A VARIANCE MINIMIZATION MODEL FOR FUZZY RANDOM MINIMUM SPANNING TREE PROBLEMS

E.-B. Mermri, H. Katagiri, M. Sakawa and H. ISHII

Abstract. In this paper, we deal with a minimum spanning tree programming problem involving fuzzy random weights and propose a fuzzy random programming model using possibility and necessity measures. First, we focus on the degree of possibility or necessity that the objective function satisfies a fuzzy goal. Next, we formulate the problem to minimize the variance of the degree that the objective function value satisfies a fuzzy goal. It is shown that the problem is transformed into a deterministic equivalent one, which is a nonlinear minimum ratio spanning tree problem with a constraint. In order to solve the problem, a Tabu Search (TS) algorithm is developed.

1 Introduction The Minimum Spanning Tree (MST) problem is to find a least cost spanning tree in an edge weighted graph. The efficient polynomial-time algorithms to solve MST problems have been developed by Kruskal [13] and Prim [17]. In the real world, MST problems are usually seen in network optimization. For instance, when designing a layout for telecommunication system, if a decision maker wish to minimize the cost for connection between cities, it is formulated as an MST problem. As other examples, the objective is to minimize the time for construction or to maximize the reliability.

Most research papers with respect to MST problems dealt with the case where each weight is constant. However, in order to investigate more realistic cases, it is necessary to consider the situation that one makes a decision on the basis of data involving randomness and fuzziness simultaneously. For instance, the cost for connection or construction often depends on the economical environment which varies randomly, and experts often estimate the cost not as a constant but as an ambiguous value. In order to take account of such situations, we deal with a minimum spanning tree problem where each edge weight is a fuzzy random variable. We call it a Fuzzy Random Minimum Spanning Tree (FRMST) problem, which is a generalized problem of fuzzy random bottleneck spanning tree problems [11].

A fuzzy random variable was first defined by Kwakernaak [14] , and its mathematical basis was developed by Puri and Ralescu [19]. Recently, some researchers considered fuzzy random linear programming problems, see [8, 15, 20, 23]. We could take various approaches to an FRMST problem according to the interpretations of the problem.

In this paper, we take a possibilistic and stochastic programming approach to fuzzy random programming problems, which is based on the idea provided by Katagiri et al. [8, 10, 12]. First we consider a degree of possibility or necessity that the total edge cost is substantially smaller than or equal to some value. Since the degree varies randomly, we formulate the problem to minimize the variance of the degree. We will show that the formulated problem is equivalent to a deterministic constrained nonlinear minimum ratio spanning tree problem, where the constraint is that the expected degree is larger or equal to a constant. This problem is generally an NP-hard problem.

2003 *Mathematics Subject Classification.* Primary 90C15, 90C59; Secondary 90C10, 90C29.

Key words and phrases. minimum spanning tree, fuzzy random variable, variance minimization model, tabu search.

For combinatorial optimization problems, there are many heuristic solution methods such as genetic algorithms, simulated annealing, ant colony optimization, TS etc. Recently, many literatures show that TS is one of the most efficient solution methods for combinatorial optimization problems, see [5, 6, 16] for instance. Cunha and Ribeiro [2] applied a tabu search algorithm to water network optimization. Blum and Blesa [1] investigated some metaheuristic approaches for edge-weighted k-cardinality tree problems and compared the performances of genetic algorithms, simulated annealing, ant colony optimization and TS. They demonstrated that TS has advantages for high cardinality. Since an MST problem is a special type of edge-weighted k-cardinality tree problems and corresponds to the highest cardinality case, we construct a solution method through a TS algorithm.

2 MST problem with fuzzy random edge weights Consider a connected undirected graph $G=(V,E)$, where $V=\{v_1,v_2,\ldots,v_n\}$ is a finite set of vertices representing terminals or telecommunication stations etc., and $E=\{e_1,e_2,\ldots,e_m\}$ is a finite set of edges representing connections between these terminals or stations. Let T be a spanning tree in the graph G and let $\boldsymbol{x}=(x_1,x_2,\ldots,x_m)$ denote its characteristic vector defined by

$$x_i=\begin{cases}1 & \text{if } e_i \text{ is an edge of } T\\ 0 & \text{otherwise.}\end{cases} \qquad i=1,\cdots,m$$

In this paper, we consider an MST problem involving fuzzy random weights as follows:

$$\left\{\begin{array}{ll}\min & \tilde{\bar{\boldsymbol{C}}}\boldsymbol{x}\\ \text{s. t.} & \boldsymbol{x}\in X,\end{array}\right. \tag{1}$$

where $\boldsymbol{x}=(x_1,\ldots,x_m)^t$ is a decision variable column vector, $\tilde{\bar{\boldsymbol{C}}}=(\tilde{\bar{C}}_1,\ldots,\tilde{\bar{C}}_m)$ is a coefficient vector and X stands for the set of characteristic vectors representing all possible spanning trees of the graph G. Each $\tilde{\bar{C}}_j$ is a fuzzy random variable with the following membership function:

$$\mu_{\tilde{\bar{C}}_j}(t)=\begin{cases}\max\left\{0,1-\dfrac{\bar{c}_j-t}{\alpha_j}\right\}, & \text{if } t\le\bar{c}_j\\ \max\left\{0,1-\dfrac{t-\bar{c}_j}{\beta_j}\right\}, & \text{if } t\ge\bar{c}_j\end{cases} \qquad j=1,\ldots,m, \tag{2}$$

where $\bar{c}_j$ denotes a random variable (or a scenario variable) whose realization under the scenario s is c_{js}. Parameters α_j and β_j denote the left and right spread of a fuzzy number, respectively. Let p_s be the probability that a scenario s occurs, and let S denote the number of scenarios. We assume that $\sum_{s=1}^{S}p_s=1$ holds.

Since the coefficients of the objective function are triangular fuzzy random variables, the objective function also becomes the same type of fuzzy random variable $\tilde{\bar{Y}}$ with the following membership function:

$$\mu_{\tilde{\bar{Y}}}(y)=\begin{cases}\max\left\{0\ ,\ 1-\dfrac{\sum_{j=1}^{m}\bar{c}_jx_j-y}{\sum_{j=1}^{m}\alpha_jx_j}\right\} & \text{if } y\le\sum_{j=1}^{m}\bar{c}_jx_j\\ \max\left\{0\ ,\ 1-\dfrac{y-\sum_{j=1}^{m}\bar{c}_jx_j}{\sum_{j=1}^{m}\beta_jx_j}\right\} & \text{if } y\ge\sum_{j=1}^{m}\bar{c}_jx_j.\end{cases} \tag{3}$$

Considering the imprecision or fuzziness of the decision maker's judgment, for each objective function of problem (1), we introduce a fuzzy goal $\tilde{G}$ with the membership function expressed as:

$$\mu_{\tilde{G}}(y) = \begin{cases} 0, & y > g^0 \\ \dfrac{y - g^0}{g^1 - g^0}, & g^1 \le y \le g^0 \\ 1, & y < g^1. \end{cases} \tag{4}$$

3 Variance minimization model using a possibility measure For problems involving ambiguous coefficients, Dubois and Prade [4] considered possibilistic programming which is based on the possibility theory introduced by Zadeh [24].

Since the membership function $\mu_{\tilde{\bar{Y}}}$ is regarded as a possibility distribution, the degree of possibility $\Pi_{\tilde{\bar{Y}}}(\tilde{G})$ that the objective function value satisfies the fuzzy goal $\tilde{G}$ is defined by

$$\Pi_{\tilde{\bar{Y}}}(\tilde{G}) = \sup_y \min \left\{ \mu_{\tilde{\bar{Y}}}(y),\ \mu_{\tilde{G}}(y) \right\}. \tag{5}$$

Accordingly, instead of the original MST problem with fuzzy random edge weights, we consider the following spanning tree problem to maximize the degree of possibility:

$$\begin{cases} \max & \Pi_{\tilde{\bar{Y}}}(\tilde{G}) \\ \text{s. t.} & \boldsymbol{x} \in X. \end{cases} \tag{6}$$

In this research, we calculate $g^{\max}$ and $g^{\min}$ defined by

$$g^{\max} = \max_s \max_{\boldsymbol{x} \in X} \sum_{j=1}^m c_{js} x_j,$$
$$g^{\min} = \min_s \min_{\boldsymbol{x} \in X} \sum_{j=1}^m c_{js} x_j.$$

Assume that g^1 and g^0 are determined by a decision maker so as to satisfy the condition that $g^{\min} \ge g^1$ and $g^{\max} \le g^0$. Then we have

$$g^1 \le \sum_{j=1}^m \bar{c}_j x_j \le g^0 \tag{7}$$

Using membership functions (3) and (4), and relation (7), then it is easy to show that the degree of possibility (5) is attained at a point y^* which satisfies the following equation

$$\mu_{\tilde{\bar{Y}}}(y^*) = \mu_{\tilde{G}}(y^*), \quad g^1 \le y^* \le \sum_{j=1}^m \bar{c}_j x_j.$$

Hence

$$y^* = \frac{g^1 \sum_{j=1}^m \alpha_j x_j + (g^0 - g^1) \sum_{j=1}^m \bar{c}_j x_j}{\sum_{j=1}^m \alpha_j x_j - g^1 + g^0}. \tag{8}$$

Replace y^* by its value in the membership functions (3) or (4), then the degree of posibility is represented as follows:

$$\Pi_{\tilde{\bar{Y}}} = \frac{\sum_{j=1}^{m} \{\alpha_j - \bar{c}_j\} x_j + g^0}{\sum_{j=1}^{m} \alpha_j x_j - g^1 + g^0}. \tag{9}$$

It should be noted here that the degree of possibility in problem (6) varies randomly because the degree of possibility includes the random variables $\bar{c}_j$ as it is shown in (9). Therefore, problem (6) is regarded as a stochastic MST problem. Katagiri et al. [9] considered an FRMST problem, which is to maximize the expected degree of possibility or necessity that the objective function value satisfies a fuzzy goal. This model is useful for decision making under fuzzy stochastic environments; however, in the obtained solution based on this model, there are often cases where the degree corresponding to a certain scenario is fairly small because the variance of the degree is not considered. Therefore, in this section, we propose the model to minimize the variance of degree of possibility under the condition that the expected degree is larger or equal to some constant. Then the problem to be considered is formulated as follows:

$$\begin{cases} \min & Var[\Pi_{\tilde{\bar{Y}}}(\tilde{G})] \\ \text{s. t.} & E\left[\Pi_{\tilde{\bar{Y}}}(\tilde{G})\right] \geq \delta, \; \boldsymbol{x} \in X, \end{cases} \tag{10}$$

where $E[\cdot]$ and $Var[\cdot]$ denote the expectation and the variance functions, respectively. The expectation and the variance of degrees of possibility are calculated as follows:

$$E[\Pi_{\tilde{\bar{Y}}}(\tilde{G})] = \frac{\sum_{s=1}^{S} p_s \left[\sum_{j=1}^{m} \{\alpha_j - c_{js}\} x_j + g^0\right]}{\sum_{j=1}^{m} \alpha_j x_j - g^1 + g^0},$$

$$Var[\Pi_{\tilde{\bar{Y}}}(\tilde{G})] = \frac{1}{\left(\sum_{j=1}^{m} \alpha_j x_j - g^1 + g^0\right)^2} \, Var\left[\sum_{j=1}^{m} \bar{c}_j x_j\right].$$

Let V denote the variance-covariance matrix of $\bar{\boldsymbol{c}}$. Then the problem to minimize the variances of degrees of possibility is formulated as:

$$\begin{cases} \min & z(\boldsymbol{x}) = \dfrac{1}{\left(\sum_{j=1}^{m} \alpha_j x_j - g^1 + g^0\right)^2} \, \boldsymbol{x}^T V \boldsymbol{x} \\ \text{s. t.} & \sum_{j=1}^{m} \left\{ \sum_{s=1}^{S} p_s c_{js} + (\delta - 1)\alpha_j \right\} x_j \leq (1-\delta) g^0 + \delta g^1 \\ & \boldsymbol{x} \in X. \end{cases} \tag{11}$$

The variance-covariance matrix is expressed by

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1m} \\ v_{21} & v_{22} & \cdots & v_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mm} \end{bmatrix},$$

where

$$v_{jj} = Var[\bar{c}_j] = \sum_{s=1}^{S} p_s \{c_{js}\}^2 - \left\{ \sum_{s=1}^{S} p_s c_{js} \right\}^2, \quad j = 1, \dots, m,$$

$$v_{jl} = Cov[\bar{c}_j, \bar{c}_l] = E[\bar{c}_j, \bar{c}_l] - E[\bar{c}_j]E[\bar{c}_l], \quad j \neq l,\ j, l = 1, \dots, m$$

and

$$E[\bar{c}_j, \bar{c}_l] = \sum_{s=1}^{S} p_s c_{js} c_{ls}$$

4 Variance minimization model using a necessity measure In the case where the decision maker prefers to hedge a risk or find a pessimistic solution, the model using a possibility measure may not be appropriate because a solution of the model using a possibility measure might be too optimistic for such a decision maker. Therefore, in this section, we consider the variance minimization model using a necessity measure.

The degree of necessity $N_{\tilde{\bar{Y}}}(\tilde{G})$ that the objective function values satisfy fuzzy goals is expressed as:

$$N_{\tilde{\bar{Y}}}(\tilde{G}) = \inf_y \max \left\{ 1 - \mu_{\tilde{\bar{Y}}}(y),\ \mu_{\tilde{G}}(y) \right\}. \tag{12}$$

If the decision maker prefers to maximize the degree of necessity, then problem (1) is reformulated as

$$\begin{cases} \max N_{\tilde{\bar{Y}}}(\tilde{G}) \\ \text{s. t. } x \in X. \end{cases} \tag{13}$$

By using membership functions (3) and (4), and relation (7), it is easy to see that the degree of necessity (12) is attained at a point y^* which satisfies the following equation

$$1 - \mu_{\tilde{\bar{Y}}}(y^*) = \mu_{\tilde{G}}(y^*), \quad \sum_{j=1}^{m} \bar{c}_j x_j \leq y^* \leq g^0.$$

Hence

$$y^* = \frac{-g^0 \sum_{j=1}^{m} \beta_j x_j + (g^1 - g^0) \sum_{j=1}^{m} \bar{c}_j x_j}{-\sum_{j=1}^{m} \beta_j x_j + g^1 - g^0}. \tag{14}$$

Replace y^* by its value in the membership functions (3) or (4), then the degree of posibility is expressed as follows:

$$N_{\tilde{\bar{Y}}}(\tilde{G}) = \frac{g^0 - \sum_{j=1}^{m} \bar{c}_j x_j}{\sum_{j=1}^{m} \beta_j x_j - g^1 + g^0}. \tag{15}$$

Since $N_{\tilde{\bar{Y}}}(\tilde{G})$ varies randomly due to the randomness of $\bar{c}_j$, then problem (13) can be regarded as a stochastic maximization model. In this section, we apply the variance minimization model to this problem and reformulate it to minimize the variance of the degree of necessity:

$$\begin{cases} \min & Var[N_{\tilde{\bar{Y}}}(\tilde{G})] \\ \text{s. t.} & E[N_{\tilde{\bar{Y}}}(\tilde{G})] \geq \delta,\ \boldsymbol{x} \in X, \end{cases} \tag{16}$$

Which can be expressed as:

$$(17) \quad \begin{cases} \min & \dfrac{1}{\left(\sum_{j=1}^{m} \beta_j x_j - g^1 + g^0\right)^2} \, \boldsymbol{x}^T V \boldsymbol{x} \\ \text{s. t.} & \sum_{j=1}^{m} \left\{ \sum_{s=1}^{S} p_s c_{js} + \delta\beta_j \right\} x_j \le (1-\delta) g^0 + \delta g^1, \ \boldsymbol{x} \in X, \end{cases}$$

5 Tabu search algorithm TS is a metaheuristic method that has proven to be very effective for many combinatorial optimization problems such as scheduling, vehicle routing, traveling salesman problem, etc. Hanafi and Freville [7] considered a TS algorithm based on strategic oscillation and showed that the proposed algorithm was well performed for the 0-1 multidimensional knapsack problems.

In this section, we shall construct a solution algorithm based on TS incorporating strategic method. The algorithm starts from an initial spanning tree solution constructed by an adequate algorithm. Then the improvement strategy, which consists of exchanging a pair of edges, generates the neighborhood of the current solution. In TS method a pair of exchanged edges is called move. Hence the resulting solution is a spanning tree. In order to prevent cycling between the same solutions, certain exchanges can be forbidden by earning them the status of "tabu move", and the set of tabu moves defines the tabu list. Tabu moves are not permanent; a short-term memory function enables them to leave the tabu list. The use of an aspiration criterion permits certain moves on the tabu list to overcome any tabu status. In the proposed algorithm, we use strategic oscillation to intensively explore the region around the current neighborhood. Strategic oscillation was originally introduced to provide an effective interplay between intensification and diversification over the intermediate to long term memory. In addition to a short-term memory, we use a frequency-based memory as a long-term memory. On the other hand, the intensification undertakes to create solutions aggressively encouraging the incorporating of solutions from an elite solution set. The process goes on until the termination criterion is satisfied.

Let $G =$ be a graph with n vertices and m edges, and let $E = \{e_1, e_2, \ldots, e_m\}$ be the set of edges of the graph G. Let T be a spanning tree in the graph G and let $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$ denote its characteristic vector defined by

$$x_i = \begin{cases} 1 & \text{if } e_i \text{ is an edge of } T \\ 0 & \text{otherwise.} \end{cases} \qquad i = 1, \cdots, m$$

Problems (11) and (17) can be written on the following form:

$$(18) \quad \begin{cases} \min z(\boldsymbol{x}) = \dfrac{\boldsymbol{x}^t V \boldsymbol{x}}{(F(\boldsymbol{x}))^2} \\ \text{s. t. } \boldsymbol{c}\boldsymbol{x} \le b, \ \boldsymbol{x} \in X, \end{cases}$$

where F is an affine function, $\boldsymbol{c}$ is an m-vector of real numbers and b is a real number. In what follows we say that a solution $\boldsymbol{x} \in X$ is feasible if the condition $\boldsymbol{c}\boldsymbol{x} \le b$ is satisfied.

In the TS algorithm for solving FRMST problem, we shall use the following notations and parameters:

$\boldsymbol{x}^c$: Current solution of the algorithm.
T^c: Current spanning tree with characteristic vector $\boldsymbol{x}^c$.
$\boldsymbol{x}^b$: Best solution found so far.

T^b: Best spanning tree with characteristic vector $\boldsymbol{x}^b$.

NIO: Number of iterations in the oscillation procedure.
MAX_NIO: Threshold of the counter NIO.
k: Counts the iterations where the best solution is unrenewed during the improvement strategy process.
MAX_k: Threshold of the counter k.
$UNRIter$: Counts the iterations of the algorithm where the best solution is unrenewed.
MAX_Iter: Threshold of the counter $UNRIter$.

The proposed TS algorithm is described in the following steps, which are followed by the description of each feature implemented in this algorithm.

Step 0 (Initial solution)
Set $NIO = UNRIter = k = 0$. Generate an initial solution $\boldsymbol{x}^0$ corresponding to an initial spanning tree T^0. If the solution $\boldsymbol{x}_0$ is feasible then denote it by $\boldsymbol{x}_f^0$; otherwise move it to a feasible solution x_f^0 by using the rule described in section 5.2. Set $\boldsymbol{x}^c := \boldsymbol{x}_f^0$ and $\boldsymbol{x}^b := \boldsymbol{x}_f^0$.

Step 1 (Improvement)
If $k > MAX\_k$, then go to step 3. Otherwise, improve the obtained solution by the *improvement strategy.*

Step 2 If $z(\boldsymbol{x}^c) < z(\boldsymbol{x}^b)$, then set $k = 0$ and $\boldsymbol{x}^b := \boldsymbol{x}^c$, and return to step 1. Otherwise, set $k := k + 1$ and return to step 1.

Step 3 (Strategic oscillation)
If $NIO > MAX\_NIO$, then go to step 5. Otherwise, add a_1 edges among $E \setminus T^c$ by using the *edge addition rule* and continue to remove one of the edges in a cycle by using the *edge remove rule* until a spanning tree is formed. Where $E \setminus T^c$ denotes the set of edges in G which are not elements of T^c. If this new solution is infeasible then move it to a feasible one by using the rule in section 5.2.

Step 4 If $z(\boldsymbol{x}^c) < z(\boldsymbol{x}^b)$, then set $NIO = k = 0$, $\boldsymbol{x}^b := \boldsymbol{x}^c$, and return to step 1. Otherwise, set $NIO := NIO + 1$ and return to step 3.

Step 5 (**Intensification by elite solutions**)
Select a set of edges (SE) that are in most of the last M elite solutions. Then, starting from an edge chosen uniformly at random from T^c, construct a new solution by adding edges from T^c and and SE, using edge addition rule in section 5.7, until a spanning tree is formed. If the constructed solution is not feasible then move it to the feasible region. Set $UNRIter := UNRIter + 1$. If $UNRIter > Max\_Iter$, then terminate. Otherwise go the next step.

Step 6
If $z(\boldsymbol{x}^b) < z(\boldsymbol{x}^c)$, then set $\boldsymbol{x}^b := \boldsymbol{x}^c$, $UNRIter = k = 0$ and return to step 1. Otherwise, set $k = 0$ and go to step 1.

The essential features that have been considered in building this TS algorithm for solving a minimum spanning tree problem are: generating an initial solution, move from infeasible to feasible region, the neighborhood structure, the improvement strategies, short-term and long-term memories, diversification procedure and oscillation strategy, intensification, termination criterion.

5.1 Initial solution Let $SCC(i)$ denote a Set of Connected Component that consists of i edges. To construct a spanning tree T, first, an edge $e \in E$ is chosen uniformly at random. With this edge, a subtree $SCC(1)$ which consists of only one edge is created. Then, a set of connected component $SCC(k+1)$ is constructed by adding an edge $e \leftarrow \text{argmin}\{z(SCC(k)+e') - z(SCC(k)) | e' \in E_{NC}(SCC(k))\}$ to the current $SCC(k)$ under construction, where $E_{NC}(SCC(k))$ is defined as follows:

$$E_{NC}(SCC(k)) = \{e \in E | SCC(k) + e \text{ has no cycle}\}.$$

5.2 Move from infeasible to a feasible region strategy Suppose that a solution $\boldsymbol{x}$ is in the infeasible region, i.e., $\boldsymbol{cx} > b$. In this procedure we move $\boldsymbol{x}$ to an element in the feasible region. Let y be an m-vector. We set $y := x$. Let $j = \text{argmax}\{c_i | y_i = 1\}$ and let $k = \text{argmin}\{c_i | y_i = 0 \text{ and } T \backslash - e_j + e_i \text{ is a spanning tree}\}$, provided k exists. If k does not exist, then set $y_i = 0$ and repeat this step until j and k are found. Then set $x_j := 0$ and $x_k := 1$. Repeat this procedure until the inequality $\boldsymbol{cx} \leq b$ is satisfied.

5.3 Neighborhood structure Let T be a set of edges which forms a spanning tree, and let $\mathcal{T}$ be the family of all possible spanning trees in the given graph. A neighborhood $N(T)$ is defined as a set of all spanning trees which can be generated by removing an edge e from T and by adding an edge e' from the set $E_{NH}(T-e) \setminus \{e\}$, where $E_{NH}(T-e)$ is defined as follows:

$$E_{NH}(T-e) = \{e' \in E | T - e + e' \in \mathcal{T}\}.$$

5.4 Improvement strategy Let T^c be the current solution spanning tree. Randomly select a subset of feasible solutions from the neighborhood $N(T^c)$. Then choose a tree with the lowest objective function and non tabu status, to form a new solution. The tabu status can be overriden if an aspiration criterion is satisfied. If a feasible solution is not found in $N(T^c)$, then move the current solution $\boldsymbol{x}^c$ to a feasible one by the rule in in section 5.2.

5.5 Short-term memory TS uses a short-term memory to escape from local minima and to avoid cycling. The short-term memory is implemented as a set of tabu lists that store solution attributes. Attributes usually refer to components of solutions, moves, or differences between two solutions. The use of tabu lists prevents the algorithm from returning to recently visited solutions.

Our TS approach is to tackle the MST problem uses only one tabu list denoted by $TabuList$. The attribute it stores is the index of the edges that were recently added or removed. Every move involves removing one edge e from the current spanning tree T^c, and adding a different edge to $T^c - e$. The status of the forbidden moves are explained as: If an edge e_j is in $TabuList$ and $x_j = 0$, then adding the edge e_j is forbidden. In addition, if an edge e_i is in $TabuList$ and $x_i = 1$, then removing the edge e_i is forbidden.

5.6 Aspiration criterion An aspiration criterion is activated to overcome the tabu status of a move whenever the solution then produced is better than the best historical solution achieved. This criterion will be effective only after a local optimum is reached.

5.7 Strategic oscillation procedure The strategic oscillation approaches by adding or removing edges to a boundary which is represented by a set of spanning trees. Instead of stopping in the boundary, it crosses over the boundary by the modified evaluation criteria for selecting moves. In this paper, we use one type of strategic oscillation approach for the problem, which recedes the boundary by continuing to add edges to a spanning tree and then approaches to the boundary by continuing to remove edges until a spanning tree is formed. Adding edges proceeds for a specified depth beyond the boundary, and turns around. At this point the boundary is again approached and is reached by removing edges.

Edge addition rule Let $SCC(i)$ denote a set of connected component that consists of i edges. Then $SCC(k+1)$ is constructed by adding an edge $e \leftarrow \text{argmin}\{z(SCC(k) + e') - z(SCC(k)) | e' \notin SCC(k)\}$ to the current set $SCC(k)$ under construction.

Edge remove rule Let $SCC(k)$ be a connected set of k edges. Then we construct a spanning tree from the set $SCC(k)$ by performing the procedure used to construct the initial solution, where we use $SCC(k)$ instead of E.

5.8 Long-term memory The roles of intensification and diversification in TS are especially relevant in longer term search processes. Frequency-based memory is one of the long-term memories and consists of gathering pertinent information about the search process so far. In our algorithm, we use residence frequency memory, which keeps track of the number of iterations where no improvement has been done and keeps in memory a number of elite solutions . By using the residence frequency memory, we provide the following diversification and intensification processes.

1. **Diversification procedure** The diversification derives the search into a new region. It begins at the situation that some spanning tree is formed. In order to explore a new search region a number of edges are removed and replaced by other edges. This procedure is illustrated in the strategic oscillation steps of the algorithm. If the strategic oscillation procedure is iterated in MAX_NIO times, then the intensification procedure is started.

2. **Intensification procedure using** The intensification procedure begins at the condition that no edge is selected after a long tune. It forces the current solution to be improved. We tray to construct a better solution from the last M best solution by using the edge addition rule.

5.9 Termination criterion The counter $UNRIter$ counts the iterations of the algorithm where the best solution T^b is unrenewed. The proposed algorithm terminates if $UNRIter$ is greater than the threshold Max_Iter. The quality of the final solution and the computer running time are both influenced by the termination criterion.

6 Genetic algorithm In order to compare the performance of the proposed TS method with another heuristic search method, we consider a genetic algorithm (GA) approach inspired from the work of Zhou and Gen [25] to solve quadratic minimum spanning tree problems by GA. Their approach uses Prüfer number for solution encoding, uniform crossover and mutation, and mixed strategy with $(\mu + \lambda)$ selection and roulette wheel selection. In this section we briefly describe this genetic algorithm approach.

The choice of an encoding solution is one of the most important step in the application of genetic algorithm to a problem. One of the classical theorems in enumeration is Cayley's theorem [3], which says that in a complete undirected graph with n vertices there are n^{n-2} distinct labeled trees. Prüfer [18] provided a constructive proof of Cayley's theorem by establishing a one to one correspondence between such spanning trees and the set of all permutations of $n-2$ digits. Prüfer numbers are $n-2$ digit sequences, $P = [p_1, p_2, \cdots, p_{n-2}]$, where the digits p_i, $1 \leq i \leq n-2$, are numbers between 1 and n.

Prüfer numbers consist one of the most efficient encoding method for spanning trees in genetic algorithm search, since they are unbiased, cover the hall space of spanning trees and represent only spanning trees. However, this representation method has little locality, since even a small change in a parent P which is represented by a Prüfer number, may result an offspring with a tree not in the neighborhood of the parent tree.

In the uniform crossover operation individual bits in the strings of two parents are swapped with a fixed probability p_c. For each crossover operation one generates a random binary string with the same size as of chromosome, with respect to the probability p_c of a string being 1. Then genes in two parents which their positions in the string mask take the digit 1 are swapped. In the mutation operation each gene can be selected with a probability p_m, to be replaced with a random digit in the set of all possible digits. The mixed strategy with $(\mu+\lambda)$ selection and roulette wheel selection selects μ best chromosomes from μ parents and λ offsprings. If there are no μ different chromosomses available, then the vacant pool of population are filled up with roulette wheel selection. For further details on this approach see the work in [25].

7 Computational experiments Let G be a complete undirected graph with n vertices and m edges, and let X be the set of all possible spanning trees of the graph G, represented by $0-1$ m-vectors as it is described in section 5. In this section we apply the proposed Tabu Search algorithm (TS) and the Genetic Algorithm (GA) described in the previous section to solve FRMST problem (11):

$$\begin{aligned} \min \quad & z(\boldsymbol{x}) = \frac{1}{\left(\sum_{j=1}^{m} \alpha_j x_j - g^1 + g^0 \right)^2} \boldsymbol{x}^T V \boldsymbol{x} \\ \text{s. t.} \quad & \sum_{j=1}^{m} \left\{ \sum_{s=1}^{S} p_s c_{js} + (\delta - 1)\alpha_j \right\} x_j \leq (1-\delta) g^0 + \delta g^1 \\ & \boldsymbol{x} \in X. \end{aligned}$$

The experiments were conducted on complete undirected graphs with different number of vertices, n. Data of these experiments are generated randomly. The scenarios (c_{js}), $j = 1, \cdots, m$, $s = 1, 2, 3$, are random real numbers distributed uniformly over [10,13]. The left spreads (α_j) are random real numbers distributed uniformly over (0,1]. The probabilities that each scenario s occurs are given as, $p_1 = 0.41$, $p_2 = 0.28$ and $p_3 = 0.31$. The matrix V and parameters g_0 and g_1 are computed by formulas given in section 3. We assume that the expected degree is larger or equal to $\delta = 0.6$ for $n = 5$, and $\delta = 0.8$ for $n = 10$ and over.

The TS parameters are set as follows: maximum of iterations in the improvement strategy $MAX\_k = 500$, maximum of iterations in the oscillation strategy $MAX\_NIO = 5$, $MAX\_UNRIter = 10$, number of ellite solutions $M = 10$ and number of add remove edge $a_1 = 3$. In the GA approache, the parameters are set as follows: crossover probability $p_c = 0.4$, mutation probability $p_m = 0.01$, population size 100, and maximum number of generations 800.

The algorithms were coded in C++ programming language and implemented on a computer with a CPU 1.7GHz and RAM 256MB. We have run each experiment ten times. The following tables illustrates computation results of two instances of the above problem.

Example 1.

	Best values		Average values		Time (second)	
Nodes	TS	GA	TS	GA	TS	GA
5	1.32×10^{-2}	1.32×10^{-2}	1.32×10^{-2}	1.32×10^{-2}	–	0.06
10	2.65×10^{-3}	2.65×10^{-3}	2.65×10^{-3}	2.65×10^{-3}	0.04	2.2
15	5.76×10^{-6}	5.97×10^{-6}	5.76×10^{-6}	17.31×10^{-6}	0.84	8.11
20	2.00×10^{-10}	3.67×10^{-10}	10.40×10^{-10}	184×10^{-10}	9.7	26
30	2.28×10^{-11}	94.6×10^{-11}	8.7×10^{-11}	357×10^{-11}	83.8	133

Example 2.

	Best values		Average values	
Nodes	TS	GA	TS	GA
5	6.55×10^{-3}	6.55×10^{-3}	6.55×10^{-3}	6.55×10^{-3}
10	2.18×10^{-3}	2.18×10^{-3}	2.18×10^{-3}	2.18×10^{-3}
15	2.63×10^{-6}	2.63×10^{-6}	2.63×10^{-6}	3.92×10^{-6}
20	6.75×10^{-7}	37×10^{-7}	8.41×10^{-7}	77.3×10^{-7}
30	2.19×10^{-11}	198×10^{-11}	3.01×10^{-11}	122×10^{-11}

From these computational results, we remark that GA approach is quite uncompetitive with the constructed TS method for solving the problem in question. In addition, the best and average values obtained by TS are very close or the same for number of nodes less or equal than 15. One conclude that the proposed TS method has good performances for solving FRMST problems.

8 Conclusion In this paper, we have considered FRMST problems. Introducing a fuzzy goal, we formulated the problem to minimize the variance of the degree of possibility or necessity that an objective function satisfies the fuzzy goal. It has been shown that the problem was transformed into a deterministic equivalent nonlinear minimum ratio spanning tree programming problem. In order to solve the problem, we have constructed a TS algorithm based on oscillation strategy, intensification by elite solutions and so on.

In the future, we will consider the fractile criterion optimization model for FRMST problems and try to extend and apply the proposed method to the problem. If the problem is formulated based on the fractile criterion optimization model, the objective will be to maximize a satisfaction level under the condition that the degree of possibility or necessity is greater than or equal to the satisfaction level. In order to solve the problem efficiently, it is expected that the solution technique for inverse MST problems can be applied.

References

[1] Blum, C. and Blesa, M.J. 2005. New metaheuristic approaches for the edge-weighed k-cardinality tree problem, *Computer & Operations Research*, vol. 32, pp. 1355–1377.

[2] Cunha, M.C. and Ribeiro, L. 2004. Tabu search algorithms for water network optimization, *European Journal of Operational Research* vol. 157, pp. 746-758.

[3] Cayley, A. 1889. A theorem on trees, *Quartery Journal of Mathematics*, vol. 23, pp. 376-378.

[4] Dubois, D. and Prade, H. 1980. *Fuzzy Sets and Systems*, Academic Press, New York.

[5] Glover, F. 1989. Tabu Search-Part I., *ORSA Journal on Computing*, vol. 1, pp. 190–206.

[6] Glover, F. 1990. Tabu Search-Part II., *ORSA Journal on Computing*, vol. 2, pp. 4–32.

[7] Hanafi, S. and Freville, A. 1998. An efficient tabu search approach for the 0-1 multidimensional knapsack problem, *European Journal of Operational Research*, vol. 106, pp. 659–675.

[8] Katagiri, H. and Ishii, H. 2000. Linear programming problem with fuzzy random constraint, *Mathematica Japonica*, vol. 52, pp. 123–129.

[9] Katagiri, H., Mermri, E.B., Sakawa, M., Kato, K. and Nishizaki, I. 2005. A possibilistic and stochastic programming approach to fuzzy random MST problems, *IEICE Transaction on Information and Systems*, vol. E88-D, pp. 1912–1919.

[10] Katagiri, H. and Sakawa, M. 2004. An interactive fuzzy satisficing method for fuzzy random linear multiobjective 0-1 programming problems through the expectation optimization model, *Mathematical and Computer Modelling*, vol. 40, pp. 411–421.

[11] Katagiri, H., Sakawa, M. and Ishii, H. 2004. Fuzzy random bottleneck spanning tree problems, *European Journal of Operational Research*, vol. 152, pp. 88–95.

[12] Katagiri, H., Sakawa, M., Kato, K. and Nishizaki, I. 2008. Interactive multiobjective fuzzy random linear programming: Maximization of possibility and probability, *European Journal of Operational Research*, vol. 188, pp. 530–539.

[13] Kruskal, J.B. Jr. 1956. On the shortest spanning subtree of a graph and traveling salesman problem, *Proceedings of ACM* 7/1, pp. 48–50.

[14] Kwakernaak, H. 1978. Fuzzy random variable-1, *Information Sciences*, vol. 15, pp. 1–29.

[15] Luhandjula, M.K. and Gupta, M.M. 1996. On fuzzy stochastic optimization, *Fuzzy Sets and Systems*, vol. 81, pp. 47–55.

[16] Mermri, E.B., Katagiri, H., Sakawa, M. and Kato, K. 2007. Remarks on the application of genetic algorithm and tabu search method to nonlinear spanning tree problems, *Applied Mathematics and Computation*, vol. 188, pp. 1071–1086.

[17] Prim, R.C. 1957. Shortest connection networks and some generations, *Bell System Technical Journal*, vol. 36, pp. 1389–1401.

[18] Prüfer, H., 1918. Neuer beweis eines satzes uber per mutationen, *Archiv derMathematik und Physik* vol. 27, pp. 742-744.

[19] Puri, M.L., and Ralescu, D.A. 1986. Fuzzy random variables, *Journal of Mathematical Analysis and Applications*, vol. 14, pp. 409–422.

[20] Qiao, Z., Zhang Y., and Wang, G. 1994. On fuzzy random linear programming, *Fuzzy Sets and Systems*, vol. 65, pp. 31–49.

[21] Sollin, M. 1965. La trace de canalisation, in: *Programming, Games, and Transportation Networks*, C. Berge, A Ghouilla-Houri Eds. Wiley, New York.

[22] Souza, M.C., Duhamel C. and Ribeiro, C.C. March 2002. A GRASP heuristic for the capacitated minimum spanning tree problem using a memory-based local search strategy, Optimization Online.

[23] Wang, G.Y. and Qiao, Z. 1993. Linear programming with fuzzy random variable coefficients, *Fuzzy Sets and Systems*, vol. 57, pp. 295–311.

[24] Zadeh, L.A. 1978. Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems*, vol. 1, pp. 3–28.

[25] Zhou, G. and Gen, M. 1998, An effective genetic algorithm approach to the quadratic minimum spanning tree problem, *Computer & Operations Research.*, vol. 25, pp. 229–237.

Hideki Katagiri
Graduated School of Engineering, Hiroshima University
Kagamiyama 1-4-1, Higashi-Hiroshima City,
Hiroshima, 739-8527 JAPAN
E-mail: katagiri-h@hiroshima-u.ac.jp