_i - \{a\}, \forall i$$

step 4: Calculate the flow f_l^k and the cost c_l^k of the path P_l^k .

$$f_l^k \leftarrow f_{l-1}^k + \min\{u_{ij} \mid (i, j) \in P_l^k\}$$

$$c_l^k \leftarrow c_{l-1}^k + \sum_{i=1}^m \sum_{j=1}^m c_{ij} (f_l^k - f_{l-1}^k)$$

step 5: Perform the flow capacity u_{ij} of each arc update. Make a new flow capacity \tilde{u}_{ij} as follows:

$$\tilde{u}_{ij} \leftarrow u_{ij} - \min\{u_{ij} \mid (i, j) \in P_l^k\}$$

step 6: **If** the flow capacity $\tilde{u}_{ij} = 0$, perform the set of nodes S_i update which the node j adjacent to node i .

$$S_i \leftarrow S_i - \{j\}, (i, j) \in P_l^k \ \& \ \tilde{u}_{ij} = 0$$

step 7: **Output** number of paths $L_k \leftarrow l - 1$, the flow f_i^k and the cost c_i^k of each path, $i \in L_k$.

D. Fitness Assignment

The weighted-sum approach can be viewed as an extension of methods used in the multiobjective optimization to the GAs. It assigns weights to each objective function and combines the weighted objectives into a single objective function. Recently, three weight setting mechanisms: the fixed weight approach (FWA), the random weights approach (RWA), and the adaptive weights approach (AWA) have been proposed [3]. In this paper, the fitness of each individual in a generation is calculated using AWA. Adaptive evaluation function based on the AWA is given in algorithm 4.

algorithm 4: Adaptive Weight Approach

input: chromosome v_k , $k \in \text{popSize}$, the flow f_i^k and the cost c_i^k of each v_k

output: fitness value $eval(v_k)$, $k \in \text{popSize}$

step 1: Define two extreme points: the maximum extreme point z^+ and the minimum extreme point z^- in criteria space as $z^+ = \{z_1^{\max}, z_2^{\max}\}$, $z^- = \{z_1^{\min}, z_2^{\min}\}$. Where z_1^{\max} , z_2^{\max} , z_1^{\min} and z_2^{\min} are the maximal value and minimal value for objective 1 and objective 2 in the current population. They are defined as follows:

$$z_1^{\max} = \max\{f_i^k \mid i \in L_k, k \in \text{pop_size}\},$$

$$z_2^{\max} = \max\{-c_i^k \mid i \in L_k, k \in \text{pop_size}\},$$

$$z_1^{\min} = \min\{f_i^k \mid i \in L_k, k \in \text{pop_size}\},$$

$$z_2^{\min} = \min\{-c_i^k \mid i \in L_k, k \in \text{pop_size}\}$$

step 2: The adaptive weight for objective 1 and objective 2 are calculated by the following equation:

$$w_1 = \frac{1}{z_1^{\max} - z_1^{\min}}, \quad w_2 = \frac{1}{z_2^{\max} - z_2^{\min}}$$

step 3: Calculate the fitness value for each individual.

$$eval(v_k) = \frac{\sum_{i=1}^{L_k} (w_1(f_i^k - z_1^{\min}) - w_2(c_i^k + z_2^{\min}))}{L_k}, \quad \forall k \in pop_size$$

E. Genetic Operators

Genetic operators mimic the process of heredity of genes to create new offspring at each generation. Using the different genetic operators has very large influence on GA performance [11]. Therefore it is important to examine different genetic operators.

1) Crossover Operator: In this paper, we proposed a new crossover operator, *weight mapping crossover* (WMX) that can be viewed as an extension of one-point crossover for permutation representation. As in one-point crossover, firstly cut-point is determined randomly and parents' left segments from cut point are copied to offspring, then remapping is realized to obtain offspring' right segment using weight of other parent's right segment. Fig. 3 shows an example of the WMX procedure.

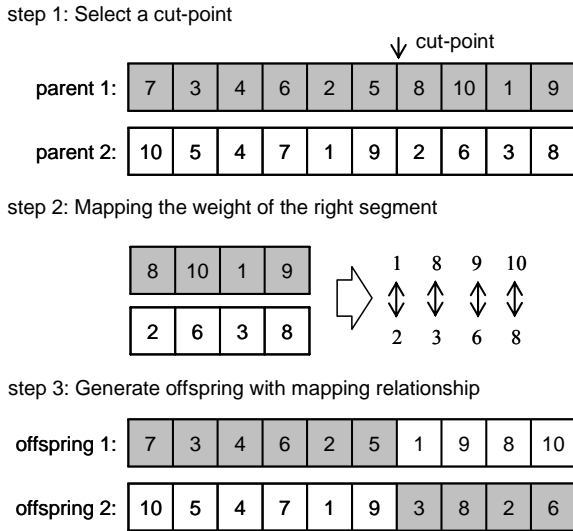


Fig. 3. Illustration of the WMX procedure.

2) Mutation Operators: In this paper, *Insertion Mutation* has been adopted. In this mutation, a gene is randomly selected and inserted a position, which is determined randomly. Fig. 4 shows an example of the Insertion Mutation procedure.

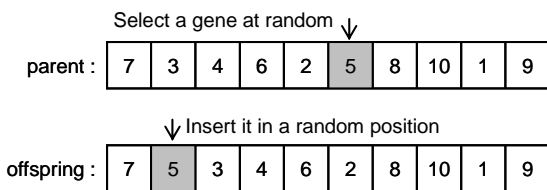


Fig. 4. Illustration of the Insertion Mutation procedure.

3) Immigration Operator: M. C. Moed *et al.* [12] proposed an immigration operator which, for certain types of functions, allows increased exploration while maintaining nearly the same level of exploitation for the given population size. The algorithm is modified considering immigration operator. In this operator, in each generation, $\mu \cdot popSize$ members are randomly generated, after evaluating them, they are replaced with $\mu \cdot popSize$ worst members of the population (μ is immigration rate).

4) Selection: the roulette wheel selection, a type of fitness-proportional selection, is adopted in this paper.

IV. EXPERIMENTS AND DISCUSSION

In this section, to show the effectiveness of priority-based encoding method that is able to find good results with respect to path optimality (quality of solution) and convergence speed, the proposed GA is compared with Ahn *et al.*'s [13] algorithm in solving the SPP. In addition, the effectiveness of AWA for solving several BNP is experimentally investigated. All experiments were realized using Java on Pentium 4 processor (1.5-GHz clock).

A. Comparison with Different Encoding Methods

1) Test Problems: For examining the effect of different encoding methods, we applied Ahn *et al.*'s method and priority-based encoding method on 6 test problems [13][14]. Dijkstra's algorithm have been used to obtain optimal solutions for the problems and the solution qualities of the proposed GA and Ahn *et al.*'s algorithm are investigated using optimal solution. Each algorithm was run 20 times using different initial seeds for each test problems. Two different stopping criteria are used. One of them is number of maximum generations. But, if the algorithm didn't improve the best solution in successive 100 runs, it is stopped to save computation time.

Population size: $popSize = 20$;
Crossover probability: $p_C = 0.70$;
Mutation probability: $p_M = 0.50$;
Immigration rate: $\mu = 3$;
Maximum generation: $max_gen = 1000$;
Terminating condition: 100 generations with same fitness.

2) Discussion of the Results: The results of each GA is given in Table 1. While in first three problems, proposed GA and Ahn's algorithm are reached to optimal solution, in second three test problems Ahn's algorithm couldn't reach optimum solution. For last four problems, because of the second stopping criterion, Ahn's algorithm is faster than the proposed algorithm. But its solution quality also decreases.

Table 1. Performance comparisons with Ahn's algorithm [13] and Proposed algorithm.

Test Problems (# of nodes/ # of arcs)	Optimal Solutions	Best Solutions		CPU Times (ms)		Generation Num. of Obtained best result	
		Prop. Alg.	Ahn's Alg.	Prop. Alg.	Ahn's Alg.	Prop. Alg.	Ahn's Alg.
20/49	142.00	142.00	142.00	23.37	40.60	9	2
80/120	389.00	389.00	389.00	96.80	118.50	4	4
80/632	291.00	291.00	291.00	118.50	109.50	10	19
160/2544	284.00	284.00	286.20	490.50	336.20	26	31
320/1845	394.00	394.00	403.40	1062.50	779.80	11	44
320/10208	288.00	288.00	288.90	1498.50	1028.30	26	38

Table 2. Comparison with the four approaches using the $D1_R$ measure.

Test Problems (# of nodes/ # of arcs)	$D1_R$ measure				CPU Times			
	AWA	RWA	SPEA	NSGA-II	AWA	RWA	SPEA	NSGA-II
25/49	143.58	191.21	315.61	228.65	14170	15122	17635	15693
25/56	141.43	203.96	224.40	185.89	14961	11918	16684	15981

Table 3. Comparison with the four approaches using the $R_{NDS}(S_j)$ measure.

Test Problems (# of nodes/ # of arcs)	$R_{NDS}(S_j)$				CPU Times			
	AWA	RWA	SPEA	NSGA-II	AWA	RWA	SPEA	NSGA-II
25/49	0.61	0.57	0.54	0.39	14170	15122	17635	15693
25/56	0.53	0.41	0.34	0.36	14961	11918	16684	15981

Table 4. Comparison with the four approaches using the $|S_j|$ measure.

Test Problems (# of nodes/ # of arcs)	$ S_j $				CPU Times			
	AWA	RWA	SPEA	NSGA-II	AWA	RWA	SPEA	NSGA-II
25/49	49	52	57	43	14170	15122	17635	15693
25/56	43	43	44	55	14961	11918	16684	15981

B. Experimental Result of Different Solution Approaches by Multiobjective GAs

1) Test Problems: The effect of different solution approaches on multiobjective GAs is investigated using SPEA[16], NSGA II [17], MOGLS[18] and AWA to the 2 test problems [15]. In this comparison, the following parameter specifications have been used.

Population size: $popSize = 20$;
Crossover probability: $p_C = 0.70$;
Mutation probability: $p_M = 0.50$;
Immigration rate: $\mu = 0.15$;
Stopping conditions: Evaluation of 5000 solutions.

2) Performance Measures: We mainly use a performance measure based on the distance from a reference solution set (*i.e.*, the Pareto-optimal solution set or a near Pareto-optimal solution set) for evaluation the solution set S_j . This measure was used in [18] and referred to as $D1_R$, the ratio of nondominated solutions $R_{NDS}(S_j)$, and the number of obtained solutions $|S_j|$. Let S^* be the reference solution set.

The $D1_R$ measure can be written as follows:

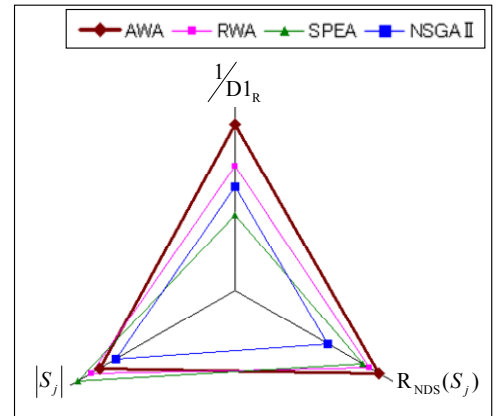
$$D1_R(S_j) = \frac{1}{|S^*|} \sum_{y \in S^*} \min \{d_{xy} | x \in S_j\}$$

where d_{xy} is the distance between a solution x and a reference solution y in the 2-objective space.

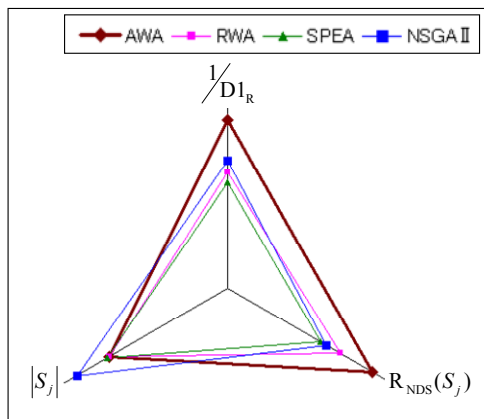
The $R_{NDS}(S_j)$ measure can be written as follows:

$$R_{NDS}(S_j) = \frac{|S_j - \{x \in S_j | \exists y \in S^*: y \prec x\}|}{|S_j|}$$

3) Discussion of the Results: The results of the four solution approaches are given in Table 2-4. While in first problem, AWA got the shortest distance $D1_R$, and also is faster than others. In second problem, RWA is faster than AWA, but its solution (distance $D1_R$) quality also decreases. AWA gives better performance than others by $R_{NDS}(S_j)$ measure. However, AWA did not effective method combine the number of obtained solutions $|S_j|$.



(a)



(b)

Fig. 5. Comparison with the four approaches using 2 test problems (a) Test problem 1 (25/49)
(b) Test problem 2 (25/56)

V. CONCLUSION

In this paper we presented a GA approach used a priority-based chromosome for solving the MXF/MCF problem. It is easy to verify that any permutation of the encoding corresponds to the paths, so that most existing genetic operators can easily be applied to the encoding. Also, any path has a corresponding encoding; therefore, any point in solution space is accessible for genetic search. This paper also combines an adaptive evaluation function based on the AWA. The fitness values of all individuals are calculated according to this adaptive evaluation function. In each generation, the set of Pareto solutions is updated by deleting all dominated solutions and adding all newly generated Pareto solutions.

Computer simulations show the several numerical experiments by using several network optimization problems, and show the effectiveness of the proposed method.

REFERENCES

- [1] A. Warburton: "Approximation of Pareto Optima in Multiple Objective Shortest Path Problems", *Operations Research*, No. 35, pp.70-79, 1987.
- [2] J.B. Smith & D. Shier: "An Empirical Investigation of Some Bicriterion Shortest Path Algorithms", *European J. of Opera. Research*, No.43, pp.216-224, 1989.
- [3] M. Gen & R. Cheng: *Genetic Algorithms and Engineering Design*. New York: John Wiley & Sons, 1997.
- [4] M. Gen & R. Cheng: "Evolutionary Network Design: Hybrid Genetic Algorithms Approach", *Inter. J. of Computational Intelligence and Applications*, Vol.3, No.4, pp.357-380, 2003.
- [5] M. Gen, G. Zhou & J. Kim: "Genetic Algorithms for Solving Network Design Problem: state-of-the-art-survey", *Evolutionary Optimization*, Vol.1, No.2, pp.121-141, 1999.
- [6] M. Gen & R. Cheng: *Genetic Algorithms and Engineering Optimization*, New York: John Wiley & Sons, 2000.
- [7] M. Gen, R. Cheng & S.S. Oren: "Network Design Techniques using Adapted Genetic Algorithms", *Advances in Engineering Software*, No.32, pp.731-744, 2001.
- [8] G. Zhou & M. Gen: "A Genetic Algorithm Approach on Tree-like Telecommunication Network Design Problem", *J. of Operational Research Society*, Vol.54, No.3, pp.248-254, 2003.
- [9] R. Cheng & M. Gen: "Resource Constrained Project Scheduling Problem using Genetic Algorithm", *Inter. J. of Intelligent Auto. and Soft Compu.*, No.3, pp.273-286, 1997.
- [10] M. Gen, R. Cheng & D. Wang: "Genetic Algorithms for Solving Shortest Path Problems", *Proc. of IEEE Int. Conf. on Evol. Comput., Indianapolis, Indiana*, pp.401-406, 1997.
- [11] C. Hsinghua, G. Premkumar & C. Chu: "Genetic Algorithms for Communications Network Design – An Empirical Study of the Factors that Influence Performance", *IEEE Trans. on Evol. Comput.*, Vol.5, No.3, pp.236-249, 2001.
- [12] C.M. Michael, C.V. Stewart & R. B. Kelly: "Reducing the Search Time of A Steady State Genetic Algorithm Using the Immigration Operator", *Proc. of IEEE Int. Conf. on Tools for AI San Jose, CA*, pp.500-501, 1991.
- [13] C.W. Ahn & R. Ramakrishna: "A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations", *IEEE Trans. on Evol. Comput.*, Vol.6, No.6, pp.566-579, 2002.
- [14] OR-Notes. [Online]: <http://mscmga.ms.ic.ac.uk/jeb/or/orweb.html>
- [15] T. Munakata, T. & D. J. Hashier: "A genetic algorithm applied to the maximum flow problem", *Proc. of the 5th Inter. Conf. on Genetic Algorithms, San Francisco*, pp.488-493, 1993.
- [16] E. Zitzler & L. Thiele: "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach", *IEEE Trans. on Evol. Comput.*, Vol.3, No.4, pp.257-271, 1999.
- [17] K. Deb, A. Pratap, S. Agarwal & T. Meyarivan: "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA- II ", *IEEE Trans. on Evol. Comput.*, Vol.6, No.2, pp.182-197, 2002.
- [18] H. Ishibuchi, T. Yoshida & T. Murata: "Balance Between Genetic Search and Local Search in Memetic Algorithms for Multiobjective Permutation Flowshop Scheduling", *IEEE Trans. on Evol. Comput.*, Vol.7, No.2, pp.204-223, 2003.
- [19] V. Chankong & Y.Y. Haimes: *Multiobjective Decision Making Theory and Methodology*, New York: North-Holland, 1983.
- [20] A. Warbuton: "Approximation of Pareto Optima in Multiple-objective, Shortest Path Problems", *Operations Research*, Vol.35, pp.70-79, 1987.
- [21] R. Hassin: "Approximation Schemes for the Restricted Shortest Path Problem", *Mathematics of Operations Research*, Vol.17, No.1, pp.36-42, 1992.
- [22] R. Ravi: "Rapid Rumor Ramification: Approximating the Minimum Broadcast Time", *Proc. of the 35th Annual IEEE Foundations of Computer Science*, pp.202-213, 1994.
- [23] J.L. Ganley, M.J. Golin & A. Bernstein: "Multicast Communication in Network Computers", *IEEE Software*, Vol.2, No.3, pp.49-61, 1985.
- [24] M.V. Marathe, R. Ravi, R. Sundaram, S. Ravi, D.J. Rosenkrantz & H. Hunt: "Bicriteria Network Design Problems", *J. of Algorithms*, No.28, pp.142-171, 1998.
- [25] Runwei Cheng & Mitsuo Gen: "Evolution Program for Resource Constrained Project Scheduling Problem", *Inter. Conf. on Evolutionary Computation*, pp.736-741, 1994