Genetic Algorithm Approach on Multi-criteria Minimum Spanning Tree Problem

Gengui Zhou and **Mitsuo Gen***
Department of Industrial and Systems Engineering
Ashikaga Institute of Technology, Ashikaga 326, Japan
E-mail: {zhou, gen}@genlab.ashitech.ac.jp

Abstract

Abstract: Minimum Spanning Tree (MST) problem is of high importance in network optimization. The multi-criteria MST (mc-MST) is a more realistic representation of the practical problem in the real-world, but it is difficult for the traditional network optimization technique to deal with. In this paper, a genetic algorithm (GA) approach is developed to deal with this problem. Without neglecting its network topology, the proposed method adopts the Prüfer number as the tree encoding and applies the Multiple Criteria Decision Making (MCDM) technique and nondominated sorting technique to make the GA search give out all Pareto optimal solutions either focused on the region near the ideal point or distributed all along the Pareto frontier. Compared with the enumeration method of Pareto optimal solution, the numerical analysis shows the efficiency and effectiveness of the GA approach on the mc-MST problem.

Keywords: Genetic Algorithms, Minimum Spanning Tree, Multiple Criteria Decision Making.

1 Introduction

The Minimum Spanning Tree (MST) problem is to find a least cost spanning tree in an edge weighted graph. This problem has been well studied and many efficient polynomial-time algorithms have been developed by Dijkstra [6], Kruskal [13], Prim [16] and Sollin [20]. More recently, some extended formulations of the MST problem have received great attention due to some practical demands, such as the degree-constrained MST by Narula and Ho (1980) [15], the stochastic MST by Ishii, Shiode and Nishida (1981) [12], the probabilistic MST by Bertsimas (1990) [3] and the quadratic MST by Xu (1995) [23]. As to these extended MST problems, Varieties of heuristics including genetic algorithms (GAs) methods have been used. The GA approaches, in particular, have shown to be highly effective in solving the probabilistic MST [1], the degree-constrained MST [24][25] and the quadratic MST [26]. However, the MST problem with multi-criteria has seldom received attention in network optimization.

In the real-world, there are usually such cases that one has to consider simultaneously multi-criteria in determining a MST because there are multiple attributes defined on each

*To whom correspondence would be addressed.

edge. The problem may arise, for instance, when designing a layout for telecommunication system, besides the cost for connection between cities or terminals, other factors such as the time for communication or construction, the complexity for construction and even the reliability are all important and have to be taken into consideration. In all these cases, the MST with multi-criteria is a more realistic representation of the practical problem.

The multi-criteria Minimum Spanning Tree (mc-MST) problem is not simply an extension of MST from single objective to multiple objectives. In general, we can not get the optimal solution of the problem because these multiple objectives usually conflict with each other in practice. The real solutions to this problem are a set of Pareto optimal solutions[5] but the calculation of it is a difficult task because it is a NP-hard problem and no previous work in this area has been reported in literature. A common solution to this problem is to simplify multiple objectives into a single objective because those polynomial-time MST algorithms are only effective to the MST with single objective. But only one single-point solution in the sense of Pareto optimality can be obtained, not to mention the difficulty in properly transforming the multiple objectives into a single objective. The decision maker in practice, however, may prefer one (Pareto optimal) point over the others depending on the situation. Consequently, it may be useful to have (all) other possible Pareto optimal solutions.

As one of the Evolutionary Computation (EC) techniques, the GAs have been receiving great attention and successfully applied in many research field in the last decade [7] [9] [14], including the effective approaches on the multiobjective optimization problems [8][18][19][21]. In this paper we propose a Genetic Algorithm (GA) approach to deal with the mc-MST problem. In keeping its network characteristic, we adopt the Prüfer number as the tree encoding[17]. This encoding is capable of equally and uniquely representing all possible spanning trees for a graph and easily operated by GA approach. The proposed GA approach can easily obtain a set of Pareto optimal solutions instead of only one for decision maker to choose. Combined with the multiple criteria decision making (MCDM) technique [5], the proposed method can enforce the GA approach to give out the Pareto optimal solutions close to the *ideal point* as much as possible. The ideal point is a solution satisfying all objectives but not obtainable. Combined with the nondominated sorting technique[21], the proposed method can make the GA approach maintain all Pareto optimal solutions distributed along the Pareto frontier, which gives out enough alternatives to reflect the achievements of each objective among multiple objectives. In order to verify the effectiveness of the proposed GA approach, we present an exact algorithm to give out all Pareto optimal solutions for the mc-MST problem. The numerical analysis shows that most of the results got by the proposed GA approach are real Pareto optimal solution. Therefore, the proposed method in this paper is really effective to deal with the mc-MST problem and capable of providing enough alternatives for decision maker to choose.

The paper is organized as follows: In Section 2 the mc-MST problem is defined. Some techniques in MCDM are described in Section 3. Section 4 discusses in detail our GA approach for the mc-MST problem. In Section 5 the numerical experiments on the proposed method are studied and the conclusion follows in Section 6.

2 Problem Description

Consider a connected undirected graph $G = (V, E)$, where $V = \{v_1, v_2, \cdots, v_n\}$ is a finite set of *vertices* representing terminals or telecommunication stations *etc.*, and $E = \{e_1, e_2, \cdots, e_m\}$

is a finite set of *edges* representing connections between these terminals or stations. Each edge has p associated positive real numbers, representing p attributes defined on it and denoted with $w_i = (w_{1i}, w_{2i}, \cdots, w_{pi})$ $(i = 1, 2, \cdots, m)$. In practice $w_{ki}(k = 1, 2, \cdots, p)$ may represent the distance, cost and so on.

Let $\boldsymbol{x} = (x_1 x_2 \cdots x_m)$ be defined as follows:

$$x_i = \begin{cases} 1, & \text{if edge } e_i \text{ is selected} \\ 0, & \text{otherwise} \end{cases}$$

Then a spanning tree of graph G can be expressed by the vector $\boldsymbol{x}$. Let X be the set of all such vectors corresponding to spanning trees in graph G, the mc-MST problem can be formulated as:

$$\begin{array}{lcl} \min\ z_1(\boldsymbol{x}) & = & \sum_{i=1}^{m} w_{1i} x_i \\ \min\ z_2(\boldsymbol{x}) & = & \sum_{i=1}^{m} w_{2i} x_i \\ \quad\cdots & & \\ \min\ z_p(\boldsymbol{x}) & = & \sum_{i=1}^{m} w_{pi} x_i \\ \text{s. t.} \qquad \boldsymbol{x} & \in & X \end{array} \tag{1}$$

where $z_i(\boldsymbol{x})$ is the ith objective to be minimized for the problem.

Compared with the traditional MST problem, the problem (1) only differs in the number of objectives. However, it is because these multiple objectives exist and usually conflict with each other, we can not determine which edge has the least weights and span one by one to form a spanning tree like the process of vertex growth or edge growth. If we transform the multiple objectives into a single objective according to some criteria, we really can easily solve the problem by using any MST algorithm but only one single solution in the sense of Pareto optimality can be obtained. Actually, this transformation is not a easy work both for the decision maker and the system analyzer in practice.

On the other hand, the problem (1) can be transformed into a multiple objective integer programming and then we can make use of the techniques in multiple objective integer programming to cope with[11]. But with the increase of the problem scale (vertices), it is still difficult to deal with. Moreover, the transformation neglects the topological characteristic of a tree structure in network and makes the problem even more complicated. Therefore, it is necessary to develop a new approach to cope with this multi-criteria problem while not neglecting its tree structure and give out enough alternatives for the decision maker to choose. In this paper the GA is adopted as the new approach for this problem because the GA can handle any kind of objective functions and any kind of constraints as a evolutionary computation method. Before discussing the GA approach on the mc-MST problem, some techniques in multiple criteria decision making have to be discussed.

3 Multiple Criteria Decision Making

A basic understanding of MCDM can be illustrated by the following definitions [5].

Definition 1: Given a set of feasible solution $S = \{\boldsymbol{x} \mid \boldsymbol{x} \in X\}$, solution $\boldsymbol{x}^*$ is denoted as the **dominating solution** for the problem (1) if and only if all other solution $\boldsymbol{x} \in S$, the following conditions hold:

$$z_k(\boldsymbol{x}) \geq z_k(\boldsymbol{x}^*), \qquad k = 1, 2, \cdots, p$$

In most cases of MCDM, such a dominating solution does not exist because these multiple objectives usually conflict with each other and no one can dominate others. However, if we only consider each objective respectively, we can figure out an ideal optimal solution.

Definition 2: As to the problem (1), the point $\boldsymbol{z}^\circ(\boldsymbol{x}) = (z_1^\circ(\boldsymbol{x}), z_2^\circ(\boldsymbol{x}), \cdots, z_p^\circ(\boldsymbol{x}))$ in criteria space is denoted as the **ideal point**, where

$$z_k^\circ(\boldsymbol{x}) = \min_{\boldsymbol{x} \in X} z_k(\boldsymbol{x}), \qquad k = 1, 2, \ldots, p$$

In the MST with single objective, the ideal point is the dominated solution or optimal solution. But in the mc-MST problem, actually the ideal point does not exist because we can not obtain the solution corresponding to the ideal point. For the multi-criteria problem, people usually adopt the concept of nondominated solution or Pareto optimal solution to define its solution.

Definition 3: Given a set of feasible solution $S = \{\boldsymbol{x} \mid \boldsymbol{x} \in X\}$, solution $\boldsymbol{x}'$ is denoted as the **nondominated solution** or **Pareto optimal solution** for the problem (1) if and only if there is no any other solution $\boldsymbol{x} \in S$, satisfying the following conditions:

$$\begin{array}{ll} z_q(\boldsymbol{x}) < z_q(\boldsymbol{x}'), & \text{for some } \ q \in \{1, 2, \cdots, p\} \\ z_k(\boldsymbol{x}) \leq z_k(\boldsymbol{x}'), & \text{for all } \ k \neq q, k = 1, 2, \ldots, p \end{array}$$

It is obviously that for the mc-MST problem there are usually a set of Pareto optimal solutions instead of one. All these Pareto optimal solutions can form a Pareto frontier.

3.1 Method of Objective Weighting

In MCDM, the common approach is to transform multiple objectives into a single objective according to some criteria and then solve the single objective problem. Generally different criteria result in different MCDM methods but simpler method is to aggregate multiple objectives with the help of a set of weighting coefficients which can represent the decision maker's preference on each objective or the attainment degree of all objectives in optimization. The following method is one of such methods.

As to the problem (1), in optimizing each objective: $\min_{\boldsymbol{x} \in X} z_k(\boldsymbol{x}), k = 1, 2, \cdots, p$, there are respectively p optimal solution $\boldsymbol{x}^j, j = 1, 2, \cdots, p$, and $p \times p$ corresponding objective values:

$$z_{jk} = z_k(\boldsymbol{x}^j), \qquad k = 1, 2, \cdots, p, \quad j = 1, 2, \cdots, p$$

In criteria space, through these p points $\boldsymbol{z}_k = (z_{1k}, z_{2k}, \cdots, z_{pk})$ $(k = 1, 2, \cdots, p)$, there is a hyperplane which satisfies:

$$\sum \lambda_k \ \boldsymbol{z}_k = \alpha$$

where α *and* λ_k $(k = 1, 2, \cdots, p)$ are the solution of the following equations:

$$\begin{array}{lcl} \sum_{k=1}^{p} \lambda_k & = & 1 \\ \sum_{k=1}^{p} z_{jk} \ \lambda_k - \alpha & = & 0, \quad j = 1, 2, \cdots, p \end{array} \tag{2}$$

The $p+1$ equations (2) have $(p+1)$ variables. It is easy to verify that there is an unique solution if there does not exist dominated solution for the problem (1). With the help of this set of weighting coefficients $\lambda_k \quad (k=1,2,\cdots,p)$, the problem (1) can be transformed into the following problem with single objective:

$$\begin{array}{lrcl} \min & z(\boldsymbol{x}) & = & \sum_{k=1}^{p} \lambda_k z_k(\boldsymbol{x}) \\ \text{s. t.} & \boldsymbol{x} & \in & X \end{array} \tag{3}$$

Obviously, we can only get one Pareto optimal solution in this way. If we really need other Pareto optimal solutions for decision making, we have to change the set of weighting coefficients but it is a hard task and even impossible in most cases.

3.2 Method of Pareto Optimal Enumeration

Simulating the process of edge growth in obtaining a spanning tree, we can enumerate all such spanning trees as the non-dominated solutions of the mc-MST problem. In the process of edge growth, we can not determine which edge has the least weight, but we can determine which subtree is dominated or not if one edge is added into that subtree. Therefore, once an edge is spanned into a subtree, the set of nondominated subtree is updated. Finally we can obtain the set of all nondominated solutions until all subtrees have $n-1$ edges.

Two main factors should be taken into consideration for enumerating all Pareto optimal solutions: (1) for any subtree in the set of nondominated subtrees, all adjacent edges are checked but only one edge is spanned into the subtree in each step, and (2) because each subtree will result in more than one nondominated subtrees, all those nondominated subtrees generated from different subtrees with the same number of edges have to be checked again in the sense of Pareto optimality and then the set of nondominated subtree has to be updated.

Supposed that $T_r^{(t)}$ is the rth subtree containing t edges, $r \in I^{(t)}$ which is the index set representing the number of such subtrees, $W_k(T_r^{(t)})$ is the total weights of the subtree $T_r^{(t)}$ in the kth objective, and $E(T_r^{(t)})$ is the set of all those edges adjacent to the subtree $T_r^{(t)}$.

procedure: enumeration

step 1: Set $t \leftarrow 0$ and $I^{(t)} \leftarrow \{1\}$. Select any one vertex in V, say v_1, let $T_1^{(t)} \leftarrow \{v_1\}$ and $W_k(T_1^{(t)}) \leftarrow 0, k=1,2,\ldots,p$.

step 2: Set $r \leftarrow 1$, $s \leftarrow 0$ and $t \leftarrow t+1$.

step 3: If $r \in I^{(t-1)}$, continue; otherwise goto **step 5**.

step 4: Determine e_l which satisfies that $e_l \in E(T_r^{(t-1)})$ and no any other edge $e_i \in E(T_r^{(t-1)})$ holds the following condition:

$$\begin{array}{ll} w_{qi} + W_q(T_r^{(t-1)}) < w_{ql} + W_q(T_r^{(t-1)}), & \text{for some } \ q \in \{1,2,\cdots,p\} \\ w_{ki} + W_k(T_r^{(t-1)}) \leq w_{ki} + W_k(T_r^{(t-1)}), & \text{for all } \ k \neq q, k=1,2,\ldots,p \end{array}$$

If there exists such edge e_l but adding edge e_l results in a cycle with the edges in $T_r^{(t-1)}$, set $E(T_r^{(t-1)}) \leftarrow E(T_r^{(t-1)}) \backslash \{e_l\}$ and return to **step 4**; If there exists such edge e_l and adding edge e_l results in no cycle with the edges in $T_r^{(t-1)}$, let $s \leftarrow s+1$, $T_s^{(t)} \leftarrow T_r^{(t-1)} \bigcup \{e_l\}$, $W_k(T_s^{(t)}) \leftarrow W_k(T_r^{(t-1)}) + w_{kl}, k=1,2,\ldots,p$, and add s into $I^{(t)}$, then goto **step 4**; If there exits no such edge e_l, let $r \leftarrow r+1$ and goto **step 3**.

step 5: Compare with all subtrees $T_s^{(t)}, s \in I^{(t)}$ in the sense of Pareto optimality, delete those dominated subtrees and compact the index set of $I^{(t)}$.

step 6: If $t < n-2$, goto **step 2**; otherwise stop.

When all those subtrees $T_s^{(t)}, s \in I^{(t)}$, contain $n-1$ edges, we get the final Pareto optimal solutions for the mc-MST problem.

Although this method can help us to obtain all Pareto optimal solutions, obviously it is also impossible for enumeration in all cases. It has been shown [4] that the number of possible trees in a complete graph on n vertices is $n^{(n-2)}$. Therefore, the Pareto optimal set may be very large (possible exponential in size) and computational and storage requirements tend to increase rapidly with the increase in problem size and number of objective functions. However, in this paper we present this algorithm just for the verification of the results obtained by the GA approach.

4 GA Approach

Genetic algorithms manipulate a population of potential solutions to an optimization (or search) problem. Specially they operate on encoded representations of the solutions, equivalent to the genetic material of individuals in nature, and not directly on the solutions themselves. Moreover, they improve all potential solutions step by step through biological evolutionary process like crossover and mutation *etc.* Therefore, we need not worry about how to form a solution of the mc-MST problem like the process of vertex growth or edge growth in the conventional MST algorithms. As to the mc-MST problem, it is also possible to give out a set of candidate solutions or Pareto optimal solutions instead of only one. Because the whole process in the GA approach is not wholly operated randomly but includes both directed and stochastic search embedded with a survival of the fittest mechanism, it is possible to enforce the search to reach the optimal solution in most cases. In this section, we focus on the design of the GA approach for the mc-MST problem.

4.1 Chromosome Representation

For the GA approach on any problem, it is important to figure out the adequate chromosome representation of the problem. One of the classical theorems in graphical enumeration is Cayley's theorem[4]. It states that there are $n^{(n-2)}$ distinct labeled trees for a complete graph with n vertices. Prüfer provided a constructive proof of Cayley's theorem by establishing a one-to-one correspondence between such trees and the set of all permutations of $n-2$ digits[17]. This means that we can use only a permutation of $n-2$ digits in order to uniquely represent a tree with n vertices where each digit is an integer between 1 and n inclusive. This permutation is usually known as Prüfer number.

For any tree in a complete graph, there are always at least two leaf vertices. By the leaf vertex is meant that there is only one edge connected on it. Based on this observation we can easily construct the Prüfer number according to the following procedure. Here we build the permutation by appending digits to the right, and thus the permutation is built and read from left to right.

procedure: encoding

step 1: Let vertex j be the smallest labeled leaf vertex in a labeled tree T.
step 2: Set k to be the first digit in the permutation if vertex k is incident to vertex j.
step 3: Remove vertex j and the edge from j to k, we have a tree with $n-1$ vertices.
step 4: Repeat above steps until one edge is left and produce the Prüfer number or permutation with $n-2$ digits in order.

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 $\overline{P}$ be the set of all vertices not included in P.
step 2: Let j be the vertex with the smallest label in $\overline{P}$, and let k be the leftmost digit of P. Add the edge from j to k into the tree. Remove j from $\overline{P}$ and k from P. If k does not occur anywhere in the remainder of P, put it into $\overline{P}$. Repeat the process until no digits left in P.
step 3: If no digits remain in P, there are exactly two vertices, r and s, in $\overline{P}$. Add edge from r to s into the tree and form a tree with $n-1$ edges.

An example is given to illustrate this encoding. The Prüfer number [2 5 6 8 2 5] corresponds to a spanning tree on a 8-vertex complete graph represented in Figure 1. The construction of the Prüfer number is described as follows: locate the leaf vertex having the smallest label. In this case, it is vertex 1. Since vertex 2 (the only vertex) is incident to vertex 1 in the tree, assign 2 to the first digit in the permutation, then remove vertex 1 and edge (1,2). Now vertex 3 is the smallest labeled leaf vertex and vertex 5 is incident to it, assign 5 to the second digit in the permutation and then remove vertex 3 and edge (3,5). Repeat the process on the subtree until edge (5,8) is left and the Prüfer number of this tree with 6 digits is finally produced.

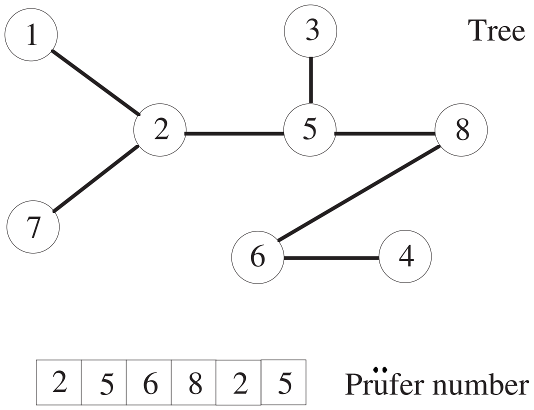

Figure 1: A tree and its Prüfer number

Conversely, we can construct the corresponding tree using the Prüfer number, $P = [\ 2\ 5\ 6\ 8\ 2\ 5\ ]$. As the vertex 1, 3, 4 and 7 are not included in P, they are set as $\overline{P} = \{1,3,4,7\}$. Firstly, vertex 1 is the smallest digit in $\overline{P}$ and vertex 2 is the leftmost digit in P. Add edge (1,2) to the tree, remove vertex 1 from $\overline{P}$ and the leftmost digit 2 of P leaving $P = [\ 5\ 6\ 8\ 2\ 5\ ]$. Secondly, vertex 3 is now the smallest digit in $\overline{P}$ and vertex 5 is the leftmost digit in remaining P. Then add edge (3,5) to the tree, remove vertex 3 from $\overline{P}$ and the leftmost digit 5 from P leaving P

= [6 8 2 5]. Repeat the process until P = [5] and vertices 2 and 8 are in $\overline{P}$. Add edge (2,5) to the tree, remove the last digit 5 of P and vertex 2 from $\overline{P}$. As vertex 5 now no longer occurs in the remaining P, it needs putting into $\overline{P}$ leaving $\overline{P} = \{5{,}8\}$. Finally, P is empty and only vertices 5 and 8 are in $\overline{P}$. Thus add edge (5,8) to the tree and stop. The tree in Figure 1 is formed.

It is clearly that the Prüfer number is a skillful encoding for spanning tree. The encoding length is only $n-2$, the search space size is n^{n-2} and the relation between Prüfer number and spanning tree is one-to-one mapping, therefore the probability to randomly produce a spanning tree is definitely 1, which means that this encoding is capable of equally and uniquely representing all possible trees and any initial population or offspring from crossover and mutation operation still keep a tree. As to the mc-MST problem, we can easily calculate each objective for each chromosome or individual according to its corresponding Prüfer number. Therefore, the Prüfer number encoding is adopted as the chromosome representation.

4.2 Crossover and Mutation

Crossover and mutation are two crucial factors in the biological evolutionary process. In this paper we adopt the uniform crossover operator which has been shown to be superior to traditional crossover strategies for combinatorial optimization problem [22]. Uniform crossover firstly generates a random crossover mask and then exchanges relative genes between parents according to the mask. A crossover mask is simply a binary string with the same size of chromosome. The parity of each bit in the mask determines, for each corresponding bit in an offspring, which parent it will receive that bit from. The operation can be illustrated in Figure 2.

Mutation is a background operator, which produces spontaneous random changes in various chromosomes. A simple way to achieve mutation would be to alter one or more genes. Here the mutation is performed as random perturbation within the permissive range of integers from 1 to n (n is the number of vertices in graph) as illustrated in Figure 3.

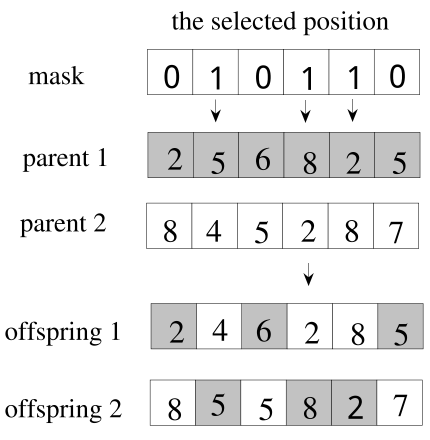

Figure 2: Illustration of crossover operation

4.3 Evaluation and Selection

As in nature, it is necessary to provide the possible driving mechanism for better individuals to survive. Evaluation is to associate each individual or chromosome with a fitness value that

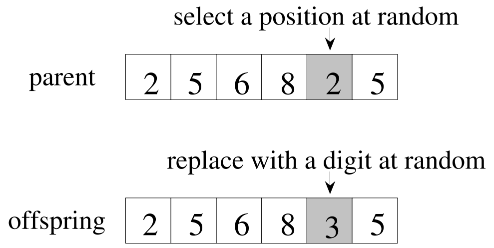

Figure 3: Illustration of mutation operation

reflects how good it is based upon its achievement of the objectives. The higher the fitness value of an individual, the higher its chances of survival and reproduction and the larger its representation in the subsequent generation. Selection is to select individuals to be parents for the next generation according to their own fitness values. Therefore, the evaluation and selection play a very important role in the evolutionary process. In the search process, however, the optimal solution to the mc-MST problem is a set of solutions — Pareto optimal solutions. Generally, different evaluation and selection strategies will reach the different Pareto optimal solutions. In this paper we propose two different evaluation and selection strategies which reflect the preference of decision maker in the multi-criteria approach like the mc-MST problem.

Strategy I: In order to evaluate the fitness of each individual in the GA approach for the mc-MST problem, we can design a adaptive evaluation function based upon the method of objective weighting discussed in Section 3.1. The fitness values of all individuals are calculated according to this adaptive evaluation function. In each generation the set of Pareto solution candidates is updated and the adaptive evaluation function is reformed again and again with the search process. In criteria space the adaptive evaluation function acts as a hyperplane. Under the pressure of selection the hyperplane enforces all Pareto solution candidates to move along its negative gradient direction if the problem is minimized and evolves them toward the ideal point as close as possible. The process can be illustrated by Figure 4.

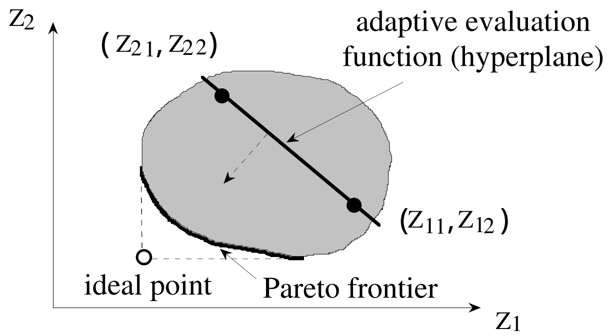

Figure 4: Illustration of the evolutionary process of Strategy I

The evaluation procedure can be operated as follows:

procedure: evaluation for Strategy I
step 1: Decode all chromosomes and calculate their objective values in each objective function.
step 2: Determine $\lambda_k (k = 1, 2, \cdots, p)$ according to Definition 4.

step 3: Determine the fitness value eval(T) of all individuals according to the following formula:

$$\text{eval}(T) = \sum_{k=1}^{p} \lambda_k \; z_k$$

As to the selection procedure, we adopt the idea of the $(\mu + \lambda)$ - selection in evolution strategy [2]. In order to avoid the premature convergence of the GA operation, our selection strategy simply selects μ best individuals from μ parents and λ offspring. If there are no μ individuals available, the vacant pool of population is filled up with renewal individuals generated randomly.

Strategy II: In Strategy I, because the selective pressure is put upon toward the ideal point, it is possible to make all Pareto optimal solutions focused on the region near the ideal point. Actually, this is not always desirable in practice. In other cases, we may hope to obtain all Pareto optimal solutions distributed along the Pareto frontier, which would supply with enough alternatives for decision maker to choose. Here, we adopt the nondominated sorting technique suggested by N. Srinivas and K. Deb [21] to deal with this situation.

Before the selection is performed in nondominated sorting, the population is ranked on the basis of an individual's nondomination. The nondominated individuals present in the population are first identified from the current population. Then all these individuals are assumed to constitute the first nondominated front in the population and assigned a large dummy fitness value. The same fitness value is assigned to give an equal reproductive potential to all these nondominated individuals. To maintain diversity in the population, these classified individuals are then *shared* [10] with their dummy fitness values. Sharing is achieved by performing selection operation using degraded fitness values that are obtained by dividing the original fitness value of an individual by a quantity proportional to the number of individuals around it. This causes multiple Pareto optimal points co-exist in the population.

After sharing, these nondominated individuals are ignored temporarily to precess the rest of the population in the same way to identify individuals for the second nondominated front. These nondominated points are then assigned a new dummy fitness value that is kept smaller than the minimum shared dummy fitness of the previous front. This process is continued until the entire population is classified into several fronts. The population is then reproduced according to the dummy fitness values.

The whole procedure can be concisely formulated as follows:

procedure: evaluation for Strategy II

step 1: Determine all nondominated individuals P_c from the current population, and assign a large dummy fitness value to them.

step 2: Calculate each individual's *niche count* m_j:

$$m_j = \sum_{k \in P_c} \text{sh}(d_{jk})$$

where

$$\text{sh}(d_{jk}) = \begin{cases} 1 - (\frac{d_{jk}}{\sigma_{\text{share}}})^2, & \text{if } d_{jk} < \sigma_{\text{share}} \\ 0, & \text{otherwise} \end{cases}$$

and d_{jk} is the phenotypic distance between two individuals j and k in the current nondominated solution set and σ_{share} is the maximum phenotypic distance allowed between any two individuals to become members of a niche.

step 3: Calculate the shared fitness value of each individual by dividing its dummy fitness value by its niche count.

step 4: Ignore all sorted nondominated individuals, go to Step 1 and continue the process until the entire population is all sorted.

As to the selection, only the *roulette wheel* selection is adopted. Because the other dummy fitness value for the following sorted nondominated individuals is just kept smaller than the minimum shared dummy fitness of the previous nondominated set, the individuals in the first nondominated set have more chance to be selected in the next generation than those in the rest of nondominated set. This is intended to search for the whole nondominated regions or Pareto optimal frontier.

4.4 mc-MST Genetic Algorithm

The overall pseudo-code procedure for the mc-MST problem is outlined as follows:

procedure: genetic algorithm for mc-MST
begin
  $t \leftarrow 0$;
  initialize the population of parents $P(0)$;
  determinate the set of nondominated solutions $E(0)$;
  while (**not** termination condition) **do**
  begin
    recombine $P(t)$ to yield the population of offspring $C(t)$;
    update $E(t)$;
    if (preference on Strategy I) **then**
      evaluate $P(t)$ and $C(t)$ by Strategy I;
      select $P(t+1)$ from $P(t)$ and $C(t)$ by Strategy I;
    else
      evaluate $P(t)$ and $C(t)$ by Strategy II;
      select $P(t+1)$ from $P(t)$ and $C(t)$ by Strategy II;
    end
    $t \leftarrow t+1$;
  end
end

5 Numerical Example

The performance of mc-MST by the GAs approach is tested on five numerical examples from 10-vertex to 50-vertex complete graph generated randomly. All of them have two objectives. Supposed that one is to minimize the cost spent on the construction of the spanning tree; the other is to minimize the cost spent on the maintenance of it. The weights defined on edges for two objectives are generated randomly and respectively distributed uniformly over $[10, 100]$ and $[10, 50]$.

The parameters for the GA are set as follows: population size $pop\_size = 200$; crossover probability $p_c = 0.2$; mutation probability $p_m = 0.05$; maximum generation $max\_gen = 500$; and run by 10 times.

Table 1 gives out the general results got by the GA approach with two strategies and the method of Pareto optimal enumeration. They are all run on EWS4800/360PX workstation.

Table 1: General results by the GA and enumeration approach

Problem size	Enumeration		GA with Strategy I		GA with Strategy II	
(vertices)	TPO	runtimes	percent	runtimes	percent	runtimes
10	56	20 (min.)	100 (%)	3.02 (min.)	100 (%)	4.20 (min.)
20	123	56 (min.)	100 (%)	4.31 (min.)	100 (%)	6.50 (min.)
30	262	2.2 (hours)	90 (%)	6.18 (min.)	94 (%)	9.32 (min.)
40	417	5.6 (hours)	86 (%)	8.05 (min.)	91 (%)	12.12 (min.)
50	635	12.4 (hours)	80 (%)	10.96 (min.)	85 (%)	15.58 (min.)

TPO: the total number of Pareto optimal solutions;
percent: the percentage of the real Pareto optimal solutions.

Table 1 shows that the GA approaches are much more efficient than the method of Pareto optimal enumeration. Compared with the results obtained by the method of Pareto optimal enumeration, all the results obtained by the GA approaches are real Pareto optimal solutions when the problem scale is small, and most of them are real Pareto optimal solutions when the problem scale is larger.

Table 2 gives out the ideal points and extreme Pareto optimal solutions for two tested mc-MST examples. The results are obtained directly by the Prim's MST algorithm [16] and expressed in the objective space.

Table 2: Ideal points and extreme Pareto optimal solutions for mc-MST

Problem size	ideal point	extreme Pareto optima	
(vertices)	(z_1°, z_2°)	(z_1°, z_2)	(z_1, z_2°)
10	(156, 112)	(156, 195)	(387, 112)
50	(583, 513)	(583, 1433)	(2366, 513)

The average results in 10 trials by the GA approach with Strategy I are illustrated in Figures 5 and 6. Figures 5 and 6 give out both the ideal points and all Pareto optimal solutions. They clearly show that all Pareto optimal solutions are close to the ideal point as much as possible under the action of the adaptive evaluation function (hyperplane). In the case of 50 vertices as shown in Figure 6, it is particularly obvious that the adaptive evaluation function (hyperplane) really makes all Pareto optimal solutions evolve toward the ideal point and focus on the Pareto frontier near the ideal point. This reflects such a decision makers' preference, *i.e.* all objectives have the same importance, and the closer all objectives to the ideal point and the better.

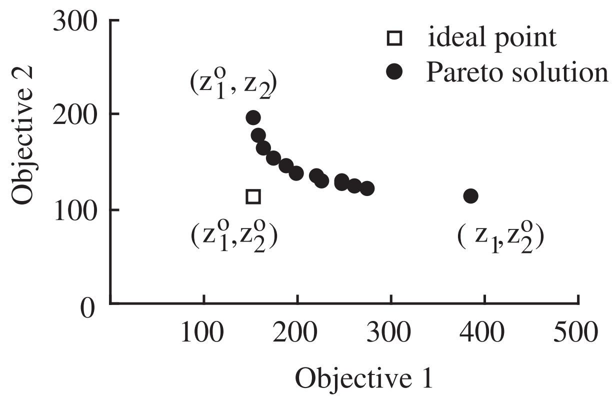

Figure 5: Illustration of GA on mc-MST with 10-vertex by Strategy I

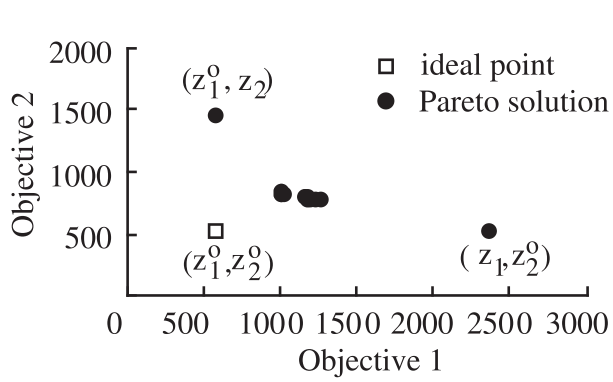

Figure 6: Illustration of GA on mc-MST with 50-vertex by Strategy I

However, sometimes we also hope to have the Pareto optimal solutions distributed all along the Pareto frontier to provide enough information about alternative solutions for the decision maker. Therefore, we use the same two numerical examples and test them by using Strategy II. According to the two extreme Pareto optimal solutions, the parameter σ_{share} for two cases are set as 25 and 200 respectively. Figures 7 and 8 give out the results by Strategy II.

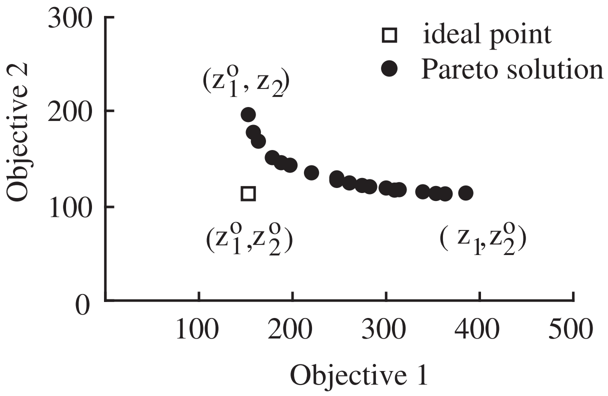

Figure 7: Illustration of GA on mc-MST with 10-vertex Strategy II

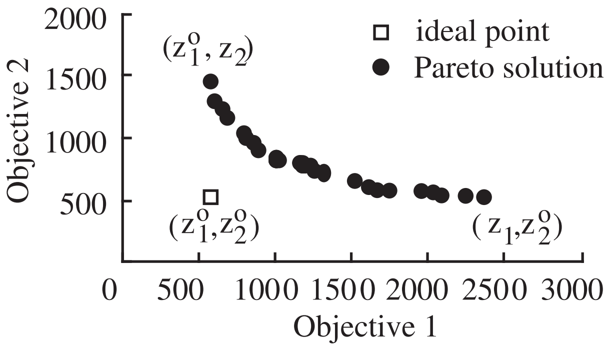

Figure 8: Illustration of GA on mc-MST with 50-vertex Strategy II

Compared with Figures 5 and 6, the results shown in Figures 7 and 8 are all distributed along the Pareto frontier instead of focusing on a specific regions. Actually, it really provides enough information and Pareto optimal solutions for decision maker to choose because there are all possible alternatives to reflect the importance and attainment of each objective respectively. Surely it is beneficial in practice.

6 Conclusion

This paper developed a new GA approach for the mc-MST problem. This problem has not been well dealt with in network optimization. In traditional network optimization techniques, it is difficult to span a tree according to multiple attributes defined on each edge. Unlike traditional incremental approach to MST format, the GA approach directly performs a search on coding space and evolve them to the Pareto optimal solutions through the genetic operations. Combined with the MCDM techniques and nondominated sorting technique, the results of the proposed method show its high effectiveness in dealing with the mc-MST problem. According to the preference of decision maker, the proposed GA approach can obtain all Pareto optimal solutions close to the ideal point or produce a set of solutions distributed along the whole Pareto frontier. Although this paper has only dealt with the classical MST problem with multi-criteria, it is easy to extend the proposed method to solve those degree-constrained MST, stochastic MST, probabilistic MST and quadratic MST problems with multi-criteria. On the whole, the research work not only develops the classical MST problem but also shows the potential power of GAs for network optimization problems with multi-criteria.

Acknowledgment

This research work was partially supported by the International Scientific Research Program (No. 07045032: 1995.4 – 1998.3), Grant-in-Aid for Scientific Research by the Ministry of Education, Science and Culture of the Japanese Government.

References

[1] Abuali, F.N., Wainwright, R.L., and Schoenefeld, D.A., "Determinant factorization: a new encoding scheme for spanning trees applied to the probabilistic minimum spanning tree problem", in: L.J. Eshelman (ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, San Francisco, CA, 1995, 470-475.

[2] Bäck, T., "Selective pressure in evolutionary algorithms: a characterization of selection mechanisms", in D. Fogel (ed.), *Proceedings of the First IEEE Conference on Evolutionary Computation*, IEEE Press, Florida, 1994, 57-62.

[3] Bertsimas, D.J., "The probabilistic minimum spanning tree problem", *Networks* 20 (1990) 245-275.

[4] Cayley, A., "A theorem on trees", *Quarterly Journal of Mathematics* 23 (1889) 376-378.

[5] Chankong, V., and Haimes, Y.Y., *Multiobjective Decision Making Theory and Methodology*, North-Holland, New York, 1983.

[6] Dijkstra, E.W., "A note on two problems in connection with graphs", *Numerische Mathematik* 1 (1959) 269-271.

[7] Fogel, D., *Evolution Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press, Piscataway, NJ, 1995.

[8] Fonesca, C.M. and Fleming, P.J., "Genetic algorithms for multiobjective optimization: formulation, discussion and generalization", in: S. Forrest (ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, San Mateo, CA, 1993, 42-50.

[9] Gen, M., and Cheng, R., *Genetic Algorithms and Engineering Design*, John Wiley & Sons, New York, 1997.

[10] Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Reading, MA: Addison-Wesley, 1989.

[11] Hwang, C., and Yoon, K., *Multiple Attribute Decision Making: Methods and Applications*, Springer-Verlag, Berlin, 1981.

[12] Ishii, H., Shiode, H., and Nishida, T., "Stochastic spanning tree problem", *Discrete Applied Mathematics* 3 (1981) 263-273.

[13] Kruskal Jr, J.B., "On the shortest spanning subtree of a graph and the traveling salesman problem", *Proc. ACM* 7/1 (1956) 48-50.

[14] Michalewicz, Z., *Genetic Algorithms + Data Structure = Evolution Program*, 2nd ed. Springer-Verlag, Berlin Heidelberg, 1994.

[15] Narula, S.C., and Ho, C.A., "Degree-constrained minimum spanning tree", *Computer &. Operations Research* 7 (1980) 239-249.

[16] Prim, R.C., "Shortest connection networks and some generalizations", *Bell Sys. Tech. J.* 36 (1957) 1389-1401.

[17] Prüfer, H., "Neuer beweis eines satzes über permutationen", *Arch. Math. Phys.* 27 (1918) 742-744.

[18] Reynolds, R., and Nazzal, A., "Using cultural algorithm with evolutionary computing to extract site location decisions from Spatio-Temporal databases", in P.J. Angeline (eds.), *Lecture Notes in Computer Science: Evolutionary Programming VI*, Springer- Verlang, Berlin, 1997, 443-456.

[19] Schaffer, J.D., "Some experiments in machine learning suing vector evaluated genetic algorithms", Ph.D. dissertation (TCGA file No. 00314), Vanderbilt University, Nashville, TN, 1984.

[20] Sollin, M., "Le trace de canalisation", in C. Berge and A. Ghouilla-Houri (eds.), *Programming, Games, and Transportation Networks*, John Wiley & Sons, New York, 1965.

[21] Srinivas, N., and Deb, K., "Multiobjective optimization using nondominated sorting in genetic algorithms", *Evolutionary Computation* 2/3 (1995) 221-248.

[22] Syswerda, G., "Uniform crossover in genetic algorithms", in J. Schaffer (ed.), *Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, San Mateo, CA, 1989, 2-9.

[23] Xu, W., "On the quadratic minimum spanning tree problem", in M. Gen and W. Xu (eds.), *Proceedings of 1995 Japan-China International Workshops on Information Systems*, Ashikaga, Japan, 1995, 141-148.

[24] Zhou, G., and Gen, M., "Approach to the degree-constrained minimum spanning tree problem using genetic algorithms", *Engineering Design and Automation* 3/2 (1997) 156-165.

[25] Zhou, G., and Gen, M., "A note on genetic algorithm approach to the degree-constrained spanning tree problems", *Networks* 30 (1997) 91-95.

[26] Zhou, G., and Gen, M., "An effective genetic algorithm approach to the quadratic minimum spanning tree problem", *Computer & Operations Research*, forthcoming issue.