WEIGHTED SUM-BASED GENETIC ALGORITHM FOR BICRITERIA NETWORK DESIGN PROBLEM

Mitsuo Gen and Lin Lin

Graduate School of Information Production and System
Waseda University, Japan

gen@waseda.jp

ABSTRACT
This paper proposes a new Multiobjective Genetic Algorithm (MOGA) approach for Bicriteria Network Design (BND) Problem. 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 BND problem.

Key Words: Multiobjective genetic algorithm, priority-based encoding, Pareto optimal solutions, shortest path problem, bicriteria network design problem.

1. 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 Dijsktra (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), 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 (BND) problem 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.

2. BICRITERIA NETWORK DESIGN (BND) PROBLEM

The bicriteria shortest path problem is one of BND problems, 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 (1987) presented the first fully polynomial approximation scheme (FPAS) for it. Hassin (1992) provided a strongly polynomial FPAS for the problem which improved the running time of Warburton (1987). Gen and Cheng (1997) presented a compromise approach-based genetic algorithm for the problem.

The general classes of BND problems with minimum two objectives (under different cost functions) are defined and extended to the more multi-criteria network design problems. Ravi et al. (1994) presented an approximation algorithm for finding good broadcast networks. Ganley et al. (1985) consider a more general problem with more than two objective functions. Marathe et al. (1998) consider three different criteria of network and presented the first polynomial-time approximation algorithms for a large class of BND problem.

In this paper, we study the complexity case of BND problems. 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 BND problem 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 $(i, j) \in A$ 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 BND 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 $(i, j)\in A$. BND is a multiobjective optimization model formulated as follows:
max \quad z_1 = f \\
min \quad z_2 = \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} \\
\text{s. t.} \quad \sum_{j=1}^{m} x_{ij} - \sum_{k=1}^{m} x_{ki} = \begin{cases} 
\quad f \quad (i = 1) \\
\quad 0 \quad (i = 2, 3, \ldots, n - 1) \\
\quad -f \quad (i = n) 
\end{cases} \\
0 \leq x_{ij} \leq u_{ij} \quad , \quad \forall (i, j) \in A \\
f \geq 0

3. NEW APPROACH OF MULTIOBJECTIVE GA

In this section, we present the proposed approach of multiobjective GA, which uses priority-based encoding method (in Section 3.1), show the paths growth procedure (in Section 3.2) that can find various paths by one chromosome for solving BNP. We present a new crossover operators (in Section 3.4). Also combines adaptive weights approach (AWA) (in Section 3.3).

3.1. Genetic Representation

How to encode a solution of the problem into a chromosome is a key issue for GAs. For any application case, it is necessary to perform analysis carefully to ensure an appropriate representation of solutions together with meaningful and problem-specific genetic operators (Gen & Cheng 1997). One of the basic features of GAs is that they work on coding space and solution space alternatively: genetic operations work on coding space (chromosomes), while evaluation and selection work on solution space. For the nonstring coding approach, three critical issues emerged concerning with the encoding and decoding between chromosomes and solutions (or the mapping between phenotype and genotype): (1) The feasibility of a chromosome; (2) The legality of a chromosome; (3) The uniqueness of mapping.

Feasibility refers to the phenomenon of whether a solution decoded from a chromosome lies in the feasible region of a given problem. Legality refers to the phenomenon of whether a chromosome represents a solution to a given problem. The illegality of chromosomes originates from the nature of encoding techniques. For many combinatorial optimization problems, problem-specific encodings are used and such encodings usually yield to illegal offspring by a simple one-cut-point crossover operation. Because an illegal chromosome cannot be decoded to a solution, it means that such chromosomes cannot be evaluated. Repairing techniques are usually adopted to convert an illegal chromosome to a legal one. The mapping from chromosomes to solutions (decoding) may belong to one of the following three cases: 1-to-1 mapping, n-to-1 mapping and 1-to-n mapping. The 1-to-1 mapping is the best one among three cases and 1-to-n mapping is the most undesired one. We need to consider these problems carefully when designing a new non-binary-string coding so as to build an effective GA.

![Fig. 1. A simple undirected graph with 7 nodes and 12 edges](image-url)
Cheng and Gen (1997) proposed a priority-based encoding method firstly for solving resource-constrained project scheduling problem (rPSP) and they also adopted this method for solving SPP in Gen et al. (1997). The priority-based encoding method is an indirect approach: encode some guiding information for constructing a path, but a path itself, in a chromosome. As it is known, a gene in a chromosome is characterized by two factors: locus, i.e., the position of gene located within the structure of chromosome, and allele, i.e., the value the gene takes. In this encoding method, the position of a gene is used to represent node ID and its value is used to represent the priority of the node for constructing a path among candidates. A path can be uniquely determined from this encoding.

**Example:** An example of generated chromosome and its decoded path as shown in Fig. 2, for the undirected graph shown in Fig. 1.

![Figure 2. Example of generated chromosome and its decoded path](image)

**Advantage:** Any permutation of the encoding corresponds to a path (legality); most existing genetic operators can be easily applied to the encoding. Any path has a corresponding encoding (completeness); any point in solution space is accessible for genetic search.

**Disadvantage:** At some case, n-to-1 mapping may occur for the encoding.

### 3.2. Population Initialization

**Genetic Representation:** As is described in Section 3.1, priority-based encoding method is adopted in this paper. Pseudocode for priority-based encoding is given in procedure 1. Where $n$ is number of nodes in the network, $v_k$ is $k$th chromosome in the initial population.

**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 (Gen et al. 1999). Therefore, random initialization is adopted in this paper.

**procedure 1:** priority-based encoding

**input:** number of nodes $n$

**output:** chromosome $v_k$

**step 0:** for $j = 1$ to $n$

\[ v_k(j) \leftarrow j; \]

\[ \left\lceil \frac{n}{2} \right\rceil \]

**step 1:** for $i = 1$ to $n$

\[ j \leftarrow \text{random}(1,n); \]

\[ l \leftarrow \text{random}(1,n); \]

if $l \neq j$ then

\[ \text{swap}(v_k(j), v_k(l)); \]

**step 2:** Output the chromosome $v_k$. 
3.3. Decoding Method

To describe this decoding method, we first define a one-path growth procedure that decodes one path based on the generated chromosome with given network; and then present an overall-path growth procedure that obtains overall possible paths for the given chromosome.

3.3.1. One-path growth procedure

The path is generated by One-path growth procedure that is given in procedure 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.

**procedure 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 \( i \leftarrow 1 \), \( P_k \leftarrow \phi \)

**step 1:** if \( S_i = \phi \), 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 \( P_k \).

\( P_k = \{x_{i1}, x_{i2}, x_{i3}, \ldots, x_{i+ld} \} \)

3.3.2. Overall-path growth procedure

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 \( u_{ij} \). With the one-path growth procedure, we can obtain the second path. By repeating this procedure we can obtain the maximum flow for the given chromosome till no new network can be defined in this way.

3.4. 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 (Gen & Cheng 1997). 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 procedure 3.

**procedure 3:** Adaptive Weight Approach

**input:** chromosome \( v_k \), \( k \in \text{pop}_\text{size} \), the flow \( f_i \) and the cost \( c_i \) of each \( v_k \)

**output:** fitness value \( \text{eval}(v_k) \), \( k \in \text{pop}_\text{size} \)

**step 1:** Define two extreme points: the maximum extreme point \( z^+ \) and the minimum extreme point \( z^- \) in criteria space as \( z^- = \{z_1^{\min}, z_2^{\min}\} \) and \( z^+ = \{z_1^{\max}, z_2^{\max}\} \). Where \( z_1^{\max}, z_2^{\max}, z_1^{\min}, z_2^{\min} \) are the maximal value and minimal value for objective 1 and objective 2 in the current population. They are defined as follows:
\n\[z_{i}^{\max} = \max \{f_{i}^{k} | i \in L_{k}, k \in \text{pop\_size}\},\]
\[z_{i}^{\min} = \min \{-c_{i}^{k} | i \in L_{k}, k \in \text{pop\_size}\},\]

**Step 2:** The adaptive weight for objective 1 and objective 2 are calculated by the following equation:
\[
w_{1} = \frac{1}{z_{i}^{\max} - z_{i}^{\min}}, \quad w_{2} = \frac{1}{z_{2}^{\max} - z_{2}^{\min}}
\]

**Step 3:** Calculate the fitness value for each individual.
\[
eval(v_{k}) = \frac{\sum_{i=1}^{L_{k}} (w_{1}(f_{i}^{k} - z_{i}^{\min}) - w_{2}(c_{i}^{k} + z_{i}^{\min}))}{L_{k}}, \quad \forall k \in \text{pop\_size}
\]

### 3.5. 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 (Hsinghua et al. 2001). Therefore it is important to examined different genetic operators.

#### 3.5.1. 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 crossover procedure.

#### 3.5.2. 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.

#### 3.5.3. Immigration Operator

Michael et. al. (1991) 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, \(\mu\cdot\text{pop\_size}\) members are randomly generated, after evaluating them, they are replaced with \(\mu\cdot\text{pop\_size}\) worst members of the population (\(\mu\) is immigration rate).

#### 3.5.4. Selection

The roulette wheel selection, a type of fitness-proportional selection, is adopted in this paper.
4. 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 (2002) 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).

4.1. Comparison with Different Encoding Methods

4.1.1. 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 (Ahn et al. 2002) (OR-Notes, Online). Dijskstra'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: \( \text{pop\_size} = 20 \);

Crossover probability: \( p_c = 0.70 \);

Mutation probability: \( p_m = 0.50 \);

Immigration rate: \( \mu = 3 \);

Maximum generation: \( \text{max\_gen} = 1000 \);

Terminating condition: 100 generations with same fitness.

### 4.1.2. 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.
4.2. Experimental Result of Different Solution Approaches by Multiobjective GAs

4.2.1. Test Problems

The effect of different solution approaches on multiobjective GAs is investigated using SPEA (Zitzler & Thiele 1999), NSGA II (Deb et al. 2002), MOGLS (Ishibuchi et al. 2003) and AWA to the 2 test problems in Munakata & Hashier (1993). In this comparison, the following parameter specifications have been used.

- Population size: \( pop\_size = 20 \);
- Crossover probability: \( p_C = 0.70 \);
- Mutation probability: \( p_M = 0.50 \);
- Immigration rate: \( \mu = 3 \);
- Stopping conditions: Evaluation of 5000 solutions.

4.2.2. 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 evaluating the solution set \( S_j \). This measure was used in Ishibuchi et al. (2003) and referred to as \( D_{1R} \), 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 \( D_{1R} \) measure can be written as follows:

\[
D_{1R}(S_j) = \frac{1}{|S^*|} \sum_{y \in S^*} \min\{d_{xy}| x \in S_j \}
\]

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:

\[
R_{NDS}(S_j) = \frac{|S_j - \{x \in S_j | \exists y \in S^*: y \preceq x\}|}{|S|}.
\]

![Fig. 4. Comparison with the four approaches using 2 test problems](image)

(a) Test problem 1 (25/49)
(b) Test problem 2 (25/56)
5. 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 $D_{1R}$, and also is faster than others. In second problem, RWA is faster than AWA, but its solution (distance $D_{1R}$) quality also decreases. AWA gives better performance than others by $R_{NDS}(S)$ measure. However, AWA did not effective method combine the number of obtained solutions $|S_j|$

REFERENCES


OR-Notes. [Online]: http://mscmga.ms.ic.ac.uk/jeb/or/orweb.html


