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

S. Aliannezhadi\textsuperscript{1} and A. Abbasi Molai\textsuperscript{2}

\textsuperscript{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 of 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) \cite{15, 18}, fuzzy relation inequalities (FRI) \cite{10, 12, 21, 26}, and fuzzy relation programming \cite{30, 31, 32, 37} have extensively been studied by many researchers since fuzzy relation equations were suggested by Sanchez \cite{15} for the first time. The max-min fuzzy relation programming with a linear function was firstly studied by Fang and Li \cite{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 \cite{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 \cite{62}, peer-to-peer network system \cite{61, 67}, and economic analysis, environmental engineering, and business administration \cite{58}. Motivated by the applications, the researchers in \cite{16, 17, 19, 20, 25, 28, 35, 36, 40, 42} studied the latticized fuzzy relation programming and fuzzy relation geometric programming problem with single-term exponents. The FREs and FRI 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 was firstly considered by Freson et al. \cite{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 \cite{10}. A path-based algorithm was designed to find the complete solution set of the bipolar max-Lukasiewicz FRE system by Yang \cite{27}. The bipolar max-product FRE systems have been studied by the product negation in \cite{6} and the standard negation in \cite{5}. Li and Liu \cite{11} also studied the linear optimization problem provided to the bipolar max-\textit{T}_\textit{L} FREs and converted it into a 0-1 integer optimization. Some procedures and rules were designed to reduce its structure in \cite{5, 6}. The nonlinear optimization problem subject to bipolar FRE constraints has firstly been considered by Zhou et al. \cite{41}. 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, 23, 30, 32], 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 to the following questions:

1. If the sufficient conditions aren’t 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 [23, 21, 11] 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 its 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:

\[
\text{Minimize } Z(x) = \bigvee_{j=1}^{n} (c_j x_j^r) \\
\text{Subject to } x \in S(A^+, A^{-}, b) := \{x \in [0, 1]^n| A^+ \circ x \lor A^- \circ \neg x = b\},
\]

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

\[
\max_{j \in J} \max \left\{a_{ij}^+, x_j, a_{ij}^-, (1-x_j)\right\} = b_i, \quad \forall i \in I.
\]

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 \(\hat{x} = (\hat{x}_1, \ldots, \hat{x}_n)^T\) is the lower bound on the solution set of system (2) where \(\hat{x}_j = \max_{i \in I} \left\{1 - \frac{b_i}{a_{ij}^+} | a_{ij}^- > b_i\right\}\), for each \(j \in J\). Also, the vector of \(\hat{\bar{x}} = (\bar{x}_1, \ldots, \bar{x}_n)^T\) is the upper bound on the solution set of system (2) where \(\bar{x}_j = \min_{i \in I} \left\{\frac{b_i}{a_{ij}^+} | a_{ij}^- > b_i\right\}\), for each \(j \in J\), where max \(\emptyset = 0\) and min \(\emptyset = 1\).
Remark 2.2. If \( S(A^+, A^-, b) \neq \emptyset \), then \( \hat{x} \leq \check{x} \). Also, if \( x \in S(A^+, A^-, b) \), then \( \hat{x} \leq x \leq \check{x} \), but its converse is not true.

Proposition 2.3. (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 \( \mathcal{I} = \{ i \in I \mid x_{i} = \check{x}_{i} \} \) and assume that \( \mathcal{I} \neq \emptyset \). Then we can remove the constraints \( i, i \in \mathcal{I} \), 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) The characteristic matrix \( Q = (q_{ij})_{m \times n} \) is defined as follows:

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

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

\[
q_{ij}^+ = \begin{cases} 
1, & \text{if } \check{x}_j \in \hat{q}_{ij}, \\
0, & \text{otherwise,}
\end{cases}
\quad
q_{ij}^- = \begin{cases} 
1, & \text{if } \check{x}_j \in \hat{q}_{ij}, \\
0, & \text{otherwise.}
\end{cases}
\]

for each \( i \in I \) and \( j \in J \).

(3) Let \( J^+_i(x) = \{ j \in J \mid x_j = \check{x}_j \} \), \( J^+_{ij}(x) = \{ j \in J \mid x_j = \check{x}_j \} \) and \( q_{ij}^+ = 1 \}, \), \( I^-_i(x) = \{ i \in I \mid x_i = \hat{x}_j \} \) and \( q_{ij}^- = 1 \}, \), and \( J^-_{ij}(x) = \{ j \in J \mid x_j = \hat{x}_j \} \) 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. A vector \( x \in [0,1]^n \) is a solution for the system of bipolar max-product FRE (2) if and only if \( \hat{x} \leq x \leq \check{x} \) and the induced binary matrix \( Q^x = (q_{ij}^x)_{m \times n} \) doesn’t have any zero rows where

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

Notation 2.6. (i) Let \( I^+_i = I^+_i(\check{x}) \), \( J^+_i = J^+_i(\check{x}) \), \( I^-_i = I^-_i(\check{x}) \), 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 2.4 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. 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 2.4, for each \( j \in J \), the value \( \check{x}_j \) is labeled with the positive literal \( y_j \) and the value \( \hat{x}_j \) is labeled with the negative literal \( \neg y_j \), respectively.

Theorem 2.8. Let \( \hat{x} \) and \( \check{x} \) 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 \( x_j = \check{x}_j \)) or \( I_j(x) = I^-_j \) (when \( x_j = \hat{x}_j \)) or \( I_j(x) = \emptyset \) (when \( \check{x}_j < x_j < \hat{x}_j \)).
3 Some sufficient conditions for simplification of problem (II)

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

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

It is easily seen that one of the minimal solutions is an optimal solution of problem (II) with regard to Ref. [II]. In the problem (II), 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, 31] cannot be applied to solve problem (II).

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 \( x \) is not difficult. If \( \bigcup_{j \in J} I_j^- = I \), then \( x = \hat{x} \) is an optimal solution for problem (II).

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

**Lemma 3.1.** Assume that \( S(A^+, A^-, b) \neq \emptyset \). Then there exists an optimal solution \( x^* = (x_1^*, \ldots, x_n^*)^T \) for problem (II) 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 (II). 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 (II). 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 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 (II), we have to set \( x_k^* = \hat{x}_k \) (or \( x_k^* = \tilde{x}_k \)) with regard to Theorem 4.3 and part (ii) of Notation 4.6.

**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. \( \bigcup_{j \in K_1} I_j^- \cup \bigcup_{j \in K_2} I_j^+ \subseteq \bigcup_{j \in J} I_j^+ \),
3. \( \bigcup_{j \in K_1} I_j^+ \subseteq \bigcup_{j \in K_2} I_j^- \cup \bigcup_{j \in J} I_j^+ \),
4. \( c_j, \tilde{x}_j \geq \bigvee_{j \in K_2} (c_j, \hat{x}_j^*) \), \( \forall j \in K_1 \),

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

**Proof.** Suppose that \( x^{**} \) is an optimal solution for problem (II) and there exists an index set \( \emptyset \neq E_1 \subseteq K_1 \) such that \( \tilde{x}_j < x_j^{**} < \hat{x}_j \), for each \( j \in E_1 \). Without loss of generality, assume that \( x_j^{**} = \hat{x}_j \), 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 vector \( x^* \) is a feasible solution for system (II) and \( Z(x^*) \leq Z(x^{**}) \) with regard to Theorems 4.3 and 4.6 and part (ii) of Notation 4.6. Otherwise, we have \( x_j^* = \tilde{x}_j \), for each \( j \in E_1 \) 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 \( x_j^* = \hat{x}_j \), for each \( j \in E_1 \), \( x_j^* = \tilde{x}_j \), 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 \( \hat{x} \leq x^* < \tilde{x} \). It is easily seen that \( \bigcup_{j \in F_1} I_j^+ \subseteq \bigcup_{j \in K_1} I_j^+ \subseteq \bigcup_{j \in K_2} I_j^+ \) and \( \bigcup_{j \in E_2} I_j^- \subseteq \bigcup_{j \in K_2} I_j^- \subseteq \bigcup_{j \in K_1} I_j^- \).
with regard to conditions 2 and 3. So, we have

\[
\bigcup_{j \in J} I_j(x^{**}) = \bigg( \bigcup_{j \in J \setminus (K_1 \cup K_2)} I_j(x^{**}) \bigg) \cup \bigg( \bigcup_{j \in F_1} I_j^+ \bigg) \cup \bigg( \bigcup_{j \in K_1 \setminus E_1} I_j^- \bigg) \cup \bigg( \bigcup_{j \in K_2 \setminus E_2} I_j^- \bigg) \cup \bigg( \bigcup_{j \in J} I_j^{**} \bigg)
\]

\[
\subseteq \bigg( \bigcup_{j \in J \setminus (K_1 \cup K_2)} I_j(x^{**}) \bigg) \cup \bigg( \bigcup_{j \in F_1} I_j^+ \bigg) \cup \bigg( \bigcup_{j \in K_1 \setminus E_1} I_j^- \bigg) \cup \bigg( \bigcup_{j \in K_2 \setminus E_2} I_j^- \bigg) \cup \bigg( \bigcup_{j \in J} I_j^{**} \bigg)
\]

\[
\subseteq \bigg( \bigcup_{j \in J \setminus (K_1 \cup K_2)} I_j(x^{**}) \bigg) \cup \bigg( \bigcup_{j \in K_2} I_j^+ \bigg) \cup \bigg( \bigcup_{j \in K_1 \setminus E_1} I_j^- \bigg) \cup \bigg( \bigcup_{j \in K_2 \setminus E_2} I_j^- \bigg) \cup \bigg( \bigcup_{j \in J} I_j^{**} \bigg)
\]

\[
= \bigg( \bigcup_{j \in K_2} I_j(x^{**}) \bigg) \cup \bigg( \bigcup_{j \in K_2} I_j^+ \bigg) \cup \bigg( \bigcup_{j \in K_1 \setminus E_1} I_j^- \bigg) \cup \bigg( \bigcup_{j \in K_2 \setminus E_2} I_j^- \bigg) = \bigcup_{j \in J} I_j^+(x^*)
\]

Since \( \hat{x} \leq x^* \leq \hat{x} \) and \( \bigcup_{j \in J} I_j(x^*) \supseteq \bigcup_{j \in J} I_j(x^{**}) \), vector \( x^* \) is a feasible solution for system (4) with regard to Theorem 4.7 and part (ii) of Notation 4.8. Furthermore, we have:

\[
Z(x^*) = \bigg( \bigvee_{j \in J \setminus (E_1 \cup E_2)} (c_j, x_j^*) \bigg) \vee \bigg( \bigvee_{j \in F_1} (c_j, x_j^*) \bigg) \vee \bigg( \bigvee_{j \in K_2} (c_j, x_j^*) \bigg)
\]

\[
\subseteq \bigg( \bigvee_{j \in J \setminus (E_1 \cup E_2)} (c_j, x_j^*) \bigg) \vee \bigg( \bigvee_{j \in F_1} (c_j, x_j^*) \bigg) \vee \bigg( \bigvee_{j \in K_1 \setminus E_1} (c_j, x_j^*) \bigg) \vee \bigg( \bigvee_{j \in K_2 \setminus E_2} (c_j, x_j^*) \bigg)
\]

\[
\subseteq \bigg( \bigvee_{j \in J \setminus (E_1 \cup E_2)} (c_j, x_j^*) \bigg) \vee \bigg( \bigvee_{j \in F_1} (c_j, x_j^*) \bigg) \vee \bigg( \bigvee_{j \in F_1} (c_j, x_j^*) \bigg) \vee \bigg( \bigvee_{j \in K_1 \setminus E_1} (c_j, x_j^*) \bigg) \vee \bigg( \bigvee_{j \in K_2 \setminus E_2} (c_j, x_j^*) \bigg) = Z(x^{**})
\]

The first inequality is derived from \( \bigvee_{j \in F_1} (c_j, x_j^*) \leq \bigvee_{j \in J \setminus (E_1 \cup E_2)} (c_j, x_j^*) \bigg) \vee \bigg( \bigvee_{j \in F_1} (c_j, x_j^*) \bigg) \vee \bigg( \bigvee_{j \in K_1 \setminus E_1} (c_j, x_j^*) \bigg) \vee \bigg( \bigvee_{j \in K_2 \setminus E_2} (c_j, x_j^*) \bigg) \), for each \( j \in F_1 \) and \( c_j, x_j^* \geq \bigvee_{j \in K_1 \setminus E_1} (c_j, x_j^*) \bigg) \geq \bigvee_{j \in K_2 \setminus E_2} (c_j, x_j^*) \), for each \( j \in K_1 \). Therefore, we have \( Z(x^*) \leq Z(x^{**}) \). Hence, such an optimal solution \( x^* \) must exist with \( x_j^* = \hat{x}_j \), 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 \( x_j^* = \hat{x}_j \), for each \( j \in E_1 \) and \( x_j^* = x_j^* \), for each \( j \in J \setminus E_1 \). We have:

\[
\bigcup_{j \in J} I_j(x^{**}) = \bigg( \bigcup_{j \in J \setminus (K_1 \cup K_2)} I_j(x^{**}) \bigg) \cup \bigg( \bigcup_{j \in F_1} I_j^+ \bigg) \cup \bigg( \bigcup_{j \in K_1 \setminus E_1} I_j^- \bigg) \cup \bigg( \bigcup_{j \in K_2 \setminus E_2} I_j^- \bigg) \cup \bigg( \bigcup_{j \in J} I_j^{**} \bigg)
\]

\[
\subseteq \bigg( \bigcup_{j \in J \setminus (K_1 \cup K_2)} I_j(x^{**}) \bigg) \cup \bigg( \bigcup_{j \in F_1} I_j^+ \bigg) \cup \bigg( \bigcup_{j \in K_1 \setminus E_1} I_j^- \bigg) \cup \bigg( \bigcup_{j \in K_2 \setminus E_2} I_j^- \bigg) \cup \bigg( \bigcup_{j \in J} I_j^{**} \bigg)
\]

\[
= \bigg( \bigcup_{j \in J \setminus (K_1 \cup K_2)} I_j(x^{**}) \bigg) \cup \bigg( \bigcup_{j \in K_2} I_j^+ \bigg) \cup \bigg( \bigcup_{j \in K_1 \setminus E_1} I_j^- \bigg) \cup \bigg( \bigcup_{j \in K_2 \setminus E_2} I_j^- \bigg) = \bigcup_{j \in J} I_j(x^*)
\]

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

\[
Z(x^*) = \bigg( \bigvee_{j \in J \setminus E_1} (c_j, x_j^*) \bigg) \vee \bigg( \bigvee_{j \in E_1} (c_j, x_j^*) \bigg) \leq \bigg( \bigvee_{j \in J \setminus E_1} (c_j, x_j^*) \bigg) \vee \bigg( \bigvee_{j \in E_1} (c_j, x_j^{**}) \bigg) = Z(x^{**})
\]

So, such an optimal solution \( x^* \) must also exist with \( x_j^* = \hat{x}_j \), for each \( j \in K_1 \), in this case.
A direct result of Theorems 3.3 and 3.4 and Remark 3.5 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. \( c_j \hat{x}_j \geq c_k \hat{x}_k', \forall j \in J_i^+ \setminus \{k\} \),

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

Theorem 3.6. If the value \( \hat{x}_k \) (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 \( J_i^+ \setminus I_h^+ \subseteq I_h^- \) (or \( I_i^+ \)), then there exists an optimal solution \( x^* = (x_j^*)_{j \in J} \) such that \( x_h^* = \hat{x}_h \).

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

\[
\bigcup_{j \in J} I_j(x^{**}) = \left( \bigcup_{j \in J \setminus \{h,k\}} I_j(x^{**}) \right) \cup (I_h(x^{**})) \cup (I_h^-) \subseteq \left( \bigcup_{j \in J \setminus \{h,k\}} I_j(x^{**}) \right) \cup (I_h^+) \cup (I_h^-) = \bigcup_{j \in J} I_j(x^*).
\]

On the other hand, \( \hat{x} \leq x^* \leq \hat{x} \). So, vector \( x^* \) is a feasible solution for system (I) with regard to Theorem 3.3 and part (ii) of Notation 3.4. Moreover, we can write:

\[
Z(x^*) = \left( \bigvee_{j \in J \setminus \{h\}} (c_j x_j^{**}) \right) \vee (c_h \hat{x}_h') \leq \left( \bigvee_{j \in J \setminus \{h\}} (c_j x_j^{**}) \right) \vee (c_h x_h^{**}) = Z(x^{**}).
\]

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 \( x_h^* = \hat{x}_h \). \( \square \)

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

4. An algorithm for resolution of problem (II)

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

Considering Theorem 3.8 and Lemma 3.9, there exists an optimal solution \( x^* \) for the optimization problem (II) 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 \( M = (m_{ij})_{|I_1| \times |2| \times |J|} \) as follows:

\[
m_{4,2j-1} = \begin{cases} c_j \hat{x}_j', & \text{if } q_{ij} = 1, \\ \infty, & \text{otherwise}, \end{cases} \quad \text{and} \quad m_{4,2j} = \begin{cases} c_j \hat{x}_j', & \text{if } q_{ij} = 1, \\ \infty, & \text{otherwise}, \end{cases}
\]
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 (II) with the value matrix $M$ according to [2]. We now propose an algorithm to find an optimal solution of the optimization problem (II).

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

**Step 1.** Calculate the lower and upper bound $\bar{x}$ and $\hat{x}$ with regard to Lemma 2.5.
**Step 2.** If $b_i > 0$, for each $i \in I$ and $\bar{x}_j < \hat{x}_j$, 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 2.3.
**Step 4.** Find the index sets $I^+_i$, $I^-_i$, $J^+_i$, and $J^-_i$ with regard to part (3) of Definition 2.3 and part (i) of Notation 2.4.
**Step 5.** Check the consistency of bipolar system (II) using Theorem 3.6 and part (i) of Notation 2.4. If it is inconsistent, then stop! Otherwise, go to Step 6.
**Step 6.** Implement the following substeps.

6.1. If $\bigcup_{j \in J} J^-_j = I$, then the vector $x^* = \hat{x}$ 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 (II) (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 4.6. Go to Step 10.

6.3. If there exists some $i \in I$ such that $|J^+_i| + |J^-_i| = 1$, i.e., $J^+_i = \{i\} \text{ and } J^-_i = \emptyset$ (or $J^-_i = \{i\}$ and $J^+_i = \emptyset$), then for any optimal solution $x^* = (x^*_j)_{j \in J}$, we have $x^*_j = \hat{x}_j$ (or $x^*_j = \bar{x}_j$) with regard to Theorem 5.7. If the conditions of Theorem 5.7 are satisfied, then under the condition, let $x^*_h = \hat{x}_h$. Moreover, remove their corresponding row(s) and column(s) from the matrices $Q^+$ and $Q^-$ according to Remark 5.8.

6.4. If the conditions of Corollary 5.8 holds, then some components of an optimal solution can be determined by Corollary 5.8. Also, check the conditions of Theorem 5.9. Then remove row(s) and column(s) of matrices $Q^+$ and $Q^-$ applying Remark 5.8.

6.5. Check the conditions of Theorem 5.9. If the conditions are satisfied, then some of the optimal variables of problem (II) can be determined. If the conditions of Theorem 5.9 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 5.8.

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 $\hat{x}_j$ to $x^*_j$ and go to Step 10.
**Step 8.** Determine the value matrix $M$ using relation (II).
**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 (II) 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 (II), 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 $T_5 = O(mn^2n)$.
**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: $T_{6,1} = O(mn^2)$.
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 $T^i = m \times O(1) = O(m)$.

The required time for setting $x^*_k = \tilde{x}_k$ for each $i \in I$, is $T^i = m \times O(1) = O(m)$. On the other hand, the required time for setting $x^*_k = \tilde{x}_k$ for each $i \in I$, is $O(1)$. Hence, the computational complexity for checking the conditions of Theorem 5.6 is as $T^2 = T^1 + O(1) = O(m) + O(1) = O(m)$.

For the required time for checking the condition $I^+_k \setminus I^+_h \subseteq I^+_h$ (or $I^+_h \setminus I^+_k$), for one fixed $k$ and $h \in J \setminus \{k\}$ in Theorem 5.6, is as follows. For finding the elements of set $I^+_h \setminus I^+_k$, we need to compare each element of set $I^+_h$ with all of the elements of set $I^+_k$. 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 $I^+_k \setminus I^+_h \subseteq I^+_h$ or $I^+_h \setminus I^+_k \subseteq I^+_k$ needs to compare each element of set $I^+_h \setminus I^+_k$ with $I^+_k$ (or $I^+_h$). Hence, it needs $T^4 = 2O(m^2) = O(m^2)$ time, in the worst case, due to $|I^+_k \setminus I^+_h| \leq m$, $|I^+_h| \leq m$, and $|I^+_k| \leq m$. In the worst case, the computational complexity for computing $I^+_k \setminus I^+_h$ and checking $I^+_h \setminus I^+_k \subseteq I^+_k$ (or $I^+_h$) is as $T^5 = T^3 + T^4 = O(m^2) + O(m^2) = O(m^4)$. Therefore, the required time for checking the conditions of Theorem 5.6 is as $T^6 = (2n) \times (n - 1) \times T^3 = (2n) \times (n - 1) \times O(m^2) = O(m^2 n^2)$, because two values $\tilde{x}_k$ (or $\tilde{x}_h$) can be replaced instead of $x_k$, for $k = 1, \ldots, n$, and $|J \setminus \{k\}| = (n - 1)$.

Also, the required time for removing the rows and columns according to Remark 5.6 as follows. The required time for removing one row or one column is assumed $O(1)$ time. If $\tilde{x}_k$ (or $\tilde{x}_h$), for $k = 1, 2, \ldots, n$, is selected for the $k$th component of an optimal solution $x^*$, then row(s) $i \in I^+_h$ (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 is less or equal to $n$, $|I^+_k| \leq m$, $|I^+_h| \leq m$, and the row(s) $i \in I^+_h$ (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 $T^6 = T^2 + T^6 + T^7 = O(m) + O(m^2 n^2) + O(mn) = O(m^2 n^2)$.

6.4. The required time for checking the conditions of Corollary 5.6 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 $2(n - 2) \times O(m^2) = O(mn^2)$, 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 $\bigcup_{j \in J^+_i \setminus \{k\}} I^+_j$ and $\bigcup_{j \in J^+_i \setminus \{k\}} I^-_j$ 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 $T^2 = O(m^2) + O(m^2) + O(m^2) = O(mn^2)$.

- In Condition 3, the required time for creating $\bigcup_{j \in J^+_i \setminus \{k\}} I^+_j$ and $\bigcup_{j \in J^+_i \setminus \{k\}} I^-_j$ are $O(mn^2)$ and $O(m^2)$, respectively, in the worst case. The required time for checking $\bigcup_{j \in J^+_i \setminus \{k\}} I^+_j \geq I^-_j \setminus I^+_k$ 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 $T^3 = O(mn^2) + O(m^2) + O(m^2) = O(mn^2)$. Condition 4 needs at most $(n - 1)$ comparison operations as $c_j \tilde{x}^*_j \geq c_k \tilde{x}^*_k$ 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 5.6, for each $i \in I$ and $k \in J^+_i$, is as: $T^5 = n \times n \times (T^1 + T^2 + T^3 + T^4) = mn \times (O(1) + O(mn^2) + O(mn^2) + O(n)) = mn \times O(mn^2) = O(mn^3)$. Under the conditions of Corollary 5.6, we put $x^*_k = \tilde{x}_k$ and $x^*_j = \tilde{x}_j$, 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 5.6 and implementation of Remark 5.6 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 5.6, 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 5.6 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 $3(n - 2) \times O(m^2) = O(mn^2)$, in the worst case. Therefore, the computational complexity for creating and checking Condition 2 is as $T^2 = O(mn^2) + O(m^2) + O(m^2) = O(mn^2)$. With the same reasons, the computational complexity of creating and checking Condition 3 is as $T^3 = O(mn^2)$. Obviously, the computational complexity of checking Condition 4 is as $T^4 = O(n^2)$. Moreover, we put $x^*_j = \tilde{x}_j$, for each $j \in K_1$. Hence, we need $O(n)$ time for this work. Thus, the computational complexity for checking Theorem 5.6
is as follows:

\[ T^5 = (2^n - 2)(2^n - 3) \times (T^1 + T^2 + T^3 + T^4) + O(n) \]
\[ = (2^n - 2)(2^n - 3) \times (O(n^2) + O(nm^2) + O(n^m) + O(n^2)) + O(n) \]
\[ = O(2^{2n}n^2 + 2^{2n}m^2). \]

Considering \( T^5 \) and the obtained computational complexities of checking Theorem \( 20 \) and implementation of Remark \( 22 \) in Substep 6.3, the computational complexity of Substep 6.5 is as follows:

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

because \( 2^{2n}nm^2 \geq n^2m^2 \geq nm \) for \( \forall n, m \in N = \{1, 2, 3, \ldots\}. \)

**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 \( \hat{x}_j \) 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 \( T_8 = O(nnm) \).

**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

\[ T = O(2^{2n}n^2 + 2^{2n}nm^2 + n^2m^3 + 2^mnm^m). \]

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 (8) 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 (9). 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 \( a_{ij}^+ \) denote 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 \( A^+ = (a_{ij}^+)_{6 \times 5} \). 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 \( a_{ij}^+ x^*_j \) 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 \( a_{ij}^+ (1 - x^*_j) \), 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 \( A^− = (a_{ij}^−)_{5 \times 5} \). The manager estimates levels \( b_i \), for \( i = 1, \ldots, 5 \), such that if he fulfills the satisfaction levels \( b_i \), \( i = 1, \ldots, 5 \), 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 \( A^+ \circ x \lor A^- \circ \neg x = b \) where “\( \circ \)” and “\( \lor \)” denote the max-product and maximum operator, respectively. Moreover, if he expends cost \( x^*_j \) for zones \( j = 1, \ldots, 6 \), to overcome the requirement from the type of \( i \), \( i = 1, \ldots, 5 \), in zone \( j \), then the investment risk in zone \( j \) is obtained by \( x^* j \), where \( r_j \in R^+ \). Now, the manager intends to minimize the maximum of risks subject to the system \( A^+ \circ x \lor A^- \circ \neg x = b \) and \( x \in [0, 1]^6 \). This problem can equivalently be modelled as follows:

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

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

**Example 6.2.** Consider the following problem.

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

(6)

(7)
where the matrices $A^+$, $A^-$, and $b$ are as follows:

$$A^+ = \begin{pmatrix}
0.35 & 0.21 & 0.14 & 0.72 & 0.30 \\
0.51 & 0.80 & 0.39 & 0.20 & 0.75 \\
0.45 & 0.30 & 0.22 & 0.90 & 0.40 \\
0.10 & 0.09 & 0.11 & 0.24 & 0.15 \\
0.25 & 0.40 & 0.26 & 0.11 & 0.07 \\
0.55 & 0.20 & 0.40 & 0.49 & 0.30 \\
0.40 & 0.23 & 0.70 & 0.44 & 0.33
\end{pmatrix}, \quad A^- = \begin{pmatrix}
0.29 & 0.17 & 0.33 & 0.60 & 0.09 \\
0.48 & 0.30 & 0.40 & 1.00 & 0.50 \\
0.12 & 0.26 & 0.31 & 0.40 & 0.45 \\
0.08 & 0.03 & 0.05 & 0.06 & 0.10 \\
0.60 & 0.25 & 0.21 & 0.14 & 0.16 \\
0.25 & 1.00 & 0.14 & 0.50 & 0.22 \\
0.44 & 0.25 & 0.37 & 0.18 & 0.41
\end{pmatrix},$$

and $b = (0.36 \ 0.60 \ 0.45 \ 0.12 \ 0.30 \ 0.55 \ 0.49)^T$.

We are now ready to solve this problem by Algorithm 1.

**Step 1.** The lower and upper bound are as follows: $\bar{x} = (0.50, 0.45, 0, 0.40, 0)^T$ and $\bar{x} = (1, 0.75, 0.70, 0.50, 0.80)^T$.

**Step 2.** Since $b_i > 0$, for each $i \in I = \{1, \ldots, 7\}$ and $\bar{x}_j < \bar{x}_j$, for each $j \in J = \{1, \ldots, 5\}$, we go to Step 3.

**Step 3.** The characteristic matrices $Q^+$ and $Q^-$ are as follows:

$$Q^+ = \begin{pmatrix}
1 & 2 & 3 & 4 & 5 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
7 & 0 & 0 & 1 & 0
\end{pmatrix} \quad \text{and} \quad Q^- = \begin{pmatrix}
1 & 2 & 3 & 4 & 5 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
7 & 0 & 0 & 0 & 0
\end{pmatrix}.$$
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}_1, \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, \hat{x}_5\} which only \hat{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 satisfy 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 \(x^*_1 = \hat{x}_4 = 0.50, x^*_5 = \hat{x}_5 = 0.80\) 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 (11)-(12) 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 \(Z(x^*) = \sum_{j=1}^{5} (c_{ij}x^*_{ij}) = 6.12\).

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 \(T_0 = O((2n)m)\). If we apply Algorithm 1, then Step 6 consumes a computational cost as \(T_6 = O(2^m n^2 + 2^m nm^2 + n^2 m^3)\) 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: \(T_0 = O(2^m n^2)\) 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 \(7 \times 5\). Hence, the values \(m\) and \(n\) are as \(m = 7\) and \(n = 5\). Therefore, the computational cost in this case is as \(T_1 = T_9 = O(2^m n^2) = O(2^7 \times 5^2) = 10^7 \times O(1)\). But if we apply Algorithm 1 to solve Example 6.2, then Step 6 will consume the time \(T_6 = O(2^m n^2 + 2^m nm^2 + n^2 m^3) = O(2^{10} \times 5^2 + 2^{10} \times 5 \times 7^2 + 5^2 \times 7^2) = 285055 \times O(1)\), 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


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


