An Efficient Algorithm to Computing Max-Min Post-inverse Fuzzy Relation for Abductive Reasoning

Sumantra Chakraborty[1], Amit Konar[1], and Ramadoss Janarthanan[2]

[1] Artificial Intelligence Laboratory,
Dept. of Electronics and Tele-Communication Engineering,
Jadavpur University, Calcutta- 32, India
[2] Department of IT, Jaya Engg. College, Chennai, India
suman_ch54@rediffmail.com, konaramit@yahoo.co.in,
srmjana_73@yahoo.com

Abstract. This paper provides an alternative formulation to computing the max-min post-inverse fuzzy relation by minimizing a heuristic (objective) function to satisfy the inherent constraints of the problem. An algorithm for computing the max-min post-inverse fuzzy relation as well as the trace of the algorithm is proposed here. The algorithm exposes its relatively better computational accuracy and higher speed in comparison to the existing technique for post-inverse computation. The betterment of computational accuracy of the max-min post-inverse fuzzy relation leads more accurate result of fuzzy abductive reasoning, because, max-min post-inverse fuzzy relation is required for abductive reasoning.

Keywords: Max–min inverse fuzzy relation, Heuristic Function, Abductive reasoning.

1 Introduction

A fuzzy relation R(x, y) usually describes a mapping from universe X to universe Y (i.e. X →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).

1.1 Fuzzy Max-Min Post-inverse Relation

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

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 505–519, 2011.

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

where x ∈ X, y ∈ Y and z ∈ Z.

For brevity we would use '∧' and '∨' to denote min and max respectively. Thus,

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

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_2 as the post-inverse to R_1. It is easy to note that when $R_1 = R_2$ = I, $R_1 \circ R_2$ = I follows. However, when $R_1 \neq$ I, we cannot find any R_2 that satisfies $R_1 \circ R_2$ = I. It is apparent from the last statements that we can only evaluate approximate max-min post-inverse to R_1, when we know R_1.

1.2 Review

The origin of the proposed max-min fuzzy inverse computation problem dates back to the middle of 1970's, 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 [36]. The work was later studied and extended by Prevot [31], Czogala, Drewniak and Pedrycz [8], Lettieri and Liguori [20], Luoh et al. [22], and Yeh [41] for finite fuzzy sets [16]. Cheng-Zhong [7] and Wang et al. [40] proposed two distinct approaches to computing intervals of solutions for each element of an inverse fuzzy relation. Higashi and Klir in [14] 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 [10], [12], [13], [15], [29], [43], [44], [30], [21] need special mention. The early approaches mentioned above unfortunately is not directly applicable to find general solutions R_2, satisfying the relational equation: $R_1 \circ R_2$ = I. Interestingly, there are problems like fuzzy backward/abductive [17] reasoning, where $R_1 \neq$ I, but R_2 needs to be evaluated. This demands a formulation to determine a suitable R_2, such that the given relational equation is *best satisfied*.

Several direct (or indirect) formulations of the *max-min* pre-inverse computing problem have been addressed in the literature [2], [6], [7], [24], [25], [26], [27], [34], [35], [37], [45]. A first attempt to compute fuzzy pre-inverse with an aim to satisfy all the underlying constraints in the relational equation using a heuristic objective function is addressed in [35]. 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. All the limitations in [35] are fully resolved by [45].

This paper is an extended part of [45], where the proposed work illustrates an alternative formulation to computing the max-min post-inverse fuzzy relation.

The rest of the paper is organized as follows. In section 2, we provide *Strategies* used to solve the post-inverse computational problem. The algorithm is presented in section 3. The analysis of algorithm and example is given section 3.1 and 3.2 respectively.

2 Proposed Computational and Approach to Fuzzy Max-Min Post-inverse 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 R o Q = I′ ≈ I, where I denotes identity matrix of dimension ($m \times m$). Let q_j be the j^{th} column of Q matrix. The following strategies have been adopted to solve the equation R o Q = I′ ≈ I for known R.

Strategy 1: Decomposition of R o Q ≈ I into $[R \text{ o } q_j]_j \approx 1$ *and* $[R \text{ o } q_j]_{l, l \neq j} \approx 0$.

Since, R o Q ≈ I, R o $q_j \approx j^{th}$ column of I matrix, therefore, the j^{th} element of R o q_j, denoted by $[R \text{ o } q_j]_j \approx 1$ and the l^{th} element (where $l \neq j$) of R o q_j, denoted by $[R \text{ o } q_j]_{l, l \neq j} \approx 0$.

Strategy 2: Determination of the effective range of q_{ij}, $\forall i$ *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 $[R \text{ o } q_j]_j \approx 1$, the range of q_{ij}, $\forall i$ virtually becomes $[0, r_{ji}]$ by Lemma 1.

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

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

Proof: $[R \text{ o } q_j]_j$

$$= \bigvee_{i=1}^{n} (r_{ji} \wedge q_{ij}) \tag{4}$$

Since $[\bigvee_{i=1}^{n} (r_{ji} \wedge q_{ij})]_{q_{ij} > r_{ji}} = [\bigvee_{i=1}^{n} (r_{ji} \wedge q_{ij})]_{q_{ij} = r_{ji}}$, The minimum value of q_{ij} that maximizes [R o q_j $]_j$ towards 1 is r_{ji}. Setting q_{ij} beyond r_{ji} is of no use in connection with maximization of [R o q_j $]_j$ towards 1. Therefore, the effective range of q_{ij} reduces from [0, 1] to [0, r_{ji}]. □

Strategy 3: Replacement of the constraint [R o q_j $]_j \approx 1$, by $q_{kj} \approx 1$, where $q_{kj} \geq q_{ij}$, $\forall i$

We first prove [R o q_j $]_j = q_{kj}$ for $q_{kj} \geq q_{ij}$, $\forall i$ by Lemma 2, and then argue that [R o q_j $]_j \approx 1$ can be replaced by $q_{kj} \approx 1$.

Lemma 2: If $q_{kj} \geq q_{ij}$, $\forall i$, then [R o q_j $]_j = q_{kj}$.

Proof: $$[R \circ q_j]_j = \bigvee_{i=1}^{n} (r_{ji} \wedge q_{ij}) \tag{5}$$

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

Therefore, $$(r_{ji} \wedge q_{ij}) = q_{ij} \tag{6}$$

Substituting expression (6) in expression (5), yields the resulting expression as

$$[R \circ q_j]_j = \bigvee_{i=1}^{n} (q_{ij}) \tag{7}$$

$$= q_{kj} \quad \text{as } q_{kj} \geq q_{ij}, \forall i. \tag{8}$$

□

The maximization of [R o q_j $]_j$, therefore, depends only on q_{kj}, and the maximum value of [R o q_j $]_j = q_{kj}$. Consequently, the constraint [R o q_j $]_j \approx 1$ is replaced by $q_{kj} \approx 1$. 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_{kj} and other element q_{ij} (for $i \neq k$) in q_j, the j^{th} column of Q-matrix, by separate procedures. For evaluation of q_{kj}, we first need to identify the positional index k of q_{kj} so that maximization of [R o q_j $]_j$ and minimization of [R o q_j $]_{l, l \neq j}$ occur jointly for a suitable selection of q_{kj}. This is taken care of in Strategy 5 and 6. In Strategy 5, we determine k for the possible largest element q_{kj}, whereas in Strategy 6 we evaluate q_{kj}. To determine q_{ij} (for $i \neq k$), we only need to minimize [R o q_j]$_{l,l \neq j}$. This is considered in Strategy 4. It is indeed important to note that selection of q_{ij} (for $i \neq k$) to minimize [R o q_j]$_{l,l \neq j}$ does not hamper maximization of [R o q_j]$_j$ as [R o q_j]$_j = q_{ik}$, vide Lemma 2.

Strategy 4: Evaluation of q_{ij}*, $i \neq k$, where* $q_{kj} \geq q_{ij}$*, $\forall i$.*

The details of the above strategy are taken up in Theorem 1.

Theorem 1: If $q_{kj} \geq q_{ij}$*, $\forall i$, then the largest value of* $q_{ij}\Big|_{i \neq k}$ *that minimizes [R o* q_j *]*$_{l,l \neq j}$ *towards 0 is given by* $(\bigwedge_{\substack{l=1 \\ l \neq j}}^{m} r_{lk}) \wedge q_{kj}$.

Proof: [R o q_j]$_{l,l \neq j}$

$$= \bigvee_{i=1}^{n} (r_{li} \wedge q_{ij}), \quad \forall\, l,\, l \neq j \tag{9}$$

$$= \bigvee_{\substack{i=1 \\ i \neq k}}^{n} (r_{li} \wedge q_{ij}) \vee (r_{lk} \wedge q_{kj}), \quad \forall\, l,\, l \neq j \tag{10}$$

$$= (r_{lk} \wedge q_{kj}) \quad \forall\, l,\, l \neq j,$$

$$\text{if } (r_{lk} \wedge q_{kj}) \geq \bigvee_{\substack{i=1 \\ i \neq k}}^{n} (r_{li} \wedge q_{ij}). \tag{11}$$

Therefore,

$$Min\ [\mathrm{R\ o}\ q_j\ ]_{l,l \neq j} = \underset{\forall l, l \neq j}{Min} (r_{lk} \wedge q_{kj})$$

$$= \underset{\forall l, l \neq j}{Min} \{ r_{lk} \} \wedge q_{kj}$$

$$= (\bigwedge_{\substack{l=1 \\ l \neq j}}^{m} r_{lk}) \wedge q_{kj} \tag{12}$$

Since $Min[R \circ q_j]_{l,l\neq j} = (\bigwedge_{l=1, l\neq j}^{m} r_{lk}) \wedge q_{kj}$ and the largest value in $[R \circ q_j]_{l,l\neq j} = (r_{lk} \wedge q_{kj})$, therefore, $Min[R \circ q_j]_{l,l\neq j}$, will be the largest among $(r_{li} \wedge q_{ij}) \forall i, i \neq k$, if

$$Min[R \circ q_j]_{l,l\neq j} = (\bigwedge_{\substack{l=1\\ l\neq j}}^{m} r_{lk}) \wedge q_{kj}$$

$$\geq \bigvee_{\substack{i=1\\ i\neq k}}^{n} (r_{li} \wedge q_{ij}), \tag{13}$$

which is same as, $(\bigwedge_{l=1, l\neq j}^{m} r_{lk}) \wedge q_{kj} \geq (r_{li} \wedge q_{ij}) \quad \forall i, i \neq k.$ (14)

The *largest value of* q_{ij} for $i \neq k$ can be obtained by setting equality in equation (14), and the resulting equality condition is satisfied when

$$q_{ij}\Big|_{i\neq k} = (\bigwedge_{\substack{l=1\\ l\neq j}}^{m} r_{lk}) \wedge q_{kj}\,. \tag{15}$$

□

Strategy 5: Determining the positional index k for the element q_{kj} *(* $\geq q_{ij}, \forall j$ *) in* q_j.

To determine the position k of q_{kj} in q_j, we first need to construct a heuristic function $h(q_{kj})$ that satisfies two constraints:

i) Maximize $[R \circ q_j]_j$ (16)

ii) Minimize $[R \circ q_j]_{l,l\neq j}$, (17)

and then determine the index k, such that $h(q_{kj}) \geq h(q_{ij})$, $\forall i$. In other words, we need to determine the positional index k for the possible largest element q_{kj} in the j^{th}

column of Q-matrix, as the largest value of $h(q_{kj})$ ensures maximization of the heuristic function $h(q_{kj})$, and thus best satisfies the constraints (16) and (17).

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

One simple heuristic cost function that satisfies (16) and (17) is

$$h_1(q_{kj}) = q_{kj} - \frac{1}{(m-1)} \sum_{\substack{l=1 \\ l \neq j}}^{m} (r_{lk} \wedge q_{kj}),$$

where $q_{kj} \geq q_{ij}$, $\forall i$, by Theorem 2. Next we find k such that

$$\underset{q_{kj} \in [0, r_{jk}]}{Max} h_1(q_{kj}) \geq \underset{q_{ij} \in [0, r_{ji}]}{Max} h_1(q_{ij}), \text{ for all } i.$$

Theorem 2: If $q_{kj} \geq q_{ij}$, $\forall i$, then maximization of $[R \text{ o } q_j]_j$ and minimization of $[R \text{ o } q_j]_{l, l \neq j}$ can be represented by a heuristic function,

$$h_1(q_{kj}) = q_{kj} - \frac{1}{(m-1)} \sum_{\substack{l=1 \\ l \neq j}}^{m} (r_{lk} \wedge q_{kj}).$$

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

$$[R \text{ o } q_j]_j = q_{kj} \tag{18}$$

Further, $[R \text{ o } q_j]_{l, l \neq j}$

$$= (r_{l1} \wedge q_{1j}) \vee (r_{l2} \wedge q_{2j}) \vee \ldots \vee (r_{lk} \wedge q_{kj}) \vee \ldots \vee (r_{ln} \wedge q_{nj}), \text{ for } \forall l,\ l \neq j. \tag{19}$$

Now by Theorem 1 we have $q_{ij}\Big|_{i \neq k} = (\bigwedge_{\substack{l=1 \\ l \neq j}}^{m} r_{lk}) \wedge q_{kj}$ and substituting this value in equation (19) we have

$$[R \text{ o } q_j]_{l, l \neq j} = (r_{lk} \wedge q_{kj}), \text{ for } \forall l,\ l \neq j \tag{20}$$

Now to jointly satisfy maximization of $[R \text{ o } q_j]_j$ and minimization of $[R \text{ o } q_j]_{l, l \neq j}$ we design a heuristic function,

$h_1(q_{kj}) = [\mathrm{R} \circ q_j]_j - \mathrm{f}((r_{1k} \wedge q_{kj}),\ldots,(r_{j-1,k} \wedge q_{kj}),(r_{j+1,k} \wedge q_{kj}),\ldots\ldots,$
$(r_{mk} \wedge q_{kj}))$.

Now, for any monotonically non-decreasing function f(·), maximization of $[\mathrm{R} \circ q_j]_j$ and minimization of the *min* terms $(r_{1k} \wedge q_{kj})$, ..., $(r_{j-1,k} \wedge q_{kj})$, $(r_{j+1,k} \wedge q_{kj})$, ..., $(r_{mk} \wedge q_{kj})$ calls for maximization of $h_1(q_{kj})$. Since averaging (Avg) is a monotonically increasing function, we replace f(·) by Avg(·). Thus,

$h_1(q_{kj}) = [\mathrm{R} \circ q_j]_j - \mathrm{Avg}((r_{1k} \wedge q_{kj}),\ldots,(r_{j-1,k} \wedge q_{kj}),(r_{j+1,k} \wedge q_{kj}),$
$\ldots\ldots,(r_{mk} \wedge q_{kj}))$

$$= q_{kj} - \frac{1}{(m-1)} \sum_{\substack{l=1 \\ l \neq j}}^{m} (r_{lk} \wedge q_{kj}). \tag{21}$$

Although apparent, it may be added for the sake of completeness that $n>1$ in (21). □

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

$$\underset{q_{kj} \in [0, r_{jk}]}{Max} h_1(q_{kj}) \geq \underset{q_{ij} \in [0, r_{ji}]}{Max} h_1(q_{ij}), \text{ for all } i \tag{22}$$

Consequently, determination of k satisfying the inequality (22) yields the largest element q_{kj} in the q_j, that maximizes the heuristic function $h_1(.)$.

Corollary 1: $h_2(q_{kj}) = q_{kj} - \sum_{\substack{l=1 \\ l \neq j}}^{m} (r_{lk} \wedge q_{kj})$*, too is a heuristic function that maximizes* $[R \circ q_j]_j$ *and minimizes* $[R \circ q_j]_{l, l \neq j}$.

Proof: Since $\sum_{\substack{l=1 \\ l \neq j}}^{m} (r_{lk} \wedge q_{kj})$ is a monotonically non-decreasing 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 non-decreasing function of q_{ij} by Theorem 3. Then we can easily verify that for q_{ij} in [0, r_{ji}], $h_1(q_{ij})\Big|_{q_{ij}=r_{ji}}$ is the largest, i.e., $h_1(q_{ij})\Big|_{q_{ij}=r_{ji}} \geq h_1(q_{ij})\Big|_{q_{ij}\in[0,r_{ji}]}$.

Theorem 3: $h_1(q_{ij}) = q_{ij} - \dfrac{1}{(m-1)} \sum_{\substack{l=1 \\ l\neq j}}^{m} (r_{li} \wedge q_{ij})$, *is a monotonically non-decreasing function of q_{ij} in [0, r_{ji}].*

Proof: We consider two possible cases:

Case 1: If $q_{ij} > r_{li}$ $\forall$ *l*, $l \neq j$, then,

$$h_1'(q_{ij}) = \frac{dh_1(q_{ij})}{dq_{ij}} = 1 - \frac{1}{(m-1)} \sum_{\substack{l=1 \\ l\neq i}}^{m} \frac{d(r_{li})}{dq_{ij}} = 1\ (>0). \qquad \text{(since, } \frac{d(r_{li})}{dq_{ij}} = 0\text{)}.$$

Case 2: Let $q_{ij} \leq r_{li}$ for at least one *l* (say *t* times), then,

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

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

$$= 1 - \frac{t}{(m-1)} \geq 0 \quad \text{as } t \leq (m-1).$$

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

Strategy 7: Finding Q-matrix

Evaluation of each row q_j of Q-matrix is considered independently. For given row q_j, we need to determine the largest element $q_{kj} \geq q_{ij}$, $\forall i$. After q_{kj} is evaluated, we evaluate q_{ij}, $i \neq k$ in the subsequent phase.

Since maximization of $h_1(\cdot)$ ensures satisfaction of the constraints (16) and (17) in strategy 5, to determine q_{kj}, we look for an index *k*, such that

$$\underset{q_{kj}\in[0,r_{jk}]}{Max}\; h_1(q_{kj}) \geq \underset{q_{ij}\in[0,r_{ji}]}{Max}\; h_1(q_{ij}), \text{ for all } i.$$

Further, since $h_1(\cdot)$ is a monotonically non-decreasing function, above inequality reduces to,

$$h_1(q_{kj})\Big|_{q_{kj}=r_{jk}} \geq h_1(q_{ij})\Big|_{q_{ij}=r_{ji}} \tag{23}$$

If a suitable value of k is found satisfying (23), we say q_{ik} is the largest element in q_i and the value of $q_{kj} = r_{jk}$.

For other element q_{ij} in q_j, $i \neq k$, we evaluate

$$q_{ij}\Big|_{i\neq k} = (\bigwedge_{\substack{l=1\\ l\neq j}}^{m} r_{lk}) \wedge q_{kj}$$

$$= (\bigwedge_{\substack{l=1\\ l\neq j}}^{m} r_{lk}) \wedge r_{jk} \qquad \text{(Since, } q_{kj} = r_{jk}\text{)}$$

$$= \bigwedge_{l=1}^{m} r_{lk}\,. \tag{24}$$

The principle for evaluation of q_{ij} for a given j can be repeated for j = 1 to m to determine Q-matrix.

3 Proposed Fuzzy Max-Min Post-inverse Computation Algorithm

The results obtained from the strategies in Section 2 are used here to construct Algorithm for Max-Min Post-inverse computation for a given fuzzy relational matrix.

ALGORITHM 1

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 RoQ is close enough to I//

Begin

For j = 1 to m

```
            Evaluate q_kj (.);        //Determine the position k and value of the largest
                                      //element q_kj in column j of Q.//
            For i = 1 to n
                If( i ≠ k )
                    Then q_ij = Min_l { r_lk };
                              //Determine all the elements in the j^th column of Q matrix
                              //except q_kj //
                End If;
            End For;
        End For;
    End.
```

Evaluate q_{kj} (.)

Begin

For i = 1 to n

$$h_1(q_{ij}) = r_{ji} - \frac{1}{(m-1)} \sum_{\substack{l=1 \\ l \neq j}}^{m} (r_{li} \wedge r_{ji}); \quad \text{//Evaluate } h_1(q_{ij}).\text{//}$$

End For

If $h_1(q_{kj}) \geq h_1(q_{ij})$ for all i

Then return k and $q_{kj} = r_{jk}$; //Return the position k of q_{kj} , and its value//

End.

3.1 Explanation of Algorithm I

Algorithm 1 evaluates the elements in q_j , i.e., q_{1j}, q_{2j},……., q_{mj} in a single pass by determining the position k of the largest element q_{kj} in the j[th] column and then its value r_{jk} . Next, we determine the other elements in q_j , which is given by

$$q_{ij}\Big|_{i \neq k} = \bigwedge_{l=1}^{m} r_{lk}\,.$$

The outer **For**-loop in the algorithm sets j = 1 to m with an increment in j by 1, and evaluation of q_j takes place for each setting of j. The most important step inside this outer **For**-loop is determining positional index k of the largest element q_{kj} in q_j and evaluation of its value. This has been taken care of in function Evaluate q_{kj} (.).

Table 1. Trace of the algorithm

$j=1$

Evaluate q_{kj} (j)

When $q_{11} = r_{11} = 0.9$; $h(q_{11}) = r_{11} - \{(r_{21} \wedge r_{11}) + (r_{31} \wedge r_{11})\}/2 = 0.8$

When $q_{21} = r_{12} = 0.4$; $h(q_{21}) = r_{12} - \{(r_{22} \wedge r_{12}) + (r_{32} \wedge r_{12})\}/2 = 0.05$

When $q_{31} = r_{13} = 0.3$; $h(q_{31}) = r_{13} - \{(r_{23} \wedge r_{13}) + (r_{33} \wedge r_{13})\}/2 = 0.05$

Max{ $h(q_{11})$, $h(q_{21})$, $h(q_{31})$ }= $h(q_{11})$; Return $q_{11}=0.9$ and k=1

Evaluate the largest q_{ij} for all i except k =1

$q_{21} = \text{Min}(r_{11}, r_{21}, r_{31}) = 0.1$; $q_{31} = \text{Min}(r_{11}, r_{21}, r_{31}) = 0.1$;

$j=2$

Evaluate_ q_{kj} (j)

When $q_{12} = r_{21} = 0.1$; $h(q_{12}) = r_{21} - \{(r_{11} \wedge r_{21}) + (r_{31} \wedge r_{21})\}/2 = 0.0$

When $q_{22} = r_{22} = 0.3$; $h(q_{22}) = r_{22} - \{(r_{12} \wedge r_{22}) + (r_{32} \wedge r_{22})\}/2 = 0.0$

When $q_{32} = r_{23} = 0.8$; $h(q_{32}) = r_{23} - \{(r_{13} \wedge r_{23}) + (r_{33} \wedge r_{23})\}/2 = 0.55$

Max{ $h(q_{12})$, $h(q_{22})$, $h(q_{32})$ }= $h(q_{32})$; Return $q_{32} = 0.8$ and k=3

Evaluate the largest q_{ij} for all i except k = 3

$q_{12} = \text{Min}(r_{13}, r_{23}, r_{33}) = 0.2$; $q_{22} = \text{Min}(r_{13}, r_{23}, r_{33}) = 0.2$;

$j=3$

Evaluate_ q_{kj} (j)

When $q_{13} = r_{31} = 0.1$; $h(q_{13}) = r_{31} - \{(r_{11} \wedge r_{31}) + (r_{21} \wedge r_{31})\}/2 = 0.0$

When $q_{23} = r_{32} = 1.0$; $h(q_{23}) = r_{32} - \{(r_{12} \wedge r_{32}) + (r_{22} \wedge r_{32})\}/2 = 0.65$

When $q_{33} = r_{33} = 0.2$; $h(q_{33}) = r_{33} - \{(r_{13} \wedge r_{33}) + (r_{23} \wedge r_{33})\}/2 = 0.0$

Max{ $h(q_{13})$, $h(q_{23})$, $h(q_{33})$ }= $h(q_{23})$; Return $q_{23} = 1.0$ and k=2

Evaluate the largest q_{ij} for all i except k = 2

$q_{13} = \text{Min}(r_{12}, r_{22}, r_{32}) = 0.3$; $q_{33} = \text{Min}(r_{12}, r_{22}, r_{32}) = 0.3$;

3.2 Example

Given $R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} 0.9 & 0.4 & 0.3 \\ 0.1 & 0.3 & 0.8 \\ 0.1 & 1.0 & 0.2 \end{bmatrix}$, we now provide a trace of the Algorithm in Table-1.

$$\therefore \quad Q = \begin{bmatrix} 0.9 & 0.2 & 0.3 \\ 0.1 & 0.2 & 1.0 \\ 0.1 & 0.8 & 0.3 \end{bmatrix}.$$

References

1. Arnould, T., Tano, S.: Interval-valued fuzzy backward reasoning. IEEE Trans. Fuzzy Systems 3(4), 425–437 (1995)
2. Arnould, T., Tano, S.: The inverse problem of the aggregation of fuzzy sets. Int. J. Uncertainity Fuzzyness Knowledge Based System 2(4), 391–415 (1994)
3. Arnould, T., Tano, S., Kato, Y., Miyoshi, T.: Backward chaining with fuzzy "if..then...rules". In: Proc. of the Second IEEE International Conference on Fuzzy Systems, San Francisco, CA, pp. 548–533 (1993)
4. Bender, E.A.: Mathematical Methods in Artificial Intelligence. IEEE Computer Society Press Silver Spring, MD (1996)
5. Bourke, M.M., Fisher, D.G.: Solution algorithms for fuzzy relational equations with max-product composition. Fuzzy Sets and Systems 94, 61–69 (1998)
6. Cen, J.: Fuzzy matrix partial ordering and generalized inverses. Fuzzy Sets and Systems 105, 453–458 (1999)
7. Cheng-Zhong, L.: Generalized inverses of fuzzy matrix. Approximate Reasoning in Decis. Anal., 57–60 (1982)
8. Czogala, E., Drewniak, J., Pedrycz, W.: Fuzzy relation equations on a finite set. Fuzzy Sets and Systems 12, 89–101 (1982)
9. Di Nola, A., Sessa, S.: On the set of solutions of composite fuzzy relation equations. Fuzzy Sets and Systems 9, 275–285 (1983)
10. Drewniak, J.: Fuzzy relation inequalities. Cybernet. Syst. 88, 677–684 (1988)
11. Drewniak, J.: Fuzzy relation equations and inequalities. Fuzzy Sets and Systems 14, 237–247 (1984)
12. Gottwald, S.: Generalized solvability criteria for fuzzy equations. Fuzzy Sets and Systems 17, 285–296 (1985)
13. Gottwald, S., Pedrycz, W.: Solvability of fuzzy relational equations and manipulation of fuzzy data. Fuzzy Sets and Systems 18, 45–65 (1986)
14. Higashi, M., Klir, G.J.: Resolution of finite fuzzy relation equations. Fuzzy Sets and Systems 13, 65–82 (1984)
15. Kim, K.H., Roush, F.W.: Generalized fuzzy matrices. Fuzzy Sets and Systems 4, 293–315 (1980)
16. Klir, G.J., Folger, T.A.: Fuzzy Sets, Uncertainity, and Information, ch. 1. Prentice-Hall, Englewood Cliffs (1988)

17. Konar, A.: Computational Intelligence: Principles, Techniques and Applications. Springer, Heidelberg (2005)
18. Kosko, B.: Fuzzy Engineering. Prentice-Hall, Englewood Cliffs (1997)
19. Leotamonphong, J., Fang, S.-C.: An efficient solution procedure for fuzzy relational equations with max-product composition. IEEE Trans. Fuzzy Systems 7(4), 441–445 (1999)
20. Lettieri, A., Liguori, F.: Characterization of some fuzzy relation equations provided with one solution in a finite set. Fuzzy Sets and Systems 13, 83–94 (1984)
21. Li, X., Ruan, D.: Novel neural algorithms based on fuzzy δ -rules for solving fuzzy relational equations, part III. Fuzzy Sets and Systems 109, 355–362 (2000)
22. Luoh, L., Wang, W.-J., Liaw, Y.-K.: Matrix-pattern-based computer algorithm for solving fuzzy relational equations. IEEE Trans. on Fuzzy Systems 11(1) (February 2003)
23. Miyakoshi, M., Shimbo, M.: Solutions of composite fuzzy relational equations with triangular norms. Fuzzy Sets and Systems 16, 53–63 (1985)
24. Pappis, C.P.: Multi-input multi-output fuzzy systems and the inverse problem. European J. of Operational Research 28, 228–230 (1987)
25. Pappis, C.P., Adamopoulos, G.I.: A computer algorithm for the solution of the inverse problem of fuzzy systems. Fuzzy Sets and Systems 39, 279–290 (1991)
26. Pappis, C.P., Sugeno, M.: Fuzzy relational equation and the inverse problem. Fuzzy Sets and Systems 15(1), 79–90 (1985)
27. Pedrycz, W.: Inverse problem in fuzzy relational equations. Fuzzy Sests and Systems 36, 277–291 (1990)
28. Pedrycz, W.: Fuzzy Set Calculus. In: Ruspini, E.H., Bonissone, P.P., Pedrycz, W. (eds.) Handbook of Fuzzy Computation. IOP Publishing, Bristol (1998)
29. Pedrycz, W.: Fuzzy relational equations with generalized connectives and their applications. Fuzzy Sets and Systems 10, 185–201 (1983)
30. Perfilieva, I., Tonis, A.: Compatibility of systems of fuzzy relation equations. Int. J. General Systems 29(4), 511–528 (2000)
31. Prevot, M.: Algorithm for the solution of fuzzy relations. Fuzzy Sets and Systems 5, 319–322 (1981)
32. Rich, E., Knight, K.: Artificial Intelligence. McGraw-Hill, New York (1991)
33. Ross, T.J.: Fuzzy Logic with Engg. Applications. McGraw-Hill, New York (1995)
34. Saha, P.: Abductive Reasoning with Inverse fuzzy discrete Relations, Ph. D. Dissertation, Jadavpur University, India (2002)
35. Saha, P., Konar, A.: A heuristic algorithm for computing the max-min inverse fuzzy relation. Int. J. of Approximate Reasoning 30, 131–147 (2002)
36. Sanchez, E.: Resolution of composite fuzzy relation equations. Inf. Control 30, 38–48 (1976)
37. Sanchez, E.: Solution of fuzzy equations with extended operations. Fuzzy Sets and Systems 12, 237–247 (1984)
38. Sessa, S.: Some results in the setting of fuzzy relation equation theory. Fuzzy Sets and Systems 14, 281–297 (1984)
39. Togai, M.: Application of fuzzy inverse relation to synthesis of a fuzzy controller for dynamic systems. In: Proc. 23rd Conf. on Decision and Control, Las Vegs, NV (December 1984)
40. Wang, S., Fang, S.-C., Nuttle, H.L.W.: Solution set of interval valued fuzzy relational equation. Fuzzy Optim. Decision Making 2(1), 41–60 (2003)

41. Wu, Y.-K., Guu, S.-M.: An efficient procedure for solving a fuzzy relational equation with Max-Archimedean t-norm composition. IEEE Trans. on Fuzzy Systems 16(1) (February 2008)
42. Yeh, C.-T.: On the minimal solutions of max-min fuzzy relational equations. Fuzzy Sets and Systems 159(1) (January 2008)
43. Zadeh, L.A.: Knowledge representation in fuzzy logic. IEEE Trans. Knowledge Data Engg. (1988)
44. Zimmerman, H.J.: Fuzzy Set Theory and its Applications. Kluwer Academic Publishers, Dordrecht (1996)
45. Chakraborty, S., Konar, A., Jain, L.C.: An Efficient Algorithm to Computing Max-Min Inverse Fuzzy Relation for Abductive Reasoning. IEEE Trans. Sys. Man Cybernet (A) 40(1) (January 2010)