A tabu search algorithm for fuzzy random minimum spanning tree problems through the probability maximization model using possibility and necessity measures

El Bekkaye Mermri[1,2], Hideki Katagiri[1], Masatoshi Sakawa[1], and Kosuke Kato[1]

Graduate School of Engineering, Hiroshima University[1]
Kagamiyama 1-4-1, Higashi-Hiroshima, Hiroshima, 739-8527 Japan[1]
email: mermri@msl.sys.hiroshima-u.ac.jp[2]

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 maximize the probability that the degree is greater than or equal to a safisficing level. It is shown that the problem is transformed into the deterministic equivalent one, which is a nonlinear minimum ratio spanning tree problem. In order to solve the problem, a Tabu Search (TS) algorithm is developed.

I. 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 [1], Prim [2] and Sollin [3]. 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.

A fuzzy random variable was first defined by Kwakernaak [4] and Puri et al. [5]. Recently, some researchers [6–9] considered fuzzy random linear programming problems. 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, which is based on the idea provided in [9]. 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 maximize the probability that the degree is greater than or equal to some satisficing level. As is shown later, the formulated problem is transformed into the deterministic equivalent nonlinear minimum ratio spanning tree problem, which is generally an NP-hard problem.

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 [10,11] is one of the most efficient solution methods for combinatorial optimization problems. Cunha et al. [13] applied a tabu search algorithm to water network optimization. Blum et al. [14] 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.

II. MST problem with fuzzy random edge costs

Consider a connected undirected graph $\mathcal{G} = (V, E)$, 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 $\boldsymbol{x} =$

$(x_1, x_2, \ldots, x_m)$ be a vector defined by

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

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

$$\left. \begin{array}{l} \min \tilde{\bar{\boldsymbol{C}}}\boldsymbol{x} \\ \text{s. t. } \boldsymbol{x} \in X \end{array} \right\} \quad (1)$$

where $\tilde{\bar{\boldsymbol{C}}} = (\tilde{\bar{C}}_1, \ldots, \tilde{\bar{C}}_n)$, $\boldsymbol{x} = (x_1, \ldots, x_n)^t$ and A is an $m \times n$ matrix (a_{ij}). 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 $\tilde{\bar{C}}_j(\omega)$:

$$\mu_{\tilde{\bar{C}}_j(\omega)}(t) = \begin{cases} L\left(\dfrac{\bar{d}_j(\omega) - t}{\bar{\alpha}_j(\omega)}\right) \ (t \leq \bar{d}_j(\omega), \ \forall \omega) \\ \\ R\left(\dfrac{t - \bar{d}_j(\omega)}{\bar{\beta}_j(\omega)}\right) \ (t > \bar{d}_j(\omega), \ \forall \omega), \end{cases}$$

where $L(t) \triangleq \max\{0, \ l(t)\}$ and $R(t) \triangleq \max\{0, \ r(t)\}$. The functions $l(t)$ and $r(t)$ are strictly decreasing functions satisfying $l(0) = r(0) = 1$. The parameters $\bar{d}_j, \bar{\alpha}_j$ and $\bar{\beta}_j, \ j = 1, \ldots, n$ are the normal random variables with mean m_j^d, m_j^α and m_j^β. The variance-covariance matrix of the vector $q = (d, \alpha, \beta)$ is denoted by V,

$$V = \begin{bmatrix} V_1 & V_4 & V_5 \\ V_4^T & V_2 & V_6 \\ V_5^T & V_6 & V_3 \end{bmatrix},$$

where $V_i, \ i = 1, \cdots, 6$ are an $m \times m$ matrices and V_i^T denotes the transposition matrix of V_i.

By applying the calculation formula [18] with respect to L - R fuzzy numbers based on the extension principle [19] to the fuzzy number $\tilde{\bar{Y}}(\omega)$ for each ω, it is easily shown that $\tilde{\bar{Y}}$ is a fuzzy random variable with the following membership function:

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

III. Possibilistic programming approach

In problem (1), the total edge weights 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* [16]. 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 [17], is the concept related to vagueness of the realization of a random variable because the realization is not fuzzy but crisp, and the degree to which an element belongs to a fuzzy set is imprecise.

Dubois et al. [18] considered possibilistic programming which is based on the possibility theory introduced by Zadeh [19], and Inuiguchi et al. [20] developed modality constrained programming models. Katagiri et al. [9] 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 developed 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:

$$\mathit{\Pi}_{\tilde{\bar{Y}}}(\tilde{G}) = \sup_y \min\left\{\mu_{\tilde{\bar{Y}}}(y), \mu_{\tilde{G}}(y)\right\} \quad (2)$$

where $\mathit{\Pi}$ is a possibility measure [19].

IV. Probability maximization model using a possibility measure

If a decision maker wish to maximize a degree of possibility that the fuzzy goal is satisfied, problem (1) is reformulated as the following problem:

$$\left. \begin{array}{l} \max \mathit{\Pi}_{\tilde{\bar{Y}}}(\tilde{G}) \\ \text{s. t. } \boldsymbol{x} \in X \end{array} \right\} \quad (3)$$

Since the value of $\mathit{\Pi}_{\tilde{\bar{Y}}}(\tilde{G})$ varies randomly due to the randomness of $\mu_{\tilde{\bar{Y}}}(y)$, we regard problem (3) as a stochastic programming problem. In stochastic programming, Dantzig [21] and Beale [22] introduced two-stage problems; Charnes and Cooper [23] introduced several stochastic programming model such as the expected optimization model, the variance minimization model and the probability maximization model. Kataoka [24] and Geoffrion [25] 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:

$$\left.\begin{array}{l} \max \; Pr\left(\omega \middle| \mathit{\Pi}_{\tilde{\bar{Y}}(\omega)}(\tilde{G}) \geq h\right) \\ \text{s. t. } \boldsymbol{x} \in X \end{array}\right\} \tag{4}$$

where h 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, $\mathit{\Pi}_{\tilde{\bar{Y}}(\omega)}(\tilde{G}) \geq h$ is transformed as follows:

$$\begin{aligned} & \sup_y \min\left\{\mu_{\tilde{\bar{Y}}(\omega)}(y),\ \mu_{\tilde{G}}(y)\right\} \geq h \\ & \Longleftrightarrow \exists\, y : \mu_{\tilde{\bar{Y}}(\omega)}(y) \geq h,\ \mu_{\tilde{G}}(y) \geq h \\ & \Longleftrightarrow \exists\, y : L\left(\frac{\bar{\boldsymbol{d}}(\omega)\boldsymbol{x} - y}{\bar{\boldsymbol{\alpha}}(\omega)\boldsymbol{x}}\right) \geq h,\ R\left(\frac{y - \bar{\boldsymbol{d}}(\omega)\boldsymbol{x}}{\bar{\boldsymbol{\beta}}(\omega)\boldsymbol{x}}\right) \geq h, \\ & \qquad\quad \mu_{\tilde{G}}(y) \geq h \\ & \Longleftrightarrow \exists\, y : \{\bar{\boldsymbol{d}}(\omega) - L^*(h)\bar{\boldsymbol{\alpha}}(\omega)\}\boldsymbol{x} \leq y \\ & \qquad\quad \leq \{\bar{\boldsymbol{d}}(\omega) + R^*(h)\bar{\boldsymbol{\beta}}(\omega)\}\boldsymbol{x},\ y \leq \mu^*_{\tilde{G}}(h) \\ & \Longleftrightarrow \{\bar{\boldsymbol{d}}(\omega) - L^*(h)\bar{\boldsymbol{\alpha}}(\omega)\}\boldsymbol{x} \leq \mu^*_{\tilde{G}}(h) \end{aligned}$$

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

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

Accordingly,

$$\begin{aligned} & Pr\left(\mathit{\Pi}_{\tilde{\bar{Y}}}(\tilde{G}) \geq h\right) \\ & \Longleftrightarrow Pr\left(\frac{\left\{\bar{\boldsymbol{d}} - L^*(h)\bar{\boldsymbol{\alpha}}\right\}\boldsymbol{x} - \left\{\boldsymbol{m}^d - L^*(h)\boldsymbol{m}^\alpha\right\}\boldsymbol{x}}{\sqrt{\boldsymbol{x}^t V_1 \boldsymbol{x} - 2L^*(h)\boldsymbol{x}^t V_4 \boldsymbol{x} + \{L^*(h)\}^2 \boldsymbol{x}^t V_2 \boldsymbol{x}}} \right. \\ & \qquad \left. \leq \frac{-\left\{\boldsymbol{m}^d - L^*(h)\boldsymbol{m}^\alpha\right\}\boldsymbol{x} + \mu^*_{\tilde{G}}(h)}{\sqrt{\boldsymbol{x}^t V_1 \boldsymbol{x} - 2L^*(h)\boldsymbol{x}^t V_4 \boldsymbol{x} + \{L^*(h)\}^2 \boldsymbol{x}^t V_2 \boldsymbol{x}}}\right). \end{aligned}$$

By noting that the left-hand side of the above inequality becomes a standard normal random variable, problem (4) is transformed into the following deterministic equivalent problem:

$$\left.\begin{array}{l} \max \dfrac{-\{\boldsymbol{m}^d - L^*(h)\boldsymbol{m}^\alpha\}\boldsymbol{x} + \mu^*_{\tilde{G}}(h)}{\sqrt{\boldsymbol{x}^t V_1 \boldsymbol{x} - 2L^*(h)\boldsymbol{x}^t V_4 \boldsymbol{x} + \{L^*(h)\}^2 \boldsymbol{x}^t V_2 \boldsymbol{x}}} \\ \text{s. t. } \boldsymbol{x} \in X \end{array}\right\} \tag{5}$$

V. Probability maximization model using a necessity measure

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

$$\left.\begin{array}{l} \max \; Pr\left(\omega \middle| N_{\tilde{\bar{Y}}(\omega)}(\tilde{G}) \geq h\right) \\ \text{s. t. } \boldsymbol{x} \in X \end{array}\right\} \tag{6}$$

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

$$N_{\tilde{\bar{Y}}(\omega)}(\tilde{G}) = \inf_y \max\left\{1 - \mu_{\tilde{\bar{Y}}(\omega)}(y), \mu_{\tilde{G}}(y)\right\} \tag{7}$$

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

$$\begin{aligned} & \inf_y \max\{1 - \mu_{\tilde{\bar{Y}}(\omega)}(y),\ \mu_{\tilde{G}}(y)\} \geq h \\ & \qquad \Longleftrightarrow \forall y : 1 - \mu_{\tilde{\bar{Y}}(\omega)}(y) < h \Rightarrow \mu_{\tilde{G}}(y) \geq h \\ & \qquad \Longleftrightarrow \forall y : \left\{\bar{\boldsymbol{d}}(\omega) - L^*(1-h)\bar{\boldsymbol{\alpha}}(\omega)\right\}\boldsymbol{x} < y \\ & \qquad\qquad < \left\{\bar{\boldsymbol{d}}(\omega) + R^*(1-h)\bar{\boldsymbol{\beta}}(\omega)\right\}\boldsymbol{x} \Rightarrow y \leq \mu^*_{\tilde{G}}(h) \\ & \qquad \Longleftrightarrow \left\{\bar{\boldsymbol{d}}(\omega) + R^*(1-h)\bar{\boldsymbol{\beta}}(\omega)\right\}\boldsymbol{x} \leq \mu^*_{\tilde{G}}(h). \end{aligned}$$

where

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

Accordingly, it holds that

$$\begin{aligned} & \frac{\left\{\bar{\boldsymbol{d}} + R^*(1-h)\bar{\boldsymbol{\beta}}\right\}\boldsymbol{x} - \left\{\boldsymbol{m}^d + R^*(1-h)\boldsymbol{m}^\beta\right\}\boldsymbol{x}}{\sqrt{\boldsymbol{x}^t V_1 \boldsymbol{x} + 2R^*(1-h)\boldsymbol{x}^t V_5 \boldsymbol{x} + \{R^*(1-h)\}^2 \boldsymbol{x}^t V_3 \boldsymbol{x}}} \\ & \leq \frac{-\left\{\boldsymbol{m}^d + R^*(1-h)\boldsymbol{m}^\beta\right\}\boldsymbol{x} + \mu^*_{\tilde{G}}(h)}{\sqrt{\boldsymbol{x}^t V_1 \boldsymbol{x} + 2R^*(1-h)\boldsymbol{x}^t V_5 \boldsymbol{x} + \{R^*(1-h)\}^2 \boldsymbol{x}^t V_3 \boldsymbol{x}}} \end{aligned}$$

Then, since the left-hand side of the above inequality is the standard normal random variable, problem (6) is equivalently transformed into the following:

$$\left.\begin{array}{l} \max \dfrac{-\{\boldsymbol{m}^d + R^*(1-h)\boldsymbol{m}^\beta\}\boldsymbol{x} + \mu^*_{\tilde{G}}(h)}{\sqrt{\boldsymbol{x}^t V_1 \boldsymbol{x} + 2R^*(1-h)\boldsymbol{x}^t V_5 \boldsymbol{x} + \{R^*(1-h)\}^2 \boldsymbol{x}^t V_3 \boldsymbol{x}}} \\ \text{s. t. } \boldsymbol{x} \in X. \end{array}\right\} \tag{8}$$

VI. 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 et al. [12] considered a TS algorithm based on strategic oscillation and showed that the proposed algorithm was well performed for the 0-1 multidimensional knapsack problems. Souza [15] proposed a GRASP with a path-relinking heuristics for a capacitated minimum spanning tree problem, which is based on a memory-based local search strategy. It was illustrated that the computational results on benchmark problem was quite positive and that the proposed method improved the best know solution for some of the largest benchmark problem.

In this section, we shall construct a solution algorithm based on TS incorporating strategic oscillation and path-relinking method. Starting from an initial spanning tree, the improvement strategy which consists of exchanging a pair of two edges generates the neighborhood of the current solution. In order to prevent cycling between the same solutions, certain exchanges which are call "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 produce 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 T^c and $\boldsymbol{x}^c$ be the current spanning tree and the corresponding solution, and let T^b and $\boldsymbol{x}^b$ be the best spanning tree and the corresponding solution. We denote the objective function value of (5) or (8) for $\boldsymbol{x}$ by $z(\boldsymbol{x})$. Then, a TS algorithm for solving fuzzy random minimum spanning tree problem is as follows:

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

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

Step 2 (Strategic oscillation, small depthj
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 remove rule* until an spanning tree is formed.

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

Step 4 iStrategic oscillationClarge depthj
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 formed. 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 $NILarge = NILarge + 1$ and return to step 4.

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

Step 7 iDiversificationj
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 formed. Return to step 1.

Step 8 iIntensification by elite solutionsj
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 formed. 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 iPath relinking methodj
If $UNRIter > Max\_Iter$, then terminate. Otherwise, generate an initial solution by the path relinking method and return to step 1.

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. In the preceding section, we describe the detail of those features.

A. 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:

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

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 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 $\boldsymbol{x}^c$, here are two major improvement strategies. One is a first improvement, which scans the neighborhood $N(T^c)$ of a current spanning tree T^c and chooses the first 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, 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 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 $e \in T^c$ from the current spanning tree T^c, and adding a different edge to $T^c - e$. $TabuList$ is the list to keep memory of the removed or added edges. 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 then 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

Strategic oscillation approaches by adding or removing edges to a boundary which is represented by a set of spanning trees. Instead of stopping 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 it is spanning. 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 is described as follows.

Edge addition rule Again, we use $SCC(i)$, a set of connected component that consist of i edges, defined in the previous section. Then, by the edge addition rule, $SCC(k+1)$ is constructed by adding an edge $e \leftarrow \text{argmin}\{z(SCC(k)+e') - z(SCC(k)) | e' \in E\}$ to the current $SCC(k)$ under construction.

Edge remove rule By the edge remove rule, $SCC(k-1)$ is constructed by removing an edge $e \leftarrow \text{argmax}\{z(SCC(k)-e') - z(SCC(k)) | e' \in SCC(k)\}$ from the current $SCC(k)$ under construction.

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 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.

1. Diversification procedure

The diversification procedure begins at the situation that some spanning tree is formed. Then it removes from the spanning tree a_3 edges which have been a part of spanning trees for a long time. Next, a spanning tree is again formed with the edges which have not been added so far by the *edge addition rule*. The diversification derives the search into a new region. Then, the strategic oscillation procedure begins at the new search region. If the strategic oscillation procedure is iterated in Max_k times, then the intensification procedure is started.

2. 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 selecting 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 ones that are not in most of elite solutions, are continued by the edge addition rule until a spanning tree is formed.

H. Path relinking method

Path relinking is initiated by selecting two solutions $\boldsymbol{x}'$ and $\boldsymbol{x}''$ from a collection of elite solutions produced during previous search phases. A path is then generated from $\boldsymbol{x}'$ to $\boldsymbol{x}''$, producing a solution sequence $\boldsymbol{x}' = \boldsymbol{x}'(1), \boldsymbol{x}'(2), \ldots,$ $\boldsymbol{x}'(r) = \boldsymbol{x}''$, where $\boldsymbol{x}'(i+1)$ is created from $\boldsymbol{x}'$ at each step by choosing a move that leaves the fewest number of moves remaining to reach $\boldsymbol{x}''$. Finally, once the path is completed, one or more of the solutions $\boldsymbol{x}''(i)$ is selected as a solution to initiate a new search phase.

I. Termination criterion

The counter $UNRIter$ counts the iterations 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.

Conclusion

In this paper, we have considered fuzzy random spanning tree problem. Introducing a fuzzy goal, we 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 was transformed into the deterministic equivalent nonlinear minimum ratio spanning programming problem. In order 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 try to extend and apply this method to the problems to minimize the variance of the degree of possibility or necessity. Since the problems include 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. J.B. Kruskal, Jr., On the shortest spanning subtree of a graph and traveling salesman problem, *Proceedings of ACM* 7/1, pp. 48–50, 1956.
2. R.C. Prim, Shortest connection networks and some generations, *Bell System Technical Journal*, vol. 36, pp. 1389–1401, 1957.
3. M. Sollin, Le trace de canalisation, in: *Programming, Games, and Transportation Networks*, C. Berge, A Ghouilla-Houri Eds. Wiley, New York, 1965.
4. H. Kwakernaak, Fuzzy random variable-1, *Information Sciences*, vol. 15, pp. 1–29, 1978.
5. M.L. Puri, and D.A. Ralescu, Fuzzy random variables, *Journal of Mathematical Analysis and Applications*, vol. 14, pp. 409–422, 1986.
6. G.-Y. Wang, and Z. Qiao, Linear programming with fuzzy random variable coefficients, *Fuzzy Sets and Systems*, vol. 57, pp. 295–311, 1993.
7. Z. Qiao, Y. Zhang and G. Wang, On fuzzy random linear programming, *Fuzzy Sets and Systems*, vol. 65, pp. 31–49, 1994.
8. M.K. Luhandjula, and M.M. Gupta, On fuzzy stochastic optimization, *Fuzzy Sets and Systems*, vol. 81, pp. 47–55, 1996.
9. H. Katagiri, and H. Ishii, Linear programming problem with fuzzy random constraint, *Mathematica Japonica*, vol. 52, pp. 123–129, 2000.
10. F. Glover, Tabu Search-Part I., *ORSA Journal on Computing*, vol.1, pp. 190–206, 1989.
11. F. Glover, Tabu Search-Part II., *ORSA Journal on Computing*, vol.2, pp. 4–32, 1990.
12. S. Hanafi and A. Freville, An efficient tabu search approach for the 0-1 multidimensional knapsack problem, *European Journal of Operational Research*, vol. 106, pp. 659–675, 1998.
13. M.C. Cunha and L. Ribeiro, Tabu search algorithms for water network optimization, *European Journal of Operational Research* (to appear).
14. C. Blum and M.J. Blesa, New metaheuristic approaches for the edge-weighed k-cardinality tree problem, *Computer & Operations Research* (to appear).
15. M.C. Souza, C. Duhamel and C.C. Ribeiro, A GRASP heuristic for the capacitated minimum spanning tree problem using a memory-based local search strategy, Optimization Online, March, 2002.
16. G.J. Klir and T.A. Folger, *Fuzzy Sets, Uncertainty, and Information*, Prentice Hall, 1988.
17. L.A. Zadeh, Probability measure of fuzzy events, *Journal of Mathematical Analysis and Applications*, vol. 23, pp. 421–427, 1968.
18. D. Dubois, H. Prade, *Fuzzy Sets and Systems*, Academic Press, New York, 1980.
19. L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems*, vol. 1, pp. 3–28, 1978.
20. M. Inuiguchi, and J. Ramik, Possibilistic linear programming: a brief review of fuzzy mathematical programming and a comparison with stochastic programming in portfolio selection problem, *Fuzzy Sets and Systems*, vol. 111, pp. 3–28, 2000.
21. G.B. Dantzig, Linear programming under uncertainty, *Management Science*, vol.1, 197–206, 1995.
22. M.L. Beale, On optimizing a convex function subject to linear inequalities, *Journal of Royal Statistics Society*, vol.17, 173–184, 1955.
23. A. Charnes and W.W. Cooper, Deterministic equivalents for optimizing and satisficing under chance constraints, *Operations Research*, vol.11, pp. 18–39, 1963.
24. S. Kataoka, A stochastic programming model, *Econometrica*, vol. 31, pp. 181-196, 1963.
25. A.M. Geoffrion, stochastic programming with aspiration or fractile criteria, *Management Science*, vol. 13, pp.672–679, 1967.