25–29 July, 2004 • Budapest, Hungary

Chance-Constrained Programming for Fuzzy Quadratic Minimum Spanning Tree Problem

Jinwu Gao
Department of Mathematical Sciences
Tsinghua University
Beijing 100084, China
E-mail: jgao@orsc.edu.cn

Mei Lu
Department of Mathematical Sciences
Tsinghua University
Beijing 100084, China
E-mail: mlu@math.tsinghua.edu.cn

Linzhong Liu
School of Traffic and Transportation
Lanzhou Jiaotong University
Lanzhou 730070, China
E-mail: lzliu@orsc.edu.cn

Abstract—**This paper investigates a minimum spanning tree (MST) problem with fuzzy costs and quadratic cost structure, which we call the fuzzy quadratic minimum spanning tree problem (FQMST). After formulating the FQMST problem as a chance-constrained programming model based on a credibility measure, the deterministic equivalent is proposed when the fuzzy direct costs and fuzzy interactive costs are characterized by trapezoidal fuzzy numbers. Then, a genetic algorithm is designed for solving FQMST problems. Finally, a numerical example is provided for illustrating the effectiveness of the genetic algorithm.**

I. INTRODUCTION

The minimum spanning tree (MST) problem has important applications in transportation, communications, distribution systems, etc. This problem has been investigated by many researchers such as Dijkstra [3], Kruskal [8], Prim [16] and Gabow [5]. Extensions of MST problem has also been discussed by some researchers. For example, Xu [18] introduced a MST problem with quadratic cost structure caused by the interactive costs that occur when certain pairs of edges are selected simultaneously. Ishii *et al.* [7] investigated a MST problem with random edge costs. In this paper, we consider a minimum spanning tree problem with quadratic cost structure and fuzzy costs, which we call the fuzzy quadratic minimum spanning tree problem (FQMST), and formulated the problem a fuzzy chance-constrained programming model using the concept of credibility measure. When the fuzzy costs are characterized by trapezoidal fuzzy numbers, we give its crisp equivalent. Furthermore, a genetic algorithm approach is designed for solving the chance-constrained programming model as well as its deterministic equivalent.

This paper is arranged as follows. In the following section, we recall some preliminaries in the credibility theory. Then Section III introduces a FQMST problem and formulates it as a fuzzy chance-constrained programming model. Section IV discusses the deterministic equivalent of the fuzzy chance-constrained programming model. Nextly, we design a genetic algorithm for solving the fuzzy chance-constrained programming as well as its deterministic equivalent. Lastly, a numerical example is provided for illustrating the effectiveness of the genetic algorithm.

II. PRELIMINARIES

Possibility theory was proposed by Zadeh [19], and developed by many researchers such as Dubois and Prade [4]. Recently, credibility theory was developed by Liu [15] as a branch of mathematics that studies the behavior of fuzzy events.

Let Θ be a nonempty set, $\mathcal{P}(\Theta)$ be the power set of Θ, and Pos is a possibility measure. The triplet $(\Theta, \mathcal{P}(\Theta), \text{Pos})$ is called a possibility space. Then for each fuzzy event $A \in \mathcal{P}(\Theta)$, its possibility to occur is $\text{Pos}\{A\}$, while its necessity to occur is defined by $\text{Nec}\{A\} = 1 - \text{Pos}\{A^c\}$. It is obvious that a fuzzy event may fail even though its possibility achieves 1, and hold even though its necessity is 0.

Recently, Liu and Liu [12] suggested a credibility measure Cr as follows.

Definition 1: (Liu and Liu [12]) Let $(\Theta, \mathcal{P}(\Theta), \text{Pos})$ be a possibility space, and A a set in $\mathcal{P}(\Theta)$. Then the credibility measure of A is defined by

$$\text{Cr}\{A\} = \frac{1}{2}(\text{Pos}\{A\} + \text{Nec}\{A\}). \tag{1}$$

It is easy to verify that Cr has the following properties:
(i) $\text{Cr}\{\emptyset\} = 0$, $\text{Cr}\{\Theta\} = 1$;
(ii) $\text{Cr}\{A\} \le \text{Cr}\{B\}$ whenever $A,\ B \in \mathcal{P}(\Theta)$, and $A \subset B$;
(iii) $\text{Pos}\{A\} \ge \text{Cr}\{A\} \ge \text{Nec}\{A\}$ for all $A \in \mathcal{P}(\Theta)$;
(iv) $\text{Cr}\{A\} + \text{Cr}\{A^c\} = 1$ for all $A \in \mathcal{P}(\Theta)$.
Directly from these basic properties, we can see that the credibility measure may play the role of probability measure more appropriately than the possibility measure and necessity measure. Then based on the credibility measure, Liu [14][15] developed the credibility theory including credibility distribution, independent and identical distribution, expected value operator, optimistic and pessimistic values, and so on. Furthermore, a framework of fuzzy programming [14] including expected value model, chance-constrained programming model and dependent-chance programming model is presented. In this paper, chance-constrained programming model based on the credibility measure is used to formulate the FQMST problem.

III. PROBLEM FORMULATION

Let $G = (V, E)$ be an undirected graph with vertex set $V = \{v_1, v_2, \cdots, v_n\}$ and edge set $E = \{e_1, e_2, \cdots, e_m\}$. A spanning tree $T = (V, S)$ is a subgraph of G such that $S \subseteq E$, $|S| = n - 1$ (where $|S|$ denote the cardinality of set S) and T is connected. In real world, the transportation, communication, and distribution networks may be characterized by spanning trees, and the construction or running cost on edge e_i may be represented by a weight w_i, $i = 1, 2, \cdots, m$. Let x be a binary decision variable defined as:

$$x_i = \begin{cases} 1, & \text{if edge } e_i \text{ is selected} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Then the cost of a spanning tree $\boldsymbol{x}$ is denoted by $\sum_{i=1}^{m} w_i x_i$. Let Γ be the set of all spanning trees corresponding to the graph G. Then the ordinary minimal spanning tree problem can be formulated as follows:

$$\min \left\{ z(\boldsymbol{x}) = \sum_{i=1}^{m} w_i x_i | \boldsymbol{x} \in \Gamma \right\}. \tag{3}$$

Sometimes, due to some interactive effects, an interactive cost will occur when a pair of edges are selected simultaneously within a tree. This may explain two interactional activities in real decision systems such as transportation, communication, and distribution networks. Denote the interactive cost due to a pair of edges (e_i, e_k) by c_{ik} and let $c_{ii} = 0$, then the cost function of a spanning tree is no longer linear but quadratic, and the minimum spanning tree problem can be formulated as follows:

$$\min \left\{ z(\boldsymbol{x}) = \sum_{i=1}^{m} \sum_{k=1}^{m} c_{ik} x_i x_k + \sum_{i=1}^{m} w_i x_i | \boldsymbol{x} \in \Gamma \right\}. \tag{4}$$

In real-world decision systems, the decision-maker often faces with some uncertain situations like insufficient information about the construction or running costs. For these cases, the direct costs and interactive costs may be specified as fuzzy variables according to the expert system. Then the cost function $\sum_{i=1}^{m} \sum_{k=1}^{m} c_{ik} x_i x_k + \sum_{i=1}^{m} w_i x_i$ becomes fuzzy too, which makes model (4) meaningless. A natural idea is to provide a confidence level α at which it is desired that $\sum_{i=1}^{m} \sum_{k=1}^{m} c_{ik} x_i x_k + \sum_{i=1}^{m} w_i x_i \leq f$. And the objective of the leader is to minimize f with a chance constraint as follows,

$$\mathrm{Cr} \left\{ \sum_{i=1}^{m} \sum_{k=1}^{m} c_{ik} x_i x_k + \sum_{i=1}^{m} w_i x_i \leq f \right\} \geq \alpha, \tag{5}$$

where f is referred to as the α-pessimistic value of the cost function. Then the FQMST problem can be formulated as the following chance-constrained programming model:

$$\begin{cases} \min f \\ \text{subject to:} \\ \qquad \mathrm{Cr} \left\{ \sum_{i=1}^{m} \sum_{k=1}^{m} c_{ik} x_i x_k + \sum_{i=1}^{m} w_i x_i \leq f \right\} \geq \alpha \\ \qquad x_i = 0 \text{ or } 1 \\ \qquad \boldsymbol{x} \in \Gamma \end{cases} \tag{6}$$

IV. Crisp Equivalent

Suppose that the direct costs and the interactive costs are all trapezoidal fuzzy numbers. In this section, we give the deterministic equivalent of fuzzy chance-constrained programming model 6.

Firstly, let us investigate some properties of the trapezoidal fuzzy numbers.

Lemma 1: Let $\xi = (r_1, r_2, r_3, r_4)$ and $\eta = (q_1, q_2, q_3, q_4)$ be two trapezoidal fuzzy variables, and a, b two nonnegative number. Then we have that

$$a\xi + b\eta = (ar_1 + bq_1, ar_2 + bq_2, ar_3 + bq_3, ar_4 + bq_4). \tag{7}$$

Proof. This result can be proved directly by Zadeh's extension principle and the proof is omitted here.

Let $\xi = (r_1, r_2, r_3, r_4)$ be a trapezoidal fuzzy variable. From the definition of credibility, it is easy to obtain

$$\mathrm{Cr}\{\xi \leq f\} = \begin{cases} 1, & \text{if } r_4 \leq f \\ \dfrac{f + r_4 - 2r_3}{2(r_4 - r_3)}, & \text{if } r_3 \leq f \leq r_4 \\ \frac{1}{2}, & \text{if } r_2 \leq f \leq r_3 \\ \dfrac{f - r_1}{2(r_2 - r_1)}, & \text{if } r_1 \leq f \leq r_2 \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Then, based on this credibility distribution function, we can prove the following Lemma easily.

Lemma 2: Let $\xi = (r_1, r_2, r_3, r_4)$ be a trapezoidal fuzzy variable, and α a given confidence level. Then we have(a) when $\alpha \leq 1/2$, $\mathrm{Cr}\{\xi \leq f\} \geq \alpha$ if and only if $(1 - 2\alpha)r_1 + 2\alpha r_2 \leq f$;(b) when $\alpha > 1/2$, $\mathrm{Cr}\{\xi \leq f\} \geq \alpha$ if and only if $(2 - 2\alpha)r_3 + (2\alpha - 1)r_4 \leq f$.

Proof. Part (a): If $\alpha \leq 1/2$ and $\mathrm{Cr}\{\xi \leq f\} \geq \alpha$, then we have $r_2 \leq f \leq r_3$ or $(r_2 - \overline{f})/2(r_2 - r_1) \geq \alpha$. When $(r_2 - f)/2(r_2 - r_1) \geq \alpha$, we have $(1 - 2\alpha)r_1 + 2\alpha r_2 \leq f$. When $r_2 \leq \overline{f} \leq r_3$, since $\alpha \leq 1/2$, $(1 - 2\alpha)r_1 + 2\alpha r_2 \leq f$ holds too. Conversely, if $r_2 \leq f$, then $\mathrm{Cr}\{\xi \leq f\} \geq 1/2 \geq \alpha$. If $(1 - 2\alpha)r_1 + 2\alpha r_2 \leq f$, then $f - r_1/2(r_2 - r_1) \geq \alpha$. Thus $\mathrm{Cr}\{\xi \leq \overline{f}\} \geq \alpha$.

Part (b): If $\alpha > 1/2$ and $\mathrm{Cr}\{\xi \leq f\} \geq \alpha$, then we have $r_4 \leq f$ or $(f + r_4 - 2r_3)/2(r_4 - r_3) \geq \alpha$. It is easy to verify that $(2 - 2\alpha)r_3 + (2\alpha - 1)r_4 \leq \overline{f}$. Conversely, if $r_4 \leq f$, then $\mathrm{Cr}\{\xi \leq \overline{f}\} = 1 \geq \alpha$. If $(2 - 2\alpha)r_3 + (2\alpha - 1)r_4 \leq \overline{f}$, then we also have $(2r_3 - r_4)/2(r_3 - r_4) \geq \alpha$. Thus $\mathrm{Cr}\{\xi \leq f\} \geq \alpha$.

Theorem 1: Let w_i in fuzzy chance-constrained programming model (6) be trapezoidal fuzzy numbers $(s_{i1}, s_{i2}, s_{i3}, s_{i4})$, $i = 1, 2, \cdots, m$, c_{ij} trapezoidal fuzzy numbers $(t_{ij1}, t_{ij2}, t_{ij3}, t_{ij4})$, $i, j = 1, 2, \cdots, m$. If $\alpha > 0.5$, then the crisp equivalent of the fuzzy CCP model (6) is given by

$$\begin{cases} \min & \begin{aligned}&(2-2\alpha)(\sum_{i=1}^{m}\sum_{j=1}^{m} t_{ij3}x_ix_j + \sum_{i=1}^{m} s_{i3}x_i)\\ &+(2\alpha-1)(\sum_{i=1}^{m}\sum_{j=1}^{m} t_{ij4}x_ix_j + \sum_{i=1}^{m} s_{i4}x_i)\end{aligned} \\ \text{subject to:} & \\ & x_i = 0 \text{ or } 1 \\ & \boldsymbol{x} \in \Gamma. \end{cases} \tag{9}$$

Proof. Since $x_i \geq 0$ for $i = 1, 2, \cdots, m$, it follows from Lemma 1 that the cost function

$$\sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij}x_ix_j + \sum_{i=1}^{m} w_ix_i \tag{10}$$

is also a trapezoidal fuzzy number, and determined by the quadruple

$$\begin{pmatrix} \sum_{i=1}^{m}\sum_{j=1}^{m} t_{ij1}x_ix_j + \sum_{i=1}^{m} s_{i1}x_i) \\ \sum_{i=1}^{m}\sum_{j=1}^{m} t_{ij2}x_ix_j + \sum_{i=1}^{m} s_{i2}x_i) \\ \sum_{i=1}^{m}\sum_{j=1}^{m} t_{ij3}x_ix_j + \sum_{i=1}^{m} s_{i3}x_i) \\ \sum_{i=1}^{m}\sum_{j=1}^{m} t_{ij4}x_ix_j + \sum_{i=1}^{m} s_{i4}x_i) \end{pmatrix}^T. \tag{11}$$

Then it follows from Lemma 2 that the chance constraint

$$\mathrm{Cr}\left\{\sum_{i=1}^{m}\sum_{k=1}^{m} c_{ik}x_ix_k + \sum_{i=1}^{m} w_ix_i \leq f\right\} \geq \alpha \tag{12}$$

is equivalent to

$$\begin{aligned}&(2-2\alpha)(\sum_{i=1}^{m}\sum_{j=1}^{m} t_{ij3}x_ix_j + \sum_{i=1}^{m} s_{i3}x_i)\\ &\quad+(2\alpha-1)(\sum_{i=1}^{m}\sum_{j=1}^{m} t_{ij4}x_ix_j + \sum_{i=1}^{m} s_{i4}x_i).\end{aligned} \tag{13}$$

That is, the fuzzy chance-constrained programming model (6) is equivalent to model (9).

V. Genetic Algorithm Approach

Xu [18] proved that the MST problem with quadratic cost structure is NP-hard and gave two heuristic algorithms which can only get some local optima. Zhou and Gen [20] showed that genetic algorithm can find much better solutions than Xu's heuristic algorithms. As an fuzzy extension of MST problem with quadratic cost structure, the FQMST problem is more difficult to solve. In the following, we will design a simulation-based genetic algorithm for solving general FQMST problems.

A. Chromosome Representation

When talk of genetic algorithm, we must start with the representation of chromosome. In the FQMST problem, a chromosome should represent a spanning tree.

Firstly, we refer to Cayley's theorem in graphical enumeration, which shows that there are n^{n-2} distinct labelled trees on a complete graph with n vertices. Prüfer [17] proved this theorem by constructing one-to-one correspondence between such trees and the set of permutation of $n-2$ digit. And this permutation is known as the Prüfer number. Then based on the one-to-one correspondence, we use the Prüfer number as chromosome, which uniquely represents a spanning tree. For detailed discussion of encoding method of tree, the reader may consult the book [6] by Gen and Cheng.

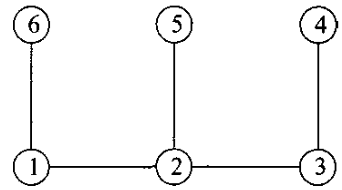

Fig. 1. The spanning tree associated with Prüfer number (3,2,2,1)

For any tree, there are at least two leaf vertices. By leaf vertex we mean that there is only one edge connected to the vertex. The following encoding and decoding procedure were proposed by Zhou and Gen [20].

Encoding Procedure

Step 1.Find the smallest labeled leaf vertex, say i, in a labeled tree.

Step 2.Let j be the first digit in the encoding if j is incident to vertex i. Here we build the encoding by appending digits to the right, and thus the encoding is built from left to right.

Step 3.Remove vertex i and the edge from i to j, we get a tree with $n-1$ vertices.

Step 4.Repeat the above steps until one edge is left and produce the Prüfer number with $n-2$ digits between 1 and n inclusive.

Decoding Procedure

Step 1.Let P be the original Prüfer number, $\overline{P}$ be the set of all vertices not included in P, and be designated as eligible for consideration.

Step 2.Find the smallest labeled leaf vertex, say i, in $\overline{P}$, and the leftmost digit of P, say j. Add the edge from i to j in to the tree. Remove i from $\overline{P}$ and j from P. If j does not occur anywhere in the remain part of P, designate j as eligible and put it in to $\overline{P}$.

Step 3.Repeat the above two steps until there are only two vertices, r and s, eligible for consideration. Add the edge from r to s into the tree and form a tree with $n-1$ edges.

Now we convert a spanning tree (Figure 1) to a Prüfer number for illustrating the procedure of encoding process. In Figure 1, the leaf vertex with smallest label is 4, and vertex 3 is incident to it. So we remove the vertex 4 and the edge from vertex 4 to 3 from the tree, and let 3 be the leftmost digit in the Prüfer number. Nextly, find the smallest leaf vertex 3, and the vertex incident to it in the subtree, vertex 2. Let 2 be the second digit in the Prüfer number, and remove the vertex 3 and edge from vertex 3 to 2 from the tree. Repeat the above process on the subtree, we finally get a Prüfer number with four digits $(3,2,2,1)$ and leave only one edge from vertex 1 to 6.

The decoding procedure of the Prüfer number $(3,2,2,1)$ will be demonstrated in Section VI.

B. Simulation for Pessimistic Value

If the fuzzy chance-constrained programming can be converted to its deterministic equivalent, then it is easy for us to compute the objective function. Otherwise, we may resort to the fuzzy simulation technique [14][15]. Now we give a fuzzy simulation procedure for computing the pessimistic value

$$\inf\{f \mid \mathrm{Cr}\{\sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij}x_ix_j + \sum_{i=1}^{m} w_ix_i \le f\} \ge \alpha\}. \tag{14}$$

Fuzzy Simulation for α-Pessimistic Value

Step 1.Set $k = 1$.

Step 2.Randomly generate u_{ijk} from the ε-level set of c_{ij}, and v_{ik} from the ε-level set of w_i, $i,j = 1,2,\cdots,m$, respectively, where ε is a sufficiently small positive number.

Step 3.Set $\nu_k = \min_{i,j}\{\mu_{c_{ij}}(u_{ijk})\} \wedge \min_i\{\mu_{w_i}(v_{ik})\}$.

Step 4.Repeat Steps 2–3 until $k > N$.

Step 5.Find the maximal value b such that $L(b) \ge b$ holds, where

$$\begin{aligned} L(b) = \frac{1}{2}\Big(&\max_{1\le k\le N}\{\nu_k \mid f_k(\boldsymbol{x}) \le b\} \\ &+ \min_{1\le k\le N}\{1-\nu_k \mid f_k(\boldsymbol{x}) > b\}\Big), \end{aligned} \tag{15}$$

and

$$f_k(\boldsymbol{x}) = \sum_{i=1}^{m}\sum_{j=1}^{m} u_{ijk}x_ix_j + \sum_{i=1}^{m} v_{ik}x_i. \tag{16}$$

Step 6.Return b.

C. Crossover and Mutation Operation

Crossover and mutation operation are two crucial factors in the biological evolutionary process. In the genetic algorithm approach, they may guarantee the diversity of the population. Hence the population have a great chance to be evolved to the optimal solution. Since a Prüfer number can always represent a labelled tree, we select a simple way for crossover and mutation operation. For two chromosomes to crossover, we just exchange their digits at randomly selected positions. And for a chromosome to mutate, randomly select a position and randomly generate an integer between 1 and n including 1 and n to replace the original.

D. Evaluation and Selection Process

Evaluation and selection process play an important role in genetic algorithm. They may be regarded as the exploration for genetic algorithm to converge to the optimal or near-optimal solution. In our genetic algorithm approach for stochastic QMST problem, the evaluation perform the following functions: (i) decoding all the chromosomes and calculating their α-pessimistic value; (ii) assigning each chromosome a fitness by a rank-based method according to its objective value. Then in the selection process, by spinning the roulette wheel *pop_size* times, we get a new population to go further.

E. Genetic Algorithm Procedure

Following selection, crossover, and mutation, the new population is ready for its next evaluation. The genetic algorithm will terminate after a given number of cyclic repetitions of the above steps or a suitable solution has been found. A general genetic algorithm procedure is given as follows.

Genetic algorithm procedure

Step 1.Initialize *pop_size* chromosomes randomly.

Step 2.Update the chromosomes by crossover and mutation operations.

Step 3.Calculate the objective values for all chromosomes by fuzzy simulation.

Step 4.Compute the fitness of each chromosome according to the objective values.

Step 5.Select the chromosomes by spinning the roulette wheel.

Step 6.Repeat the second to fifth steps for a given number of cycles.

Step 7.Report the best chromosome as the optimal solution.

We also note that the genetic algorithm procedure can also apply to model (4) and (9).

VI. Numerical Example

Zhou and Gen [20] have illustrated by some numerical example that genetic algorithm with this representation structure is effective in finding much better solutions than existing heuristics. So in this section, we only give a numerical example for illustrating purpose.

Suppose that the decision maker has set the confidence level $\alpha = 0.90$. Now we go into particulars for a FQMST problem with 6 vertices. For a complete graph, we label its 6 vertices with integers $1,2,\cdots,6$, respectively. Then it has $C_6^2 = 15$ edges. The direct costs w_i are characterized by trapezoidal fuzzy numbers $(s_{i1}, s_{i2}, s_{i3}, s_{i4})$, where s_{ik} are randomly generated from $[2,10]$, $i = 1,2,\cdots,6$, $k = 1,2,3,4$. The interactive costs c_{ij} are also characterized by trapezoidal fuzzy numbers $(t_{ij1}, t_{ij2}, t_{ij3}, t_{ij4})$, where t_{ijk} are randomly generated from $[0,4]$, $i,j = 1,2,\cdots,6$, $k = 1,2,3,4$. After a run of the genetic algorithm with *pop_size* $= 30$, we get the optimal chromosome $(3,2,2,1)$, and the optimum is 43.2829. Now, we convert the chromosome to a solution by the decoding procedure.

1) Let $P = (3, 2, 2, 1)$ and $\overline{P} = \{4, 5, 6\}$.
2) Remove the smallest integer 4 from $\overline{P}$, and the leftmost integer 3 from P. Add the edge from 3 to 4 to the tree. Since 3 does not occur at other places of P, we have $P = (2, 2, 1)$, $\overline{P} = \{3, 5, 6\}$.
3) Remove the smallest integer 3 from $\overline{P}$, and the leftmost integer 2 from P. Add the edge from 2 to 3 to the tree. Since 2 occurs at other places of P, we have $P = (2, 1)$ and $\overline{P} = \{5, 6\}$.
4) Remove the smallest integer 5 from $\overline{P}$, and the leftmost integer 2 from P. Add the edge from 2 to 5 to the tree. Since 2 does not occur at other places of P, we have $P = (1)$, $\overline{P} = \{2, 6\}$.
5) Remove the smallest integer 2 from $\overline{P}$, and the leftmost integer 1 from P. Add the edge from 1 to 2 to the tree. Now we have $\overline{P} = \{1, 6\}$, $P = \emptyset$.
6) Add the edge from vertex 1 to 6 to the tree, and we get a tree with five edges, i.e., a spanning tree in Figure 1.

VII. Conclusions

In this paper, a FQMST problem was formulated it as fuzzy chance-constrained programming. The crisp equivalent of the fuzzy chance-constrained programming was given when the costs are characterized by trapezoidal fuzzy numbers. Moreover, a simulation-based genetic algorithm was designed for solving the fuzzy chance-constrained programming as well as its crisp equivalent. A numerical example was also provided for illustrating purpose.

Acknowledgments

This work was supported by Qinglan talent fund of Lanzhou Jiaotong University, and Specialized Research Fund for the Doctoral Program of Higher Education (No.20020003009).

References

[1] J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications*, Macmillan, London and Elsevier, New York, 1976.
[2] N. Christofides, *Graph Theory: An Algorithmic Approach*, Academic Press, New York, 1975.
[3] E.W. Dijkstra, A note on two problems in connection with graphs, *Numerische Mathematik*, 1 (1959), 269–271.
[4] D. Dubois, and H. Prade, *Possibility Theory*, Plenum, New York, 1988.
[5] H.N. Gabow, Z. Galil, T. Spencer and R.E. Tarjan, Efficient algorithms for finding minimum spanning trees in undirected and directed graphs, *Combinatorica*, 6(1986), 109–122.
[6] M. Gen and R. Cheng, *Genetic Algorithms & Engineering Optimization*, John Wiley & sons, New York, 2000.
[7] H. Ishii, and T. Nishida, Stochastic bottleneck spanning tree problem, *Networks*, 13(1983), 443–449.
[8] J.B.Jr. Kruskal, On the shortest spanning subtree of a graph and traveling salesman problem, *Proceeding of AMS*, 7(1956), 45–50.
[9] B. Liu, and K. Iwamura, Chance constrained programming with fuzzy parameters, *Fuzzy Sets and Systems*, 94(1998), 227–237.
[10] B. Liu, Minimax chance-constrained programming models for fuzzy decision systems, *Information Sciences*, 112(1998), 25–38.
[11] B. Liu, Fuzzy random chance-constrained programming, *IEEE Transactions on Fuzzy Systems*, 9:5(2001), 713–720.
[12] B. Liu, and Y.-K. Liu, Expected value of fuzzy variable and fuzzy expected value models, *IEEE Transactions on Fuzzy Systems*, 10(2002), 445–450.
[13] B. Liu, Toward fuzzy optimization without mathematical ambiguity, *Fuzzy Optimization and Decision Making*, 1(2002), 43–63.
[14] B. Liu, *Theory and Practice of Uncertain Programming*, Physica-Verlag, Heidelberg, 2002.
[15] B. Liu, *Uncertainty Theory: An Introduction to its Axiomatic Foundations*, Springer-Verlag, Heidelberg, 2004.
[16] R.C. Prim, Shortest connection networks and some generalizations, *Bell System Technical Journal*, 36 (1957), 1389–1401.
[17] H. Prüfer, Neuer Beweis Eines Satzes über Permutation, *Archives of Mathematical Physics*, 27 (1918), 742–744.
[18] W. Xu, *Quadratic minimum spanning tree problems and related topics*, Ph.D. dissertation, University of Maryland, 1984.
[19] L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems*, 1(1978), 3–28.
[20] G. Zhou and M. Gen, An effective genetic algorithm approach to the quadratic minimum spanning tree problem, *Computers and Operations Research*, 25 (1998), 229–247.