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

$$R(x,y) = \{((x,y), \mu_R(x,y))\,|(x,y) \in X \times Y\} \tag{1}$$

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 $R_1\, o\, R_2$, produces a fuzzy relation defined by

$$R_1\, o\, R_2 = \left\{(x,z), \max_y \{\min\{\mu_{R_1}(x,y), \mu_{R_2}(y,z)\}\}\right\} \tag{2}$$

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

For brevity, we would use "$\wedge$" and "$\vee$" to denote min and max, respectively. Thus

$$R_1\, o\, R_2 = \left\{(x,z), \bigvee_y \{\mu_{R_1}(x,y) \wedge \mu_{R_2}(y,z)\}\right\}. \tag{3}$$

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

$$\mu_{R_1\, o\, R_2}(x,z) = \bigvee_y \{\mu_{R_1}(x,y) \wedge \mu_{R_2}(y,z)\}. \tag{4}$$

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 $R_1\, o\, R_2 = I$, 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 $R_1 = R_2 = I$, $R_1\, o\, R_2 = I$ follows. However, when $R_2 \neq I$, we cannot find any R_1 that satisfies $R_1\, o\, R_2 = I$. Analogously, when $R_1 \neq I$, we do not have any R_2, such that $R_1\, o\, R_2 = I$. 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 $R_1\, o\, R_2 = I'$, where I' is sufficiently close to I. The *closeness* of $R_1\, o\, R_2$ to I can be measured by a new distance norm defined by

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

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

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 $R_1\, o\, R_2 = I$, 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).

Manuscript received April 3, 2008; revised October 25, 2008. First published November 3, 2009; current version published December 16, 2009. This paper was recommended by Associate Editor T. H. Lee.

S. Chakraborty is with the Artificial Intelligence Laboratory, Department of Electronics and Telecommunication Engineering, Jadavpur University, Calcutta 700 032, India (e-mail: suman_ch54@rediffmail.com).

A. Konar is with the Artificial Intelligence Laboratory, Department of Electronics and Telecommunication Engineering, Jadavpur University, Calcutta 700 032, India (e-mail: konaramit@yahoo.co.in).

L. C. Jain is with the Knowledge-Based Intelligent Engineering Systems Centre, University of South Australia, Adelaide, SA 5095, Australia (e-mail: lakhmi.jain@unisa.edu.au).

Digital Object Identifier 10.1109/TSMCA.2009.2030784

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: $R_1 \, o \, R_2 = I$, 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 $(R_1 \, o \, R_2)$ (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 interdependence 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 $R_1 \, o \, R_2 = I$. Unlike in [37], where concurrently maximizing all the heuristic functions resulted in large values in the nondiagonal elements of $R_1 \, o \, R_2$, 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 $R_1 \, o \, R_2$ 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 Cen [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 t-norm 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 $Q \, o \, R = I' \approx I$, where I denotes identity matrix of dimension $(n \times n)$. Let q_i be the ith row of Q matrix. The following strategies have been adopted to solve the equation $Q \, o \, R = I' \approx I$ for known R.

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

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

$[q_i\, o\, R]_i \approx 1$, 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:

$$[q_i\, o\, R]_i = \bigvee_{j=1}^{m} (q_{ij} \wedge r_{ji}).$$

Since

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

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:

$$[q_i\, o\, R]_i = \bigvee_{j=1}^{m} (q_{ij} \wedge r_{ji}). \tag{7}$$

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

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

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

$$[q_i\, o\, R]_i = \bigvee_{j=1}^{m} (q_{ij}) = q_{ik} \text{ as } q_{ik} \geq q_{ij} \qquad \forall j. \tag{9}$$

■

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 ith 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 $[q_i\, o\, R]_{l,l\neq i}$ 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_{ij} (for $j \neq k$), we only need to minimize $[q_i\, o\, R]_{l,l\neq i}$. This is considered in Strategy 4. It is indeed important to note that selection of q_{ij} (for $j \neq k$) to minimize $[q_i\, o\, R]_{l,l\neq i}$ 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 $[q_i\, o\, R]_{l,l\neq i}$ toward zero is given by $q_{ik} \wedge (\bigwedge_{l=1, l\neq i}^{n} r_{kl})$.

Proof:

$$\begin{aligned}[q_i\, o\, R]_{l,l\neq i} &= \bigvee_{j=1}^{m} (q_{ij} \wedge r_{jl}) \qquad \forall l,\ l \neq i \\ &= \bigvee_{\substack{j=1\\ j\neq k}}^{m} (q_{ij} \wedge r_{jl}) \vee (q_{ik} \wedge r_{kl}) \qquad \forall l,\ l \neq i. \\ &= (q_{ik} \wedge r_{kl}) \qquad \forall l,\ l \neq i \\ &\qquad \text{if } (q_{ik} \wedge r_{kl}) \geq \bigvee_{\substack{j=1\\ j\neq k}}^{m} (q_{ij} \wedge r_{jl}).\end{aligned} \tag{10}$$

Therefore

$$\begin{aligned}\mathrm{Min}[q_i\, o\, R]_{l,l\neq i} &= \mathop{\mathrm{Min}}_{\forall l,\, l\neq i} (q_{ik} \wedge r_{kl}) \\ &= q_{ik} \wedge \mathop{\mathrm{Min}}_{\forall l,\, l\neq i} \{r_{kl}\} \\ &= q_{ik} \wedge \left(\bigwedge_{\substack{l=1\\ l\neq i}}^{n} r_{kl}\right).\end{aligned} \tag{11}$$

Since $\mathrm{Min}[q_i\, o\, R]_{l,l\neq i} = q_{ik} \wedge (\bigwedge_{l=1, l\neq i}^{n} r_{kl})$ and the largest value in $[q_i\, o\, R]_{l,l\neq i} = (q_{ik} \wedge r_{kl})$, therefore, $\mathrm{Min}[q_i\, o\, R]_{l,l\neq i}$, will be the largest among $(q_{ij} \wedge r_{jl}) \forall j,\ j \neq k$ if

$$\mathrm{Min}[q_i\, o\, R]_{l,l\neq i} = q_{ik} \wedge \left(\bigwedge_{\substack{l=1\\ l\neq i}}^{n} r_{kl}\right) \geq \bigvee_{\substack{j=1\\ j\neq k}}^{m} (q_{ij} \wedge r_{jl}) \tag{12}$$

which is the same as

$$q_{ik} \wedge \left(\bigwedge_{\substack{l=1\\ l\neq i}}^{n} r_{kl}\right) \geq (q_{ij} \wedge r_{jl}) \qquad \forall j,\ j \neq k. \tag{13}$$

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

$$q_{ij}|_{j\neq k} = q_{ik} \wedge \left(\bigwedge_{\substack{l=1\\ l\neq i}}^{n} r_{kl}\right). \tag{14}$$

■

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) $$\text{Maximize } [q_i\, o\, R]_i \tag{15}$$

ii) $$\text{Minimize } [q_i\, o\, R]_{l,l\neq i} \tag{16}$$

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 ith 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

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

where $q_{ik} \geq q_{ij} \qquad \forall j$

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 $[q_i\, o\, R]_i$ and minimization of $[q_i\, o\, R]_{l, l \neq i}$ can be represented by a heuristic function

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

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

$$[q_i\, o\, R]_i = q_{ik}.$$

Further

$$[q_i\, o\, R]_{l, l \neq i} = (q_{i1} \wedge r_{1l}) \vee (q_{i2} \wedge r_{2l}) \vee \cdots \vee (q_{ik} \wedge r_{kl}) \vee \cdots \vee (q_{im} \wedge r_{ml}), \qquad \text{for } \forall l,\ l \neq i. \quad (17)$$

Now, by Theorem 1, we have $q_{ij}|_{j \neq k} = q_{ik} \wedge (\bigwedge_{\substack{l=1 \\ l \neq i}}^{n} r_{kl})$ and substituting this value in (17), we have $[q_i\, o\, R]_{l, l \neq i} = (q_{ik} \wedge r_{kl})$, for $\forall l,\ l \neq i$.

Now, to jointly satisfy maximization of $[q_i\, o\, R]_i$ and minimization of $[q_i\, o\, R]_{l, l \neq i}$, we design a heuristic function

$$h_1(q_{ik}) = [q_i\, o\, R]_i - f\left((q_{ik} \wedge r_{k1}), \ldots, (q_{ik} \wedge r_{k,i-1}), (q_{ik} \wedge r_{k,i+1}), \ldots, (q_{ik} \wedge r_{kn})\right).$$

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

$$\begin{aligned} h_1(q_{ik}) &= [q_i\, o\, R]_i - \text{Avg}\left((q_{ik} \wedge r_{k1}), \ldots, (q_{ik} \wedge r_{k,i-1}), (q_{ik} \wedge r_{k,i+1}), \ldots, (q_{ik} \wedge r_{kn})\right) \\ &= q_{ik} - \frac{1}{(n-1)} \sum_{\substack{l=1 \\ l \neq i}}^{n} (q_{ik} \wedge r_{kl}). \end{aligned} \quad (18)$$

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

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

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

Corollary 1: $h_2(q_{ik}) = q_{ik} - \sum_{\substack{l=1 \\ l \neq i}}^{n} (q_{ik} \wedge r_{kl})$ too is a heuristic function that maximizes $[q_i\, o\, R]_i$ and minimizes $[q_i\, o\, R]_{l, l \neq i}$.

Proof: Since $\sum_{\substack{l=1 \\ l \neq i}}^{n} (q_{ik} \wedge r_{kl})$ 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 $[0, r_{ji}]$, $h_1(q_{ij})|_{q_{ij} = r_{ji}}$ is the largest, i.e., $h_1(q_{ij})|_{q_{ij} = r_{ji}} \geq h_1(q_{ij})|_{q_{ij} \in [0, r_{ji}]}$.

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

Proof: We consider two possible cases.

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

$$\begin{aligned} h_1'(q_{ij}) &= \frac{dh_1(q_{ij})}{dq_{ij}} \\ &= 1 - \frac{1}{(n-1)} \sum_{\substack{l=1 \\ l \neq i}}^{n} \frac{d(r_{jl})}{dq_{ij}} \\ &= 1 (< 0) \qquad \left(\text{since } \frac{d(r_{jl})}{dq_{ij}} = 0\right). \end{aligned}$$

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

$$\begin{aligned} h_1'(q_{ij}) &= \frac{dh_1(q_{ij})}{dq_{ij}} \\ &= 1 - \frac{1}{(n-1)} \frac{d}{dq_{ij}} \sum_{t\text{-times}} (q_{ij}) \\ &= 1 - \frac{t}{(n-1)} \geq 0, \qquad t \leq (n-1) \end{aligned}$$

$\therefore\ h_1'(q_{ij}) \geq 0$, 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}, $j \neq k$ in the subsequent phase.

Since 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

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

Furthermore, since $h_1(\cdot)$ is a monotonically nondecreasing function, aforementioned inequality reduces to

$$h_1(q_{ik})|_{q_{ik}=r_{ki}} \geq h_1(q_{ij})|_{q_{ij}=r_{ji}}. \tag{20}$$

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

$$D_i = |[q_i \, o \, R]_j - I_{ij}| \tag{21}$$

$$= \{1 - [q_i \, o \, R]_i\} + \sum_{\substack{l=1 \\ l \neq i}}^{n} \{[q_i \, o \, R]_l - 0\}$$

$$= \{1 - q_{ik}\} + \sum_{\substack{l=1 \\ l \neq i}}^{n} (q_{ik} \wedge r_{kl}) \tag{22}$$

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

$$q_{ij}|_{j \neq k} = q_{ik} \wedge \left(\bigwedge_{\substack{l=1 \\ l \neq i}}^{n} r_{kl} \right)$$

$$= r_{ki} \wedge \left(\bigwedge_{\substack{l=1 \\ l \neq i}}^{n} r_{kl} \right), \qquad (\text{since } q_{ik} = r_{ki})$$

$$= \bigwedge_{l=1}^{n} r_{kl}. \tag{23}$$

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: $Q = [q_{ij}]_{n \times m}$ 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_$q_{ik}(.)$; //Determine the position k and value of the largest element q_{ik} in row i of Q.//
    For $j = 1$ to m
      If $(j \neq k)$
        Then $q_{ij} = \mathrm{Min}_l\{r_{kl}\}$; //Determine all the elements in the ith row of Q matrix except q_{ik}//
      End If;
    End For;
  End For;
End.
Evaluate_$q_{ik}(.)$
Begin
  For $j = 1$ to m
    Initialize: $q_{ij} \leftarrow r_{ji}$;
    $h_1(q_{ij}) = q_{ij} - (1/(n-1)) \sum_{l=1, l \neq i}^{n} (q_{ij} \wedge r_{jl})$;
    //Evaluate $h_1(q_{ij})$.//
  End For;
  For $j = 1$ to m
  For $k \in \{x, y, z, \ldots\} \subseteq \{1, 2, \ldots, m\}$
    If for a unique k, $h_1(q_{ik}) \geq h_1(q_{ij})$ for all j
    Then return k and $q_{ik} = r_{ki}$; //Return the position k of q_{ik}, and its value//
    Else If $q_{ix} \leq q_{ik} \; \forall k$ //in case multiple k exist//
      Then return $k = x$ and $q_{ik} = q_{ix} = r_{xi}$;
        End If;
      End If;
    End For;
  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 ith row and then its value r_{ki}. Next, we determine the other elements in q_i, which is given by $q_{ij}|_{j \neq k} = \bigwedge_{l=1}^{n} r_{kl}$. 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 $q_{ik}(.)$.

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

$$\therefore \quad Q = \begin{bmatrix} 0.9 & 0.3 & 0.3 \\ 0.6 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 \end{bmatrix}.$$

TABLE I
TRACE OF ALGORITHM I

i =1

Evaluate_q_{ik} (i)

With $q_{11}=r_{11}=0.9$; $h_1(q_{11}) = q_{11} - \{(q_{11}\wedge r_{12})+(q_{11}\wedge r_{13})\}/2 = 0.45$

With $q_{12}=r_{21}=1.0$; $h_1(q_{12}) = q_{12} - \{(q_{12}\wedge r_{22})+(q_{12}\wedge r_{23})\}/2 = 0.35$

With $q_{13}=r_{31}=1.0$; $h_1(q_{13}) = q_{13} - \{(q_{13}\wedge r_{32})+(q_{13}\wedge r_{33})\}/2 = 0.1$

Max{ $h_1(q_{11})$, $h_1(q_{12})$, $h_1(q_{13})$}= $h_1(q_{11})$; Return $q_{11}=r_{11}=0.9$ and k=1.

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

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

i =2

Evaluate_q_{ik} (i)

With $q_{21}=r_{12}=0.6$; $h_1(q_{21}) = q_{21} - \{(q_{21}\wedge r_{11})+(q_{21}\wedge r_{13})\}/2 = 0.15$

With $q_{22}=r_{22}=0.8$; $h_1(q_{22}) = q_{22} - \{(q_{22}\wedge r_{21})+(q_{22}\wedge r_{23})\}/2 = 0.15$

With $q_{23}=r_{32}=1.0$; $h_1(q_{23}) = q_{23} - \{(q_{23}\wedge r_{31})+(q_{23}\wedge r_{33})\}/2 = 0.1$

Since, Max{ $h_1(q_{21})$, $h_1(q_{22})$, $h_1(q_{23})$ }= $h_1(q_{21}) = h_1(q_{22})$, and $q_{21} < q_{22}$
return $q_{21} = r_{12} = 0.6$ and k=1.

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

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

i =3

Evaluate_q_{ik} (i)

With $q_{31}=r_{13}=0.3$; $h_1(q_{31}) = q_{31} - \{(q_{31}\wedge r_{11})+(q_{31}\wedge r_{12})\}/2 = 0.0$

With $q_{32}=r_{23}=0.5$; $h_1(q_{32}) = q_{32} - \{(q_{32}\wedge r_{21})+(q_{32}\wedge r_{22})\}/2 = 0.0$

With $q_{33}=r_{33}=0.8$; $h_1(q_{33}) = q_{33} - \{(q_{33}\wedge r_{31})+(q_{33}\wedge r_{32})\}/2 = 0.0$

Since, Max{ $h_1(q_{31})$, $h_1(q_{32})$, $h_1(q_{33})$}= $h_1(q_{31}) = h_1(q_{32}) = h_1(q_{33})$ and $q_{31} < q_{32}$
and $q_{31} < q_{33}$, return $q_{31} = r_{13} = 0.3$ and k=1.

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

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

C. Time Complexity

Since for finding the largest element q_{ik} in the ith 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 $D_i = \{1 - q_{ik}\} + \sum_{\substack{l=1\\ l\neq i}}^{n} (q_{ik} \wedge r_{kl})$ 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 $h_2(q_{ik}) = q_{ik} - \sum_{\substack{l=1\\ l\neq i}}^{n} (q_{ik} \wedge r_{kl})$ ensures minimization of the condition for optimality $D = \sum_{i=1}^{n} D_i$, where

$$D_i = \{1 - q_{ik}\} + \sum_{\substack{l=1\\ l\neq i}}^{n} (q_{ik} \wedge r_{kl}).$$

Proof: Since

$$D_i = \{1 - q_{ik}\} + \sum_{\substack{l=1 \\ l \neq i}}^{n} (q_{ik} \wedge r_{kl}) = 1 - h_2(q_{ik}) \quad (24)$$

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 $h_2(q_{ij}) = q_{ij} - \sum_{\substack{l=1 \\ l \neq i}}^{n} (q_{ij} \wedge r_{jl})$ for $q_{ij} \in [0, r_{ji}]$, the following inequalities hold depending on positivity of $h_2(.)$ as indicated:

$$h_2(q_{ij}) \leq h_2(r_{ji}), \qquad \text{if } h_2(q_{ij}) > 0$$

$$h_2(q_{ij}) \leq h_2(0), \qquad \text{if } h_2(q_{ij}) \leq 0.$$

Proof: Let $h_2(q_{ij}) > 0$. Then, following the definition of $h_2(q_{ij})$, we have

$$q_{ij} > \sum_{\substack{l=1 \\ l \neq i}}^{n} (q_{ij} \wedge r_{jl}). \quad (25)$$

Inequality (25) fails, if $q_{ij} \leq r_{jl}$. Consequently, to satisfy the inequality, we set $q_{ij} > r_{jl}$ $\forall l$, $l \neq i$ in (25), which finally reduces to

$$\sum_{\substack{l=1 \\ l \neq i}}^{n} (q_{ij} \wedge r_{jl}) = \sum_{\substack{l=1 \\ l \neq i}}^{n} r_{jl} < q_{ij}.$$

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, $[h_2(q_{ij})]_{\text{Max}} = h_2(r_{ji})$. 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, $h_2(q_{ij})|_{q_{ij}=0} = 0$ following definition of $h_2(.)$. 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 $h_2(q_{ik}) = q_{ik} - \sum_{\substack{l=1 \\ l \neq i}}^{n} (q_{ik} \wedge r_{kl})$. 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

$$\operatorname*{Max}_{q_{ik} \in [0, r_{ki}]} h_2(q_{ik}) \geq \operatorname*{Max}_{q_{ij} \in [0, r_{ji}]} h_2(q_{ij}), \qquad \text{for all } j.$$

Now, by Theorem 5, it is clear that $\text{Max}_{q_{ij} \in [0, r_{ji}]} h_2(q_{ij}) = h_2(r_{ji})$ when $h_2(q_{ij}) > 0$. If $h_2(q_{ij}) \leq 0$, then $\text{Max}_{q_{ij} \in [0, r_{ji}]} h_2(q_{ij}) = h_2(0)$. 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:

$$q_{ij}\Big|_{j \neq k} = q_{ik} \wedge \left(\bigwedge_{\substack{l=1 \\ l \neq i}}^{n} r_{kl} \right). \quad (26)$$

If $h_2(r_{ki}) \leq 0$, then $q_{ik} = 0$ and

$$q_{ij}|_{j \neq k} = (0) \wedge \left(\bigwedge_{\substack{l=1 \\ l \neq i}}^{n} r_{kl} \right) = 0$$

otherwise, $q_{ik} = r_{ki}$ and

$$q_{ij}|_{j \neq k} = (r_{ki}) \wedge \left(\bigwedge_{\substack{l=1 \\ l \neq i}}^{n} r_{kl} \right) = \bigwedge_{l=1}^{n} r_{kl}. \quad (27)$$

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()
    Begin
        For j = 1 to m
            Initialize: q_ij ← r_ji;
            h_2(q_ij) = q_ij − Σ_{l=1, l≠i}^{n} (q_ij ∧ r_jl);
            //Evaluate h_2(q_ij)//
            If (h_2(q_ij) ≤ 0)
            Then h_2(q_ij) = 0;
        End if;
      End For;
      If (h_2(q_ik) ≥ h_2(q_ij)) ∀j, ∃k
        Then
            If (h_2(r_ki) ≤ 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

i = 1

Evaluate_q_{ik} (i)

$q_{11} = r_{11} = 0.9$; $h_2(q_{11}) = q_{11} - \{(q_{11} \wedge r_{12}) + (q_{11} \wedge r_{13})\} = 0.2$

$q_{12} = r_{21} = 1.0$; $h_2(q_{12}) = q_{12} - \{(q_{12} \wedge r_{22}) + (q_{12} \wedge r_{23})\} = -0.3$; Since $h_2(q_{12}) < 0$, $\therefore h_2(q_{12}) = 0$

$q_{13} = r_{31} = 0.2$; $h_2(q_{13}) = q_{13} - \{(q_{13} \wedge r_{32}) + (q_{13} \wedge r_{33})\} = -0.2$; Since $h_2(q_{13}) < 0$, $\therefore h_2(q_{13}) = 0$

Max$\{h_2(q_{11}), h_2(q_{12}), h_2(q_{13})\} = h_2(q_{11})$; Return $q_{11} = r_{11} = 0.9$ and k=1.

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

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

i = 2

Evaluate_q_{ik} (i)

$q_{21} = r_{12} = 0.4$; $h_2(q_{21}) = q_{21} - \{(q_{21} \wedge r_{11}) + (q_{21} \wedge r_{13})\} = -0.3$; Since $h_2(q_{21}) < 0$, $\therefore h_2(q_{21}) = 0$

$q_{22} = r_{22} = 0.8$; $h_2(q_{22}) = q_{22} - \{(q_{22} \wedge r_{21}) + (q_{22} \wedge r_{23})\} = -0.5$; Since $h_2(q_{22}) < 0$, $\therefore h_2(q_{22}) = 0$

$q_{23} = r_{32} = 1.0$; $h_2(q_{23}) = q_{23} - \{(q_{23} \wedge r_{31}) + (q_{23} \wedge r_{33})\} = 0.6$

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.

$q_{21} = \text{Min}(r_{31}, r_{32}, r_{33}) = 0.2$; $q_{22} = \text{Min}(r_{31}, r_{32}, r_{33}) = 0.2$.

i = 3

Evaluate_q_{ik} (i)

$q_{31} = r_{13} = 0.3$; $h_2(q_{31}) = q_{31} - \{(q_{31} \wedge r_{11}) + (q_{31} \wedge r_{12})\} = -0.3$; Since $h_2(q_{31}) < 0$, $\therefore h_2(q_{31}) = 0$

$q_{32} = r_{23} = 0.5$; $h_2(q_{32}) = q_{32} - \{(q_{32} \wedge r_{21}) + (q_{32} \wedge r_{22})\} = -0.5$; Since $h_2(q_{32}) < 0$, $\therefore h_2(q_{32}) = 0$

$q_{33} = r_{33} = 0.2$; $h_2(q_{33}) = q_{33} - \{(q_{33} \wedge r_{31}) + (q_{33} \wedge r_{32})\} = -0.2$; Since $h_2(q_{33}) < 0$, $\therefore h_2(q_{33}) = 0$

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 arbitarily select k = 3 and return $q_{33} = 0$.

Evaluate q_{ij} for all j except k = 3.

$q_{31} = 0$; $q_{32} = 0$.

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

Example 1

$$\text{Given } R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} 0.9 & 0.4 & 0.3 \\ 1.0 & 0.8 & 0.5 \\ 0.2 & 1.0 & 0.2 \end{bmatrix}.$$

Trace of the Algorithm II to evaluate Q using $h_2(\cdot)$ is given in Table II.

$$\therefore \quad Q = \begin{bmatrix} 0.9 & 0.3 & 0.3 \\ 0.2 & 0.2 & 1.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix}$$

which is optimal.

It can easily be verified that time complexity of Algorithm II is also of $O(n^2)$.

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 $Q \, o \, R = I$ 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 $q_{ij}|_{j \neq k} = \bigwedge_{l=1}^{n} r_{kl}$. 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_{ik} may be zero when $q_{ik} \geq q_{ij}$, for all j, and $q_{ij}|_{j \neq k} = 0$ 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 $h_2(.) \geq 0$ for at least one q_{ij} (where $j = 1$ to m) in q_i. However, $h_2(q_{ij}) \geq 0$ yields the following:

$$\text{Or, } h_2(q_{ij}) = r_{ji} - \sum_{\substack{l=1 \\ l \neq i}}^{n} (r_{ji} \wedge r_{jl}) \geq 0$$

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

$$\text{Or, } r_{ji} \geq \sum_{\substack{l=1 \\ l \neq i}}^{n} (r_{ji} \wedge r_{jl}). \tag{28}$$

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

$$r_{ji} \geq \sum_{\substack{l=1 \\ l \neq i}}^{n} (r_{jl}). \tag{29}$$

Since for a given column i

$$r_{ji} \geq \sum_{\substack{l=1 \\ l \neq i}}^{n} (r_{jl})$$

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.

Given:	If x is A Then y is B
Given:	y is B′
Find:	x is A′

The procedure for evaluating the membership of the inference: x is A′}, is well known as GMT [18], [30], [35].

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

speed (m/s) \ height (feet)		5	6	7
	7	0.9	0.6	0.3
R(x, y) =	8	1.0	0.8	0.5
	10	1.0	1.0	0.8

Fig. 1. Fuzzy relational matrix constructed by Lukasiewicz implication function.

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

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

$$A' = B'\,o\,R^{-1}. \tag{30}$$

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) *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

$$\begin{aligned} B' &= \neg B \\ &= [\,(1 - 0.2) \quad (1 - 0.4) \quad (1 - 0.7)\,] \\ &= [\,0.8 \quad 0.6 \quad 0.3\,]. \end{aligned}$$

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 ¬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 $R^{-1} = R^{\mathrm{T}}$, as used in GMT [17].

VI. Conclusion

This paper presents a new formulation of computing inverse fuzzy relation by determining the inherent constraints

TABLE III
PERFORMANCE ANALYSIS

Method used for evaluating Q, pre-inverse to R	Evaluation of A′ = B′ o Q, where B′=[0.8 0.6 0.3]	Error = ‖ A′ − ¬ A ‖ $= [\sum_{i=1}^{n} (A_i' - \neg A_i)^2]^{1/2}$, where, ¬ A = [0.7 0.4 0.1]
Q = R^{-1} evaluated by Algrithm I (proposed method). $Q = \begin{bmatrix} 0.9 & 0.3 & 0.3 \\ 0.6 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 \end{bmatrix}$	A′ = [0.8 0.3 0.3]	Error = $[(0.8\text{-}0.7)^2+(0.3\text{-}0.4)^2+(0.3\text{-}0.1)^2]^{1/2}$ **= 0.245**
Q = R^{-1} evaluated by Algorithm II (Optimal solution). $Q = \begin{bmatrix} 0.9 & 0.3 & 0.3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	A′ = [0.8 0.3 0.3]	Error = $[(0.8\text{-}0.7)^2+(0.3\text{-}0.4)^2+(0.3\text{-}0.1)^2]^{1/2}$ **= 0.245**
Q = R^T = Transpose of R (classical [17]). $Q = \begin{bmatrix} 0.9 & 1.0 & 1.0 \\ 0.6 & 0.8 & 1.0 \\ 0.3 & 0.5 & 0.8 \end{bmatrix}$	A′ = [0.8 0.8 0.8]	Error = $[(0.8\text{-}0.7)^2+(0.8\text{-}0.4)^2+(0.8\text{-}0.1)^2]^{1/2}$ **= 0.812**
Q = R^{-1} by the method introduced in [37]. $Q = \begin{bmatrix} 0.9 & 1.0 & 0.8 \\ 0.3 & 0.8 & 0.8 \\ 0.3 & 0.5 & 0.8 \end{bmatrix}$	A′ = [0.8 0.8 0.8]	Error = $[(0.8\text{-}0.7)^2+(0.8\text{-}0.4)^2+(0.8\text{-}0.1)^2]^{1/2}$ **= 0.812**

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 $h_1(\cdot)$, 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 with inverse computation by Algorithm I exhibits superiority 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} includes 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 each row of R is greater than or equal to the sum of the remaining elements in the same row, and the largest elements of 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 fizzy 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_{ik}. 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_{ik}.

Theorem 9: Maximization of the heuristic function

$$h(q_{ij}) = (q_{ij} \wedge r_{ji}) - \bigvee_{\substack{l=1 \\ l \neq i}}^{n} (q_{ij} \wedge r_{jl})$$

to obtain q_{ij}, does not ensure minimization of all the $\min$ terms $q_{ij} \wedge r_{jl}$, $l = 1$ to n, $l \neq i$.

Proof: Maximization of $h(q_{ij})$, indicates maximization of $(q_{ij} \wedge r_{ji})$ and minimization of $\vee_{\substack{l=1 \\ l \neq i}}^{n} (q_{ij} \wedge r_{jl})$ for $j = 1$ to m. However

$$\bigvee_{\substack{l=1 \\ l \neq i}}^{n} (q_{ij} \wedge r_{jl}) = (q_{ij} \wedge r_{jl'}) \vee \left(\bigvee_{\substack{l=1 \\ l \neq i,\, l'}}^{n} (q_{ij} \wedge r_{jl}) \right)$$

when $(q_{ij} \wedge r_{jl'}) \geq (q_{ij} \wedge r_{jl})$ for $l = 1$ to n, $l \neq i$, $\bigvee_{\substack{l=1 \\ l \neq i}}^{n} (q_{ij} \wedge r_{jl})$ reduces to $(q_{ij} \wedge r_{jl'})$.

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

Theorem 10: Maximization of the heuristic functions, $h(q_{ij}) = (q_{ij} \wedge r_{ji}) - \bigvee_{\substack{l=1 \\ l \neq i}}^{n} (q_{ij} \wedge r_{jl})$, $j = 1$ to m, to obtain

q_{ij}, attempting to satisfy $[q_i\, o\, R]_i \approx 1$ and $[q_i\, o\, R]_{l,l\neq i} \approx 0$; ultimately sets the worst case (largest) value of

$$[q_i\, o\, R]_l = \bigvee_{j=1}^{m} (r_{ji} \wedge r_{jl}), \qquad l = 1 \text{ to } n \text{ and } l \neq i.$$

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

$$\begin{aligned}[q_i\, o\, R]_{l,l\neq i} &= \bigvee_{j=1}^{m} (q_{ij} \wedge r_{jl}), \qquad l = 1 \text{ to } n \text{ and } l \neq i \\ &= \bigvee_{j=1}^{m} (r_{ji} \wedge r_{jl}), \qquad l = 1 \text{ to } n \text{ and } l \neq i\end{aligned}$$

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 $q_{ik} = q_{ix}$, denoted by $D_i|_{q_{ix}}$, is smaller than the error norm induced by $q_{ik} = q_{iy}$, denoted by $D_i|_{q_{iy}}$, where index k is the position of the largest element in q_i.

Proof: From (18), we have $h_1(q_{ik}) = q_{ik} - (1/(n-1)) \sum_{l=1, l\neq i}^{n} (q_{ik} \wedge r_{kl})$.

Given $h_1(q_{ix}) = h_1(q_{iy})$, so we have

$$\begin{aligned} & q_{ix} - \frac{1}{(n-1)} \sum_{\substack{l=1\\l\neq i}}^{n} (q_{ix} \wedge r_{xl}) = q_{iy} - \frac{1}{(n-1)} \sum_{\substack{l=1\\l\neq i}}^{n} (q_{iy} \wedge r_{yl}) \\ \text{or, } & (n-1)q_{ix} - \sum_{\substack{l=1\\l\neq i}}^{n} (q_{ix} \wedge r_{xl}) = (n-1)q_{iy} - \sum_{\substack{l=1\\l\neq i}}^{n} (q_{iy} \wedge r_{yl}) \\ \text{or, } & (n-2)q_{ix} + q_{ix} - \sum_{\substack{l=1\\l\neq i}}^{n} (q_{ix} \wedge r_{xl}) \\ & = (n-2)q_{iy} + q_{iy} - \sum_{\substack{l=1\\l\neq i}}^{n} (q_{iy} \wedge r_{yl}) \\ \text{or, } & (n-2)q_{ix} + \left\{ (1-q_{iy}) + \sum_{\substack{l=1\\l\neq i}}^{n} (q_{iy} \wedge r_{yl}) \right\} \\ & = (n-2)q_{iy} + \left\{ (1-q_{ix}) + \sum_{\substack{l=1\\l\neq i}}^{n} (q_{ix} \wedge r_{xl}) \right\} \\ \text{or, } & (n-2)q_{ix} + D_i|_{q_{iy}} = (n-2)q_{iy} + D_i|_{q_{ix}} \\ \text{or, } & D_i|_{q_{iy}} - D_i|_{q_{ix}} = (n-2)\{q_{iy} - q_{ix}\}. \end{aligned} \tag{31}$$

Now, from (31), it is obvious that, as $q_{ix} < q_{iy}$, thus $D_i|_{q_{ix}} < D_i|_{q_{iy}}$, i.e., error will be less when we will assign q_{ik} 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 $D_i|_{q_{ix}}$, is equal to the error norm induced by $q_{ik} = q_{iy}$, denoted by $D_i|_{q_{iy}}$, 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 $D_i|_{q_{ix}} = D_i|_{q_{iy}}$ [using (24)]. Therefore, even if $q_{ix} < q_{iy}$, the error norm $D_i|_{q_{ix}} = D_i|_{q_{iy}}$. Hence, the theorem is proved. ■

REFERENCES

[1] T. Arnould and S. Tano, "Interval-valued fuzzy backward reasoning," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 4, pp. 425–437, Nov. 1995.

[2] T. Arnould and S. Tano, "The inverse problem of the aggregation of fuzzy sets," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 2, no. 4, pp. 391–415, 1994.

[3] T. Arnould, S. Tano, Y. Kato, and T. Miyoshi, "Backward chaining with fuzzy 'if ... then ... rules'," in *Proc. 2nd IEEE Int. Conf. Fuzzy Syst.*, San Francisco, CA, 1993, pp. 533–548.

[4] E. A. Bender, *Mathematical Methods in Artificial Intelligence*. Silver Spring, MD: IEEE Comput. Soc. Press, 1996.

[5] M. M. Bourke and D. G. Fisher, "Solution algorithms for fuzzy relational equations with max-product composition," *Fuzzy Sets Syst.*, vol. 94, no. 1, pp. 61–69, Feb. 1998.

[6] J. Cen, "Fuzzy matrix partial ordering and generalized inverses," *Fuzzy Sets Syst.*, vol. 105, no. 3, pp. 453–458, Aug. 1999.

[7] A. Chakraborty, A. Konar, U. K. Chakraborty, and A. Chatterjee, "Emotion recognition from facial expressions and its control using fuzzy logic," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 4, pp. 726–743, Jul. 2009.

[8] L. Cheng-Zhong, "Generalized inverses of fuzzy matrix," in *Approximate Reasoning in Decision Analysis*. Amsterdam, The Netherlands: North Holland, 1982, pp. 57–60.

[9] E. Czogala, J. Drewniak, and W. Pedrycz, "Fuzzy relation equations on a finite set," *Fuzzy Sets Syst.*, vol. 7, no. 1, pp. 89–101, Jan. 1982.

[10] A. Di Nola and S. Sessa, "On the set of solutions of composite fuzzy relation equations," *Fuzzy Sets Syst.*, vol. 9, no. 1–3, pp. 275–285, 1983.

[11] J. Drewniak, "Fuzzy relation inequalities," *Cybern. Syst.*, vol. 88, pp. 677–684, 1988.

[12] J. Drewniak, "Fuzzy relation equations and inequalities," *Fuzzy Sets Syst.*, vol. 14, no. 3, pp. 237–247, Dec. 1984.

[13] S. Gottwald, "Generalized solvability criteria for fuzzy equations," *Fuzzy Sets Syst.*, vol. 17, no. 3, pp. 285–296, Dec. 1985.

[14] S. Gottwald and W. Pedrycz, "Solvability of fuzzy relational equations and manipulation of fuzzy data," *Fuzzy Sets Syst.*, vol. 18, no. 1, pp. 45–65, Jan. 1986.

[15] M. Higashi and G. J. Klir, "Resolution of finite fuzzy relation equations," *Fuzzy Sets Syst.*, vol. 13, no. 1, pp. 65–82, May 1984.

[16] K. H. Kim and F. W. Roush, "Generalized fuzzy matrices," *Fuzzy Sets Syst.*, vol. 4, no. 3, pp. 293–315, Nov. 1980.

[17] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty, and Information*. Englewood Cliffs, NJ: Prentice-Hall, 1988, ch. 1.

[18] A. Konar, *Computational Intelligence: Principles, Techniques and Applications*. Berlin, Germany: Springer-Verlag, 2005.

[19] B. Kosko, *Fuzzy Engineering*. Englewood Cliffs, NJ: Prentice-Hall, 1997.

[20] M. Krysander, J. Aslund, and M. Nyberg, "An efficient algorithm for finding minimal over constrained subsystems for model-based diagnosis," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 1, pp. 197–217, Jan. 2009.

[21] J. Leotamonphong and S.-C. Fang, "An efficient solution procedure for fuzzy relational equations with max-product composition," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 4, pp. 441–445, Aug. 1999.

[22] A. Lettieri and F. Liguori, "Characterization of some fuzzy relation equations provided with one solution in a finite set," *Fuzzy Sets Syst.*, vol. 13, no. 1, pp. 83–94, May 1984.

[23] X. Li and D. Ruan, "Novel neural algorithms based on fuzzy δ rules for solving fuzzy relational equations: Part III," *Fuzzy Sets Syst.*, vol. 109, no. 3, pp. 355–362, Feb. 2000.

[24] L. Luoh, W.-J. Wang, and Y.-K. Liaw, "Matrix-pattern-based computer algorithm for solving fuzzy relational equations," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 1, pp. 100–108, Feb. 2003.

[25] M. Miyakoshi and M. Shimbo, "Solutions of composite fuzzy relational equations with triangular norms," *Fuzzy Sets Syst.*, vol. 16, no. 1, pp. 53–63, May 1985.

[26] C. P. Pappis, "Multi-input multi-output fuzzy systems and the inverse problem," *Eur. J. Oper. Res.*, vol. 28, no. 2, pp. 228–230, Feb. 1987.

[27] C. P. Pappis and G. I. Adamopoulos, "A computer algorithm for the solution of the inverse problem of fuzzy systems," *Fuzzy Sets Syst.*, vol. 39, no. 3, pp. 279–290, Feb. 1991.

[28] C. P. Pappis and M. Sugeno, "Fuzzy relational equation and the inverse problem," *Fuzzy Sets Syst.*, vol. 15, no. 1, pp. 79–90, Feb. 1985.

[29] W. Pedrycz, "Inverse problem in fuzzy relational equations," *Fuzzy Sets Syst.*, vol. 36, no. 2, pp. 277–291, Jun. 1990.

[30] W. Pedrycz, "Fuzzy set calculus," in *Handbook of Fuzzy Computation*, E. H. Ruspini, P. P. Bonissone, and W. Pedrycz, Eds. Bristol, U.K.: IOP Publ., 1998.

[31] W. Pedrycz, "Fuzzy relational equations with generalized connectives and their applications," *Fuzzy Sets Syst.*, vol. 10, no. 1–3, pp. 185–201, 1983.

[32] I. Perfilieva and A. Tonis, "Compatibility of systems of fuzzy relation equations," *Int. J. Gen. Syst.*, vol. 29, no. 4, pp. 511–528, 2000.

[33] M. Prevot, "Algorithm for the solution of fuzzy relations," *Fuzzy Sets Syst.*, vol. 5, no. 3, pp. 319–322, May 1981.

[34] E. Rich and K. Knight, *Artificial Intelligence*. New York: McGraw-Hill, 1991.

[35] T. J. Ross, *Fuzzy Logic With Engineering Applications*. New York: McGraw-Hill, 1995.

[36] P. Saha, "Abductive reasoning with inverse fuzzy discrete relations," Ph.D. dissertation, Jadavpur Univ., Calcutta, India, 2002.

[37] P. Saha and A. Konar, "A heuristic algorithm for computing the max–min inverse fuzzy relation," *Int. J. Approx. Reason.*, vol. 30, no. 3, pp. 131–147, Sep. 2002.

[38] E. Sanchez, "Resolution of composite fuzzy relation equations," *Inf. Control*, vol. 30, no. 1, pp. 38–48, 1976.

[39] E. Sanchez, "Solution of fuzzy equations with extended operations," *Fuzzy Sets Syst.*, vol. 12, no. 3, pp. 237–247, Apr. 1984.

[40] S. Sessa, "Some results in the setting of fuzzy relation equation theory," *Fuzzy Sets Syst.*, vol. 14, no. 3, pp. 281–297, Dec. 1984.

[41] S. Singh, A. Kodali, K. Choi, K. R. Pattipati, S. M. Namburu, S. C. Sean, D. V. Prokhorov, and L. Qiao, "Dynamic multiple fault diagnosis: Mathematical formulations and solution techniques," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 1, pp. 160–176, Jan. 2009.

[42] M. Togai, "Application of fuzzy inverse relation to synthesis of a fuzzy controller for dynamic systems," in *Proc. 23rd Conf. Decision Control*, Las Vegas, NV, Dec. 1984, pp. 904–905.

[43] S. Wang, S.-C. Fang, and H. L. W. Nuttle, "Solution set of interval-valued fuzzy relational equation," *Fuzzy Optim. Decis. Making*, vol. 2, no. 1, pp. 41–60, Mar. 2003.

[44] Y.-K. Wu and S.-M. Guu, "An efficient procedure for solving a fuzzy relational equation with Max-Archimedean t-norm composition," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 1, pp. 73–84, Feb. 2008.

[45] C.-T. Yeh, "On the minimal solutions of max–min fuzzy relational equations," *Fuzzy Sets Syst.*, vol. 159, no. 1, pp. 23–39, Jan. 2008.

[46] L. A. Zadeh, "Knowledge representation in fuzzy logic," *IEEE Trans. Knowl. Data Eng.*, vol. 1, no. 1, pp. 89–100, Mar. 1988.

[47] H. J. Zimmerman, *Fuzzy Set Theory and Its Applications*. Dordrecht, The Netherlands: Kluwer, 1996.

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.

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 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 journal 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.

Lakhmi 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.