

Priority-based Genetic Algorithm for Bicriteria Network Optimization Problem

Mitsuo Gen* Lin Lin** Runwei Cheng***

* Graduate School of Information, Production & Systems, Waseda University
gen@waseda.jp

** Graduate School of Information, Production & Systems, Waseda University
lin@ruri.waseda.jp

*** JANA Solutions Co.
qiang@mb2.ksp.fujixerox.co.jp

Abstract - In recent years, several researchers have presented the extensive research reports on network optimization problems. In our real life applications, many important network problems are typically formulated as a Maximum Flow model (MXF) or a Minimum Cost Flow model (MCF). In this paper, we propose a Genetic Algorithm (GA) approach used a priority-based chromosome for solving the bicriteria network optimization problem including MXF and MCF models (MXF/MCF).

I. INTRODUCTION

The MXF problem is to find the maximum flow from a given source to a given sink. The MCF problem is also to find how to supply flows from the sources to the sinks with the minimum cost [1][3]. In this paper, the objective is to find the set of Pareto optimal solutions that give possible maximum flow with minimum cost. Priority-based chromosomes [2] have been used for encoding the network optimization problem. This paper also combines Adaptive Weight Approach (AWA) that utilizes some useful information from the current population to readjust weights for obtaining a search pressure toward a positive ideal point. It is based on the Gen & Cheng's Approach [4]. Computer simulations show the several numerical experiments by using several network optimization problems, and show the effectiveness of the proposed method.

II. MXF/MCF PROBLEM FORMULATION

In a network with flow capacities and costs on the arcs, the MXF/MCF problem is to determine the maximum possible flow z_1 with minimum cost z_2 from a given source to a given sink. An undirected graph $G=(V,A)$ comprises a set of nodes $V=\{1, 2, \dots, m\}$ and a set of arcs $A \in V \times V$ connecting nodes in V . Corresponding to each arc are two nonnegative numbers c_{ij} and u_{ij} representing the cost and flow capacity, and others of interest, from node i to node j . The MXF/MCF problem can be formulated as follows:

$$\begin{aligned} \max \quad & z_1 = f \\ \min \quad & z_2 = \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij} \end{aligned}$$

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

III. PRIORITY-BASED GENETIC ALGORITHM APPROACH

A. Genetic Representation

A gene in a chromosome is characterized by two factors: locus, the position of the gene within the structure of chromosome, and allele, the value of the gene takes. The position of a gene is used to represent a node, and the value is used to represent the priority of the node for constructing a path among candidates. The encoding method is denoted as *priority-based encoding* [2]. It is also 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.

procedure 1: priority-based encoding

input data: number of nodes m
output data: chromosome v_k
step 0: allele $v_k(\cdot) \leftarrow 0$, the priority value $p \leftarrow -1$.
step 1: If $p > m$, go to step 4; otherwise, continue.
step 2: $j \leftarrow \text{random}(1, m)$
step 3: If $v_k(j) = 0$, $v_k(j) \leftarrow p$, $p \leftarrow p+1$, return to step 1;
otherwise, return to step 2.
step 4: Output the chromosome v_k .

procedure 2: one path growth

input data: number of nodes m , chromosome v_k , the set of nodes S_i with all nodes adjacent to node i
output data: path P_k
step 0: the source node $i \leftarrow 1$, $P_k \leftarrow \phi$
step 1: If $S_i = \phi$, go to step 3; otherwise, continue.
step 2: Select l from the set of nodes S_i with the highest priority, $P_k \leftarrow P_k \cup \{x_{il}\}$, $i \leftarrow l$.
step 3: Output the complete path P_k

$$P_k = \{x_{1,l_1}, x_{1,l_2}, x_{2,l_3}, \dots, x_{l_{m-1},l_m}\}$$

procedure 3: overall paths growth

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

output data: 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 $S_l = \emptyset$, go to step 7; otherwise, $l \leftarrow l + 1$, continue.

step 2: The implementation of path P_l^k growth is based on **procedure 2**. Select the sink node a of path P_l^k .

step 3: If the sink node $a = m$, continue; otherwise, perform the set of nodes S_i update as follows, return to step 1.

$$s_i \leftarrow s_i - \{a\}, \quad \forall i$$

step 4: Calculate the flow f_i^k and the cost c_i^k of the path P_l^k .

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

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

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 .

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

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, $i \in L_k$.

B. Adaptive Evaluation Function

In order to evaluate the fitness of each individual in the GA approach for the MXF/MCF problem, we can design an adaptive evaluation function based on the AWA [4] (Gen-Cheng2000, pp.127-131). The fitness values of all individuals are calculated according to this adaptive evaluation function.

procedure 4: Adaptive Weight Approach

input data: chromosome v_k , $k \in pop_size$, number of paths L_k , the flow f_i^k and the cost c_i^k of each path, $i \in L_k$

output data: fitness value $eval(v_k)$, $k \in pop_size$

step 1: Define two extreme points: the maximum extreme point z^+ and the minimum extreme point z^- in criteria space as follows:

$$z^+ = \{z_1^{\max}, z_2^{\max}\} \quad z^- = \{z_1^{\min}, z_2^{\min}\}$$

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:

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

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

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

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

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

$$w_1 = \frac{1}{z_1^{\max} - z_1^{\min}}$$

$$w_2 = \frac{1}{z_2^{\max} - z_2^{\min}}$$

step 3: Calculate the fitness value for each individual.

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

C. Pareto Optimal Solutions

In the case of multiple objectives, there does not necessarily exist a solution that is best with respect to all objectives because of incommensurability and conflict among objectives. A solution may be best in one objective but worst in other objectives. Therefore, there usually exist a set of solutions for the multiple-objective case which cannot simply be compared with each other. For such solutions, called *Pareto optimal* solutions, no improvement in any objective function is possible without sacrificing at least one of the other objective functions. In this paper, Pareto solutions are identified at each generation.

procedure 5: identify Pareto solutions

input data: chromosome v_k , $k \in pop_size$, number of paths L_k , the flow f_i^k and the cost c_i^k of each path, $i \in L_k$

output data: number of Pareto solutions J , Pareto solution $e_j = \{v_j, p, z_j^1, z_j^2\}$, $e_j \in E$

step 0: initialize $i \leftarrow 0, j \leftarrow 0$.

step 1: $i \leftarrow i + 1$, if $i > L_k$, go to step 7; otherwise, continue.

step 2: $switch \leftarrow true, j \leftarrow j + 1$, if $j \leq J$, continue; otherwise, return to step 1.

step 3: If $f_i^k = z_j^1$ or $c_i^k = z_j^2$, let $switch \leftarrow false$, continue; otherwise, go to step 5.

step 4: If $c_i^k < z_j^2$ or $f_i^k > z_j^1$, update the set of Pareto solutions E and J , go to step 6.

$$E \leftarrow E \setminus e_j$$

$$J \leftarrow J - 1$$

$$switch \leftarrow true$$

step 5: If $f_i^k > z_j^1$ and $c_i^k < z_j^2$, update the set of Pareto solutions E and J , go to step 6; otherwise, return to step 2.

$$E \leftarrow E \setminus e_j$$

$$J \leftarrow J - 1$$

step 6: If $switch = true$, adding newly generated Pareto solution as follows, return to step 1.

$$J \leftarrow J + 1$$

$$e_j \leftarrow \{v_k(\cdot), i, f_i^k, c_i^k\}$$

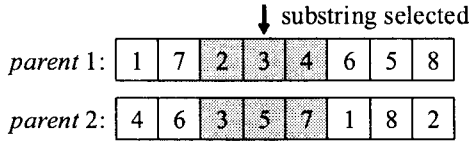
$$E \leftarrow E \cup e_j$$

step 7: Output number of Pareto solutions J , Pareto solution $e_j = \{v_j, p, z_j^1, z_j^2\} e_j \in E$

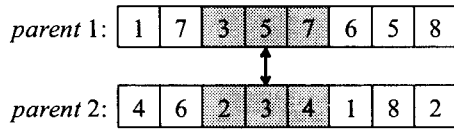
D. Genetic Operators

Crossover: Here the position-based crossover operator proposed by PMX (Partial Mapped Crossover) [2] (Gen-Cheng97, pp.119-125) was adopted. It can be viewed as an extension of two-point crossover for binary string to permutation representation. It uses a special repairing procedure to resolve the illegitimacy caused by the simple two-point crossover as shown in Figure 1.

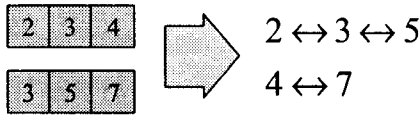
step 1: select the substring at random



step 2: exchange substrings between



step 3: determine mapping relationship



step 4: legalize offspring with mapping relationship

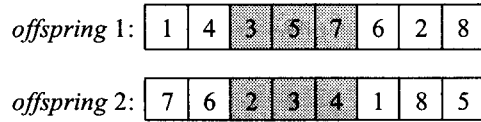


Figure 1: Illustration of the PMX operator.

Mutation: The swap mutation operator was used here, in which two positions are selected at random and their contents are swapped as shown in Figure 2.

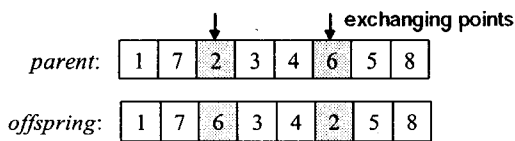


Figure 2: Illustration of the swap mutation operator.

Selection: The roulette wheel approach [2], a type of fitness-proportional selection, was adopted.

E. Overall Procedure for the priority-based GA

- step 0: Set genetic parameters and read in the data of a given instance.
- step 1: Generate the initial population based on procedure 1 (priority-based encoding).
- step 2: Decode the chromosomes into paths based on procedure 3 (overall paths growth), calculate the fitness values for each decoded individual based on procedure 4 (Adaptive Weight Approach).

step 3: Identify the set of Pareto solutions based on procedure 5 (identify Pareto solutions).

step 4: Produce offspring with crossover and mutation, decode the offspring and calculate the fitness values for each decoded offspring.

step 5: The current generation $gen \leftarrow gen+1$, update the set of Pareto solutions, select the next generation using the roulette wheel method.

step 6: If the maximal generation is reached $gen > genmax$, stop; otherwise, return to step 4.

IV. NUMERICAL EXAMPLES

A. Example 1

The first numerical example, presented by T. Munakata & D.J. Hashier [1], was adopted. The problem comprises 25 nodes and 49 arcs. It is given as shown in Figure 3.

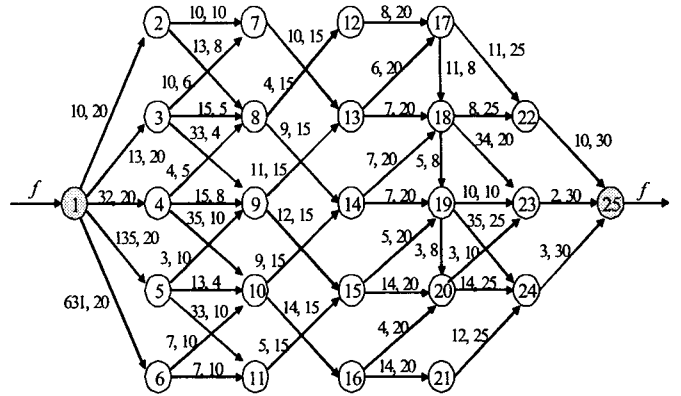


Figure 3: Illustration of Example 1

The parameters for the proposed GA approach are set as follows:

- Population size: $pop_size = 20$
- Crossover probability: $p_C = 0.20$
- Mutation probability: $p_M = 0.02$
- Maximum generation: $maxgen = 1000$

Table 1 gives out the all Pareto optimal solutions. Figures 4 give out the ideal points and all Pareto optimal solutions.

Table 1: The Pareto optimal solutions of Example 1

z_1	z_2	z_1	z_2	z_1	z_2
4	300	30	2470	72	7703
5	345	33	2786	73	8382
8	600	38	2926	75	9762
10	696	40	3046	78	11799
12	993	43	3274	80	13147
15	1001	47	3674	82	14531
18	1226	52	4074	85	17115
20	1568	56	4830	87	17941
21	1629	59	5406	88	19254
23	1833	66	6575	89	19333
28	2178	69	7145	90	20007

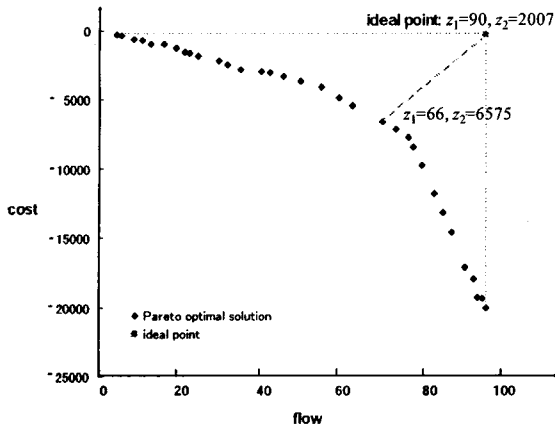


Figure 4: The Pareto optimal solutions of Example 1

B. Example 2

The second numerical Example, presented by T. Munakata & D.J. Hashier [1], was adopted. The problem comprises 25 nodes and 56 arcs. It is given as shown in Figure 5.

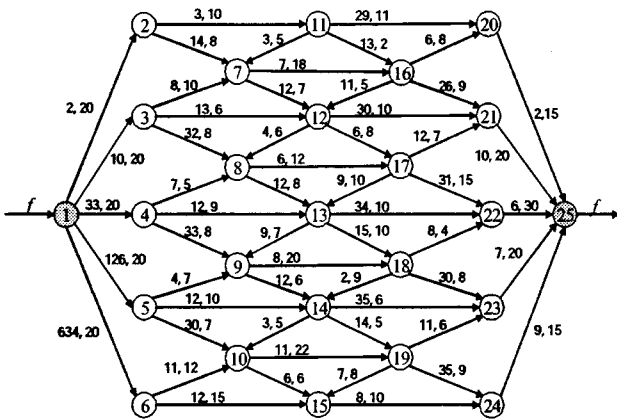


Figure 5: Illustration of Example 2

The parameters for the proposed GA approach are set as follows:

- Population size: $pop_size = 20$
- Crossover probability: $p_C = 0.20$
- Mutation probability: $p_M = 0.02$
- Maximum generation: $maxgen = 1000$

Table 2 gives out the all Pareto optimal solutions. Figures 6 give out the ideal points and all Pareto optimal solutions.

Table 2: The Pareto optimal solutions of Example 2

z_1	z_2	z_1	z_2	z_1	z_2	z_1	z_2
2	52	32	1633	54	3872	73	6944
8	248	34	1909	55	3990	74	7192
10	340	36	1937	58	4146	75	7402
15	495	38	2077	61	4671	76	7532
18	692	40	2485	63	5153	78	7847
19	1012	41	2581	65	5463	80	9228
20	1111	43	2731	66	5704	82	10395
21	1220	47	3080	67	6323	83	12508
25	1292	49	3302	68	6422	85	12610
26	1406	51	3551	71	6537	86	13151
27	1457	52	3739	72	6748	91	16752
28	1475						

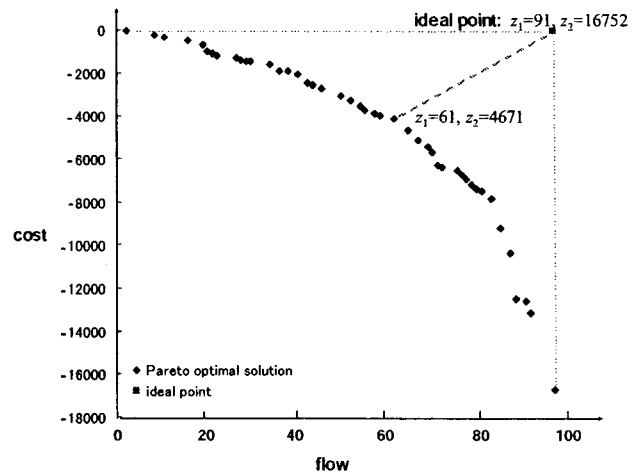


Figure 6: The Pareto optimal solutions of Example 2

V. CONCLUSION

In this paper presented, a GA approach used a priority-based chromosome for solving the MXF/MCF problem. Priority-based chromosome is also 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] T. Munakata & D.J. Hashier: "A Genetic Algorithm Applied to the Maximum Flow Problem", *Proc. of 5th International Conference on Genetic Algorithms*, pp.488-493, 1993.
- [2] M. Gen & R. Cheng: *Genetic Algorithms and Engineering Design*, John Wiley & Sons, New York, 1997.
- [3] M. Gen & K. Ida: *Neural Networks and Optimization with Mathematica*, Kyoritsu Shuppan, 1998 in Japanese.
- [4] M. Gen & R. Cheng: *Genetic Algorithms and Engineering Optimization*, John Wiley & Sons, New York, 2000.
- [5] M. Gen, R. Cheng & S.S. Oren: "Network design techniques using adapted genetic algorithms", *Advances in Engineering Software*, Vol.32, pp.731-744, 2001.
- [6] C.W. Ahn & R. S. Ramakrishna: "A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations", *IEEE Trans. Evol. Comput.*, Vol. 6, No. 6, pp.566-579, 2002.