Available online at www.sciencedirect.com
SCIENCE DIRECT®

Fuzzy Sets and Systems 150 (2005) 147–162

FUZZY sets and systems
www.elsevier.com/locate/fss

Minimizing a linear function under a fuzzy max–min relational equation constraint

Yan-Kuen Wu[a,1], Sy-Ming Guu[b,*,2]

[a]*Department of Industrial Management, Vanung University, Taoyuan, Taiwan, 320, ROC*
[b]*Department of Business Administration, Yuan Ze University, Taoyuan, Taiwan, 320, ROC*

Received 18 October 2002; received in revised form 17 September 2004; accepted 20 September 2004
Available online 18 October 2004

Abstract

In this paper we investigate the problem of minimizing a linear objective function subject to a fuzzy relational equation constraint. A necessary condition for optimal solution is proposed. Based on this necessary condition, we propose three rules to simplify the work of computing an optimal solution. Numerical examples are provided to illustrate the procedure. Experimental results are reported showing that our new procedure systematically outperforms our previous work.

Keywords: Fuzzy optimization; Fuzzy relational equation; Max–min composition

1. Introduction

In this paper, we consider the following mathematical model:

$$\text{Minimize } Z(x) = \sum_{i=1}^{m} c_i x_i, \tag{1}$$

* Corresponding author. Department of Business Administration, Yuan Ze University, 135 Far East Road, Nei-Li, Taoyuan, 32026, Taiwan. Tel.:+886 3 463 0381; fax: +886 3 455 7040.
E-mail addresses: ykw@vnu.edu.tw (Y.-K. Wu), iesmguu@saturn.yzu.edu.tw (S.-M. Guu).
[1] Research is supported under the Grant of NSC 92-2416-H-238-003.
[2] Research is supported under the Grant of NSC 92-2416-H-155-021, Taiwan, The Republic of China.

0165-0114/$ - see front matter
doi:10.1016/j.fss.2004.09.010

$$\text{subject to } x \in X(A,b) := \{x \in [0,1]^m \,|\, x \circ A = b\}, \tag{2}$$

where $c_i > 0$, $A = (a_{ij})_{m \times n}$ with $0 \leqslant a_{ij} \leqslant 1$, $b = (b_1, \ldots, b_n)$ is an n-dimensional vector with $0 \leqslant b_j \leqslant 1$, and the operation "$\circ$" represents the max–min composition.

As an application, model (1)–(2) has been employed for the streaming media provider seeking a minimum cost while fulfilled the requirements assumed by a three-tier framework. This three-tier framework consists of an original multimedia server, m regional servers, and n clients. The multimedia server is the original streaming data provider which services n clients through m regional servers. The jth client has access to the ith regional server through a network connection with bandwidth a_{ij}. Each client is guaranteed to have at least one way of receiving the multimedia streaming data that meets its quality level. Client j has quality requirement b_j. The multimedia server sends streaming data with quality level x_i to the ith regional server through the virtual circuit. The min operations involved in constraints reflect the situation that "if the streaming data transmitted from ith regional server to jth client has a quality level x_i lower than the bandwidth a_{ij}, the streaming data will be delivered without losing any information. In case the streaming data has a quality level better than the bandwidth a_{ij}, some data will be dropped which means no streaming data that has a quality level higher than the bandwidth can be delivered completely." The max operations reflect that each client needs at least one regional server to fulfill its quality requirement. In this application, all the quality levels x_i, b_j, and bandwidth a_{ij} have been normalized to be within $[0, 1]$. The objective function is the service cost per unit time, measured in dollars per second. For detailed description, we refer to [12].

Let $\mathcal{I} = \{1, 2, \ldots, m\}$ and $\mathcal{J} = \{1, 2, \ldots, n\}$ be two index sets, the constraint part of model (1)–(2) is to find a set of solution vectors $x \in [0, 1]^m$ such that

$$\max_{i \in \mathcal{I}} \min(a_{ij}, x_i) = b_j, \forall j \in \mathcal{J}. \tag{3}$$

Finding solutions of equation (3) belongs to the topic of fuzzy relational equation problem [1,17,20,22]. Let $x^1 = (x_i^1)$, $x^2 = (x_i^2)$ be two vectors in $[0, 1]^m$. If we assume $x^1, x^2 \in X(A, b)$ and define $x^1 \leqslant x^2$ if and only if $x_i^1 \leqslant x_i^2$ for all $i \in \mathcal{I}$, then the operator "$\leqslant$" forms a partial order relation on $X(A, b)$. A solution $\bar{x} \in X(A, b)$ is called the maximum solution if $x \leqslant \bar{x}$ for all $x \in X(A, b)$. On the other hand, an $\underline{x} \in X(A, b)$ is a minimal solution if $\forall x \in X(A, b)$, $x \leqslant \underline{x}$ implies that $x = \underline{x}$. A solution $x^* \in X(A, b)$ is *optimal* for problem (1)–(2) if $Z(x^*) \leqslant Z(x)$ for all $x \in X(A, b)$.

It is well-known [11] when the solution set $X(A, b)$ is nonempty, the $X(A, b)$ can be completely determined by the unique maximum solution and a finite number of minimal solutions [5,9,22]. Moreover, the maximum solution can be computed easily by the following Goedel implication [22]:

$$\bar{x} = A \diamond b = (\min_{j \in \mathcal{J}}(a_{ij} \diamond b_j))_{i \in I}, \tag{4}$$

where

$$a_{ij} \diamond b_j := \begin{cases} 1 & \text{if } a_{ij} \leqslant b_j; \\ b_j & \text{if } a_{ij} > b_j. \end{cases}$$

Note that $X(A, b) \neq \emptyset$ if and only if the vector $A \diamond b$ satisfies all equations in (3). We assume in this paper that $X(A, b)$ is nonempty. Although the maximum solution of (3) can be easily computed, the procedure of finding all minimal solutions may be tedious. We refer to [4,5,9,16,18,19,21] for algorithms to find all minimal solutions of (3). Recently, there is a growing interest for more general research on fuzzy relational equations with max-t-norm composition [2,3,24,26]. Applications of fuzzy relational equations can be found in [5,6,10,21,23].

Our main aim for model (1)–(2) is to find its optimal solution. We note that the optimal solution of model (1)–(2) is among the minimal solutions of $X(A, b)$. Therefore, one possible way to find an optimal solution is to compute *all* minimal solutions first (with the aid of algorithms in literature) and then by enumeration to find the optimal solution.

The other approach to find an optimal solution has been documented in Fang and Li [7]. They showed that problem (1)–(2) can be converted into a 0-1 integer programming problem. Furthermore, Fang and Li solved this associated 0-1 integer programming problem by branch and bound method with jump-tracking technique. Wu et al. [28] improved Fang and Li's method by providing an initial upper bound for the branch and bound part. Testing examples showed that their initial upper bound is sharp. In addition, with this upper bound and rearranging the structure of problem (1)–(2), the branch and bound part in Wu, Guu and Liu's procedure visited much less nodes of the solution tree than that in Fang and Li. The initial upper bound employed in [28] is easy to compute, yet this upper bound is "fixed" in their procedure. That is, in Wu, Guu and Liu's procedure the initial upper bound is not updated by a better bound when possible. In the present paper we shall update the current bound when a better bound is generated.

Variants of model (1)–(2) can be found in literature. If the objective function in (1) becomes $Z(x) = \max_{i\in\mathcal{I}}\{\min(c_i, x_i)\}$ with $c_i \in [0, 1]$, the model is called the latticized linear programming problem [27]. On the other hand, Wang [25] explored the same mathematical problem (1)–(2) with multiple linear objective functions. Wang characterized some properties of efficient points and transformed the problem as a multi-attribute decision problem. Recently, Loetamonphong et al. [14] have studied nonlinear multi-objective optimization problem with a fuzzy relational equation constraint. And a genetic algorithm was employed to find the Pareto optimal solutions. Lu and Fang [15] proposed a genetic algorithm to solve the problem (1)–(2) with single nonlinear objective function. We refer to [8,13] for model (1)–(2) with max-product composition in place of max-min composition.

The rest of this paper is organized as follows. Section 2 contains theorems in which necessary conditions for an optimal solution are stated. We then derive three rules to simplify the work of finding an optimal solution. Procedure for finding an optimal solution will be presented. Section 3 contains two examples to illustrate the procedure. Conclusion is in Section 4.

2. Rules for reducing the problem

In this section, we shall present new results for optimal solution of problem (1)–(2). As in [28], we hereinafter rearrange the coefficients in c and b in increasing order, namely we require $0 < c_1 \leqslant c_2 \leqslant \cdots \leqslant c_m$ and $b_1 \leqslant b_2 \leqslant \cdots \leqslant b_n$.

Lemma 1 (*Peeva [21]*). *Let $x \in X(A, b)$. Then for each $j \in \mathcal{J}$, there exists at least one index i such that* $\min(x_i, a_{ij}) = b_j$.

Definition 1. For any solution $x \in X(A, b)$, the x_i is called a *binding* variable if $\min(x_i, a_{ij}) = b_j$ for some $j \in \mathcal{J}$.

Let $x \in X(A, b)$ and x_i be a binding variable. The index set $\{j \mid \min(x_i, a_{ij}) = b_j, \forall j \in \mathcal{J}\}$ is denoted by $J_i(x)$. The $J_i^*(x)$ denotes the largest index in a nonempty $J_i(x)$. Note that x_i is nonbinding if and only if $J_i(x) = \emptyset$.

Lemma 2. *For any optimal solution $x^* \in X(A, b)$, if x_i^* is a binding variable, then$x_i^* = b_{J_i^*(x^*)}$.*

Proof. Since x_i^* is a binding variable, we have $\min(x_i^*, a_{ij}) = b_j$ for all $j \in J_i(x^*)$. Hence, we have $x_i^* \geqslant b_j$ for all $j \in J_i(x^*)$. Since the b_js are increasing in j, we have in particular, $x_i^* \geqslant b_{J_i^*(x^*)}$. Since $c_i > 0$ and x^* is an optimal solution, the x_i^* must be as small as possible. Hence, we have $x_i^* = b_{J_i^*(x^*)}$. □

Theorem 1. *Let x^* be an optimal solution of problem* (1)–(2). *Then $x_i^* = 0$ or $x_i^* = b_{J_i^*(x^*)}$.*

Proof. If x_i^* is not a binding variable, we can assign 0 to x_i^* due to $c_i > 0$. On the other hand, if x_i^* is a binding variable, then $x_i^* = b_{J_i^*(x^*)}$ by Lemma 2.

The analysis so far implies that for any optimal solution x^*, if x_i^* is nonbinding, then x_i^* can be assigned to be zero. It turns out that the maximum solution $\bar{x}$ provides information in searching for nonbinding decision variables, to which we now turn.

Theorem 2. *Let $\bar{x}$ be the maximum solution of $X(A, b)$. If $\bar{x}_i$ is not a binding variable, then x_i is nonbinding for any $x \in X(A, b)$. On the other hand, if $\bar{x}_i$ is a binding variable, then*

$$J_i(x) \subset J_i(\bar{x}) \, \forall x \in X(A, b).$$

Proof. If $\bar{x}_i$ is nonbinding, then we have $\min(\bar{x}_i, a_{ij}) < b_j$ for all $j \in \mathcal{J}$. Since $\bar{x}$ is the maximum solution, we have $x_i \leqslant \bar{x}_i$ for all $x \in X(A, b)$. This implies that

$$\min(x_i, a_{ij}) \leqslant \min(\bar{x}_i, a_{ij}) < b_j \text{ for all } j \in \mathcal{J}.$$

Hence, x_i is nonbinding as well for all $x \in X(A, b)$. □

On the other hand, suppose that $\bar{x}_i$ is a binding variable. Then $J_i(\bar{x})$ is nonempty. For any solution $x \in X(A, b)$, if $J_i(x)$ is empty, then the theorem holds obviously. Suppose that $J_i(x)$ is nonempty and consider $j \in J_i(x)$, we need to show that $j \in J_i(\bar{x})$. Since $j \in J_i(x)$, we have $\min(x_i, a_{ij}) = b_j$. We have two cases to be considered. Case 1: if $a_{ij} = b_j$, then $x_i \in [b_j, 1]$. Since $\bar{x}$ is the maximum solution, we have $\bar{x}_i \geqslant x_i \geqslant b_j$. Hence, we have $\min(\bar{x}_i, a_{ij}) = b_j$. Case 2: if $a_{ij} > b_j$, then $x_i = b_j$. Since $\bar{x}$ is the maximum solution, we have $\bar{x}_i \geqslant x_i = b_j$. On the other hand, since $a_{ij} > b_j$ and $\bar{x}_i = \min_{j \in J}\{a_{ij} \diamond b_j\}$, we have $\bar{x}_i \leqslant a_{ij} \diamond b_j = b_j$. Therefore, we have $\bar{x}_i = b_j$. It follows that $\min(\bar{x}_i, a_{ij}) = b_j$. Both cases imply that $j \in J_i(\bar{x})$.

It follows from Theorem 2 that if $\bar{x}_i$ is nonbinding, then we can simply set $x_i^* = 0$, where $x^* = (x_i^*)$ is any optimal solution. If $\bar{x}_i$ is binding, then $J_i(x^*) \subseteq J_i(\bar{x})$ for any optimal solution x^*. If x_i^* is binding, then $x_i^* = b_{J_i^*(x^*)}$ by Theorem 1. However, we have to first identify the index set $J_i(x^*)$ in order to compute the x_i^*. Since $J_i(\bar{x})$ contains $J_i(x^*)$, we can limit our search within $J_i(\bar{x})$, to which we now turn.

Define a value matrix $M = (m_{ij})$ with $i \in \mathcal{I}$ and $j \in \mathcal{J}$ by

$$m_{ij} = \begin{cases} c_i b_j & \text{if } j \in J_i(\bar{x}); \\ \infty & \text{otherwise.} \end{cases}$$

The numerical elements in the ith row of M correspond to the contributions to the objective by x_i^*, where each of the b_j with $j \in J_i(\bar{x})$ is a possible candidate for x_i^*.

One of the new results in this paper is that with the matrix M we can create rules to reduce the problem. Since for any optimal solution x^*, x_i^* is either 0 or $b_{J_i^*(x^*)}$ (by Theorem 1), the rules for reducing the problem are to set as many optimal decision variables as possible to 0 or $b_{J_i^*(x^*)}$.

Lemma 3. *Let x_i be a binding variable of a solution x. Then $a_{ij} = b_j$ for all $j \in J_i(x) \setminus B_i(x)$, where $B_i(x) = \{j \in J_i(x) | b_j = b_{J_i^*(x)}\}$.*

Proof. Since x_i is binding, the index set $J_i(x)$ is nonempty. And $\min(x_i, a_{ij}) = b_j$ for all $j \in J_i(x)$. Since b_js are increasing with respect to j and $x_i \geqslant b_j$ for all $j \in B_i(x)$, we have

$$a_{ij} = b_j \text{ for all } j \in J_i(x) \setminus B_i(x). \qquad \square$$

Rules for reducing the problem: We are ready to present the rules to determine the values of as many decision variables as possible in the optimal solution.

Rule 1. If $\emptyset \neq J_s(\bar{x}) \subseteq J_t(\bar{x})$ for some s and t with $t < s$, then there exists an optimal solution x^* with $x_s^* = 0$.

Proof. Let x^* be any optimal solution. If $x_s^* = 0$, then we are done. Suppose to the contrary that $x_s^* \neq 0$, we shall establish a solution with a better objective value than x^*, hence a contradiction. We first note that since $\emptyset \neq J_s(\bar{x}) \subseteq J_t(\bar{x})$, by Lemma 3, we have

$$a_{sj} = b_j \text{ for all } j \in J_s(\bar{x}) \setminus B_s(\bar{x}) \text{ and } a_{tj} = b_j \text{ for all } j \in J_t(\bar{x}) \setminus B_t(\bar{x}).$$

Furthermore, $J_s(\bar{x}) \setminus B_s(\bar{x}) \subseteq J_t(\bar{x}) \setminus B_t(\bar{x})$.

Since x_s^* is nonzero, we have x_s^* a binding variable with $J_s(x^*) \neq \emptyset$. Moreover, by Theorem 1 and Theorem 2, we have

$$x_s^* = b_{J_s^*(x^*)} \text{ and } \min(x_s^*, a_{sj}) = b_j \ \forall\, j \in J_s(x^*). \qquad \square$$

Case 1: If $x_t^* = 0$, we consider the vector $x = (x_i)$ which equals to x^* except $x_t = x_s^*$ and $x_s = 0$. We then have $\min(x_t, a_{tj}) = b_j$ for all $j \in J_s(x^*)$. Hence the feasibility of x_s^* is maintained by x_t. Therefore,

x is a solution of the problem. Moreover

$$\sum_{i=1}^{n} c_i x_i^* - \sum_{i=1}^{n} c_i x_i = c_s x_s^* - c_t x_s^* \geqslant 0.$$

If $c_s x_s^* - c_t x_s^* < 0$, we have a contradiction to the assumption of x^*. If $c_s x_s^* - c_t x_s^* = 0$, then x is an optimal solution with zero in its sth element.

Case 2: If $x_t^* > 0$, then x_t^* is binding as well and $x_t^* = b_{J_t^*(x^*)}$. We have two subcases to be considered. Case 2-1: if $J_t^*(x^*) \geqslant J_s^*(x^*)$, since $J_s(\bar{x}) \setminus B_s(\bar{x}) \subseteq J_t(\bar{x}) \setminus B_t(\bar{x})$, the constraints satisfied by x_s^* can be sustained by x_t^*. Hence, x_s^* is redundant. Since x^* is optimal, the x_s^* should be zero. Case 2-2: if $J_t^*(x^*) < J_s^*(x^*)$, we can select $x_t = b_{J_s^*(x^*)}$. Then the constraints satisfied by x_t^* and x_s^* can be sustained by x_t. It follows that if we adjust x^* by setting $x_s^* = 0$ and x_t^* by x_t (denote the adjusted x^* by x^{**}), then

$$\sum_{i=1}^{n} c_i x_i^* - \sum_{i=1}^{n} c_i x_i^{**} = c_t x_t^* + c_s x_s^* - c_t x_t = c_t x_t^* + (c_s - c_t) x_s^* > 0.$$

Therefore, the x^* is not an optimal solution, a contradiction.

Rule 2. Suppose that the model (1)–(2) has solution. Let $I_j := \{i \in \mathcal{I} \mid \min\{\bar{x}_i, a_{ij}\} = b_j\}, \forall j \in \mathcal{J}$. If for some $j \in \mathcal{J}$ the $I_j = \{i\}$ is a singleton set, then for any optimal solution x^*, we have $x_i^* \geqslant b_j$. If $a_{ij} > b_j$, then $x_i^* = b_j$. Moreover, we can delete the kth constraint from further consideration if $k < j$ and $i \in I_k$.

Proof. By Theorem 2, a singleton set $I_j = \{i\}$ implies that for any optimal solution x^* the jth constraint is satisfied only by the decision variable x_i^*. That is, $\min(x_i^*, a_{ij}) = b_j$ and $\min(x_r^*, a_{rj}) < b_j$ for all $r \neq i$. This implies that $x_i^* \geqslant b_j$. Obviously, if $a_{ij} > b_j$, then $x_i^* = b_j$. On the other hand, since the model (1)–(2) has solution and $I_j = \{i\}$, the $\bar{x}_i$ is a binding variable. Together with $k < j$ and $i \in I_k$, we have

$$\min\{\bar{x}_i, a_{ij}\} = b_j \text{ and } \min\{\bar{x}_i, a_{ik}\} = b_k. \qquad \square$$

We have two cases to discuss.

Case 1: If $\bar{x}_i = b_j$, then it follows that $x_i^* = b_j$ for any optimal solution x^*. Hence,

$$\min\{x_i^*, a_{ik}\} = \min\{\bar{x}_i, a_{ik}\} = b_k.$$

Case 2: If $\bar{x}_i > b_j$, then by $b_j \geqslant b_k$ and $\min\{\bar{x}_i, a_{ik}\} = b_k$, we have $a_{ik} = b_k$. Since $x_i^* \geqslant b_j$, we have $\min\{x_i^*, a_{ik}\} = b_k$. Both cases imply that the kth constraint can be satisfied automatically by x_i^* as long as we detect the $\bar{x}_i$ is the only binding variable in constraint j.

Rule 3. Compute an initial upper bound for the optimal objective value. The procedure of computing an initial upper bound for the optimal value is essentially the one in [28] (see appendix.) In the current paper, the Rule 3 cooperates with Rules 1 and 2 to reduce the problem. Precisely, if there exists, say, an entry m_{ij} of M strictly larger than the current upper bound, we then set $m_{ij} = \infty$. That is, the x_i^* can NOT be binding in the *j*th constraint. We note that the branch and bound part of Wu, Guu and Liu's method did not update their upper bound. Indeed, during the branch and bound part, the initial upper bound may be improved by a better solution. We shall illustrate this point by Example 1 in next section.

We are ready to present a procedure for solving the problem.

Step 1: Rename the variables and arrange the order of constraints so that the c_is and b_js are ordered increasingly, if necessary.

Step 2: Compute the vector $A \diamond b$ by (4).

Step 3: Check the consistency of equations (2) by verifying whether $(A \diamond b) \circ A = b$. If it is inconsistent, then stop. Otherwise, set the maximum solution $\bar{x} = A \diamond b$.

Step 4: (Optional) Compute I_j for each $j \in \mathcal{J}$.

Step 5: Generate the value matrix *M*.

Step 6: From the (current) matrix *M* compute the index sets $J_i = \{j \in \mathcal{J} | m_{ij} \neq \infty\}$ for all remaining decision variables x_i.

Step 7: Apply the Rule 1 and Rule 2 to determine the values of as many decision variables as possible. Delete the corresponding rows and/or columns in *M* (Hence, the size of the problem is reduced.) Denote the remaining submatrix by *M* again. If all decision variables have been set, go to Step 10.

Step 8: Compute the initial upper bound from *M*. Apply Rule 3 to set some entries of *M* by ∞, if any. If some entries of *M* are set by ∞, then go back to Step 6. Otherwise, go to Step 9.

Step 9: Take the (remaining) value matrix *M*. Employ the backward branch and bound method with jump-tracking technique to solve for the remaining undecided decision variables (Details of this part are illustrated in Example 1. In addition, the initial upper bound will be improved by a better solution during the branch and bound part if possible.)

Step 10: Generate an optimal solution for the original problem.

3. Two examples

In this section, we shall give two examples to illustrate our procedure. In particular, Example 1 is given to illustrate that we may be able to improve the initial upper bound (as computed by Wu, Guu and Liu's method) so that the visited nodes of the branch and bound part can be decreased. Example 2 is given to show the merit of our three rules. With these rules, we may be able to solve some problems without invoking the branch and bound part in new procedure.

Example 1. Consider the following problem.

$$\text{Minimize} \quad Z(y) = 1.2y_1 + 2.5y_2 + 0.6y_3 + y_4 + 1.1y_5$$

$$\text{subject to} \quad \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 \end{bmatrix} \circ \begin{bmatrix} 0.5 & 0.8 & 1.0 & 0.8 & 0.7 \\ 0.5 & 0.9 & 1.0 & 0.8 & 0.2 \\ 0.4 & 0.9 & 1.0 & 0.6 & 0.7 \\ 0.5 & 0.95 & 0.7 & 0.3 & 0.6 \\ 0.45 & 0.6 & 1.0 & 0.8 & 0.7 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5 & 0.9 & 1.0 & 0.8 & 0.7 \end{bmatrix}.$$

Step 1: To make the c_is and b_js be ordered increasingly, we rename the variables $y_3 \to x_1$, $y_4 \to x_2$, $y_5 \to x_3$, $y_1 \to x_4$ and $y_2 \to x_5$ and arrange the order of constraints. The renamed and arranged problem is as follows:

$$\text{Minimize} \quad Z(x) = 0.6x_1 + x_2 + 1.1x_3 + 1.2x_4 + 2.5x_5$$

$$\text{subject to} \quad \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix} \circ \begin{bmatrix} 0.4 & 0.7 & 0.6 & 0.9 & 1.0 \\ 0.5 & 0.6 & 0.3 & 0.95 & 0.7 \\ 0.45 & 0.7 & 0.8 & 0.6 & 1.0 \\ 0.5 & 0.7 & 0.8 & 0.8 & 1.0 \\ 0.5 & 0.2 & 0.8 & 0.9 & 1.0 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.7 & 0.8 & 0.9 & 1.0 \end{bmatrix}.$$

Step 2: Compute the vector $A \diamond b = \begin{bmatrix} 1.0 & 0.9 & 1.0 & 1.0 & 1.0 \end{bmatrix}$.

Step 3: Direct computation shows that $(A \diamond b) \circ A = b$. Hence, the system is consistent and $X(A, b) \neq \emptyset$. Set the maximum solution $\bar{x} = A \diamond b$.

Step 4: (Optional) Compute the index sets I_j for all j. And they are

$$I_1 = \{2, 4, 5\},\ I_2 = \{1, 3, 4\},\ I_3 = \{3, 4, 5\},\ I_4 = \{1, 2, 5\},\ I_5 = \{1, 3, 4, 5\}.$$

Step 5: Generate the value matrix M.

$$\begin{array}{rl} \text{equation} \to & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \end{array}$$

$$M = \begin{array}{c} (x_1^*) \\ (x_2^*) \\ (x_3^*) \\ (x_4^*) \\ (x_5^*) \end{array} \begin{bmatrix} \infty & 0.42 & \infty & 0.54 & 0.6 \\ 0.5 & \infty & \infty & 0.9 & \infty \\ \infty & 0.77 & 0.88 & \infty & 1.1 \\ 0.6 & 0.84 & 0.96 & \infty & 1.2 \\ 1.25 & \infty & 2.0 & 2.25 & 2.5 \end{bmatrix}.$$

Step 6: From the current matrix M compute the index sets J_i for all i. And they are

$$J_1 = \{2, 4, 5\},\ J_2 = \{1, 4\},\ J_3 = \{2, 3, 5\},\ J_4 = \{1, 2, 3, 5\},\ J_5 = \{1, 3, 4, 5\}.$$

Step 7: The Rules 1 and 2 cannot be applied.

Step 8: Compute the initial upper bound from M. First, we compute IZ_i, $i \in \mathcal{I} = \{1, 2, 3, 4, 5\}$ by

$$IZ_i = \max_{j \in J_i} \{c_i b_j\} + c_{t_1^i} b_{g_1^i} + \cdots + c_{t_{k_i}^i} b_{g_{k_i}^i}.$$

$$IZ_1 = \max_{j \in J_1 = \{2,4,5\}} \{c_1 b_j\} + c_2 b_1 + c_3 b_3 = \max\{0.42, 0.54, 0.6\} + 0.5 + 0.88 = 1.98,$$

$$IZ_2 = \max_{j \in J_2 = \{1,4\}} \{c_2 b_j\} + c_3 b_3 + c_1 b_5 = \max\{0.5, 0.9\} + 0.88 + 0.6 = 2.38,$$

$$IZ_3 = \max_{j \in J_3 = \{2,3,5\}} \{c_3 b_j\} + c_2 b_1 + c_1 b_4 = \max\{0.77, 0.88, 1.1\} + 0.5 + 0.54 = 2.14,$$

$$IZ_4 = \max_{j \in J_4 = \{1,2,3,5\}} \{c_4 b_j\} + c_1 b_4 = \max\{0.6, 0.84, 0.96, 1.2\} + 0.54 = 1.74,$$
$$IZ_5 = \max_{j \in J_5 = \{1,3,4,5\}} \{c_5 b_j\} + c_1 b_2 = \max\{1.25, 2.0, 2.25, 2.5\} + 0.42 = 2.92.$$

Then, set

$$\text{initial upper bound} = \min\{IZ_i : i \in \mathcal{I}\} = 1.74.$$

By Rule 3, we can set elements m_{53}, m_{54}, m_{55} of M which are larger than 1.74 to be ∞. Therefore, the resulting value matrix (still denoted by M)

$$\begin{array}{ll} \text{equation} \rightarrow & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \end{array}$$
$$M = \begin{array}{c} (x_1^*) \\ (x_2^*) \\ (x_3^*) \\ (x_4^*) \\ (x_5^*) \end{array} \begin{bmatrix} \infty & 0.42 & \infty & 0.54 & 0.6 \\ 0.5 & \infty & \infty & 0.9 & \infty \\ \infty & 0.77 & 0.88 & \infty & 1.1 \\ 0.6 & 0.84 & 0.96 & \infty & 1.2 \\ 1.25 & \infty & \infty & \infty & \infty \end{bmatrix}.$$

With this value matrix M, we go back Step 6 to compute the current index sets J_i and discover $J_5 = \{1\} \subset J_2 = \{1, 4\}$. By Rule 1, we can set $x_5^* = 0$. Hence, the 5th row of M can be deleted from further consideration. After deletion, the value matrix becomes

$$\begin{array}{ll} \text{equation} \rightarrow & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \end{array}$$
$$M = \begin{array}{c} (x_1^*) \\ (x_2^*) \\ (x_3^*) \\ (x_4^*) \end{array} \begin{bmatrix} \infty & 0.42 & \infty & 0.54 & 0.6 \\ 0.5 & \infty & \infty & 0.9 & \infty \\ \infty & 0.77 & 0.88 & \infty & 1.1 \\ 0.6 & 0.84 & 0.96 & \infty & 1.2 \end{bmatrix}.$$

Now, Rules 1, 2 and 3 can not be applied. We go to Step 9.

Step 9: Employ the backward branch and bound method with jump-tracking technique to solve for the remaining decision variables. The detailed steps within Step 9 is summarized in Fig. 1. On each node, the branching process should be stopped when objective value there is larger than the current upper bound. The "backward" indicates that the branch and bound method starts from the last column and toward to the first column of value matrix. Since we add positive entry to the objective value during each branching step and the finite entries in each row of the value matrix M are columnwise increasing, we expect that the solution tree generated by *backward* branch and bound process should be "smaller" than that by *forward* branch and bound method.

Now given the current value matrix M, we have $I_5 = \{1, 3, 4\}$ from last column. This implies that $\{x_1, x_3, x_4\}$ are three candidates for binding variables. (We can see three branches generated from Node 0 in Fig. 1.) If we select x_1, then its contribution to the objective is 0.6. On the other hand, if we select

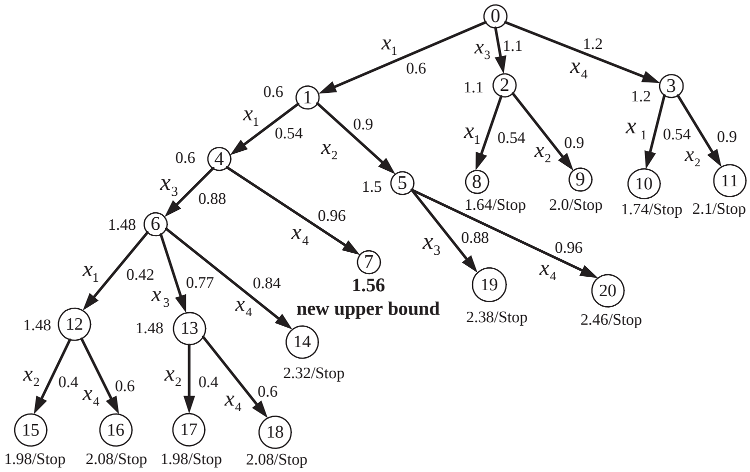

Fig. 1. Backward branch and bound method.

x_3 (x_4), then the objective value is 1.1 (1.2). So far, we have Nodes 1, 2, and 3 with objective values 0.6, 1.1, and 1.2, respectively. Along the branches to each node, we need to check whether the computed binding variables satisfy all equations or not. So far, the identified binding variable on each of Nodes 1, 2, and 3 does not satisfy all equations, respectively. We need further branching to generate solutions. The jump-tracking technique requires us to branch on the node with least objective value. Therefore, we shall branch on Node 1.

We now move backward to the 4th column of M. Since $I_4 = \{1, 2\}$, $\{x_1, x_2\}$ are two candidates for binding variables. In Fig. 1, the branch from Node 1 to Node 4 (Node 5) is generated if x_1 (x_2) is selected. Since variable x_1 is currently binding, the objective value in Node 4 is still 0.6. On the other hand, the objective value in Node 5 is updated to 1.5. The identified binding variables corresponding to the branches to Node 4 and Node 5 do not satisfy all equations, respectively. Further branching is needed to generate solutions. And up to now, we have four nodes (Nodes 2,3,4, and 5) to select for the next branching process. By the jump-tracking technique, we select Node 4 to branch further because of least objective value there.

We move back to the 3rd column of M. Since $I_3 = \{3, 4\}$, $\{x_3, x_4\}$ are two candidates for binding variables. In Fig. 1 we have two branches from Node 4 (that is, Nodes 6 and 7). And the objective value in Node 6 (7) is updated to 1.48 (1.56). In each node, we need to check whether the computed binding variables satisfy all equations or not. Along the branches to Node 6, we have identified $\{x_1, x_3\}$ as binding variables, yet not enough to satisfy all equations. Along the branches to Node 7, the corresponding binding variables $\{x_1, x_4\}$ satisfy all equations. Hence $x_1 = 1.0$, $x_4 = 0.8$, $x_2 = x_3 = x_5 = 0$ is a solution with objective value 1.56, which is better than the initial upper bound 1.74. We update the current upper bound as 1.56. Now, we have four nodes (Nodes 2,3,5, and 6) to select for the next branching process. Again, we select Node 2 due to jump-tracking technique. Since Node 2 corresponds to the 5th column of M, we move back to the 4th column and generate Nodes 8 and 9, respectively. Note that the objective values in both nodes are larger than current upper bound 1.56. Both nodes are fathomed. By continuing this process, a tree with 20 nodes is generated as in Fig. 1.

Step 10: Generate an optimal solution. From Fig. 1, the optimal objective value is 1.56, and the optimal solution is

$$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*) = (1.0, 0, 0, 0.8, 0).$$

The optimal solution of original problem is

$$(y_1^*, y_2^*, y_3^*, y_4^*, y_5^*) = (0.8, 0, 1.0, 0, 0).$$

Indeed, Wu, Guu, and Liu's method needs to visit 34 nodes for solving this problem. This simple example illustrates that our new procedure visits less nodes than our previous method. It is our three rules, the new upper bound and its updating that reduce the branches of the solution tree.

Example 2. Consider the following problem.

$$\begin{aligned} \text{Minimize} \quad & Z(x) = 0.45x_1 + 0.5x_2 + 0.7x_3 + x_4 + 1.1x_5 + 1.4x_6 + 1.5x_7 + 2x_8 + 3.6x_9 \\ \text{subject to} \quad & \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 \end{bmatrix} \circ A \\ & = \begin{bmatrix} 0.2 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 & 0.9 & 0.95 & 1.0 \end{bmatrix}, \end{aligned}$$

$$\text{where} \quad A = \begin{bmatrix} 0.25 & 0.4 & 0.4 & 0.6 & 0.1 & 0.8 & 0.8 & 0.9 & 1.0 \\ 0.12 & 0.1 & 0.5 & 0.5 & 0.6 & 0.8 & 0.95 & 0.8 & 0.8 \\ 0.2 & 0.15 & 0.2 & 0.6 & 0.7 & 0.5 & 0.1 & 0.4 & 0.4 \\ 0.15 & 0.2 & 0.5 & 0.6 & 0.5 & 0.8 & 0.3 & 0.9 & 1.0 \\ 0.1 & 0.14 & 0.5 & 0.6 & 0.6 & 0.2 & 0.9 & 0.4 & 1.0 \\ 0.2 & 0.4 & 0.4 & 0.2 & 0.7 & 0.8 & 0.9 & 0.98 & 0.5 \\ 0.15 & 0.25 & 0.5 & 0.6 & 0.2 & 0.8 & 0.8 & 0.75 & 1.0 \\ 0.2 & 0.4 & 0.45 & 0.2 & 0.7 & 0.85 & 0.5 & 0.9 & 0.6 \\ 0.15 & 0.35 & 0.5 & 0.6 & 0.65 & 0.7 & 0.9 & 0.95 & 0.9 \end{bmatrix}.$$

Step 1: Both of the c_is and b_js are ordered increasingly.
Step 2: Compute $A \diamond b = \begin{bmatrix} 0.2 & 0.9 & 1.0 & 1.0 & 1.0 & 0.95 & 1.0 & 0.8 & 1.0 \end{bmatrix}$.
Step 3: Direct computation illustrates that $(A \diamond b) \circ A = b$. Hence, the system is consistent and $X(A, b) \neq \emptyset$. Set the maximum solution $\bar{x} = A \diamond b$.
Step 4: (Optional) Compute the index sets I_j for all j. And they are

$$I_1 = \{1, 3, 6, 8\}, I_2 = \{6, 8\}, I_3 = \{2, 4, 5, 7, 9\}, I_4 = \{3, 4, 5, 7, 9\}, I_5 = \{3, 6, 8\},$$

$$I_6 = \{2, 4, 6, 7, 8\}, I_7 = \{2, 5, 6, 9\}, I_8 = \{6, 9\}, I_9 = \{4, 5, 7\}.$$

Step 5: Generate the value matrix M.

$$\begin{array}{ll} & \text{equation} \rightarrow \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \\ M = \begin{array}{c} (x_1^*) \\ (x_2^*) \\ (x_3^*) \\ (x_4^*) \\ (x_5^*) \\ (x_6^*) \\ (x_7^*) \\ (x_8^*) \\ (x_9^*) \end{array} & \begin{bmatrix} 0.09 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 0.25 & \infty & \infty & 0.4 & 0.45 & \infty & \infty \\ 0.14 & \infty & \infty & 0.42 & 0.49 & \infty & \infty & \infty & \infty \\ \infty & \infty & 0.5 & 0.6 & \infty & 0.8 & \infty & \infty & 1.0 \\ \infty & \infty & 0.55 & 0.66 & \infty & \infty & 0.99 & \infty & 1.1 \\ 0.28 & 0.56 & \infty & \infty & 0.98 & 1.12 & 1.26 & 1.33 & \infty \\ \infty & \infty & 0.75 & 0.9 & \infty & 1.2 & \infty & \infty & 1.5 \\ 0.4 & 0.8 & \infty & \infty & 1.4 & 1.6 & \infty & \infty & \infty \\ \infty & \infty & 1.8 & 2.16 & \infty & \infty & 3.24 & 3.42 & \infty \end{bmatrix}. \end{array}$$

Step 6: From the current matrix M compute the index sets J_i for all i. And they are

$$J_1 = \{1\},\ J_2 = \{3, 6, 7\},\ J_3 = \{1, 4, 5\},\ J_4 = \{3, 4, 6, 9\},\ J_5 = \{3, 4, 7, 9\},$$

$$J_6 = \{1, 2, 5, 6, 7, 8\},\ J_7 = \{3, 4, 6, 9\},\ J_8 = \{1, 2, 5, 6\},\ J_9 = \{3, 4, 7, 8\}.$$

Step 7: Apply the Rules 1 and 2 to fix as many as possible the decision variables. We first note that

$$J_7 = J_4 \text{ and } J_8 \subset J_6.$$

By Rule 1, we set $x_7^* = x_8^* = 0$. Hence, we can delete the 7th and 8th rows of M. After deletion, the M becomes

$$\begin{array}{ll} & \text{equation} \rightarrow \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \\ M = \begin{array}{c} (x_1^*) \\ (x_2^*) \\ (x_3^*) \\ (x_4^*) \\ (x_5^*) \\ (x_6^*) \\ (x_9^*) \end{array} & \begin{bmatrix} 0.09 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 0.25 & \infty & \infty & 0.4 & 0.45 & \infty & \infty \\ 0.14 & \infty & \infty & 0.42 & 0.49 & \infty & \infty & \infty & \infty \\ \infty & \infty & 0.5 & 0.6 & \infty & 0.8 & \infty & \infty & 1.0 \\ \infty & \infty & 0.55 & 0.66 & \infty & \infty & 0.99 & \infty & 1.1 \\ 0.28 & 0.56 & \infty & \infty & 0.98 & 1.12 & 1.26 & 1.33 & \infty \\ \infty & \infty & 1.8 & 2.16 & \infty & \infty & 3.24 & 3.42 & \infty \end{bmatrix}. \end{array}$$

We note that now the index set $I_2 = \{6\}$. By Rule 2, we have $x_6^* \geqslant b_2 = 0.4$. Note that x_6^* is binding in the first and second columns of M now. This implies that at the optimal solution the corresponding equations to these two columns are satisfied by binding variable x_6^*. Hence the first and second columns can be deleted from further consideration. After deleting the first and second columns from M, we note the remaining elements in the first row are all ∞. It implies that the x_1^* is nonbinding in any columns, hence we set $x_1^* = 0$. Therefore the first row of M is deleted as well. The remaining value matrix M becomes

$$
\begin{array}{lll}
 & \text{equation} \rightarrow & \begin{array}{ccccccc} 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\
M = & \begin{array}{c} (x_2^*) \\ (x_3^*) \\ (x_4^*) \\ (x_5^*) \\ (x_6^*) \\ (x_9^*) \end{array} & \begin{bmatrix} 0.25 & \infty & \infty & 0.4 & 0.45 & \infty & \infty \\ \infty & 0.42 & 0.49 & \infty & \infty & \infty & \infty \\ 0.5 & 0.6 & \infty & 0.8 & \infty & \infty & 1.0 \\ 0.55 & 0.66 & \infty & \infty & 0.99 & \infty & 1.1 \\ \infty & \infty & 0.98 & 1.12 & 1.26 & 1.33 & \infty \\ 1.8 & 2.16 & \infty & \infty & 3.24 & 3.42 & \infty \end{bmatrix}.
\end{array}
$$

Step 8: At this stage, since there are some undecided decision variables, we need to compute an upper bound for the optimal value of this reduced problem. By Wu, Guu and Liu's method, an upper bound is 3. By Rule 3, we find both $m_{97} = 3.24$ and $m_{98} = 3.42$ are larger than this upper bound. We shall set ∞ to m_{97} and m_{98}. Then the value matrix becomes

$$
\begin{array}{lll}
 & \text{equation} \rightarrow & \begin{array}{ccccccc} 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\
M = & \begin{array}{c} (x_2^*) \\ (x_3^*) \\ (x_4^*) \\ (x_5^*) \\ (x_6^*) \\ (x_9^*) \end{array} & \begin{bmatrix} 0.25 & \infty & \infty & 0.4 & 0.45 & \infty & \infty \\ \infty & 0.42 & 0.49 & \infty & \infty & \infty & \infty \\ 0.5 & 0.6 & \infty & 0.8 & \infty & \infty & 1.0 \\ 0.55 & 0.66 & \infty & \infty & 0.99 & \infty & 1.1 \\ \infty & \infty & 0.98 & 1.12 & 1.26 & 1.33 & \infty \\ 1.8 & 2.16 & \infty & \infty & \infty & \infty & \infty \end{bmatrix}.
\end{array}
$$

From the current M, we have the 8th constraint containing exact one binding variable x_6^*. That is, $I_8 = \{6\}$. By Rule 2, we shall set $x_6^* = b_8 = 0.95$. We note that the x_6^* is binding in constraints 5,6,7 as well. We then delete the 5th, 6th, 7th, and 8th constraints from further consideration. The reduced value matrix M becomes

$$
\begin{array}{lll}
 & \text{equation} \rightarrow & \begin{array}{ccc} 3 & 4 & 9 \end{array} \\
M = & \begin{array}{c} (x_2^*) \\ (x_3^*) \\ (x_4^*) \\ (x_5^*) \\ (x_9^*) \end{array} & \begin{bmatrix} 0.25 & \infty & \infty \\ \infty & 0.42 & \infty \\ 0.5 & 0.6 & 1.0 \\ 0.55 & 0.66 & 1.1 \\ 1.8 & 2.16 & \infty \end{bmatrix}.
\end{array}
$$

With the current matrix M, we go back Step 6 to compute index set J_i and discover

$$J_5 = J_4 \text{ and } J_9 \subset J_4.$$

It follows from Rule 1 that we set $x_5^* = x_9^* = 0$. And the corresponding rows are deleted. The reduced matrix M becomes

$$\begin{array}{ll} \text{equation} \rightarrow & \quad 3 \qquad 4 \qquad 9 \\ M = \begin{array}{c} (x_2^*) \\ (x_3^*) \\ (x_4^*) \end{array} & \begin{bmatrix} 0.25 & \infty & \infty \\ \infty & 0.42 & \infty \\ 0.5 & 0.6 & 1.0 \end{bmatrix}. \end{array}$$

Note again that $I_9 = \{4\}$. Hence, by Rule 2 we set $b_9 = 1.0$ to x_4^*. Since x_4^* appears in constraints 3 and 4 as a binding variable, we can delete the 3rd and 4th constraints from further consideration. Now there is no element after deletion in M. And two decision variables x_2^* and x_3^* are set to be zero.

Step 10: An optimal solution yielded by our procedure is

$$x^* = (0, 0, 0, 1.0, 0, 0.95, 0, 0, 0) \text{ and the optimal value is } Z(x^*) = 2.33.$$

Now, using Wu, Guu and Liu's method to solve Example 2 again, we find that their backward branch and bound part has to be implemented. Moreover, there are 37 visited nodes in solution tree. This example illustrates that our new procedure may solve the problem without invoking the branch and bound part of the procedure.

Numerical experiment: In Table 1, we have compared the performance of Wu, Guu and Liu's approach and our new procedure. Here we use the same test examples recorded in Appendix of [28]. The current experiment was programmed by Visual Basic 6.0 on a Pentium III PC with 1000 MHZ and 256-MB RAM. Each testing was terminated if the number of the visited nodes is more than 200,000. Note that since our new procedure utilizes the rules to simplify the problem and update the upper bound as further as possible, the "solution tree" generated by the branch and bound part is "smaller" than that by Wu, Guu and Liu's procedure. Table 1 also illustrates that the new procedure visits less nodes in solution tree.

Note that the number with "$*$" means that the initial upper bound is equal to the optimal objective value. The initial upper bound of testing problems 1, 5–6, 8, 11–13 is updated by the new procedure. Problems 12 and 13 can not be solved by Wu, Guu and Liu's procedure because their procedure visits more than 200,000 nodes, yet our new procedure solves them successfully. Problem 14 remains a challenge because both procedures fail to report the final results.

4. Conclusion

In this paper, we considered an optimization problem involving the minimization of a linear cost function subject to a fuzzy relational equation with max–min composition. We added new theoretical results for the minimization problem. In particular, we established necessary conditions possessed by its optimal solution. For the algorithmic consideration, we proposed three rules to simplify the problem. Testing examples illustrated that our new procedure improved our previous work in a sense of visiting less nodes in solution tree.

Table 1
Performance between Wu, Guu and Liu's procedure and the new procedure

No.	Size of problem (m, n)	Wu et al.'s procedure		New procedure		Optimal value
		No. of nodes visited	Initial upper bound	No. of nodes visited	Incumbent upper bound	
1	(5,5)	46	2.0300	40	1.8200	1.8200
2	(6,6)	23	* 2.6000	23	* 2.6000	2.6000
3	(7,7)	97	* 3.0700	75	* 3.0700	3.0700
4	(8,8)	88	* 2.2750	60	* 2.2750	2.2750
5	(9,9)	135	2.5000	65	2.2600	2.2600
6	(10,10)	157	1.9250	119	1.7200	1.7200
7	(11,10)	37	* 2.0410	5	* 2.0410	2.0410
8	(12,10)	190	1.9250	145	1.7200	1.7200
9	(15,15)	246	* 2.7971	225	* 2.7971	2.7971
10	(16,15)	366	* 1.0189	340	* 1.0189	1.0189
11	(20,20)	8,377	24.8570	6,434	23.5720	23.5720
12	(30,30)	Over 200,000	125.7491	304	125.4599	125.4599
13	(40,40)	Over 200,000	95.4612	34,282	73.0557	73.0557
14	(50,50)	Over 200,000	638.9283	Over 200,000	600.9779	

Appendix

For easy reference, we shall cite the procedure of Wu, Guu, and Liu as follows. Let m_j^* denote the minimum value in column j of matrix M. Precisely, $m_j^* = \min\{m_{ij} : i \in I_j\}$ for all $j \in \mathcal{J}$. The $arg[m_j^*]$ denotes the least i in column j such that $m_j^* = m_{arg[m_j^*]j}$. Consider the following m solutions defined as: for each $i = 1, 2, \ldots, m$, $y_{ij} = 1$ for $j \in J_i$ and $y_{arg[m_j^*]j} = 1$ for $j \notin J_i$. Assume that there are k_i distinct $arg[m_j^*]$ for all $j \notin J_i$, denoted by

$$t_1^i, \ldots, t_{k_i}^i.$$

Let $G_s^i = \{j \notin J_i : arg[m_j^*] = t_s^i\}$ for all $s = 1, 2, \ldots, k_i$. Let g_s^i denote the largest element in G_s^i. Then the objective value, denoted by IZ_i, yielded by the ith solution is computed by

$$IZ_i = \max_{j \in J_i} \{c_i b_j\} + c_{t_1^i} b_{g_1^i} + \cdots + c_{t_{k_i}^i} b_{g_{k_i}^i}.$$

We select our initial upper bound to be

$$\text{initial upper bound} = \min\{IZ_1, \ldots, IZ_m\}.$$

References

[1] G.I. Adamopoulos, C.P. Pappis, Some results on the resolution of fuzzy relation equations, Fuzzy Sets Syst. 60 (1993) 83–88.

[2] B.D. Baets, Analytical solution methods for fuzzy relational equations, in: D. Dubois, H. Prade (Eds.), Fundamentals of Fuzzy Sets, Kluwer, Norwell, MA, 2000, pp. 291–340.
[3] B.D. Baets, E. Tsiporkova, R. Mesiar, Conditioning in possibility theory with strict order norms, Fuzzy Sets Syst. 106 (1999) 221–229.
[4] L. Chen, P.P. Wang, Fuzzy relation equations (I): the general and specialized solving algorithms, Soft Comput. 6 (2002) 428–435.
[5] E. Czogala, J. Drewniak, W. Pedrycz, Fuzzy relation equations on a finite set, Fuzzy Sets Syst. 7 (1982) 89–101.
[6] A. Di Nola, S. Sessa, W. Pedrycz, E. Sanchez, Fuzzy Relational Equations and Their Applications in Knowledge Engineering, Kluwer Academic Press, Dordrecht, 1989.
[7] S.-C. Fang, G. Li, Solving fuzzy relation equations with a linear objective function, Fuzzy Sets Syst. 103 (1999) 107–113.
[8] S.-M. Guu, Y.-K. Wu, Minimizing a linear objective function with fuzzy relation equation constraints, Fuzzy Optim. Decision Making 1 (4) (2002) 347–360.
[9] M. Higashi, G.J. Klir, Resolution of finite fuzzy relation equations, Fuzzy Sets Syst. 13 (1984) 65–82.
[10] G.J. Klir, T.A. Folger, Fuzzy Sets, Uncertainty, and Information, Prentice-Hall, NJ, 1988.
[11] G.J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall, PTR, Englewood Cliffs, NJ, 1995.
[12] H.-C. Lee, S.-M. Guu, On the optimal three-tier multimedia streaming services, Fuzzy Optim. Decision Making 2 (31) (2002) 31–39.
[13] J. Loetamonphong, S.-C. Fang, Optimization of fuzzy relational equations with max-product composition, Fuzzy Sets Syst. 118 (2001) 509–517.
[14] J. Loetamonphong, S.-C. Fang, R.E. Young, Multi-objective optimization problems with fuzzy relation equation constraints, Fuzzy Sets Syst. 127 (2002) 141–164.
[15] J. Lu, S.-C. Fang, Solving nonlinear optimization problems with fuzzy relation equations constraints, Fuzzy Sets Syst. 119 (2001) 1–20.
[16] L. Luoh, W.-J. Wang, Y.-K. Liaw, Matrix-pattern-based computer algorithm for solving fuzzy relation equations, IEEE Trans. Fuzzy Syst. 11 (1) (2003) 100–108.
[17] M. Miyakoshi, M. Shimbo, Sets of solution-set-invariant coefficient matrices of simple fuzzy relation equations, Fuzzy Sets Syst. 21 (1987) 59–83.
[18] C.P. Pappis, G.I. Adamopoulos, A computer algorithm for the solution of the inverse problem of fuzzy systems, Fuzzy Sets Syst. 39 (1991) 279–290.
[19] C.P. Pappis, G.I. Adamopoulos, A software routine to solve the generalized inverse problem of fuzzy systems, Fuzzy Sets Syst. 47 (1992) 319–322.
[20] C.P. Pappis, M. Sugeno, Fuzzy relation equations and the inverse problem, Fuzzy Sets Syst. 15 (1985) 79–90.
[21] K. Peeva, Fuzzy linear systems, Fuzzy Sets Syst. 49 (1992) 339–355.
[22] E. Sanchez, Resolution of composite fuzzy relation equations, Inform. Control 30 (1976) 38–48.
[23] E. Sanchez, Solutions in composite fuzzy relation equations, in: M.M. Gupta, G.N. Saridis, B.R. Gaines (Eds.), Fuzzy Automata and Decision Processes, North-Holland, New York, 1977, pp. 221–234.
[24] G.B. Stamou, S.G. Tzafestas, Resolution of composite fuzzy relation equations based on Archimedean triangular norms, Fuzzy Sets Syst. 120 (2001) 395–407.
[25] H.-F. Wang, A multi-objective mathematical programming problem with fuzzy relation constraints, J. Multi-Criteria Decision Anal. 4 (1995) 23–35.
[26] S. Wang, S.-C. Fang, H.L.W. Nuttle, Solutions sets of interval-valued fuzzy relational equations, Fuzzy Optim. Decision Making 2 (1) (2003) 41–60.
[27] P.Z. Wang, D.Z. Zhang, E. Sanchez, E.S. Lee, Latticized linear programming and fuzzy relation inequalities, J. Math. Anal. Appl. 159 (1) (1991) 72–87.
[28] Y.-K. Wu, S.-M. Guu, J.Y.-C. Liu, An accelerated approach for solving fuzzy relation equations with a linear objective function, IEEE Trans. Fuzzy Syst. 10 (4) (2002) 552–558.