

A new algorithm for geometric optimization with a single-term exponent constrained by bipolar fuzzy relation equations

S. Aliannezhadi¹ and A. Abbasi Molai²

^{1,2}*School of Mathematics and Computer Sciences, Damghan University, P.O.Box 36715-364, Damghan, Iran.*

s.Aliannezhadi@std.du.ac.ir, a_abbasimolai@du.ac.ir

Abstract

A geometric programming problem subject to bipolar max-product fuzzy relation equation constraints is studied in this paper. Some necessary and sufficient conditions are given for its solution existence. A lower and upper bound on the solution set of its feasible domain is obtained. Some sufficient conditions are proposed to determine some its optimal components without its resolution. A modified branch-and-bound method is extended to solve the problem. Moreover, an efficient algorithm is proposed to solve the problem based on the simplification operations and the modified branch-and-bound method. Its computational complexity is carefully analyzed. Some examples are given to show the importance of the problem and to illustrate the process of the algorithm. Finally, an analytic and comparative study is done to show the efficiency of the simplification procedures.

Keywords: Bipolar fuzzy relation equation, geometric programming, max-product composition, modified branch-and-bound method.

1 Introduction

The problems related to fuzzy relation equations (FREs) [17, 18, 24], fuzzy relation inequalities (FRIs) [12, 27, 28, 32, 34], and fuzzy relation programming [14, 26, 33, 39] have extensively been studied by many researchers since fuzzy relation equations were suggested by Sanchez [15] for the first time. The max-min fuzzy relation programming with a linear function was firstly studied by Fang and Li [7]. They converted the problem to a 0-1 integer programming problem and solved it by the branch-and-bound method. Some researchers accelerated their approach by providing some rules [4, 9, 21, 22]. However, all applied problems in the real world cannot be formulated by linear objective functions. The geometric programming problem has always been applied as one of the most important models of the real world in the areas of wireless communication [36], peer-to-peer network system [31, 37], and economic analysis, environmental engineering, and business administration [38]. Motivated by the applications, the researchers in [16, 19, 20, 25, 30, 35, 36, 40, 41] studied the latticized fuzzy relation programming and fuzzy relation geometric programming problem with single-term exponents. The FREs and FRIs applied in the above problems are increasing with respect to the variables. In some application areas such as management, economics, and covering, the variables and their negations participate in FRE system, simultaneously. These systems are called Bipolar FREs (BFREs). The linear programming problem subject to bipolar *FREs* or *FRI*s was firstly considered by Freson et al. [8]. They investigated the structure of its feasible domain. Its nonempty feasible domain can be determined by a finite number of maximal and minimal point pairs. The system of bipolar *FRE* was further considered by Li and Jin [10]. A path-based algorithm was designed to find the complete solution set of the bipolar max-Lukasiewicz *FRE* system by Yang [29]. The bipolar max-product *FRE* systems have been studied by the product negation in [5] and the standard negation in [6]. Li and Liu [11] also studied the linear optimization problem provided to the bipolar max- T_L *FREs* and converted it into a 0-1 integer optimization. Some procedures and rules were designed to reduce its structure in [3, 13]. The nonlinear optimization problem subject to bipolar *FRE* constraints has firstly been considered by Zhou et al. [42]. They converted the problem to a 0-1 mixed

integer optimization. Its resolution by integer programming techniques has a high computational complexity. As an extension of the models proposed in [20, 36, 40, 41], the geometric programming problem with the single-term exponent provided to the max-product BFRE constraints was studied in [2]. Its main focus is to design a new algorithm to solve the problem based on the modified branch-and-bound approach to solve the problem using a value matrix. First of all, some of its special cases were detected in terms of several sufficient conditions. Under the sufficient conditions, its optimal solution can be found without implementing the modified branch-and-bound method on the problem. The main motivation is to answer the following questions:

1. If the sufficient conditions are not satisfied for some instances of the problem, can we detect other sufficient conditions to find some of its optimal components?
2. If the answer is yes, how can we simplify the original problem and obtain a reduced problem with a smaller size or dimension?
3. How much do the simplification procedures reduce the rate of computations?

As we will show later, the simplification procedures can considerably reduce the rate of computations. We also compare the rate of computations in two cases: (1) applying the simplification procedures and (2) without using them as [2]. It will show the efficiency of the proposed simplification procedures or rules in the algorithm. The main contribution of the present paper is to design some rules to simplify the original problem. The rules are presented in terms of some sufficient conditions. Under the conditions, some of the optimal components are determined before implementing the modified branch and bound method on the original problem. Then the original problem is simplified and its dimensions are reduced using the rules as much as possible. Motivated by the researches in [4, 21, 23] for fuzzy relation programming, some rules are proposed to cut down the dimensions of the original problem due to its particular structure. An efficient algorithm is finally designed to find one of its optimum points using the rules and the modified branch-and-bound method. The structure of the paper is organized as follows. The geometric fuzzy relation programming problem with bipolar max-product *FRE* constraints is formulated in Section 2 and the structure of its solution set is investigated. Some theorems and corollaries are presented to detect some of its optimum components without solving it by the modified branch-and-bound method in Section 3. In Section 4, a step-by-step procedure is proposed for its resolution based on a value matrix and the modified branch-and-bound approach. Its computational complexity is analyzed in Section 5. Some examples are given to show its importance and to illustrate the process of the algorithm in Section 6. An analytic and comparative study is done to show the efficiency of the proposed algorithm and the simplification procedures in Section 7. Finally, conclusions are given in Section 8.

2 The formulation of the problem and its structure of feasible domain

The geometric programming problem with single-term exponents subject to bipolar max-product *FRE* constraints can be formulated as follows:

$$\begin{aligned} \text{Minimize } Z(x) &= \bigvee_{j=1}^n (c_j \cdot x_j^{r_j}), \\ \text{Subject to } x &\in S(A^+, A^-, b) := \{x \in [0, 1]^n \mid A^+ \circ x \vee A^- \circ \neg x = b\}, \end{aligned} \quad (1)$$

where $A^+ = (a_{ij}^+)$ and $A^- = (a_{ij}^-)$ are two $m \times n$ fuzzy relation matrices with $0 \leq a_{ij}^+, a_{ij}^- \leq 1$, for each $i \in I = \{1, 2, \dots, m\}$ and $j \in J = \{1, 2, \dots, n\}$. Also, assume that $b = (b_1, \dots, b_m)^T$, $c = (c_1, \dots, c_n)$, and $r = (r_1, \dots, r_n)$, where $b_i \in [0, 1]$, $c_j \geq 0$, and $r_j \geq 0$, for each $i \in I$ and $j \in J$. Moreover, $x = (x_1, \dots, x_n)^T$ and $\neg x = (1-x_1, \dots, 1-x_n)^T$ are called the decision variable vector and the negation of decision variable vector, respectively. The operator of "o" represents the max-product composition. In this paper, we express the results for case $c \geq 0$, and $r \geq 0$, but $c \neq 0$ and $r \neq 0$, throughout this paper. The constraint part of problem (1) is to find a set of solution vectors $x \in [0, 1]^n$ such that

$$\max_{j \in J} \max \{a_{ij}^+ \cdot x_j, a_{ij}^- \cdot (1-x_j)\} = b_i, \quad \forall i \in I. \quad (2)$$

A system of bipolar max-product *FRE* (2) is called consistent if $S(A^+, A^-, b) \neq \emptyset$. Otherwise, it is inconsistent.

Lemma 2.1. [2] *The vector of $\tilde{x} = (\tilde{x}_1, \dots, \tilde{x}_n)^T$ is the lower bound on the solution set of system (2) where $\tilde{x}_j = \max_{i \in I} \left\{ 1 - \frac{b_i}{a_{ij}^-} \mid a_{ij}^- > b_i \right\}$, for each $j \in J$. Also, the vector of $\hat{x} = (\hat{x}_1, \dots, \hat{x}_n)^T$ is the upper bound on the solution set of system (2) where $\hat{x}_j = \min_{i \in I} \left\{ \frac{b_i}{a_{ij}^+} \mid a_{ij}^+ > b_i \right\}$, for each $j \in J$, where $\max \emptyset = 0$ and $\min \emptyset = 1$.*

Archive of SID

Remark 2.2. [11] If $S(A^+, A^-, b) \neq \emptyset$, then $\check{x} \leq \hat{x}$. Also, if $x \in S(A^+, A^-, b)$, then $\check{x} \leq x \leq \hat{x}$, but its converse is not true.

Proposition 2.3. [2] (1) Assume that $S(A^+, A^-, b) \neq \emptyset$. If there exists $j_0 \in J$ such that $\check{x}_{j_0} = \hat{x}_{j_0}$, then for all $x \in S(A^+, A^-, b)$, we have $x_{j_0} = \check{x}_{j_0} = \hat{x}_{j_0}$. Moreover, let $\bar{I} = \{i \in I \mid \max\{a_{ij_0}^+ \cdot x_{j_0}, a_{ij_0}^- \cdot (1 - x_{j_0})\} = b_i\}$ and assume that $\bar{I} \neq \emptyset$. Then we can remove the constraints $i, i \in \bar{I}$, and the variable x_{j_0} from system (2) when $\check{x}_j \leq x_j \leq \hat{x}_j, \forall j \in J$. (2) Suppose that $S(A^+, A^-, b) \neq \emptyset$ and $I_0 = \{i \in I \mid b_i = 0\}$. Then we can remove the constraints $i, i \in I_0$, from system (2) when $\check{x}_j \leq x_j \leq \hat{x}_j, \forall j \in J$.

Without loss of generality, suppose that $\check{x}_j < \hat{x}_j$, for each $j \in J$ and $b_i > 0$, for each $i \in I$.

Definition 2.4. (1) [2] The characteristic matrix $\tilde{Q} = (\tilde{q}_{ij})_{m \times n}$ is defined as follows:

$$\tilde{q}_{ij} = \begin{cases} \{\check{x}_j\}, & \text{if } a_{ij}^- \cdot (1 - \check{x}_j) = b_i \neq a_{ij}^+ \cdot \hat{x}_j, \\ \{\hat{x}_j\}, & \text{if } a_{ij}^- \cdot (1 - \check{x}_j) \neq b_i = a_{ij}^+ \cdot \hat{x}_j, \\ \{\check{x}_j, \hat{x}_j\}, & \text{if } a_{ij}^- \cdot (1 - \check{x}_j) = b_i = a_{ij}^+ \cdot \hat{x}_j, \\ \emptyset, & \text{otherwise.} \end{cases} \quad (3)$$

(2) [11] Matrices $Q^+ = (q_{ij}^+)_{m \times n}$ and $Q^- = (q_{ij}^-)_{m \times n}$ are defined as follows:

$$q_{ij}^+ = \begin{cases} 1, & \text{if } \hat{x}_j \in \tilde{q}_{ij}, \\ 0, & \text{otherwise,} \end{cases} \quad \text{and} \quad q_{ij}^- = \begin{cases} 1, & \text{if } \check{x}_j \in \tilde{q}_{ij}, \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

for each $i \in I$ and $j \in J$.

(3) [2] Let $I_j^+(x) = \{i \in I \mid x_j = \hat{x}_j \text{ and } q_{ij}^+ = 1\}$, $J_i^+(x) = \{j \in J \mid x_j = \hat{x}_j \text{ and } q_{ij}^+ = 1\}$, $I_j^-(x) = \{i \in I \mid x_j = \check{x}_j \text{ and } q_{ij}^- = 1\}$, and $J_i^-(x) = \{j \in J \mid x_j = \check{x}_j \text{ and } q_{ij}^- = 1\}$, for each $i \in I$ and $j \in J$. Moreover, let $I_j(x) = I_j^+(x) \cup I_j^-(x)$, for each $j \in J$.

Theorem 2.5. [11] A vector $x \in [0, 1]^n$ is a solution for the system of bipolar max-product FRE (2) if and only if $\check{x} \leq x \leq \hat{x}$ and the induced binary matrix $Q^x = (q_{ij}^x)_{m \times n}$ doesn't have any zero rows where

$$q_{ij}^x = \begin{cases} 1, & \text{if } x_j \in \tilde{q}_{ij}, \\ 0, & \text{otherwise.} \end{cases}$$

Notation 2.6. [2] (i) Let $I_j^+ = I_j^+(\hat{x})$, $J_i^+ = J_i^+(\hat{x})$, $I_j^- = I_j^-(\check{x})$, and $J_i^- = J_i^-(\check{x})$, for each $i \in I$ and $j \in J$.

(ii) Considering part (2) of Definition 2.4, the statement of "the induced binary matrix $Q^x = (q_{ij}^x)_{m \times n}$ doesn't have any zero rows" in Theorem 2.5 is equivalent to " $\bigcup_{j \in J} I_j(x) = I$ ".

We can apply the following theorem to check the consistency of bipolar max-product FRE (2).

Theorem 2.7. [10] A system of bipolar max-product FRE (2) is consistent if and only if its characteristic boolean formula $C = \bigwedge_{i \in I} C_i$ is well-defined and satisfiable, where $C_i = \bigvee_{j \in J_i^+} y_j \vee \bigvee_{j \in J_i^-} \neg y_j$ and $y_j, \neg y_j \in \{0, 1\}$.

Note that in Theorem 2.7, for each $j \in J$, the value \hat{x}_j is labeled with the positive literal y_j and the value \check{x}_j is labeled with the negative literal $\neg y_j$, respectively.

Theorem 2.8. [2] Let \check{x} and \hat{x} be the lower and upper bound, respectively. Then

$$I_j(x) \subseteq I_j^+ \cup I_j^-, \quad \forall x \in S(A^+, A^-, b), \quad \forall j \in J.$$

Exactly, we have $I_j(x) = I_j^+$ (when $x_j = \hat{x}_j$) or $I_j(x) = I_j^-$ (when $x_j = \check{x}_j$) or $I_j(x) = \emptyset$ (when $\check{x}_j < x_j < \hat{x}_j$).

3 Some sufficient conditions for simplification of problem (1)

In this section, some sufficient conditions are suggested to simplify the problem (1). Under the conditions, we can determine some components of an optimal solution without solving problem (1), directly.

From now on, we assume that the bipolar *FRE* (2) is consistent.

It is easily seen that one of the minimal solutions is an optimal solution of problem (1) with regard to Ref. [40]. In the problem (1), we have two binary characteristic matrices instead of one matrix and we can select either I_j^+ or I_j^- , for each $j \in J$ (if $I_j^+ \neq I_j^-$). Hence, the used methods in [20, 40] cannot be applied to solve problem (1).

One possible method for finding an optimal solution is to compute all the minimal solutions and compare their corresponding objective values to obtain an optimal solution. We know that computing all the minimal solutions is difficult, but obtaining the lower bound \tilde{x} is not difficult. If $\bigcup_{j \in J} I_j^- = I$, then $x^* = \tilde{x}$ is an optimal solution for problem

(1). There exists an interesting property among components of an optimal solution of problem (1) and lower and upper bound vector \hat{x} and \tilde{x} as follows.

Lemma 3.1. [2] *Assume that $S(A^+, A^-, b) \neq \emptyset$. Then there exists an optimal solution $x^* = (x_1^*, \dots, x_n^*)^T$ for problem (1) such that for each $j \in J$ either $x_j^* = \hat{x}_j$ or $x_j^* = \tilde{x}_j$.*

We now present some sufficient conditions to determine some of the optimal variables of problem (1). First of all, the following remark is expressed.

Remark 3.2. *Suppose that some components of an optimal solution x^* are determined under some sufficient conditions in the resolution process of problem (1). If the value \tilde{x}_k is selected for the k^{th} component of an optimal solution x^* , then row(s) $i \in I_k^-$ and column k can be deleted from the matrices Q^+ and Q^- . Also, if the value \hat{x}_k is selected for the k^{th} component of an optimal solution x^* , then row(s) $i \in I_k^+$ and column k can be removed from the matrices Q^+ and Q^- .*

We are now ready to express the following theorem.

Theorem 3.3. *If there exists some $i \in I$ such that $|J_i^+| + |J_i^-| = 1$, i.e., $J_i^+ = \{k\}$ and $J_i^- = \emptyset$ (or $J_i^- = \{k\}$ and $J_i^+ = \emptyset$), then for any optimal solution $x^* = (x_j^*)_{j \in J}$ we have $x_k^* = \hat{x}_k$ (or $x_k^* = \tilde{x}_k$).*

Proof. Since each optimal solution $x^* = (x_j^*)_{j \in J}$ is a feasible solution for system (2), we have to set $x_k^* = \hat{x}_k$ (or $x_k^* = \tilde{x}_k$) with regard to Theorem 2.5 and part (ii) of Notation 2.6. \square

Theorem 3.4. *If there exists a pair of nonempty index sets $K_1, K_2 \subset J$ such that the following conditions are satisfied:*

1. $K_1 \cap K_2 = \emptyset$,
2. $\bigcup_{j \in K_1} I_j^+ \setminus \bigcup_{j \in K_1} I_j^- \subseteq \bigcup_{j \in K_2} I_j^+$,
3. $\bigcup_{j \in K_1} I_j^- \supseteq \bigcup_{j \in K_2} I_j^- \setminus \bigcup_{j \in K_2} I_j^+$,
4. $c_j \cdot \hat{x}_j^{r_j} \geq \bigvee_{j \in K_2} (c_j \cdot \hat{x}_j^{r_j}), \forall j \in K_1$,

then there exists an optimal solution $x^ = (x_j^*)_{j \in J}$ such that $x_j^* = \tilde{x}_j$, for each $j \in K_1$.*

Proof. Suppose that x^{**} is an optimal solution for problem (1) and there exists an index set $\emptyset \neq E_1 \subseteq K_1$ such that $\tilde{x}_j < x_j^{**} \leq \hat{x}_j$, for each $j \in E_1$. Without loss of generality, assume that $x_j^{**} = \tilde{x}_j$, for each $j \in K_1 \setminus E_1$. If $\tilde{x}_j < x_j^{**} < \hat{x}_j$, for each $j \in E_1$, then set $x_j^* = \tilde{x}_j$, for each $j \in E_1$ and $x_j^* = x_j^{**}$, for each $j \in J \setminus E_1$. It is easily seen that the vector x^* is a feasible solution for system (2) and $Z(x^*) \leq Z(x^{**})$ with regard to Theorems 2.5 and 2.8 and part (ii) of Notation 2.6. Otherwise, we have $x_j^{**} = \hat{x}_j$, for each $j \in F_1$ where $\emptyset \neq F_1 \subseteq E_1$, $\tilde{x}_j < x_j^{**} < \hat{x}_j$, for each $j \in E_1 \setminus F_1$, and $x_j^{**} = \tilde{x}_j$, for each $j \in K_1 \setminus E_1$. Now, we can consider two following cases:

Case 1. $\tilde{x}_j \leq x_j^{**} < \hat{x}_j$, for each $j \in E_2$, where $\emptyset \neq E_2 \subseteq K_2$.

Case 2. $x_j^{**} = \hat{x}_j$, for each $j \in K_2$.

In the both cases, we will change some components of vector x^{**} to obtain a feasible solution x^* with $Z(x^*) \leq Z(x^{**})$ as follows:

Case 1. In this case, without loss of generality, assume that $x_j^{**} = \hat{x}_j$, for each $j \in K_2 \setminus E_2$. With regard to condition 1, put $x_j^* = \tilde{x}_j$, for each $j \in E_1$, $x_j^* = \hat{x}_j$ for each $j \in E_2$, and $x_j^* = x_j^{**}$, for each $j \in J \setminus (E_1 \cup E_2)$. It is obvious that $\tilde{x} \leq x^* \leq \hat{x}$. It is easily seen that $\bigcup_{j \in F_1} I_j^+ \subseteq \bigcup_{j \in K_1} I_j^+ \subseteq (\bigcup_{j \in K_2} I_j^+) \cup (\bigcup_{j \in K_1} I_j^-)$ and $\bigcup_{j \in E_2} I_j^- \subseteq \bigcup_{j \in K_2} I_j^- \subseteq (\bigcup_{j \in K_2} I_j^+) \cup (\bigcup_{j \in K_1} I_j^-)$

Archive of SID

with regard to conditions 2 and 3. So, we have

$$\begin{aligned}
\bigcup_{j \in J} I_j(x^{**}) &= \left(\bigcup_{j \in J \setminus (K_1 \cup K_2)} I_j(x^{**}) \right) \cup \left(\bigcup_{j \in F_1} I_j^+ \right) \cup \left(\bigcup_{j \in K_1 \setminus E_1} I_j^- \right) \cup \left(\bigcup_{j \in E_2} I_j(x^{**}) \right) \cup \left(\bigcup_{j \in K_2 \setminus E_2} I_j^+ \right) \\
&\subseteq \left(\bigcup_{j \in J \setminus (K_1 \cup K_2)} I_j(x^{**}) \right) \cup \left(\bigcup_{j \in F_1} I_j^+ \right) \cup \left(\bigcup_{j \in K_1 \setminus E_1} I_j^- \right) \cup \left(\bigcup_{j \in E_2} I_j^- \right) \cup \left(\bigcup_{j \in K_2 \setminus E_2} I_j^+ \right) \\
&\subseteq \left(\bigcup_{j \in J \setminus (K_1 \cup K_2)} I_j(x^{**}) \right) \cup \left(\bigcup_{j \in K_2} I_j^+ \right) \cup \left(\bigcup_{j \in K_1} I_j^- \right) \cup \left(\bigcup_{j \in K_1 \setminus E_1} I_j^- \right) \cup \left(\bigcup_{j \in K_2} I_j^+ \right) \cup \left(\bigcup_{j \in K_1} I_j^- \right) \cup \left(\bigcup_{j \in K_2 \setminus E_2} I_j^+ \right) \\
&= \left(\bigcup_{j \in J \setminus (K_1 \cup K_2)} I_j(x^{**}) \right) \cup \left(\bigcup_{j \in K_2} I_j^+ \right) \cup \left(\bigcup_{j \in K_1} I_j^- \right) \\
&= \bigcup_{j \in J} I_j(x^*).
\end{aligned}$$

Since $\check{x} \leq x^* \leq \hat{x}$ and $\bigcup_{j \in J} I_j(x^*) \supseteq \bigcup_{j \in J} I_j(x^{**})$, vector x^* is a feasible solution for system (2) with regard to Theorem 2.5 and part (ii) of Notation 2.6. Furthermore, we have:

$$\begin{aligned}
Z(x^*) &= \left(\bigvee_{j \in J \setminus (E_1 \cup E_2)} (c_j \cdot x_j^{*r_j}) \right) \vee \left(\bigvee_{j \in E_1} (c_j \cdot \check{x}_j^{r_j}) \right) \vee \left(\bigvee_{j \in E_2} (c_j \cdot \hat{x}_j^{r_j}) \right) \\
&\leq \left(\bigvee_{j \in J \setminus (E_1 \cup E_2)} (c_j \cdot x_j^{*r_j}) \right) \vee \left(\bigvee_{j \in F_1} (c_j \cdot \hat{x}_j^{r_j}) \right) \vee \left(\bigvee_{j \in E_1 \setminus F_1} (c_j \cdot x_j^{**r_j}) \right) \vee \left(\bigvee_{j \in F_1} (c_j \cdot \hat{x}_j^{r_j}) \right) \\
&\leq \left(\bigvee_{j \in J \setminus (E_1 \cup E_2)} (c_j \cdot x_j^{*r_j}) \right) \vee \left(\bigvee_{j \in F_1} (c_j \cdot \hat{x}_j^{r_j}) \right) \vee \left(\bigvee_{j \in E_1 \setminus F_1} (c_j \cdot x_j^{**r_j}) \right) \vee \left(\bigvee_{j \in E_2} (c_j \cdot x_j^{**r_j}) \right) = Z(x^{**}).
\end{aligned}$$

The first inequality is derived from $\bigvee_{j \in E_1} (c_j \cdot \check{x}_j^{r_j}) \leq \left(\bigvee_{j \in F_1} (c_j \cdot \hat{x}_j^{r_j}) \right) \vee \left(\bigvee_{j \in E_1 \setminus F_1} (c_j \cdot x_j^{**r_j}) \right)$, $\bigvee_{j \in F_1} (c_j \cdot \hat{x}_j^{r_j}) \geq c_j \cdot \hat{x}_j^{r_j}$, for each $j \in F_1$ and $c_j \cdot \hat{x}_j^{r_j} \geq \bigvee_{j \in K_2} (c_j \cdot \hat{x}_j^{r_j}) \geq \bigvee_{j \in E_2} (c_j \cdot \hat{x}_j^{r_j})$, for each $j \in K_1$. Therefore, we have $Z(x^*) \leq Z(x^{**})$. Hence, such an optimal solution x^* must exist with $x_j^* = \check{x}_j$, for each $j \in K_1$, in this case.

Case 2. In this case, its proof is similar to the proof of Case 1. We give only the main ideas of the proof. With regard to condition 1, put $x_j^* = \check{x}_j$, for each $j \in E_1$ and $x_j^* = x_j^{**}$, for each $j \in J \setminus E_1$. We have:

$$\begin{aligned}
\bigcup_{j \in J} I_j(x^{**}) &= \left(\bigcup_{j \in J \setminus (K_1 \cup K_2)} I_j(x^{**}) \right) \cup \left(\bigcup_{j \in F_1} I_j^+ \right) \cup \left(\bigcup_{j \in K_1 \setminus E_1} I_j^- \right) \cup \left(\bigcup_{j \in K_2} I_j^+ \right) \\
&\subseteq \left(\bigcup_{j \in J \setminus (K_1 \cup K_2)} I_j(x^{**}) \right) \cup \left(\bigcup_{j \in K_2} I_j^+ \right) \cup \left(\bigcup_{j \in K_1} I_j^- \right) \cup \left(\bigcup_{j \in K_1 \setminus E_1} I_j^- \right) \cup \left(\bigcup_{j \in K_2} I_j^+ \right) \\
&= \left(\bigcup_{j \in J \setminus (K_1 \cup K_2)} I_j(x^{**}) \right) \cup \left(\bigcup_{j \in K_2} I_j^+ \right) \cup \left(\bigcup_{j \in K_1} I_j^- \right) = \bigcup_{j \in J} I_j(x^*).
\end{aligned}$$

The inequality $Z(x^*) \leq Z(x^{**})$ can easily be obtained as follows:

$$Z(x^*) = \left(\bigvee_{j \in J \setminus E_1} (c_j \cdot x_j^{*r_j}) \right) \vee \left(\bigvee_{j \in E_1} (c_j \cdot \check{x}_j^{r_j}) \right) \leq \left(\bigvee_{j \in J \setminus E_1} (c_j \cdot x_j^{*r_j}) \right) \vee \left(\bigvee_{j \in E_1} (c_j \cdot x_j^{**r_j}) \right) = Z(x^{**}).$$

So, such an optimal solution x^* must also exist with $x_j^* = \check{x}_j$, for each $j \in K_1$, in this case.

A direct result of Theorems 3.3 and 3.4 and Remark 3.2 is the following corollary.

Corollary 3.5. *If there exist $i \in I$ and $k \in J_i^+$ such that the following conditions are satisfied:*

1. $J_i^- = \emptyset$,
2. $\bigcup_{j \in J_i^+ \setminus \{k\}} I_j^+ \setminus \bigcup_{j \in J_i^+ \setminus \{k\}} I_j^- \subseteq I_k^+$,
3. $\bigcup_{j \in J_i^+ \setminus \{k\}} I_j^- \supseteq I_k^- \setminus I_k^+$,
4. $c_j \cdot \hat{x}_j^{r_j} \geq c_k \cdot \hat{x}_k^{r_k}$, $\forall j \in J_i^+ \setminus \{k\}$,

then there exists an optimal solution $x^ = (x_j^*)_{j \in J}$ such that $x_k^* = \hat{x}_k$ and $x_j^* = \check{x}_j$, for each $j \in J_i^+ \setminus \{k\}$.*

Theorem 3.6. *If the value \check{x}_k (or \hat{x}_k) is selected for the k^{th} component of an optimal solution and there exists $h \in J \setminus \{k\}$ such that $I_h^+ \setminus I_h^- \subseteq I_k^-$ (or I_k^+), then there exists an optimal solution $x^* = (x_j^*)_{j \in J}$ such that $x_h^* = \check{x}_h$.*

Proof. Suppose that x^{**} is an optimal solution for problem (1). Also, suppose that $x_k^{**} = \check{x}_k$, and $\check{x}_h < x_h^{**} \leq \hat{x}_h$. We give the proof only for the first case (\check{x}_k). The proof of the other case (\hat{x}_k) is similar. We will change x^{**} to obtain a feasible solution x^* without increasing the objective value. Solution x^* can be built by putting $x_h^* = \check{x}_h$ and $x_j^* = x_j^{**}$, for each $j \in J \setminus \{h\}$. Since $I_h^+ \subseteq I_k^- \cup I_h^-$ we have:

$$\begin{aligned} \bigcup_{j \in J} I_j(x^{**}) &= \left(\bigcup_{j \in J \setminus \{h,k\}} I_j(x^{**}) \right) \cup (I_h(x^{**})) \cup (I_k^-) \subseteq \left(\bigcup_{j \in J \setminus \{h,k\}} I_j(x^{**}) \right) \cup (I_h^+) \cup (I_k^-) \\ &\subseteq \left(\bigcup_{j \in J \setminus \{h,k\}} I_j(x^{**}) \right) \cup (I_k^-) \cup (I_h^-) \cup (I_k^-) = \bigcup_{j \in J} I_j(x^*). \end{aligned}$$

On the other hand, $\check{x} \leq x^* \leq \hat{x}$. So, vector x^* is a feasible solution for system (2) with regard to Theorem 2.5 and part (ii) of Notation 2.6. Moreover, we can write:

$$Z(x^*) = \left(\bigvee_{j \in J \setminus \{h\}} (c_j \cdot x_j^{**r_j}) \right) \vee (c_h \cdot \check{x}_h^{r_h}) \leq \left(\bigvee_{j \in J \setminus \{h\}} (c_j \cdot x_j^{**r_j}) \right) \vee (c_h \cdot x_h^{**r_h}) = Z(x^{**}).$$

The inequality is derived from $c_h \geq 0$ and $r_h \geq 0$. Hence, there exists such an optimal solution $x^* = (x_j^*)_{j \in J}$ with $x_h^* = \check{x}_h$. \square

Up to now, some sufficient conditions were presented to determine some components of an optimal solution x^* of problem (1). Delete the determined variables and the corresponding constraints from problem (1) to reduce the problem. Set $J^* = \{j \in J \mid x_j^* \text{ has been determined}\}$ and $L^* = \bigvee_{j \in J^*} (c_j \cdot x_j^{**r_j})$ and also update the index set J as $J = J \setminus J^*$. Obviously, $J \cup J^* = \{1, \dots, n\}$ and $J \cap J^* = \emptyset$. Without loss of generality, assume that $\bigvee_{j \in J} (c_j \cdot x_j^{r_j})$ is the objective value of the reduced problem. Note that for each $x = (x_j)_{j \in J}$, we have $Z(x) = \bigvee_{j \in J} (c_j \cdot x_j^{r_j}) \vee L^*$ where it is not necessarily equal to $\bigvee_{j \in J} (c_j \cdot x_j^{r_j})$.

4 An algorithm for resolution of problem (1)

In this section, we will define a value matrix M to capture all the properties of problem (1) based on the lower and upper bound \check{x} and \hat{x} . Also, we will explain the modified branch-and-bound method with the jump-tracking technique to solve the optimization problem (1). Finally, we will express the results associated with the previous sections and the above concepts in terms of an algorithm.

Considering Theorem 2.8 and Lemma 3.1, there exists an optimal solution x^* for the optimization problem (1) such that $I_j(x^*) = I_j^+$ or $I_j(x^*) = I_j^-$, for each $j \in J$. With regard to the above concepts, we can define the value matrix $M = (m_{ij})_{|I| \times 2|J|}$ as follows:

$$m_{i,2j-1} = \begin{cases} c_j \cdot \hat{x}_j^{r_j}, & \text{if } q_{ij}^+ = 1, \\ \infty, & \text{otherwise,} \end{cases} \quad \text{and} \quad m_{i,2j} = \begin{cases} c_j \cdot \check{x}_j^{r_j}, & \text{if } q_{ij}^- = 1, \\ \infty, & \text{otherwise,} \end{cases} \quad \text{www.SID.ir} \quad (5)$$

Archive of SID

for each $i \in I$ and $j \in J$.

We will employ the branch-and-bound method with the jump-tracking technique to solve problem (1) with the value matrix M according to [2]. We now propose an algorithm to find an optimal solution of the optimization problem (1).

Algorithm 1. An algorithm for solving problem (1).

Step 1. Calculate the lower and upper bound \tilde{x} and \hat{x} with regard to Lemma 2.1.

Step 2. If $b_i > 0$, for each $i \in I$ and $\tilde{x}_j < \hat{x}_j$, for each $j \in J$, then go to Step 3. Otherwise, apply Proposition 2.3.

Step 3. Produce two characteristic matrices Q^+ and Q^- by parts (1) and (2) of Definition 2.4.

Step 4. Find the index sets I_j^+ , I_j^- , J_i^+ , and J_i^- with regard to part (3) of Definition 2.4 and part (i) of Notation 2.6.

Step 5. Check the consistency of bipolar system (2) using Theorem 2.7 and part (i) of Notation 2.6. If it is inconsistent, then stop! Otherwise, go to Step 6.

Step 6. Implement the following substeps.

6.1. If $\bigcup_{j \in J} I_j^- = I$, then the vector $x^* = \tilde{x}$ is an optimal solution by the point mentioned in Section 3 and go to

Step 10.

6.2. If $c = 0$ or $r = 0$, then any $x \in S(A^+, A^-, b)$ is an optimal solution for problem (1) (in this case, the objective function becomes a fixed number). Such a feasible solution x can be created by the characteristic boolean formula in Theorem 2.7. Go to Step 10.

6.3. If there exists some $i \in I$ such that $|J_i^+| + |J_i^-| = 1$, i.e., $J_i^+ = \{k\}$ and $J_i^- = \emptyset$ (or $J_i^- = \{k\}$ and $J_i^+ = \emptyset$), then for any optimal solution $x^* = (x_j^*)_{j \in J}$, we have $x_k^* = \hat{x}_k$ (or $x_k^* = \tilde{x}_k$) with regard to Theorem 3.3. If the conditions of Theorem 3.6 are satisfied, then under the condition, let $x_h^* = \tilde{x}_h$. Moreover, remove their corresponding row(s) and column(s) from the matrices Q^+ and Q^- according to Remark 3.2.

6.4. If the conditions of Corollary 3.5 holds, then some components of an optimal solution can be determined by Corollary 3.5. Also, check the conditions of Theorem 3.6. Then remove row(s) and column(s) of matrices Q^+ and Q^- applying Remark 3.2.

6.5. Check the conditions of Theorem 3.4. If the conditions are satisfied, then some of the optimal variables of problem (1) can be determined. If the conditions of Theorem 3.6 are satisfied, then use it to determine the optimal values of variables. Also, remove row(s) and column(s) of matrices Q^+ and Q^- according to Remark 3.2.

It is noticeable that if we reduced the problem by one of rules 6.1–6.5, then we should again check the rules 6.1–6.5 for the reduced problem. If the conditions are satisfied, then we can use their results for further simplifications.

Step 7. If $Q^+ = Q^- = \emptyset$, then assign \tilde{x}_j to x_j^* and go to Step 10.

Step 8. Determine the value matrix M using relation (5).

Step 9. Employ the modified branch-and-bound method with the jump-tracking technique on matrix M to solve the reduced problem.

Step 10. Determine the optimal solution and the optimal objective value of problem (1) from the above steps. End. In the next section, the computational complexity of Algorithm 1 is analyzed.

5 Computational complexity of algorithm 1

In this section, we analyze the computational complexity of Algorithm 1. To do this, we firstly compute the computational cost for each step of the algorithm. The computational costs of steps 1-5, substeps 6.1-6.2, and steps 8-10 are similar to [2]. Therefore, we briefly mention their computational costs. The computational costs of substeps 6.3-6.5 and step 7 are completely discussed because they have not been analyzed up to now. Finally, the computational complexity of Algorithm 1 is obtained with regard to that of its steps. The parameters m and n denote the number of rows and columns of matrices A^+ (or A^-) in problem (1), respectively.

Step 1. The total required time for Step 1, is as $T_1 = O(mn)$.

Step 2. The total required time for Step 2, in the worst case, is as $T_2 = O(m + n)$.

Step 3. The total required time for Step 3, is as $T_3 = O(mn)$.

Step 4. The total required time for Step 4, is as $T_4 = O(mn)$.

Step 5. The computational complexity of this step is as $T_5 = O(mn2^n)$.

Step 6. This step contains five substeps. To compute its computational complexity, we should compute the computational cost of each substep.

6.1. Its computational complexity is as: $T_{6.1} = O(nm^2)$.

6.2. Its computational complexity is as: $T_{6.2} = O(n)$.

6.3. At first, we should check whether $|J_i^+| + |J_i^-| = 1$ or not, for each $i \in I$. The required time for check-

ing the condition is assumed as: $O(1)$. The computational complexity of checking the condition, for each $i \in I$, is $T^1 = m \times O(1) = O(m)$. On the other hand, the required time for setting $x_k^* = \hat{x}_k$ (or $x_k^* = \check{x}_k$) is $O(1)$. Hence, the computational complexity for checking the conditions of Theorem 3.3 is as $T^2 = T^1 + O(1) = O(m) + O(1) = O(m)$.

The required time for checking the condition $I_h^+ \setminus I_h^- \subseteq I_k^-$ (or I_k^+), for one fixed k and $h \in J \setminus \{k\}$ in Theorem 3.6, is as follows. For finding the elements of set $I_h^+ \setminus I_h^-$, we need to compare each element of set I_h^+ with all of the elements of set I_h^- . Hence, its computational complexity is as $T^3 = m^2 \times O(1) = O(m^2)$ where $O(1)$ is the required time for comparing two elements. Also, checking condition $I_h^+ \setminus I_h^- \subseteq I_k^-$ or $I_h^+ \setminus I_h^- \subseteq I_k^+$ needs to compare each element of set $I_h^+ \setminus I_h^-$ with I_k^- (or I_k^+). Hence, it needs $T^4 = 2O(m^2) = O(m^2)$ time, in the worst case, due to $|I_h^+ \setminus I_h^-| \leq m$, $|I_k^-| \leq m$, and $|I_k^+| \leq m$. In the worst case, the computational complexity for computing $I_h^+ \setminus I_h^-$ and checking $I_h^+ \setminus I_h^- \subseteq I_k^-$ (or I_k^+) is as $T^5 = T^3 + T^4 = O(m^2) + O(m^2) = O(m^2)$. Therefore, the required time for checking the conditions of Theorem 3.6 is as $T^6 = (2n) \times (n-1) \times T^5 = (2n) \times (n-1) \times O(m^2) = O(m^2 n^2)$, because two values \check{x}_k (or \hat{x}_k) can be replaced instead of x_k , for $k = 1, \dots, n$, and $|J \setminus \{k\}| = (n-1)$.

Also, the required time for removing the rows and columns according to Remark 3.2 is as follows. The required time for removing one row or one column is assumed $O(1)$ time. If \check{x}_k (or \hat{x}_k), for $k = 1, 2, \dots, n$, is selected for the k^{th} component of an optimal solution x^* , then row(s) $i \in I_k^-$ (or $i \in I_k^+$) and column k can be deleted from the matrices Q^+ and Q^- . Therefore, the required time is as $T^7 = n \times (2m) \times 2 \times O(1) = O(mn)$, in the worst case, because the number of variables are less or equal to n , $|I_k^-| \leq m$, $|I_k^+| \leq m$, and the row(s) $i \in I_k^-$ (or $i \in I_k^+$) and column k are removed from two matrices Q^+ and Q^- , simultaneously. Hence, the total required time or computational complexity for Substep 6.3 is as $T_{6,3} = T^2 + T^6 + T^7 = O(m) + O(m^2 n^2) + O(mn) = O(m^2 n^2)$.

6.4. The required time for checking the conditions of Corollary 3.5 for a fixed $i \in I$ and a fixed $k \in J_i^+$ is as follows:

Condition 1 needs $T^1 = O(1)$ time. In Condition 2, the required time of creating $\bigcup_{j \in J_i^+ \setminus \{k\}} I_j^+$ and $\bigcup_{j \in J_i^+ \setminus \{k\}} I_j^-$ is as

$2(n-2) \times O(m^2) = O(nm^2)$, in the worst case, because $|I_j^+| \leq m$, $|I_j^-| \leq m$, and $|J_i^+ \setminus \{k\}| \leq n-1$. Each of the operations of the creating $\bigcup_{j \in J_i^+ \setminus \{k\}} I_j^+ \setminus \bigcup_{j \in J_i^+ \setminus \{k\}} I_j^-$ and checking $\bigcup_{j \in J_i^+ \setminus \{k\}} I_j^+ \setminus \bigcup_{j \in J_i^+ \setminus \{k\}} I_j^- \subseteq I_k^+$ needs $O(m^2)$ time

with the similar reasons to Substep 6.3, in the worst case. Therefore, the total required time for checking Condition 2 is as $T^2 = O(nm^2) + O(m^2) + O(m^2) = O(nm^2)$.

In Condition 3, the required times for creating $\bigcup_{j \in J_i^+ \setminus \{k\}} I_j^-$ and $I_k^- \setminus I_k^+$ are $O(nm^2)$ and $O(m^2)$, respectively, in the worst

case. The required time for checking $\bigcup_{j \in J_i^+ \setminus \{k\}} I_j^- \supseteq I_k^- \setminus I_k^+$ is as $O(m^2)$. The reasons of above points are similar to that of

Substep 6.4. Therefore, the total required time for checking Condition 3 is as $T^3 = O(nm^2) + O(m^2) + O(m^2) = O(nm^2)$.

Condition 4 needs at most $(n-1)$ comparison operations as $c_j \hat{x}_j^{r_j} \geq c_k \hat{x}_k^{r_k}$ because $|J_i^+ \setminus \{k\}| \leq n-1$. Therefore, $T^4 = (n-1) \times O(1) = O(n)$, where $O(1)$ is the required time to compare two elements together. With regard to the above points, the total required time for checking the conditions of Corollary 3.5, for each $i \in I$ and $k \in J_i^+$, is as: $T^5 = m \times n \times (T^1 + T^2 + T^3 + T^4) = mn \times (O(1) + O(nm^2) + O(nm^2) + O(n)) = mn \times O(nm^2) = O(n^2 m^3)$. Under the conditions of Corollary 3.5, we put $x_k^* = \hat{x}_k$ and $x_j^* = \check{x}_j$, for each $j \in J_i^+ \setminus \{k\}$, which needs $O(n)$ time. With regard to Substep 6.3, the computational complexities of checking Theorem 3.6 and implementation of Remark 3.2 are $O(n^2 m^2)$ and $O(nm)$, respectively. Hence, the computational complexity of Substep 6.4 is as: $T_{6,4} = T^5 + O(n) + O(n^2 m^2) + O(nm) = O(n^2 m^3)$.

6.5. To check the conditions of Theorem 3.4, we firstly consider the different choices of sets $K_1, K_2 \subset J$ where sets K_1 and K_2 are not equal to empty or J . The number of subsets of set J without considering empty set and J is $2^n - 2$. The number of different cases for choice of sets $K_1, K_2 \subset J$ is as follows. The set K_1 can be selected among $2^n - 2$ subsets of set J . Then, we remove set K_1 and select the set K_2 among $2^n - 3$ remaining subsets. Thus, there exist $(2^n - 2)(2^n - 3)$ different cases to select the sets K_1 and K_2 . We should now check four conditions of Theorem 3.4 for each of the cases.

Using the similar reasons to Substep 6.3, Condition 1 needs $T^1 = O(n^2)$ time because $|K_1| \leq n$ and $|K_2| \leq n$. Also, the total required time for creating sets $\bigcup_{j \in K_1} I_j^+$, $\bigcup_{j \in K_1} I_j^-$, and $\bigcup_{j \in K_2} I_j^+$ is as $3(n-2) \times O(m^2) = O(nm^2)$, in the worst case.

Therefore, the computational complexity for creating and checking Condition 2 is as $T^2 = O(nm^2) + O(m^2) + O(m^2) = O(nm^2)$. With the same reasons, the computational complexity of creating and checking Condition 3 is as $T^3 = O(nm^2)$. Obviously, the computational complexity of checking Condition 4 is as $T^4 = O(n^2)$. Moreover, we put $x_j^* = \check{x}_j$, for each $j \in K_1$. Hence, we need $O(n)$ time for this work. Thus, the computational complexity for checking Theorem 3.4

Archive of SID

is as follows:

$$\begin{aligned} T^5 &= (2^n - 2)(2^n - 3) \times (T^1 + T^2 + T^3 + T^4) + O(n) \\ &= (2^n - 2)(2^n - 3) \times (O(n^2) + O(nm^2) + O(nm^2) + O(n^2)) + O(n) \\ &= O(2^{2n}n^2 + 2^{2n}nm^2). \end{aligned}$$

Considering T^5 and the obtained computational complexities of checking Theorem 3.6 and implementation of Remark 3.2 in Substep 6.3, the computational complexity of Substep 6.5 is as follows:

$$T_{6.5} = T^5 + O(n^2m^2) + O(nm) = O(2^{2n}n^2 + 2^{2n}nm^2),$$

because $2^{2n}nm^2 \geq n^2m^2 \geq nm$ for $\forall n, m \in N = \{1, 2, 3, \dots\}$.

Step 7. The required time for checking $Q^+ = \emptyset$ and $Q^- = \emptyset$ is $2 \times O(1)$. Moreover, we need $O(n)$ time to assign \check{x}_j to x_j^* , in the worst case. Therefore, the computational complexity of this step is as $T_7 = O(n)$.

Step 8. The total required time for this step is as $T_8 = O(mn)$.

Step 9. The total required time for this step is as $T_9 = O((2n)^m)$.

Step 10. This step needs $T_{10} = O(1)$ time.

With regard to the computational costs in each step, the computational complexity of the algorithm is as

$$T = O(2^{2n}n^2 + 2^{2n}nm^2 + n^2m^3 + 2^m n^m).$$

We are now ready to illustrate Algorithm 1 by a numerical example and to express its importance.

6 Examples

In this section, an example is presented to express the importance of the problem (1) in the covering problem. A numerical example is also given to explain its resolution process by Algorithm 1.

Example 6.1. A manager intends to cover six zones by promoting the quality level of his school [1]. To do this, he considers five quality criteria as: 1. cultural activities, 2. laboratories, 3. athletic-recreational facilities, 4. educational, and 5. cleanliness. Some plans are implemented for improvement of each potentially poor criterion. Let a_{ij}^+ denotes the required quality level of criterion (i) for students' parents in zone (j). This matrix for the six zones and the five criteria is as $A^+ = (a_{ij}^+)_{5 \times 6}$. He estimates that if he expends cost x_j (x_j has been normalized in $[0, 1]$) to overcome the requirement from type of (i), then he will obtain the quality level $a_{ij}^+ \cdot x_j$ to satisfy the parents in zone (j) from criterion (i). On the other hand, the manager receives the costs from the parents. If the costs increase, their satisfaction level will decrease to $a_{ij}^- \cdot (1 - x_j)$, where a_{ij}^- denotes the satisfaction level of parents from criterion (i) in zone (j). The satisfaction levels of a_{ij}^- are given in terms of matrix $A^- = (a_{ij}^-)_{5 \times 6}$. The manager estimates levels b_i , for $i = 1, \dots, 5$, such that if he fulfills the satisfaction levels b_i , $i = 1, \dots, 5$, for criterion (i) at least for the parents of one of the zones, then he will overcome the difficulties in the requirements from the kind of (i). The range of costs that he should pay to fulfill the quality levels of $b = (b_1, b_2, b_3, b_4, b_5)^T$ is obtained by resolution of system $A^+ \circ x \vee A^- \circ \neg x = b$ where "o" and "v" denote the max-product and maximum operator, respectively. Moreover, if he expends cost x_j for zones $j = 1, \dots, 6$, to overcome the requirement from the type of (i), $i = 1, \dots, 5$, in zone (j), then the investment risk in zone (j) is obtained by $x_j^{r_j}$ where $r_j \in R^+$. Now, the manager intends to minimize the maximum of risks subject to the system $A^+ \circ x \vee A^- \circ \neg x = b$ and $x \in [0, 1]^6$. This problem can equivalently be modelled as follows:

$$\begin{aligned} \min \quad & Z(x) = \bigvee_{j=1}^6 x_j^{r_j}, \\ \text{s.t.} \quad & A^+ \circ x \vee A^- \circ \neg x = b, \\ & x \in [0, 1]^6. \end{aligned}$$

Now, we will give an example to illustrate Algorithm 1.

Example 6.2. Consider the following problem.

$$\begin{aligned} \text{Minimize} \quad & Z(x) = (6.x_1^4) \vee (7.x_2) \vee (8.x_3^{\frac{3}{2}}) \vee (3.x_4^{\frac{1}{5}}) \vee (6.x_5^4) \\ \text{Subject to} \quad & x \in S(A^+, A^-, b) := \{x \in [0, 1]^5 \mid A^+ \circ x \vee A^- \circ \neg x = b\}, \end{aligned} \tag{6}$$

Archive of SID

Since node 2 has the less cost with regard to other nodes, we branch on it. The second constraint can be satisfied by each of the values $\{\hat{x}_4, \hat{x}_5\}$. Two nodes 3 and 4, respectively, with costs 2.38 and 2.45 are created. The stop criterion is not hold for nodes 1,3, and 4. So, we choose the node with the least cost, i.e., node 3. The third constraint can be hold by each of the values $\{\hat{x}_4, \tilde{x}_5\}$ which only \tilde{x}_5 can be selected in this node. The cost of node 5 is 2.38. The stop criterion is not again hold. Hence, the fourth constraint can be satisfied by each of the values $\{\hat{x}_4, \hat{x}_5\}$ which none of them cannot be chosen. Therefore, the process in node 5 ends. Now, node 4 has the least cost between two nodes 1 and 4. Since we have no candidates to satisfy the third constraint in node 4, the process ends in this node. We continue the process from node 1. Only value \hat{x}_5 can be selected to hold the second constraint in node 1. The cost in node 6 becomes 2.61 and the process ends in this node. Therefore, the optimal solution $x_4^* = \hat{x}_4 = 0.50$, $x_5^* = \hat{x}_5 = 0.80$ happens in this node. The processes have been shown in a tree with 7 nodes as Figure 1.

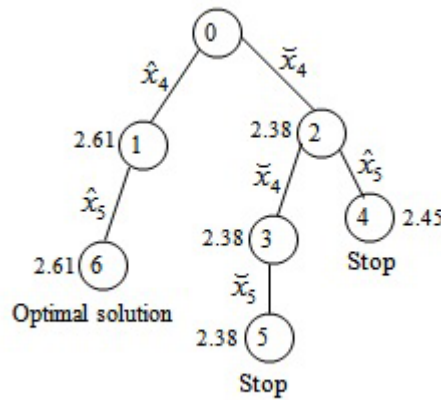


Figure 1: Modified branch-and-bound method.

Step 10. The optimal solution and the optimal objective value of problem (6)-(7) are as follows:
 $x^* = (x_1^*, x_2^*, x_3^*, x_4^*, x_5^*)^T = (0.5, 0.45, 0.7, 0.5, 0.8)^T$ and $Z(x^*) = \sqrt[5]{\prod_{j=1}^5 (c_j \cdot x_j^{r_j})} = 6.12$.

Now, we will measure the rate of computations in two cases (1) applying the simplification procedures and (2) without using them. In the next section, the effects of the simplification procedures are studied and an analytic and comparative study is done to show the efficiency of the proposed algorithm.

7 Analysis and comparison

If we apply Algorithm 1 without using the simplification procedures, i.e., Step 6, then the algorithm will need a computational cost as $T_9 = O((2n)^m)$. If we apply Algorithm 1, then Step 6 consumes a computational cost as $T_6 = O(2^{2n}n^2 + 2^{2n}nm^2 + n^2m^3)$ and simplifies the original problem. This work reduces the dimensions of the original problem. The reduced problem is solved using Step 9. With regard to the high computational cost of Step 9, each reduction on the dimensions of original problem can decrease the computational rate, considerably. For instance, we compare the rate of computations between Algorithm 1 without using the simplification procedures and Algorithm 1 in Example 6.2. The rate of computations of Algorithm 1 without using the simplification procedure, i.e., Step 6, for Example 6.2 is as: $T_9 = O((2n)^m)$ where m and n are the number of rows and columns of matrix A^+ (or A^-), respectively. The matrix A^+ (or A^-) in Example 6.2 is 7×5 . Hence, the values m and n are as $m = 7$ and $n = 5$. Therefore, the computational cost in this case is as $T^{(1)} = T_9 = O((2n)^m) = O((2 \times 5)^7) = 10^7 \times O(1)$. But if we apply Algorithm 1 to solve Example 6.2, then Step 6 will consume the time $T_6 = O(2^{2n}n^2 + 2^{2n}nm^2 + n^2m^3) = O(2^{10} \times 5^2 + 2^{10} \times 5 \times 7^2 + 5^2 \times 7^3) = 285055 \times O(1)$, to simplify the original problem. Using Step 6, the dimensions of simplified problem are reduced to $m = 4$ and $n = 2$. Hence, Step 9 spends $T_9 = O((2n)^m) = O((2 \times 2)^4) = O(256) = 256 \times O(1)$ time to solve the simplified problem. Overall, the Algorithm 1 needs to $T^{(2)} = T_6 + T_9 = (285055 + 256) \times O(1) = 285311 \times O(1)$ time.

If we compare the rate of computations in two cases, the proposed algorithm solves Example 6.2 in the time $T^{(2)} = 285311 \times O(1)$ and the algorithm without using Step 6 solves Example 6.2 in time $T^{(1)} = 10^7 \times O(1)$. It shows that the proposed algorithm solves Example 6.2, 35 times faster than the algorithm without Step 6. In fact,

the simplification procedures or Step 6 can accelerate the algorithm in [2], considerably. It shows the efficiency of our algorithm, specially, the efficiency of the simplification procedures in reduction of the computations.

8 Conclusions

The geometric optimization with single-term exponent subject to bipolar max-product *FRE* constraints was studied in this paper. The structure of its feasible domain was investigated. Some sufficient conditions were proposed to determine some the optimal components without solving the optimization problem by the branch-and-bound method. These conditions can reduce the size of the problem, considerably. The modified branch-and-bound method is then applied to solve the reduced problem. The computation complexity of the algorithm was discussed and the analytic and comparative study showed the efficiency of the proposed algorithm. Its importance was illustrated by the covering problem.

Acknowledgement

We are very grateful to the editor and anonymous reviewers for their valuable comments and suggestions for the improvement of the manuscript.

References

- [1] A. Abbasi Molai, E. Khorram, *An algorithm for solving fuzzy relation equations with max-T composition operator*, Information Sciences, **178** (2008), 1293-1308.
- [2] S. Aliannezhadi, A. Abbasi Molai, *Geometric programming with a single-term exponent subject to bipolar max-product fuzzy relation equation constraints*, Fuzzy Sets and Systems, **397** (2020), 61-83.
- [3] S. Aliannezhadi, A. Abbasi Molai, B. Hedayatfar, *Linear optimization with bipolar max-parametric hamacher fuzzy relation equation constraints*, Kybernetika, **52**(4) (2016), 531-557.
- [4] C. W. Chang, B. S. Shieh, *Linear optimization problem constrained by fuzzy max-min relation equations*, Information Sciences, **234** (2013), 71-79.
- [5] M. E. Cornejo, D. Lobo, J. Medina, *On the solvability of bipolar max-product fuzzy relation equations with the product negation*, Journal of Computational and Applied Mathematics, **354** (2019), 520-532.
- [6] M. E. Cornejo, D. Lobo, J. Medina, *On the solvability of bipolar max-product fuzzy relation equations with the standard negation*, Fuzzy Sets and Systems, to appear, (2020). Doi: 10.1016/j.fss.2020.02.010.
- [7] S. C. Fang, G. Li, *Solving fuzzy relation equations with a linear objective function*, Fuzzy Sets and Systems, **103** (1999), 107-113.
- [8] S. Freson, B. De Baets, H. De Meyer, *Linear optimization with bipolar max-min constraints*, Information Sciences, **234** (2013), 3-15.
- [9] S. M. Guu, Y. K. Wu, *Minimizing a linear objective function under a max-t-norm fuzzy relational equation constraint*, Fuzzy Sets and Systems, **161** (2010), 285-297.
- [10] P. Li, Q. Jin, *Fuzzy relational equations with min-biimplication composition*, Fuzzy Optimization and Decision Making, **11** (2012), 227-240.
- [11] P. Li, Y. Liu, *Linear optimization with bipolar fuzzy relational equation constraints using the Lukasiewicz triangular norm*, Soft Computing, **18** (2014), 1399-1404.
- [12] G. Li, J. Qiu, G. Xiao, H. Qu, X. P. Yang, *Optimal strong solution of the minimax problem with two-sided fuzzy relation inequality constraints*, IEEE Access, **7** (2019), 177571-177584.
- [13] C. C. Liu, Y. Y. Lur, Y. K. Wu, *Linear optimization of bipolar fuzzy relational equations with max-Lukasiewicz composition*, Information Sciences, **360** (2016), 149-162.

- [14] J. Qiu, H. Xue, G. Li, X. P. Yang, *Fuzzy relation bilevel optimization model in the wireless communication station system*, IEEE Access, **8** (2020), 60811-60823.
- [15] E. Sanchez, *Resolution of composite fuzzy relation equations*, Information and Control, **30** (1976), 38-48.
- [16] E. Shivanian, E. Khorram, *Monomial geometric programming with fuzzy relation inequality constraints with max-product composition*, Computers and Industrial Engineering, **56** (2009), 1386-1392.
- [17] F. Sun, X. B. Qu, X. P. Wang, L. Zhu, *On pre-solution matrices of fuzzy relation equations over complete Brouwerian lattices*, Fuzzy Sets and Systems, **384** (2020), 34-53.
- [18] E. Turunen, *Necessary and sufficient conditions for the existence of solution of generalized fuzzy relation equations $A \Leftrightarrow X = B$* , Information Sciences, **536** (2020), 351-357.
- [19] P. Z. Wang, D. Z. Zhang, E. Sanchez, E. S. Lee, *Latticized linear programming and fuzzy relation inequalities*, Journal of Mathematical Analysis and Applications, **159** (1991), 72-87.
- [20] Y. K. Wu, *Optimizing the geometric programming problem with single-term exponents subject to max-min fuzzy relational equation constraints*, Mathematical and Computer Modelling, **47** (2008), 352-362.
- [21] Y. K. Wu, S. M. Guu, *Minimizing a linear function under a fuzzy max-min relational equation constraint*, Fuzzy Sets and Systems, **150** (2005), 147-162.
- [22] Y. K. Wu, S. M. Guu, *An efficient procedure for solving a fuzzy relational equation with max archimedean t-norm composition*, IEEE Transactions on Fuzzy Systems, **16** (2008), 73-84.
- [23] Y. K. Wu, S. M. Guu, J. Y. C. Liu, *An accelerated approach for solving fuzzy relation equations with a linear objective function*, IEEE Transactions on Fuzzy Systems, **10**(4) (2002), 552-558.
- [24] G. Xiao, T. Zhu, Y. Chen, X. P. Yang, *Linear searching method for solving approximate solution to system of max-min fuzzy relation equations with application in the instructional information resources allocation*, IEEE Access, **7** (2019), 65019-65028.
- [25] X. P. Yang, *Linear programming method for solving semi-latticized fuzzy relation geometric programming with max-min composition*, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, **23**(5) (2015), 781-804.
- [26] X. P. Yang, *Optimal-vector-based algorithm for solving min-max programming subject to addition-min fuzzy relation inequality*, IEEE Transactions on Fuzzy Systems, **25**(5) (2017), 1127-1140.
- [27] X. P. Yang, *Leximax minimum solution of addition-min fuzzy relation inequalities*, Information Sciences, **524** (2020), 184-198.
- [28] X. P. Yang, *Solutions and strong solutions of min-product fuzzy relation inequalities with application in supply chain*, Fuzzy Sets and Systems, **384** (2020), 54-74.
- [29] X. P. Yang, *Resolution of bipolar fuzzy relation equations with max-Lukasiewicz composition*, Fuzzy Sets and Systems, **397** (2020), 41-60.
- [30] J. Yang, B. Cao, *Monomial geometric programming with fuzzy relation equation constraints*, Fuzzy Optimization and Decision Making, **6** (2007), 337-349.
- [31] X. P. Yang, H. T. Lin, X. G. Zhou, B. Y. Cao, *Addition-min fuzzy relation inequalities with application in BitTorrent-like Peer-to-Peer file sharing system*, Fuzzy Sets and Systems, **343** (2018), 126-140.
- [32] X. B. Yang, X. P. Yang, K. Hayat, *A new characterisation of the minimal solution set to max-min fuzzy relation inequalities*, Fuzzy Information and Engineering, **9** (2017), 423-435.
- [33] X. P. Yang, D. H. Yuan, B. Y. Cao, *Lexicographic optimal solution of the multi-objective programming problem subject to max-product fuzzy relation inequalities*, Fuzzy Sets and Systems, **341** (2018), 92-112.
- [34] X. P. Yang, G. Z. Zheng, X. G. Zhou, B. Y. Cao, *Lexicography minimum solution of fuzzy relation inequalities: applied to optimal control in P2P file sharing system*, International Journal of Machine Learning and Cybernetics, **8**(5) (2017), 1555-1563.

- [35] X. P. Yang, X. G. Zhou, B. Y. Cao, *Single-variable term semi-latticized fuzzy relation geometric programming with max-product operator*, Information Sciences, **325** (2015), 271-287.
- [36] X. P. Yang, X. G. Zhou, B. Y. Cao, *Latticized linear programming subject to max-product fuzzy relation inequalities with application in wireless communication*, Information Sciences, **358-359** (2016), 44-55.
- [37] X. P. Yang, X. G. Zhou, B. Y. Cao, Y. H. Hong, *Variable substitution method for solving single-variable term fuzzy relation geometric programming problem and its application*, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, **27**(4) (2019), 537-557.
- [38] C. Zener, *Engineering design by geometric programming*, John Wiley, New York, 1971.
- [39] Y. B. Zhong, G. Xiao, X. P. Yang, *Fuzzy relation lexicographic programming for modelling P2P file sharing system*, Soft Computing, **23** (2019), 3605-3614.
- [40] X. G. Zhou, R. Ahat, *Geometric programming problem with single-term exponents subject to max-product fuzzy relational equations*, Mathematical and Computer Modelling, **53** (2011), 55-62.
- [41] X. G. Zhou, B. Y. Cao, X. P. Yang, *The set of optimal solutions of geometric programming problem with max-product fuzzy relational equations constraints*, International Journal of Fuzzy Systems, **18** (2016), 436-447.
- [42] J. Zhou, Y. Yu, Y. Liu, Y. Zhang, *Solving nonlinear optimization problems with bipolar fuzzy relational equation constraints*, Journal of Inequalities and Applications, **126** (2016), 1-10. Doi: 10.1186/s13660-016-1056-6.

A new algorithm for geometric optimization with a single-term exponent constrained by bipolar fuzzy relation equations

S. Aliannezhadi, and A. Abbasi Molai

یک الگوریتم جدید برای بهینه‌سازی هندسی با توان تک جمله‌ای با محدودیت‌های معادلات رابطه فازی دو قطبی

چکیده. در این مقاله، یک مساله برنامه‌ریزی هندسی با محدودیت‌های معادله رابطه فازی دو قطبی با عملگر ماکزیمم ضرب مورد مطالعه قرار می‌گیرد. تعدادی شرط لازم و کافی برای وجود جواب آن ارائه می‌شود. یک کران پایین و بالا روی مجموعه جواب دامنه شدنی آن بدست آورده می‌شود. تعدادی شرط کافی برای تعیین چند مؤلفه بهینه آن بدون حل مساله پیشنهاد می‌گردد. یک روش شاخه و کران اصلاح شده برای حل مساله گسترش داده می‌شود. بعلاوه، الگوریتم کارایی برای حل مساله براساس عملیات ساده‌سازی و روش شاخه و کران اصلاح شده پیشنهاد می‌شود. پیچیدگی محاسباتی آن به دقت مورد تجزیه و تحلیل قرار می‌گیرد. چندین مثال برای نشان دادن اهمیت مساله و توضیح روند الگوریتم ارائه می‌شود. در پایان، یک بررسی تحلیلی و مقایسه‌ای برای اثبات کارایی رویه‌های ساده‌سازی انجام می‌شود.