An Efficient Algorithm to Computing Max–Min Inverse Fuzzy Relation for Abductive Reasoning

Sumantra Chakraborty, Amit Konar, Member, IEEE, and Lakhmi C. Jain

Abstract—This paper provides an alternative formulation to computing the max–min inverse fuzzy relation by embedding the inherent constraints of the problem into a heuristic (objective) function. The optimization of the heuristic function guarantees maximal satisfaction of the constraints, and consequently, the condition for optimality yields solution to the inverse problem. An algorithm for computing the max–min inverse fuzzy relation is proposed. An analysis of the algorithm indicates its relatively better computational accuracy and higher speed in comparison to the existing technique for inverse computation. The principle of fuzzy abduction is extended with the proposed inverse formulation, and the better relative accuracy of the said abduction over existing works is established through illustrations with respect to a predefined error norm.

Index Terms—Abductive reasoning, heuristic function, max–min inverse fuzzy relation.

I. INTRODUCTION

A FUZZY relation $R(x, y)$ describes a mapping from universe $X$ to universe $Y$ (i.e., $X \rightarrow Y$), and is formally represented by

$$R(x, y) = \{(x, y), \mu_R(x, y) \} | (x, y) \in X \times Y \}$$

where $\mu_R(x, y)$ denotes the membership of $(x, y)$ to belong to the fuzzy relation $R(x, y)$.

Let $X$, $Y$, and $Z$ be three universes and $R_1(x, y)$, for $(x, y) \in X \times Y$ and $R_2(y, z)$, for $(y, z) \in Y \times Z$ be two fuzzy relations. Then, max–min composition operation of $R_1$ and $R_2$, denoted by $R_1 \circ R_2$, produces a fuzzy relation defined by

$$R_1 \circ R_2 = \left\{(x, z), \min_y \{\mu_R_1(x, y), \mu_R_2(y, z)\} \right\}$$

where $x \in X$, $y \in Y$, and $z \in Z$.

For brevity, we would use “$\land$” and “$\lor$” to denote min and max, respectively. Thus

$$R_1 \circ R_2 = \left\{(x, z), \min_y \{\mu_R_1(x, y) \land \mu_R_2(y, z)\} \right\}.$$ (3)

The membership function of $(x, z)$ in the max–min composition relation $R_1 \circ R_2$ is often denoted by $\mu_{R_1 \circ R_2}$. Formally, it is defined by

$$\mu_{R_1 \circ R_2}(x, z) = \min_y \{\mu_R_1(x, y) \land \mu_R_2(y, z)\}.$$ (4)

A. Fuzzy Max–Min Inverse Relation

Let $R_1$ and $R_2$ be two fuzzy relational matrices of dimension $(n \times m)$ and $(m \times n)$, respectively. When $R_1 \circ R_2 = I$, the identity relation, we define $R_1$ as the preinverse to $R_2$, and $R_2$ as the postinverse to $R_1$. It is easy to note that when $R_1 = R_2 = I$, $R_1 \circ R_2 = I$ follows. However, when $R_2 \neq I$, we cannot find any $R_1$ that satisfies $R_1 \circ R_2 = I$. Analogously, when $R_1 \neq I$, we do not have any $R_2$, such that $R_1 \circ R_2 = I$.

It is apparent from the last two statements that we can only evaluate approximate max–min preinverse to $R_2$ or postinverse to $R_1$, when we know $R_1$ or $R_2$, respectively.

Suppose, we consider the preinverse computation problem. Therefore, given $R_2$, we need to find $R_1$, such that $R_1 \circ R_2 = I'$, where $I'$ is sufficiently close to $I$. The closeness of $R_1 \circ R_2$ to $I$ can be measured by a new distance norm defined by

$$D = \sum_{i=1}^{n} \sum_{l=1}^{n} | (R_1 \circ R_2)_{i,i} - I_{i,i} |$$ (5)

$$D = \sum_{i=1}^{n} \left[ (I_{i,i} - (R_1 \circ R_2)_{i,i} ) + \sum_{l=1}^{n} \left( (R_1 \circ R_2)_{i,i} - I_{i,i} \right) \right]$$ (6)

where $D$ should not exceed a small predefined real number. The definition of approximate postinverse relation to $R_1$ may also be given analogously.

B. Review

One simple approach to computing approximate preinverse $R_1$ satisfying $R_1 \circ R_2 = I$, is to exhaustively generate the elements of $R_1$ in the interval $[0, 1]$, and then identify the best $R_1$ that minimizes the distance norm $D$, given by (6).
Computational cost for such algorithm would be exceedingly high, and thus is not amenable for implementation in practice. In this paper, we provide an alternative formulation of the preinverse computation problem by a heuristic approach that gives a solution (and optimal solution too) with a time complexity of $O(n^2)$, where $n$ denotes the number of rows in matrix $R_1$.

The origin of the proposed max–min fuzzy inverse computation problem dates back to the middle of 1970s, when the researchers took active interest to find a general solution to fuzzy relational equations involving max–min composition operation. The pioneering contribution of solving max–min composition-based relational equation goes to Sanchez [38]. The work was later studied and extended by Prevot [33], Czogala et al. [9], Lettieri and Liguori [22], Luoh et al. [24], and Wu and Guu [44] for finite fuzzy sets [17]. Cheng–Zhong [8] and Wang et al. [43] proposed two distinct approaches to computing intervals of solutions for each element of an inverse fuzzy relation. Higashi and Klir [15] introduced a new approach to computing maximal and minimal solutions to a fuzzy relational equation. Among the other well-known approaches to solve fuzzy relational equations, the works presented in [11], [13], [14], [16], [23], [31], [32], [46], and [47] need special mention. The early approaches previously mentioned unfortunately is not directly applicable to find general solutions $R_1$, satisfying the relational equation: $R_1 \circ R_2 = I$, as feasible solution exists only for the special case $R_1 = R_2 = I$. Interestingly, there are problems like fuzzy backward/abductive [18] reasoning, where $R_2 \neq I$, but $R_1$ needs to be evaluated. This demands a formulation to determine a suitable $R_1$, such that the given relational equation is best satisfied. Since satisfying the relational equation refers to satisfying the underlying constraints, we need to construct a suitable objective function, involving the constraints, so that optimization of the objective function ensures optimal (best) satisfaction of the constraints.

Several direct (or indirect) formulations of the max–min preinverse computing problem have been addressed in the literature [2], [6], [8], [26]–[29], [36], [37], [39]. A first attempt to compute fuzzy preinverse with an aim to satisfy all the underlying constraints in the relational equation using a heuristic objective function is addressed in [37]. The work, however, is not free from limitations as the motivations to optimize the heuristic (objective) function to optimally satisfy all the constraints are not fully realized due to the following reasons. The heuristic function employed in [37] constrains only the largest min term, although the motivation was to constrain all the necessary $(n-1)$ min terms (see Theorem 9, Appendix). Second, an attempt to independently maximizing the heuristic functions to determine elements in $R_1$, ultimately sets in large values for the nondiagonal elements of $(R_1 \circ R_2)$ (see Theorem 10, Appendix), consequently failing to satisfy the necessary constraints.

This papers, however, overcomes both the limitations first by a suitable selection of the heuristic function, capable of constraining all the necessary min terms. Second, the independence of the elements in $R_1$ has been considered here to optimally derive a solution for $R_1$ that satisfies all the underlying constraints in $R_1 \circ R_2 = I$. Unlike in [37], where concurrently maximizing all the heuristic functions resulted in large values in the nondiagonal elements of $R_1 \circ R_2$, we here attempt to maximize the heuristic function to obtain only the largest element in each row of $R_1$. Consequently, the problem of concurrently maximizing the heuristic function addressed earlier does not arise here. Furthermore, to obtain other elements, except the largest, in each row of $R_1$, we attempt to minimize all the $(n-1)$ min-terms introduced above with an ultimate aim to minimize the nondiagonal elements of $R_1 \circ R_2$ toward zero.

The fuzzy inverse we introduced so far was pivoted around max–min composition operator. However, there exist other forms of fuzzy inverse defined with respect to triangular norms (T-norms) and max-product operators. Some of the significant contributions in this regard are due to Pedrycz [30], Sanchez [39], and Miyakoshi and Shimbo [25]. The work of Chen [6] in determining the relationship between T-ordering and the generalized inverses in this regard needs mentioning. Generalizations to L-fuzzy relations were explored by Di Nola and Sessa [10], Sessa [40], and Drewniak [12]. Bourke and Fisher [5] proposed a solution to max-product inverse fuzzy relation. Leotamonphong et al. [21] also proposed an interesting solution to the same problem. Wu and Guu [44] presented an efficient procedure for solving a fuzzy relational equation with Max–Archimedean ternorm composition operator. A detailed discussion on these works falls beyond the scope of this paper.

The work presented in this paper has been applied to fuzzy abductive (backward) reasoning [1], [3]. It may be mentioned that the classical fuzzy abduction employs generalized modus tollens (GMT), where the transpose of the implication relation for the given rule is used to extract the inference [7], [17]. In this paper, we employ the inverse of the implication relation to determine the fuzzy inference for the abductive reasoning problem. Such relational inverse-based abduction yields better accuracy in inference, and thus the proposed technique for abductive reasoning is expected to meet the demand of many engineering application such as fault diagnosis [20], [41].

The rest of this paper is organized as follows. In Section II, we provide Strategies used to solve the inverse computational problem. The algorithm is presented in Section III with numerical examples and analysis of time complexity. The issues of the optimal solution with an alternative heuristic function are addressed in Section IV. The application of max–min inverse fuzzy relation is discussed in Section V. Conclusions are listed in Section VI. The limitations of an existing algorithm [37] are indicated in the Appendix.

II. PROPOSED COMPUTATIONAL APPROACH TO FUZZY MAX–MIN PREINVERSE RELATION

Given a fuzzy relational matrix $R$ of dimension $(m \times n)$, we need to evaluate a $Q$ matrix of dimension $(n \times m)$ such that $Q \circ R = I \approx I$, where $I$ denotes identity matrix of dimension $(n \times n)$. Let $q_i$ be the $i$th row of $Q$ matrix. The following strategies have been adopted to solve the equation $Q \circ R = I \approx I$ for known $R$.

Strategy 1—Decomposition of $Q \circ R \approx I$ into $[q_i, o R]_l \approx 1$ and $[q_i, o R]_{l \neq i} \approx 0$. Since $Q \circ R \approx I$, $q_i \circ R \approx$ $i$th row of $I$ matrix, therefore, the $i$th element of $q_i \circ R$, denoted by $[q_i, o R]_l \approx 1$ and the $l$th element (where $l \neq i$) of $q_i \circ R$, denoted by $[q_i, o R]_{l \neq i} \approx 0$.

Strategy 2—Determination of the Effective Range of $q_{ij}$ \forall $j$ in $[0, r_{ij}]$: Since $Q$ is a fuzzy relational matrix, its elements $q_{ij} \in [0, 1]$ for \forall $i, j$. However, to satisfy the constraint
occurs jointly for a suitable selection of $q_{ij}$, the range of $q_{ij} \forall j$ virtually becomes $[0, r_{ji}]$ by Lemma 1.

This range is hereafter referred to as effective range of $q_{ij}$.

**Lemma 1:** The constraint $|q_{i} o R|_i \approx 1$, sets the effective range of $q_{ij}$ in $[0, r_{ji}]$.

**Proof:**

$$\left[q_{i} o R\right]_i = \bigvee_{j=1}^{m} (q_{ij} \land r_{ji}).$$

Since

$$\left[\bigvee_{j=1}^{m} (q_{ij} \land r_{ji})\right]_{q_{ij}>r_{ji}} = \left[\bigvee_{j=1}^{m} (q_{ij} \land r_{ji})\right]_{q_{ij}=r_{ji}}$$

the minimum value of $q_{ij}$ that maximizes $|q_{i} o R|_i$ toward one is $r_{ji}$. Setting $q_{ij}$ beyond $r_{ji}$ is of no use in connection with maximization of $|q_{i} o R|_i$ toward one. Therefore, the effective range of $q_{ij}$ reduces from $[0, 1]$ to $[0, r_{ji}]$.

**Strategy 3—Replacement of the Constraint** $|q_{i} o R|_i \approx 1$, By $q_{ik} \approx 1$, Where $q_{ik} \geq q_{ij} \forall j$: We first prove $|q_{i} o R|_i = q_{ik}$ for $q_{ik} \geq q_{ij} \forall j$ by Lemma 2, and then argue that $|q_{i} o R|_i \approx 1$ can be replaced by $q_{ik} \approx 1$.

**Lemma 2:** If $q_{ik} \geq q_{ij} \forall j$, then $|q_{i} o R|_i = q_{ik}$.

**Proof:**

$$\left[q_{i} o R\right]_i = \bigvee_{j=1}^{m} (q_{ij} \land r_{ji}).$$

By Lemma 1, we can write $0 \leq q_{ij} \leq r_{ji} \forall j$. Therefore

$$(q_{ij} \land r_{ji}) = q_{ij}.$$ (8)

Substituting (8) in (7), yields the resulting expression as

$$\left[q_{i} o R\right]_i = \bigvee_{j=1}^{m} (q_{ij}) = q_{ik} \land q_{ij} \forall j.$$ (9)

The maximization of $|q_{i} o R|_i$, therefore, depends only on $q_{ik}$, and the maximum value of $|q_{i} o R|_i = q_{ik}$. Consequently, the constraint $|q_{i} o R|_i \approx 1$ is replaced by $q_{ik} \approx 1$. Discussion on Strategy 3 ends here. A brief justification to Strategy 4–6 is outlined next.

**Justification of Strategies 4 to 6:** In this paper, we evaluate the largest element $q_{ik}$ and other element $q_{ij}$ (for $j \neq k$) in $q_i$, the $i$th row of $Q$-matrix, by separate procedures. For evaluation of $q_{ik}$, we first need to identify the positional index $k$ of $q_{ik}$ so that maximization of $|q_{i} o R|_i$ and minimization of $|q_{i} o R|_{i,l \neq i}$ occur jointly for a suitable selection of $q_{ik}$. This is taken care of in Strategy 5 and 6. In Strategy 5, we determine $k$ for the possible largest element $q_{ik}$, whereas in Strategy 6, we evaluate $q_{ik}$. To determine $q_{ik}$ (for $j \neq k$), we only need to minimize $|q_{i} o R|_{i,l \neq i}$. This is considered in Strategy 4. It is indeed important to note that selection of $q_{ij}$ (for $j \neq k$) to minimize $|q_{i} o R|_{i,l \neq i}$ does not hamper maximization of $|q_{i} o R|_i$, as $|q_{i} o R|_i = q_{ik}$ (see Lemma 2).

**Strategy 4—Evaluation of $q_{ij}, \; j \neq k$, Where $q_{ik} \geq q_{ij} \forall j$:** The details of the aforementioned strategy are taken up in Theorem 1.

**Theorem 1:** If $q_{ik} \geq q_{ij} \forall j$, then the largest value of $q_{ij}|_{j \neq k}$ that minimizes $|q_{i} o R|_{i,l \neq i}$ toward zero is given by $q_{ik} \land \bigwedge_{l \neq i} r_{kl}$.

**Proof:**

$$\left[q_{i} o R\right]_{i,l \neq i} = \bigvee_{j=1}^{m} (q_{ij} \land r_{ji}) \quad \forall l, l \neq i$$

$$= \bigvee_{j=1}^{m} (q_{ij} \land r_{ji}) \lor (q_{ik} \land r_{kl}) \quad \forall l, l \neq i.$$ (10)

Therefore

$$\text{Min}[q_{i} o R]_{i,l \neq i} = \text{Min}_{\forall l, l \neq i} (q_{ik} \land r_{kl}) = q_{ik} \land \text{Min}_{\forall l, l \neq i} \{r_{kl}\}.$$ (11)

Since $\text{Min}[q_{i} o R]_{i,l \neq i} = q_{ik} \land \bigwedge_{l \neq i} r_{kl}$ and the largest value in $|q_{i} o R|_{i,l \neq i} = (q_{ik} \land r_{kl})$, therefore, $\text{Min}[q_{i} o R]_{i,l \neq i}$ will be the largest among $(q_{ij} \land r_{ji})$, $j \neq k$.

$$\text{Min}[q_{i} o R]_{i,l \neq i} = q_{ik} \land \bigwedge_{l \neq i} r_{kl}.$$ (12)

which is the same as

$$q_{ik} \land \bigwedge_{l \neq i} r_{kl} \geq (q_{ij} \land r_{ji}) \quad \forall j, j \neq k.$$ (13)

The largest value of $q_{ij}$ for $j \neq k$ can be obtained by setting equality in (13), and the resulting equality condition is satisfied when

$$q_{ij}|_{j \neq k} = q_{ik} \land \bigwedge_{l \neq i} r_{kl}.$$ (14)

**Strategy 5—Determining the Positional Index $k$ for the Element $q_{ik}$ ($\geq q_{ij} \forall j$) in $q_{i}$:** To determine the position $k$ of $q_{ik}$ in $q_{i}$, we first need to construct a heuristic function $h(q_{ik})$ that satisfies two constraints

i) Maximize $|q_{i} o R|_i$.

ii) Minimize $|q_{i} o R|_{i,l \neq i}$.
and then determine the index $k$, such that $h(q_{ik}) \geq h(q_{ij}) \forall j$.
In other words, we need to determine the positional index $k$ for the possible largest element $q_{ik}$ in the $i$th row of $Q$-matrix, as the largest value of $h(q_{ik})$ ensures maximization of the heuristic function $h(q_{ik})$, and thus best satisfies the constraints (15) and (16).

Formulation of the heuristic function is considered first, and the determination of $k$ satisfying $h(q_{ik}) \geq h(q_{ij}) \forall j$ is undertaken next.

One simple heuristic cost function that satisfies (15) and (16) is

$$h_1(q_{ik}) = q_{ik} - \frac{1}{n-1} \sum_{l \neq i} q_{ik} \land r_{kl}$$

where $q_{ik} \geq q_{ij} \ \forall j$ by Theorem 2. Next, we find $k$ such that $\text{Max}_{q_{ik} \in [0,r_{ki}]} h_1(q_{ik}) \geq \text{Max}_{q_{ij} \in [0,r_{ji}]} h_1(q_{ij})$, for all $j$.

**Theorem 2:** If $q_{ik} \geq q_{ij} \ \forall j$, then maximization of $[q_{i} \circ R]_i$ and minimization of $[q_{i} \circ R]_i$ can be represented by a heuristic function

$$h_1(q_{ik}) = q_{ik} - \frac{1}{n-1} \sum_{l \neq i} q_{ik} \land r_{kl}$$

Proof: Given $q_{ik} \geq q_{ij}$, for all $j$, Thus, from Lemma 2, we have

$$[q_{i} \circ R]_i = q_{ik}.$$ Further

$$\left[q_{i} \circ R\right]_{l \neq i} = (q_{i1} \land r_{1l}) \lor (q_{i2} \land r_{2l}) \lor \cdots \lor (q_{im} \land r_{ml}) \lor \cdots \lor (q_{im} \land r_{ml}), \quad \text{for } \forall l, l \neq i.$$ Now, by Theorem 1, we have $q_{ij}, j \neq k = q_{ik} \land (\bigvee_{l \neq i} r_{kl})$ and substituting this value in (17), we have $[q_{i} \circ R]_i \land [q_{i} \circ R]_i = q_{ik} \land r_{kl}$, for all $l, l \neq i$.

Now, to jointly satisfy maximization of $[q_{i} \circ R]_i$ and minimization of $[q_{i} \circ R]_i \land [q_{i} \circ R]_i$, we design a heuristic function

$$h_1(q_{ik}) = \left[q_{i} \circ R\right]_i - f((q_{ik} \land r_{k1}), \ldots, (q_{ik} \land r_{ki-1}), (q_{ik} \land r_{ki+1}), \ldots, (q_{ik} \land r_{kn})).$$

Now, for any monotonically nondecreasing function $f(\cdot)$, maximization of $[q_{i} \circ R]_i$ and minimization of the $n$ terms $(q_{ik} \land r_{k1}), \ldots, (q_{ik} \land r_{ki-1}), (q_{ik} \land r_{ki+1}), \ldots, (q_{ik} \land r_{kn})$ calls for maximization of $h_1(q_{ik})$. Since averaging (Avg) is a monotonically increasing function, we replace $f(\cdot)$ by $\text{Avg}(\cdot)$. Thus

$$h_1(q_{ik}) = \text{Avg}((q_{ik} \land r_{k1}), \ldots, (q_{ik} \land r_{ki-1}), (q_{ik} \land r_{ki+1}), \ldots, (q_{ik} \land r_{kn}))$$

$$= q_{ik} - \frac{1}{n-1} \sum_{l \neq i} q_{ik} \land r_{kl}. \quad (18)$$

Although apparent, it may be added for the sake of completeness that $n < 1$ in (18).

The determination of index $k$, such that $h_1(q_{ik}) \geq h_1(q_{ij}) \ \forall j$ can be explored now. Since $q_{ij} \in [0,r_{ji}]$ and $q_{ik} \in [0,r_{ki}]$, therefore $h_1(q_{ik}) \geq h_1(q_{ij}) \ \forall j$ can be transformed to

$$\text{Max}_{q_{ik} \in [0,r_{ki}]} h_1(q_{ik}) \geq \text{Max}_{q_{ij} \in [0,r_{ji}]} h_1(q_{ij}), \quad \text{for all } j. \quad (19)$$

Consequently, determination of $k$ satisfying the inequality (19) yields the largest element $q_{ik}$ in the $q_i$ that maximizes the heuristic function $h_1(\cdot)$.

**Corollary 1:** $h_2(q_{ik}) = q_{ik} - \sum_{l \neq i} (q_{ik} \land r_{kl})$ too is a heuristic function that maximizes $[q_{i} \circ R]_i$ and minimizes $[q_{i} \circ R]_i \land [q_{i} \circ R]_i$.

Proof: Since $\sum_{l \neq i} (q_{ik} \land r_{kl})$ is a monotonically nondecreasing function, thus the Corollary can be proved similar to Theorem 2.

**Strategy 6—Finding the Maximum Value of $h_1(q_{ij})$ for $q_{ij} \in [0,r_{ji}]$:** We first of all prove that $h_1(q_{ij})$ is a monotonically nondecreasing function of $q_{ij}$ by Theorem 3. Then, we can easily verify that for $q_{ij} \in [0,r_{ji}]$, $h_1(q_{ij})$ is the largest, i.e., $h_1(q_{ij}) = h_1(q_{ij}) \geq h_1(q_{ij})$, is the largest of $h_1(q_{ij})$ for $q_{ij} \in [0,r_{ji}]$.

**Theorem 3:** $h_1(q_{ij}) = q_{ij} - (1/(n-1)) \sum_{l \neq i} (q_{ij} \land r_{ji})$, is a monotonically nondecreasing function of $q_{ij} \in [0,r_{ji}]$.

Proof: We consider two possible cases.

Case 1) If $q_{ij} < r_{ji} \ \forall l, l \neq i$, then

$$h_1'(q_{ij}) = \frac{dh_1(q_{ij})}{dq_{ij}}$$

$$= 1 - \frac{1}{n-1} \sum_{l \neq i} \frac{d(r_{ji})}{dq_{ij}}$$

$$= 1(< 0) \quad \text{(since } \frac{d(r_{ji})}{dq_{ij}} = 0).$$

Case 2) Let $q_{ij} \leq r_{ji}$ for at least one $l$ (for example $t$ times), then

$$h_1'(q_{ij}) = \frac{dh_1(q_{ij})}{dq_{ij}}$$

$$= 1 - \frac{1}{n-1} \frac{d}{dq_{ij}} \sum_{t \text{times}} (q_{ij})$$

$$= 1 - \frac{t}{n-1} \geq 0, \quad t \leq (n-1)$$

Thus, $h_1'(q_{ij}) \geq 0$, and therefore, $h_1(\cdot)$ is a monotonically nondecreasing function of $q_{ij}$ in $[0,r_{ji}]$.

**Strategy 7—Finding $q$-Matrix:** Evaluation of each row $q_i$ of $Q$-matrix is considered independently. For given row $q_i$, we need to determine the largest element $q_{ik} \geq q_{ij} \ \forall j$. After $q_{ik}$ is evaluated, we evaluate $q_{ij}, j \neq k \ \forall j$. Subsequently, since the maximization of $h_1(\cdot)$ ensures satisfaction of the constraints (15) and (16) in Strategy 5, to determine $q_{ik}$, we look for an index $k$, such that

$$\text{Max}_{q_{ik} \in [0,r_{ki}]} h_1(q_{ik}) \geq \text{Max}_{q_{ij} \in [0,r_{ji}]} h_1(q_{ij}), \quad \text{for all } j.$$
Furthermore, since \( h_1(\cdot) \) is a monotonically nondecreasing function, aforementioned inequality reduces to

\[
h_1(q_{ik})|_{q_{ik} = r_{ki}} \geq h_1(q_{ij})|_{q_{ij} = r_{kj}}, \tag{20}
\]

If a suitable value of \( k \) is found satisfying (20), we say that \( q_{ik} \) is the largest element in \( q_i \) and the value of \( q_{ik} = r_{ki} \). In case there exist more than one value of \( k \in \{x, y, z, \ldots \} \subseteq \{1, 2, \ldots, m\} \) that satisfy (20), we select \( k = k' \) that minimizes distance norm

\[
D_i = ||q_i o R||_j - I_{ij}
\]

\[
= \{1 - [q_i o R]_i\} + \sum_{i=1}^n ([q_i o R]_i - 0)
\]

\[
= \{1 - q_{ik}\} + \sum_{i=1}^n (q_{ik} \land r_{kl})
\]

where \( q_{ik} \geq q_{ij} \) for all \( j \). For convenience, we sometimes write \( D_i \) as \( D_i[q_{ik}] \). Minimization of the distance norm \( D_i \) ensures optimal selection of \( q_{ik} \) and hence \( q_{ij} \), where \( q_{ik} \geq q_{ij} \) for all \( j \). Suppose, for \( k = x \) and \( k = y \), \( h_1(q_{ix}) = h_1(q_{iy}) \) and \( q_{ix} < q_{iy} \), then it can be proved by Theorem 11 (see Appendix) that \( D_i[q_{ix}] < D_i[q_{iy}] \), which suggests that selection of \( k = x \) is a better choice. Consequently, when \( k \in \{x, y, z, \ldots \} \subseteq \{1, 2, \ldots, m\} \), we select \( k = k' \), for which \( D_i[q_{ik}] \leq D_i[q_{ik}] \) for all \( k \).

To determine other element \( q_{ij} \) in \( q_i, j \neq k \), we evaluate

\[
q_{ij}|_{j \neq k} = q_{ik} \land \left( \bigwedge_{i=1}^n r_{kl} \right)
\]

\[
= r_{ki} \land \left( \bigwedge_{i=1}^n r_{kl} \right), \quad \text{(since } q_{ik} = r_{ki} \text{)}
\]

\[
= \bigwedge_{i=1}^n r_{kl}. \tag{23}
\]

The principle for evaluation of \( q_{ij} \) for a given \( i \) can be repeated for \( i = 1 \) to \( n \) to determine \( Q \)-matrix.

III. PROPOSED FUZZY MAX–MIN PREINVERSE COMPUTATION ALGORITHM

The results obtained from the strategies in Section II are used here to construct Algorithm I for Max–Min preinverse computation for a given fuzzy relational matrix.

**ALGORITHM I**

**Input:** \( R = [r_{ij}]_{m \times n} \) where \( 0 \leq r_{ij} \leq 1 \) \( \forall i, j; \)

**Output:** \( Q = [q_{ij}]_{m \times n} \) where \( 0 \leq q_{ij} \leq 1 \) \( \forall i, j, \) such that \( Q o R \) is close enough to \( I; \)

**Begin**

For \( i = 1 \) to \( n \)

Evaluate \( q_{ik}(\cdot); \) //Determine the position \( k \) and value of the largest element \( q_{ik} \) in row \( i \) of \( Q.//

For \( j = 1 \) to \( m \)

If \( (j \neq k) \)

\[ q_{ij} = \text{Min}_l(r_{kl}); \] //Determine all the elements in the \( i \)th row of \( Q \) matrix except \( q_{ik}.//

End If;

End For;

End If;

End For;

End End For;

End.

Evaluate \( q_{ik}(\cdot) \)

**Begin**

For \( j = 1 \) to \( m \)

Initialize: \( q_{ij} \leftarrow r_{ji}; \)

\[ h_1(q_{ij}) = q_{ij} - (1/(n - 1)) \sum_{l=1}^n (q_{lj} \land r_{jl}); \]

//Evaluate \( h_1(q_{ij}). //

End For;

End If;

End For;

End For;

End End For;

End.

A. Explanation of Algorithm I

Algorithm I evaluates the elements in \( q_i \), i.e., \( q_{i1}, q_{i2}, \ldots, q_{im} \) in a single pass by determining the position \( k \) of the largest element \( q_{ik} \) in the \( i \)th row and then its value \( r_{ki} \). Next, we determine the other elements in \( q_i \), which is given by \( q_{ij}|_{j \neq k} = \bigwedge_{i=1}^n r_{kl} \). The outer For-loop in the algorithm sets \( i = 1 \) to \( n \) with an increment in \( i \) by 1, and evaluation of \( q_i \) takes place for each setting of \( i \).

The most important step inside this outer For-loop is determining positional index \( k \) of the largest element \( q_{ik} \) in \( q_i \) and evaluation of its value. This has been taken care of in function Evaluate \( q_{ik}(\cdot) \).

B. Example

Given

\[
R = \begin{bmatrix}
  r_{11} & r_{12} & r_{13} \\
  r_{21} & r_{22} & r_{23} \\
  r_{31} & r_{32} & r_{33}
\end{bmatrix} = \begin{bmatrix}
  0.9 & 0.6 & 0.3 \\
  1.0 & 0.8 & 0.5 \\
  1.0 & 1.0 & 0.8
\end{bmatrix}
\]

we now provide a trace of the Algorithm I in Table I. For a given loop index \( i, 1 \leq i \leq 3 \), in the algorithm, we have shown the traces of evaluation of \( q_{i1}, q_{i2} \), and \( q_{i3} \) in a separate box. The results of evaluation of \( [q_{ij}] \) is finally assembled together

\[
Q = \begin{bmatrix}
  0.9 & 0.3 & 0.3 \\
  0.6 & 0.3 & 0.3 \\
  0.3 & 0.3 & 0.3
\end{bmatrix}.
\]
TABLE I
TRACE OF ALGORITHM I

<table>
<thead>
<tr>
<th>j=1</th>
<th>Evaluate ( q_{ik}(i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>With ( q_{11} = r_{11} = 0.9 ), ( h(q_{11}) = q_{11} ) (</td>
<td>{(q_{11} \land r_{12}) \land (q_{11} \land r_{13})} / 2 = 0.45 )</td>
</tr>
<tr>
<td>With ( q_{12} = r_{12} = 1.0 ), ( h(q_{12}) = q_{12} ) (</td>
<td>{(q_{12} \land r_{22}) \land (q_{12} \land r_{23})} / 2 = 0.35 )</td>
</tr>
<tr>
<td>With ( q_{13} = r_{13} = 1.0 ), ( h(q_{13}) = q_{13} ) (</td>
<td>{(q_{13} \land r_{32}) \land (q_{13} \land r_{33})} / 2 = 0.1 )</td>
</tr>
<tr>
<td>Max( h(q_{11}), h(q_{12}), h(q_{13}) = h(q_{11}) ). Return ( q_{11} = r_{11} = 0.9 ) and ( k=1 ).</td>
<td></td>
</tr>
</tbody>
</table>

Evaluate \( q_{ij} \) for all j except \( k=1 \).

\( q_{12} = \text{Min}(r_{11}, r_{12}, r_{13}) = 0.3 \); \( q_{13} = \text{Min}(r_{11}, r_{12}, r_{13}) = 0.3 \).

<table>
<thead>
<tr>
<th>j=2</th>
<th>Evaluate ( q_{ik}(i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>With ( q_{21} = r_{21} = 0.6 ), ( h(q_{21}) = q_{21} ) (</td>
<td>{(q_{21} \land r_{12}) \land (q_{21} \land r_{13})} / 2 = 0.15 )</td>
</tr>
<tr>
<td>With ( q_{22} = r_{22} = 0.8 ), ( h(q_{22}) = q_{22} ) (</td>
<td>{(q_{22} \land r_{22}) \land (q_{22} \land r_{23})} / 2 = 0.15 )</td>
</tr>
<tr>
<td>With ( q_{23} = r_{23} = 1.0 ), ( h(q_{23}) = q_{23} ) (</td>
<td>{(q_{23} \land r_{32}) \land (q_{23} \land r_{33})} / 2 = 0.1 )</td>
</tr>
<tr>
<td>Since, Max( h(q_{21}), h(q_{22}), h(q_{23}) = h(q_{21}) ). and ( q_{21} &lt; q_{22} )</td>
<td></td>
</tr>
<tr>
<td>return ( q_{21} = r_{21} = 0.6 ) and ( k=1 ).</td>
<td></td>
</tr>
</tbody>
</table>

Evaluate \( q_{ij} \) for all j except \( k=1 \).

\( q_{22} = \text{Min}(r_{11}, r_{12}, r_{13}) = 0.3 \); \( q_{23} = \text{Min}(r_{11}, r_{12}, r_{13}) = 0.3 \).

<table>
<thead>
<tr>
<th>j=3</th>
<th>Evaluate ( q_{ik}(i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>With ( q_{31} = r_{31} = 0.3 ), ( h(q_{31}) = q_{31} ) (</td>
<td>{(q_{31} \land r_{12}) \land (q_{31} \land r_{13})} / 2 = 0.0 )</td>
</tr>
<tr>
<td>With ( q_{32} = r_{32} = 0.5 ), ( h(q_{32}) = q_{32} ) (</td>
<td>{(q_{32} \land r_{22}) \land (q_{32} \land r_{23})} / 2 = 0.0 )</td>
</tr>
<tr>
<td>With ( q_{33} = r_{33} = 0.8 ), ( h(q_{33}) = q_{33} ) (</td>
<td>{(q_{33} \land r_{32}) \land (q_{33} \land r_{33})} / 2 = 0.0 )</td>
</tr>
<tr>
<td>Since, Max( h(q_{31}), h(q_{32}), h(q_{33}) = h(q_{31}) ). and ( q_{31} &lt; q_{32} ) and ( q_{32} &lt; q_{33} ) return ( q_{31} = r_{31} = 0.3 ) and ( k=1 ).</td>
<td></td>
</tr>
</tbody>
</table>

Evaluate \( q_{ij} \) for all j except \( k=1 \).

\( q_{32} = \text{Min}(r_{11}, r_{12}, r_{13}) = 0.3 \); \( q_{33} = \text{Min}(r_{11}, r_{12}, r_{13}) = 0.3 \).

C. Time Complexity

Since for finding the largest element \( q_{ik} \) in the \( i \)th row of the \( Q \) matrix, the outer for loop in Algorithm I is executed \( n \)-times, and for evaluation of \( q_{ij} \) for \( j \neq k \) for \( j \), the inner for loop is executed \( (n-1) \) times the time complexity of the algorithm is \( O(n + (n-1)) = O(2n-1) \). Thus, for \( i = 1 \) to \( n \), the complexity of the algorithm becomes \( O((2n-1)n) = O(2n^2 - n) \approx O(n^2) \).

In the existing work on fuzzy inverse [37], time complexity was found to be \( O(n^3) + K.O(n^2) \), where the first term is due to computation of all possible inverses of a given matrix, and the last term corresponds to evaluation of the best inverse among \( K \) number of inverses. Thus, the proposed algorithm saves an additional complexity by a factor greater than \( n \).

IV. Optimality of the Solution

In Section II, we proposed two heuristic functions \( h_1(\cdot) \) and \( h_2(\cdot) \). In this section, we demonstrate that \( h_2(\cdot) \) leads to optimal solution to the inverse problem, but the solution obtained by using \( h_2(\cdot) \) may not always be acceptable for abductive reasoning.

Given a fuzzy relational matrix \( R \), we say \( Q \) to be the optimal solution for the equation \( Q o R = I \), if the distance metric

\[
D = \sum_{i=1}^{n} D_i
\]

where \( D_i = \{1 - q_{ik}\} + \sum_{l=1 \atop l \neq 1}^{n} (q_{ik} \land r_{kl}) \) is minimum. We refer this as the condition for optimality.

The heuristic function \( h_2(\cdot) \), supports the condition for optimality (see Theorem 4).

Theorem 4: The maximization of the heuristic function \( h_2(q_{ik}) = q_{ik} - \sum_{l=1 \atop l \neq 1}^{n} (q_{ik} \land r_{kl}) \) ensures minimization of the condition for optimality \( D = \sum_{i=1}^{n} D_i \), where

\[
D_i = \{1 - q_{ik}\} + \sum_{l=1 \atop l \neq 1}^{n} (q_{ik} \land r_{kl}).
\]
Proof: Since
\[ D_i = \{1 - q_{ik}\} + \sum_{i=1}^{n} (q_{ik} \land r_{kl}) = 1 - h_2(q_{ik}) \] (24)
maximization of \( h_2(q_{ik}) \) ensures minimization of \( D_i \). Consequently, maximization of \( h_2(q_{ik}) \) for all \( i = 1 \) to \( n \) minimizes \( D = \sum_{i=1}^{n} D_i \).

A. Properties of \( h_2(\cdot) \)

Theorem 5: Given \( h_2(q_{ij}) = q_{ij} - \sum_{i=1}^{n} (q_{ij} \land r_{jl}) \) for \( q_{ij} \in [0, r_{ij}] \), the following inequalities hold depending on positivity of \( h_2(\cdot) \) as indicated:
\[
\begin{align*}
  h_2(q_{ij}) &\leq h_2(r_{ji}), \quad \text{if } h_2(q_{ij}) > 0 \\
  h_2(q_{ij}) &\leq h_2(0), \quad \text{if } h_2(q_{ij}) \leq 0.
\end{align*}
\]

Proof: Let \( h_2(q_{ij}) > 0 \). Then, following the definition of \( h_2(q_{ij}) \), we have
\[
q_{ij} > \sum_{i=1}^{n} (q_{ij} \land r_{jl}).
\]
(25)

Inequality (25) fails, if \( q_{ij} \leq r_{ji} \). Consequently, to satisfy the inequality, we set \( q_{ij} > r_{ji} \) \( \forall l, l \neq i \) in (25), which finally reduces to
\[
\sum_{i=1}^{n} (q_{ij} \land r_{jl}) = \sum_{i=1}^{n} r_{jl} < q_{ij}.
\]

As a consequence, \( h_2'(q_{ij}) = (dh_2(q_{ij})/dq_{ij}) = 1 > 0 \), i.e., \( h_2(q_{ij}) \) is a monotonically increasing function of \( q_{ij} \). Furthermore, since \( 0 \leq q_{ij} \leq r_{ji} \) by Lemma 1, therefore, \( [h_2(q_{ij})]_{\max} = h_2(r_{ji}) \). Thus, \( h_2(q_{ij}) \leq h_2(r_{ji}), \) if \( h_2(q_{ij}) > 0 \). This proves the first part of the theorem.

To prove the second part of the theorem, we presume \( h_2(q_{ij}) \leq 0 \). Again, \( h_2(q_{ij}) \big|_{q_{ij}=0} = 0 \) following definition of \( h_2(\cdot) \). Therefore, \( h_2(q_{ij}) \leq h_2(0) \). This proves the second part of the theorem.

B. Evaluation of \( Q \) Using \( h_2(\cdot) \) for Optimal Solution

To get optimal solution, we follow the first five strategies, listed in Section II, but consider a different heuristic function \( h_2(q_{ik}) = q_{ik} - \sum_{i=1}^{n} (q_{ik} \land r_{kl}) \). This heuristic function was obtained assuming \( q_{ik} \geq q_{ij} \), for all \( j \). Strategy 6 given below may be used to find an optimal \( Q \).

1) Strategy 6—Finding Optimal \( Q \)-Matrix: Evaluation of each row \( q_i \) of \( Q \)-matrix is considered independently. For a given row \( q_i \), we need to determine the largest element \( q_{ik} \geq q_{ij} \) \( \forall j \). After \( q_{ik} \) is evaluated, we evaluate \( q_{ij}, j \neq k \) in the subsequent phase.

Since \( q_{ik} \in [0, r_{ki}] \) and \( q_{ij} \in [0, r_{ji}] \), therefore, to obtain \( q_{ik} \geq q_{ij} \forall j \) in \( q_i \) by \( h_2(\cdot) \), we need to identify the value of \( q_{ik} \) such that
\[
\max_{q_{ik} \in [0, r_{ki}]} h_2(q_{ik}) = \max_{q_{ij} \in [0, r_{ij}]} h_2(q_{ij}), \quad \text{for all } j.
\]

Now, by Theorem 5, it is clear that \( \max_{q_{ij} \in [0, r_{ji}]} h_2(q_{ij}) = h_2(r_{ji}) \) when \( h_2(q_{ij}) > 0 \). If \( h_2(q_{ij}) \leq 0 \), then \( \max_{q_{ij} \in [0, r_{ji}]} h_2(q_{ij}) = h_2(0) \). Consequently, for positive \( h_2(q_{ik}), q_{ik} = r_{ki} \); and \( q_{ik} = 0 \) otherwise.

To evaluate \( q_{ij}(j \neq k) \) in \( q_i \), we use the following result by Theorem 5:
\[
q_{ij} \big|_{j \neq k} = q_{ik} \land \left( \bigwedge_{l=1}^{n} r_{kl} \right). \tag{26}
\]

If \( h_2(r_{ki}) \leq 0 \), then \( q_{ik} = 0 \) and
\[
q_{ij} \big|_{j \neq k} = (0) \land \left( \bigwedge_{l=1}^{n} r_{kl} \right) = 0
\]
otherwise, \( q_{ik} = r_{ki} \) and
\[
q_{ij} \big|_{j \neq k} = (r_{ki}) \land \left( \bigwedge_{l=1}^{n} r_{kl} \right) = \bigwedge_{l=1}^{n} r_{kl}. \tag{27}
\]

The principle for evaluation of \( q_{ij} \) for a given \( i \) can be repeated for \( i = 1 \) to \( n \) to determine \( Q \)-matrix.

C. Algorithm for Optimal Solution

The algorithm to obtain \( Q \) is similar with the given one in Section III, except the procedure for evaluating \( q_{ik} \), which is given by

**ALGORITHM II**

**Evaluate\_q_{ik}(\cdot)**

**Begin**

For \( j = 1 \) to \( m \)

Initialize: \( q_{ij} \leftarrow r_{ji}; \)

\( h_2(q_{ij}) = q_{ij} - \sum_{i=1}^{n} (q_{ij} \land r_{jl}); \)

//Evaluate \( h_2(q_{ij})//\)

If \( h_2(q_{ij}) \leq 0 \)

Then \( h_2(q_{ij}) = 0; \)

End if;

End For;

If \( h_2(q_{ij}) \geq h_2(q_{ij}) \forall j, \exists k \)

Then

If \( h_2(r_{ki}) \leq 0 \)

Then return \( k \) and \( q_{ik} = 0; //\)Return the position \( k \) of \( q_{ik} \), and its value//

Else return \( k \) and \( q_{ik} = r_{ki}. \)

End if;

End if;

**End.**

It is indeed important to note that if \( h_2(q_{ik}) \geq h_2(q_{ij}) \) for more than one \( k \), for example, \( k_1, k_2, k_3, \ldots \), then we can select any
TABLE II
TRACE OF ALGORITHM II

<table>
<thead>
<tr>
<th>i = 1</th>
<th>Evaluate ( q_{ik}(i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_{11} - r_{21} = 0.9 )</td>
<td>( h_2(q_{11}) = q_{11} \times ((q_{11} \cap r_{12}) \cap (q_{11} \cap r_{13})) = 0.2 )</td>
</tr>
<tr>
<td>( q_{12} - r_{22} = 0.3 )</td>
<td>( h_2(q_{12}) = q_{12} \times ((q_{12} \cap r_{22}) \cap (q_{12} \cap r_{23})) = 0.3 ); Since ( h_2(q_{12}) = 0 ); ( \vdots ); ( h_2(q_2) = 0 )</td>
</tr>
<tr>
<td>( q_{13} - r_{23} = 0.2 )</td>
<td>( h_2(q_{13}) = q_{13} \times ((q_{13} \cap r_{23}) \cap (q_{13} \cap r_{33})) = 0.2 ); Since ( h_2(q_{13}) = 0 ); ( \vdots ); ( h_2(q_3) = 0 )</td>
</tr>
<tr>
<td>Max ( { h_2(q_{11}), h_2(q_{12}), h_2(q_{13}) } = h_2(q_{11}) ); Return ( q_{11} = r_1 = 0.9 ) and ( k = 1 ), Evaluate ( q_{ij} ) for all ( j ) except ( k = 1 ),</td>
<td></td>
</tr>
<tr>
<td>( q_{12} = \text{Min}(r_{11}, r_{12}, r_{13}) = 0.3 ); ( q_{13} = \text{Min}(r_{11}, r_{12}, r_{13}) = 0.3 ) .</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>i = 2</th>
<th>Evaluate ( q_{ik}(i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_{21} - r_{21} = 0.4 )</td>
<td>( h_2(q_{21}) = q_{21} \times ((q_{21} \cap r_{12}) \cap (q_{21} \cap r_{13})) = 0.3 ); Since ( h_2(q_{21}) = 0 ); ( \vdots ); ( h_2(q_2) = 0 )</td>
</tr>
<tr>
<td>( q_{22} - r_{22} = 0.8 )</td>
<td>( h_2(q_{22}) = q_{22} \times ((q_{22} \cap r_{22}) \cap (q_{22} \cap r_{23})) = 0.5 ); Since ( h_2(q_{22}) = 0 ); ( \vdots ); ( h_2(q_2) = 0 )</td>
</tr>
<tr>
<td>( q_{23} - r_{23} = 1.0 )</td>
<td>( h_2(q_{23}) = q_{23} \times ((q_{23} \cap r_{23}) \cap (q_{23} \cap r_{33})) = 0.6 )</td>
</tr>
<tr>
<td>Max ( { h_2(q_{21}), h_2(q_{22}), h_2(q_{23}) } = h_2(q_{23}) ); Return ( q_{23} = r_{32} = 1.0 ) and ( k = 3 ), Evaluate ( q_{ij} ) for all ( j ) except ( k = 3 ),</td>
<td></td>
</tr>
<tr>
<td>( q_{21} = \text{Min}(r_{31}, r_{32}, r_{33}) = 0.2 ); ( q_{22} = \text{Min}(r_{31}, r_{32}, r_{33}) = 0.2 ) .</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>i = 3</th>
<th>Evaluate ( q_{ik}(i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_{31} - r_{31} = 0.3 )</td>
<td>( h_2(q_{31}) = q_{31} \times ((q_{31} \cap r_{12}) \cap (q_{31} \cap r_{13})) = 0.3 ); Since ( h_2(q_{31}) = 0 ); ( \vdots ); ( h_2(q_3) = 0 )</td>
</tr>
<tr>
<td>( q_{32} - r_{32} = 0.5 )</td>
<td>( h_2(q_{32}) = q_{32} \times ((q_{32} \cap r_{22}) \cap (q_{32} \cap r_{23})) = 0.5 ); Since ( h_2(q_{32}) = 0 ); ( \vdots ); ( h_2(q_3) = 0 )</td>
</tr>
<tr>
<td>( q_{33} - r_{33} = 0.2 )</td>
<td>( h_2(q_{33}) = q_{33} \times ((q_{33} \cap r_{33}) \cap (q_{33} \cap r_{33})) = 0.2 ); Since ( h_2(q_{33}) = 0 ); ( \vdots ); ( h_2(q_3) = 0 )</td>
</tr>
<tr>
<td>Max ( { h_2(q_{31}), h_2(q_{32}), h_2(q_{33}) } = h_2(q_{31}) = h_2(q_{32}) = h_2(q_{33}) ); we arbitrarily select ( k = 3 ) and return ( q_{33} = 0 ), Evaluate ( q_{ij} ) for all ( j ) except ( k = 3 ),</td>
<td></td>
</tr>
<tr>
<td>( q_{31} = 0 ); ( q_{32} = 0 ) .</td>
<td></td>
</tr>
</tbody>
</table>

value of \( k = k_1 \) or \( k_2 \) or \( k_3 \), for the reason indicated in Theorem 12 (see Appendix).

D. Relative Quality of Solutions Obtained by Algorithm I and Algorithm II

In this section, we attempt to compare the quality of solutions obtained by using Algorithms I and II, respectively. In abductive reasoning to be discussed in Section V, we need to identify the preinverse matrix \( Q \) with fewer or preferably no zeros. Fortunately, the Algorithm I never returns a \( Q \) with zero elements, when \( R \) contains no zeros. However, the Algorithm II may return a \( Q \)-matrix with one or more zeros, even when \( R \) contains no zeros. In this section, we study these properties by Theorem 6, 7, and 8, and compare the relative performance of the two Algorithms in the next section.

Theorem 6: If \( r_{ji} \neq 0 \) for any \( i, j \) then the obtained solutions \( q_{ij} \), for the equation \( QoR = I \) employing \( h_1(\cdot) \), are nonzero.

Proof: According to Algorithm I [using \( h_1(\cdot) \)] \( q_{ik} = r_{ki} \) when \( q_{ik} \geq q_{ij} \), for all \( j \), and \( q_{ij} | j \neq k = \bigwedge_{i=1}^{n} r_{ki} \). Thus, when \( r_{ji} \neq 0 \) for any \( i, j \), \( q_{ik} \), and \( q_{ij} | j \neq k \) are nonzero. Since this is valid for any row \( i \) of \( Q \)-matrix, therefore, \( Q \)-matrix contains no zero elements, when \( r_{ji} \neq 0 \) for all \( i, j \).
Theorem 7: If \( r_{ji} \neq 0 \) for any \( i, j \) then the solutions \( q_{ij} \), for the equation \( Q o R = I \) employing \( h_2(\cdot) \), need not be essentially nonzero.

Proof: According to Algorithm II, the optimal solution [using \( h_2(\cdot) \)] \( q_{ij} \) may be zero when \( q_{ik} \geq q_{ij} \), for all \( j \) and \( q_{ij} \neq k = 0 \) when \( r_{ji} \neq 0 \) for all \( i, j \). Therefore, \( Q \)-matrix need not essentially have nonzero elements, when \( r_{ji} \neq 0 \) for all \( i, j \).

Theorem 8: The Algorithm II returns a nonzero optimal preinverse \( Q \) to relational matrix \( R \) (having all nonzero elements) if the largest element in each row of \( R \) is \( \geq \) the sum of the remaining elements in the same row, subject to the condition that the largest elements of any two rows should not fall in the same column of \( R \).

Proof: By Algorithm II, to obtain nonzero optimal preinverse \( Q \) to relational matrix \( R \), we need to have \( h_2(\cdot) \geq 0 \) for at least one \( q_{ij} \) where \( j = 1 \) to \( m \) in \( q_i \). However, \( h_2(q_{ij}) \geq 0 \) yields the following:

\[
\text{Or, } h_2(q_{ij}) = r_{ji} - \sum_{l \neq i} (r_{ji} \land r_{jl}) \geq 0
\]

[if \( h_2(\cdot) \geq 0 \), \( q_{ij} = r_{ji} \), by Theorem 5].

\[
\text{Or, } r_{ji} \geq \sum_{l \neq i} (r_{ji} \land r_{jl}).
\]

Equation (28) will be valid when \( r_{ji} > r_{jl} \forall l, l \neq i \). Therefore, substituting this condition in (28), we have

\[
r_{ji} \geq \sum_{l \neq i} (r_{ji}).
\]

Since for a given column \( i \)

\[
r_{ji} \geq \sum_{l \neq i} (r_{jl})
\]

therefore \( r_{jl} \leq r_{ji} \) for all \( l \), except \( l = i \). Thus, to satisfy inequality (29) for all column \( i \), it can be ascertained that no two largest elements of row \( j \) and \( k \) can fall in the same column for \( j \neq k \). Consequently, the statement of the Theorem follows.

V. Application in Abductive Reasoning

The algorithm for preinverse introduced in Section II is useful for solving the well-known problems for fuzzy abductive reasoning [18], [19], [34], [39], outlined below.

Given a fuzzy production rule and the consequence as follows, we need to evaluate the premise.

\[ \text{Given: If } x \text{ is } A \text{ Then } y \text{ is } B \]

\[ \text{Given: } y \text{ is } B' \]

\[ \text{Find: } x \text{ is } A' \]

The procedure for evaluating the membership of the inference: \( x \) is \( A' \), is well known as GMT [18], [30], [35].

Let \( x = \text{height} \) and \( y = \text{speed} \) be two fuzzy linguistic variables in the universes \( X \) and \( Y \), respectively. \( A = \text{TALL} \) be a fuzzy set, where \( A \subseteq X \) and \( B = \text{HIGH} \) be a fuzzy set where \( B \subseteq Y \).

Let us consider instances of membership distribution as \( \mu_A(x) = \{0.3/'5', 0.6/'6', 0.9/'7'\} \) and \( \mu_B(y) = \{0.2/'7 \text{ m/s, } 0.4/'8 \text{ m/s, } 0.7/'10 \text{ m/s}\} \), which for convenience of our analysis is represented by two vectors: \( A = [0.3 \ 0.6 \ 0.9] \) and \( B = [0.2 \ 0.4 \ 0.7] \), respectively.

Given the observed distribution for \( B' = \neg B \), the abductive reasoning problem attempts to determine the membership distribution \( A' \) [37] by using

\[ A' = B' o R^{-1}. \]  

Here, \( R \) by Lukasiewicz implication function \( A \rightarrow B \) given in Fig. 1 and the preinverse to \( R \) as obtained in Section II is reproduced below

\[ R^{-1} = \begin{bmatrix} 0.9 & 0.3 & 0.3 \\ 0.6 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 \end{bmatrix}. \]

It may be added here that abductive reasoning, which holds good in fuzzy logic, is not allowed in classical (propositional [4], [30], [35]) logic. However, in classical logic when \( B' = \neg B \) (i.e., one’s complement of all components of \( B \)) and the rule: \( A \rightarrow B \) is supplied, we find \( A' = \neg A \) using (classical) \( \text{modus tollens} \). Thus, in a similar manner, we now test how good is our \( R^{-1} \) by evaluating \( A' \), when \( B' = \neg B \), and then comparing the results with the existing works. For instance, let

\[ B' = \neg B \]

\[ = [(1 - 0.2) \ (1 - 0.4) \ (1 - 0.7)] \]

\[ = [0.8 \ 0.6 \ 0.3]. \]

Given the relational matrix for the implication rule: \( A \rightarrow B \), Table I provides a comparative estimate of the error in retrieval of \( A' = \neg A \) from \( B' = \neg B \). The error in the present context is defined as the Euclidean distance between the computed vector \( A' \) from the known vector \( \neg A \). It is apparent from the table that the proposed method to evaluate fuzzy modus tollens yields the best results, when \( R^{-1} \) is computed by using Algorithm I or II, instead of the one reported in [37] and the usual definition of \( R^{-1} = R^T \), as used in GMT [17].

VI. Conclusion

This paper presents a new formulation of computing inverse fuzzy relation by determining the inherent constraints
in the problem definition, and attempting to satisfy them together without overconstraining or relaxing either. Two possible heuristic functions have been used to find a useful and feasible solution, and also an optimal solution to the inverse problem. Algorithm I employing the first heuristic function yields a feasible solution to fuzzy inverse, and such inverse computation method when applied to fuzzy abductive reasoning yields good accuracy with respect to a predefined Euclidean norm of error. It is interesting to note that abductive reasoning carried out with inverse computation by Algorithm I exhibits superiority over existing methods for fuzzy abduction [17], [37].

The Algorithm II gives optimal solution to the inverse computation problem. However, the inverse solutions obtained by Algorithm II are acceptable for abductive reasoning, when \( R^{-1} \) contains no or fewer zeros. The conditions for having nonzero \( R^{-1} \) have been determined. It states that the \( R^{-1} \) obtained by Algorithm II includes no zeros, if the largest element in any two rows should not fall in the same column of \( R \). When \( R^{-1} \) contains no zero elements, the computations of the elements of \( R^{-1} \) by the two proposed methods employ the same mathematical functions, and thus yield the same results. Consequently, either of the two algorithms for \( R^{-1} \) computation can be used for abductive reasoning, when the precondition for \( R^{-1} \) containing no zeros is ensured. However, when this precondition fails, Algorithm I is found to yield better \( R^{-1} \), and consequently abductive reasoning performed with the \( R^{-1} \) obtained by Algorithm I yields the best performance. This is established experimentally as well in Table III.

Time complexity of our proposed algorithm is only \( O(n^2) \), where \( n \) denotes number of rows in the fuzzy relational matrix. The proposed algorithms for fuzzy inverse thus make sense in fuzzy abductive reasoning, as their performance in solution quality far exceeds other formulation within a small computational complexity.

### APPENDIX

In this Appendix, we present the limitations of the work reported in [37], and the optimal selection of \( k \) in \( q_{jk} \). Theorems 9 and 10 justify the limitations of the work in [37], while Theorems 11 and 12 provide strategies for the selection of optimal \( k \) in \( q_{jk} \).

**Theorem 9:** Maximization of the heuristic function

\[
h(q_{ij}) = (q_{ij} \land r_{ji}) - \sum_{l=1}^{n} (q_{ij} \land r_{lj})
\]

to obtain \( q_{ij} \), does not ensure minimization of all the min terms \( q_{ij} \land r_{lj}, l = 1 to n, l \neq i \).

**Proof:** Maximization of \( h(q_{ij}) \), indicates maximization of \( (q_{ij} \land r_{ji}) \) and minimization of \( \bigvee_{l=1}^{n} (q_{ij} \land r_{lj}) \) for \( j = 1 to m \).

However

\[
\sum_{l=1}^{n} (q_{ij} \land r_{lj}) = (q_{ij} \land r_{ji}) \bigvee \left( \sum_{l=1}^{n} (q_{ij} \land r_{lj}) \right)
\]

when \( (q_{ij} \land r_{ji}) \geq (q_{ij} \land r_{lj}) \) for \( l = 1 to n, l \neq i \), \( \bigvee_{l=1}^{n} (q_{ij} \land r_{lj}) \) reduces to \( (q_{ij} \land r_{ji}) \).

Thus, minimization of \( \bigvee_{l=1}^{n} (q_{ij} \land r_{lj}) \) imposes a restriction on the upper bound term of its \( (q_{ij} \land r_{lj}) \) only, but not on all min terms \( (q_{ij} \land r_{lj}) \) for \( l = 1 to n, l \neq i \).

**Theorem 10:** Maximization of the heuristic functions,

\[
h(q_{ij}) = (q_{ij} \land r_{ji}) - \bigvee_{l=1}^{n} (q_{ij} \land r_{lj}), j = 1 to m,
\]
to obtain

<table>
<thead>
<tr>
<th>Method used for evaluating ( Q ), pre-inverse to ( R )</th>
<th>Evaluation of ( A' = B' \circ Q ), where ( B'=[0.8 \ 0.6 \ 0.3] )</th>
<th>Error = ( \left| A' - \gamma A \right| )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q = R^{-1} ) evaluated by Algorithm I (proposed method).</td>
<td>( A' = [0.8 \ 0.3 \ 0.3] )</td>
<td>( \left| \sum_{l=1}^{n} (A' - \gamma A)^2 \right|^{1/2} )</td>
</tr>
<tr>
<td>( Q = R^{-1} ) evaluated by Algorithm II (Optimal solution).</td>
<td>( A' = [0.8 \ 0.3 \ 0.3] )</td>
<td>( \left| \sum_{l=1}^{n} (A' - \gamma A)^2 \right|^{1/2} )</td>
</tr>
<tr>
<td>( Q = R^T ) Transpose of ( R ) (classical [17]).</td>
<td>( A' = [0.8 \ 0.8 \ 0.8] )</td>
<td>( \left| \sum_{l=1}^{n} (A' - \gamma A)^2 \right|^{1/2} )</td>
</tr>
<tr>
<td>( Q = R^{-1} ) by the method introduced in [37].</td>
<td>( A' = [0.8 \ 0.8 \ 0.8] )</td>
<td>( \left| \sum_{l=1}^{n} (A' - \gamma A)^2 \right|^{1/2} )</td>
</tr>
</tbody>
</table>
$q_{ij}$, attempting to satisfy $[q_{i} \circ R]_{ij} \approx 1$ and $[q_{i} \circ R]_{l \neq i j} \approx 0$; ultimately sets the worst case (largest) value of

$$[q_{i} \circ R]_{l} = \bigvee_{j=1}^{m} (r_{ij} \land r_{jl}), \quad l = 1 \text{ to } n \text{ and } l \neq i.$$ 

**Proof:** Larger the value of $q_{ij}$ for all j, closer the value of $[q_{i} \circ R]_{l}$ to 1. On the other hand, smaller the value of $q_{ij}$, close to the value of $[q_{i} \circ R]_{l \neq i j}$ to 0. Furthermore, the smallest value of $q_{ij}$ that maximizes $[q_{i} \circ R]_{l}$ is $q_{ij} = r_{ij}$. Setting $q_{ij} = r_{ij}$ for all j in $[q_{i} \circ R]_{l \neq i}$, we obtain

$$[q_{i} \circ R]_{l \neq i} = \bigvee_{j=1}^{m} (q_{ij} \land r_{jl}), \quad l = 1 \text{ to } n \text{ and } l \neq i$$

which, however, is much more than zero. Hence, the theorem follows.

**Theorem 11:** If $h_{1}(q_{ix}) = h_{1}(q_{iy})$ and $q_{ix} < q_{iy}$, then the error norm induced by $q_{ix}$, denoted by $D_{1}[q_{ix}]$, is smaller than the error norm induced by $q_{iy}$, denoted by $D_{1}[q_{iy}]$, where index $k$ is the position of the largest element in $q_{i}$. 

**Proof:** From (18), we have $h_{1}(q_{ik}) = q_{ik} - (1/(n-1)) \sum_{j=1}^{n} (q_{ik} \land r_{kj})$.

Given $h_{1}(q_{ix}) = h_{1}(q_{iy})$, so we have

$$q_{ix} = \frac{1}{(n-1)} \sum_{i=1}^{n} (q_{ij} \land r_{xi}) = \frac{1}{(n-1)} \sum_{i=1}^{n} (q_{ij} \land r_{yi})$$

or, $(n-1)q_{ix} = \sum_{i=1}^{n} (q_{ij} \land r_{xi}) = (n-1)q_{iy} - \sum_{i=1}^{n} (q_{ij} \land r_{yi})$

or, $(n-2)q_{ix} + q_{iy} = \sum_{i=1}^{n} (q_{ij} \land r_{xi})$

$$= (n-2)q_{iy} + q_{ix} - \sum_{i=1}^{n} (q_{ij} \land r_{yi})$$

or, $(n-2)q_{iy} + q_{ix} = \sum_{i=1}^{n} (1 - q_{ij} + q_{iy} \land r_{yi})$

$$= (n-2)q_{iy} + \sum_{i=1}^{n} (q_{ij} \land r_{xi})$$

or, $(n-2)q_{iy} + D_{1}[q_{iy}] = (n-2)q_{ix} + D_{1}[q_{ix}]$

or, $D_{1}[q_{ix}] - D_{1}[q_{iy}] = (n-2)(q_{ix} - q_{iy}).$ \hspace{1cm} (31)

Now, from (31), it is obvious that, as $q_{ix} < q_{iy}$, thus $D_{1}[q_{ix}] < D_{1}[q_{iy}]$, i.e., error will be less when we will assign $q_{ix}$ as lesser value. Here, $q_{ix}$ is the smaller than $q_{iy}$ and $h_{1}(q_{ix}) = h_{1}(q_{iy})$, so error norm for taking $x$ as a $k$ index is smaller over $y$ as a $k$ index.

**Theorem 12:** If $h_{2}(q_{ix}) = h_{2}(q_{iy})$ and $q_{ix} < q_{iy}$, then the error norm induced by $q_{ik} = q_{ix}$, denoted by $D_{1}[q_{ix}]$, is equal to the error norm induced by $q_{ik} = q_{iy}$, denoted by $D_{1}[q_{iy}]$, where index $k$ is the position of the largest element in $q_{i}$. 

**Proof:** $h_{2}(q_{ix}) = h_{2}(q_{iy})$. Therefore, $1 - h_{2}(q_{ix}) = 1 - h_{2}(q_{iy})$ or $D_{1}[q_{ix}] = D_{1}[q_{iy}]$ [using (24)]. Therefore, even if $q_{ix} < q_{iy}$, the error norm $D_{1}[q_{ix}] = D_{1}[q_{iy}]$. Hence, the theorem is proved. 

**REFERENCES**


Sumantra Chakraborty was born in Habra, India, on May 3, 1981. He received the B.E. degree in electronics and communication engineering from the MCKV Institute of Engineering, Vidyasagar University, Midnapur, India, in 2004 and the M.Tech. degree in intelligent automation and robotics from Jadavpur University, Calcutta, India, in 2008, where he is currently working toward the Ph.D. degree under his teacher A. Konar on Abductive Reasoning and Type-2 fuzzy logic.

He was with the IBM India Software Laboratory for around a year, and his working area was Power Processor (P7) Validation. He is currently a visiting faculty member with Jadavpur University, where he offers postgraduate-level courses on Intelligent Automation and Robotics. He is also a visiting faculty member with Jadavpur University, where he offers postgraduate-level courses on intelligent automation and robotics. His current research interests include artificial intelligence, fuzzy abduction, Q-learning, and robotics.

Lakshmi C. Jain is currently a Professor of knowledge-based engineering and the Director/Founder of the Knowledge-Based Intelligent Engineering Systems Centre, University of South Australia, Adelaide, Australia. His interests focus on artificial intelligence paradigms and their applications in complex systems, art-science fusion, e-education, e-healthcare, robotics, unmanned air vehicles, and intelligent agents.

Prof. Jain is a Fellow of the Institution of Engineers, Australia.

Amit Konar (M’97) received the B.E. degree from Bengal Engineering and Science University, Howrah, India, in 1983 and the M.E. Tel E., M.Phil., and Ph.D. (Engineering) degrees from Jadavpur University, Calcutta, India, in 1985, 1988, and 1994, respectively.

In 2006, he was a Visiting Professor for the Summer Courses with the University of Missouri, St. Louis. He is currently a Professor with the Department of Electronics and Telecommunication Engineering (ETCE), Jadavpur University. He is a founding coordinator of the M.Tech. program on Intelligent Automation and Robotics, offered by ETCE Department, Jadavpur University. He has supervised ten Ph.D. theses. He has around 200 publications in international journals and conferences. He is an author of six books, including two popular texts: Artificial Intelligence and Soft Computing (CRC Press, 2000) and Computational Intelligence: Principles, Techniques and Applications (Springer, 2005).

He serves as the Editor-in-Chief of the International Journal of Artificial Intelligence and Soft Computing. He is a member of the editorial board of five other international journals. His research areas include the study of computational intelligence algorithms and their applications to the entire domain of electrical engineering and computer science. Specifically, he works on fuzzy sets and logic, neurocomputing, evolutionary algorithms, Dempster–Shafer theory and Kalman filtering, and applied the principles of computational intelligence in image understanding, very large scale integration design, mobile robotics, and pattern recognition.

Dr. Konar was the recipient of the All India Council for Technical Education-accredited 1997–2000 Career Award for Young Teachers for his significant contribution in teaching and research.