POSSIBILITY-BASED PROBABILITY MAXIMIZATION MODELS FOR FUZZY RANDOM MINIMUM SPANNING TREE PROBLEMS

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

Received February 14, 2011; revised March 6, 2011

Abstract. This paper deals with a minimum spanning tree programming problem involving fuzzy random weights. New decision making models are proposed to maximize the probability that the degree of possibility or necessity that the fuzzy goal for the fuzzy random objective function is satisfied is greater than or equal to a satisficing level. It is shown that the original problems involving fuzzy random variables are transformed into deterministic equivalent ones which are nonlinear maximum spanning tree problems through the proposed models. An efficient tabu search algorithm is developed to solve the resulting nonlinear maximum spanning tree problems.

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 [24, 27, 31]. In the real world, MST problems are usually seen in network optimization. For instance, when designing a layout for telecommunication systems, if a decision maker wishes to minimize the cost for connection between cities, such a decision making situation is formulated as an MST problem.

Most research papers with respect to MST problems dealt with the case where each weight is constant. However, we are often faced with the situation that one makes a decision on the basis of uncertain data. For handling such MST problems under uncertainty, Ishii et al. [13, 14, 15] considered stochastic MST problems in which the weights attached to edges are expressed by random variables, and constructed polynomial-time algorithms for solving deterministic equivalent problems obtained through the transformation of the original MST problem involving randomness. However, in order to investigate more realistic cases, it is important to deal with not only randomness but also fuzziness inherently involved in MST problems in the real world. 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 consider a minimum spanning tree problem where each edge weight is a fuzzy random variable, called a Fuzzy Random Minimum Spanning Tree (FRMST) problem.

A fuzzy random variable was first defined by Kwakernaak [25], and its mathematical basis was developed by Puri and Ralescu [29]. Recently, some researchers considered fuzzy random linear programming problems [16, 22, 19, 26, 30, 32] and combinatorial optimization problems such as 0-1 programming problems [21] and bottleneck spanning tree problems [17, 20].

Although we could take various approaches to a FRMST problem according to the different interpretations of the problem, in this paper, we take a possibilistic and stochastic programming approach. As for fuzzy random MST problems, Katagiri et al. [18] considered the possibility-based expectation model which is to maximize the expectation of degree of

2000 *Mathematics Subject Classification.* Primary 90B15, 90C70; Secondary 90C27, 90C35.
Key words and phrases. Minimum spanning tree problems, fuzzy random variables, possibility measure, necessity measure, probability maximization, tabu search.

possibility or necessity with respect to the attainment level for the fuzzy goal given by the decision maker. However, considering the diversification in individual value of decision makers, the possibility-based expectation model is not always the best one, and it is useful to propose other optimization models.

In this paper, we propose possibility-based probability maximization models for FRMST problems. First we consider a degree of possibility or necessity that the total edge cost is substantially smaller than or equal to some value. Realizing that the degree varies randomly, we formulate the problem to maximize the probability that the degree is greater than or equal to some satisficing level. As will be shown later, the formulated problem is transformed into a deterministic equivalent nonlinear maximum spanning tree problem, which is generally an NP-hard problem. For solving the nonlinear maximum spanning tree problem, we construct an effective tabu search-based approximate algorithm. In order to show the efficiency of the proposed algorithm, we compare the performance of the proposed algorithm with those of the existing algorithms [35].

2 MST problem with fuzzy random edge costs Consider a connected undirected graph $\mathcal{G} = (\mathcal{V}, E)$, where $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ is a finite set of vertices representing terminals or telecommunication stations etc., and $E = \{e_1, e_2, \dots, e_m\}$ is a finite set of edges representing connections between these terminals or stations. Let $\boldsymbol{x} = (x_1, x_2, \dots, x_m)^t$ be a vector defined by

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

In this paper, we consider a minimum spanning tree problem involving fuzzy random weights as follows:

$$\left.\begin{array}{rl} \text{minimize} & \tilde{\bar{\boldsymbol{C}}}\boldsymbol{x} \\ \text{subsect to} & \boldsymbol{x} \in X, \end{array}\right\} \tag{1}$$

where $\tilde{\bar{\boldsymbol{C}}} = (\tilde{\bar{C}}_1, \dots, \tilde{\bar{C}}_m)$ is a coefficient vector and X stands for the set of a 0-1 m-dimensional vectors representing all possible spanning trees of the given graph $\mathcal{G}$. Each $\tilde{\bar{C}}_j$ is a fuzzy random variable taking a fuzzy number $\tilde{\bar{C}}_j(\omega)$ as a realization for each ω, where ω is an elementary event of the universal event Ω. The following is the membership function characterizing $\tilde{\bar{C}}_j(\omega)$:

$$\mu_{\tilde{\bar{C}}_j(\omega)}(t) \quad = \quad \begin{cases} L\left(\dfrac{\bar{d}_j(\omega) - t}{\alpha_j}\right) & (t \leq \bar{d}_j(\omega)) \\[2ex] R\left(\dfrac{t - \bar{d}_j(\omega)}{\beta_j}\right) & (t > \bar{d}_j(\omega)), \end{cases}$$

where $L(t)$ is a strictly non-increasing function on $[0, +\infty)$, satisfying

(i) $L(0) = 1$

(ii) There exists a t_0^L such as $L(t) = 0$ for any $t \geq t_0^L$.

Function R also satisfies the same condition as L. Parameters $\bar{d}_j$, $j = 1, \cdots, m$, denote normal random variables with mean $\mathcal{M}_j^d$, and parameters α_j and β_j, $j = 1, \cdots, m$, are real numbers denoting the left and right spread, respectively. The variance-covariance matrix of the vector $\bar{\boldsymbol{d}} = (\bar{d}_1, \bar{d}_2, \cdots, \bar{d}_m)$ is denoted by V.

By applying the calculation formula [7] with respect to L - R fuzzy numbers based on the extension principle [34] to the fuzzy number $\tilde{\bar{Y}}(\omega)$ for each ω, it is easily shown that $\bar{\boldsymbol{C}}\boldsymbol{x}$ is a fuzzy random variable $\tilde{\bar{Y}}$ with the following membership function for each elementary event ω:

$$\mu_{\tilde{\bar{Y}}(\omega)}(y) \quad = \quad \begin{cases} L\left(\dfrac{\bar{\boldsymbol{d}}(\omega)\boldsymbol{x}-y}{\boldsymbol{\alpha}\boldsymbol{x}}\right) & (y \leq \bar{\boldsymbol{d}}(\omega)\boldsymbol{x}) \\ R\left(\dfrac{y-\bar{\boldsymbol{d}}(\omega)\boldsymbol{x}}{\boldsymbol{\beta}\boldsymbol{x}}\right) & (y > \bar{\boldsymbol{d}}(\omega)\boldsymbol{x}). \end{cases}$$

3 Possibility-based probability maximization model In problem (1), the total edge weight represented by a fuzzy random variable cannot be minimized in the deterministic sense. Therefore, we construct an optimization criterion to take account of the uncertainty included in the problem. For constructing an optimization criterion, we focus on the concepts of *vagueness* and *ambiguity*. Vagueness is a concept representing the fuzziness concerning the degree to which the element of a set belongs to the set. Ambiguity is related to fuzziness of the value. From this point of view, fuzzy random variables are considered as the concepts dealing with ambiguity of the realization of a random variable since the realization of a random variable is fuzzy. On the other hand, *fuzzy event*, which was introduced by Zadeh [33], is the concept related to vagueness of the realization of a random variable since the realization is not fuzzy but crisp, and the degree to which an element belongs to a fuzzy set is imprecise. Dubois et al. [7] considered possibilistic programming which is based on the possibility theory introduced by Zadeh [34].

Katagiri et al. [16] considered a linear programming problem where the right-hand side of a constraint is a fuzzy random variable. They first introduced a possibilistic and stochastic programming approach to fuzzy random programming problems by noting that the degree of possibility that the constraint is satisfied varies randomly.

In this paper, we shall develop the idea to the case where the coefficients of an objective function are fuzzy random variables. Considering the vagueness of the decision maker's judgment, the fuzzy goal such that the objective function value is substantially smaller than g_1 is introduced. The fuzzy goal is characterized by the following membership function:

$$\mu_{\tilde{G}}(y) = \begin{cases} 1, & y \leq g_1 \\ g(y), & g_1 \leq y \leq g_0 \\ 0, & g_0 \leq y, \end{cases}$$

where g is a strictly decreasing function. Then, a degree of possibility that the objective function value attains the fuzzy goal $\tilde{G}$ is given as follows:

$$\Pi_{\tilde{\bar{Y}}(\omega)}(\tilde{G}) = \sup_{y} \min \left\{ \mu_{\tilde{\bar{Y}}(\omega)}(y), \mu_{\tilde{G}}(y) \right\}. \tag{2}$$

It should be noted here that the value of $\Pi_{\tilde{Y}(\omega)}(\tilde{G})$ varies randomly due to the randomness of $\mu_{\tilde{\bar{Y}}(\omega)}(y)$. Assuming that the decision maker hopes to maximize the value of $\Pi_{\tilde{Y}(\omega)}(\tilde{G})$, we take stochastic programming approaches in order to handle the randomness of the degree of possibility.

3.1 Probability maximization model using a possibility measure In stochastic programming, Beale [1] and Dantzig [6] introduced two-stage problems; Charnes et al. [4] introduced several stochastic programming model such as the expected optimization model, the variance minimization model and the probability maximization model. Kataoka [23] and Geoffrion [8] separately considered another stochastic programming model which is to optimize a satisficing level under the condition that the objective function value is better than the satisficing level.

In this paper, we consider the following problem which is based on the probability maximization model:

$$\left.\begin{array}{rl}\text{maximize} & Pr\left(\omega\middle|\Pi_{\tilde{\bar{Y}}(\omega)}(\tilde{G}) \geq h\right)\\ \text{subject to} & \boldsymbol{x} \in X,\end{array}\right\} \tag{3}$$

where h $(0 < h < 1)$ is a satisficing level for the degree of possibility with respect to a fuzzy goal. This problem is to maximize the probability that the degree of possibility is greater than or equal to a satisficing level h.

For any elementary event ω, $\Pi_{\tilde{\bar{Y}}(\omega)}(\tilde{G}) \geq h$ is transformed as

$$\sup_y \min\left\{\mu_{\tilde{\bar{Y}}(\omega)}(y),\ \mu_{\tilde{G}}(y)\right\} \geq h. \tag{4}$$

By using the properties of functions $\mu_{\tilde{\bar{Y}}(\omega)}$ and $\mu_{\tilde{G}}$, it is easy to show that if an element y satisfies the following in equation:

$$\min\left\{\mu_{\tilde{\bar{Y}}(\omega)}(y),\ \mu_{\tilde{G}}(y)\right\} \geq h,$$

then there exists an element $y^* \in [0, \bar{\boldsymbol{d}}(w)\boldsymbol{x}]$ such that

$$\min\left\{\mu_{\tilde{\bar{Y}}(\omega)}(y^*),\ \mu_{\tilde{G}}(y^*)\right\} \quad \geq \min\left\{\mu_{\tilde{\bar{Y}}(\omega)}(y),\ \mu_{\tilde{G}}(y)\right\} \geq h.$$

Hence, relation (4) becomes equivalent to

$$\begin{aligned}
&\sup_{y \leq \bar{\boldsymbol{d}}(w)\boldsymbol{x}} \min\left\{\mu_{\tilde{\bar{Y}}(\omega)}(y),\ \mu_{\tilde{G}}(y)\right\} \geq h\\
&\Longleftrightarrow \exists\, y : \mu_{\tilde{\bar{Y}}(\omega)}(y) \geq h,\ \mu_{\tilde{G}}(y) \geq h, \quad y \leq \bar{\boldsymbol{d}}(w)\boldsymbol{x}\\
&\Longleftrightarrow \exists\, y : \left\{\begin{array}{l} L\left(\dfrac{\bar{\boldsymbol{d}}(\omega)\boldsymbol{x} - y}{\boldsymbol{\alpha x}}\right) \geq h, \quad y \leq \bar{\boldsymbol{d}}(w)\boldsymbol{x}\\ \mu_{\tilde{G}}(y) \geq h.\end{array}\right.
\end{aligned}$$

Since L is a decreasing function on $[0, 1]$, the above relations are equivalent to

$$\begin{aligned}
&\exists\, y \leq \bar{\boldsymbol{d}}(w)\boldsymbol{x} : \{\bar{\boldsymbol{d}}(\omega) - L^*(h)\boldsymbol{\alpha}\}\boldsymbol{x} \leq y \leq \mu^*_{\tilde{G}}(h)\\
\Longleftrightarrow\quad &\{\bar{\boldsymbol{d}}(\omega) - L^*(h)\boldsymbol{\alpha}\}\boldsymbol{x} \leq \mu^*_{\tilde{G}}(h),
\end{aligned} \tag{5}$$

where $L^*(h)$ and $\mu^*_{\tilde{G}}(h)$ are pseudo inverse functions defined as

$$\begin{aligned}
L^*(h) &= \sup\{r | L(r) > h,\ r \geq 0\},\\
\mu^*_{\tilde{G}}(h) &= \sup\{r | \mu_{\tilde{G}}(r) \geq h\}.
\end{aligned}$$

Subtracting the expectation value $E\left(\{\bar{\boldsymbol{d}} - L^*(h)\boldsymbol{\alpha}\}\boldsymbol{x}\right)$ from both sides of the inequality (5) and dividing all by this deviation $\sqrt{Var\left(\{\bar{\boldsymbol{d}} - L^*(h)\boldsymbol{\alpha}\}\boldsymbol{x}\right)}$, where Var denotes the variance operator, we obtain

$$Pr\left(\Pi_{\tilde{\bar{Y}}}(\tilde{G}) \geq h\right) \Longleftrightarrow Pr\left(\frac{\left\{\bar{\boldsymbol{d}} - \boldsymbol{\mathcal{M}}^d\right\}\boldsymbol{x}}{\sqrt{\boldsymbol{x}^t V x}} \leq \frac{-\left\{\boldsymbol{\mathcal{M}}^d - L^*(h)\boldsymbol{\alpha}\right\}\boldsymbol{x} + \mu^*_{\tilde{G}}(h)}{\sqrt{\boldsymbol{x}^t V \boldsymbol{x}}}\right). \tag{6}$$

By noting that the left-hand side of the above inequality becomes a standard normal random variable, problem (3) is transformed into the following deterministic equivalent problem:

$$\left.\begin{array}{ll}\text{maximize} & z^{\Pi}(\boldsymbol{x}) = \dfrac{-\left\{\boldsymbol{\mathcal{M}}^d - L^*(h)\boldsymbol{\alpha}\right\}\boldsymbol{x} + \mu^*_{\tilde{G}}(h)}{\sqrt{\boldsymbol{x}^t V \boldsymbol{x}}} \\ \text{subject to} & \boldsymbol{x} \in X.\end{array}\right\} \tag{7}$$

It should be noted here that this problem is a nonlinear maximum spanning tree problem.

3.2 Probability maximization model using a necessity measure In the preceding section, we have considered a model using a possibility measure, which is useful in making a decision from an optimistic viewpoint. This section is devoted to investigating a model using a necessity measure, which is based on a risk aversion model:

$$\left.\begin{array}{ll}\text{maximize} & Pr\left(\omega \middle| N_{\tilde{\bar{Y}}(\omega)}(\tilde{G}) \geq h\right) \\ \text{subject to} & \boldsymbol{x} \in X,\end{array}\right\} \tag{8}$$

where $N_{\tilde{\bar{Y}}(\omega)}$ represents the degree of necessity that the objective function value satisfies the fuzzy goal and is expressed as

$$N_{\tilde{\bar{Y}}(\omega)}(\tilde{G}) = \inf_y \max\left\{1 - \mu_{\tilde{\bar{Y}}(\omega)}(y), \mu_{\tilde{G}}(y)\right\}. \tag{9}$$

Then, $N_{\tilde{\bar{Y}}(\omega)}(\tilde{G}) \geq h$ implies

$$\inf_y \max\{1 - \mu_{\tilde{\bar{Y}}(\omega)}(y),\ \mu_{\tilde{G}}(y)\} \geq h. \tag{10}$$

By using the properties of functions $\mu_{\tilde{\bar{Y}}(\omega)}$ and $\mu_{\tilde{G}}$, it is easy to show that if an element y satisfies

$$\max\{1 - \mu_{\tilde{\bar{Y}}(\omega)}(y),\ \mu_{\tilde{G}}(y)\} \geq h,$$

then there exists an element $y^* \geq \bar{\boldsymbol{d}}(\omega)\boldsymbol{x}$ such that

$$\max\left\{1 - \mu_{\tilde{\bar{Y}}(\omega)}(y^*),\ \mu_{\tilde{G}}(y^*)\right\} \leq \max\left\{1 - \mu_{\tilde{\bar{Y}}(\omega)}(y),\ \mu_{\tilde{G}}(y)\right\} \leq h.$$

Hence, relation (10) becomes equivalent to

$$\begin{aligned} & \inf_{y \geq \bar{\boldsymbol{d}}(w)\boldsymbol{x}} \max\left\{1 - \mu_{\tilde{\bar{Y}}(\omega)}(y),\ \mu_{\tilde{G}}(y)\right\} \geq h \\ \iff \quad & \forall y \geq \bar{\boldsymbol{d}}(w)\boldsymbol{x} : 1 - \mu_{\tilde{\bar{Y}}(\omega)}(y) < h \Rightarrow \mu_{\tilde{G}}(y) \geq h \\ \iff \quad & \forall y \geq \bar{\boldsymbol{d}}(w)\boldsymbol{x} : 1 - R\left(\frac{y - \bar{\boldsymbol{d}}(\omega)\boldsymbol{x}}{\boldsymbol{\beta}\boldsymbol{x}}\right) < h \Rightarrow \mu_{\tilde{G}}(y) \geq h. \end{aligned}$$

Since R is a strictly decreasing function on $[0, 1]$ and $\mu_{\tilde{G}}(y)$ is a decreasing function on $[0, +\infty)$, the above relations are equivalent to

$$\begin{aligned} \iff \quad & \forall y \geq \bar{\boldsymbol{d}}(w)\boldsymbol{x} : y < \left\{\bar{\boldsymbol{d}}(\omega) + R^*(1-h)\boldsymbol{\beta}\right\}\boldsymbol{x} \Rightarrow y \leq \mu^*_{\tilde{G}}(h) \\ \iff \quad & \left\{\bar{\boldsymbol{d}}(\omega) + R^*(1-h)\boldsymbol{\beta}\right\}\boldsymbol{x} \leq \mu^*_{\tilde{G}}(h), \end{aligned}$$

where

$$R^*(h) \quad = \quad \sup\{r | R(r) > h,\ r \geq 0\}.$$

Accordingly, it holds that

$$\frac{\left\{\bar{\boldsymbol{d}} - \boldsymbol{\mathcal{M}}^d\right\}\boldsymbol{x}}{\sqrt{\boldsymbol{x}^t V \boldsymbol{x}}} \leq \frac{-\left\{\boldsymbol{\mathcal{M}}^d + R^*(1-h)\boldsymbol{\beta}\right\}\boldsymbol{x} + \mu^*_{\tilde{G}}(h)}{\sqrt{\boldsymbol{x}^t V \boldsymbol{x}}}.$$

Then, since the left-hand side of the above inequality is the standard normal random variable, problem (8) is equivalently transformed into

$$(11) \qquad \left.\begin{array}{ll} \text{maximize} & z^N(\boldsymbol{x}) \triangleq \dfrac{-\left\{\boldsymbol{\mathcal{M}}^d + R^*(1-h)\boldsymbol{\beta}\right\}\boldsymbol{x} + \mu^*_{\tilde{G}}(h)}{\sqrt{\boldsymbol{x}^t V \boldsymbol{x}}} \\ \text{subject to} & \boldsymbol{x} \in X. \end{array}\right\}$$

Here, it should be noted that problem (11) is a nonlinear maximum spanning tree problem, and that this problem cannot be strictly solved by some polynomial-time algorithm such as the Krukal method or the Prim method like linear cases.

4 GA-based solution algorithms As described in the previous section, the problem to be solved is a nonlinear maximum spanning tree problem. Zhou and Gen [35] considered a quadratic minimum spanning tree problem, in which the objective function is a quadratic form. Although they originally proposed a genetic algorithm for solving the quadratic minimum spanning tree problem, the proposed algorithm can be directly applicable for solving nonlinear maximum spanning tree problems. In this section, we consider genetic algorithm (GA) approaches inspired from the work of Zhou and Gen [35]. Their approach uses Prüfer number for solution encoding, uniform crossover and mutation, and mixed strategy with $(\mu + \lambda)$ selection and roulette wheel selection. First, we briefly describe this genetic algorithm.

The choice of an encoding solution is one of the most important steps 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 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 [28]. 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 are one of the most efficient encoding methods for spanning trees in genetic algorithm search, in the sense that they are unbiased, cover the hall space of spanning trees and represent only spanning trees. However, this representation method has a drawback that an offspring with a tree may not be in the neighborhood of the parent tree for a small change in a parent P which is represented by a Prüfer number,

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 whose 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 offspring. If there are no μ different chromosomes available, then the vacant pool of population are filled up with roulette wheel selection. In order to vary the selection method in this GA, we consider two other selection methods; sharing and scaling selection methods.

5 Tabu search for solving FRMST problems Tabu search [9, 10] is a metaheuristic method that has proven to be very effective for many combinatorial optimization problems [5]. It has been illustrated that the computational results on benchmark problem were quite positive and that TS-based methods improved the best known solution for the large benchmark instances in the wide varieties of combinatorial optimization problems. As one of the most effective TS algorithm, Hanafi [12] considered a TS algorithm based on strategic oscillation. Although they showed that the proposed algorithm is well performed for the 0-1 multidimensional knapsack problems, it is not directly applicable for solving nonlinear minimum (maximum) spanning tree problems.

In this section, in order to propose a more effective algorithm than existing GA-based algorithms [35], we shall construct a solution algorithm based on TS incorporating strategic oscillation. Starting from an initial spanning tree, the improvement strategy, which consists of exchanging a pair of edges, generates the neighborhood of the current solution. In order to prevent cycling between the same solutions, certain exchanges which are called "moves" can be forbidden, earning them the status of "tabu move". 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 aspiration criterion permits certain moves on the tabu list to overcome any tabu status. Strategic oscillation was originally introduced to provide an effective interplay between intensification and diversification over the intermediate to long term. In the proposed algorithm, we used strategic oscillation to intensively explore the region around the current neighborhood. In addition to a short-term memory, we use a residence frequency memory as a long-term memory. A diversification procedure, using the residence frequency memory function, will lead to the exploration of region of the solution space not previously visited. On the other hand, an intensification procedure 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.

The essential features that have been considered in building a TS algorithm for solving a fuzzy random minimum spanning tree problem are: generating an initial solution, the neighborhood structure, the improvement strategies, short-term and long-term memories, oscillation strategy, intensification by an elite solution set, diversification procedure, termination criterion. The details of these concepts or procedures are as follows:

a. Initial solution Let $SCC(i)$ denote a Set of Connected Component that consists of i edges. In this paper, we indentify $SSC(i)$ with an m-dimensional decision vector $\boldsymbol{x}$ whose i elements are equal to 1s. 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, in general, a $SCC(k+1)$ is constructed by adding an edge $e \in \text{argmax}\{z(SCC(k)) + e') - z(SCC(k)) | e' \in E_{NC}(SCC(k))\}$ to the current $SCC(k)$ under construction, where $E_{NC}(SCC(k))$ is defined as follows:

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

b. Neighborhood structure Let T be a set of edges which forms a spanning tree, and let $\mathcal{T}$ be a class of all possible spanning trees in a given graph. The neighborhood $N(T)$ consists of all possible spanning trees which can be generated by removing an edge $e \in T$ and by adding an edge from the set $E_{NH}(T-e) \setminus \{e\}$, where $E_{NH}(T-e)$ is defined as follows:

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

c. Improvement strategy In order to improve the current solution $\boldsymbol{x}^c$ corresponding to the current spanning tree T^c, there are two major improvement strategies. One is a first improvement strategy, which scans the neighborhood $N(T^c)$ and chooses the initially-discovered spanning tree T^f corresponding to the solution $\boldsymbol{x}^f$ such that $z(\boldsymbol{x}^f) > z(\boldsymbol{x}^c)$. The other is a best improvement strategy, which exhaustively explores the neighborhood and returns one of the solutions with the lowest objective function value.

At the beginning, we use the first improvement strategy. If a better solution cannot be found, we switch to the best improvement strategy.

d. 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 uses only one tabu list denoted by $TabuList$. The attribute the tabu list stores is the index of the edges that were recently added or removed. Every move consists of two steps: the first step is to remove one edge $e \in T^c$ from the current spanning tree T^c, and the second step is to add 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.

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

f. Strategic oscillation procedure The strategic oscillation approaches to a boundary by adding or removing edges, where a set of spanning trees is regarded as the boundary in our problems. 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 reformed. 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. In order to explore the search space efficiently, we use two kinds of depth: small depth and large depth. First, the Oscillation Strategy with Small Depth (OSSD) is performed, and if OSSD cannot find a better solution in $NISmall$ iterations, then the Oscillation Strategy with Large Depth (OSLD) begins to explore the search space. If the OSLD finds a better solution, then the strategy is switched to the OSSD again. If the OSLD cannot find a better solution in $NILarge$ iterations, the strategic oscillation procedure is terminated.

The rules of adding and removing edges are described as follows.

Edge addition rule The strategic oscillation procedure begins at adding an edge $e \in \operatorname{argmax}\{z(T^c + e') - z(T^c) | e' \in E \backslash Tabulist\}$ to the current minimum spanning tree T. Then, the constructed connected component can be expressed by $SCC(n)$ because any spanning tree T is represented with $SSC(n-1)$. In the same way, several edges are added by using the rule that $SCC(k+1)$ is constructed by adding an edge $e \in \text{argmax}\{z(SCC(k)+e') - z(SCC(k))|e' \in E\backslash Tabulist\}$ to $SCC(k)$. It should be noted here that the resulting connected component includes some cycles.

Edge removal rule The edge removal rule is applied to reform a spanning tree by removing several edges of the connected component with some cycles. To be more specific, by the edge removal rule, $SCC(k-1)$ is constructed by removing an edge $e \in \text{argmax}\{z(SCC(k)-e')-z(SCC(k))|e' \in CY(k)\}$ from a cycle in the current $SCC(k)$ under construction, where $CY(k)$ denotes the set of elements $e' \in SCC(k)$ which are parts of cycles in the component $SCC(k)$. This procedure is continued until a spanning tree, namely, $SCC(n-1)$ is reformed.

g. 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 a residence frequency memory, which keeps track of the number of iterations where edges has been a part of the solution. By using the residence frequency memory, we provide the following diversification and intensification processes.

i. Diversification procedure The diversification procedure begins at the situation that some spanning tree is reformed. Then, several edges that constructs the spanning tree, each of which have been a part of the explored solutions for a long time, are removed from the spanning tree. Next, another spanning tree is reformed by adding the edges which have not been involved in the explored solutions so far through the *edge addition rule*. If the strategic oscillation procedure is iterated in Max_k times, then the intensification procedure is started.

ii. Intensification procedure using an elite solution set The intensification procedure begins at the condition that no edge is selected. First, a connected component is constructed by continuing to select the edges that occur frequently in the elite solutions. The selected edges are never removed during the procedure. After constructing a connected component, the process of adding edges, except for ones that are not in most of elite solutions, are continued by the edge addition rule until a spanning tree is reformed.

h. Termination criterion The counter $UNRIter$ counts the iterations where the best solution T^b is not renewed. 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.

Now, we are ready to describe the details of the proposed algorithm based on tabu search. Let $\boldsymbol{x}^c$ be the current solution and T^c its corresponding spanning tree, and let $\boldsymbol{x}^b$ and T^b be the best solution and its corresponding spanning tree, respectively. In the proposed algorithm, we use the following parameters:

$NISmall$: Number of iterations in the small depth procedure.
MAX_Small: Threshold of the counter $NISmall$.
$NILarge$: Number of iterations in the large depth procedure.
MAX_Large: Threshold of the counter $NISmall$.
$UNRIter$: Counts the iterations where the best solution is not renewed.
MAX_Iter: Threshold of the counter $UNRIter$.
Max_k: Threshold of the counter, k, of the oscillation strategy.

The proposed 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 $NISmall = NILarge = UNRIter = k = 0$. Generate an initial solution $\boldsymbol{x}^0$ corresponding to an initial spanning tree T^0. Set $\boldsymbol{x}^c := \boldsymbol{x}^0$ and $\boldsymbol{x}^b := \boldsymbol{x}^0$. Set $k = 0$.

Step 1 (Improvement)
Improve the obtained solution by the *improvement strategy*. Set $\boldsymbol{x}^b := \boldsymbol{x}^c$.

Step 2 (Oscillation Strategy with Small Depth)
Set $k := k + 1$. If $NISmall > MAX\_Small$, then go to step 4. Otherwise, add a_1 edges among $N(T^c)$ by using the *edge addition rule* and continue to remove one of the edges in a cycle by using the *edge removal rule* until a spanning tree is reformed.

Step 3 If $z(\boldsymbol{x}^c) > z(\boldsymbol{x}^b)$, then set $\boldsymbol{x}^b := \boldsymbol{x}^c$ and $NISmall = 0$, and return to step 2. Otherwise, set $NISmall = NISmall + 1$ and return to step 2.

Step 4 (Oscillation Strategy with Large Depth)
If $NILarge > MAX\_Large$, then go to step 6. Otherwise, add a_2 $(a_1 < a_2)$ edges among $N(T^c)$ by the edge addition rule and continue to remove one of the edges in a cycle by the edge remove rule until a spanning tree is reformed. Improve the current solution by the improvement strategy.

Step 5 If $z(\boldsymbol{x}^c) > z(\boldsymbol{x}^b)$, then set $NILarge = NISmall = 0$ and return to step 2. Otherwise, set $\boldsymbol{x}^b := \boldsymbol{x}^c$, $NILarge := NILarge + 1$, and return to step 4.

Step 6 If $k > Max\_k$, then go to step 8. Otherwise, go to step 7.

Step 7 (Diversification)
Remove a_3 edges in T^c that are resident for a long time in spanning trees. Slap a long tabu tenure to the removed edges. Add the edges whose resident time are short so as not to make a cycle until a spanning tree is reformed. Return to step 1.

Step 8 (Intensification by elite solutions)
Set $k = 0$. Construct a set of connected components by adding edges that are in most of elite solutions. Add edges, except for the edges that are not in most of elite solutions, by the *edge addition rule* until a spanning tree is reformed. Improve the obtained solution by the improvement strategy. If $z(\boldsymbol{x}^c) > z(\boldsymbol{x}^b)$, then set $\boldsymbol{x}^b := \boldsymbol{x}^c$, $k = UNRIter = 0$ and return to step 2. Otherwise, set $UNRIter := UNRIter + 1$ and go to step 9.

Step 9 If $UNRIter > Max\_Iter$, then terminate. Otherwise, return to step 1.

6 Numerical experiments Let $\mathcal{G}$ 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 $\mathcal{G}$ represented by m-dimensional $0-1$ decision vectors. In this section we apply the proposed TS method and GAs described in this paper, to solve FRMST problem (7).

The experiments are conducted on complete undirected graphs with different number of nodes, and the values of vectors $\boldsymbol{\mathcal{M}}^d$ and $\boldsymbol{\alpha}$ are generated randomly. The parameter h is set as 0.7.

In the GA approaches, the parameters are set as follows: crossover probability $p_c = 0.4$, mutation probability $p_m = 0.01$, population size 120, and maximum number of generations

1000. Iteration parameters in GA and TS algorithms are set so as to make their run times close to each other, and therefore a comparison of their performances make sense.

We have run each experiment ten times. The following tables show the experimental results of the above problem by using the TS method and GAs described in this paper. These algorithms are coded in C++ programming language and implemented on a computer with a CPU Celeron 1.7GHz and RAM 252MB. The expressions GA-Sc, GA-Sh and GA-$(\mu+\lambda)$ denote, respectively, the GA with scaling selection method, sharing selection method and the mixed strategy with $(\mu+\lambda)$ selection and roulette wheel selection proposed in the study by Zhou and Gen [35].

Table 1. Best objective function values.

Nodes	TS	GA-Sc	GA-Sh	GA-$(\mu+\lambda)$
6	0.729	0.729	0.729	0.729
7	1.141	1.141	1.141	1.141
8	2.062	2.062	2.062	1.870
9	1.543	1.412	1.432	1.372
10	1.485	1.399	1.346	1.275
15	2.367	1.600	1.576	1.719
30	4.958	2.046	2.200	2.042
50	6.865	1.702	1.739	1.925

Table 2. Average objective function values.

Nodes	TS	GA-Sc	GA-Sh	GA-$(\mu+\lambda)$
6	0.729	0.729	0.729	0.729
7	1.141	1.141	1.141	1.141
8	2.062	2.000	1.931	1.774
9	1.543	1.322	1.302	1.238
10	1.485	1.320	1.277	1.228
15	2.367	1.505	1.488	1.448
30	4.918	1.946	1.948	1.870
50	6.771	1.650	1.668	1.670

Table 3. Average computational time in second.

Nodes	TS	GA-Sc	GA-Sh	GA-$(\mu+\lambda)$
6	–	0.3	1.2	0.6
7	–	0.5	1.5	0.9
8	0.1	0.7	1.8	1.1
9	0.13	1	2.2	1.5
10	0.5	1.3	2.7	2
15	4	5	7	7.2
30	82	85	92	120
50	1359	1095	1098	1374

The sign ”–” means a very short time.

From these computational results, we remark that GAs with the three selection methods are quite uncompetitive with the TS method for solving the problem. In addition, the best and average values obtained by TS are very close or the same for number of nodes less than or equal to 15. It is observed that the proposed TS method has generally better performances than the existing methods based on genetic algorithms.

As a more general problem than minimum spanning tree problem, there exists a k-minimum spanning tree problem [11], which is to seek a subtree with exact k edges whose objective function is minimal. Blum and Blesa [2] proposed several approximate solution methods and showed that a TS-based method is the best in case of large k. Realizing that the minimum spanning tree corresponds to the largest $k(= n - 1)$-minimum spanning tree problem, our observation that our TS algorithm is better than GA-based ones is consistent with or supported by the experimental results shown by Blum and Blesa. Thus, we can conclude that our TS algorithm is better than the existing GA-based algorithms.

7 Conclusion In this paper, we have considered fuzzy random minimum spanning tree problem. Introducing a fuzzy goal, we have formulated the problem to maximize the probability that the degree of possibility or necessity that an objective function satisfies the fuzzy goal. It has been shown that the problem can be transformed into a deterministic equivalent nonlinear maximum spanning problem. In order to solve the problem, we have constructed a TS algorithm based on oscillation strategy, intensification by elite solution set and diversification by residence frequency and so on.

In the future, we will extend the proposed method to other decision making models for fuzzy random minimum spanning tree problems. For instance, we will consider the case where not only the probability but also the satisficing level (denoted by h) for the degree of possibility or necessity. As another future work, we will try to solve the problem of minimizing the variance of the degree of possibility or necessity. Since such a problem includes the constraint with respect to the expected degree of possibility or necessity, we need to extend our method in order to deal with the constraint by changing a part of the oscillation strategy.

References

[1] Beale, M.L. 1995. On optimizing a convex function subject to linear inequalities, *Journal of Royal Statistics Society*, vol. 17:173–184.

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

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

[4] Charnes A. and Cooper, W.W. 1963. Deterministic equivalents for optimizing and satisficing under chance constraints, *Operations Research*, vol. 11:18–39.

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

[6] Dantzig, G.B. 1995. Linear programming under uncertainty, *Management Science*, vol.1:197–206.

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

[8] Geoffrion, A.M. 1967. stochastic programming with aspiration or fractile criteria, *Management Science*, vol. 13:672–679.

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

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

[11] Hamacher, H.W., Jörnsten, K., Maffioli, F. 1991. Weighted k-cardinality trees, *Technical Report 91.023*, Politecnico di Milano, Dipartimento di Elettronica, Italy.

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

[13] Ishii, H., and Nishida, T. 1983. Stochastic bottleneck spanning tree problem, *Networks*, vol. 13:443-449.

[14] Ishii, H., Shiode, S., Nishida, T. and Namasuya, Y. 1981. Stochastic spanning tree problem, *Discrete Applied Mathematics*, vol. 3:263–273.

[15] Ishii, H., Shiode, S. and Nishida, T. 1981. Chance constrained spanning tree problem, *Journal of the Operations Research Society of Japan*, vol. 24:147–157.

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

[17] Katagiri, H. and Ishii, H. 2000. Chance constrained bottleneck spanning tree problem with fuzzy random edge costs, *Journal of the Operations Research Society of Japan*, vol. 43:128–137.

[18] 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:1912–1919.

[19] Katagiri, H. and Sakawa, M. 2011. Interactive multiobjective fuzzy random programming through the level set-based probability model, *Information Sciences*, vol. 181:1641–1650.

[20] Katagiri, H., Sakawa, M. and Ishii, H. 2004. Fuzzy random bottleneck spanning tree problems using possibility and necessity measures, *European Journal of Operational Research*, vol. 152:88–95.

[21] Katagiri, H., Sakawa, M., Kato, K. and Nishizaki, I. 2004. A fuzzy random multiobjective 0-1 programming based on the expectation optimization model using possibility and necessity measures, *Mathematical and Computer Modelling*, vol. 40:411–421.

[22] 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. 188:530–539.

[23] Kataoka, S. 1963. A stochastic programming model, *Econometrica*, vol. 31:181-196.

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

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

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

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

[28] Prüfer, H., 1918. Neuer beweis eines satzes uber permutationen, Arch. Math. Phys. vol. 27:742-744.

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

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

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

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

[33] Zadeh, L.A. 1968. Probability measure of fuzzy events, *Journal of Mathematical Analysis and Applications*, vol. 23:421–427.

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

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

Faculty of Engineering, Hiroshima University, 1-4-1, Kagamiyama, Higashi-Hiroshima, 739-8527 JAPAN

E-mail : katagiri-h@hiroshima-u.ac.jp