ELSEVIER International Journal of Approximate Reasoning 30 (2002) 131–147

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

www.elsevier.com/locate/ijar

A heuristic algorithm for computing the max–min inverse fuzzy relation

Parbati Saha [a,*], Amit Konar [b]

[a] *Department of Mathematics, B.E. College (Deemed University), Howrah 711103, India*
[b] *E.T.C.E. Department, Jadavpur University, Calcutta 700032, India*

Received 1 July 2000; received in revised form 1 July 2001; accepted 1 March 2002

Abstract

The paper addresses a classical problem of computing approximate max–min inverse fuzzy relation. It is an NP-complete problem for which no polynomial time algorithm is known till this date. The paper employs a heuristic function to reduce the search space for finding the solution of the problem. The time-complexity of the proposed algorithm is $O(n^3)$, compared to $O(k^n)$, which is required for an exhaustive search in the real space of $[0, 1]$ at k regular intervals of interval length $(1/k)$.

Keywords: Algorithms; Max–min inverse fuzzy relation; Heuristics

1. Introduction

Let X and $Y \subseteq$ r be two universal sets. A fuzzy relation that describes a mapping from X to Y $(X \to Y)$ generally is a fuzzy subset of $X \times Y$, where '$\times$' denotes a cartesian product [18]. Formally, a fuzzy relation $\mathbf{R}$ is defined by

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

where $\mu_{\mathbf{R}}(x, y)$ refers to the membership of (x, y) to belong to the fuzzy relation $\mathbf{R}(x, y)$. Fuzzy 'composition' [8] is an operation, by which fuzzy relations in different product space can be combined with each other. There exist different

* Corresponding author.
E-mail address: parbati_saha@hotmail.com (P. Saha).

PII: S0888-613X(02)00069-5

versions of 'composition'. The 'max–min' composition, which is most popular among them is defined below. Let $X, Y, Z \subseteq$ r be three universal sets and $R_1(x,y), (x,y) \in X \times Y$ and $R_2(y,z), (y,z) \in Y \times Z$ be two fuzzy relations. The max–min composition of $\mathbf{R}_1$ and $\mathbf{R}_2$, denoted by $\mathbf{R}_1 \circ \mathbf{R}_2$ is then a fuzzy set, is defined by

$$\mathbf{R}_1 \circ \mathbf{R}_2 = \left\{ (x,z), \max_y \{\min\{\mu_{\mathbf{R}_1}(x,y), \mu_{\mathbf{R}_2}(y,z)\}\} \right\}, \tag{2}$$

where $x \in X,\ y \in Y$ and $z \in Z$. For brevity, we shall use '$\wedge$' and '$\vee$ ' to denote 'min' and 'max' operators, respectively. Thus expression (2) can be re-written as

$$\mathbf{R}_1 \circ \mathbf{R}_2 = \left\{ (x,z), \bigvee_y \{\mu_{\mathbf{R}_1}(x,y) \wedge \mu_{\mathbf{R}_2}(y,z)\} \right\}. \tag{3}$$

We use $\mu_{\mathbf{R}_1 \circ \mathbf{R}_2}(x,z)$ to denote the membership function of (x,z) in the max–min composition relation $\mathbf{R}_1 \circ \mathbf{R}_2$ is defined by

$$\mu_{\mathbf{R}_1 \circ \mathbf{R}_2}(x,z) = \bigvee_y \{\mu_{\mathbf{R}_1}(x,y) \wedge \mu_{\mathbf{R}_2}(y,z)\} \tag{4}$$

1.1. Fuzzy max–min inverse relations

Let $X = \{x_1, x_2, \ldots, x_n\},\ Y = \{y_1, y_2, \ldots, y_m\}$ and $Z = \{z_1, z_2, \ldots, z_l\}$ be three universal sets and $\mathbf{R}_1, \mathbf{R}_2$ be two fuzzy relations on $X \times Y$ and $Y \times Z$, respectively. Again, let $\mathbf{R}_1 \circ \mathbf{R}_2 = \mathbf{I}$, where $\mathbf{I}$ denotes an identity relation, such that $\mu_{\mathbf{R}_1 \circ \mathbf{R}_2}(x,z) = \mathbf{I},$ when $x = x_i \in X$ and $z = z_i \in Z$ and $\mu_{\mathbf{R}_1 \circ \mathbf{R}_2}(x,z) = 0,$ otherwise. Under this circumstances, we call R_1, *the max–min pre-inverse relation to* $\mathbf{R}_2$ and $\mathbf{R}_2$, *the max–min post-inverse relation to* $\mathbf{R}_1$. Unfortunately, $\mathbf{R}_1 \circ \mathbf{R}_2 = \mathbf{I}$ is true, only when $\mathbf{R}_1 = \mathbf{R}_2 = \mathbf{I}$. We thus define $\mathbf{R}_1$ as the *approximate max–min pre-inverse relation to* $\mathbf{R}_2$, when $\mathbf{R}_1 \circ \mathbf{R}_2 = \mathbf{I}'$, such that $\mathbf{I}'$ is sufficiently close to $\mathbf{I}$ with respect to a Euclidean norm of the difference $(\mathbf{I} - \mathbf{I}')$, estimated by

$$D = \left[\sum_{\forall z} \sum_{\forall x} \{\mu_{\mathrm{I}}(x,z) - \mu_{\mathrm{I}'}(x,z)\}^2 \right]^{1/2},$$

where D should not exceed a small pre-defined real number. The definition of *approximate post-inverse relation* to $\mathbf{R}_1$ may also be given analogously.

1.2. Best approximate pre-inverse relation

Let $\mathbf{Q}$ be a set of fuzzy relations of $\mathbf{R}_1$, such that for all $\mathbf{R}_1 \in \mathbf{Q}$, there exists an $\mathbf{R}_2$ with $\mathbf{R}_1 \circ \mathbf{R}_2 = \mathbf{I}'$ and

$$\text{Distance} = \left[\sum_{\forall z}\sum_{\forall x}\{\mu_{\mathbf{I}}(x,z) - \mu_{\mathbf{I}'}(x,z)\}^2\right]^{1/2} \leqslant D. \tag{5}$$

Let for $\mathbf{R}_1 = \mathbf{Q}_i \in \mathbf{Q}$, the distance $D_i \leqslant D$ and is the smallest among all possible distances, computed for $\mathbf{Q}_i \in \mathbf{Q}$. Then $\mathbf{Q}_i$ is called the best *approximate pre-inverse relation to* $\mathbf{R}_2$. *Analogously*, we can define the best *approximate post-inverse relation to* $\mathbf{R}_1$.

2. A heuristic approach to determine the inverse fuzzy relation

The paper aims at estimating $\mathbf{R}_1$ when $\mathbf{R}_1 \circ \mathbf{R}_2 = \mathbf{I}'$ and the measure of distance between $\mathbf{I}$ and $\mathbf{I}'$ is $\leqslant D$. Given a relation $\mathbf{R}_2$, we can find $\mathbf{R}_1$ by a random search of $\mu_{\mathbf{R}_1}(x,y)$ $\forall x$, $\forall y$ in the interval $[0,1]$ that jointly satisfy the distance criterion, mentioned above. This, however, requires a massive search in the interval $[0,1]$. For instance, if $\mathbf{R}_1$ has $n(=9)$ elements and we pick up elements from $[0,1]$ at k regular intervals of interval length $= 1/k$ $(= 0.1$ say), then to search the n $(= 9)$ elements we require to evaluate distances $(k+1)^n \approx \mathrm{O}(k^n) = 10^9$ times and then select the relation $\mathbf{R}_1 = \mathbf{Q}_i$, as *the best approximate pre-inverse relation*. In fact, finding the pre-inverse is an *NP-complete problem* [4], for which no 'polynomial time' algorithm is known till this date. The fuzzy max–min relational equation has been solved by a number of researchers [1–3,5–7,10,11,14–16] in different ways. However, the problem presented in the paper is new and cannot be solved by any of the existing approaches.

An alternative approach to solve the problem is to employ a *heuristic function* which is expected to yield good results in most cases but is not guaranteed to yield the best pre-inverse [12]. To construct the heuristic function, let us denote the kth row and ith column of $(\mathbf{Q} \circ \mathbf{R}_2)$ by $(\mathbf{Q} \circ \mathbf{R}_2)_{k,i}$, where

$$(\mathbf{Q} \circ \mathbf{R}_2)_{k,i} = \bigvee_{j=1}^{n} (q_{kj} \wedge r_{ji}), \tag{6}$$

where q_{kj} and r_{ji} are the (k,j)th and (j,i)th elements of the relational matrices $\mathbf{Q}$ and $\mathbf{R}_2$, respectively. Also r_{ji} (given), $q_{kj} \in [0,1]$ $\forall i,j,k$.

For estimating $\mathbf{Q}$, to satisfy $\mathbf{Q} \circ \mathbf{R}_2 = \mathbf{I}'$, we require

$$\bigvee_{j=1}^{n} (q_{kj} \wedge r_{jk}) \quad \text{to be close to 1}$$

and

$$\bigvee_{\substack{j=1\\ i\neq k}}^{n} (q_{kj} \wedge r_{ji}) \quad \text{to be close to 0.}$$

Now, $\bigvee_{j=1}^{n}(q_{kj} \wedge r_{jk})$ will be close to 1, if we keep each element $(q_{k1} \wedge r_{1k})$, $(q_{k2} \wedge r_{2k}), \ldots, (q_{kn} \wedge r_{nk})$ individually close to 1, vide Lemma 1 presented in Appendix A. On the other hand $\bigvee_{\substack{j=1\\ i\neq k}}^{n}(q_{kj} \wedge r_{ji})$ will be close to 0, if $(q_{k1} \wedge r_{1i}), (q_{k2} \wedge r_{2i}), \ldots, (q_{k,i-1} \wedge r_{i-1,i}), (q_{k,i+1} \wedge r_{i+1,i}), \ldots, (q_{kn} \wedge r_{ni})$ all are individually set close to zero, vide Lemma 2 presented in Appendix A.

Further it will be proved, without any loss of generality, vide Lemma 3 presented in Appendix A that q_{kj} may be selected from $\{r_{j1}, r_{j2}, \ldots, r_{jk}, \ldots, r_{jn}\}$ instead of the entire interval $[0, 1]$. This is a significant issue as it causes a reduction in search space. Thus summarisingly, $q_{kj}\ \forall k, j$ is chosen from $\{r_{j1}, r_{j2}, \ldots, r_{jn}\}$ by using the following two criteria:
(i) $(q_{kj} \wedge r_{jk})$ is to be maximised and
(ii) $(q_{kj} \wedge r_{ji})[i \neq k]$ is to be minimised.

The last two criteria can be combined to a single criterion as formulated below:

$$\left[(q_{kj} \wedge r_{jk}) - \bigvee_{\substack{i=1\\ i\neq k}}^{n} (q_{kj} \wedge r_{ji})\right] \quad \text{is to be maximised.}$$

The above function is called a heuristic, which is expected to yield good q_{kj}, given the $r_{jk}\ \forall j, k$ in the interval $[0, 1]$.

3. A heuristic algorithm for computing the fuzzy pre-inverse relation

The heuristic function defined in Section 2, will be employed to compute **Q**, the pre-inverse matrices to $\mathbf{R}_2$. Since the number of possible values of q_{kj} is unpredictable, prior to execution of the algorithm, declaring large static storage for them at the beginning of the program is not suggestive. Rather, a dynamic allocation of memory by pointers is preferred. We thus define the node n_w (in Fig. 1(a)) and the linked list structure for the overall computation (in Fig. 1(b)), using the node definition of Fig. 1(a). The field definitions of the structure used to define node n_w in Fig. 1(a) is clearly explained and therefore is not elaborated further in the text.

The algorithm for estimating **Q**, the pre-inverse to $\mathbf{R}_2$, is now presented below. Let us for the time being assume $\mathbf{R}_2$ to be of dimension $(n \times n)$.

Procedure fuzzy pre-inversion (Q, R$_2$)
begin
node $\leftarrow$ n$_1$;

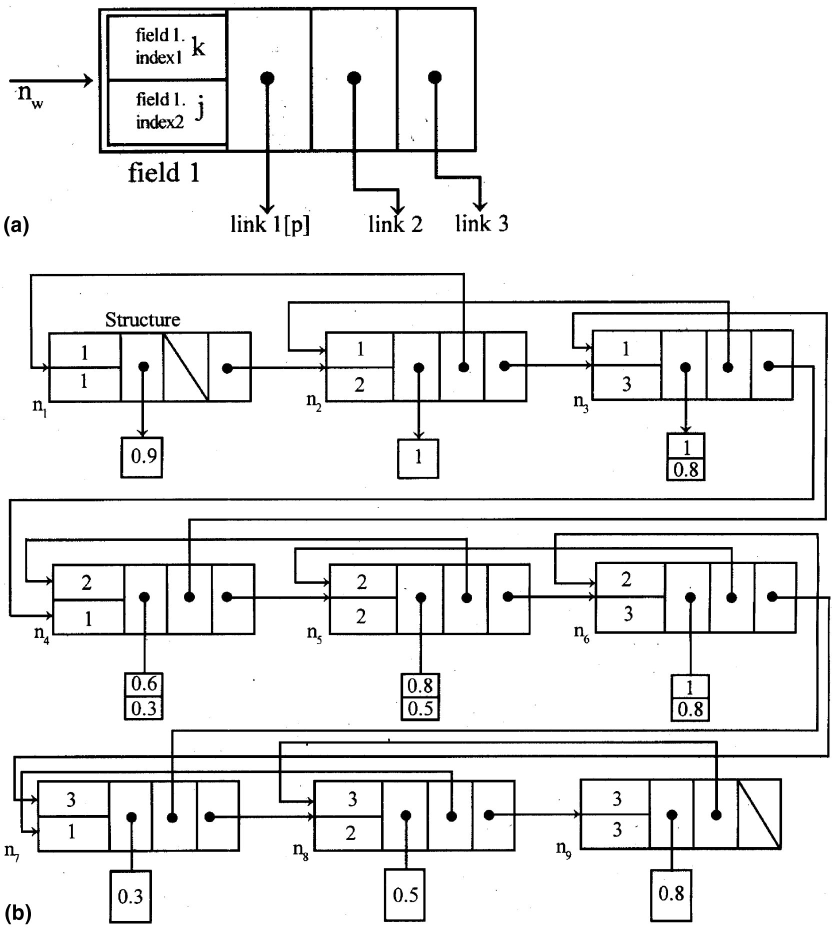

Fig. 1. (a) The field definition of the data structure, wheren n_w denotes the address of a node; field 1.index 1 and field 1.index 2 are integers, denoting the position of an element in a matrix, link 1[p] represents a one dimensional array with index p; link 2 (link 3) points to the predecessor (successor) node in the linked list structure. (b) The linked structure used for the evaluation of the inverse of matrix **R** (of Example 1) by employing nodes like (a).

repeat the following steps

Step 1: $k \leftarrow$ node . field 1 . index 1;

$j \leftarrow$ node . field 1 . index 2;

for $w = 1$ to n

$$\alpha[w] \leftarrow r[j,w] \wedge r[j,k] - \bigvee_{\substack{i=1 \\ i \neq k}}^{n} (r[j,w] \wedge r[j,i])$$

end for;

Step 2: set $\beta \leftarrow$ max . of α [w], $1 \leqslant \forall\, w \leqslant n$;

```
            p ← 1;
            for w = 1 to n
                  if β = α [w]
                    then do
                    begin
                    node . link 1 [p] ← r[j, w];
                    p ← p + 1 ;
                  end if;
            end for;
     Step 3: node ← node . link 3; // for next node //
    until node . link 3 = nil
  end.
```

Time complexity: Since for finding each element of **Q** we require to compute $\alpha[w]$ n-times, we require a search time $O(n)$ in array $\alpha[w]$. Thus for n^2 elements in the matrix, the search time is $O((n^2)n) = O(n^3)$.

The above algorithm preserves the connectivity among the nodes and explains the method to compute and store the values of q_{kj} $\forall k, j$ in the data structure. The statements for returning the q_{kj}s are intensionally omitted to keep the algorithm brief.

In the above algorithm, we did not intensionally restrict the valuation space of D to determine the entire set of **Q** matrices. It may further be noted that the above method for computing pre-inverse for square matrices can be easily extended for evaluating it for non-square relational matrices, vide Property 4 in Section 4.

After the set $\mathbf{Q} = \{\mathbf{Q}_k\}$ is evaluated, we can determine $\mathbf{Q}_{\text{best}}$, the best among them by determining the Euclidean distance $\|\mathbf{I} - \mathbf{I}'\|$ forall $\mathbf{Q}_k$ and selecting that $\mathbf{Q}_k = \mathbf{Q}_{\text{best}}$ for which the distance is minimum. The procedure for computing $\mathbf{Q}_{\text{best}}$ is presented below. It may, however, be noted that $\mathbf{Q}_{\text{best}}$ does not mean the globally best pre-inverse. We can never guarantee the globally best inverse by a heuristic estimator.

```
  Procedure find-best (Q, Q_best);
  begin
    for k = 1 to number-of-inverses
     for Q_k ∈ Q
        sum ← 0
          D ← I − Q_k ∘ R;
             for all i
                for all j
                   sum ← sum + d_ij^2          //D = [d_ij]//
                end for;
             end for;
             store [k] ← sum;
     end for;
```

end for;
find – min-store [k]; //this procedure determines k = m, where store [k], for $1 \leqslant k \leqslant$ number-of-inverses, is minimum//
$\mathbf{Q}_{\text{best}} \leftarrow \mathbf{Q}_{\text{m}}$;
end.

Example 1. The algorithm for computing fuzzy pre-inverse is applied to $\mathbf{R}_2$, given below, for estimation of $\mathbf{Q}$.

$$\mathbf{R}_2 = [r_{kj}] = \begin{bmatrix} 0.9 & 0.6 & 0.3 \\ 1.0 & 0.8 & 0.5 \\ 1.0 & 1.0 & 0.8 \end{bmatrix}.$$

The trace of the procedure fuzzy-pre-inversion is presented in Table 1. We finally get 16 $\mathbf{Q}_k s$: $\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_{16}$, given below, out of which $\mathbf{Q}_{13}, \mathbf{Q}_{14}, \mathbf{Q}_{15}$ and $\mathbf{Q}_{16}$ have minimum distance measure.

$$\mathbf{Q}_1 = \begin{bmatrix} 0.9 & 1.0 & 1.0 \\ 0.6 & 0.8 & 1.0 \\ 0.3 & 0.5 & 0.8 \end{bmatrix}, \quad \mathbf{Q}_2 = \begin{bmatrix} 0.9 & 1.0 & 1.0 \\ 0.3 & 0.8 & 1.0 \\ 0.3 & 0.5 & 0.8 \end{bmatrix},$$

$$\mathbf{Q}_3 = \begin{bmatrix} 0.9 & 1.0 & 1.0 \\ 0.6 & 0.5 & 1.0 \\ 0.3 & 0.5 & 0.8 \end{bmatrix}, \quad \mathbf{Q}_4 = \begin{bmatrix} 0.9 & 1.0 & 1.0 \\ 0.3 & 0.5 & 1.0 \\ 0.3 & 0.5 & 0.8 \end{bmatrix},$$

$$\mathbf{Q}_5 = \begin{bmatrix} 0.9 & 1.0 & 0.8 \\ 0.6 & 0.8 & 1.0 \\ 0.3 & 0.5 & 0.8 \end{bmatrix}, \quad \mathbf{Q}_6 = \begin{bmatrix} 0.9 & 1.0 & 0.8 \\ 0.3 & 0.8 & 1.0 \\ 0.3 & 0.5 & 0.8 \end{bmatrix},$$

$$\mathbf{Q}_7 = \begin{bmatrix} 0.9 & 1.0 & 0.8 \\ 0.6 & 0.5 & 1.0 \\ 0.3 & 0.5 & 0.8 \end{bmatrix}, \quad \mathbf{Q}_8 = \begin{bmatrix} 0.9 & 1.0 & 0.8 \\ 0.3 & 0.5 & 1.0 \\ 0.3 & 0.5 & 0.8 \end{bmatrix},$$

$$\mathbf{Q}_9 = \begin{bmatrix} 0.9 & 1.0 & 1.0 \\ 0.6 & 0.8 & 0.8 \\ 0.3 & 0.5 & 0.8 \end{bmatrix}, \quad \mathbf{Q}_{10} = \begin{bmatrix} 0.9 & 1.0 & 1.0 \\ 0.3 & 0.8 & 0.8 \\ 0.3 & 0.5 & 0.8 \end{bmatrix},$$

$$\mathbf{Q}_{11} = \begin{bmatrix} 0.9 & 1.0 & 1.0 \\ 0.6 & 0.5 & 0.8 \\ 0.3 & 0.5 & 0.8 \end{bmatrix}, \quad \mathbf{Q}_{12} = \begin{bmatrix} 0.9 & 1.0 & 1.0 \\ 0.3 & 0.5 & 0.8 \\ 0.3 & 0.5 & 0.8 \end{bmatrix},$$

Table 1
Trace of the procedure fuzzy-pre-inversion

Node	k	j	w	$\alpha[w] = r[j,w] \wedge r[j,k] - \bigvee_{\substack{i=1 \\ i \neq k}}^{3} (r[j,w] \wedge r[j,i])$	$\beta = \max_{w} \alpha[w]$	p	node.link $1[p] = r[j,w]$ if $\beta = \alpha[w]$
n_1	1	1	1	$\alpha[1] = 0.9 \wedge 0.9 - \vee (0.9 \wedge 0.6, 0.9 \wedge 0.3) = 0.3$	$\beta = 0.3$	1	node.link $1[1] = r[1,1] = 0.9$
			2	$\alpha[2] = 0.6 \wedge 0.9 - \vee (0.6 \wedge 0.6, 0.6 \wedge 0.3) = 0.0$			
			3	$\alpha[3] = 0.3 \wedge 0.9 - \vee (0.3 \wedge 0.6, 0.3 \wedge 0.3) = 0.0$			
n_2	1	2	1	$\alpha[1] = 1.0 \wedge 1.0 - \vee (1.0 \wedge 0.8, 1.0 \wedge 0.5) = 0.2$	$\beta = 0.2$	1	node.link $1[1] = r[2,1] = 1.0$
			2	$\alpha[2] = 0.8 \wedge 1.0 - \vee (0.8 \wedge 0.8, 0.8 \wedge 0.5) = 0.0$			
			3	$\alpha[3] = 0.5 \wedge 1.0 - \vee (0.5 \wedge 0.8, 0.5 \wedge 0.5) = 0.0$			
n_3	1	3	1	$\alpha[1] = 1.0 \wedge 1.0 - \vee (1.0 \wedge 1.0, 1.0 \wedge 0.8) = 0.0$	$\beta = 0.0$	1	node.link $1[1] = r[3,1] = 1.0$
			2	$\alpha[2] = 1.0 \wedge 1.0 - \vee (1.0 \wedge 1.0, 1.0 \wedge 0.8) = 0.0$		2	node.link $1[2] = r[3,2] = 1.0$
			3	$\alpha[3] = 0.8 \wedge 1.0 - \vee (0.8 \wedge 1.0, 0.8 \wedge 0.8) = 0.0$		3	node.link $1[3] = r[3,3] = 0.8$
n_4	2	1	1	$\alpha[1] = 0.6 \wedge 0.9 - \vee (0.9 \wedge 0.9, 0.3 \wedge 0.9) = -.3$		1	node.link $1[1] = r[1,2] = 0.6$
			2	$\alpha[2] = 0.6 \wedge 0.6 - \vee (0.9 \wedge 0.6, 0.3 \wedge 0.6) = 0.0$	$\beta = 0.0$	2	node.link $1[2] = r[1,3] = 0.3$
			3	$\alpha[3] = 0.6 \wedge 0.3 - \vee (0.9 \wedge 0.3, 0.3 \wedge 0.3) = 0.0$			

n_5	2	2	1	$\alpha[1] = 0.8 \wedge 1.0 - \vee(1.0 \wedge 1.0, 0.5 \wedge 1.0) = -.2$		1	node.link $1[1] = r[2,2] = 0.8$
			2	$\alpha[2] = 0.8 \wedge 0.8 - \vee(1.0 \wedge 0.8, 0.5 \wedge 0.8) = 0.0$	$\beta = 0.0$	2	node.link $1[2] = r[2,3] = 0.5$
			3	$\alpha[3] = 0.8 \wedge 0.5 - \vee(1.0 \wedge 0.5, 0.5 \wedge 0.5) = 0.0$			
n_6	2	3	1	$\alpha[1] = 0.1 \wedge 1.0 - \vee(1.0 \wedge 1.0, 0.8 \wedge 1.0) = 0.0$	$\beta = 0.0$	1	node.link $1[1] = r[3,1] = 1.0$
			2	$\alpha[2] = 1.0 \wedge 1.0 - \vee(1.0 \wedge 1.0, 0.8 \wedge 1.0) = 0.0$		2	node.link $1[2] = r[3,2] = 1.0$
			3	$\alpha[3] = 1.0 \wedge 0.8 - \vee(1.0 \wedge 0.8, 0.8 \wedge 0.8) = 0.0$		3	node.link $1[3] = r[3,3] = 0.8$
n_7	3	1	1	$\alpha[1] = 0.3 \wedge 0.9 - \vee(0.9 \wedge 0.9, 0.6 \wedge 0.9) = -.6$			
			2	$\alpha[2] = 0.3 \wedge 0.6 - \vee(0.9 \wedge 0.6, 0.6 \wedge 0.6) = -.3$			
			3	$\alpha[3] = 0.3 \wedge 0.3 - \vee(0.9 \wedge 0.3, 0.6 \wedge 0.3) = 0.0$	$\beta = 0.0$	1	node.link $1[1] = r[1,3] = 0.3$
n_8	3	2	1	$\alpha[1] = 0.5 \wedge 1.0 - \vee(1.0 \wedge 1.0, 0.8 \wedge 1.0) = -.5$			
			2	$\alpha[2] = 0.5 \wedge 0.8 - \vee(1.0 \wedge 0.8, 0.8 \wedge 0.8) = -.3$			
			3	$\alpha[3] = 0.5 \wedge 0.5 - \vee(1.0 \wedge 0.5, 0.8 \wedge 0.5) = 0.0$	$\beta = 0.0$	1	node.link $1[1] = r[2,3] = 0.5$
n_9	3	3	1	$\alpha[1] = 0.8 \wedge 1.0 - \vee(1.0 \wedge 1.0, 1.0 \wedge 1.0) = -.2$			
			2	$\alpha[2] = 0.8 \wedge 1.0 - \vee(1.0 \wedge 1.0, 1.0 \wedge 1.0) = -.2$			
			3	$\alpha[3] = 0.8 \wedge 0.8 - \vee(1.0 \wedge 0.8, 1.0 \wedge 0.8) = 0.0$	$\beta = 0.0$	1	node.link $1[1] = r[3,3] = 0.8$

$$\mathbf{Q}_{13} = \begin{bmatrix} 0.9 & 1.0 & 0.8 \\ 0.6 & 0.8 & 0.8 \\ 0.3 & 0.5 & 0.8 \end{bmatrix}, \quad \mathbf{Q}_{14} = \begin{bmatrix} 0.9 & 1.0 & 0.8 \\ 0.3 & 0.8 & 0.8 \\ 0.3 & 0.5 & 0.8 \end{bmatrix},$$

$$\mathbf{Q}_{15} = \begin{bmatrix} 0.9 & 1.0 & 0.8 \\ 0.6 & 0.5 & 0.8 \\ 0.3 & 0.5 & 0.8 \end{bmatrix}, \quad \mathbf{Q}_{16} = \begin{bmatrix} 0.9 & 1.0 & 0.8 \\ 0.3 & 0.5 & 0.8 \\ 0.3 & 0.5 & 0.8 \end{bmatrix}.$$

4. Analysis of the algorithm

The following properties have been derived analytically from the algorithm presented in Section 3.

Property 1. *For a given relational matrix R, the inverse matrix with respect to the fuzzy max–min composition operator is not unique, in general.*

Proof. By the proof of Lemma 3, α attains its maximum value when $q_{1j} = r_{j1}$, $q_{nj} = r_{jn}$, $j = 1, 2, \ldots, n$ and $q_{kj} = r_{j1}, r_{j2}, \ldots, r_{jk}$, where $k \neq 1$, $\neq n$; $j = 1, 2, \ldots, n$.

Here, q_{kj} being non-unique, the inverse matrix $\mathbf{Q} = [q_{kj}]$ is not unique, in general. Hence the statement of the property follows. □

Property 2. *$\boldsymbol{R}^{\mathrm{T}}$ is one of the possible inverse fuzzy relational matrices of $\boldsymbol{R}$ with respect to the fuzzy max–min composition operator.*

Proof. By the proof of Property 1, $\alpha_{\max}$ can be obtained by choosing $q_{kj} = r_{jk}$ also. With this choice of q_{kj}, the matrix $\mathbf{Q} = [r_{jk}] = \mathbf{R}^{\mathrm{T}}$. Hence the statement of the property follows. □

Property 3. *For a given relational matrix $\boldsymbol{R}$, the best inverse matrix with respect to the fuzzy max–min composition operator is not unique, in general.*

Proof. Let $\mathbf{Q} = [q_{kj}]$ where $q_{st} = \alpha, 1 \leqslant s, t \leqslant n$ with $0 < \alpha < 1$ and $\alpha > r_{tj}$ for $j = 1, 2, \ldots, n$, be the best fuzzy inverse of R with respect to the fuzzy max–min composition operator and $\mathbf{Q}'$ be a matrix which differs from $\mathbf{Q}$ only in the element lying at the intersection of sth row and tth column, defined as follows:

$\mathbf{Q}' = [q'_{kj}]$, where $q'_{st} = \beta \neq \alpha, 0 < \beta < 1$ and $\beta > r_{ij}$ for $j = 1, 2, \ldots, n$ and $q'_{kj} = q_{kj}$ otherwise. Now since $q_{st} \neq q'_{st}$, the elements of the product matrices $(\mathbf{Q} \circ \mathbf{R})$ and $(\mathbf{Q}' \circ \mathbf{R})$ can differ only in the sth row.

$$(\mathbf{Q} \circ \mathbf{R})_{s,j} = (q_{s1} \wedge r_{1j}) \vee (q_{s2} \wedge r_{2j}) \vee \cdots \vee (\alpha \wedge r_{tj}) \vee \cdots \vee (q_{sn} \wedge r_{nj})$$
$$= (q_{s1} \wedge r_{1j}) \vee (q_{s2} \wedge r_{2j}) \vee \cdots \vee (r_{tj}) \vee \cdots \vee (q_{sn} \wedge r_{nj})$$
$$[\because \alpha > r_{tj} \text{ for } j = 1, 2, \ldots, n],$$

$$(\mathbf{Q}' \circ \mathbf{R})_{s,j} = (q_{s1} \wedge r_{1j}) \vee (q_{s2} \wedge r_{2j}) \vee \cdots \vee (\beta \wedge r_{tj}) \vee \cdots \vee (q_{sn} \wedge r_{nj})$$
$$= (q_{s1} \wedge r_{1j}) \vee (q_{s2} \wedge r_{2j}) \vee \cdots \vee (r_{tj}) \vee \cdots \vee (q_{sn} \wedge r_{nj})$$
$$[\because \beta > r_{tj} \text{ for } j = 1, 2, \ldots, n],$$

$$\therefore (\mathbf{Q} \circ \mathbf{R})_{s,j} = (\mathbf{Q}' \circ \mathbf{R})_{s,j}, \quad j = 1, 2, \ldots, n,$$

$$\therefore (\mathbf{Q} \circ \mathbf{R}) = (\mathbf{Q}' \circ \mathbf{R})$$
$$[\because (\mathbf{Q} \circ \mathbf{R})_{k,j} = (\mathbf{Q}' \circ \mathbf{R})_{k,j} \text{ for } k \neq s;\ j = 1, 2, \ldots, n].$$

Thus there exists another best inverse of **R** viz. $\mathbf{Q}'(\neq \mathbf{Q})$. So **Q** is not unique. Hence the statement of the property follows. □

It is evident from the definition of fuzzy relational matrices that it need not be square matrix. Property 4 presents a clue for inversion of non-square fuzzy relational matrices.

Property 4. *The algorithm for inversion of square matrices is equally applicable to non-square matrices.*

Proof. The algorithm for fuzzy inversion can be applied to non-square matrices if α_w's in the algorithm can be estimated. For a given **R** of dimension $m \times n$, the fuzzy inverse of **R**, denoted by $\mathbf{Q} = [q_{kj}]$, is to be of dimension $n \times m$, where

$$\alpha_w = (q_{kj} \wedge r_{jk}) - \bigvee_{\substack{i=1 \\ i \neq k}}^{n} (q_{kj} \wedge r_{ji}) \quad \text{with } q_{kj} \in \{r_{j1}, r_{j2}, \ldots, r_{jk}, \ldots, r_{jn}\}.$$

Since α_w's, as shown above, can be computed, the statement of the property follows. □

Another interesting result that needs special mention is the structure of the **R** matrix that yields *unique* fuzzy inverse. The structural constraints on **R** matrix that yields unique inverse is presented in Theorem 1.

Theorem 1. *The **R** matrix yields a unique inverse matrix **Q** if all the columns of the matrix **R** are equal to a given arbitrary column vector.*

Proof. By the proof of Property 1, q_{kj} becomes unique when r_{ji}'s are equal for $i = 1, 2, \ldots, n$ i.e., elements lying in the jth row of **R** be equal. Hence, the relational matrix **R** will have unique fuzzy inverse with respect to the fuzzy max–min composition operator, if all the elements lying in the same row of **R** be equal i.e., if all the columns of the matrix **R** is equal to a given arbitrary column vector. Hence the statement of the theorem follows. □

5. Application in abductive reasoning

The proposed algorithm for approximate pre-inversion is useful for solving the well-known fuzzy abductive reasoning problems [9,13,17], presented below.

For instance, consider the fuzzy sets $A \subseteq X$ and $B \subseteq Y$, where X and Y are two universal sets.

Given a rule and the consequent as follows, we want to evaluate the premise.

Given:	If x is **A**, then y is **B**
Given:	y is $\mathbf{B}'$
Find:	x is $\mathbf{A}'$

In the above example, $\mathbf{B}' \subseteq Y$ and $\mathbf{A}' \subseteq X$ and x, y are fuzzy variables such that $x \in X$ and $y \in Y$. The problem of determining $\mathbf{A}'$ for known values of **A, B** and $\mathbf{B}'$ is the primary basis of the fuzzy abductive reasoning problem, also called the Generalised Modus Tollens (GMT).

Let

$$x = \text{height}$$

$$y = \text{speed}$$

be the fuzzy variables satisfying

$$x \in \mathbf{A} = \text{Height-is-Tall} \subseteq X = \text{Fuzzy universal set Height}$$

and

$$y \in \mathbf{B} = \text{Speed-is-High} \subseteq Y = \text{Fuzzy universal set Speed.}$$

With two given instances of **A** and **B** as

$$\mathbf{A} = [0.3 \quad 0.6 \quad 0.9],$$

and

$$\mathbf{B} = [0.2 \quad 0.4 \quad 0.7]$$

we now compute the relational matrix $\mathbf{R}(x, y)$ for the implication $\mathbf{A} \rightarrow \mathbf{B}$ by Lukasiewicz implication function and thus find

$$\mathbf{R}(x, y) = \begin{array}{c|ccc} \text{speed (m/s)} \diagdown \text{height (feet)} & 5 & 6 & 7 \\ \hline 7 & 0.9 & 0.6 & 0.3 \\ 8 & 1.0 & 0.8 & 0.5 \\ 10 & 1.0 & 1.0 & 0.8 \end{array}$$

Then, for abductive reasoning, let the observed distribution $\mathbf{B}' = \mathbf{B}$. Consequently, we find

$$\mathbf{A}' = \mathbf{B}' \circ \mathbf{R}^{-1}.$$

In the present context,

$$\mathbf{R} = \begin{bmatrix} 0.9 & 0.6 & 0.3 \\ 1.0 & 0.8 & 0.5 \\ 1.0 & 1.0 & 0.8 \end{bmatrix}$$

and $\mathbf{R}^{-1} = \mathbf{Q}$ returns 16 matrices of which the following four yields minimum norm. Let the four matrices be denoted by

$$\mathbf{Q}_1 = \begin{bmatrix} 0.9 & 1.0 & 0.8 \\ 0.6 & 0.8 & 0.8 \\ 0.3 & 0.5 & 0.8 \end{bmatrix}, \quad \mathbf{Q}_2 = \begin{bmatrix} 0.9 & 1.0 & 0.8 \\ 0.3 & 0.8 & 0.8 \\ 0.3 & 0.5 & 0.8 \end{bmatrix},$$

$$\mathbf{Q}_3 = \begin{bmatrix} 0.9 & 1.0 & 0.8 \\ 0.6 & 0.5 & 0.8 \\ 0.3 & 0.5 & 0.8 \end{bmatrix}, \quad \mathbf{Q}_4 = \begin{bmatrix} 0.9 & 1.0 & 0.8 \\ 0.3 & 0.5 & 0.8 \\ 0.3 & 0.5 & 0.8 \end{bmatrix}.$$

Let

$$\mathbf{A}'_i = \mathbf{B}' \circ \mathbf{Q}_i, \quad i = 1, 2, 3, 4 = \mathbf{B} \circ \mathbf{Q}_i \text{ [by using } \mathbf{B}' = \mathbf{B}].$$

Thus,

$$\mathbf{A}'_1 = [0.4 \quad 0.5 \quad 0.7],$$

$$\mathbf{A}'_2 = [0.3 \quad 0.5 \quad 0.7],$$

$$\mathbf{A}'_3 = [0.4 \quad 0.5 \quad 0.7],$$

$$\mathbf{A}'_4 = [0.3 \quad 0.5 \quad 0.7].$$

Consequently,

$$\|\mathbf{A}'_1 - \mathbf{A}\| = 0.245,$$

$$\|\mathbf{A}'_2 - \mathbf{A}\| = 0.224,$$

$$\|\mathbf{A}_3' - \mathbf{A}\| = 0.245,$$
$$\|\mathbf{A}_4' - \mathbf{A}\| = 0.224$$

of which $\|\mathbf{A}_k' - \mathbf{A}\|$ for $k = 2, 4$ is the minimum. The principle presented here to illustrate the use of $\mathbf{R}^{-1}$ in abductive reasoning can be easily extended to chained Modus Ponens [4].

6. Conclusions

We defined a heuristic function to determine the approximate fuzzy max–min compositional inverse and employed it in classical abductive reasoning problems. The proposed method is fast as the time complexity of the procedure fuzzy-pre-inverse is $O(n^3)$ compared to exhaustive search which requires to evaluate Euclidean distance $O(k^n)$ times, where we take k discrete values from $[0, 1]$ at a regular interval of $(1/k)$. Thus when $k = 11$ and $n = 3$, we require a computational time of $O(3^3)$ and $O(11^3)$ in the respective two cases and obviously the former is much less than the latter. Further, experimental evidences [12] show that the best matrix we found by our method does not differ much from the globally best inverse.

Acknowledgements

The authors thank Prof. A.K. Nath of ETCE Dept., J.U. for pointing out the topic of research and Prof. Amaljyoti Sengupta, Vice-Chancellor, B.E. College (A Deemed University) and Prof. B. Mukhopadhyay, Head, Dept. of Maths., B.E. College (A Deemed University) for providing a good research environment.

Appendix A

Lemmas 1, 2 and 3 which have been referred in Section 2 are in order:
The lemma requires to prove formally the following inequality:

Lemma 1. *For any square matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ of dimension $n \times n$, we have*

$$\bigvee_{q_{k1}, q_{k2}, \ldots, q_{kn}, \in [0,1]} [(q_{k1} \wedge r_{1k}) \vee (q_{k2} \wedge r_{2k}) \vee \cdots \vee (q_{kj} \wedge r_{jk}) \vee \cdots \vee (q_{kn} \wedge r_{nk})]$$
$$= \left[\bigvee_{q_{k1} \in [0,1]} (q_{k1} \wedge r_{1k})\right] \vee \left[\bigvee_{q_{k2} \in [0,1]} (q_{k2} \wedge r_{2k})\right] \vee \cdots \vee \left[\bigvee_{q_{kj} \in [0,1]} (q_{kj} \wedge r_{jk})\right]$$
$$\vee \cdots \vee \cdots \vee \left[\bigvee_{q_{kn} \in [0,1]} (q_{kn} \wedge r_{nk})\right].$$

Proof

$$\bigvee_{q_{k1},q_{k2},\dots,q_{kn}\in[0,1]} [(q_{k1}\wedge r_{1k})\vee(q_{k2}\wedge r_{2k})\vee\cdots\vee(q_{kj}\wedge r_{jk})\vee\cdots\vee(q_{kn}\wedge r_{nk})]$$

$$= (q'_{k1}\wedge r_{1k})\vee(q'_{k2}\wedge r_{2k})\vee\cdots\vee(q'_{kj}\wedge r_{jk})\vee\cdots\vee(q'_{kn}\wedge r_{nk}),$$

say where $q'_{k1}, q'_{k2}, \dots, q'_{kn}\in[0,1]$

$$= \left[\bigvee_{q_{k1}\in[0,1]}(q_{k1}\wedge r_{1k})\right]\vee\left[\bigvee_{q_{k2}\in[0,1]}(q_{k2}\wedge r_{2k})\right]\vee\cdots\vee\left[\bigvee_{q_{kj}\in[0,1]}(q_{kj}\wedge r_{jk})\right]$$

$$\vee\cdots\vee\left[\bigvee_{q_{kn}\in[0,1]}(q_{kn}\wedge r_{nk})\right]$$

since $\bigvee_{q_{kj}\in[0,1]}(q_{kj}\wedge r_{jk}) = (q'_{kj}\wedge r_{jk})$ for $j=1,2,\dots,n, q'_{kj}\in[0,1]$. Thus the Lemma 1 is valid. □

Lemma 2. *For any square matrices* $\boldsymbol{Q}$ *and* $\boldsymbol{R}$ *of dimension* $n\times n$, *we have*

$$\bigwedge_{q_{k1},q_{k2},\dots,q_{kn}\in[0,1]} [(q_{k1}\wedge r_{1i})\vee(q_{k2}\wedge r_{2i})\vee\cdots\vee(q_{kj}\wedge r_{ji})\vee\cdots\vee(q_{kn}\wedge r_{ni})]$$

$$= \left[\bigwedge_{q_{k1}\in[0,1]}(q_{k1}\wedge r_{1i})\right]\vee\left[\bigwedge_{q_{k2}\in[0,1]}(q_{k2}\wedge r_{2i})\right]\vee\cdots\vee\left[\bigwedge_{q_{kj}\in[0,1]}(q_{kj}\wedge r_{ji})\right]$$

$$\vee\cdots\vee\cdots\vee\left[\bigwedge_{q_{kn}\in[0,1]}(q_{kn}\wedge r_{ni})\right]\quad(i\neq k)$$

Proof

$$\bigwedge_{q_{k1},q_{k2},\dots,q_{kn}\in[0,1]} [(q_{k1}\wedge r_{1i})\vee(q_{k2}\wedge r_{2i})\vee\cdots\vee(q_{kj}\wedge r_{ji})\vee\cdots\vee(q_{kn}\wedge r_{ni})]$$

$$= (q''_{k1}\wedge r_{1i})\vee(q''_{k2}\wedge r_{2i})\vee\cdots\vee(q''_{kj}\wedge r_{ji})\vee\cdots\vee(q''_{kn}\wedge r_{ni}),$$

say where $q''_{k1}, q''_{k2}, \dots q''_{kj}, \dots, q''_{kn}\in[0,1]$,

$$= \left[\bigwedge_{q_{k1}\in[0,1]}(q_{k1}\wedge r_{1i})\right]\vee\left[\bigwedge_{q_{k2}\in[0,1]}(q_{k2}\wedge r_{2i})\right]\vee\cdots\vee\left[\bigwedge_{q_{kj}\in[0,1]}(q_{kj}\wedge r_{ji})\right]$$

$$\vee\cdots\vee\left[\bigwedge_{q_{kn}\in[0,1]}(q_{kn}\wedge r_{ni})\right],$$

since $\bigwedge_{q_{kj}\in[0,1]}(q_{kj}\wedge r_{ji}) = (q''_{kj}\wedge r_{ji})$ for $j=1,2,\dots,n, q''_{kj}$ being $\in[0,1]$. □

Lemma 3. *The selection of* $q_{kj} \in \{r_{j1}, r_{j2}, \ldots, r_{jk}, \ldots, r_{jn}\}$ *instead in the range* $[0, 1]$ *does not violate the determination of inverse fuzzy relational matrix.*

Proof. Let us denote $\{r_{j1}, r_{j2}, \ldots, r_{jk}, \ldots, r_{jn}\}$ and $[0, 1]$ by S and S', respectively, and

$$\alpha = (q_{kj} \wedge r_{jk}) - \bigvee_{\substack{i=1 \\ i \neq k}} (q_{kj} \wedge r_{ji}), \quad j = 1, 2, \ldots, n.$$

Without any loss of generality, we can assume that the elements in S be arranged in the following order:

$$r_{j1} \leqslant r_{j2} \leqslant \cdots \leqslant r_{jk-1} \leqslant r_{jk} \leqslant r_{jk+1} \leqslant \cdots \leqslant r_{jn}.$$

Now, we consider three possible cases.

Case 1: Let $k = 1$. Then

$$\alpha = (q_{1j} \wedge r_{j1}) - \vee[(q_{1j} \wedge r_{j2}), (q_{1j} \wedge r_{j3}), \ldots, (q_{1j} \wedge r_{jn})].$$

Let us denote the maximum value of α by $\alpha_{\max}$. Then $\alpha_{\max} = 0$ when $q_{1j} = r_{j1} \in S$ and also when $q_{1j} \in S' - S$ with $q_{1j} < r_{j1}$.

Case 2: Let $k = n$. Then

$$\alpha = (q_{nj} \wedge r_{jn}) - \vee[(q_{nj} \wedge r_{j1}), (q_{nj} \wedge r_{j2}), \ldots, (q_{nj} \wedge r_{jn-1})].$$

Here $\alpha_{\max} = r_{jn} - r_{jn-1}$, when $q_{nj} = r_{jn} \in S$ and also when $q_{nj} \in S' - S$ with $q_{nj} > r_{jn}$.

Case 3: Let $k \neq 1, \neq n$. We have

$$\alpha = (q_{kj} \wedge r_{jk}) - \vee[(q_{kj} \wedge r_{j1}), (q_{kj} \wedge r_{j2}), \ldots, (q_{kj} \wedge r_{jk-1}), \\ (q_{kj} \wedge r_{jk+1}), \ldots, (q_{kj} \wedge r_{jn})].$$

Here $\alpha_{\max} = 0$ when $q_{kj} \in S$ with $q_{kj} \leqslant r_{jk}$ and also when $q_{kj} \in S' - S$ with $q_{kj} < r_{jk}$.

From the above three cases, it is clear that if one selects q_{kj} from the set $\{r_{j1}, r_{j2}, \ldots, r_{jk}, \ldots, r_{jn}\}$ instead of the set of the numbers in the interval $[0, 1]$, loses nothing, rather saves significant computational time. Hence the statement of the Lemma 3 is valid. □

References

[1] T. Arnould, S. Tano, Interval-valued fuzzy backward reasoning, IEEE Trans. Fuzzy Systems 3 (4) (1995) 425–437.

[2] T. Arnould, S. Tano, Y. Kato, T. Miyoshi, Backward chaining with fuzzy "if...then... rules", in: Proceedings of the Second IEEE International Conference on Fuzzy Systems, San Francisco, CA, 1993, pp. 548–553.

[3] T. Arnould, S. Tano, The inverse problem of the aggregation of fuzzy sets, Int. J. Uncertainty, Fuzziness Knowledge Based Systems 2 (4) (1994) 391–415.
[4] E.A. Bendes, Mathematical Methods in Artificial Intelligence, IEE Computer Society Press, Silver Spring, MD, 1996.
[5] J. Drewniak, Fuzzy relation inequalities, Cybernet Syst. 88 (1988) 677–684.
[6] K.H. Kim, F.W. Roush, Generalized fuzzy matrices, Fuzzy Sets and Systems 4 (1980) 293–315.
[7] G.J. Klir, T.A. Folger, Fuzzy sets, Uncertainty, and Information, Prentice-Hall, Englewood Cliffs, NJ, 1988 (Chapter 1).
[8] B. Kosko, Fuzzy Engineering, Prentice-Hall, Englewood Cliffs, NJ, 1997.
[9] A. Konar, A.K. Mondal, Uncertainty management in expert systems using fuzzy petri nets, IEEE Trans. Knowledge Data Eng. 8 (1) (1996) 96–105.
[10] C.P. Pappis, G.I. Adamopoulos, A computer algorithm for the solution of the inverse problem of fuzzy systems, Fuzzy Sets and Systems 39 (1991) 279–290.
[11] W. Pedrycz, Inverse problem in fuzzy relational equations, Fuzzy Sets and Systems 36 (1990) 277–291.
[12] E. Rich, K. Knight, Artificial Intelligence, McGraw-Hill, New York, 1991.
[13] T.J. Ross, Fuzzy Logic with Engineering Applications, McGraw-Hill, New York, 1995.
[14] P. Saha, Abductive Reasoning with Inverse Fuzzy Discrete Relations, Ph.D. Dissertation, Jadavpur University, India, 2002.
[15] E. Sanchez, Resolution of composite fuzzy relation equations, Inf. Control 30 (1976) 38–48.
[16] E. Sanchez, Solution of fuzzy equations with extended operations, Fuzzy Sets and Systems 12 (1984) 237–247.
[17] L.A. Zadeh, Knowledge representation in fuzzy logic, IEEE Trans. Knowledge Data Eng. 1 (1) (1988).
[18] H.J. Zimmerman, Fuzzy Set Theory and its Applications, Kluwer Academic Publishers, Dordrecht.