Advances in Fuzzy Systems

Fuzzy Functions, Relations, and Fuzzy Transforms (2012)

Guest Editors: Salvatore Sessa, Ferdinando Di Martino, and Irina G. Perfilieva

Fuzzy Functions, Relations, and Fuzzy Transforms (2012)

Fuzzy Functions, Relations, and Fuzzy Transforms (2012)

Guest Editors: Salvatore Sessa, Ferdinando Di Martino, and Irina G. Perfilieva

Editorial Board

Contents

Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2012, Article ID 640951, 2 pages
doi:10.1155/2012/640951

Editorial

Fuzzy Functions, Relations, and Fuzzy Transforms (2012)

Salvatore Sessa,[1] Ferdinando Di Martino,[1] and Irina G. Perfilieva[2]

[1] *Dipartimento di Costruzioni e Metodi Matematici in Architettura, Università degli Studi di Napoli Federico II, Via Monteoliveto 3, 80134 Napoli, Italy*

[2] *Centre of Excellence IT4Innovations, Division University of Ostrava, Institute for Research and Applications of Fuzzy Modeling, 30 dubna 22, 701 00 Ostrava 1, Czech Republic*

Correspondence should be addressed to Salvatore Sessa, sessa@unina.it

Received 25 November 2012; Accepted 25 November 2012

Fuzzy functions, fuzzy relations, and fuzzy transforms are applied to fuzzy systems and like in the first issue published in 2011, here these terms must be intended in general sense. Indeed, the topics concerning fuzzy optimizations with fuzzy functions and relations (including also clustering and PSO algorithms), applications of fuzzy relations/transforms to fusion images, and detection coarse grained fuzzy association rules in the datasets and fuzzy convex processes are dealt with widely in this second issue. The contents of any single paper are resumed in the following sequel:

(i) in the paper of M. Vajgl et al., there is the combination of (at least) two fusion operators; both are based on the F-transform. The first fusion operator is based on a robust partition of the scene domain. The second fusion operator is applied to the residuals of scenes and is based on a finer partition of the same domain. The proposed method can be characterized as a weighted combination of those two and is computationally more effective of the previous algorithms known in the literature;

(ii) in the paper of A.-Q. Li et al., the authors propose to evaluate the production performance of reservoir management units by selecting 12 indicators from the three aspects of production task, production technology, and reservoir development. According to the principle of fuzzy analytic hierarchy process (FAHP), this paper presents a new method to get the weights of these indicators;

(iii) in the paper of A.-B. Hanchevici et al., the authors present a networked control strategy for linear SISO systems affected by variant communication delays. They adjust the command provided by the PID controller by using fuzzy logic, whose inputs for the fuzzy logic controller (FLC) are represented by the delay and the variation of delay and the output is to adapt the PID controller's command to the new value of the communication. An application is also shown;

(iv) in the first paper of F. D. Martino and S. Sessa, the authors present a new method based on fuzzy transforms for detecting coarse-grained association rules in the datasets. The fuzzy association rules are represented in the form of linguistic expressions and a preprocessing phase determines the optimal fuzzy partition of the domains of the quantitative attributes. The extraction of the fuzzy association rules is executed via the AprioriGen algorithm and a confidence index calculated via the inverse fuzzy transform;

(v) in the paper of M. Matloka, the author presents the notion of (Φ_1, Φ_2)-convex fuzzy processes and some basic properties of such processes, being Φ_1, Φ_2 suitable generalized convex-like functions of one or more real variables;

(vi) in the paper of E. Eğrioğlu, the author studies fuzzy time series and proposes a new method based on fuzzy C-means and genetic algorithms for forecasting time invariant series; then he applies this method to several datasets;

(vii) in the second paper of F. D. Martino and S. Sessa, the authors present a method by using the hierarchical cluster based multispecies particle swarm optimization to generate a fuzzy system of Tagaki-Sugeno-Kang type encapsulated in a geographical information system considered as environmental decision support for spatial analysis. The spatial area is partitioned in subzones, so the data measured in each subzone are used to extract a fuzzy rule set of the above type.

This second issue continues the editorial lines of the first issue and we hope that it can capture the attention of many readers of the journal whose study can further expand the topics discussed here.

Salvatore Sessa
Ferdinando Di Martino
Irina G. Perfilieva

Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2012, Article ID 134068, 8 pages
doi:10.1155/2012/134068

Research Article

Production Performance Appraisal Rating for Reservoir Management Units Based on Fuzzy Clustering

An-Qi Li,[1] Yan-Rong Chang,[1] and Xin-Hai Kong[2]

[1] *Changqing Oilfield Branch, PetroChina Company Limited, Shanxi, Xi'an 710018, China*
[2] *Department of Graduate, Southwest Petroleum University, Sichuan, Chengdu 610500, China*

Correspondence should be addressed to Xin-Hai Kong, kongxinhai2@163.com

Received 13 July 2012; Accepted 2 September 2012

Academic Editor: Salvatore Sessa

In view of the existing situation of oilfield development, one kind of method to evaluate the production performance of reservoir management units (RMUs) was presented in this paper. Among the commonly used indicators of oilfield development, select 12 indicators from the three aspects of production task, production technology, and reservoir development. According to the principle of fuzzy analytic hierarchy process (FAHP), this paper introduced one kind of new method to get the weights of indicators. By means of the method of TOPSIS, it is easy to obtain the rankings for all the RMUs through calculating the weighted Euclidean distance between each RMU and the positive or negative ideal RMU. Considering the gap between the differences in RMUs, the production performance appraisal ratings of RMUs are determined by fuzzy clustering. This evaluation method could constantly improve the management level of reservoir units and deepen the delicacy management of oilfield development.

1. Introduction

The oilfield companies mostly take the management concept of "Benchmarking" during the process of oilfield development [1]. According to the dynamic analysis of reservoir development, the technical section provides a kind of development scheme and sets some feasible goals that should be achieved. And the reservoir management units (RMUs) must achieve the production goals in accordance with the development requirements [2]. Currently, the development department in the process of oilfield development evaluates the production performance of the reservoir management units (RMUs) based on their own statistics data and the assessment results calculated by themselves [3]. It means that the evaluation accuracy is not high enough and the crosswise contrast is not enough. In order to make the development department accurately and timely, grasp the current situation of development and management of RMUs, it needs to establish a relatively perfect evaluation system to really respond to the management level, efficiency, and development effect of RMUs, promoting the delicacy management of oilfield development. In this paper, we first present the evaluation indicators and their computing method. In order to reasonably decide the weight of each indicator, a kind of fuzzy AHP is introduced in Section 3. Next, we introduce the method of TOPSIS to decide the comprehensive ranking of RMUs and use the Euclidean distance to describe the proximity between two RMUs. However, the proximity among the RMUs is different, so we adopt the fuzzy clustering to classify the RMUs. In order to obtain the optimal classification, the F-statistics is mentioned in Section 4.

2. Evaluation Indicators and Their Computing Method

Through the analysis, the production performance evaluation indicators of RMU are divided into three aspects of production task, reservoir development, and production technology [1–5] (see Figure 1), including 12 indicators in the following.

2.1. Production Task. The production task [4, 5], denoted as u_1, contains the task completion rate of crude oil (u_{11}) and the task completion rate of water injection (u_{12}).

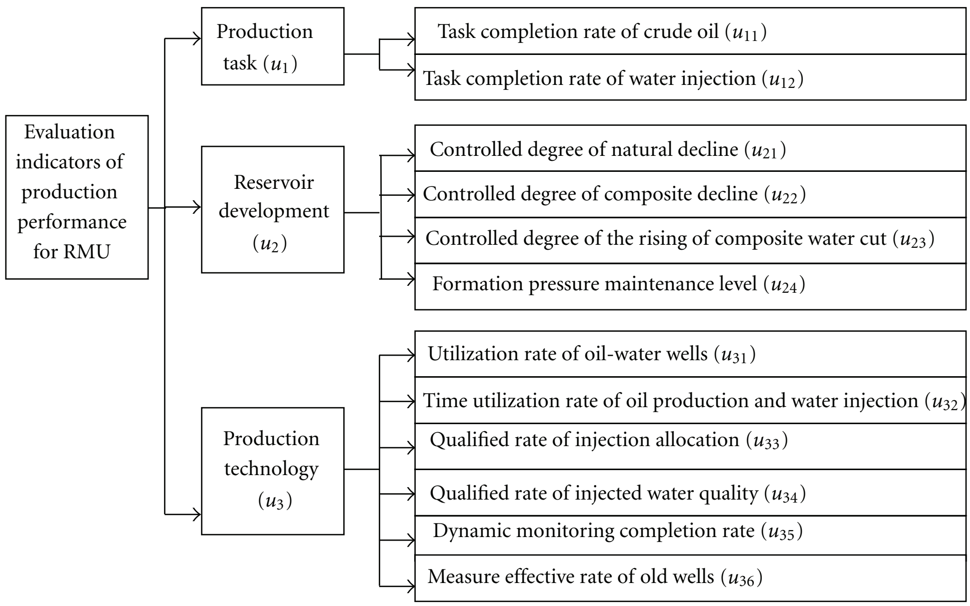


Figure 1: Hierarchical relationship of the evaluation indicators.

(1) *Task Completion Rate of Crude Oil.* $u_{11} = q_o/\widetilde{q}_o \times 100\%$, where q_o is the actual production of crude oil and $\widetilde{q}_o$ is the planned output of crude oil, unit: "tons".

(2) *Task Completion Rate of Water Injection.* $u_{12} = q_w/\widetilde{q}_w \times 100\%$, where q_w is the actual volume of water injection and $\widetilde{q}_w$ is the geology-required volume of water injection, unit: m^3.

2.2. Reservoir Development. The Reservoir development [6, 7], denoted as u_2, contains the controlled degree of natural decline (u_{21}), the controlled degree of composite decline (u_{22}), the controlled degree of the rising of composite water cut (u_{23}), and the formation pressure maintenance level (u_{24}).

(1) *Controlled Degree of Natural Decline.* $u_{21} = (\widetilde{D}_r - D_r)/\widetilde{D}_r \times 100\%$, where D_r is the actual natural decline rate and $\widetilde{D}_r$ is the control target of natural decline rate, unit: %.

(2) *Controlled Degree of Composite Decline.* $u_{22} = (\widetilde{D}_t - D_t)/\widetilde{D}_t \times 100\%$, where D_t is the actual composite decline rate and $\widetilde{D}_t$ is the control target of composite decline rate, unit: %.

(3) *Controlled Degree of the Rising of Composite Water Cut.* $u_{23} = (\Delta\widetilde{f}_w - \Delta f_w)/\Delta\widetilde{f}_w \times 100\%$, where Δf_w is the actual rising rate of composite water cut and $\Delta\widetilde{f}_w$ is the control target of the raising rate of composite water cut, unit: %.

(4) *Formation Pressure Maintenance Level.* $u_{24} = p/p_0 \times 100\%$, where p is the current formation pressure and p_0 is the original formation pressure, unit: "MPa".

2.3. Production Technology. The production technology [8], denoted as u_3, contains the utilization rate of oil-water wells (u_{31}), the hour utilization rate of oil production, and water injection (u_{32}), the qualified rate of injection allocation (u_{33}), the qualified rate of injected water quality (u_{34}), the dynamic monitoring completion rate (u_{35}), and the measure effective rate of old wells (u_{36}).

(1) *Utilization Rate of Oil-Water Wells.* $u_{31}^o = n_o/\widetilde{n}_o \times 100\%$, where u_{31}^o is the utilization rate of oil wells, n_o is the active number of oil wells, and $\widetilde{n}_o$ is the total number of oil wells; and $u_{31}^w = n_w/\widetilde{n}_w \times 100\%$, where u_{31}^w is the utilization rate of water injection wells, n_w is the active number of water injection wells, and $\widetilde{n}_w$ is the total number of water injection wells. Therefore, the utilization rate of oil-water wells can be defined by

$$u_{31} = \frac{\widetilde{n}_o}{\widetilde{n}_o + \widetilde{n}_w} \times u_{31}^o + \frac{\widetilde{n}_w}{\widetilde{n}_o + \widetilde{n}_w} \times u_{31}^w. \quad (1)$$

(2) *Time Utilization Rate of Oil Production and Water Injection.* $u_{32}^o = T_o/\widetilde{T}_o \times 100\%$, where u_{32}^o is the time utilization rate of oil production, T_o is the actual time of oil production and $\widetilde{T}_o$ is the calendar time of oil production; $u_{32}^w = T_w/\widetilde{T}_w \times 100\%$, where u_{32}^w is the time utilization rate of water injection, T_w is the actual time of water injection, and $\widetilde{T}_w$ is the calendar time of water injection, unit: "day". Therefore, the time utilization rate of oil production and water injection can be defined by

$$u_{32} = \frac{n_o}{n_o + n_w} \times u_{32}^o + \frac{n_w}{n_o + n_w} \times u_{32}^w. \quad (2)$$

(3) *Qualified Rate of Injection Allocation.* $u_{33} = c_w/\widetilde{c}_w \times 100\%$, where $\widetilde{c}_w$ is the sum of the number of wells and the

number of layers for water injection allocation and c_w is the sum of the qualified wells and the qualified layers.

(4) *Qualified Rate of Injected Water Quality.* $u_{34} = m_w/\widetilde{m}_w \times 100\%$, where $\widetilde{m}_w$ is the total number of water quality monitoring and m_w is the number of qualified water sample.

(5) *Dynamic Monitoring Completion Rate.* $u_{35}^j = k_j/\widetilde{k}_j \times 100\%$ is the dynamic monitoring completion rate of the jth project, and the comprehensive dynamic monitoring completion rate is defined as

$$u_{35} = \sum_{j=1}^{n} \left(\frac{\widetilde{k}_j}{\widetilde{K}} \times u_{35}^j \right), \tag{3}$$

where n is the number of monitoring project, $\widetilde{K}$ is the total number of planned monitoring, $\widetilde{k}_j$ is the number of planned monitoring for the jth project, and k_j is the actual number of monitoring for the jth project, unit: times.

(6) *Measure Effective Rate of Old Wells.* $u_{36} = n_{\text{eff}}/\widetilde{n} \times 100\%$, where $\widetilde{n}$ is the total number of measures and n_{eff} is the number of effective measures.

3. The Method to Determine the Weights of Evaluation Indicators

At present, with regard to determining the weights of evaluation indicators, the analytic hierarchy process (AHP) is a kind of relatively ideal method. While the traditional AHP needs to do the consistency test and constantly adjust the judgment matrix, some scholars put forward the fuzzy analytic hierarchy process (FAHP) [9–18]. We introduce a kind of fuzzy AHP to determine the weights of indicators in this section. The principle is as follows.

Definition 1 (see [9–11]). Assume that A is an n-order matrix, denoted as $A = (a_{ij})_{n\times n}$,

(a) A is called a fuzzy matrix if for all $i, j \in \{1, 2, \ldots, n\}$, A satisfies $0 \le a_{ij} \le 1$;

(b) A is called a fuzzy complementary matrix if for all $i, j \in \{1, 2, \ldots, n\}$, A also satisfies $a_{ij} + a_{ji} = 1$;

(c) A is called a fuzzy consistent judgment matrix if for any $i, j, k \in \{1, 2, \ldots, n\}$, A further satisfies $a_{ij} = a_{ik} - a_{jk} + 0.5$.

Property 1. A is a fuzzy consistent judgment matrix if and only if for any $i, j \in \{1, 2, \ldots, n\}$ and all $k \in \{1, 2, \ldots, n\}$, there exists a constant of a satisfying $a_{ik} - a_{jk} = a$.

Proof. (1) Suppose that A is a fuzzy consistent judgment matrix such that

$$a_{ij} = a_{ik} - a_{jk} + 0.5, \tag{4}$$

for any i, j and all k. Therefore, we have

$$a_{ik} - a_{jk} = a_{ij} - 0.5 = a. \tag{5}$$

(2) Suppose that for any given i, j, there exists a constant of a for each k such that

$$a_{ik} - a_{jk} = a, \tag{6}$$

and when $j = k$, we have

$$a_{ij} - a_{jj} = a. \tag{7}$$

Consequently, we get

$$a_{ik} - a_{jk} = a_{ij} - a_{jj}. \tag{8}$$

Since $a_{ij} + a_{ji} = 1$, so we can get $a_{jj} = 0.5$. Finally, we obtain

$$a_{ij} = a_{ik} - a_{jk} + 0.5. \tag{9}$$

□

Property 2. Assume A is a fuzzy complementary judgment matrix, we define a kind of fuzzy transform:

$$F : A \longrightarrow R, \tag{10}$$

$$a_{ij} \longrightarrow r_{ij} = \frac{r_i - r_j}{\alpha} + 0.5, \quad \alpha \ge 2(n-1), \tag{11}$$

where $r_i = \sum_{j=1}^{n} a_{ij}$, $i = 1, 2, \ldots, n$. Then, R is a fuzzy consistent judgment matrix.

Proof. Firstly, we prove that R is a fuzzy matrix. Since $0 \le a_{ij} \le 1$, we know that

$$\begin{gathered} 0.5 \le \sum_{k=1}^{n} a_{ik} \le n - 0.5, \\ 0.5 - n \le -\sum_{k=1}^{n} a_{jk} \le -0.5. \end{gathered} \tag{12}$$

Therefore, we have $|r_i - r_j| \le n - 1$. When $\alpha \ge 2(n-1)$, we get $0 \le r_{ij} \le 1$.

Secondly, we prove that R is a fuzzy complementary matrix.

$$r_{ij} + r_{ij} = \left(\frac{r_i - r_j}{\alpha} + 0.5 \right) + \left(\frac{r_j - r_i}{\alpha} + 0.5 \right) = 1. \tag{13}$$

Finally, we prove that R is a fuzzy consistent judgment matrix

$$\begin{aligned} r_{ik} - r_{jk} + 0.5 &= \left(\frac{r_i - r_k}{\alpha} + 0.5 \right) - \left(\frac{r_j - r_k}{\alpha} + 0.5 \right) + 0.5 \\ &= \frac{r_i - r_j}{\alpha} + 0.5 = r_{ij}. \end{aligned} \tag{14}$$

□

Property 3. Assume that $A = (a_{ij})_{n\times n}$ is a fuzzy complementary judgment matrix, and $w = (w_1, w_2, \ldots, w_n)$ is the weight vector or ordering vector, then

Table 1: The quantity scale of 0.1 ∼ 0.9.

Scale value	Definition	Explanation
0.5	Equally important	Two elements compared, equally important.
0.6	Slightly important	Two elements compared, an element is more slightly important than another element.
0.7	Obviously important	Two elements compared, an element is more obviously important than another element.
0.8	Much more important	Two elements compared, an element is much more strongly important than another element.
0.9	Extremely important	Two elements compared, an element is more extremely important than another element.
0.1, 0.2, 0.3, 0.4	Converse comparison	If the u_i is compared with u_j, get the judgment value a_{ij}, then u_j is compared with u_i, the judgment value is $a_{ji} = 1 - a_{ij}$.

Table 2: The statistical data of 12 RMUs in 2011.

RMU	u_1		u_2				u_3					
	u_{11}%	u_{12}%	u_{21}%	u_{22}%	u_{23}%	u_{24}%	u_{31}%	u_{32}%	u_{33}%	u_{34}%	u_{35}%	u_{36}%
1	99.8	96.3	−25.8	−30.3	−64.2	89.3	92.4	98.5	91.5	96.1	108.4	77.6
2	100.0	98.7	7.4	7.9	6.2	91.0	90.2	98.2	92.1	96.4	103.8	80.0
3	99.7	96.5	10.5	15.9	−1.3	92.7	93.0	98.4	93.2	95.8	97.7	82.5
4	100.3	99.0	11.1	17.0	66.2	95.3	95.2	99.0	93.9	96.7	106.7	84.8
5	99.9	96.7	3.5	20.2	28.8	92.2	89.8	98.1	92.6	93.4	98.6	81.4
6	99.3	97.4	1.1	2.5	20.9	87.2	92.1	98.1	91.9	96.5	96.5	76.5
7	99.5	98.2	6.0	8.5	76.0	90.7	90.5	98.4	89.4	97.0	97.4	79.3
8	100.1	99.1	−19.9	−22.3	−15.8	82.0	90.8	98.3	95.0	91.9	104.9	75.7
9	100.0	97.9	20.4	21.3	37.6	92.5	74.1	99.0	90.5	95.6	96.4	74.2
10	100.0	96.6	0.7	1.2	45.9	90.4	88.5	99.0	88.3	95.3	105.6	72.2
11	99.8	97.2	12.5	13.6	18.7	89.1	98.3	98.7	89.4	96.3	102.1	68.8
12	99.6	97.0	52.4	79.7	64.6	86.3	69.8	98.7	88.0	86.6	106.3	69.2

(1) when A is consistent, by using the normalizing rank aggregation method, the weight is given by

$$w_i = \frac{2}{n^2}\sum_{j=1}^{n} a_{ij}, \quad i = 1, 2, \dots, n, \tag{15}$$

(2) when A is not entirely consistent, first carry out the fuzzy consistent transformation by (11), and then through the normalizing rank aggregation method, the weight is given by

$$w_i = \frac{1}{n} - \frac{1}{\alpha} + \frac{2}{n\alpha}\sum_{j=1}^{n} a_{ij}, \quad \alpha \geq 2(n-1). \tag{16}$$

Proof. (1) When A is completely consistent,

$$\begin{aligned} w_i &= \frac{\sum_{j=1}^{n} a_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}} = \frac{\sum_{j=1}^{n} a_{ij}}{\sum_{1\leq i<j\leq n}^{n}\left(a_{ij} + a_{ji}\right) + 0.5n} \\ &= \frac{\sum_{j=1}^{n} a_{ij}}{n^2/2} = \frac{2}{n^2}\sum_{j=1}^{n} a_{ij}. \end{aligned} \tag{17}$$

(2) When A is not entirely consistent,

$$\begin{aligned} w_i &= \frac{\sum_{j=1}^{n} r_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{n} r_{ij}} = \frac{\sum_{j=1}^{n} r_{ij}}{n^2/2} \\ &= \frac{\sum_{j=1}^{n}\left(\left(\left(r_i - r_j\right)/\alpha\right) + 0.5\right)}{n^2/2} = \frac{1}{n} - \frac{1}{\alpha} + \frac{2}{n\alpha}\sum_{j=1}^{n} a_{ij}. \end{aligned} \tag{18}$$

From the above analysis, the steps for determining the weights can be summarized in the following. □

Step 1. The expert gives out the fuzzy complementary judgment matrix A by using the pairwise comparison method based on the quantity scale of 0.1 ∼ 0.9 [9] (see Table 1).

Step 2. Check whether A is consistent or not. If consistent, calculate the weights by (15); if not, calculate the weight by (16).

4. Production Performance Appraisal Rating for RMUs

In this section, we introduce the method of TOPSIS [19] to decide the comprehensive ranking of RMUs and use the

Table 3: The standardized decision data.

RMU	u_{11}	u_{12}	u_{21}	u_{22}	u_{23}	u_{24}	u_{31}	u_{32}	u_{33}	u_{34}	u_{35}	u_{36}
1	0.5000	0.0000	0.0000	0.0000	0.0000	0.5489	0.7930	0.4444	0.5000	0.9135	1.0000	0.5500
2	0.7000	0.8571	0.4246	0.3473	0.5021	0.6767	0.7158	0.1111	0.5857	0.9423	0.6167	0.7000
3	0.4000	0.0714	0.4642	0.4200	0.4486	0.8045	0.8140	0.3333	0.7429	0.8846	0.1083	0.8563
4	1.0000	0.9643	0.4719	0.4300	0.9301	1.0000	0.8912	1.0000	0.8429	0.9712	0.8583	1.0000
5	0.6000	0.1429	0.3747	0.4591	0.6633	0.7669	0.7018	0.0000	0.6571	0.6539	0.1833	0.7875
6	0.0000	0.3929	0.3440	0.2982	0.6070	0.3910	0.7825	0.0000	0.5571	0.9519	0.0083	0.4813
7	0.2000	0.6786	0.4067	0.3527	1.0000	0.6541	0.7263	0.3333	0.2000	1.0000	0.0833	0.6563
8	0.8000	1.0000	0.0754	0.0727	0.3452	0.0000	0.7368	0.2222	1.0000	0.5096	0.7083	0.4313
9	0.7000	0.5714	0.5908	0.4691	0.7261	0.7895	0.1509	1.0000	0.3571	0.8654	0.0000	0.3375
10	0.7000	0.1071	0.3389	0.2864	0.7853	0.6316	0.6561	1.0000	0.0429	0.8365	0.7667	0.2125
11	0.5000	0.3214	0.4898	0.3991	0.5913	0.5338	1.0000	0.6667	0.2000	0.9327	0.4750	0.0000
12	0.3000	0.2500	1.0000	1.0000	0.9187	0.3233	0.0000	0.6667	0.0000	0.0000	0.8250	0.0250

Table 4

(a)

	u_1	u_2	u_3
u_1	0.5	0.7	0.7
u_2	0.3	0.5	0.5
u_3	0.3	0.5	0.5

(b)

	u_{11}	u_{12}
u_{11}	0.5	0.5
u_{12}	0.5	0.5

(c)

	u_{21}	u_{22}	u_{23}	u_{24}
u_{21}	0.5	0.5	0.5	0.6
u_{22}	0.5	0.5	0.5	0.6
u_{23}	0.5	0.5	0.5	0.6
u_{24}	0.4	0.4	0.4	0.5

(d)

	u_{31}	u_{32}	u_{33}	u_{34}	u_{35}	u_{36}
u_{31}	0.5	0.5	0.6	0.7	0.9	0.5
u_{32}	0.5	0.5	0.6	0.7	0.9	0.5
u_{33}	0.4	0.4	0.5	0.6	0.8	0.4
u_{34}	0.3	0.3	0.4	0.5	0.7	0.3
u_{35}	0.1	0.1	0.2	0.3	0.5	0.1
u_{36}	0.5	0.5	0.6	0.7	0.9	0.5

Euclidean distance to describe the proximity between two RMUs. However, the proximity among the RMUs is different, for this reason, we adopt the fuzzy clustering to classify the RMUs. In order to obtain the optimal classification, the F-statistics is mentioned in this paper.

4.1. Comprehensive Ranking. Assume that there are m RMUs and n evaluation indicators, the decision data matrix is denoted by $X = (x_{ij})_{m\times n}$. According to the method of TOPSIS, the comprehensive ranking procedure for RMUs consists of the following steps.

Step 1. Standardize the decision data matrix. The standardized decision data matrix is denoted by $Y = (y_{ij})_{m\times n}$, and the transformation formula are given in the following;

(a) when the jth indicator is the benefit type,

$$y_{ij} = \frac{x_{ij} - \min_i\left\{x_{ij}\right\}}{\max_i\left\{x_{ij}\right\} - \min_i\left\{x_{ij}\right\}}. \tag{19}$$

(b) when the jth indicator is the cost type,

$$y_{ij} = \frac{\max_i\left\{x_{ij}\right\} - x_{ij}}{\max_i\left\{x_{ij}\right\} - \min_i\left\{x_{ij}\right\}}. \tag{20}$$

(c) when the jth is the target type,

$$y_{ij} = 1 - \frac{\left|x_{ij} - x_0\right|}{\max\left\{\max_i\left\{x_{ij}\right\} - x_0,\ x_0 - \min_i\left\{x_{ij}\right\}\right\}}, \tag{21}$$

where x_0 is the target, and $\min_i\{x_{ij}\} \leq x_0 \leq \max_i\{x_{ij}\}$.

Step 2. Determine the weights of indicators. The weight vector $\omega = (\omega_1, \omega_2, \ldots, \omega_n)$ can be obtained by FAHP. Furthermore, we could calculate the weighted decision matrix $Z = (z_{ij})_{m\times n}$, where $z_{ij} = w_j \cdot y_{ij}$.

Step 3. Determine the positive ideal vector and the negative ideal vector. Respectively, denoted by $z^+ = (z_1^+, z_2^+, \ldots, z_n^+)$ $z^- = (z_1^-, z_2^-, \ldots, z_n^-)$, where $z_j^+ = \max_i\{z_{ij}\}$, $z_j^- = \min_i\{z_{ij}\}$.

Step 4. Calculate the Euclidean distance from the positive ideal vector and the negative ideal vector. The Euclidean distance between the ith RMU and the positive ideal vector is denoted by

$$d_i^+ = \sqrt{\sum_{k=1}^{n}\left(z_{ik} - z_k^+\right)^2}. \tag{22}$$

Table 5: The weights of all the indicators.

Indicator	Weights of indicators	Subindicators	Weights of subindicators	Combination weights
u_1	0.4222	u_{11}	0.5	0.2111
		u_{12}	0.5	0.2111
u_2	0.2889	u_{21}	0.2625	0.0758
		u_{22}	0.2625	0.0758
		u_{23}	0.2625	0.0758
		u_{24}	0.2125	0.0614
u_3	0.2889	u_{31}	0.2056	0.0594
		u_{32}	0.2056	0.0594
		u_{33}	0.1722	0.0498
		u_{34}	0.1388	0.0401
		u_{35}	0.0722	0.0209
		u_{36}	0.2056	0.0594

Table 6: The relative closeness of 12 RMUs.

RMU	d_i^+	d_i^-	f_i^*
1	0.2762	0.1366	0.3309
2	0.1224	0.2559	0.6765
3	0.2495	0.1454	0.3683
4	0.0606	0.3325	0.8459
5	0.2223	0.1715	0.4355
6	0.2723	0.1264	0.3170
7	0.2054	0.1919	0.4829
8	0.1477	0.2823	0.6565
9	0.1448	0.2249	0.6083
10	0.2252	0.1874	0.4542
11	0.2065	0.1668	0.4468
12	0.2450	0.1594	0.3941

The Euclidean distance between the ith RMU and the negative ideal vector is denoted by

$$d_i^- = \sqrt{\sum_{k=1}^{n} \left(z_{ik} - z_k^-\right)^2}. \qquad (23)$$

Step 5. Calculate the relative closeness to the positive ideal vector. The relative closeness can be defined as

$$f_i^* = \frac{d_i^-}{\left(d_i^- + d_i^+\right)}, \quad i = 1, 2, \ldots, m. \qquad (24)$$

Step 6. Decide the ranking according to the value of f_i^*. The bigger the closeness shows the better the ranking.

4.2. Rating of RMUs. Considering the gap between the differences in RMUs, the comprehensive ranking still is not enough. It is necessary to classify the RMUs with fuzzy clustering. Therefore, further we work out the distance matrix of RMUs, denoted by $D = (d_{ij})_{m\times m}$, where

$$d_{ij} = \frac{1}{n} \left| \sum_{k=1}^{n} \left(z_{ik} - z_{jk}\right) \right|. \qquad (25)$$

From (25), we can get the dynamic fuzzy clustering and the dynamic clustering figure.

In order to reasonably determine the number of classification, we introduce a kind of F statistics [20]:

$$F = \frac{\sum_{i=1}^{r} n_i \sum_{k=1}^{m} \left(\overline{x}_{ik} - \overline{x}_k\right)^2 / (r-1)}{\sum_{i=1}^{r} \sum_{j=1}^{n_i} \sum_{k=1}^{m} \left(x_{ik} - \overline{x}_{jk}\right)^2 / (n-r)} \sim F(r-1, n-r), \qquad (26)$$

where r is the number of classification and n_i is the number of elements in the ith classification.

$$\overline{x}_{ik} = \frac{1}{n_i} \sum_{j=1}^{n_i} x_{jk}, \quad k = 1, 2, \ldots, m \qquad (27)$$

is the average value of the kth indicator in the ith classification.

$$\overline{x}_k = \frac{1}{n} \sum_{j=1}^{n} x_{jk}, \quad k = 1, 2, \ldots, m \qquad (28)$$

is the average value of the kth indicator of all the RMUs.

We can calculate the values of F for each classification scheme by (26), under a given reliability $\alpha = 0.05$, and find out the critical values of $F_{0.05}$. If $F > F_{0.05}$, the corresponding classification is feasible. Generally, take the classification number corresponding directly with $\max\{F - F_{0.05}\}$ as the optimal classification number and finally get the best classification rating.

5. Example Analysis

The statistical data of 12 reservoir management units (RMUs) of an oilfield in the year of 2011 are listed in Table 2.

According to the basic data in Table 2, we could obtain the evaluation results.

Step 1. Standardize the above decision data. The 12 indicators are all the benefit type; their standardized decision data are shown in Table 3.

Table 7: The distance between any two RMUs.

RMU	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0000	0.0258	0.0097	0.0465	0.0117	0.0012	0.0154	0.0214	0.0221	0.0126	0.0122	0.0073
2	0.0258	0.0000	0.0161	0.0207	0.0141	0.0245	0.0104	0.0044	0.0037	0.0132	0.0136	0.0185
3	0.0097	0.0161	0.0000	0.0368	0.0021	0.0084	0.0058	0.0118	0.0125	0.0029	0.0026	0.0023
4	0.0465	0.0207	0.0368	0.0000	0.0347	0.0452	0.0311	0.0250	0.0244	0.0339	0.0342	0.0392
5	0.0117	0.0141	0.0021	0.0347	0.0000	0.0105	0.0037	0.0097	0.0104	0.0008	0.0005	0.0044
6	0.0012	0.0245	0.0084	0.0452	0.0105	0.0000	0.0142	0.0202	0.0209	0.0113	0.0110	0.0061
7	0.0154	0.0104	0.0058	0.0311	0.0037	0.0142	0.0000	0.0060	0.0067	0.0028	0.0032	0.0081
8	0.0214	0.0044	0.0118	0.0250	0.0097	0.0202	0.0060	0.0000	0.0007	0.0089	0.0092	0.0141
9	0.0221	0.0037	0.0125	0.0244	0.0104	0.0209	0.0067	0.0007	0.0000	0.0095	0.0099	0.0148
10	0.0126	0.0132	0.0029	0.0339	0.0008	0.0113	0.0028	0.0089	0.0095	0.0000	0.0003	0.0053
11	0.0122	0.0136	0.0026	0.0342	0.0005	0.0110	0.0032	0.0092	0.0099	0.0003	0.0000	0.0049
12	0.0073	0.0185	0.0023	0.0392	0.0044	0.0061	0.0081	0.0141	0.0148	0.0053	0.0049	0.0000

Table 8: The values of F and the critical values of $F_{0.05}$.

Classification Number	2	3	4	5	6	7	8	9	10	11
F	0.77	1.79	1.63	15.06	28.50	1.81	1.42	2.32	0.24	0.79
$F_{0.05}$	4.96	4.26	4.07	4.12	4.39	4.95	6.09	8.85	19.39	241.88
$\Delta F = F - F_{0.05}$	−4.19	−2.47	−2.44	10.94	24.12	−3.14	−4.67	−6.52	−19.14	−241.09

Step 2. Get the judgment matrixes of all the hierarchies through expert scoring Table 4.

Therefore, we can calculate the weights for the evaluation indicators shown in Table 5.

Step 3. Calculate the relative closeness of every RMU (see Table 6).

From Table 6, we know the comprehensive ranking as follows:

$$f_4^* \succ f_2^* \succ f_8^* \succ f_9^* \succ f_7^* \succ f_{10}^* \succ f_{11}^* \succ f_5^* \succ f_{12}^* \succ f_3^* \succ f_1^* \succ f_6^*. \tag{29}$$

Step 4. Determine the best classification rating. First of all, by (25), calculate the distance between any two RMUs (see Table 7).

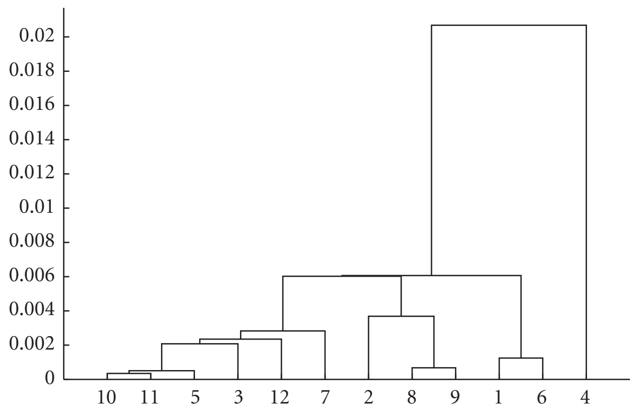


Figure 2: The dynamic clustering figure.

Next, we can draw the dynamic fuzzy clustering figure (see Figure 2).

Lastly, calculate the values of F for every kind of classification by F-statistics. The values of F are listed in Table 8.

From Table 8, it can be seen that the best classification number is "six", namely,

$$\{4\} \succ \{2\} \succ \{8,9\} \succ \{7\} \succ \{3,5,10,11,12\} \succ \{1,6\}. \tag{30}$$

6. Conclusions

Through analyzing the actual situation in the process of oilfield development, we first present some practically feasible evaluation indicators and their computing method in the second section. In order to reasonably decide the weight of each indicator, we introduce a kind of fuzzy AHP in Section 3. Next, by means of the method of TOPSIS, it is easy to decide the comprehensive rankings of RMUs through calculating the weighted Euclidean distance between every RMU and the positive or negative ideal RMU. And we use the Euclidean distance to describe the proximity between two RMUs. Considering the proximity among the RMUs is different, the fuzzy clustering is introduced to classify the RMUs, and the production performance appraisal ratings of RMUs are determined by fuzzy clustering. In order to obtain the optimal classification, the F-statistics is mentioned in Section 4. Finally, a practical example is illustrated to explain the feasibility of this method.

In order to make the development department of oilfield companies accurately and timely grasp the current situation of oilfield development and management of RMUs, it needs to establish a relatively perfect evaluation method to really respond to the management level, efficiency, and development effect of RMUs, promoting the delicacy management of oilfield development. As we know, by means of the relatively effective evaluation method to ascertain the appraisal rating of the RMUs, we cannot only know their production performance, but also it is helpful to motivate the enthusiasm of practical production for all the RMUs. By using the evaluation method proposed in this paper, the management level could be constantly improved, and the delicacy management of oilfield development can be deepened. And the oilfield companies continuously strengthen the digital construction, so it will support the accuracy and objectivity of the evaluation method.

References

[1] Y. M. Chen, *Reservoir Management*, China University of Petroleum Press, DongYing, China, 2007.

[2] F. Jahn, M. Cook, and M. Graham, *Hydrocarbon Exploration & Production*, vol. 55, Elsevier Science, Aberdeen, UK, 2nd edition, 2008.

[3] J. R. Fanchi, *Integrated Reservoir Asset Management: Principles and Best Practices*, Gulf Professional Publishing, Houston, Tex, USA, 2010.

[4] J. Y. Liu, *Geological Foundation of Oil and Gas Field Development*, Petroleum Industry Press, Beijing, China, 2006.

[5] P. T. Geoffrey, *Oil and Gas: A Practical Handbook*, Globe Law and Business, London, UK, 2009.

[6] N. Ezekwe, *Petroleum Reservoir Engineering Practice*, Prentice Hall, Upper Saddle River, NJ, USA, 2010.

[7] C. L. Li, *Fundamentals of Reservoir Engineering*, Petroleum Industry Press, Beijing, China, 2nd edition, 2011.

[8] Y. C. Li, *The Technology of Petroleum Production*, Petroleum Industry Press, Beijing, China, 2nd edition, 2009.

[9] J. J. Zhang, "Fuzzy analytic hierarchy process," *A Chinese Journal of Fuzzy Systems and Mathematics*, vol. 14, no. 2, pp. 80–88, 2000.

[10] Y. J. Lv, "Weight calculation method of fuzzy analytical hierarchy process," *A Chinese Journal of Fuzzy Systems and Mathematics*, vol. 16, no. 2, pp. 79–85, 2002.

[11] J. J. Zhang, "A new ranking method of fuzzy complementary judgment matrix," *Operations Research and Management Science*, vol. 14, no. 2, pp. 59–63, 2005.

[12] Y. J. Lu, W. L. Shi, and X. R. Guo, "The conditions of rank preservation and a general Priority formula for fuzzy complementary judgment matrix," *Mathematics in Practice and Theory*, vol. 39, no. 15, pp. 153–158, 2009.

[13] Z. P. Fan, J. Ma, and Q. Zhang, "An approach to multiple attribute decision making based on fuzzy preference information on alternatives," *Fuzzy Sets and Systems*, vol. 131, no. 1, pp. 101–106, 2002.

[14] F. Kong and H. Y. Liu, "Applying fuzzy analytic hierarchy process to evaluate success factors of E-commerce," *International Journal of Information and Systems Sciences*, vol. 1, pp. 406–412, 2005.

[15] A. N. Haq and G. Kannan, "Fuzzy analytical hierarchy process for evaluating and selecting a vendor in a supply chain model," *International Journal of Advanced Manufacturing Technology*, vol. 29, no. 7-8, pp. 826–835, 2006.

[16] A. Azadeh, S. Nazari-Shirkouhi, L. Hatami-Shirkouhi, and A. Ansarinejad, "A unique fuzzy multi-criteria decision making: computer simulation approach for productive operators' assignment in cellular manufacturing systems with uncertainty and vagueness," *International Journal of Advanced Manufacturing Technology*, vol. 56, no. 1–4, pp. 329–343, 2011.

[17] W. G. Li, Q. Yu, and R. C. Luo, "Application of fuzzy analytic hierarchy process and neural network in power transformer risk assessment," *Journal of Central South University*, vol. 19, pp. 982–987, 2012.

[18] Z. B. Liu, W. Qing, and X. H. Kong, "Fuzzy comprehensive evaluation on the effect of measures operation for oil-water well," *Advances in Fuzzy Systems*, vol. 2011, Article ID 695690, 5 pages, 2011.

[19] D. Yong, "Plant location selection based on fuzzy TOPSIS," *International Journal of Advanced Manufacturing Technology*, vol. 28, no. 7-8, pp. 839–844, 2006.

[20] Y. D. Tan, M. Fornage, and H. Xu, "Ranking analysis of F-statistics for microarray data," *BMC Bioinformatics*, vol. 9, article 142, 2008.

Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2012, Article ID 258476, 12 pages
doi:10.1155/2012/258476

Research Article

Detection of Fuzzy Association Rules by Fuzzy Transforms

Ferdinando Di Martino and Salvatore Sessa

Dipartimento di Costruzioni e Metodi Matematici in Architettura, Università degli Studi di Napoli Federico II, Via Monteoliveto 3, 80134 Napoli, Italy

Correspondence should be addressed to Salvatore Sessa, sessa@unina.it

Received 3 March 2012; Revised 25 June 2012; Accepted 25 June 2012

Academic Editor: Irina G. Perfilieva

We present a new method based on the use of fuzzy transforms for detecting coarse-grained association rules in the datasets. The fuzzy association rules are represented in the form of linguistic expressions and we introduce a pre-processing phase to determine the optimal fuzzy partition of the domains of the quantitative attributes. In the extraction of the fuzzy association rules we use the AprioriGen algorithm and a confidence index calculated via the inverse fuzzy transform. Our method is applied to datasets of the 2001 census database of the district of Naples (Italy); the results show that the extracted fuzzy association rules provide a correct coarse-grained view of the data association rule set.

1. Introduction

Fuzzy association rules extraction [1] is a fundamental process in data mining for many topics as classification and information retrieval. Many techniques have been presented for extracting fuzzy association rules in datasets and databases [1–21]; some authors are using soft computing approaches as evolutionary methods [14, 21–26] and clustering algorithms [24, 27, 28] for creating fuzzy partitions of data attribute domains. In many practical cases the user does not need to make a detailed fuzzy partition of the domain attributes and a fine exploration of fuzzy association rules between attributes in datasets. Indeed, sometimes his purpose is to acquire a more immediate coarse-grained knowledge of hidden relations in the data creating a coarse-grained fuzzy partition of each attribute domain and by estimating fuzzy association rules with evaluative linguistic expressions.

Here we propose a new approach for detecting coarse-grained fuzzy association rules in datasets, based on fuzzy transforms (for short, F-transforms), which are already used for image analysis [18, 29–34], data analysis [31, 35], and forecasting [34, 36]. In particular, in [31] a modality of extraction of fuzzy association rules in a coarse grained way is proposed by using F-transforms. In this paper we follow this approach; our framework is composed from a pre-processing phase (necessary in order to obtain the optimal cardinality of the fuzzy partitions of the data attribute domains), and of two successive processes for extracting the fuzzy association rules. Let us consider a dataset represented by Table 1.

We define the context $w_i = [a_i, b_i]$ of the given attribute X_i by setting $a_i = \min\{p_{1i}, \ldots, p_{mi}\}$ and $b_i = \max\{p_{1i}, \ldots, p_{mi}\}$. Thus we can consider a fuzzy partition of $n(i)$ fuzzy sets $F = \{A_{i1}, A_{i2}, \ldots, A_{in(i)}\}$ for each context w_i. A fuzzy association rule $S \to T$ between two disjoint sets of attributes $S = \{X_1, \ldots, X_k\}$ and $T = \{X_l, \ldots, X_z\}$ can be generally expressed as

$$\begin{aligned} &\text{IF } (X_1 \text{ is } A_{1h(1)}) \text{AND} \cdots \text{AND} (X_k \text{ is } A_{kh(k)}), \\ &\text{THEN } (X_l \text{ is } A_{lh(l)}) \text{AND} \cdots \text{AND} (X_z \text{ is } A_{zh(z)}). \end{aligned} \tag{1}$$

In [31] the F-transforms approach, for detecting fuzzy association rules from a dataset in a coarse-grained view and to construct a framework of fuzzy association rules between

Table 1: Schema of a dataset.

	X_1	$\cdots$	X_i	$\cdots$	X_r
O_1	p_{11}	$\cdots$	p_{1i}	$\cdots$	p_{1r}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
O_j	p_{j1}	$\cdots$	p_{ji}	$\cdots$	p_{jr}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
O_m	p_{m1}	$\cdots$	p_{mi}	$\cdots$	p_{mr}

$X = \{X_1, \ldots, X_i, \ldots, X_r\}$ is the set of the attributes and $O = \{O_1, O_2, \ldots, O_m\}$ is the set of the objects or instances in the dataset. A value p_{ji} belonging to the domain of the attribute X_i is called an item and a set of items is called an itemset. An association rule is represented as a directional dependence between sets of attributes in the dataset. It is indicated with an expression of the type $S \to T$, where S, T are sets of attributes. The implication operator means that if all the items in S exist in an object, then all the items in T are also in the object with a high probability [2].

Table 2: Nodes obtained for [8, 37] with $n = 4$.

x_{i-1}	x_i	x_{i+1}	Label
10	10	15	A_1
10	15	20	A_2
15	20	25	A_3
20	25	25	A_4

two disjoint sets of attributes $S = \{X_1, \ldots, X_k\}$ and $T = \{X_z\}$, is used in the form:

$$\begin{aligned} &\text{IF}(X_1 \text{ is } A_{1h(1)})\text{AND} \cdots \text{AND}(X_k \text{ is } A_{kh(k)}) \\ &\overset{F}{\sim} (\text{mean } X_z \text{ is } \aleph), \end{aligned} \tag{2}$$

where (in accordance to [18, 29]) $A_{ih(i)}$ is the h_ith basic function of the uniform partition of the ith context associated to the node X_{hi}. Each clause in the antecedent assumes the linguistic meaning "X_i is approximately LV$(A_{ih(i)})$," with LV$(A_{i,h(i)})$ being the evaluative expression assigned to the fuzzy set $(A_{i,h(i)})$ and $\aleph$ a pure evaluative linguistic expression that characterizes the component $F_{h(1)h(2)\cdots h(k)}$ of the F-transform corresponding at the fuzzy sets $A_{1h(1)}, \ldots, A_{kh(k)}$. The term "mean" in the consequent derives from the fact that the component $F_{h(1)h(2)\cdots h(k)}$ is obtained as a mean of the values of the item X_z weighted over the basic functions $A_{1h(1)}, \ldots, A_{kh(k)}$. The symbol $\overset{F}{\sim}$ represents an association between attributes obtained with the F-transforms. In other words, we can say that a fuzzy association rule expressed in the form (2) provides a synthetic valuation of the association rule between attributes, with the associations expressed linguistically with the model of [18, 29] and the fuzzy sets in the antecedent $A_{1h(1)}, \ldots, A_{kh(k)}$ being the basic functions of the uniform fuzzy partition of the corresponding context.

Following the definitions and notations of [32], let $n \geq 2$ and $x_1, x_2, \ldots, x_n$ be points of a specific context $[a,b]$, called nodes, such that $x_1 = a < x_2 < \cdots < x_n = b$. The assigned family of fuzzy sets $A_1, \ldots, A_n : [a, b] \to [0, 1]$, called basic functions, is a fuzzy partition of $[a, b]$ if the following holds:

(A) $A_i(x_i) = 1$ for every $i = 1, 2, \ldots, n$;

(B) $A_i(x_i) = 0$ if $x \notin (x_{i-1}, x_{i+1})$ for $i = 2, \ldots, n$;

(C) $A_i(x)$ is a continuous function on $[a,b]$;

(D) $A_i(x)$ strictly increases on $[x_{i-1}, x_i]$ for $i = 2, \ldots, n$ and strictly decreases on $[x_i, x_{i+1}]$ for $i = 1, \ldots, n-1$;

(E) $\sum_{i=1}^{n} A_i(x) = 1$ for every $x \in [a, b]$.

We say that the fuzzy sets $\{A_1(x), \ldots, A_n(x)\}$ form an uniform fuzzy partition if

(F) $n \geq 3$ and $x_i = a + h \cdot (i-1)$, where $h = (b-a)/(n-1)$ and $i = 1, 2, \ldots, n$ (equidistance of the nodes);

(G) $A_i(x_i - x) = A_i(x_i + x)$ for every $x \in [0, h]$ and $i = 2, \ldots, n-1$;

(H) $A_{i+1}(x) = A_i(x - h)$ for every $x \in [x_i, x_i + 1]$ and $i = 1, 2, \ldots, n-1$.

An example of basic functions is given by triangular fuzzy sets. For example, if $n = 4$, $a = x_1 = 10$, and $b = x_4 = 25$, we obtain $h = 3$. Table 2 shows the nodes that characterize each basic function.

If we define basic functions of triangular form, as example, we have for $k = 2, \ldots, n-1$:

$$\begin{aligned} A_1(x) &= \begin{cases} 1 - \dfrac{x - x_1}{h} & \text{if } x \in [x_1, x_2] \\ 0 & \text{otherwise,} \end{cases} \\ A_k(x) &= \begin{cases} \dfrac{x - x_{k-1}}{h} & \text{if } x \in [x_{k-1}, x_k] \\ 1 - \dfrac{x - x_k}{h} & \text{if } x \in [x_k, x_{k+1}] \\ 0 & \text{otherwise,} \end{cases} \\ A_n(x) &= \begin{cases} \dfrac{x - x_{n-1}}{h} & \text{if } x \in [x_{n-1}, x_n] \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{3}$$

Figure 1 shows the four basic functions forming the fuzzy partition of the context [8, 37] given in (3) for $n = 4$.

The method given in [31] can be very useful when we need to extract fuzzy association rules in an approximate way from a dataset; for each attribute a coarse-grained uniform fuzzy partition of its context is created and the evaluative linguistic expression in the consequent represents a weighted mean of the values of the attribute X_z. Nevertheless, as pointed in [35], this approach does not take into account the necessity to have the data sufficiently dense with respect to the chosen fuzzy partition, otherwise the F-transforms cannot be used. In order to avoid the choice of a fuzzy partition either too fine or too coarse of the contexts, it is necessary to define a pre-processing phase which determines the optimal fuzzy partition to choose with respect to the density of the data. Here we propose a technique based on F-transforms to detect the framework of strong fuzzy association rules from a dataset in the form (2). In our method we determine the best uniform fuzzy partition of each context constituted from triangular fuzzy sets like in Figure 1. For each context $w_i, i = 1, \ldots, r$, of the dataset in Table 1, an initial uniform fuzzy partition of n_i triangular fuzzy sets (3) is established.

To control that the data are sufficiently dense with respect to the chosen partition, we must check that for

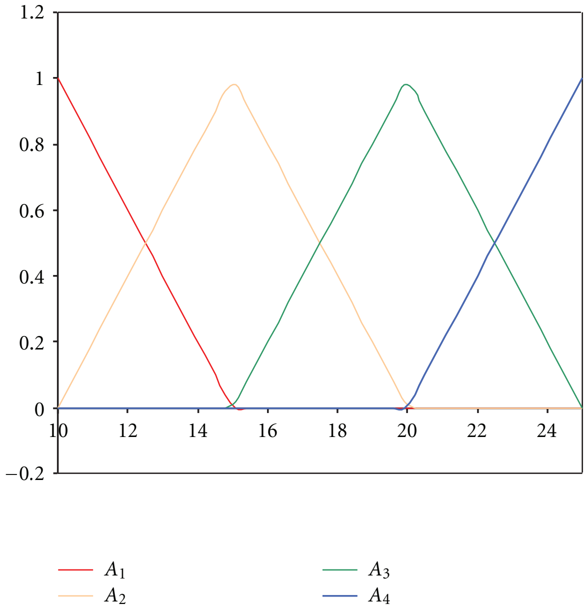


Figure 1: Example of triangular basic functions.

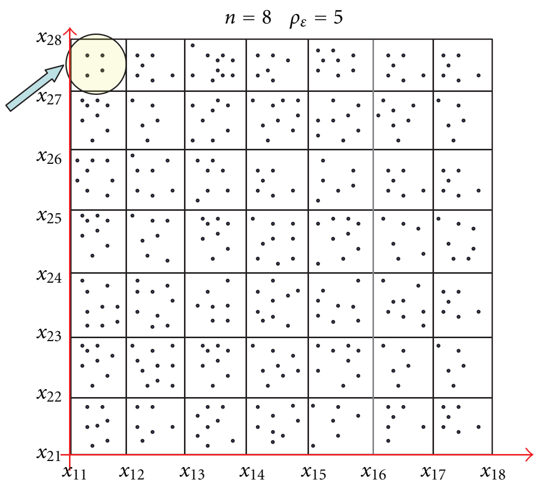


Figure 2: Example of too fine uniform fuzzy partition set.

each combination $(A_{1h(1)}, A_{2h(2)}, \ldots, A_{rh(r)}), h(1), \ldots, h(r) \in \{1, \ldots, n\}$, there exists at least one data point such that $A_{1h(1)}(x_{i1}) \cdot A_{2h(2)}(x_{i2}) \cdot \cdots \cdot A_{rh(r)}(x_{ir}) > 0$. In order to reduce the computational complexity (we should extend this control on $n_1 \cdot n_2 \cdot \cdots \cdot n_r$ combinations where n_i is the cardinality of the fuzzy partition of the context w_i), we consider n fuzzy sets for each context, thus the number of the possible combinations $(A_{1h(1)}, A_{2h(2)}, \ldots, A_{rh(r)})$ is n^r.

In Section 2 we schematize our method for extraction fuzzy association rules. In Section 3 we recall the definitions of F-transforms in k (≥ 2) variables. In Section 4 we present the extraction process of the fuzzy association rules, in Section 5 we show some results coming from the tests conducted on the databases of the 2001 census data related to the city of Naples (Italy), supplied by ISTAT (Istituto Nazionale di Statistica). These databases contain information on population, buildings, housing, family, employment work for each census zone of Naples. The reliability of the fuzzy association rules is also discussed as well.

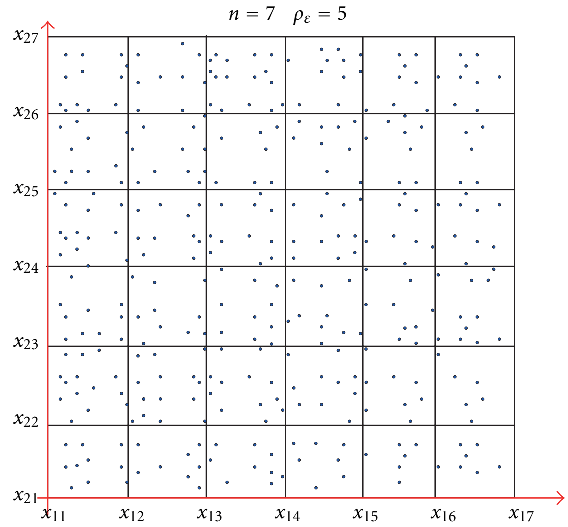


Figure 3: Example of correct uniform fuzzy partition set.

2. Pre-Processing and Extraction Fuzzy Association Rule Phases

We use a pre-processing phase that determines the optimal value of n by starting from an initial cardinality of the fuzzy partitions. Furthermore this phase ensures that the data are sufficiently dense with respect to the chosen fuzzy partitions, so that $A_{1h(1)}(x_{i1}) \cdot A_{2h(2)}(x_{i2}) \cdot \cdots \cdot A_{rh(r)}(x_{ir}) > 0$ for each combination $(A_{1h(1)}, A_{2h(2)}, \ldots, A_{rh(r)})$ of basic functions with respect to a minimal density of data points $(x_{i1}, x_{i2}, \ldots, x_{ir})$. For each combination $(A_{1h(1)}, A_{2h(2)}, \ldots, A_{rh(r)})$, we calculate the value

$$\rho_{h(1),\ldots,h(r)} = \operatorname{card}(\operatorname{supp}(A_{1h(1)} \cdot A_{2h(2)} \cdot \cdots \cdot A_{rh(r)})) \quad (4)$$

which is the number of data points $(x_{i1}, x_{i2}, \ldots, x_{ir})$ for which $A_{1h(1)}(x_{i1}) \cdot A_{2h(2)}(x_{i2}) \cdot \cdots \cdot A_{rh(r)}(x_{ir}) > 0$. The user can use a minimal density ρ_ε of data, in the sense that

$$\rho_{h(1)\cdots h(r)} \geq \rho_\varepsilon, \quad (5)$$

and the fulfillment of this constraint is to determine a value for the parameter n which is consistent with the distribution of the data correspondent to the uniform fuzzy partitions. The use of ρ_ε allows us to control how the uniform fuzzy partition set of the attributes should be finer. If $\rho_\varepsilon = 0$, we obtain the more coarse-grained fuzzy partition set in accordance to this constraint. As examples, in Figures 2 and 3 we show the case of two attributes, X_1 and X_2. Each object of the dataset is represented as a point in the Cartesian graph of X_1 and X_2 and in both examples we consider $\rho_\varepsilon = 5$. In Figure 2 we have a too fine partition for $n = 8$; in fact $\rho_{h(1)h(2)} = 4 < \rho_\varepsilon = 5$ by considering the combination of

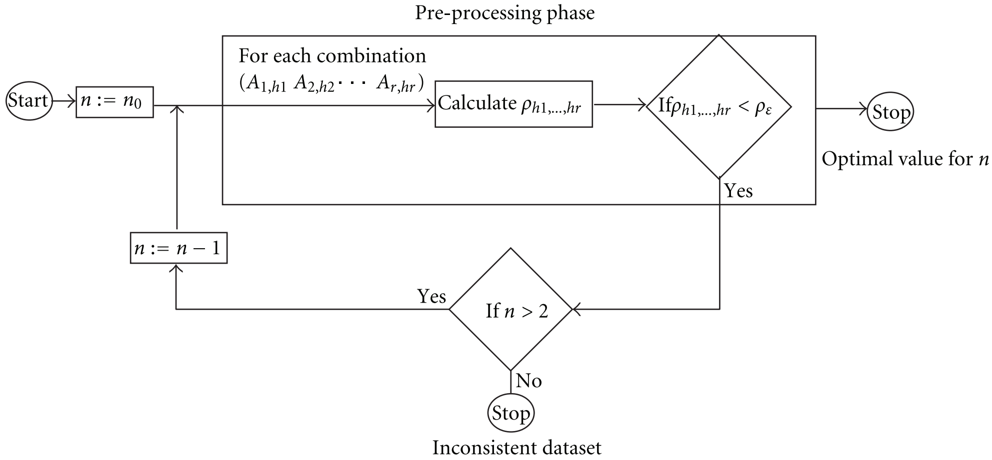


Figure 4: Schema of the pre-processing phase.

triangular fuzzy sets (A_{11}, A_{28}). In Figure 3 we have a more coarse-grained fuzzy partition for $n = 7$ which should be optimal. A finer partition set does not satisfy the constraint of minimum density of data and a coarser partition set would be under-sampled with respect to the dataset dimension. In this pre-processing phase we start with a value n_0 of the parameter n. If the partition set is too fine, we decrement the value of n by 1 until we determine the optimal partition. If $n \le 2$, the dataset is too coarse grained for the fuzzy rules extraction via F-transforms and the process is stopped (inconsistent dataset), otherwise we use the optimal partition in the successive step of fuzzy rules extraction. Figure 4 gives the schema of the pre-processing phase.

Following [31], in the extraction process we establish fuzzy association rules of the form (2) by calculating the support index as the percentage of objects in the dataset for which the antecedent in the fuzzy association rule (2) is not null, that is, $A_{1h(1)}(x_{i1}) \cdot A_{2h(2)}(x_{i2}) \cdot \cdots \cdot A_{rh(r)}(x_{ir}) > 0$ for $i = 1, \ldots, m$, and it is defined as

$$\sup_{h(1),\ldots,h(r)} = \frac{\rho_{h(1),\ldots,h(r)}}{m}, \tag{6}$$

where m is the dimension of the dataset. Then we calculate the confidence of each rule to evaluate the precision of a potential fuzzy association rule. Normally the confidence index is given by the ratio of the number of the objects in the dataset for which the antecedent and the consequent in the fuzzy association rule (2) are not null with respect to the number of objects in the data set for which the antecedent in the fuzzy association rule (2) is not null. In other words, we use the confidence index proposed in [31] given by

$$\mathrm{con}_{h(1),\ldots,h(k)} = \sqrt{\frac{\sum_{i=1}^{m}\left(H_n^F(o_i) - F_{h(1)h(2)\cdots h(k)}\right)^2 \cdot \left(A_{1h(1)}(p_{i1}) \cdot A_{2h(2)}(p_{i2}) \cdot \cdots \cdot A_{kh(k)}(p_{ik})\right)}{\sum_{i=1}^{m}\left(p_{iz} - F_{h(1)h(2)\cdots h(k)}\right)^2 \cdot \left(A_{1h(1)}(p_{i1}) \cdot A_{2h(2)}(p_{i2}) \cdot \cdots \cdot A_{kh(k)}(p_{ik})\right)}}, \tag{7}$$

where $H_n^F(o_i)$ is the value of the inverse F-transform applied on the ith data object and $F_{h(1)h(2)\cdots h(k)}$ is the fuzzy transform component associated with the fuzzy sets $(A_{1h(1)}, A_{2h(2)}, \ldots, A_{kh(k)})$. The multidimensional inverse F-Transform $H_n^F(o_i)$ has been already used in [31, 35] to model the dependency of the attribute X_z via the predictors $X_1, \ldots X_k$ like $X_z = H(X_1, \ldots, X_k)$, where H is a function estimated via a suitable fuzzy partition of the independent attribute domains. The formula (7) provides an estimate of the grade of precision of a potential fuzzy association rule. If the above index is equal to 1, then the error in the approximation of X_z obtained with the inverse fuzzy transform $H_n^F(o_i)$ is null.

In our framework we also use two sub-processes for extracting the fuzzy association rules. In the first sub-process we use the AprioriGen algorithm to extract the candidate fuzzy association rule with maximal dimension and support greater than or equal to a threshold $\sup_\varepsilon$. In the successive sub-process, the corresponding inverse fuzzy transform and the confidence index (7) are calculated for each fuzzy association rule candidate. The fuzzy association rule is extracted if the index (7) is greater than or equal to a threshold con_ε and in this case we determine the evaluative linguistic expression $\aleph$ with the component $F_{h(1)h(2)\cdots h(k)}$ of the F-transform corresponding to the fuzzy sets $(A_{1h(1)}, A_{2h(2)}, \ldots, A_{kh(k)})$ (this modality of assignment is described in Section 4 with many details). In Figure 5 we show the processes used for extracting fuzzy association rules in the form (2).

Summarizing, we can say that in the first step the AprioriGen algorithm is applied for determining the set of potential fuzzy association rules; in the successive step the direct, the inverse fuzzy transform, and the index "con"

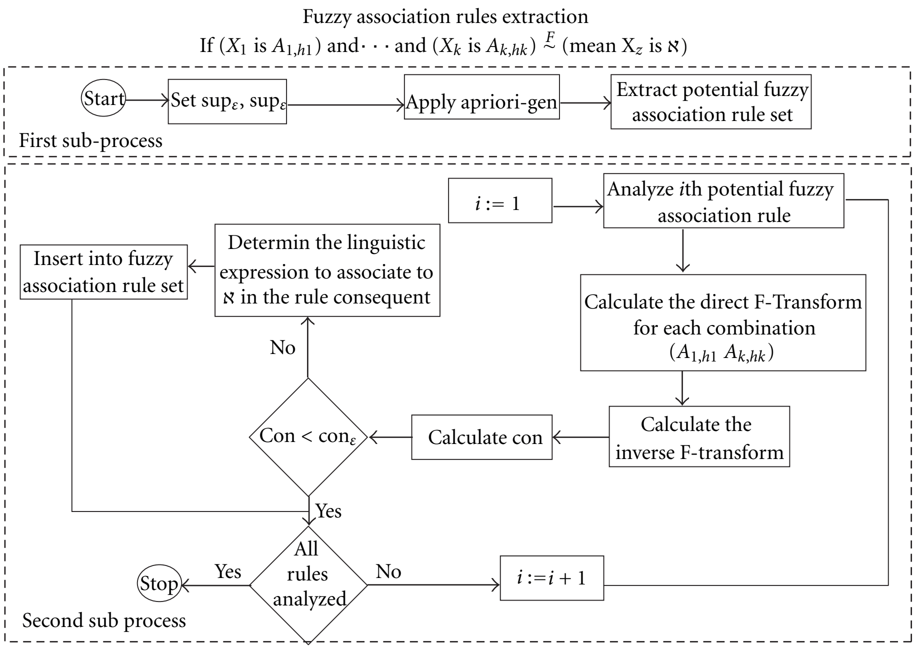


Figure 5: Extraction process of fuzzy association rules.

are calculated for each potential fuzzy association rule; if con $<$ con$_\varepsilon$ the potential fuzzy association rule is discarded, otherwise it is extracted and inserted like a strong fuzzy association rule in the final set of the rules.

3. Discrete *F*-Transforms in Several Variables

We firstly deal with functions of one variable and only the discrete case; indeed we know that a function f assumes determined values in the set of points $P = \{p_1, \ldots, p_m\}$ of the interval $[a, b]$, $a, b \geq 0$. If P is sufficiently dense with respect to the fixed partition $\{A_1, A_2, \ldots, A_n\}$, that is for each $i = 1, \ldots, n$ there exists an index $j \in \{1, \ldots, m\}$ such that $A_i(p_j) > 0$, we can define the n-tuple $[F_1, F_2, \ldots, F_n]$ as the discrete F-transform of f with respect to the basic functions $\{A_1, A_2, \ldots, A_n\}$, where each F_i is given by

$$F_{\mathrm{i}} = \frac{\sum_{j=1}^{m} f(p_j) A_i(p_j)}{\sum_{j=1}^{m} A_i(p_j)}, \tag{8}$$

for $i = 1, \ldots, n$. We also define the inverse F-transform of f with respect to the basic functions $\{A_1, A_2, \ldots, A_n\}$ by setting

$$f_{F,n}(p_j) = \sum_{i=1}^{n} F_i A_i(p_j) \tag{9}$$

for every $j \in \{1, \ldots, m\}$. We have the following theorem [32, Theorem 5].

Theorem 1. *Let $f(x)$ be a function assigned on the set $P = \{p_1, \ldots, p_m\} \subseteq [a, b]$, $a, b \geq 0$. Then for every $\varepsilon > 0$, there exists an integer $n(\varepsilon)$ and a related fuzzy partition $\{A_1, A_2, \ldots, A_{n(\varepsilon)}\}$ of $[a, b]$ such that P is sufficiently dense with respect to $\{A_1, A_2, \ldots, A_{n(\varepsilon)}\}$ and for every $p_j \in [a, b]$, $j = 1, \ldots, m$, the following inequality holds:*

$$\left| f(p_j) - f_{F,n(\varepsilon)}(p_j) \right| < \varepsilon. \tag{10}$$

Now we consider functions in k (≥ 2) variables. The universe of the discourse is given by the Cartesian product $[a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_k, b_k]$ where $[a_i, b_i]$ ($i = 1, \ldots, k$), $a_i, b_i \geq 0$, is the domain of the ith variable. Let $x_{11}, x_{12}, \ldots, x_{in(i)} \in [a_1, b_1], \ldots, x_{k1}, x_{k2}, \ldots, x_{kn(k)} \in [a_k, b_k]$ be $n(1) + \cdots + n(k)$ assigned points, called nodes, such that $x_{i1} = a_i < x_{i2} < \cdots < x_{in} = b_i$ for $i = 1, \ldots, k$. Furthermore, let $\{A_{i1}, A_{i2}, \ldots, A_{in(i)}\}$ be a fuzzy partition of $[a_i, b_i]$ for $i = 1, \ldots, k$. We assume that the function $f(x_1, x_2, \ldots, x_k)$ takes values in the set $P = \{(p_{11}, p_{12}, \ldots, p_{1k}), (p_{21}, p_{22}, \ldots, p_{2k}), \ldots, (p_{m1}, p_{m2}, \ldots, p_{mk})\}$, that is, in m points $p_j = (p_{j1}, p_{j2}, \ldots, p_{jk}) \in [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_k, b_k]$ for $j = 1, \ldots, m$. We say that P is sufficiently dense with respect to the chosen partitions $\{A_{11}, A_{12}, \ldots, A_{1n(1)}\}, \ldots, \{A_{k1}, A_{k2}, \ldots, A_{kn(k)}\}$ if for each k-tuple $\{h(1), \ldots, h(k)\} \in \{1, \ldots, n(1)\} \times \cdots \times \{1, \ldots, n(k)\}$, there exists a point $p_j = (p_{j1}, p_{j2}, \ldots, p_{jk}) \in P$ such that $A_{1h(1)}(p_{j1}) \cdot A_{2h(2)}(p_{j2}) \cdot \cdots \cdot A_{kh(k)}(p_{jk}) > 0$, $j \in \{1, \ldots, m\}$. In this case we define the function $F_{h(1)\cdots h(k)}$ as the $(h(1), \ldots, h(k))$th component of the discrete F-transform of f with respect to $\{A_{11}, A_{12}, \ldots, A_{1n(1)}\}, \ldots, \{A_{k1}, A_{k2}, \ldots, A_{kn(k)}\}$ given by

$$F_{h(1)h(2)\cdots h(k)} = \frac{\sum_{j=1}^{m} f\left(p_{j1}, p_{j2}, \dots p_{jk}\right) \cdot A_{1h(1)}\left(p_{j1}\right) \cdot A_{2h(2)}\left(p_{j2}\right) \cdot \cdots \cdot A_{kh(k)}\left(p_{jk}\right)}{\sum_{j=1}^{m} A_{1h(1)}\left(p_{j1}\right) \cdot A_{2h(2)}\left(p_{j2}\right) \cdot \cdots \cdot A_{kh(k)}\left(p_{jk}\right)}. \quad (11)$$

Now we define the inverse F-transform of f with respect to the basic functions $\{A_{11},A_{12},\dots,A_{1n(1)}\},\dots,\{A_{k1}, A_{k2},\dots,A_{kn(k)}\}$ to be the function defined by setting for each $p_j = (p_{j1},p_{j2},\dots,p_{jk}) \in [a_1, b_1] \times \cdots \times [a_k, b_k]$:

$$\begin{aligned} f^F_{n(1)\cdots n(k)}\left(p_{j1}, p_{j2}, \dots, p_{j,k}\right) &= \sum_{h(1)=1}^{n(1)} \sum_{h(2)=1}^{n(2)} \cdots \sum_{h(k)=1}^{n(k)} F_{h(1)h(2)\cdots h(k)} A_{1h(1)}\left(p_{j1}\right) \\ &\quad \cdot A_{2h(2)} \cdot \cdots \cdot A_{kh(k)}\left(p_{jk}\right) \end{aligned} \quad (12)$$

for $j = 1,\dots,m$. As in [32], it is possible to prove the following generalization of Theorem 1.

Theorem 2. *Let $f(x_1, x_2, \dots, x_k)$ be a function assigned on the set of points $P = \{(p_{11}, p_{12}, \dots, p_{1k}), (p_{21}, p_{22}, \dots, p_{2k}), \dots, (p_{m1}, p_{m2}, \dots, p_{mk})\} \subseteq [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_k, b_k]$, $a_i, b_i \geq 0$. Then for every $\varepsilon > 0$, there exist k integers $n(1, \varepsilon), \dots, n(k, \varepsilon)$ and related fuzzy partitions*

$$\{A_{11}, A_{12}, \dots, A_{1n(1,\varepsilon)}\}, \dots, \{A_{k1}, A_{k2}, \dots, A_{kn(k,\varepsilon)}\} \quad (13)$$

such that the set P is sufficiently dense with respect to fuzzy partitions (13) and for every $p_j = (p_{j1}, p_{j2}, \dots, p_{jk}) \in P$, $j = 1, \dots, m$, the following inequality holds:

$$\begin{aligned} &\Big| f\left(p_{j1}, p_{j2}, \dots, p_{jk}\right) \\ &\quad - f^F_{n(1,\varepsilon)n(2,\varepsilon)} \cdots {}_{n(k,\varepsilon)}\left(p_{j1}, p_{j2}, \dots, p_{jk}\right) \Big| < \varepsilon. \end{aligned} \quad (14)$$

Strictly speaking, Theorems 1 and 2 assure that a discrete (even multi-dimensional) function can be approximated arbitrarily with a suitable inverse (multidimensional) F. transform provided that a convenient fuzzy partition of the universe of discourse is found via related basic functions. Like pointed out in our previous papers [35, 36, 38–40], unfortunately Theorem 1 (resp. Theorem 2) is not constructive, in the sense that it does not give a tool to find an integer $n(\varepsilon)$(resp., k integers $n(1, \varepsilon), \dots, n(k, \varepsilon)$) and basic functions $\{A_1, A_2, \dots, A_{n(\varepsilon)}\}$ (resp., (13)) such that (10) (resp., (14)) holds for an arbitrary $\varepsilon > 0$. In a practical sense, we assume several values of n (resp., $n(1), \dots, n(k)$) testing that the set P is sufficiently dense with respect to the related fuzzy partition (resp., partitions) and then we use the two indexes (6) and (7) for controlling the choice of the best fuzzy partition (resp., partitions).

4. Fuzzy Association Rules Extraction Process

We use the multi-dimensional direct and inverse F-transform for extracting fuzzy association rules in the form (2) from the dataset represented as in Table 1, where $(A_{i1}, \dots, A_{i,n}), i = 1, \dots, k$, is the uniform fuzzy partition of the context $w_i = [a_i, b_i]$ of the given attribute X_i by setting $a_i = \min\{p_{1i}, \dots, p_{mi}\}$ and $b_i = \max\{p_{1i}, \dots, p_{mi}\}$, with the basic functions (3). For computational simplicity, we assume constant the cardinality of the fuzzy partition of each contexts w_i, that is card $w_i = n, i = 1, \dots, k$. Let $F_{h(1)h(2)\cdots h(k)}$ the multi-dimensional direct F-transform defined by (11) be corresponding to the combination $(A_{1h(1)}, \dots, A_{kh(k)})$ and if p_{jz} is the given (expected) value, then the formula (11) is reduced to the following one:

$$F_{h(1)h(2)\cdots h(k)} = \frac{\sum_{j=1}^{m} p_{jz} \cdot A_{1h(1)}\left(p_{j1}\right) \cdot \cdots \cdot A_{kh(k)}\left(p_{jk}\right)}{\sum_{j=1}^{m} A_{1h(1)}\left(p_{j1}\right) \cdot \cdots \cdot A_{kh(k)}\left(p_{jk}\right)}. \quad (15)$$

In other words, $F_{h(1)h(2)\cdots h(k)}$ is a mean of the values of the attribute X_z weighted over $(A_{1h(1)}, \dots, A_{kh(k)})$. Following [18, 29], ℵ is given by the combination of one of the linguistic following hedges: Ex (extremely), Si (significantly), Ve (very), empty hedge, ML (more or less), Ro (roughly), QR (quite roughly), VR (very roughly) with one of the following expressions: Sm (small), Me (medium), Bi (big). Each linguistic hedge is modelled with a continuous function ν_{abc} defined by means of three parameters a, b, c, with $0 \leq a < b < c \leq 1$, as

$$\nu_{abc}(y) = \begin{cases} 0 & \text{if } 0 \leq y < a \\ \dfrac{(y-a)^2}{(b-a)(c-a)} & \text{if } a \leq y \leq b \\ 1 - \dfrac{(c-y)^2}{(c-b)(c-a)} & \text{if } b < y < c \\ 1 & \text{if } 1 \geq y \geq c. \end{cases} \quad (16)$$

The fuzzy sets of each combination of linguistic hedges with one of the expressions "Small," "Medium," "Big" are defined by

$$\begin{aligned} \text{Int}(\langle \text{linguistichedge} \rangle \text{small}) &= \nu_{abc}(\text{LH(x)}), \\ \text{Int}(\langle \text{linguistichedge} \rangle \text{medium}) &= \nu_{abc}(\text{MH(x)}), \\ \text{Int}(\langle \text{linguistichedge} \rangle \text{big}) &= \nu_{abc}(\text{RH(x)}), \end{aligned} \quad (17)$$

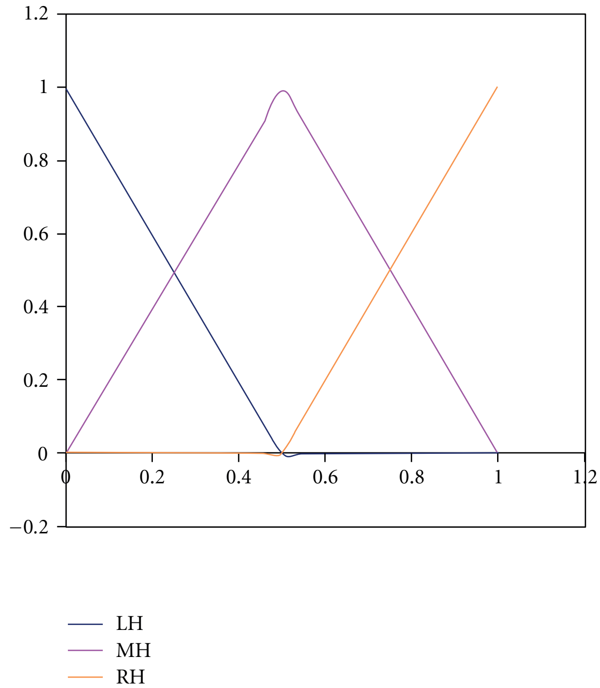


Figure 6: The functions LH, MH, and RH.

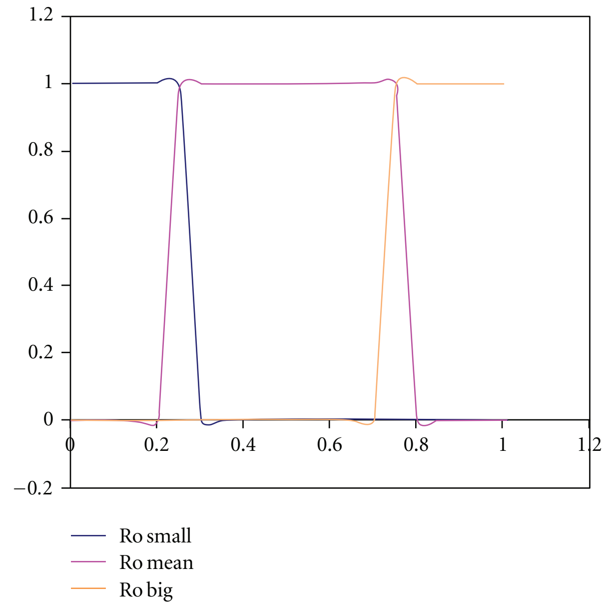


Figure 7: Ro small, Ro medium, and Ro big ($a = 0.49, b = 0.5, c = 0.51$).

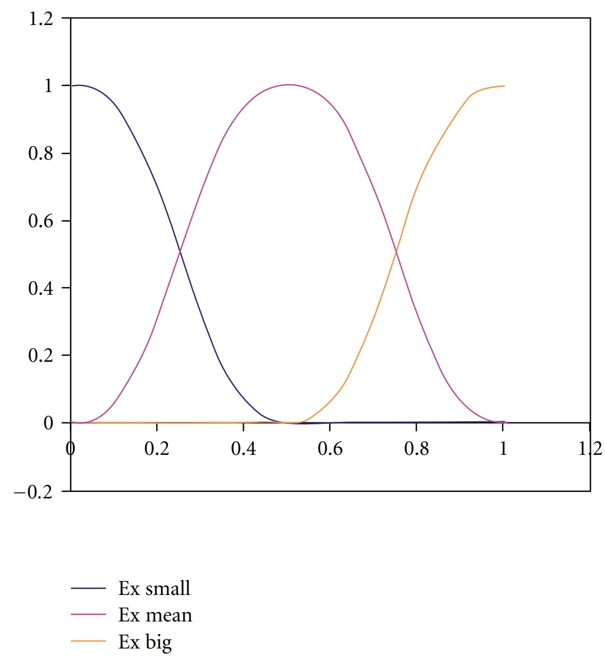


Figure 8: “Ex small,” “Ex medium,” and “Ex big” ($a = 0.03, b = 0.5, c = 0.96$).

where the locution “Int” stands for the intensity of the linguistic expression and

$$\mathrm{LH}(x) = \begin{cases} \dfrac{0.5 - x}{0.5} & \text{if } 0 \leq x \leq 0.5 \\ 0 & \text{if } 1 \geq x > 0.5, \end{cases}$$
$$\mathrm{MH}(x) = \begin{cases} \dfrac{x}{0.5} & \text{if } 0 \leq x \leq 0.5 \\ \dfrac{1 - x}{0.5} & \text{if } 1 \geq x > 0.5, \end{cases} \tag{18}$$
$$\mathrm{RH}(x) = \begin{cases} 0 & \text{if } 0 \leq x \leq 0.5 \\ \dfrac{x - 0.5}{0.5} & \text{if } 1 \geq x > 0.5. \end{cases}$$

The extension of a linguistic expression is obtained via a simple linear transformation defined, considering the context $w_z = [a_z, b_z]$ of the attribute X_z, as

$$x = \frac{(s - a_z)}{(b_z - a_z)}, \quad s \in [a_z, b_z]. \tag{19}$$

The linguistic expression ℵ represents a weighted mean of the values of the attribute X_z, in which the weights are given by the membership values of the basic functions. In Figure 6 we show the three linear functions LH, MH, RH.

As example, in Figure 7 (resp., Figure 8) we show the fuzzy sets “Ro small,” “Ro medium,” and “Ro big” (resp., “Ex small,” “Ex medium” and “Ex big”) determined by setting $a = 0.49, b = 0.5, c = 0.51$ (resp., $a = 0.03, b = 0.5, c = 0.96$). In other words, the experts can assign specific labels to the linguistic expressions which are representative of their reasoning, or can use the same label for more fuzzy sets. For example, to the fuzzy set “Ex medium” they can associate the label ”perfectly on the average”. In accordance to [18, 29], we define a partial ordering “≤” in the set of the linguistic hedges as

$$\begin{aligned} &\mathrm{Ex}\langle \exp \rangle \leq \mathrm{Si}\langle \exp \rangle \leq \mathrm{Ve}\langle \exp \rangle \leq \text{empty hedge}\langle \exp \rangle \\ &\quad \leq \mathrm{ML}\langle \exp \rangle \leq \mathrm{Ro}\langle \exp \rangle \leq \mathrm{QR}\langle \exp \rangle \leq \mathrm{VR}\langle \exp \rangle, \end{aligned} \tag{20}$$

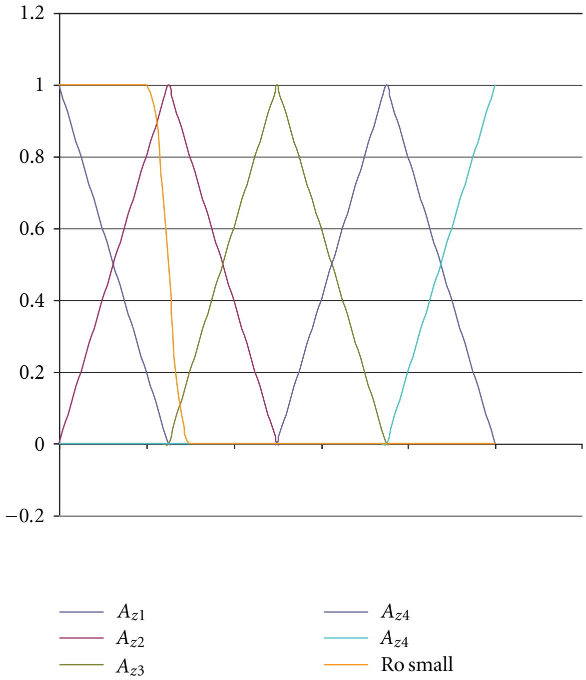


FIGURE 9: Inclusion areas for the fuzzy association rule $R2$.

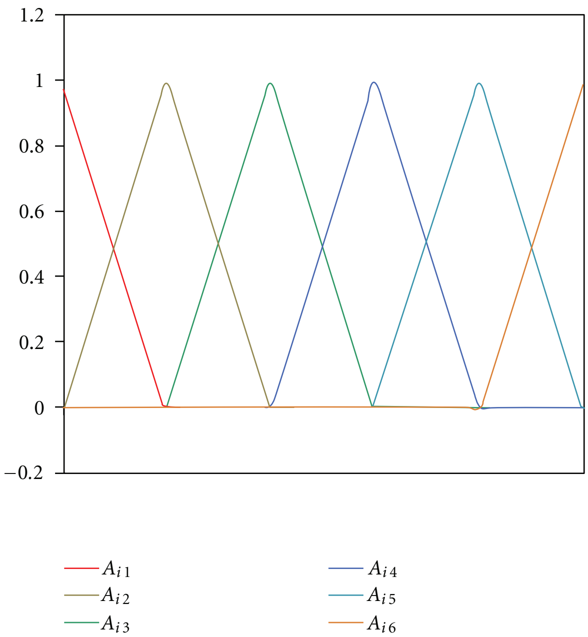


FIGURE 10: Inclusion areas for the fuzzy association rule $R3$.

where ⟨exp⟩ is one of the following expressions: "small," "medium" or "big". As mentioned above, if we obtain the same membership degree for two or more linguistic expressions, we assign to ℵ the linguistic expression of the "lowest fuzzy set," which is the sharpest evaluative expression with respect to the partial ordering "≤." For example, if we have $x = 1$, we assign the linguistic expression "Ex big" to ℵ because "Ex big" is the lowest fuzzy set among all the fuzzy

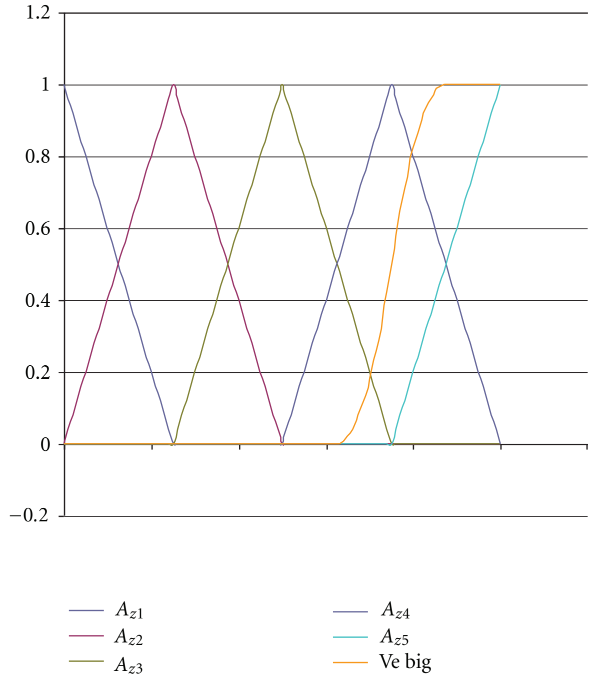


FIGURE 11: Basic functions used for the residential buildings dataset.

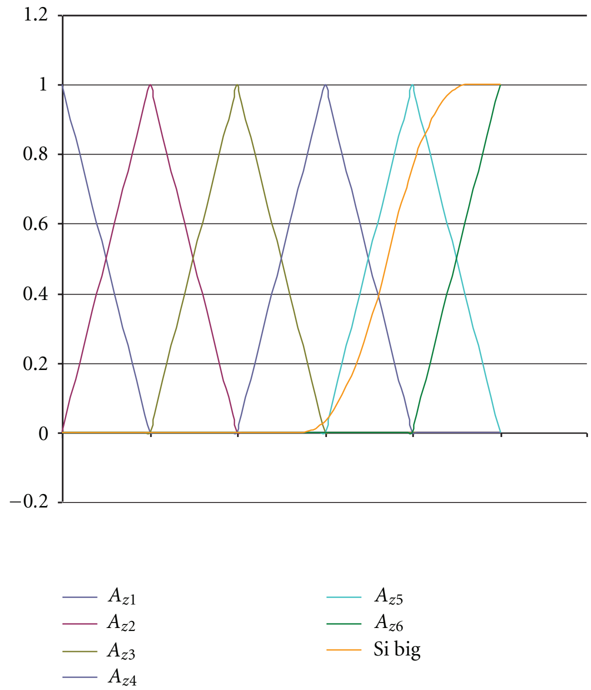


FIGURE 12: Inclusion areas for the fuzzy association rule $R1$.

sets "⟨hedge⟩ big" (like "Ex big" and "Ro big") assuming the value 1. In order to use the formulae (18), the dataset has to be sufficiently dense with respect to the chosen fuzzy partitions of basic functions.

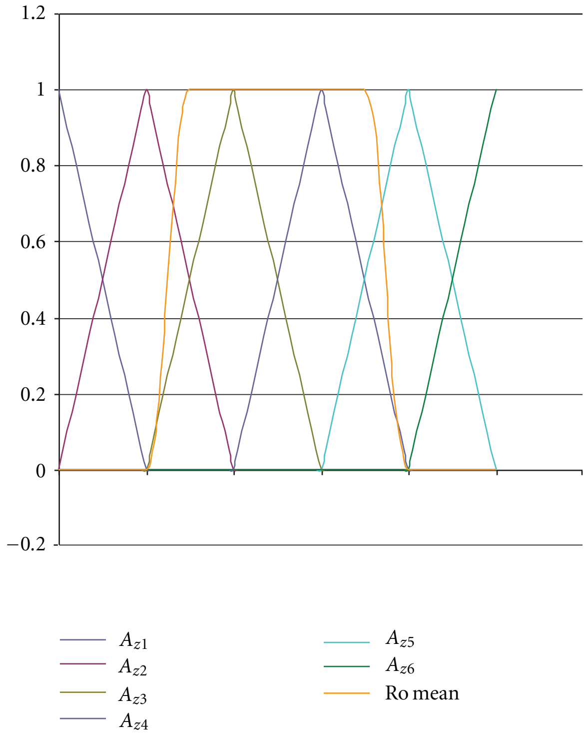


Figure 13: Inclusion areas for the fuzzy association rule $R2$.

We set the optimal value of the parameter n using the pre-processing phase schematized in Figure 4, in which we control that the number of data points $p_j = (p_{j1}, p_{j2}, \ldots, p_{jr})$ such that $A_{1h(1)}(p_{j1}) \cdot A_{2h(2)}(p_{j2}) \cdot \cdots \cdot A_{kh(r)}(p_{jr}) \geq \rho_\varepsilon$ for each combination $(A_{1h(1)}, A_{2h(2)}, \ldots, A_{rh(r)})$, where ρ_ε is a prefixed threshold otherwise the cardinality n of each fuzzy partition is decremented and the process is iterated. In this mode we impose the fulfilment of the sufficient density of the data points with respect to the fuzzy partition's set and we are sure that the fuzzy partition created is too coarse grained. We set the initial value of the cardinality of the fuzzy partitions to n_0.The pseudocode of the algorithm of the pre-processing phase is reported below.

(1) Set $n := n_0$

(2) Set the minimal point data density

(3) For each combination $(A_{1h(1)}, A_{2h(2)}, \ldots, A_{r,h(r)})$,

  (a) calculate the value $\rho_{h(1),\ldots,h(r)}$ defined in (4)

  (b) If $\rho_{h(1),\ldots,h(r)} < \rho_\varepsilon$, then

    (i) $n := n - 1$

    (ii) if $n < 2$ exit (-1)

    (iii) exit for

  (c) end if

(4) Next

(5) Return n

The successive extraction process of the fuzzy association rules is composed by two sub-processes, schematized in Figure 5. In the first sub-process we use the AprioriGen algorithm for selecting the candidate fuzzy association rule and for choosing the antecedents with maximal dimension and support greater than or equal to the threshold $\sup_\varepsilon$. The AprioriGen algorithm is composed by two steps: the join and the pruning step. In the join step an itemset is generated and formed by k attributes merging two $(k-1)$-itemsets having the same first $(k-2)$ attributes. In the pruning step all the elements not having all the first $(k-1)$ subsets as great are deleted. In the successive sub-process the fuzzy association rule is extracted if the grade of confidence (7) is greater than or equal to the threshold con_ε and it is calculated via the inverse F-transform [31, 35] given from the following formula (similar to (12)):

Table 3: Parameters for fuzzy sets associated to the linguistic expressions.

Linguistic expression	a	b	c
Ex ⟨exp⟩	0.03	0.50	0.96
Si ⟨exp⟩	0.10	0.50	0.85
Ve ⟨exp⟩	0.25	0.50	0.75
Emptyhedge ⟨exp⟩	0.30	0.50	0.70
ML ⟨exp⟩	0.35	0.50	0.65
Ro ⟨exp⟩	0.40	0.50	0.60
QR ⟨exp⟩	0.45	0.50	0.55
VR ⟨exp⟩	0.49	0.50	0.51

$$H_n^F\left(p_j\right) = \sum_{l(1)=1}^{n} \sum_{l(2)=1}^{n} \cdots \sum_{l(k)=1}^{n} F_{l(1)l(2)\ldots l(k)} \cdot A_{1l(1)}\left(p_{j1}\right) \cdot \cdots \cdot A_{kl(k)}\left(p_{jk}\right), \tag{21}$$

obtained considering all the multi-dimensional F-transform components $F_{l(1)l(2)\cdots l(k)}$ of the combination of basic functions $(A_{1l(1)}, \ldots, A_{kl(k)})$ $(l(i) = 1, \ldots, n)$. If the confidence index (7) is greater or equal than the threshold con_ε, the F-transform component correspondent to the basic functions $(A_{1l(1)}, \ldots, A_{kl(k)})$ in the antecedent of the potential fuzzy association rule is used for determining the linguistic expression of the consequent $\aleph$ of the final fuzzy association rule. If we obtain the same membership degree for two or more linguistic expressions, we assign the linguistic expression of the lowest fuzzy sets to $\aleph$ according to the above partial ordering "$\leq$." The related pseudo-code is reported below.

(1) Set $\sup_\varepsilon$

(2) Apply AprioriGen algorithm

(3) Return the potential fuzzy association rule set

(4) Next

(5) For each combination $(A_{1l(1)}, \ldots, A_{kl(k)})$, calculate the direct F-transform component $F_{l(1)l(2)\ldots l(k)}$

(6) Next

(7) For each data object p_j, calculate the inverse F-transform $H_n^F(p_j)$

Table 4: Fuzzy rules extracted from the resident dataset.

id	Fuzzy association rule	sup	con
$R1$	If (X_4 is A_{42}) $\overset{F}{\sim}$ (mean X_2 is Ve small)	0.44	0.23
$R2$	If (X_4 is A_{42}) $\overset{F}{\sim}$ (mean X_6 is Ro small)	0.44	0.23
$R3$	If (X_2 is A_{22}) and (X_5 is A_{54}) $\overset{F}{\sim}$ (mean X_4 is Ve big)	0.23	0.15
$R4$	If (X_2 is A_{24}) and (X_6 is A_{63}) $\overset{F}{\sim}$ (mean X_8 is QR small)	0.21	0.15

Table 5: Percentages of the confidence index for the fuzzy association rules.

Fuzzy association rule	X_z	Percentages				
		A_{z1}	A_{z2}	A_{z3}	A_{z4}	A_{z5}
$R1$	X_2	98	53	5	0	0
$R2$	X_6	97	47	1	1	0
$R3$	X_4	0	1	6	55	99
$R4$	X_8	97	48	1	0	0

Table 6: Percentages of the confidence index for the fuzzy association rules.

Fuzzy association rule	Fuzzy set in the consequent	Percentages of inclusion				
		A_{z1}	A_{z2}	A_{z3}	A_{z4}	A_{z5}
$R1$	Ve small	100	54	4	0	0
$R2$	Ro small	100	51	1	0	0
$R3$	Ve big	0	0	4	54	100
$R4$	Qr small	100	50	0	0	0

(8) Next

(9) Calculate the confidence index (7) and call it as "con"

(10) If con $\geq$ con$_\varepsilon$, the F-transform component $F_{l(1)l(2)\dots l(k)}$ is to be assigned to $\aleph$

(11) End if

(12) Insert the fuzzy association rule in the fuzzy association rule set

(13) Next

(14) Return the fuzzy association rule set

In Section 5 we present some results obtained from datasets relative to the 2001 census ISTAT (Istituto Nazionale di Statistica) concerning the municipalities of the district of Naples (Italy).

5. A Simulation Result

We consider a first dataset of the last ISTAT census database of the municipalities of the district of Naples (Italy). This dataset is obtained by extracting the information about residents with job and families. We use the following notation: X_1 stands for census code, X_2 stands for the percentage of not employed, X_3 for the percentage of managers and professional men, X_4 stands for the percentage of women employed, X_5 stands for the percentage of graduate employed, X_6 stands for the percentage of residential houses of property, X_7 stands for the percentage of families with two or more houses of propriety, and finally X_8 stands for the percentage of families with more than two sons. In the pre-processing phase we set $\rho_\varepsilon = 5$, obtaining $n = 5$ as optimal cardinality partition of each attribute. Each fuzzy partition is uniform and constructed by using five triangular fuzzy sets of the form (3). We set both values of sup$_\varepsilon$ and con$_\varepsilon$ to 0.1. The domain's expert has suggested the values reported in Table 3 for the parameters *a, b, c*.

After the extraction process we obtain four fuzzy association rules (cfr. Table 4). To verify the reliability of these results, we report the values (as percentages) of the confidence index given by (7), obtained for each basic function of the attribute $X_z, z = 2, 4, 6, 8$, in the consequent of the extracted fuzzy association rules in Table 5. The linguistic expression in the consequents of the fuzzy association rules in Table 4 can be roughly interpreted as a mean of the fuzzy sets $A_{z1}, \dots, A_{z5}$ weighted for the value of the confidence index. In Table 5 we report the percentages of the inclusion areas of each basic function with the fuzzy set associated to the linguistic expression in the consequent with respect to the area of the basic function. Each inclusion area is the area given from the intersection between the basic function and the fuzzy set.

From the comparison of Tables 5 and 6, the confidence index is approximately similar to the correspondent percentage of inclusion. In Figure 9 (resp., Figure 10) we show graphically the inclusion areas for the association rule $R2$ (resp., $R3$). The fuzzy set associated with the linguistic expression in the consequent is in orange colour. Then we can state that the fuzzy association rules extracted in Table 4 can be interpreted as a coarse-grained fuzzy association rules in which the linguistic expression in the consequent approximates a mean of finer fuzzy set given by the basic functions (3) for the attribute A_z. This approximation depends clearly on the the values of *a, b, c*.

The next dataset consists of attributes describing characteristics of residential buildings and houses. The notation concerning the attributes is the following: X_1 stands for census code, X_2 stands for percentage of residential buildings constructed during the last 5 years, X_3 stands for percentage of residential buildings with maintenance during the last 5 years, X_4 stands for mean year of last maintenance, X_5 stands for mean number of residential houses, X_6 stands for mean surface of residential houses, X_7 stands for percentage of residential houses whit central heating. In the pre-processing phase we set $\rho_\varepsilon = 5$, obtaining $n = 6$ as optimal cardinality partition of each attribute. In Figure 11 we show the six basic functions of the form (3) which give the uniform fuzzy partition of each context. After the extraction process we obtain three fuzzy association rules (cfr. Table 7).

Table 7: Fuzzy extraction rules extracted from the residential buildings dataset.

Id	Fuzzy association rule	Support	Confidence index
$R1$	If (X_2 is A_{26}) $\overset{F}{\sim}$ (mean X_7 is Si big)	0.19	0.58
$R2$	If (X_2 is A_{22}) $\overset{F}{\sim}$ (mean X_3 is Ro mean)	0.33	0.27
$R3$	If (X_5 is A_{52}) and (X_6 is A_{65}) $\overset{F}{\sim}$ (mean X_2 is QR small)	0.14	0.17

Table 8: Percentages of the confidence index for the fuzzy association rules.

Fuzzy association rule	X_z	Confidence index					
		A_{z1}	A_{z2}	A_{z3}	A_{z4}	A_{z5}	A_{z6}
$R1$	X_7	0	0	1	20	78	97
$R2$	X_3	0	35	96	95	31	1
$R3$	X_2	96	75	6	1	0	0

Table 9: Percentages of inclusion obtained for the fuzzy association rules.

Fuzzy association rule	Fuzzy set in the consequent	Percentages of inclusion					
		A_{z1}	A_{z2}	A_{z3}	A_{z4}	A_{z5}	A_{z6}
R1	Si big	0	0	0	19	82	100
R2	Ro mean	0	32	100	100	32	0
R3	QR small	100	74	5	0	0	0

Then we calculate the confidence index (7) for all the antecedents of the fuzzy association rules extracted in Table 6. In Table 8 we report as percentages the values of the confidence index obtained for each basic function of the attribute X_z in the consequent of the extracted fuzzy association rules.

In Table 9 we report the inclusion areas of each basic function with the fuzzy set associated to the linguistic expression in the consequent with respect to the area of the basic function.

By comparing Tables 8 and 9, we can state that also for this dataset the linguistic expression in the fuzzy association rules can be roughly interpreted as a mean of the fuzzy sets $A_{z1}, \ldots, A_{z6}$ weighted from the value of the confidence index. In Figure 12 (resp., Figure 13) we show graphically the inclusion areas for the association rule $R1$ (resp., $R2$). The fuzzy set associated with the linguistic expression in the consequent is represented in orange colour.

The results confirm that the F-transforms can be used to extract fuzzy association rules in the form (2) in a coarse-grained view from datasets. The comparison of the results suggests that the linguistic evaluation used for the attribute A_z in the consequent and calculated, using the inverse F-transform, can be estimated as a weighted mean of the finer fuzzy sets composed from basic functions (4), where the weights are given from confidence index (7).

6. Conclusions

We propose the usage of multi-dimensional F-transforms which allow to extract fuzzy association rules from datasets in a coarse-grained form. Our approach allows always to control that the set of the assigned points is sufficiently dense respect to the basic functions of the partition and we use the support and confidence indexes for selecting and analyzing fuzzy association rules.

This method can be used in data mining processes in which a fine exploration of fuzzy association rules between attributes in the datasets is not necessary. In a future work the authors intend to explore the performances of these methods for very large datasets and compare the results with the ones obtained by using other well-known existing methods such as clustering- and evolutionary-based ones.

References

[1] P. Hajek and T. Havranek, *Mechanizing Hypothesis Formation*, Mathematical Foundations for a General Theory, Springer, Berlin, Germany, 1978.

[2] K. C. C. Chan and W. H. Au, "Mining fuzzy association rules," in *Proceedings of the 6th International Conference on Information and Knowledge Management (CIKM '97)*, pp. 209–215, Las Vegas, Nev, USA, November 1997.

[3] W. H. Au and K. C. C. Chan, "Effective algorithm for discovering fuzzy rules in relational databases," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 1314–1319, Anchorage, Alaska, USA, May 1998.

[4] K. C. C. Chan and W. H. Au, "An effective algorithm for discovering fuzzy rules in relational databases," in *Proceedings of IEEE World Congress on Computational Intelligence*, vol. 2, pp. 1313–1319, Honolulu, Hawaii, USA, 2002.

[5] G. Chen, Q. Wei, and E. Kerre, "Fuzzy data mining: discovery of fuzzy generalized association rules," in *Recent Research Issues on Management of Fuzziness in Association Rules*, G. Bordogna and G. Pasi, Eds., pp. 45–66, Springer, Berlin, Germany, 2000.

[6] G. Chen and Q. Wei, "Fuzzy association rules and the extended mining algorithms," *Information Sciences*, vol. 147, no. 1–4, pp. 201–228, 2002.

[7] C. M. Kuok, A. Fu, and M. H. Wong, "Mining fuzzy association rules in databases," *SIGMOD Record*, vol. 27, no. 1, pp. 41–46, 1998.

[8] M. De Cock, C. Cornelis, and E. E. Kerre, "Fuzzy association rules: a two-sided approach," in *Proceedings of the International Conference on Fuzzy Information Processing, Theories and Applications*, pp. 385–390, Beijing, China, 2003.

[9] M. Delgado, N. Marín, D. Sánchez, and M. A. Vila, "Fuzzy association rules: general model and applications," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 2, pp. 214–225, 2003.

[10] M. Delgado, N. Manin, M. J. Martín-Bautista, D. Sanchez, and M. A. Vila, "Mining fuzzy association rules: an overview," in *Studies in Fuzziness and Soft Computing*, vol. 164, pp. 351–373, Springer, Berlin, Germany, 2005.

[11] D. Dubois, H. Prade, and T. Sudkamp, "On the representation, measurement, and discovery of fuzzy associations," *IEEE*

Transactions on Fuzzy Systems, vol. 13, no. 2, pp. 250–262, 2005.

[12] D. Dubois, E. Hüllermeier, and H. Prade, "A systematic approach to the assessment of fuzzy association rules," *Data Mining and Knowledge Discovery*, vol. 13, no. 2, pp. 167–192, 2006.

[13] A. Gyenesei and J. Teuhola, "Multidimensional fuzzy partitioning of attribute ranges for mining quantitative data," *International Journal of Intelligent Systems*, vol. 19, no. 11, pp. 1111–1126, 2004.

[14] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan-Kaufman, New York, NY, USA, 2nd edition, 2006.

[15] T. P. Hong and Y. C. Lee, "An overview of mining fuzzy association rules," *Studies in Fuzziness and Soft Computing*, vol. 220, pp. 397–410, 2008.

[16] R. J. Miller and Y. Yang, "Association rules over interval data," *SIGMOD Record*, vol. 26, no. 2, pp. 452–461, 1997.

[17] S. Mitra, S. K. Pal, and P. Mitra, "Data mining in soft computing framework: a survey," *IEEE Transactions on Neural Networks*, vol. 13, no. 1, pp. 3–14, 2002.

[18] V. Novák, I. Perfilieva, A. Dvořák, G. Chen, Q. Wei, and P. Yan, "Mining pure linguistic associations from numerical data," *International Journal of Approximate Reasoning*, vol. 48, no. 1, pp. 4–22, 2008.

[19] G. Piatecky-Shapiro and W. J. Frawley, *Knowledge Discovery in Databases*, AAAI Press/MIT Press, Cambridge, UK, 1991.

[20] H. Shen and S. Horiguchi, "Mining quantitative fuzzy association rules," in *Encyclopedia of Data Warehousing and Mining*, J. Wang, Ed., pp. 815–819, IGI Global, 2005.

[21] W. Shu and L. Ding, "Efficient data mining approach for fuzzy association rules," *Journal of Software*, vol. 6, no. 1, pp. 91–99, 2011.

[22] R. Alcalá, J. Alcalá-Fdez, and F. Herrera, "A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 4, pp. 616–635, 2007.

[23] R. Alhajj and M. Kaya, "Multi-objective genetic algorithms based automated clustering for fuzzy association rules mining," *Journal of Intelligent Information Systems*, vol. 31, no. 3, pp. 243–264, 2008.

[24] K. Hirota and W. Pedrycz, "Fuzzy computing for data mining," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1575–1600, 1999.

[25] M. Kaya, R. Alhajj, F. Polat, and A. Arslan, "Efficient automated mining of fuzzy association rules," in *Proceedings of the 13th International Conference on Database and Expert Systems Applications*, pp. 133–142, Springer, 2002.

[26] M. Kaya and R. Alhajj, "Genetic algorithm based framework for mining fuzzy association rules," *Fuzzy Sets and Systems*, vol. 152, no. 3, pp. 587–601, 2005.

[27] A. W. Fu, M. H. Wong, S. C. Sze, W. C. Wong, W. L. Wong, and W. K. Yu, "Finding fuzzy sets for the mining of fuzzy association rules for numerical attributes," in *Proceedings of the 1st International Symposium on Intelligent Data Engineering and Learning*, pp. 263–268, Hong Kong, 1998.

[28] A. Mangalampalli and V. Pudi, "Fuzzy association rule mining algorithm for fast and efficient performance on very large datasets," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 1163–1168, IEEE Press, Jeju Island, Korea, August 2009.

[29] V. Novák, "A comprehensive theory of trichotomous evaluative linguistic expressions," *Fuzzy Sets and Systems*, vol. 159, no. 22, pp. 2939–2969, 2008.

[30] W. Ouyang and Q. Huang, "Mining direct and indirect weighted fuzzy association rules in large transaction databases," in *Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '09)*, vol. 3, pp. 128–132, IEEE Press, Tianjin, China, August 2009.

[31] I. Perfilieva, V. Novàk, and A. Dvoràk, "Fuzzy transform in the analysis of data," *International Journal of Approximate Reasoning*, vol. 48, no. 1, pp. 36–46, 2008.

[32] I. Perfilieva and B. De Baets, "Fuzzy transforms of monotone functions with application to image compression," *Information Sciences*, vol. 180, no. 17, pp. 3304–3315, 2010.

[33] I. Perfilieva, "Fuzzy transforms: theory and applications," *Fuzzy Sets and Systems*, vol. 157, no. 8, pp. 993–1023, 2006.

[34] I. Perfilieva, V. Novák, V. Pavliska, A. Dvořák, and M. Štěpnička, "Analysis and prediction of time series using fuzzy transform," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '08)*, pp. 3876–3880, Hong Kong, June 2008.

[35] F. Di Martino, V. Loia, and S. Sessa, "Fuzzy transforms method and attribute dependency in data analysis," *Information Sciences*, vol. 180, no. 4, pp. 493–505, 2010.

[36] F. Di Martino, V. Loia, and S. Sessa, "Fuzzy transforms method in prediction data analysis," *Fuzzy Sets and Systems*, vol. 180, no. 1, pp. 146–163, 2011.

[37] T. P. Hong, C. H. Chen, Y. L. Wu, and Y. C. Lee, "A GA-based fuzzy mining approach to achieve a trade-off between number of rules and suitability of membership functions," *Soft Computing*, vol. 10, no. 11, pp. 1091–1101, 2006.

[38] F. Di Martino, V. Loia, I. Perfilieva, and S. Sessa, "An image coding/decoding method based on direct and inverse fuzzy transforms," *International Journal of Approximate Reasoning*, vol. 48, no. 1, pp. 110–131, 2008.

[39] F. D. Martino, V. Loia, and S. Sessa, "Fuzzy transforms for compression and decompression of color videos," *Information Sciences*, vol. 180, no. 20, pp. 3914–3931, 2010.

[40] F. Di Martino and S. Sessa, "Compression and decompression of images with discrete fuzzy transforms," *Information Sciences*, vol. 177, no. 11, pp. 2349–2362, 2007.

Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2012, Article ID 125086, 9 pages
doi:10.1155/2012/125086

Research Article

Advanced F-Transform-Based Image Fusion

Marek Vajgl, Irina Perfilieva, and Petra Hod'áková

Institute for Research and Applications of Fuzzy Modeling, University of Ostrava, 30. dubna 22, 701 03 Ostrava, Czech Republic

Correspondence should be addressed to Irina Perfilieva, irina.perfilieva@osu.cz

Received 7 August 2012; Accepted 23 August 2012

Academic Editor: Ferdinando Di Martino

We propose to use the modern technique of the F-transform in order to show that it can be successfully applied to the image fusion. We remind two working algorithms (SA—the simple algorithm, and CA—the complete algorithm) which are based on the F-transform and discuss, how they can be improved. We propose a new algorithm (ESA—the enhanced simple algorithm) which is effective in time and free of frequently encountered shortcomings.

1. Introduction

Image processing is nowadays one of the most interesting research areas where traditional and new approaches are applied together and bring significant advantages. In this contribution, we consider the image fusion, which is one of many subjects of image processing. The image fusion aims at integration of complementary distorted multisensor, multitemporal, and/or multiview scenes into one new image which contains the "best" parts of each scene. Thus, the main problem in the area of image fusion is to find the less undistorted scene for every given pixel.

A local focus measure is traditionally used for selection of an undistorted scene. The scene which maximizes the focus measure is selected. Usually, the focus measure is a measure of high frequency occurrences in the image spectrum. This measure is used when a source of distortion is connected with blurring which suppresses high frequencies in an image. In this case, it is desirable that a focus measure decreases with an increase of blurring.

There are various fusion methodologies currently in use. The methodologies differ according to different mathematical fields: statistical methods (e.g., using aggregation operators, such as the MinMax method [1]), estimation theory [2], fuzzy methods (see [3, 4]), optimization methods (e.g., neural networks, genetic algorithms [5]), and multiscale decomposition methods, which incorporate various transforms, for example, discrete wavelet transforms (for a classification of these methods see [6], a classification of wavelet-based image fusion methods can be found in [7], and for applications for blurred and unregistered images, refer to [8]).

In our approach, we propose to use the modern technique of the F-transform and to show that it can be successfully applied to the image fusion. Our previous attempts have been reported in [9–12]. The original motivation for the F-transform (a short name for the fuzzy transform) came from fuzzy modeling [13, 14]. Similarly to traditional transforms (Fourier and wavelet), the F-transform performs a transformation of an original universe of functions into a universe of their "skeleton models" (vectors of F-transform components) in which further computation is easier. Moreover, sometimes, the F-transform can be more efficient than its counterparts. The F-transform proves to be a successful methodology with various applications: image compression and reconstruction [15, 16], edge detection [17, 18], numeric solution of differential equations [19], and time-series procession [20].

The F-transform-based approach to the image fusion has been proposed in [11, 12]. The main idea is a combination of (at least) two fusion operators, both are based on the F-transform. The first fusion operator is applied to F-transform components of scenes and is based on a robust partition of the scene domain. The second fusion operator is applied to the residuals of scenes with respect to inverse F-transforms with fused components and is based on a finer partition of the same domain. Although this approach is not explicitly based on focus measures, it uses the fusion operator which is able to choose an undistorted scene among available blurred. In this contribution, we analyze two methods of fusion that have been discussed in [11, 12] and propose a new method which can be characterized as a weighted combination of those two. We show that

- the new method is computationally more effective than the complete algorithm of fusion and has better quality than the simple algorithm of fusion, both have been proposed in [11, 12].

2. F-Transform

Before going into the details of image fusion, we give a brief characterization of the F-transform technique applied herein (we refer to [13] for a complete description).

Generally speaking, the F-transform is a linear mapping from a set of ordinary continuous/discrete functions over domain P onto a set of discrete functions (vectors) defined on a fuzzy partition of P. We assume that the reader is familiar with the notion of *fuzzy set* and the way(s) of its representation. In this paper, we identify fuzzy sets with their membership functions. In the below given explanation, we will speak about the F-transform of an image function u which is a discrete function $u : P \to \mathbb{R}$ of two variables, defined over the set of pixels $P = \{(i, j) \mid i = 1, \ldots, N,\ j = 1, \ldots, M\}$ and taking value from the set of reals $\mathbb{R}$. Throughout this text, we will always assume that M, N, and u have the same meaning as above.

Let $[1, N] = \{x \mid 1 \le x \le N\}$ be an interval on the real line $\mathbb{R}$, $n \ge 2$, a number of fuzzy sets in a fuzzy partition of $[1, N]$, and $h = (N - 1)/(n - 1)$ the distance between nodes $x_1, \ldots, x_n \in [1, N]$, where $x_1 = 1$, $x_k = x_1 + (k - 1)\,h$, $k = 1, \ldots, n$. Fuzzy sets $A_1, \ldots, A_n : [1, N] \to [0, 1]$ establish an *h-uniform* fuzzy partition of $[1, N]$ if the following requirements are fulfilled.

(1) For every $k = 1, \ldots, n$, $A_k(x) = 0$ if $x \in [1, N] \setminus [x_{k-1}, x_{k+1}]$, where $x_0 = x_1$, $x_{N+1} = x_N$;

(2) for every $k = 1, \ldots, n$, A_k is continuous on $[x_{k-1}, x_{k+1}]$, where $x_0 = x_1$, $x_{N+1} = x_N$;

(3) for every $i = 1, \ldots, N$, $\sum_{k=1}^{n} A_k(i) = 1$;

(4) for every $k = 1, \ldots, n$, $\sum_{i=1}^{N} A_k(i) > 0$;

(5) for every $k = 2, \ldots, n - 1$, A_k is symmetrical with respect to the line $x = x_k$.

The membership functions of the respective fuzzy sets in a fuzzy partition are called *basic functions*. The example of triangular basic functions $A_1, \ldots, A_n$, $n \ge 2$ on the interval $[1, N]$ is given below:

$$A_1(x) = \begin{cases} 1 - \dfrac{(x - x_1)}{h}, & x \in [x_1, x_2], \\ 0, & \text{otherwise}, \end{cases}$$

$$A_k(x) = \begin{cases} \dfrac{|x - x_k|}{h}, & x \in [x_{k-1}, x_{k+1}], \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

$$A_n(x) = \begin{cases} \dfrac{(x - x_{n-1})}{h}, & x \in [x_{n-1}, x_n], \\ 0, & \text{otherwise}. \end{cases}$$

Let us remark that

(1) the shape (e.g., triangular or sinusoidal) of a basic function in a fuzzy partition is not predetermined and can be chosen according to additional requirements, for example, smoothness, and so forth, see [13];

(2) if the shape of a basic function of a uniform fuzzy partition of $[1, N]$ is chosen, then the basic function can be uniquely determined by the number n_p of points, which are "covered" by every "full" basic function A_k where

$$n_p = |\{i \in [1 < n] \mid A_k(i) > 0\}|, \quad k = 2, \ldots, n - 1, \tag{2}$$

(in this case, we assume that $n \ge 3$).

Similarly, a uniform fuzzy partition of the interval $[1, M]$ with $m \ge 2$ basic functions $B_1, \ldots, B_m$ can be defined. Then the fuzzy partition of $P = [1, N] \times [1, M]$ is obtained by $n \times m$ fuzzy sets $A_1 \cdot B_1, \ldots, A_n \cdot B_m$. Below, we will always assume that n, m denote quantities of fuzzy sets in fuzzy partitions of $[1, N]$ and $[1, M]$, respectively.

Let $u : P \to \mathbb{R}$ and fuzzy sets $A_k \cdot B_l$, $k = 1, \ldots, n$, $l = 1, \ldots, m$, establish a fuzzy partition of $[1, N] \times [1, M]$. The (direct) F-transform of u (with respect to the chosen partition) is an image of the map $F[u] : \{A_1, \ldots, A_n\} \times \{B_1, \ldots, B_m\} \to \mathbb{R}$ defined by

$$F[u](A_k \cdot B_l) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} u(i, j) A_k(i) B_l(j)}{\sum_{i=1}^{N} \sum_{j=1}^{M} A_k(i) B_l(j)}, \tag{3}$$

where $k = 1, \ldots, n$, $l = 1, \ldots, m$. The value $F[u](A_k \cdot B_l)$ is called an F-transform component of u and is denoted by $F[u]_{kl}$. The components $F[u]_{kl}$ can be arranged into the matrix representation as follows:

$$\mathbf{F}_{nm}[u] = \begin{pmatrix} F[u]_{11} & \ldots & F[u]_{1m} \\ \vdots & \vdots & \vdots \\ F[u]_{n1} & \ldots & F[u]_{nm} \end{pmatrix}. \tag{4}$$

The inverse F-transform of u is a function on P, which is represented by the following inversion formula, where $i = 1, \ldots, N$, $j = 1, \ldots, M$:

$$u_{nm}(i, j) = \sum_{k=1}^{n} \sum_{l=1}^{m} F[u]_{kl} A_k(i) B_l(j). \tag{5}$$

It can be shown that the inverse F-transform u_{nm} approximates the original function u on the domain P. The proof can be found in [13, 14].

3. The Problem of Image Fusion

Image fusion aims at the integration of various complementary image data into a single, new image with the best possible quality. The term "quality" depends on the demands of the specific application, which is usually related to its usefulness for human visual perception, computer vision, or further processing. More formally, if u is an ideal image (considered as a function of two variables) and $c_1, \ldots, c_K$ are acquired (input) images, then the relation between each c_i and u can be expressed by

$$c_i(x, y) = d_i(u(x, y)) + e_i(x, y), \quad i = 1, \ldots, K, \tag{6}$$

where d_i is an unknown operator describing the image degradation, and e_i is an additive random noise. The problem of fusion consists in finding an image $\hat{u}$ such that it is close to u and it is better (in terms of a chosen quality) than any of $c_1, \ldots, c_K$. This problem occurs, for example, if multiple photos with focuses on different objects of the same scene are taken.

4. Image Decomposition for Image Fusion

Let us explain the mechanism of fusion with the help of the F-transform. It is based on a chosen decomposition of an image. We distinguish a one-level and a higher-level decomposition. We assume that the image u is a discrete real function $u = u(x, y)$ defined on the $N \times M$ array of pixels $P = \{(i, j) \mid i = 1, \ldots, N,\ j = 1, \ldots, M\}$ so that $u : P \to \mathbb{R}$. Moreover, let fuzzy sets $A_k \cdot B_l$, $k = 1, \ldots, n$, $l = 1, \ldots, m$, where $2 \le n \le N$, $2 \le m \le M$ establish a fuzzy partition of $[1, N] \times [1, M]$.

We begin with the following representation of u on P:

$$u(x, y) = u_{nm}(x, y) + e(x, y), \quad 0 < n \le N,\ 0 < m \le M, \tag{7}$$

$$e(x, y) = u(x, y) - u_{nm}(x, y), \tag{8}$$

where u_{nm} is the inverse F-transform of u and e is the respective first difference. If we replace e in (7) by its inverse F-transform e_{NM} with respect to the finest partition of $[1, N] \times [1, M]$, the above representation can then be rewritten as follows:

$$u(x, y) = u_{nm}(x, y) + e_{NM}(x, y), \quad \forall (x, y) \in P, \tag{9}$$

We call (9) a one-level decomposition of u on P.

If function u is smooth, then the function e_{NM} is small, and the one-level decomposition (9) is sufficient for our fusion algorithm. However, images generally contain various types of degradation that disrupt their smoothness. As a result, the function e_{NM} in (9) is not negligible, and the one-level decomposition is insufficient for our purpose. In this case, we continue with the decomposition of the first difference e in (7). We decompose e into its inverse F-transform $e_{n'm'}$ (with respect to a finer fuzzy partition of $[1, N] \times [1, M]$ with $n' : n < n' \le N$ and $m' : m < m' \le M$ basic functions, resp.) and the second difference e'. Thus, we obtain the second-level decomposition of u on P:

$$\begin{aligned} u(x, y) &= u_{nm}(x, y) + e_{n'm'}(x, y) + e'(x, y), \\ e'(x, y) &= e(x, y) - e_{n'm'}(x, y). \end{aligned} \tag{10}$$

In the same manner, we can obtain a higher-level decomposition of u on P:

$$\begin{aligned} u(x, y) = {} & u_{n_1m_1}(x, y) + e^{(1)}_{n_2m_2}(x, y) + \cdots + e^{(k-2)}_{n_{k-1}m_{k-1}}(x, y) \\ & + e^{(k-1)}(x, y), \end{aligned} \tag{11}$$

where

$$\begin{gathered} 0 < n_1 \le n_2 \le \cdots \le n_{k-1} \le N, \\ 0 < m_1 \le m_2 \le \cdots \le m_{k-1} \le M, \\ e^{(1)}(x, y) = u(x, y) - u_{n_1m_1}(x, y), \\ e^{(i)}(x, y) = e^{(i-1)}(x, y) - e^{(i-1)}_{n_im_i}(x, y), \quad i = 2, \ldots, k-1. \end{gathered} \tag{12}$$

Below, we will be working with the two decompositions of u that are given by (9) and (11).

5. Two Algorithms for Image Fusion

In [12], we proposed two algorithms:

(i) the simple F-transform-based fusion algorithm (SA) and

(ii) the complete F-transform-based fusion algorithm (CA).

These algorithms are based on the decompositions (9) and (11), respectively.

The principal role in fusion algorithms CA and SA is played by the *fusion operator* $\kappa : \mathbb{R}^K \to \mathbb{R}$, defined as follows:

$$\kappa(x_1, \ldots, x_K) = x_p, \quad \text{if } \left| x_p \right| = \max(|x_1|, \ldots, |x_K|). \tag{13}$$

5.1. Simple F-Transform-Based Fusion Algorithm. In this section, we give a "block" description of the SA without technical details which can be found in [12] and not be repeated here. We assume that $K \ge 2$ input images $c_1, \ldots, c_K$ with various types of degradation are given. Our aim is to recognize undistorted parts in the given images and to fuse them into one image.

(i) Choose values n, m such that $2 \le n \le N, 2 \le m \le M$ and create a fuzzy partition of $[1, N] \times [1, M]$ by fuzzy sets $A_k \cdot B_l$, $k = 1, \ldots, n$, $l = 1, \ldots, m$.

(ii) Decompose input images $c_1, \ldots, c_K$ into inverse F-transforms and error functions according to the one-level decomposition (9).

(a) (b)

FIGURE 1: Two inputs of the image "Table." The toy is blurred in the left image, and vice versa, it is the only sharp part in the right one.

FIGURE 2: The SA fusion of the image "Table," approximate run time: 1,6 sec.

(iii) Apply the fusion operator (13) to the respective F-transform components of $c_1, \ldots, c_K$ and obtain the fused F-transform components of a new image.

(iv) Apply the fusion operator to the to the respective F-transform components of the error functions e_i, $i = 1, \ldots, K$, and obtain the fused F-transform components of a new error function.

(v) Reconstruct the fused image from the inverse F-transforms with the fused components of the new image and the fused components of the new error function.

The SA-based fusion is very efficient if we can guess values n, m, that characterize a proper fuzzy partition. Usually, this is done manually according to user's skills. The dependence on fuzzy partition parameters can be considered as a main shortcoming of this otherwise effective algorithm. Two recommendations follow from our experience.

(i) For complex images (with many small details), higher values of n, m give better results.

(ii) If a triangular shape of a basic function is chosen, then the generic choice of n, m is such that the corresponding values of n_p, m_p are equal to 3 (recall that n_p is a number of points, which are covered by every full basic function A_k).

In this section, the algorithm SA is illustrated on examples "Table" and "Castle," see Figures 2 and 4 below. There are two inputs of the image "Table" (Figure 1) and four ones of the image "Castle" (Figure 3).

5.2. Complete F-Transform-Based Fusion Algorithm. The CA-based fusion does not depend on one choice of fuzzy partition parameters (as in the case of the SA), because it runs through a sequence of increasing values n, m. The description of the CA is similar to that of the SA except for the step 4 which is repeated in a cycle. Therefore, the quality of fusion is high, but the implementation of the CA is rather slow and memory consuming, especially for large images. For illustration, see Figures 5 and 6.

5.3. Fusion Artefacts. In this section, we characterize input images, for which it is reasonable to apply the SA or CA. By doing this, we put restrictions on inputs which are acceptable by the algorithms SA and CA. First of all, input images should be taken without shifting or rotation. Secondly, blurred parts of input images should not contain many small details like leaves on trees, and so forth. If it is so, then the fusion made by SA or CA can leave "artefacts," like "ghosts" or "lakes," see the explanation below where we assume that there are two input images for the fusion.

(i) Ghosts: this happens when a sharp edge of a nondamaged input image is significantly blurred in the other one. As a result of the SA or CA, the edge is perfectly reconstructed, but its neighboring area is affected by the edge presence (see Figure 7).

(ii) Lakes: this may happen in both cases when the fusion is performed by the SA or CA. In the case of SA, a "lake" is a result of choosing neighboring areas with significantly different colors from different input images. In the case of SA, a "lake" is a result of rounding off numbers (see Figure 8).


(a)


(b)


(c)


(d)

FIGURE 3: Four inputs (sharp zones vary from the northwest to the southeast quatre) of the image "Castle."

FIGURE 4: The SA fusion of the image "Castle," approximate run time: 1,9 sec.

FIGURE 5: Table—the CA fusion of the image "Table," 2 input images, approximate run time: 111 sec.

FIGURE 6: Table—the CA fusion of the image "Castle," 4 input images, approximate run time: 359 sec.


(a)


(b)

FIGURE 7: Example of "ghost"—the white area around the left button (a) and the doubled edge of the roof (b).

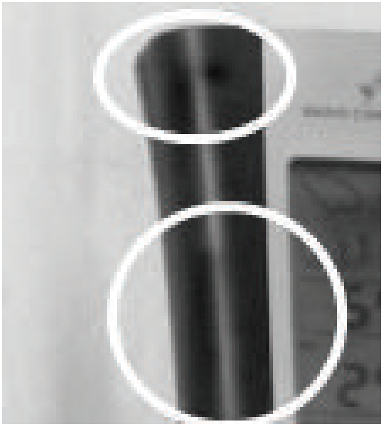


Figure 8: Example of “lake”—the color of this area is significantly different from the respective colors of input images.

Figure 9: Table—the ESA fusion of the image “Table,” 2 input images, approximate run time: 10 sec.

6. Improved F-Transform Fusion

The main purpose of this contribution is to create a method which will be as fast as the SA and as efficient as the CA. The following particular goals should be achieved.

(i) Avoid running through a long sequence of possible partitions (as in the case of CA).

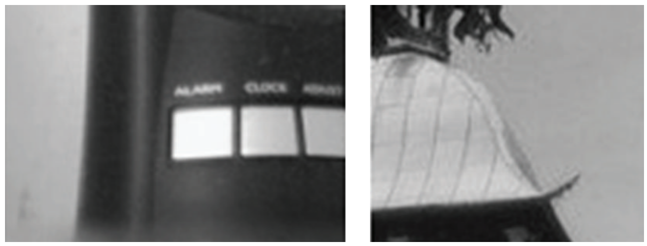
(ii) Automatically adjust parameters of a fusion algorithm according to a level of blurring and a location of a blurred area in input images.

(iii) Eliminate situation which can lead to “ghosts” and “lakes” in a fused image.

6.1. Proposed Solution. The main idea of the improved F-transform fusion is to enhance the SA by adding another run of the F-transform over the first difference (7). Our explanation is as follows: the first run of the F-transform is aimed at edge detection in each input image, while the second run propagates only sharp edges (and their local areas) to the fused image. The informal description of the enhanced simple algorithm (ESA) is given in Algorithm 1.

Although the algorithm ESA is written for gray scale input images, there is an easy way how to extend it to color

Figure 10: Table—the ESA fusion of the image “Castle,” 4 input images, approximate run time: 18 sec.



(a) (b)

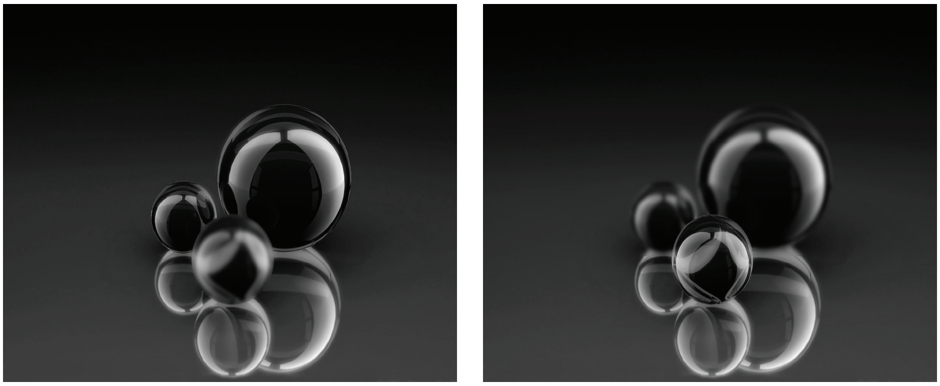
Figure 11: Example of “ghost” artefact reduction by ESA.

Figure 12: Example of “lakes” artefact reduction by ESA.

images which are represented in RGB or YUV models. Our tests were performed for both of them. In the case of RGB, the respective R, G, or B channels were processed independently and then combined. In the second case of YUV, the Y-part of the model was used to achieve weights (this part contains the most relevant information about the image intensity), while the U-part and the V-part were processed with the obtained weights.

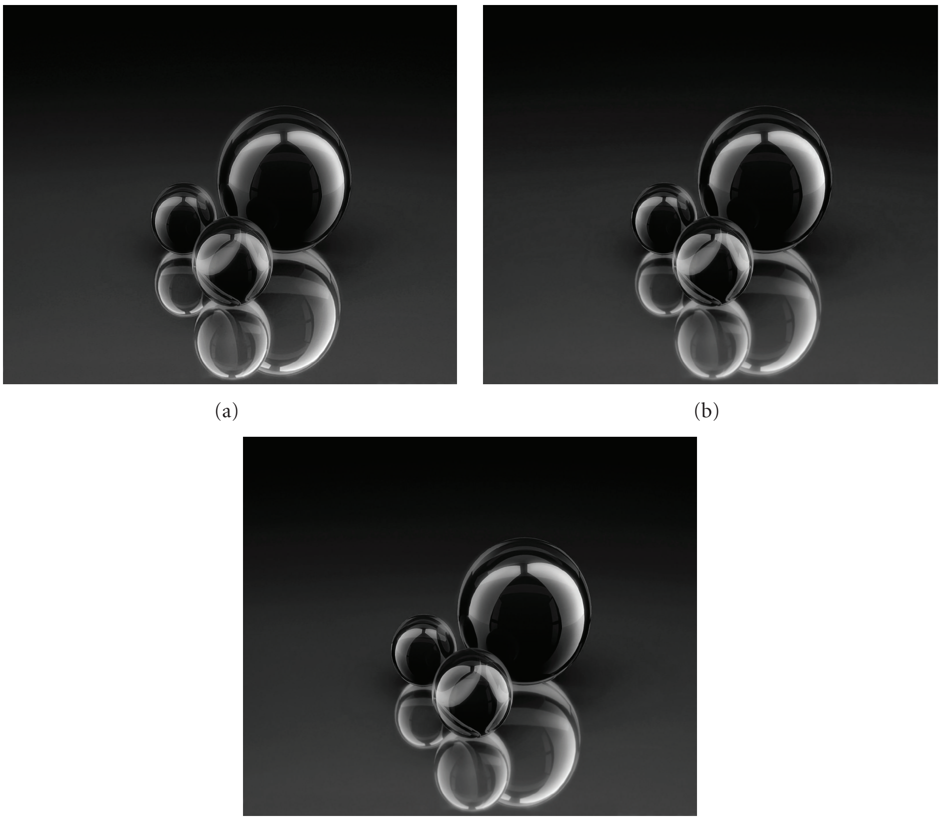
Let us remark that the ESA-fused images are (in general) better than each of the SA or CA. It can be visually seen on the chosen Figures 9 and 10. The main advantages of the ESA are as follows.

(i) Time: the executing time is smaller than in the case of the CA (in the examples above it is as follows: 11



(a) (b)

Figure 13: Two inputs of the image "Balls." The central ball is blurred in (a) and vice versa, it is the only one sharp ball (b).



(a) (b)

(c)

Figure 14: The SA (a), CA (b), and ESA (c) fusions of the image "Balls." The best quality has the ESA fusion (cf. Table 2).

Table 1: Basic characteristics of the three algorithms applied to the tested images.

Image set	Resolution	Time (s)			Memory (MB)		
		CA	SA	ESA	CA	SA	ESA
Castle	1120 × 840	359	1,9	19,0	160	35	102
Table	852 × 639	111	1,6	11,0	95	3	38
Balls	1600 × 1200	340	1,2	36	270	58	152

for all input images **do**
  Compute the *inverse F-transform*
  Compute the *first absolute difference* between the original image and the inverse F-transform of it
  Compute the *second absolute difference* between the first one and its inverse F-transform and set them as weights of pixels
end for
for all pixels in an image **do**
  Compute the value of *sow*—the sum of weights over all input images
  for all input images **do**
    Compute the value of *wr*—the ratio between the weight of a current pixel and *sow*
  end for
  Compute the fused value of a pixel in the resulting image as a weighted (by *wr*) sum of input image values
end for

Algorithm 1

Table 2: MSE and PSNR characteristics of the three fusion methods applied to the tested images.

Image set	MSE			PSNR		
	CA	SA	ESA	CA	SA	ESA
Castle	9,48	42,48	14,15	40,62	37,51	40,61
Balls	1,28	6,03	0,86	48,91	43,81	52,57

versus 111 (“Table”), 18 versus 359 (“Castle”). The quality of the ESA fusion is better than that of the SA. Examples of run times and memory consumption are presented in Table 1 (notice that the memory consumption significantly depends on memory management of implementation environment.)

(ii) Ghosts: ghosts effect is reduced. The “ghost” effects (they are seen around the tower roof in the image “Castle” and around the buttons and the clock in the image “Table”) are removed as it can be seen in Figure 11.

(iii) Lakes: lakes effect is eliminated. The “lakes” are almost eliminated as it can be seen from Figures 8, 9, and 12.

6.2. Comparison between Three Algorithms. In this section, we show that in general, the ESA fusion has better execution parameters than the SA or CA fusion. We experimented with numerous images which due to the space limitation cannot be presented in this paper. An exception is made for one image “Balls” with geometric figures to show how the fusion methods reconstruct edges. In Figure 13, two inputs of the image “Balls” are given, and in Figure 14, three fusions of the same image are demonstrated.

In Table 1, we demonstrate that the complexity (measured by the execution time or by the used memory) of the newly proposed ESA fusion is greater than the complexity of the SA and less than the complexity of the CA.

In Table 2, we demonstrate that the quality of fusion (measured by the values of MSE and PSNR) of the newly proposed ESA fusion is better (the MSE value is smaller) than the quality of the SA and in some cases (the image “Balls”) is better than the quality of the CA. Table 2 does not contain the values of MSE and PSNR for the image “Table,” because (as it happens in reality) there was no original (nondistorted) image at disposal.

7. Conclusion

In this paper, we continued our research started in [9–12] on effective fusion algorithms. We proposed the improved method of the F-transform-based fusion which is free from following imperfections: long running time, dependence on initial parameters which characterize a proper fuzzy partition, and presence of fusion artefacts, like “ghosts” or “lakes”.

Acknowledgments

This work relates to Department of the Navy Grant N62909-12-1-7039 issued by Office of Naval Research Global. The US Government has a royalty-free license throughout the world in all copyrightable material contained herein. Additional support was given also by the SGS12/PRF/2012 (Image Processing and Artefacts Detection Using Soft Computing).

References

[1] R. S. Blum, “Robust image fusion using a statistical signal processing approach,” *Information Fusion*, vol. 6, no. 2, pp. 119–128, 2005.

[2] A. Loza, D. Bull, N. Canagarajah, and A. Achim, “Non-Gaussian model-based fusion of noisy images in the wavelet domain,” *Computer Vision and Image Understanding*, vol. 114, no. 1, pp. 54–65, 2010.

[3] H. Singh, J. Raj, G. Kaur, and T. Meitzler, “Image fusion using fuzzy logic and applications,” in *Proceedings of the IEEE International Conference on Fuzzy Systems*, vol. 1, pp. 337–340, July 2004.

[4] R. Ranjan, H. Singh, T. Meitzler, and G. R. Gerhart, “Iterative image fusion technique using fuzzy and neuro fuzzy logic and applications,” in *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS ’05)*, pp. 706–710, June 2005.

[5] A. Mumtaz, A. Majid, and A. Mumtaz, “Genetic algorithms and its application to image fusion,” in *Proceedings of the 4th IEEE International Conference on Emerging Technologies 2008 (ICET ’08)*, pp. 6–10, October 2008.

[6] G. Piella, “A general framework for multiresolution image fusion: from pixels to regions,” *Information Fusion*, vol. 4, no. 4, pp. 259–280, 2003.

[7] K. Amolins, Y. Zhang, and P. Dare, “Wavelet based image fusion techniques-an introduction, review and comparison,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 62, no. 4, pp. 249–263, 2007.

[8] F. Šroubek and and J. Flusser, “Fusion of blurred images,” in *Multi-Sensor Image Fusion and Its Applications*, Z. Liu and R. Blum, Eds., Signal Processing and Communications Series, CRC Press, San Francisco, Calif, USA, 2005.

[9] M. Daňková and R. Valášek, “Full fuzzy transform and the problem of image fusion,” *Journal of Electrical Engineering*, no. 12, pp. 82–84, 2006.

[10] I. Perfilieva and M. Daňková, “Image fusion on the basis of fuzzy transforms,” in *Proceedings of the 8th International FLINS Conference on Computational Intelligence in Decision and Control*, pp. 471–476, Madrid, Spain, September 2008.

[11] I. Perfilieva, M. Daňková, P. Hod’áková, and M. Vajgl, “The use of F-transform for image fusion algorithms,” in *Proceedings of the International Conference of Soft Computing and Pattern Recognition (SoCPaR ’10)*, pp. 472–477, December 2010.

[12] P. Hodáková, I. Perfilieva, M. Daňková, and M. Vajgl, “F-transform based image fusion,” in *Image Fusion*, O. Ukimura, Ed., pp. 3–22, InTech, Rijeka, Croatia, 2011.

[13] I. Perfilieva, “Fuzzy transforms: theory and applications,” *Fuzzy Sets and Systems*, vol. 157, no. 8, pp. 993–1023, 2006.

[14] I. Perfilieva, “Fuzzy transforms: a challenge to conventional transforms,” *Advances in Imaging and Electron Physics*, vol. 147, pp. 137–196, 2007.

[15] I. Perfilieva, V. Pavliska, M. Vajgl, and B. De Baets, “Advanced image compression on the basis of fuzzy transforms,” in *Proceedings of the 12th International Conference Information Processing and Management of Uncertainty for Knowledge-Based Systems (IPMU ’08)*, pp. 1167–1174, Malaga, Spain, 2008.

[16] F. Di Martino, V. Loia, I. Perfilieva, and S. Sessa, “An image coding/decoding method based on direct and inverse fuzzy transforms,” *International Journal of Approximate Reasoning*, vol. 48, no. 1, pp. 110–131, 2008.

[17] I. Perfiljeva, M. Daňková, P. Hodáková, and M. Vajgl, “Edge detection using F-transform,” in *Proceedings of the 11th International Conference on Intelligent Systems Design and Applications (ISDA ’11)*, pp. 672–677, Cordoba, Spain, 2011.

[18] I. Perfiljeva, P. Hodáková, and P. Hurtík, “F^1-transform edge detector inspired by Canny’s algorithm,” in *Communications in Computer and Information Science. Advances on Computational Intelligence*, pp. 230–239, Springer, Heidelberg, Germany, 2012.

[19] M. Štěpnička and M. R. Valášek, “Numerical solution of partial differential equations with help of fuzzy transform,” in *Proceedings of the 2005 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE ’05)*, pp. 1153–1162, Reno, Nev, USA, 2005.

[20] I. Perfilieva, V. Novák, V. Pavliska, A. Dvořák, and M. Štěpnička, “Analysis and prediction of time series using fuzzy transform,” in *Proceedings of the International Joint Conference on Neural Networks (WCCi ’08)*, pp. 3875–3879, Hong Kong, Hong Kong.

Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2012, Article ID 290845, 4 pages
doi:10.1155/2012/290845

Research Article

A Note on $(\Phi E_1, \Phi E_2)$-Convex Fuzzy Processes

Marian Matłoka

Department of Applied Mathematics, Poznań University of Economics, Al. Niepodległości 10, 61-875 Poznań, Poland

Correspondence should be addressed to Marian Matłoka, marian.matloka@ue.poznan.pl

Received 18 June 2012; Accepted 2 September 2012

Academic Editor: Salvatore Sessa

We first define the concept of $(\Phi E_1, \Phi E_2)$-convex fuzzy processes. Second, we present some basic properties of such processes.

1. Introduction

In recent years, many generalizations of convexity have appeared in the literature aiming at applications to duality theory and optimality conditions. In 1997, Pini and Singh [1] introduced (Φ_1, Φ_2)-convex functions and studied some of their properties. They showed that some of the well-known classes of generalized convex functions (e.g., B-vex functions [2], geodesic convex functions [3], and invex functions [4]) form subclasses of the class of (Φ_1, Φ_2)-convex functions. In 1999, Youness [5] showed that many results for convex sets and convex functions actually hold for a wider class of sets and functions, called E-convex sets and E-convex functions.

Convex processes were studied first by Rockafellar [6] who was interested in extending properties of linear transformations to a large class of maps preserving convexity and which arise naturally in economic theory.

The extension of this notion to the fuzzy framework was done by Matłoka [7, 8] and was investigated by Syau et al. [9] and by Chalco-Cano et al. [10].

In this work, we extend the notion of ΦE-convex set, ΦE-convex fuzzy set, and $(\Phi E_1, \Phi E_2)$-convex fuzzy process. First, we will present the preliminary definitions and next the main properties of $(\Phi E_1, \Phi E_2)$-convex fuzzy process.

2. Preliminaries

Let X_i denote a subset of the n_i-dimensional Euclidean space R^{n_i}. Assume that Φ_i is a map satisfying the following assumption:

(i) $\Phi_i : X_i \times X_i \times [0,1] \to R^{n_i}$,

(ii) $\Phi_i(x, y, 0) = y$, $\Phi_1(x, x, \lambda) = x$, $\forall x, y \in X_i$, $\lambda \in [0,1]$.

For any subsets A, B, of X_i let us define

$$\Phi_i(A, B, \lambda) = \{\Phi_i(x, y, \lambda) : x \in A, y \in B\}. \tag{1}$$

Definition 1 (see [1]). A set X_i is Φ_i-convex if $\Phi_i(x_1, x_2, \lambda) \in X_i$ for all $x_1, x_2 \in X_i$, $\lambda \in [0,1]$.

Remark 2. The intersection of Φ_i-convex sets is still Φ_i-convex.

Remark 3. Let X_i be a convex subset of R^{n_i}, and $\Phi_i(x_1, x_2, \lambda) = \lambda x_1 + (1-\lambda)x_2$. Then a convex set is Φ_i-convex.

Remark 4. If $\eta : R^{n_i} \times R^{n_i} \to R^{n_i}$, X_i is a preinvex set with respect to η (see [11]), $\Phi_i(x_1, x_2, \lambda) = x_2 + \lambda \cdot \eta(x_1, x_2)$, then X_i is a Φ_i-convex set.

Assume that $E_1 : R^{n_i} \to R^{n_i}$.

Definition 5 (see [5]). A set $X_i \subset R^{n_i}$ is said to be E_i-convex if $(1-\lambda)E(x_1) + \lambda E(x_2) \in X_i$, for each $x_1, x_2 \in X_i$ and $\lambda \in [0,1]$.

Definition 6. A set $X_i \subset R^{n_i}$ is said to be ΦE_i-convex if $\Phi_i(E_i(x_1), E_i(x_2), \lambda) \in X_i$, for each $x_1, x_2 \in X_i$ and $\lambda \in [0,1]$.

Remark 7. If $\Phi_i(x_1, x_2, \lambda) = \lambda x_1 + (1-\lambda)\, x_2$ and E_i is the identity map then a convex set is ΦE_i-convex.

Definition 8. Let X_i be a ΦE_i-convex set. A function $f : X_i \to R$ is said to be ΦE_i-quasiconcave on X_i if for any $x_1, x_2 \in X_i$ and $\lambda \in [0,1]$

$$f(\Phi_i(E_i(x_1), E_i(x_2), \lambda)) \geq \min(f(x_1), f(x_2)). \tag{2}$$

Let $C : R^{n_i} \to [0,1]$ denote a fuzzy set in R^{n_i}.

Definition 9. A fuzzy set C is called ΦE_i-convex if and only if

$$C(\Phi_i(E_i(x_1), E_i(x_2), \lambda)) \geq \min(C(x_1), C(x_2)), \quad (3)$$

for all $x_1, x_2 \in R^{n_i}$.

Definition 10. An α-cut of a fuzzy set C is defined as follows:

$$C^\alpha = \begin{cases} \{x : C(x) \geq \alpha\} & \text{for } \alpha \in (0,1] \\ \{x : C(x) > \alpha\} & \text{for } \alpha = 0. \end{cases} \quad (4)$$

Proposition 11. *If C is ΦE_i-convex fuzzy set then C^α is ΦE_i-convex (crisp) set.*

Proof. We have to prove that if $x_1, x_2 \in C^\alpha$ then for any $\lambda \in [0,1]$, $\Phi_i(E_i(x_1), E_i(x_2), \lambda) \in C^\alpha$. So, taking into account the above definitions, we observe that if $x_1, x_2 \in C^\alpha$ then $C(x_1) \geq \alpha$ and $C(x_2) \geq \alpha$ and $\min(C(x_1), C(x_2)) \geq \alpha$. So, $C(\Phi_i(E_i(x_1), E_i(x_2), \lambda)) \geq \min(C(x_1), C(x_2)) \geq \alpha$. This means that $\Phi_i(E_i(x_1), E_i(x_2), \lambda) \in C^\alpha$. □

3. Main Results

In this section, we present the definition and some properties of the $(\Phi E_1, \Phi E_2)$-convex fuzzy processes.

Let X_i denote ΦE_i-convex set and $F(X_i)$ the set of all non-void fuzzy sets in X_i $(i = 1, 2, 3)$.

Definition 12. A mapping A from X_1 to $F(X_2)$ is called $(\Phi E_1, \Phi E_2)$-convex fuzzy process if and only if for any $x_1, x_2 \in X_1$ and $\lambda \in [0,1]$ and $y \in X_2$

$$A(\Phi_1(E_1(x_1), E_1(x_2), \lambda))(y) \geq \sup_{\substack{y_1, y_2 \in X_2: \\ y = \Phi_2(E_2(y_1), E_2(y_2), \lambda)}} \min(A(x_1)(y_1), A(x_2)(y_2)). \quad (5)$$

Example 13. Let $X_1 = R$, $X_2 = R-(-1/2, 1/2)$ and let $E_1(x) = |x|$, $E_2(y) = y$, $\Phi_1(x_1, x_2, \lambda) = \lambda x_1 + (1-\lambda)x_2$, and

$$\Phi_2(y_1, y_2, \lambda) = \begin{cases} \lambda y_1 + (1-\lambda) y_2 & \text{if } y_1 y_2 > 0 \\ (1+\lambda) y_2 - \lambda y_1 & \text{if } y_1 y_2 < 0. \end{cases} \quad (6)$$

Consider $A : X_1 \to F(X_2)$ defined by

$$A(X) := \chi_{\langle f(x), \infty)}, \quad (7)$$

where $f(x) = k|x|$, $k > 0$, and χ_C denotes the characteristics function of C.

Then a mapping A is $(\Phi E_1, \Phi E_2)$-convex fuzzy process.

Now, let us consider $A : X_1 \to F(X_2)$ defined by

$$A(x)(y) = \begin{cases} \dfrac{y}{f(x)} & \text{if } y \in \langle 0, f(x)), \\ 1 & \text{if } y \in \langle f(x), \infty), \\ 0 & \text{if } y \in< (-\infty, 0), \end{cases} \quad (8)$$

for all $x \neq 0$ and

$$A(x)(y) = \begin{cases} 1 & \text{if } y \in \langle 0, \infty), \\ 0 & \text{if } y \in (-\infty, 0), \end{cases} \quad (9)$$

for $x = 0$, where $f(x) = k|x|$, $k > 0$.

The above mapping is $(\Phi E_1, \Phi E_2)$-convex fuzzy mapping too.

Theorem 14. *If A is a $(\Phi E_1, \Phi E_2)$-convex fuzzy process from X_1 to $F(X_2)$ and E_1 is an identity mapping then for any $x \in X_1 A(x)$ is a ΦE_2-convex fuzzy set in X_2.*

Proof. We will prove that if $y_1, y_2 \in X_2$, $\lambda \in [0,1]$ then for any $x \in X_1$

$$A(x)(\Phi_2(E_2(y_1), E_2(y_2), \lambda)) \geq \min(A(x)(y_1), A(x)(y_2)). \quad (10)$$

Let us note that for any $\in X_1 E_1(x) = x = \Phi_1(x, x, \lambda)$.

So, using the definition of $(\Phi E_1, \Phi E_2)$-convex fuzzy processes, we have

$$\begin{aligned} &A(x)(\Phi_2(E_2(y_1), E_2(y_2), \lambda)) \\ &\quad = A(\Phi_1(x, x, \lambda))(\Phi_2(E_2(y), E_2(y_2), \lambda)) \\ &\quad \geq \sup_{\substack{y', y'' \in X_2: \\ \Phi_2(E_2(y_1), E_2(y_2), \lambda) \\ = \Phi_2(E_2(y'), E_2(y''), \lambda)}} \min(A(x)(y'), A(x)(y'')) \quad (11) \\ &\quad \geq \min(A(x)(y_1), A(x)(y_2)). \end{aligned}$$

□

Definition 15. The graph of a $(\Phi E_1, \Phi E_2)$-convex fuzzy process A from X_1 to $F(X_2)$, denoted G_A, is a fuzzy set in $X_1 \times X_2$ such that for any $(x, y) \in X_1 \times X_2$

$$G_A(x, y) = A(x)(y). \quad (12)$$

In the analogous way as in the Definition 9, we can define the $(\Phi E_1, \Phi E_2)$-convex fuzzy subset B of $X_1 \times X_2$, that is,

$$\begin{aligned} &B(\Phi_1(E_1(x_1), E_1(x_2), \lambda), \Phi_2(E_2(y_1), E_2(y_2), \lambda)) \\ &\quad \geq \min(B(x_1, y_1), B(x_2, y_2)), \end{aligned} \quad (13)$$

for $x_1, x_2 \in X_1$, $y_1, y_2 \in X_2$, $\lambda \in [0,1]$.

Theorem 16. *The graph of a $(\Phi E_1, \Phi E_2)$-convex fuzzy process A from X_1 to $F(X_2)$ is a $(\Phi E_1, \Phi E_2)$-convex fuzzy subset of $X_1 \times X_2$.*

Proof. Taking into account the definitions of the graph and $(\Phi E_1, \Phi E_2)$-convex fuzzy process, we observe that for any $(x_1, y_1), (x_2, y_2) \in X_1 \times X_2$ and $\lambda \in [0,1]$

$$\begin{aligned} &G_A(\Phi_1(E_1(x_1), E_1(x_2), \lambda), \Phi_2(E_2(y_1), E_2(y_2), \lambda)) \\ &\quad = A(\Phi_1(E_1(x_1), E_1(x_2), \lambda))(\Phi_2(E_2(y_1), E_2(y_2), \lambda)) \\ &\quad \geq \sup_{\substack{y', y'' \in X_2: \\ \Phi_2(E_2(y_1), E_2(y_2), \lambda) \\ = \Phi_2(E_2(y'), E_2(y''), \lambda)}} \min(A(x_1)(y'), A(x_2)(y'')) \quad (14) \\ &\quad \geq \min(A(x_1)(y_1), A(x_2)(y_2)) \\ &\quad = \min(G_A(x_1, y_1), G_A(x_2, y_2)). \end{aligned}$$

□

Definition 17. A composition of $(\Phi E_1, \Phi E_2)$-convex fuzzy process A from X_1 to $F(X_2)$ and $(\Phi E_2, \Phi E_3)$-convex fuzzy process B from X_2 to $F(X_3)$ is mapping $B \circ A$ from X_1 to $F(X_3)$ such that

$$(B \circ A)(x)(z) = \sup_{y \in X_2} \min(A(x)(y), B(y)(z)) \tag{15}$$

for any $(x, z) \in X_1 \times X_3$.

Theorem 18. *If A is $(\Phi E_1, \Phi E_2)$-convex fuzzy process from X_1 to $F(X_2)$ and B is $(\Phi E_2, \Phi E_3)$-convex fuzzy process from X_2 to $F(X_3)$, then $B \circ A$ is a $(\Phi E_1, \Phi E_3)$-convex fuzzy process from X_1 to $F(X_3)$.*

Proof. Let $x_1, x_2 \in X_1$ and $\lambda \in [0,1]$. Then for any $z \in X_3$ we have

$$\begin{aligned}
&(B \circ A)(\Phi_1(E_1(x_1), E_1(x_2), \lambda))(z) \\
&= \sup_{y \in X_2} \min(A(\Phi_1(E_1(x_1), E_1(x_2), \lambda))(y), B(y)(z)) \\
&= \sup_{\substack{y_1, y_2 \in X_2: \\ y = \Phi_2(E_2(y_1),\, E_2(y_2),\, \lambda)}} \min(A(\Phi_1(E_1(x_1), E_1(x_2), \lambda)) \\
&\qquad (\Phi_2(E_2(y_1), E_2(y_2), \lambda)), \\
&B(\Phi_2(E_2(y_1), E_2(y_2), \lambda))(z)) \\
&\geq \sup_{\substack{y_1, y_2 \in X_2: \\ y = \Phi_2(E_2(y_1),\, E_2(y_2),\, \lambda)}} \min(A(x_1)(y_1), A(x_2)(y_2), \\
&\sup_{\substack{z_1, z_2 \in X_3: \\ z = \Phi_3(E_3(z_1), E_3(z_2), \lambda)}} \min(B(y_1)(z_1), B(y_2)(z_2))) \\
&\geq \sup_{\substack{z_1, z_2 \in X_3: \\ z = \Phi_3(E_3(z_1), E_3(z_2), \lambda)}} \sup_{y_1, y_2 \in X_2} \min(A(x_1)(y_1), A(x_2)(y_2), \\
&\qquad B(y_1)(z_1), B(y_2)(z_2)) \\
&= \sup_{\substack{z_1, z_2 \in X_3: \\ z = \Phi_3(E_3(z_1), E_3(z_2), \lambda)}} \min\left(\sup_{y_1 \in X_2} \min(A(x_1)(y_1), B(y_1)(z_1)),\right. \\
&\qquad \left.\sup_{y_2 \in X_2} \min(A(x_2)(y_2), B(y_2)(z_2))\right) \\
&= \sup_{\substack{z_1, z_2 \in X_3: \\ z = \Phi_3(E_3(z_1), E_3(z_2), \lambda)}} \min((B \circ A)(x_1)(z_1), (B \circ A)(x_2)(z_2)).
\end{aligned} \tag{16}$$

So, $B \circ A$ is a $(\Phi E_1, \Phi E_3)$-convex fuzzy process.

For any set $C \subset X_1$, we put

$$A(C)(y) = \sup_{x \in C} A(x)(y) \tag{17}$$

for any $y \in X_2$. □

Theorem 19. *If A is $(\Phi E_1, \Phi E_2)$-convex fuzzy process from X_1 to $F(X_2)$ and C is a ΦE_1-convex subset of X_1 and E_1 is an identity mapping then $A(C)$ is ΦE_2-convex subset of X_2.*

Proof. Let $y_1, y_2 \in X_2$ and $\lambda \in [0,1]$. Then we have

$$\begin{aligned}
&A(C)(\Phi_2(E_2(y_1), E_2(y_2), \lambda)) \\
&\quad = \sup_{x \in C} A(x)(\Phi_2(E_2(y_1), E_2(y_2), \lambda)) \\
&\quad = \sup_{x \in C} A(\Phi_1(x, x, \lambda))(\Phi_2(E_2(y_1), E_2(y_2), \lambda)) \\
&\quad \geq \sup_{x \in C} \sup_{\substack{y', y'' \in X_2: \\ \Phi_2(E_2(y'), E_2(y''), \lambda) \\ = \Phi_2(E_2(y_1), E_2(y_2), \lambda)}} \min(A(x)(y'), A(x)(y'')) \\
&\quad = \sup_{x \in C} \min(A(x)(y_1), A(x)(y_2)) \\
&\quad = \min\left(\sup_{x \in C} A(x)(y_1), \sup_{x \in C} A(x)(y_2)\right) \\
&\quad = \min(A(C)(y_1), A(C)(y_2)).
\end{aligned} \tag{18}$$

So, $A(C)$ is a ΦE_2-convex fuzzy subset of X_2. □

Theorem 20. *If A is $(\Phi E_1, \Phi E_2)$-convex fuzzy process from X_1 to $F(X_2)$ then for any $\alpha \in [0,1]$*

$$\begin{aligned}
&[A(\Phi_1(E_1(x_1), E_1(x_2), \lambda))]^{\alpha} \\
&\quad \supset \Phi_2(E_2([A(x_1)]^{\alpha}), E_2([A(x_2)]^{\alpha}), \lambda)
\end{aligned} \tag{19}$$

for any $x_1, x_2 \in X_1$, $\lambda \in [0,1]$.

Proof. According to the definition of α-cut, we have

$$\begin{aligned}
&[A(\Phi_1(E_1(x_1), E_1(x_2), \lambda))]^{\alpha} \\
&\quad = \{y \in X_2 : A(\Phi_1(E_1(x_1), E_1(x_2), \lambda))(y) \geq \alpha\}, \\
&E_2([A(x_1)]^{\alpha}) = \{E_2(y_1) \in X_2 : A(x_1)(y_1) \geq \alpha\}, \\
&E_2([A(x_2)]^{\alpha}) = \{E_2(y_2) \in X_2 : A(x_2)(y_2) \geq \alpha\}.
\end{aligned} \tag{20}$$

Moreover,

$$\begin{aligned}
&\Phi_2(E_2([A(x_1)]^{\alpha}), E_2([A(x_2)]^{\alpha}), \lambda) \\
&\quad = \{y = \Phi_2(E_2(y_1), E_2(y_2), \lambda) : \\
&\qquad y_1 \in [A(x_1)]^{\alpha}, y_2 \in [A(x_2)]^{\alpha}\}, \\
&A(\Phi_1(E_1(x_1), E_1(x_2), \lambda))(y) \\
&\quad \geq \sup_{\substack{y_1, y_2 \in X_2: \\ y = \Phi_2(E_2(y_1), E_2(y_2), \lambda)}} \min(A(x_1)(y_1), A(x_2)(y_2)).
\end{aligned} \tag{21}$$

This means that if $y \in \Phi_2(E_2([A(x_1)]^{\alpha}), E_2([A(x_2)]^{\alpha}), \lambda)$ then $y \in [A\ (\Phi_1(E_1(x_1), E_1(x_2), \lambda)]^{\alpha}$, that is,

$$\begin{aligned}
&[A(\Phi_1(E_1(x_1), E_1(x_2), \lambda))]^{\alpha} \\
&\quad \supset \Phi_2(E_2([A(x_1)]^{\alpha}), E_2([A(x_2)]^{\alpha}), \lambda).
\end{aligned} \tag{22}$$

Now, for any $\alpha \in [0,1]$ and any ΦE_2-quasiconcave function $g : X_2 \to R$, let us define a function

$$q_g^{\alpha}(x) = \max_{y \in [A(x)]^{\alpha}} g(y) \quad \text{for } x \in X_1. \tag{23}$$

□

Theorem 21. *If A is a $(\Phi E_1, \Phi E_2)$-convex fuzzy process from X_1 to $F(X_2)$, then the function q_g^α is ΦE_1-quasiconcave.*

Proof. Let $x_1, x_2 \in X_1$, $\lambda \in [0,1]$. Then we have

$$\begin{aligned}
& q_g^\alpha(\Phi_1(E_1(x_1), E_1(x_2), \lambda)) \\
&\quad = \max_{y \in [A(\Phi_1(E_1(x_1), E_1(x_2), \lambda))]^\alpha} g(y) \\
&\quad \geq \max_{y \in \Phi_2(E_2([A(x_1)]^\alpha), E_2([A(x_2)]^\alpha),\, \lambda)} g(y) \\
&\quad = \max\{g(y) : \ y = \Phi_2(E_2(y_1), E_2(y_2), \lambda), \\
&\qquad\qquad y_1 \in [A(x_1)]^\alpha, y_2 \in [A(x_2)]^\alpha\} \\
&\quad = \max\{g(\Phi_2(E_2(y_1), E_2(y_2), \lambda) : \\
&\qquad\qquad y_1 \in [A(x_1)]^\alpha, y_2 \in [A(x_2)]^\alpha\} \\
&\quad \geq \max\{\min(g(y_1), g(y_2)) : \\
&\qquad\qquad y_1 \in [A(x_1)]^\alpha, y_2 \in [A(x_2)]^\alpha\} \\
&\quad \geq \min\left(\max_{y_1 \in [A(x_1)]^\alpha} g(y_1), \max_{y_2 \in [A(x_2)]^\alpha} g(y_2)\right) \\
&\quad = \min\left(q_g^\alpha(x_1), q_g^\alpha(x_2)\right).
\end{aligned} \tag{24}$$

□

References

[1] R. Pini and C. Singh, "$(\Phi E_1, \Phi E_2)$-convexity," *Optimization*, vol. 40, no. 2, pp. 103–120, 1997.

[2] C. R. Bector and C. Singh, "B-vex functions," *Journal of Optimization Theory and Applications*, vol. 71, no. 2, pp. 237–253, 1991.

[3] T. Rapcsák, "Geodesic convexity in nonlinear optimization," *Journal of Optimization Theory and Applications*, vol. 69, no. 1, pp. 169–183, 1991.

[4] M. A. Hanson, "On sufficiency of the Kuhn-Tucker conditions," *Journal of Mathematical Analysis and Applications*, vol. 80, no. 2, pp. 545–550, 1981.

[5] E. A. Youness, "E-convex sets, E-convex functions, and E-convex programming," *Journal of Optimization Theory and Applications*, vol. 102, no. 2, pp. 439–450, 1999.

[6] R. T. Rockafellar, "Monotone process of convex and concavetype," *Amer. Math. Soc.*, no. 77, 1967.

[7] M. Matłoka, "Convex fuzzy processes," *Fuzzy Sets and Systems*, vol. 110, no. 1, pp. 104–114, 2000.

[8] M. Matłoka, "Some properties of Φ-convex fuzzy processes," *The Journal of Fuzzy Mathematics*, vol. 7, pp. 701–706, 1999.

[9] Y.-R. Syau, C. Y. Low, and T. H. Wu, "A note on convex fuzzy processes," *Applied Mathematics Letters*, vol. 15, no. 2, pp. 193–196, 2002.

[10] Y. Chalco-Cano, M. A. Rojas-Medar, and R. Osuna-Gómez, "s-convex fuzzy processes," *Computers and Mathematics with Applications*, vol. 47, no. 8-9, pp. 1411–1418, 2004.

[11] T. Weir and B. Mond, "Pre-invex functions in multiple objective optimization," *Journal of Mathematical Analysis and Applications*, vol. 136, no. 1, pp. 29–38, 1988.

Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2012, Article ID 808361, 8 pages
doi:10.1155/2012/808361

Research Article

Fuzzy Systems Based on Multispecies PSO Method in Spatial Analysis

Ferdinando Di Martino,[1] Vincenzo Loia,[2] and Salvatore Sessa[1]

[1] *Dipartimento di Costruzioni e Metodi Matematici in Architettura, Università degli Studi di Napoli Federico II, Via Monteoliveto 3, 80134 Napoli, Italy*
[2] *Dipartimento di Matematica e Informaica, Università degli Studi di Salerno, Via Ponte don Melillo, 84084 Fisciano, Italy*

Correspondence should be addressed to Salvatore Sessa, sessa@unina.it

Received 15 April 2012; Accepted 31 May 2012

Academic Editor: Irina G. Perfilieva

We present a method by using the hierarchical cluster-based Multispecies particle swarm optimization to generate a fuzzy system of Takagi-Sugeno-Kang type encapsulated in a geographical information system considered as environmental decision support for spatial analysis. We consider a spatial area partitioned in subzones: the data measured in each subzone are used to extract a fuzzy rule set of above mentioned type. We adopt a similarity index (greater than a specific threshold) for comparing fuzzy systems generated for adjacent subzones.

1. Introduction

A geographical information system (shortly, GIS) is often used by the analyst as a decision support system for problem solving in spatial analysis (e.g., cf. [1–9]). In some works, fuzzy inference systems are encapsulated in GIS tools, where the knowledge is represented by if-then rules. In [7], the authors present an inference model, integrated in a GIS, based on Takagi-Sugeno-Kang (shortly, TSK) fuzzy system [10, 11]. In [4], a first-order Takagi-Sugeno fuzzy inference system is generated for estimating and simulating the discharge and sediment concentration in river basins. Also here we integrate in a tool GIS an inference model which generates a TSK-fuzzy system, based on the hierarchical cluster-based multiSpecies particle swarm optimization (shortly, HCMSPSO) algorithm [12] The spatial area of study is divided into subzones: for each subzone the correspondent TSK-fuzzy system is extracted by starting from the subset of patterns georeferred in that subzone. In our method, we calculate a similarity index between TSK-fuzzy systems generated for adjacent subzones: if this index is greater than a specific threshold, then the corresponding subzones and the data subsets are merged, and the TSK-fuzzy system is generated again for the new subzone. The partition of the area of study in subzones derives from the fact that the impact of the geographical characteristics necessarily involves spatial changes of the parameters forming the rule set of the fuzzy system.

A subzone represents a spatial subarea (of the area of study) with homogeneous characteristics with respect to the phenomenon investigated: we can consider as examples of subzones those areas with a specific index (pollution index, risk index, vulnerability index, traffic index, etc.). The concept of division of the area of study in homogeneous subzones has been introduced in [13–15], in which the area is partitioned in iso-reliable zones, that is, in subzones with homogeneous reliability of the spatial characteristics. Strictly speaking, the expert has provided a set of N patterns $(x_1, \ldots, x_n, y)$, where $\mathbf{x} = (x_1, \ldots, x_n)$ is the n-dimensional vector input, and y is the output, measured on georeferred locations in the area of study. This expert does not know a priori the optimal partition of the area of study in homogeneous subzones with respect to the fuzzy system to be generated; he already uses partitions defined for sociological, morphological, climatic characteristics. However, he does not know whether the same fuzzy system can be applied to two adjacent subzones. In this work, we propose a based particle swarm optimization (PSO) algorithm for optimizing the partition in homogeneous zones of the area of study: indeed we generate a TSK-fuzzy system for each subzone by using the so-called HCMSPSO algorithm [12], which is a variation of PSO, and we compare the TSK-fuzzy system in adjacent subzones. A TSK-fuzzy system [16] is composed from a set of r fuzzy rules R_i represented in the following form:

$$R_i : \text{IF}\,(x_1 \text{ is } A_{i1}) \cdots (x_n \text{ is } A_{in}), \quad \text{then}\,(y \text{ is } f_i), \quad i = 1, \dots, r, \tag{1}$$

where the fuzzy set A_{ij} assumes the following Gaussian membership function:

$$A_{ij} = \exp\left[-\left(\frac{x_j - m_{ij}}{\sigma_{ij}}\right)^2\right]. \tag{2}$$

The firing strength of the ith rule on $\mathbf{x} = (x_1, \dots, x_n)$ is given as

$$\mu_i(\mathbf{x}) = \prod_{j=1}^{n} A_{ij} = \exp\left[-\sum_{j=1}^{n}\left(\frac{x_j - m_{ij}}{\sigma_{ij}}\right)^2\right]. \tag{3}$$

In the TSK-fuzzy system of zero order, the fuzzy set f_i is generally represented in the consequent as a constant a_{i0}; the TSK-fuzzy system of one order is given by a linear combination of the input variables:

$$f_i = a_{i0} + \sum_{i=1}^{n} a_{ij} x_j, \tag{4}$$

where the coefficient a_{ij} are real numbers. The output y is calculated by the weighted average defuzzification method:

$$y' = \frac{\sum_{i=1}^{r} \mu_i(\mathbf{x}) f_i}{\sum_{i=1}^{r} \mu_i(\mathbf{x})}. \tag{5}$$

The HCMSPSO algorithm generates an optimal TSK-fuzzy system by determining the number r of rules and by optimizing the coefficient values m_{ij}, σ_{ij}, and a_{ij} in each rule. In our approach, the geographical area of study is initially partitioned in Z subzones; for each subzone we apply the HCMSPSO algorithm to generate the optimal TSK-fuzzy system. Here, we give an evolution process in which at each iteration we merge adjacent subzones with a fuzzy rule-set similarity index greater then a predefined threshold. The process stops when no fuzzy rule-set similarity index associated to couple of adjacent subzones is greater or equal to the threshold. We propose a similarity index based on the fuzzy inclusion concept given as

$$I_{ii'j}^{hk} = \frac{\int_{x_{j0}}^{x_{j1}} \min\left[A_{ij}^{h}\left(x_j\right), A_{i'j}^{k}\left(x_j\right)\right] dx_j}{\max\left[\int_{x_{j0}}^{x_{j1}} A_{ij}^{h}\left(x_j\right) dx_j, \int_{x_{j0}}^{x_{j1}} A_{i'j}^{k}\left(x_j\right) dx_j\right]}, \tag{6}$$

where x_{j0} and x_{j1} are the lower and upper bounds of the domain of x_j. If $I_{ij}^{hk} = 1$, the two fuzzy sets overlap completely.

In other words, given the hth and kth TSK-fuzzy systems correspondent to adjacent subzones and with the same number of rules, we define the following similarity index for each couple of rules r_i and $r_{i'}$, $(i, i') \in (1, \dots, r)^2$ as

$$I_{ii'}^{hk} = \frac{\sum_{j=1}^{n+1} I_{ii'j}^{hk}}{n+1}, \tag{7}$$

which is a mean of (6) on the $n + 1$ variables (n inputs and the output variable). Then, we order the rule sets of the two fuzzy systems in accordance to the following criteria:

(i) we choose the two rules r_i and $r_{i'}$ with the greatest value $I_{ii'}^{hk}$. The two rules become the first rules of the respective fuzzy systems, and the index (7) can be written as I_1^{hk};

(ii) among the remaining rules, we consider the two new rules with the index (7); these rules become the second rules of the respective fuzzy systems and the index (7) can be written as I_2^{hk}; the process is repeated up to order all the rules in the two fuzzy sets.

We calculate the similarity index of the two TSK-fuzzy systems as

$$S^{hk} = \frac{\sum_{i=1}^{r} I_i^{hk}}{r}. \tag{8}$$

The similarity value S^{hk} is a mean of the values I_i^{hk} obtained for all the rules. We use this index to decide if the hth and kth adjacent subzones can be merged. Our evolution process is a hierarchical iterative approach; initially the area of study is partitioned in a fine-grained set of subzones; the analyst divides the area of study into subzones based on significant geospatial characteristics with respect to the problem studied (e.g., he divides the area into subzones corresponding to municipalities in demographic problems, or corresponding to subzones with different climatic characteristics in meteorological problems). Since each subzone must contain a significant number of data points (otherwise the partition would be too fine with respect to the distribution of the patterns), we impose the constraint that each subzone must contain at least N_{th} patterns, where N_{th} stands for a threshold number. At each iteration, the HCMSPSO algorithm is applied to generate an optimal fuzzy system for each subzone. In Section 2, we introduce the HCMSPSO method, in Section 3 we present our method for finding the optimal partition of the area of study, in Section 4 we present some experimental results, and Section 5 is conclusive.

2. The HCMSPSO Algorithm: An Overview

The HCMSPSO algorithm [12] is a method based on the PSO algorithm for determining the optimal TSK-fuzzy system by using a set of patterns. The HCMSPSO determines the number r of rules and the optimal values of the parameters m_{ij}, σ_{ij}, and a_{ij} of the membership functions in each rule. In [12, 17–27], variations of the PSO algorithm are proposed.

The HCMSPSO method originates from the cluster-based particle swarm optimization method (shortly, CPSO) [28], in which each swarm is used independently for optimizing a set of parameters. In HCMSPSO, each swarm forms a species and the number of species is set to the number of fuzzy rules r. each species is formed from P_s particles, and each particle in a species represents a single fuzzy rule. The ith species is used for optimizing the parameters in the ith fuzzy rule. The position $\mathbf{s}_i^q$ of the qth particle in the ith species is given from the $(2n+1)$-dimensional vector:

$$\mathbf{s}_i^q = \left(m_{i1}^q, \ldots, m_{in}^q, \sigma_{i1}^q, \ldots, \sigma_{in}^q, a_{i0}^q\right) \in \Re^{2n+1} \tag{9}$$

and from the $(3n+1)$-dimensional vector:

$$\mathbf{s}_i^q = \left(m_{i1}^q, \ldots, m_{in}^q, \sigma_{i1}^q, \ldots, \sigma_{in}^q, a_{i0}^q, a_{i1}^q, \ldots, a_{in}^q\right) \in \Re^{3n+1}. \tag{10}$$

for a TSK-fuzzy system of first order. A TSK-fuzzy system of zero order is built by choosing one for each species of the P_s particles. To determine the final number of rules and generate the P_s particles in each species, the following iterative process is used:

(1) initially we set $r = 1$; we consider the first input pattern $\mathbf{x}(0)$, forming the first particle in the first species setting $m_{1j}^1 = x_j(0)$ and $\sigma_{1j}^1 = \sigma_{\mathrm{Init}}$, where σ_{Init} is a predefined value by determining the initial width of each fuzzy set;

(2) we generate all the particles in the first species, associated to the fuzzy rule R_1; the qth particle is given by the formula (for TSK-fuzzy system of first order):

$$\begin{aligned}\mathbf{s}_1^q = \Big(&m_{11} + \Delta m_{11}^q, \ldots, m_{in} + \Delta m_{1n}^q,\\ &\sigma_{11} + \Delta\sigma_{11}^q, \ldots, \sigma_{1n} + \Delta\sigma_{1n}^q, a_{10}^q, a_{11}^q, \ldots, a_{1n}^q\Big),\end{aligned} \tag{11}$$

where $q = 1, \ldots, P_s$, Δm_{1j}^q and $\Delta\sigma_{1j}^q$ represent small variations to m_{1j} and σ_{1j}, respectively, generated from the interval $[-0.1, 0.1]$; the values $a_{10}^q, a_{11}^q, \ldots, a_{1n}^q$ are obtained randomly in the output y range;

(3) for each successive pattern $\mathbf{x}(k)$, $k = 2, \ldots, N$, we consider the rule with maximum firing strength given by

$$I(k) = \arg\max_{1\le i\le r}[\mu_i(\mathbf{x}(k))]. \tag{12}$$

If results $\mu_{I(k)}(\mathbf{x}(k)) \le \mu_{\mathrm{th}}$, where μ_{th} is a predefined threshold, then a new rule is generated by setting

$$\begin{aligned} m_{(r+1)j}(k) &= x_j(k), \quad j = 1, \ldots, n,\\ \sigma_{(r+1)j}(k) &= \alpha \sum_{j=1}^{n}\left(\frac{x_j(k) - m_{I_k j}}{\sigma_{I_k j}}\right)^2, \quad j = 1, \ldots, n,\end{aligned} \tag{13}$$

where $\alpha > 0$ determines the degree of overlapping between two clusters;

(4) we generate all the particles in the $(r+1)$-species, associated to the fuzzy rule $R_{(r+1)}$; the qth particle is given by the formula (for TSK-fuzzy system of first order):

$$\begin{aligned}&\mathbf{s}_{(r+1)}^q(k)\\ &= \Big(m_{(r+1)1}(k) + \Delta m_{(r+1)1}^q(k), \ldots, m_{(r+1)n}(k)\\ &\quad + \Delta m_{(r+1)n}^q(k), \sigma_{(r+1)1}(k) + \Delta\sigma_{(r+1)1}^q(k), \ldots, \sigma_{(r+1)n}(k)\\ &\quad + \Delta\sigma_{(r+1)n}^q(k), a_{(r+1)0}^q, a_{(r+1)1}^q, \ldots, a_{(r+1)n}^q\Big),\end{aligned} \tag{14}$$

where $q = 1, \ldots, P_s$, $k = 2, \ldots, N$, $\Delta m_{(r+1)j}^q(k)$ and $\Delta\sigma_{(r+1)j}^q(k)$ represent small variations to $m_{(r+1)j}(k)$ and $\sigma_{(r+1)j}(k)$, respectively, from the interval $[-0.1, 0.1]$; the related values $a_{(r+1)0}^q, a_{(r+1)1}^q, \ldots, a_{(r+1)n}^q$ are obtained randomly from an interval identical to the output y range;

(5) we set $r = r + 1$ and iterate the steps (3) and (4) for all the patterns. At the end of the process, we have generated r rules and r species with P_s particles.

For each species, a partition of its particles is created in subspecies. A subspecies is a cluster of particles of a species. To create a subspecies partition of the particle of a species, we need to sort the particle of the species. The index used for sorting the particles of a species is the root means square error (RMSE) defined as

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{k=1}^{N}\left[y'(k) - y(k)\right]^2}{N}}, \tag{15}$$

where $y'(k)$ is the output calculated using the defuzzification formula (5), and $y(k)$ is the output value of the kth pattern. The successive steps are used for sorting the particles in each species based on the increasing value of the RMSE and partitioning each species into subspecies:

(1) for each combination of particles of each species, we determine the set of particles $(\mathbf{s}_1^g, \ldots, \mathbf{s}_r^g)$ with minimum RMSE. For each species, we set $\mathbf{s}_i^1 = \mathbf{s}_i^g$;

(2) for sorting the particles in the ith species, we calculate the RMSE produced by the combination $(\mathbf{s}_1^g, \ldots, \mathbf{s}_{i-1}^g, \mathbf{s}_i^q, \mathbf{s}_{i+1}^g, \ldots, \mathbf{s}_r^g)$ with $q = 2, \ldots, P_s$; then sorting the particles based on the increasing value of the corresponding RMSE. This step is repeated for all r species.

After sorting each species, we can do a partition of them into subspecies; the first particle of a subspecies, that is, the particle of the subspecies with least RMSE, is called leader of the subspecies. The next steps are used for partitioning each ordered species in subspecies:

(1) for the ith species, we set the number of subspecies $n_i = 1$ and create the first subspecies by setting the leader of this subspecies as the first particles of the species, $\mathbf{p}_i^{\mathrm{leader}(n_i)} = \mathbf{s}_i^g$;

(2) then we consider the successive particles of the species $\mathbf{s}_i^q$ with $q = 2, \ldots, P_s$ and calculate the index distance between the particle $\mathbf{s}_i^q$ and the leader of the first subspecies:

$$\psi\left(\mathbf{p}_i^{\mathrm{leader}(n_i)}, \mathbf{s}_i^q\right) = \sqrt{\frac{\left\|\mathbf{p}_i^{\mathrm{leader}(n_i)} - \mathbf{s}_i^q\right\|}{L}}, \tag{16}$$

Table 1: Dimension of the pattern subsets included in each subzone.

Subzone	Class	Number of buildings
1	Industrial area	64
2	Industrial area	75
3	Rural area	1306
4	Inhabited nucleus	586
5	Inhabited nucleus	126
6	Urban center	192
7	Rural area	1160

where $L = 2n$ (resp., $L = 3n$) for a TSK-fuzzy system of zero (resp., first) order. If we have $\psi(\mathbf{p}_i^{\text{leader}(n_i)}, \mathbf{s}_i^q) < \psi_{\text{th}}$, where ψ_{th} is a threshold value, then we assign the particle $\mathbf{s}_i^q$ to this subspecies; otherwise we create a new subspecies by setting $n_i = n_i + 1$;

(3) we iterate the steps (8) and (9) for all the r species.

In the last steps, the PSO algorithm is applied; we define the particles in the same subspecies as neighbours of a particle. The best global position of the neighbours of a particle $\mathbf{s}_i^q$, enclosed in the kth subspecies of the ith species at the iteration time $(t-1)$, is given from the best position of the leader of the subspecies $\mathbf{p}_i^{\text{leader}(k)}$ until the iteration time $(t-1)$. The best local position of the particle $\mathbf{s}_i^q$ enclosed in the kth subspecies of the ith species at the iteration time $(t-1)$ is given by the best position $\mathbf{p}_i^q$ of the particle until the iteration time $(t-1)$.

3. Subzones in the Generation Process

Our method is an iterative process that determines the optimal partition in subzones of the area of study; each subzone represents a zone of the area of study, homogeneous with respect to a specific TSK-fuzzy system composed from r fuzzy rules in the form (1). The expert creates an initial finer partition of the area of study according to specific local features (of type sociological, climatic, orographic, hydrological, etc.); the pattern data set is divided into subsets such that a subset is spatially included in the corresponding subzone. The successive step is performed to verify that the data distribution is consistent with the partition into subzones of the area of study; we verify that the number of patterns inside each subzone is greater or equal to a specific threshold value which can be set by the user. Clearly, the higher this value is, more the low resultant RMSE is expected; therefore, the greater the accuracy of the resulting TSK-fuzzy system will be. If N_h is the dimension of the subset of patterns inside the hth subzone and N_{th} is the threshold value, we impose the following constraint for each subzone:

$$N_h \geq N_{\text{th}}, \quad h = 1, \dots, Z, \tag{17}$$

where Z is the cardinality of the partition such that $N = \sum_{h=1}^{Z} N_h$. We consider consistently the pattern dataset with the partition into subzones of the area of study if (17) is true; otherwise, the expert creates a more coarse-grained partition; this control is iterated until each subset of patterns is consistent with respect to the corresponding subzone. For each subzone, the HCMSPSO method is applied for generating an optimal TSK-fuzzy system; we associate to each subzone the TSK-fuzzy system determined and its RMSE.

We compare the TSK-fuzzy systems of adjacent subzones, calculating the similarity index S^{hk} as in (8). If S^{hk} is greater or equal to a threshold value $S^{\text{threshold}}$, the hth and kth subzones are merged. When two or more subzones are merged in a new subzone, we group the corresponding pattern subsets together into a single subset, and we restart the HCMSPSO algorithm for the new subzone. This process is iterated until we have that $S^{hk} < S^{\text{threshold}}$ for all hth and kth adjacent subzones. As final result a thematic map is produced in which the area of study is divided into the final subzones classified according to the RMSE of the TSK-fuzzy system generated. To compare homogeneous errors, we calculate two normalized errors used in the literature:

(i) the normalized root mean square Error index (shortly, NRMSE) which is the rapport between the RMSE and the range (this last is given from the difference between maximal and the minimal values of the output variable y, in absolute value). We define the NRMSE with the following percentage:

$$\text{NRMSE} = \frac{\text{RMSE}}{|y_{\max} - y_{\min}|} \times 100; \tag{18}$$

(ii) the coefficient of variation of the RMSE error (shortly, CVRMSE) which is the rapport between the RMSE and the mean value $\overline{y}$ of the output variable y. We define the CVRMSE with the following percentage:

$$\text{CVRMSE} = \frac{\text{RMSE}}{\overline{y}} \times 100. \tag{19}$$

The NRMSE and CVRMSE are used in the creation of thematic maps of the TSK-fuzzy systems. The expert can fix a reliability threshold for the TSK-fuzzy system; in subzones with index greater than this threshold, it is necessary to use additional data and/or eliminate data with noise and outliers. The process described above can be schematized in the following steps:

(1) the expert divides initially the area of study into Z subzones, partitioning the area in homogeneous zones; this partition represents the finer partition desired by the expert;

(2) the pattern dataset is partitioned in subsets of data; each subset contains measured data georeferred into a specific subzone; if the dimension of each pattern subsets is less than a prefixed threshold N_{th}, the partition is too fine with respect to the pattern dataset and the process return to step (1); in this case the expert must create a more coarse grained subzone partition of the area of study;

(3) for each subzone, we use the HCMSPSO method to determine the number of rules and generate the correspondent species;

(4) we use the HCMSPSO method to generate the subspecies and optimize the parameters in each rule;

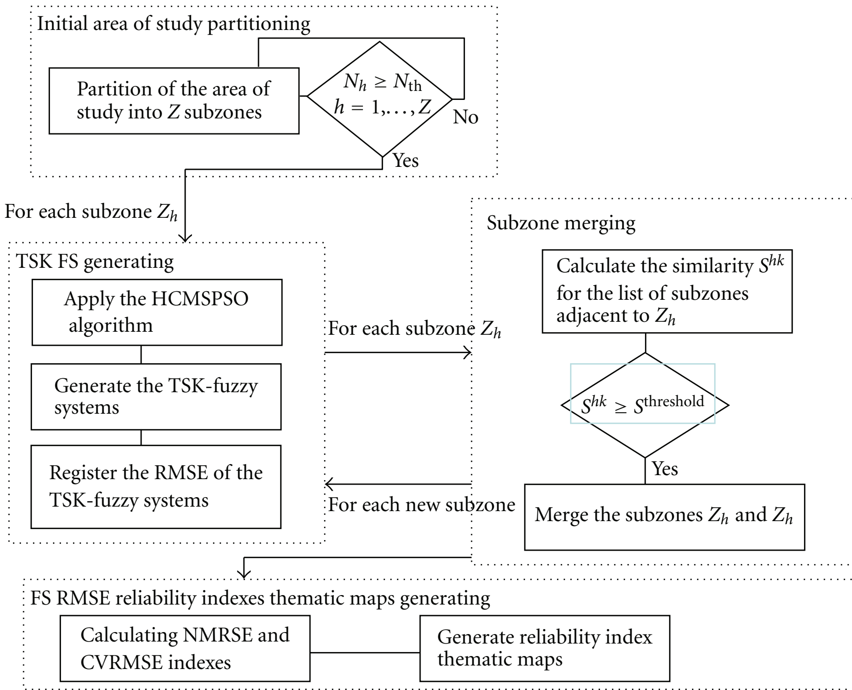


Figure 1: Schema of the subzones generation process.

Table 2: Final number of rules and RMSE obtained for each TSK-fuzzy system.

Subzone	Number of rules	RMSE	NRMSE	CVRMSE	t_p
1	7	15.281	2.942%	5.736%	1.2×10^5
2	7	22.214	4.738%	8.175%	1.3×10^5
3	9	239.023	45.020%	83.677%	1.5×10^5
4	7	87.727	18.465%	32.460%	1.3×10^5
5	7	38.369	9.820%	14.518%	1.4×10^5
6	7	53.223	10.467%	18.835%	1.4×10^5
7	8	162.931	30.688%	57.169%	1.5×10^5

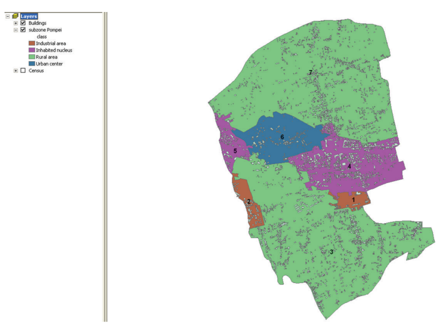


Figure 2: Initial subzones of Pompeii based on the city planning class.

(5) we compare the TSK-fuzzy system calculated to two adjacent subzones z_h and z_k, by calculating the similarity index S^{hk}; if $S^{hk} \geq S^{\text{threshold}}$, where $S^{\text{threshold}}$ is a predefined

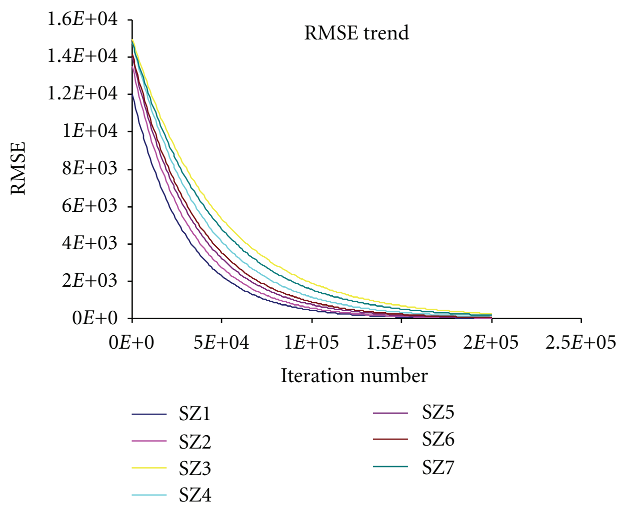


Figure 3: RMSE trend obtained for the seven subzones.

Table 3: Similarity index between adjacent subzones.

Subzone 1	Subzone 2	S^{hk}
4	6	0.388
5	6	0.326
1	4	0.078
2	5	0.073

Table 4: Breaks of the three thematic classes of NRMSE and CVRMSE.

Class	NRMSE class breaks	CVRMSE class breaks
Low	0%–10%	0%–20%
Mean	10,001%–30%	20,001%–50%
High	Over 30%	Over 50%

threshold, the two subzones, are merged in one subzone and the number of subzones is $Z-1$;

(6) if there are two or more subzones merged, we iterate steps (3), (4), and (5) for each new subzone;

(7) two thematic maps are created by showing for each final subzone the reliability class correspondent to the final NRMSE and CVRMSE.

In Figure 1, we have schematized the above process. If we are interested to analyze whether the final distribution of the final subzones is or not approximately uniform, we can calculate the coefficient of variation of this distribution; this index is extracted by calculating the mean and standard deviation of the distribution of the N_h's as

$$m_N = \frac{\sum_{h=1}^{Z} N_h}{Z},$$
$$\sigma_N = \sqrt{\frac{\sum_{h=1}^{Z}\left(N_h - m_N\right)^2}{Z-1}}. \tag{20}$$

The coefficient of variation, expressed in percentage, is given by

$$\mathrm{CV}_N = \frac{\sigma_N}{m_N} \times 100. \tag{21}$$

We can consider the pattern data distribution approximately uniform if $\mathrm{CV}_N \leq 50\%$; this control can be used at the end of the process by the expert if he intends to verify if the final pattern dataset is approximately uniform with respect to the final partition of the area of study. This analysis is useful to see that significant differences in the RMSE on the area of study can be due to substantial changes in the cardinality of the final subsets of patterns inside each final subzone. In Section 4, we present some results of our tests applied on spatial datasets. The HCMSPSO algorithm has been implemented and encapsulated in the tool GIS ESRI/ArcGIS, release 10. A first experiment is made on test spatial data firstly by comparing the results obtained with the HCMSPSO method and the ones obtained using other PSO-based methods, secondly for verifying the accuracy of our method in the subzones merging process. Then, we present the results obtained by applying our method on a problematic related to the valuation of the costs for building maintenance in the city of Pompeii (Italy).

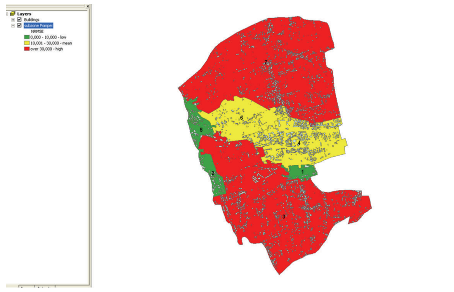


Figure 4: Thematic map of NRMSE.

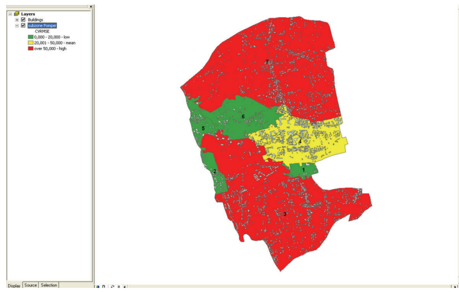


Figure 5: Thematic map of CVRMSE.

4. Results of Tests

Now, we apply our method to geospatial data. The test concerns the buildings of the municipality of Pompeii, which is a famous touristic city for the presence of an important Sanctuary and of a large famous archeological heritage. The municipal area is partitioned into four classes: rural area, urban center, inhabited nucleus, and industrial area. An expert's goal is to plan the cost of maintenance of buildings, based to the building maintenance data. The data are extracted from the related dataset, and they are the following: x_1 : year of construction, x_2 : year of last maintenance, x_3 : extent of the damage with respect to the volume of the building (0 : null extension and 1 : complete extension), x_4 : gravity of the damage (0 : null gravity and 1 : maximum gravity), y : cost of the maintenance (calculated in thousands of Euro). Each pattern corresponds to a building georeferred to a specific census microzone. The initial subzones are formed by union of adjacent microzones with a same city planning class, and we obtain $Z = 7$ subzones. The thematic map in Figure 2 shows the seven subzones obtained by the subdivision of the area of Pompeii based on the four classes.

The dataset is formed by $N = 3445$ buildings (cf. Table 1); we set $N_{\mathrm{th}} = 50$. We have that N_h is greater than $N_{\mathrm{th}} = 50$ for every $h = 1, \ldots, Z$, we can assume that the set of data is suitable for the partition. Then we use this subdivision in 7 subzones of Pompeii for generating the relative TSK-fuzzy system of zero order. In the HCMSPSO method, we set the threshold values μ_{th} and ψ_{th} to 0.03 and 2, respectively; the number of iterations is 200.000. The graph in Figure 3 shows the RMSE trend with respect to the iteration number obtained for the seven subzones. For all the seven TSK-fuzzy systems generated, the RMSE trend reaches a plateau after about 1.5×10^5 iterations. Table 2 shows the results, and we report the final number of rules, the RMSE, NRMSE, CVRMSE, and the index t_p. The results show that the TSK-fuzzy systems related to the subzones 3 and 7 have RMSE and t_p greater values with respect to the ones related to the other subzones. Then, we calculated the similarity indexes between adjacent subzones; we set $S^{\text{threshold}} = 0.7$.

The similarity values between two TSK-fuzzy systems with same number of rules, associated with adjacent subzones are given in Table 3. Table 4 contains breaks of the three thematic classes of NRMSE and CVRMSE: we consider not sufficiently reliable the results obtained in subzones belonging to the class "high," that is, subzones with NMRSE greater then 30% or CVRMSE greater than 50%. The results in Figures 4 and 5 show that the resultant TSK-fuzzy system obtained for the rural areas 1 and 7 are not sufficiently reliable. These subzones contain the greatest number of patterns; the results confirm that the costs of maintenance of a building in rural areas are probably not significantly related to parameters like the last year of construction and maintenance.

5. Conclusions

In this paper, we present a method based on the HCMSPSO algorithm for extracting a TSK-fuzzy system from a dataset of georeferred measured data for spatial analysis. The area of study is partitioned initially into subzones by the expert; for each subzone is extracted the TSK-fuzzy system by using the subset of patterns georeferred into the subzone. We merge subzones with similar TSK-fuzzy system and by recreating the TSK-fuzzy system for the new subzone. For each TSK-fuzzy system, the reliability is evaluated using indexes based on the final RMSE. The algorithm has been implemented in the tool ESRI/ArcGIS, release 10; the results of our tests show that this method can be well used in a GIS platform and encapsulated as decision support system for optimizing fuzzy systems related to subzones of the area of study.

References

[1] O. Al-Jarrah and H. Abu-Qdais, "Municipal solid waste landfill siting using intelligent system," *Waste Management*, vol. 26, no. 3, pp. 299–306, 2006.

[2] A. Ertug Gunes and J. P. Kovel, "Using GIS in emergency management operations," *Journal of Urban Planning and Development*, vol. 126, no. 3, pp. 136–149, 2000.

[3] P. Kumar, V. Singh, and D. Reddy, "Advanced traveler information system for Hyderabad City," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 26–37, 2005.

[4] P. C. Nayak and S. K. Jain, "Modelling runoff and sediment rate using a neuro-fuzzy technique," *Proceedings of the Institution of Civil Engineers*, vol. 164, no. 4, pp. 201–209, 2011.

[5] E. V. Ocalir, O. Y. Ercoskun, and R. Tur, "An integrated model of GIS and fuzzy logic (FMOTS) for location decisions of taxicab stands," *Expert Systems with Applications*, vol. 37, no. 7, pp. 4892–4901, 2010.

[6] A. P. Tang, L. L. Xie, and X. X. Tao, "Application of GIS to build earthquake emergency response system for urban area," *Journal of Harbin Institute of Technology*, vol. 9, no. 1, pp. 38–42, 2002.

[7] C. H. Tsai, W. C. Wang, C. W. Chen, H. L. Chen, and M. L. Lin, "The construction of decision model for tourism disaster evacuation based on GIS and fuzzy theory," *World Academy of Science, Engineering and Technology*, vol. 54, pp. 81–85, 2009.

[8] M. H. Vahidnia, A. A. Alesheikh, and A. Alimohammadi, "Hospital site selection using fuzzy AHP and its derivatives," *Journal of Environmental Management*, vol. 90, no. 10, pp. 3048–3056, 2009.

[9] D. Zheng and and W. Kainz, "Fuzzy rule extraction from GIS data with a neural fuzzy system for decision making," in *Proceedings of the 7th International Symposium on Advances in Geographic Information Systems (ACM-GIS '99)*, pp. 79–84, Kansas, Kan, USA, November 1999.

[10] M. Sugeno and G. T. Kang, "Structure identification of fuzzy model," *Fuzzy Sets and Systems*, vol. 28, no. 1, pp. 15–33, 1988.

[11] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.

[12] C. F. Juang, C. M. Hsiao, and C. H. Hsu, "Hierarchical cluster-based multispecies particle-swarm optimization for fuzzy-system optimization," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 1, pp. 14–26, 2010.

[13] F. Di Martino, S. Sessa, and V. Loia, "A fuzzy-based tool for modelization and analysis of the vulnerability of aquifers: a case study," *International Journal of Approximate Reasoning*, vol. 38, no. 1, pp. 99–111, 2005.

[14] F. Di Martino, V. Loia, S. Sessa, and M. Giordano, "An evaluation of the reliability of a GIS based on fuzzy logic, in a concrete case study," in *Fuzzy Modeling with Spatial Information for Geographic Problems*, F. E. Petry, V. B. Robinson, and M. A. Cobb, Eds., pp. 185–208, Springer, Heidelberg, Germany, 2005.

[15] F. Di Martino, V. Loia, S. Sessa, and S. B, "By using fuzzy rules for applications to risk analysis in the Vesuvian area," *Contemporary Engineering Sciences*, vol. 4, no. 5, pp. 55–78, 2011.

[16] D. Kukolj, "Design of adaptive Takagi-Sugeno-Kang fuzzy models," *Applied Soft Computing Journal*, vol. 2, no. 2, pp. 89–103, 2002.

[17] P. Bajpai and S. N. Singh, "Fuzzy adaptive particle swarm optimization for bidding strategy in uniform price spot market," *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 2152–2160, 2007.

[18] D.-B Chen, F. Zou, and Y.-T. Wang, "A multi-objective endocrine PSO algorithm and application," *Applied Soft Computing*, vol. 11, no. 8, pp. 4508–4520, 2011.

[19] X. Chen and Y. Li, "A modified PSO structure resulting in high exploration ability with convergence guaranteed," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 37, no. 5, pp. 1271–1289, 2007.

[20] C. F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 34, no. 2, pp. 997–1006, 2004.

[21] D. Parrott and X. Li, "Locating and tracking multiple dynamic optima by a particle swarm model using speciation," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 440–458, 2006.

[22] K. E. Parsopóulos and M. N. Vrahatis, "Particle swarm optimization method in multiobjective problems," in *Proceedings of ACM Symposium on Applied Computing*, pp. 603–607, Madrid, Spain, March 2002.

[23] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.

[24] Y. Shi and R. Eberhart, "Modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, May 1998.

[25] W. Jingxuan and W. Yuping, "Multi-objective fuzzy particle swarm optimization based on elite archiving and its convergence," *Journal of Systems Engineering and Electronics*, vol. 19, no. 5, pp. 1035–1040, 2008.

[26] Z. Xinchao, "A perturbed particle swarm algorithm for numerical optimization," *Applied Soft Computing Journal*, vol. 10, no. 1, pp. 119–124, 2010.

[27] W. Zhang and Y. Liu, "Multi-objective reactive power and voltage control based on fuzzy optimization strategy and fuzzy adaptive particle swarm," *International Journal of Electrical Power and Energy Systems*, vol. 30, no. 9, pp. 525–532, 2008.

[28] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to participle swam optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.

Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2012, Article ID 785709, 6 pages
doi:10.1155/2012/785709

Research Article

A New Time-Invariant Fuzzy Time Series Forecasting Method Based on Genetic Algorithm

Erol Eğrioğlu

Department of Statistics, Faculty of Arts and Science, University of Ondokuz Mayıs, 55139 Samsun, Turkey

Correspondence should be addressed to Erol Eğrioğlu, erole@omu.edu.tr

Received 8 April 2012; Revised 3 May 2012; Accepted 13 May 2012

Academic Editor: Ferdinando Di Martino

In recent years, many fuzzy time series methods have been proposed in the literature. Some of these methods use the classical fuzzy set theory, which needs complex matricial operations in fuzzy time series methods. Because of this problem, many studies in the literature use fuzzy group relationship tables. Since the fuzzy relationship tables use order of fuzzy sets, the membership functions of fuzzy sets have not been taken into consideration. In this study, a new method that employs membership functions of fuzzy sets is proposed. The new method determines elements of fuzzy relation matrix based on genetic algorithms. The proposed method uses first-order fuzzy time series forecasting model, and it is applied to the several data sets. As a result of implementation, it is obtained that the proposed method outperforms some methods in the literature.

1. Introduction

The fuzzy time series were firstly defined in Song and Chissom [1]. Many time series in the real life have uncertainty observations. This kind of the time series is called fuzzy time series. For example, some of these time series are stock index data, air pollution data, enrollment data, and temperature data. The observations of these time series are convertible to fuzzy sets. The fuzzy time series separate two classes which are time-variant and time-invariant. The time-invariant fuzzy time series have time-invariant relationship of lagged fuzzy time series variables. This relationship is proved from an "R" matrix, which is invariant in the time space. The first method for forecasting time-invariant time series is proposed in Song and Chissom [2], in which the membership of observations is determined, subjectively. Chen [3] proposed fuzzy time series method which does not need complex matricial operations, and it uses fuzzy group relationship tables. Since these use order of fuzzy sets, the membership functions of fuzzy sets are not taken into consideration. Sullivan and Woodall [4] proposed fuzzy time series method based on Markov Chains. All of these methods forecast fuzzy time series based on first-order fuzzy time series forecasting model.

In the literature, many fuzzy time series forecasting method are based on high-order fuzzy time series model, bivariate fuzzy time series model, and multivariate fuzzy time series model. Some of these methods are due to Chen [5], Aladag et al. [6], and Egrioglu et al. [7]. Generally, fuzzy time series methods are based on three stages. These are fuzzification, determination of fuzzy relation, and defuzzification. In the literature, the fuzzy time series methods are improved by employing various artificial intelligence techniques in these three stages. The genetic algorithms, particle swarm optimization, and fuzzy c-means methods are used in the fuzzification stage. Feed forward neural networks are used determining fuzzy relation and defuzzification stage.

Many proposed methods in the literature neglected membership values of fuzzy sets. In this study, we proposed new fuzzy time series forecasting method for first order fuzzy time series forecasting model. The proposed method takes into account of membership values. Moreover, the proposed method does not need complex matricial operations and outperforms the well-known methods in the literature.

In Section 2 are given some important definitions. In Section 3, the fuzzy c-means method is summarized. In Section 4, genetic algorithm is briefly explained. In Section 5, we present details of our proposed methods. In Section 6, the proposed method is applied to some data sets, and we present results of implementation.

2. The Some Definitions Related to Fuzzy Time Series

The fuzzy time series were firstly defined in Song and Chissom [1]. The time-variant and time-invariant fuzzy time series definitions are given below.

Definition 1. Let $Y(t)$ $(t = \ldots, 0, 1, 2, \ldots)$, a subset of real numbers, be the universe of discourse on which fuzzy sets $f_j(t)$ are defined. If $F(t)$ is a collection of $f_1(t)$, $f_2(t), \ldots$, then $F(t)$ is called a fuzzy time series defined on $Y(t)$.

Definition 2. Suppose $F(t)$ is implied by $F(t-1)$ only, that is, $F(t-1) \rightarrow F(t)$. Then this relation can be expressed as $F(t) = F(t-1) \circ R(t, t-1)$, where $R(t, t-1)$ is the fuzzy relationship between $F(t-1)$ and $F(t)$, and $F(t) = F(t-1) \circ R(t, t-1)$ is called the first order model of $F(t)$.

Definition 3. Suppose $R(t, t-1)$ is a first-order model of $F(t)$. If for any t, $R(t, t-1)$ is independent of t, that is, for any t, $R(t, t-1) = R(t-1, t-2)$, then $F(t)$ is called a time-invariant fuzzy time series, otherwise it is called a time-variant fuzzy time series.

The symbol "$\circ$" stands for max-min composition of fuzzy sets. Song and Chissom [2] firstly introduced an algorithm based on the first order model for forecasting time invariant $F(t)$. In Song and Chissom [2], the fuzzy relationship matrix $R(t, t-1) = R$ is obtained by many matricial operations. The fuzzy forecasts are obtained based on max-min composition as follows:

$$F(t) = F(t-1) \circ R. \tag{1}$$

The dimension of R is depending on number of fuzzy sets which are partition number of universe and discourse. If we use more fuzzy sets, we need more matricial operations for obtaining R. In this situation, Song and Chissom's [2] method is getting more complex.

3. The Fuzzy C-Means Clustering Method: An Overview

Song and Chissom [1] method uses decomposition of the universe discourse in the stage of fuzzification. There are two problems: the number of intervals of arbitrary length and the arbitrary choice of membership degrees. In order to overcome these problems, Cheng et al. [8] and Li et al. [9] fuzzy c-means (FCM) clustering method was used. FCM clustering method was firstly introduced by Bezdek [10]. Let u_{ij} be the membership, v_i the center of cluster, n the number of variables, and c the number of clusters. Then the objective function, which is minimized in FCM, is

$$J_\beta(X, V, U) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{\beta} d^2\left(x_j, v_i\right), \tag{2}$$

Table 1: Enrollment data.

Years	Actual	Years	Actual
1971	13055	1982	15433
1972	13563	1983	15497
1973	13867	1984	15145
1974	14696	1985	15163
1975	15460	1986	15984
1976	15311	1987	16859
1977	15603	1988	18150
1978	15861	1989	18970
1979	16807	1990	19328
1980	16919	1991	19337
1981	16388	1992	18876

where β is a constant greater than 1 and $d(x_j, v_i)$ is a distance between the observation and the center of the cluster. J_β is minimized with subject the constraints:

$$\begin{gathered} 0 \le u_{ij} \le 1, \quad \forall i, j, \\ 0 < \sum_{j=1}^{n} u_{ij} \le n, \quad \forall i, \\ \sum_{i=1}^{c} u_{ij} = 1, \quad \forall j. \end{gathered} \tag{3}$$

In this method, the minimization is done by an iterative algorithm. In each repetition, the values of u_{ij} and v_i are updated by the formulas given in (4):

$$\begin{gathered} v_i = \frac{\sum_{j=1}^{n} u_{ij}^{\beta} x_j}{\sum_{j=1}^{n} u_{ij}^{\beta}}, \\ u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(d(x_j, v_i)/d(x_j, v_k)\right)^{2/(\beta-1)}}. \end{gathered} \tag{4}$$

4. Genetic Algorithm: An Overview

Chen and Chung [11], Lee et al. [12] used genetic algorithm in fuzzification stage. The genetic algorithms were first proposed by Holland [13]. The genetic algorithms have population size, evaluation function, cross-over rate, mutation rate and maximum generation number. The genetic algorithm researches an optimal solution with many chromosomes. In a chromosome, there are many genes. Generally, the genetic algorithm starts with random population whose size is determined by user according to the problem under study. For example, Chen and Chung [11] used 50 as population size and the new generation is produced by various techniques in the iterations. Some of these techniques are crossover, mutation, and natural selection, summarized as follows.

Crossover. The system randomly selects two chromosomes from a population and randomly selects a crossover point

Table 2: centers of clusters are obtained from FCM algorithm in Step 1.

Centers of clusters						
v_1	v_2	v_3	v_4	v_5	v_6	v_7
13480,6299	15003,98	15455,4	15965,12	16837,86	18161,53	19144,4

Table 3: The memberships of the observations to clusters are obtained from FCM algorithm in Step 1.

Years	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
1971	0,8889	0,0424	0,0279	0,0190	0,0113	0,0062	0,0043
1972	0,9925	0,0032	0,0019	0,0012	0,0006	0,0003	0,0002
1973	0,8072	0,0932	0,0478	0,0274	0,0137	0,0065	0,0043
1974	0,0486	0,7570	0,1245	0,0446	0,0157	0,0060	0,0036
1975	0,0000	0,0001	0,9998	0,0001	0,0000	0,0000	0,0000
1976	0,0048	0,1716	0,7757	0,0378	0,0069	0,0020	0,0011
1977	0,0039	0,0485	0,7993	0,1328	0,0114	0,0027	0,0014
1978	0,0017	0,0135	0,0601	0,9116	0,0104	0,0019	0,0009
1979	0,0001	0,0003	0,0005	0,0013	0,9971	0,0005	0,0002
1980	0,0005	0,0018	0,0030	0,0071	0,9821	0,0042	0,0013
1981	0,0093	0,0409	0,0900	0,4378	0,3869	0,0249	0,0103
1982	0,0001	0,0027	0,9950	0,0018	0,0003	0,0001	0,0000
1983	0,0004	0,0070	0,9835	0,0078	0,0009	0,0002	0,0001
1984	0,0057	0,7977	0,1647	0,0236	0,0055	0,0017	0,0010
1985	0,0066	0,7367	0,2179	0,0290	0,0066	0,0021	0,0012
1986	0,0001	0,0004	0,0013	0,9977	0,0005	0,0001	0,0000
1987	0,0000	0,0001	0,0002	0,0006	0,9987	0,0003	0,0001
1988	0,0000	0,0000	0,0000	0,0000	0,0001	0,9997	0,0001
1989	0,0010	0,0018	0,0023	0,0032	0,0063	0,0438	0,9416
1990	0,0009	0,0017	0,0022	0,0029	0,0052	0,0239	0,9632
1991	0,0010	0,0019	0,0024	0,0031	0,0057	0,0258	0,9601
1992	0,0021	0,0041	0,0052	0,0072	0,0147	0,1196	0,8472

from the two selected chromosomes to exchange genes after the crossover point. The crossover operation is depending on crossover rate. The random number generates uniform distribution, then the crossover operation is applied if random number is bigger than crossover rate.

Mutation. The user must determine a mutation rate. Then a chromosome is randomly selected. If the system randomly generates a real value between zero and one, which is smaller than or equal to the mutation rate, then the system performs the mutation operation with a randomly selected gene from the chromosomes.

Natural Selection. Each chromosome of any generation is evaluated according to evaluation function. All chromosomes are ordered according to evaluation function value. The best chromosomes are transferred to the next generation. Some worst chromosomes are discarded from generations, and the new chromosomes are admitted to the new generation.

5. The Proposed Method for Forecasting First Order Fuzzy Time Series Method

Here, we study forecasting time-invariant fuzzy time series. The FCM is used for fuzzification of time series, and genetic algorithm is used for finding the fuzzy relation. The proposed method has the following important advantages:

(i) the membership values of the fuzzy sets are taken into consideration;

(ii) there is no necessity of complicated matricial operations;

(iii) FCM no needs subjective decisions by using partition of the universe of discourse.

The algorithm is given below in step by step as follows.

Step 1 (Time series are fuzzified by FCM). Let c be the number of fuzzy sets such that $2 \leq c \leq n$. FCM initially assigns crisp values to the fuzzy sets. After this application, the center of each cluster L_r is determined and denoted by v_r, $r = 1, 2, \ldots, c$.

Step 2 (Determination of the parameters of genetic algorithm). The evaluation function is the RMSE as follows:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(x_t - \hat{x}_t)^2}{n}}, \tag{5}$$

Table 4: The forecasts of the proposed method.

Years				Fuzzy forecasts				Defuzzified forecasts
1972	0,8694	0,8279	0,7866	0,3820	0,2837	0,1199	0,0430	13480,63
1973	0,8694	0,8279	0,7866	0,3820	0,2837	0,1199	0,0430	13480,63
1974	0,8072	0,8072	0,7866	0,3820	0,2837	0,1199	0,0478	14242,00
1975	0,3643	0,5920	0,7570	0,7570	0,6596	0,4888	0,1245	15710,00
1976	0,5000	0,9998	0,3584	0,9998	0,8535	0,3184	0,2807	15484,55
1977	0,5000	0,7757	0,3584	0,7757	0,7757	0,3184	0,2807	15935,65
1978	0,5000	0,7993	0,3584	0,7993	0,7993	0,3184	0,2807	15935,65
1979	0,4219	0,7284	0,2815	0,7736	0,9116	0,0601	0,9043	16837,86
1980	0,0022	0,9009	0,0558	0,3246	0,9971	0,9971	0,5000	17499,69
1981	0,0071	0,9009	0,0558	0,3246	0,9821	0,9821	0,5000	17499,69
1982	0,4219	0,4378	0,2815	0,4378	0,4378	0,3869	0,4378	16737,00
1983	0,5000	0,9950	0,3584	0,9950	0,8535	0,3184	0,2807	15484,55
1984	0,5000	0,9835	0,3584	0,9835	0,8535	0,3184	0,2807	15484,55
1985	0,3643	0,5920	0,7977	0,7977	0,6596	0,4888	0,1647	15710,00
1986	0,3643	0,5920	0,7367	0,7367	0,6596	0,4888	0,2179	15710,00
1987	0,4219	0,7284	0,2815	0,7736	0,9977	0,0496	0,9043	16837,86
1988	0,0022	0,9009	0,0558	0,3246	0,9987	0,9987	0,5000	17499,69
1989	0,0001	0,0001	0,5000	0,5000	0,5000	0,5000	0,9997	19144,40
1990	0,0032	0,0063	0,0438	0,0438	0,0438	0,5000	0,5621	19144,40
1991	0,0029	0,0052	0,0239	0,0239	0,0239	0,5000	0,5621	19144,40
1992	0,0031	0,0057	0,0258	0,0258	0,0258	0,5000	0,5621	19144,40

Table 5: The RMSE values of the some first order methods and proposed method.

Methods	RMSE
Song and Chissom [2]	642,26
Sullivan and Woodall [4]	621,33
Chen [3]	638,36
Proposed method	484,61

being x_t the crisp time series, $\hat{x}_t$ the defuzzified forecast, and n is the number of forecasts. The other parameters are selected as follows:

population size: 30;

crossover rate: 1;

mutation rate: 0.01;

maximum generation: 300;

number of discarded chromosomes in the natural selection: 10.

Step 3 (Initialization). The genes are the elements of the crisp relation (Figure 1):

$$R = \left[r_{ij}\right], \quad i = 1, 2, \ldots, c, \; j = 1, 2, \ldots, c. \tag{6}$$

Step 4 (Crossover operation). The crossover operation is applied if the random number is bigger than the crossover rate. The system randomly selects two chromosomes from a population and randomly selects a crossover point from the two selected chromosomes to exchange genes after the crossover point. The crossover operation is shown in Figure 2.

r_{11}	r_{12}	$\cdots$	r_{1c}	r_{21}	r_{22}	$\cdots$	r_{2c}	$\cdots$	r_{c1}	r_{c2}	$\cdots$	r_{cc}

Figure 1: The structure of one chromosome.

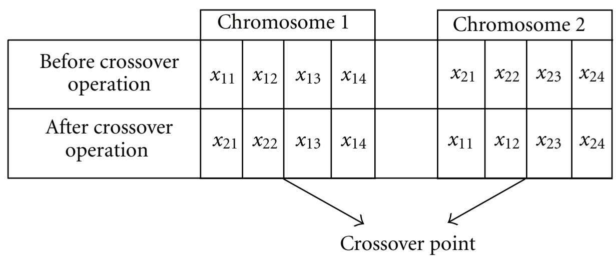


Figure 2: An example of crossover operation.

Step 5. If the system randomly generates a real value between zero and one which is smaller than or equal to the mutation rate, then the system performs the mutation operation with a randomly selected gene from the chromosomes. The mutation operation is demonstrated in Figure 3. x_{new} is randomly generated from $[x_{11}, x_{13}]$ interval.

Step 6. The method for calculating RMSE for any chromosome is given in Substeps 6.1–6.4.

Table 6: The results of the methods for IMKB 100 data.

Method		Number of fuzzy sets									Min	Mean
		7	8	9	10	11	12	13	14	15		
Chen [3]	RMSE (training)	1420	1178	1340	1164	1281	1233	1152	1205	1090	1090	1252
	RMSE (test)	1238	1132	1340	1191	1155	1131	1218	1125	1186	1125	1210
Song and Chissom [1, 2]	RMSE (training)	1462	1178	1340	1164	1281	1233	1140	1209	1105	1105	1257
	RMSE (test)	1571	1132	1340	1191	1155	1131	1249	1121	1132	1121	1240
Proposed Method	RMSE (training)	1379	1271	1391	1360	1176	1176	1152	1089	1107	1089	1234
	RMSE (test)	1565	1317	1291	1154	1118	1126	1039	1036	1022	1022	1185

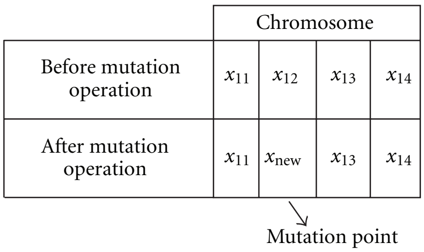


Figure 3: An example of mutation operation.

Substep 6.1. c-gens constitute in the rows of R.

Substep 6.2. Fuzzy forecasts are obtained using (1). After R is estimated, fuzzy forecasts can be obtained by using FCM algorithm. For example, for any t, the $(t-1)$th fuzzy observation is given as follows:

$$F(t-1) = [1\ 0.5\ 0.3\ 0\ 0], \qquad R = \begin{bmatrix} 1 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{7}$$

The fuzzy forecast for t time is computed by (1) as follows:

$$\begin{aligned} \widehat{F}(t) &= F(t-1) \circ R = [1\ 0.5\ 0.3\ 0\ 0] \circ \begin{bmatrix} 1 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \\ &= [1\ 0.5\ 0.5\ 0.3\ 0]. \end{aligned} \tag{8}$$

Substep 6.3. If the membership values of a fuzzy forecast have only one maximum, then select the center of this cluster as the defuzzified forecasted value. If the membership values of an fuzzy forecast has two or more maximum, then select the arithmetic mean of centers of the corresponding clusters as the defuzzified forecasted value. For example, if $\widehat{F}(t) = [0.7\ \ 0.5\ \ 0.3\ \ 0.2\ \ 0.4]$, then the maximum is 0.7 and then the fuzzified forecast is (v_1), center of L_1. If $\widehat{F}(t) = [0.7\ \ 0.5\ \ 0.7\ \ 0.2\ \ 0.4]$, then the maximum is 0.7 and then fuzzified forecast is arithmetic mean of v_1 center of L_1 cluster and v_3, center of L_3.

Figure 4: The IMKB 100 data between 28/12/2010 and 29/02/2012.

Substep 6.4. RMSE is computed by (5).

Step 7 (Natural selection operation). All chromosomes are ordered according to the RMSE value. The best 20 chromosomes are transferred to the next generation. 10 chromosomes are discarded from the generation. The new 10 randomly generated chromosomes are replaced in the new generation.

Step 8. Steps 4 and 7 are repeated 300 times.

6. Implementation

The proposed method is firstly applied to Alabama University Enrollment data (1971–1992) which is well known in the literature. The Enrollment data are given Table 1.

The proposed method is programmed in Matlab 7.9 version. In the application, we use seven fuzzy sets ($c = 7$) in Step 1. The centers of clusters and the membership values from FCM algorithm are given Tables 2 and 3, respectively. The fuzzy and defuzzified forecasts are given in Table 4.

Consider

$$R = \begin{bmatrix} 0,8694 & 0,8279 & 0,7866 & 0,3820 & 0,2837 & 0,1199 & 0,0430 \\ 0,3643 & 0,5920 & 1,0000 & 1,0000 & 0,6596 & 0,4888 & 0,0000 \\ 0,5000 & 1,0000 & 0,3584 & 1,0000 & 0,8535 & 0,3184 & 0,2807 \\ 0,4219 & 0,7284 & 0,2815 & 0,7736 & 1,0000 & 0,0496 & 0,9043 \\ 0,0022 & 0,9009 & 0,0558 & 0,3246 & 1,0000 & 1,0000 & 0,5000 \\ 0,0000 & 0,0000 & 0,5000 & 0,5000 & 0,5000 & 0,5000 & 1,0000 \\ 0,0000 & 0,0000 & 0,0000 & 0,0000 & 0,0000 & 0,5000 & 0,5621 \end{bmatrix}. \tag{9}$$

The RMSE of well-known methods [2–4] is given in Table 5.

The proposed method is successively applied to daily Istanbul stock market time series data between 28/12/2010 and 29/02/2012 dates (IMKB 100). The graph of the time series is given in Figure 4. The data between 28/12/2010 and 10/10/2011 dates are used as training data, the data between 11/10/2011 and 29/02/2012 dates are used as test data.

The number of fuzzy clusters is changed from 7 to 15. The obtained results from three methods are given in Table 6.

The RMSE values of the proposed method are smaller than Chen's [3] and Song and Chissom's [1, 2] methods.

References

[1] Q. Song and B. S. Chissom, "Fuzzy time series and its models," *Fuzzy Sets and Systems*, vol. 54, no. 3, pp. 269–277, 1993.

[2] Q. Song and B. S. Chissom, "Forecasting enrollments with fuzzy time series—part I," *Fuzzy Sets and Systems*, vol. 54, no. 1, pp. 1–9, 1993.

[3] S. M. Chen, "Forecasting enrollments based on fuzzy time series," *Fuzzy Sets and Systems*, vol. 81, no. 3, pp. 311–319, 1996.

[4] J. Sullivan and W. H. Woodall, "A comparison of fuzzy forecasting and Markov modeling," *Fuzzy Sets and Systems*, vol. 64, no. 3, pp. 279–293, 1994.

[5] S. M. Chen, "Forecasting enrollments based on high-order fuzzy time series," *Cybernetics and Systems*, vol. 33, no. 1, pp. 1–16, 2002.

[6] C. H. Aladag, M. A. Basaran, E. Egrioglu, U. Yolcu, and V. R. Uslu, "Forecasting in high order fuzzy times series by using neural networks to define fuzzy relations," *Expert Systems with Applications*, vol. 36, no. 3, pp. 4228–4231, 2009.

[7] E. Egrioglu, C. H. Aladag, U. Yolcu, V. R. Uslu, and M. A. Basaran, "A new approach based on artificial neural networks for high order multivariate fuzzy time series," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10589–10594, 2009.

[8] C. H. Cheng, G. W. Cheng, and J. W. Wang, "Multi-attribute fuzzy time series method based on fuzzy clustering," *Expert Systems with Applications*, vol. 34, no. 2, pp. 1235–1242, 2008.

[9] S. T. Li, Y. C. Cheng, and S. Y. Lin, "A FCM-based deterministic forecasting model for fuzzy time series," *Computers and Mathematics with Applications*, vol. 56, no. 12, pp. 3052–3063, 2008.

[10] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, NY, USA, 1981.

[11] S. M. Chen and N. Y. Chung, "Forecasting enrollments using high-order fuzzy time series and genetic algorithms," *International Journal of Intelligent Systems*, vol. 21, no. 5, pp. 485–501, 2006.

[12] L. W. Lee, L. H. Wang, and S. M. Chen, "Temperature prediction and TAIFEX forecasting based on fuzzy logical relationships and genetic algorithms," *Expert Systems with Applications*, vol. 33, no. 3, pp. 539–550, 2007.

[13] J. H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, Mass, USA, 1975.

Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2012, Article ID 217068, 8 pages
doi:10.1155/2012/217068

Research Article

A Hybrid PID-Fuzzy Control for Linear SISO Systems with Variant Communication Delays

Adrian-Bogdan Hanchevici, Monica Patrascu, and Ioan Dumitrache

Department of Automatic Control and Systems Engineering, University Politehnica of Bucharest, 060042 Bucharest, Romania

Correspondence should be addressed to Adrian-Bogdan Hanchevici, adrian.hanchevici@upb.ro

Received 11 April 2012; Accepted 30 April 2012

Academic Editor: Salvatore Sessa

We propose one networked control strategy for linear SISO systems affected by variant communication delays. The purpose of this approach is to adjust, by using the fuzzy logic, the command provided by the PID controller. The input for the fuzzy logic controller (FLC) is represented by the delay and the variation of delay, and the output is used to adapt the PID controller's command to the new value of the communication delay which occurs in the network. Simulation and implementation results are presented. The results show performance improvement when using our control strategy. The authors present the implementation for an axis of a 3D crane, using the Ethernet/IP (TCP/IP) communication protocol.

1. Introduction

Networked Control Systems (NCSs) are a new type of feedback control systems, where the information from the sensors about the controlled parameters and the commands for the actuators are exchanged through a communication network like Ethernet, CAN, and wireless. In Figure 1 is presented the general architecture of a NCS [1].

In Figure 1, S1 to Sn represent the sensor nodes, A1 to An are the actuator nodes, and C1 to Cn are the controller nodes. There is no compulsory to be as many controllers as actuators or sensors. The main problems of this networked control systems are the variant communication delays and package lose (information lose).

In [2, 3] are presented methods to reduce the traffic through the network and so it drops the possibility to appear communication delays. For this purpose, event-based control is used, which means that the control signals are not modified until a condition is met.

According to the communication delay must be used the proper control strategy to ensure the desired QoC (quality of control). Moreover, the parameters of the network can be adapted (QoS: quality of service) by analyzing the performances of all closed-loop systems that are sharing the network [1].

In [4] are presented the challenges and opportunities in Wireless Sensor/Actuator Networks (WSAN). One main issue described is that the WSAN has to support the QoS necessary for the applications which are using the WSAN.

When we want to control a process with variant parameters, we can use the multimodel control strategy. This solution has the advantage of breaking a difficult problem into several easy problems to solve [5–8].

The adaptive control systems represent another strategy that is used when the process has structural and parametric modifications. In this case, the controllers have the capability to identify the modifications of the process and to tune their parameters, accordingly [6, 9–15].

In [16, 17] were used the genetic algorithms for online tuning of the PID controller for linear SISO systems with random communication delay. The communication delay was included in the process model.

Recently were developed new concepts which are using a combination of multimodel and adaptive control. Viable solutions were proposed for the online control of complex processes characterized by nonlinear models [18, 19].

The paper is organized as follows. In Section 2, we start by presenting the proposed control strategy. Section 3 describes the case study which includes the presentation of the experimental setup and the mathematical model of the

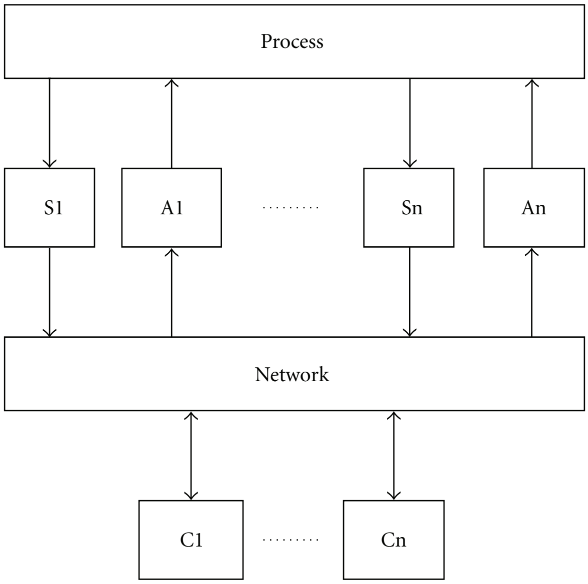


Figure 1: The architecture of a networked control system.

process, and also the simulation and experimental results. Finally, Section 4 contains the conclusions and the directions for future research.

2. Proposed Control Strategy

2.1. Networked Control Systems. A network of sensors and actuators is a group of sensors and actuators that are geographically distributed and interconnected. The sensors gather information about the process and the actuators react to this information by performing the appropriate actions. These networks allow such a management system to monitor and manipulate the behavior of a process. The quality of communication between nodes (Quality of Service (QoS)) plays an important role in the network.

Networks of sensors are used to gather information about a process and then submit those to interested users. Sensors cannot interfere with the process and in most applications it is not enough just to make the monitorization of parameters, also it is needed to perform actions on the process. This need for action makes necessary the coexistence in the network of sensors and actuators. The quality of communication between the nodes of the network must be provided to ensure user satisfaction with system services.

From the end user point of view, applications that use networks have their own specific requirements on the quality of communication offered by the network infrastructure. In some cases, the information has to be exchanged vary fast between nodes and must be complete.

Depending on the type of target application, quality of communication within the network can be characterized by robustness, availability, security, and so forth. Some parameters can be used to measure the satisfaction level of services, such as bandwidth, delay, and loss of information.

Bandwidth is the amount of information carried within a period of time. In general, as long as the bandwidth of the network is greater, the system performance is better. Those nodes that frequently generate large amounts of data, such as video cameras, often require high bandwidth.

Delay is the time required for a packet to get from sender to recipient. Applications that are sensitive to network delays are those that require delivering data packets in real time. Real time does not mean fast computation or high speed communication. A real time communication system is unique because it must work with speed to enable the synchronization requirements. Packet loss rate is the percentage of packets that are lost in the transmission process. It can be used to represent the probability to lose the packets. A package may be lost due to congestion and less successful connection.

These networks are not simple sensor network because of coexistence of sensors and actuators. Sensor nodes generally have low cost, low power consumption, and small size and are equipped with limited processing module and communication (for instance, enables communication with other nodes at low speeds). Due to the limitation of communication capacity, the bandwidth is limited. In particular, energy conservation is very important to extend the operation of the network, because it is not desirable to change the batteries of a sensor when it is operational. Nodes with actuators generally have a greater capacity for processing and communication. Also it is allocated more energy.

In the presence of resource constraints, network communication quality may suffer due to lack of computing resources and/or communication. For example, a number of nodes which want to transmit messages via the same network must compete for limited bandwidth that the network is able to provide. As a consequence, some data will suffer significant delays, resulting in a low quality of communication. Because of limited memory, some data packets may be lost before the node can send them to the destination. It is very important that the given limited resources available are to be used effectively.

Sensors and actuators do not have the same resource constraints. Because they were designed by using different technologies and with different purposes, they are not similar in several aspects, such as processing capacities, communication, and functionality. In a large-scale system, hardware and network technologies used in the network may be different from one subsystem to another. This is true because there are very few standard sensors and dedicated networks and actuators such of products in the market, often have different features. This heterogeneity of the network makes use of high level of available resources to be very difficult. Consequently, efficient use of resources cannot be maximized in most situations.

2.2. Hybrid PID-Fuzzy Control. The control strategy presented in this paper is proposed for linear SISO systems which have variant communication delays because of the communication network.

The control strategy is based on fuzzy logic. The fuzzy logic controller is used to adjust the command provided by the PID controller.

In Figure 2 is presented the structure of the control strategy using the fuzzy logic, where: y_k represents the controlled variable, r_k is the setpoint for the controlled variable, u_k is the control signal, $\hat{\tau}$ represents the estimated delay, and $\hat{\dot{\tau}}$ is the change in delay. The delay estimator module performs the estimation of the delay and change in delay. On every sampling time is performed the estimation of the current communication delay and the adjustment the PID controller's command (u_k^0) according to the fuzzy logic (Δu_k).

In order to determine the current communication delay, the controller requests data from the S (sensor) node. The controller knows that when there is no communication delay, the sensor node will provide the requested data after a certain period of time, t_1. When there is the communication delay, the sensor node will provide the requested data in a higher period of time, t_1', where $t_1' = t_1 + \tau'$. We have considered the case when the sensor and actuator nodes are in the same geographical area, which means that the communication delay between the sensor node and controller is the same with the communication delay between the controller and the actuator node. After all this considerations, the communication delay of the closed-loop system on a moment of time is $\tau = \tau' + \tau'$ and we have included this in the process model. We have analyzed the delay introduced by command computation and after comparing it with the communication delay, we have decided that it can be neglected.

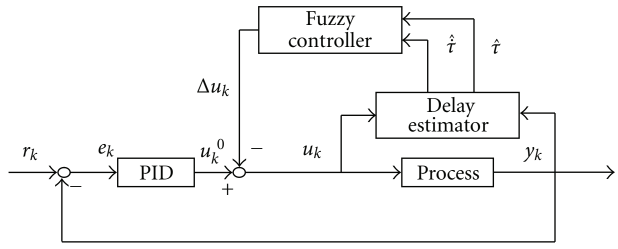


Figure 2: The control strategy.

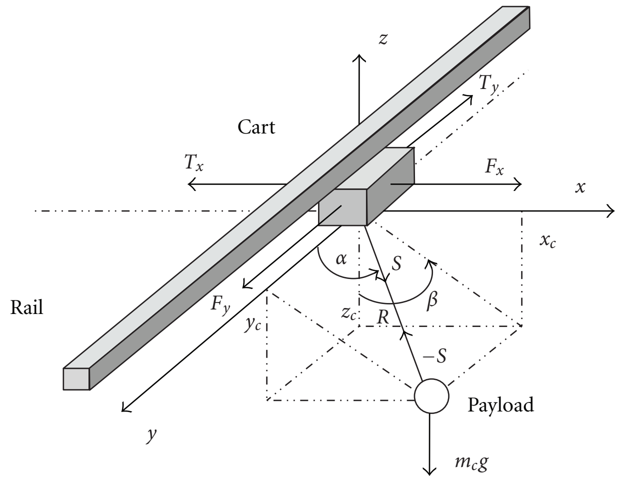


Figure 3: The crane system [20].

3. Case Study

3.1. Experimental Setup and Crane Model. The nonlinear SISO system taken into study in this paper is the Oy axis of a 3-dimensional (3D) Crane. This electromechanical system has a complex dynamic behavior, being controlled from a process computer. The 3D Crane setup consists of a payload (lifted and lowered in the Oz direction by a motor mounted on a cart) hanging on a pendulum-like liftline. The cart is mounted on a rail, giving the system capability of horizontal motion in the Ox and Oy directions. Thus, the payload can move in three dimensions. As actuators, the system makes use of three RH158.24.75 DC motors. Data acquisition is performed by means of five encoders that measure five process variables: the three coordinated of the payload in space and two deviation angles of the liftline, with a resolution of 4096 pulses per rotation (ppr) for the spatial coordinates and 0.0015 rad for the deviation angles. A breakout box contains an interface module that amplifies the control signals which are transmitted from the PC to the DC motors [20].

For our application (3D Crane), the data acquisition is performed by five sensor nodes. The actuation is performed by three actuator nodes (DC motors).

On each sensor and actuator node there is an ATMEL microcontroller, namely ATMEGA32, an ENC28J60 module, and a clock signal generator. The microcontrollers from actuation nodes are used to convert the controller outputs into PWM signals used to command the motors [21]. The microcontrollers from sensor nodes count the rising and falling edges of the signals provided by the encoders and send the values from the counters in digital format to the process computer. The ENC28J60 module is used for the communication between the process computer and the sensor nodes, also between the process computer and the actuation nodes. The clock signal generator sets the working frequency of the microcontroller at the value of 16 MHz.

The 3D Crane is a MIMO system, having three input variables (control signals for the three DC motors) and five output variables (encoder signals). The nonlinear characteristic is given by the pendulum-like motion of the payload in space, with a variable line length. Figure 3 (INTECO (2000)) presents the crane system, where: x_w (not represented) is the distance of the rail with the cart from the center of the construction frame; y_w (not represented) is the distance of the cart from the center of the rail; R is the length of the liftline; α represents the angle between the Oy axis and the liftline; β represents the angle between the negative direction on the Oz axis and the projection of the liftline onto the Oxz plane; m_c is the mass of the payload; m_w is the mass of the cart; m_s is the mass of the moving rail; x_c, y_c, z_c are the coordinates of the payload; $S = F_R - T_R$ represents the reaction force in the liftline acting on the cart; F_x is the force driving the rail with cart; F_y is the force driving the cart along the rail; F_R is the force controlling the length of the liftline; T_x, T_y, T_R are friction forces.

If we define the state variables and the relations of them we get a mathematical model as

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\\ \dot{x}_5\\ \dot{x}_6\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & s_5^2\dot{\beta}^2+\dot{\alpha}^2+Z & 0\end{bmatrix}\cdot\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\end{bmatrix}$$

$$+\begin{bmatrix}0 & 0 & 0\\ 0 & 1 & \mu_1 s_5 s_7\\ 0 & 0 & 0\\ 1 & 0 & \mu_1 c_5\\ 0 & 0 & 0\\ -c_5 & -s_5 c_7 & -\left(1+\mu_1 c_5^2+\mu_1 s_5^2 s_7^2\right)\end{bmatrix}\cdot\begin{bmatrix}N_1\\ N_2\\ N_3\end{bmatrix},$$

$$\begin{bmatrix}x_c\\ y_c\\ z_c\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0 & s_5 s_7 & 0\\ 0 & 0 & 1 & 0 & c_5 & 0\\ 0 & 0 & 0 & 0 & 0 & -s_5 c_7\end{bmatrix}\cdot\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\end{bmatrix}, \tag{1}$$

where the following notations have been used: $x_1 = x_w$; $x_2 = \dot{x}_1$; $x_3 = y_w$; $x_4 = \dot{x}_3$; $x_5 = R$; $x_6 = \dot{x}_5$; $\mu_1 = m_c/m_w$; $\mu_2 = m_c/(m_w+m_s)$; $N_1 = (F_y - T_y)/m_w$; $N_2 = (F_x - T_x)/(m_w+m_c)$; $N_3 = (F_R - T_R)/m_c$; $s_5 = \sin\alpha$; $c_5 = \cos\alpha$; $s_7 = \sin\beta$; $c_7 = \cos\beta$; g is the gravitational acceleration; $Z = (g s_5 c_7)/R$ is a nonlinear function.

A simulation model of the system was implemented. Details of this model can be found in [20].

3.2. Simulation and Experimental Study. The nonlinear SISO system taken into study in this paper can be approximated by the rational s-transfer function:

$$H_P(s) = \frac{B(s)}{A(s)} = \frac{K_P}{s(T_P\cdot s+1)}, \tag{2}$$

where K_P represents the gain and T_P describes the dynamics of the process. The communication delay is included in the process model and $H_P(s)$ becomes

$$H_P'(s) = \frac{B(s)}{A(s)}e^{-\tau\cdot s} = \frac{K_P}{s(T_P\cdot s+1)}e^{-\tau\cdot s}, \tag{3}$$

where τ is variant.

We have considered the linear approximation of the process, as presented above. This approximation was made experimentally. After making this linearization, we have tuned the PID controller, for the case when there is no communication delay ($\tau = 0$ sec). The tuning was made offline, in a simulated environment.

After the offline tuning was achieved, the PID controller has been tested, in simulation. We have continued and considered the communication delay $\tau = 0.6$ seconds and $\tau = 1.0$ seconds. Below, in Figure 4, are presented the simulated responses of the system using the same PID controller and different communication delays. By analyzing the three responses, we can see that there is a difference between them, in terms of settling time and overshoot, while there is no difference in terms of steady state error.

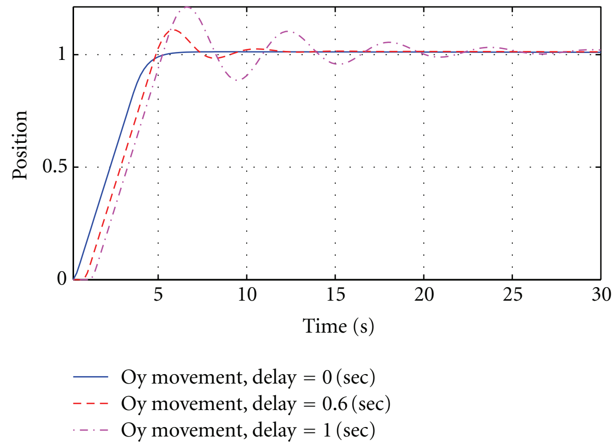


Figure 4: Simulated responses using the same PID controller and different communication delays.

Command u		Delay P0	P1	P2	P3	P4	P5	P6	P7	P8
Change in delay	P4	P6	P7	P8	P8	P8	P8	P8	P8	P8
	P3	P5	P6	P7	P8	P8	P8	P8	P8	P8
	P2	P4	P5	P6	P7	P8	P8	P8	P8	P8
	P1	P3	P4	P5	P6	P7	P8	P8	P8	P8
	Z	P2	P3	P4	P5	P6	P7	P8	P8	P8
	N1	P1	P2	P3	P4	P5	P6	P7	P8	P8
	N2	P0	P1	P2	P3	P4	P5	P6	P7	P8
	N3	P0	P0	P1	P2	P3	P4	P5	P6	P7
	N4	P0	P0	P0	P1	P2	P3	P4	P5	P6

Figure 5: The rule base of the fuzzy logic controller.

This mismatch is generated by the fact that when the communication delay appears, the PID controller designed for the case when there is no communication delay is not able to perform as requested and the closed-loop system's performances are getting worse as the communication delay is getting higher values.

In order to prevent the decrease of performances of the closed-loop system, we have adopted the control strategy presented in Figure 2.

The rule base of the fuzzy logic controller used in simulation and experimental study is presented in Figure 5.

For the delay and command are used the P0–P8 (positive) membership functions, and for the change in delay are used the N4, N3, N2, N1 (negative), Z (zero), P1, P2, P3, and P4 membership functions.

For our study, we have considered that the highest bound for the delay is 2 seconds.

The system works with a sampling time of 0.2 seconds.

Table 1: Simulation study analysis.

	Settling time(s)	Overshoot (%)	Steady state error (%)
	Oy	*Oy*	*Oy*
PID (fixed delay)			
$\tau = 0.0$ sec (S)	4.4	1.3	0
$\tau = 0.4$ sec (S)	6.2	6.2	0
$\tau = 0.6$ sec (S)	7.0	11.0	0
$\tau = 1.0$ sec (S)	18.6	21.3	0
$\tau = 1.4$ sec (S)		unstable	
$\tau = 1.8$ sec (S)		unstable	
$\tau = 2.0$ sec (S)		unstable	
PID and fuzzy (fixed delay)			
$\tau = 0.0$ sec (S)	4.8	1.3	0
$\tau = 0.4$ sec (S)	4.8	1.9	0
$\tau = 0.6$ sec (S)	5.0	3.7	0
$\tau = 1.0$ sec (S)	8.6	8.6	0
$\tau = 1.4$ sec (S)	10.2	12.4	0
$\tau = 1.8$ sec (S)	12.0	12.2	0
$\tau = 2.0$ sec (S)	13.2	9.3	0
PID (variant delay)			
S1 ($\tau_{max} = 2.0$ sec)	24.2	23.8	0
S2 ($\tau_{max} = 2.0$ sec)	31.2	26.8	0
S3 ($\tau_{max} = 2.0$ sec)	8.8	17.1	0
S4 ($\tau_{max} = 2.0$ sec)	16.0	27.2	0
S5 ($\tau_{max} = 2.0$ sec)	10.0	42.9	0
S6 ($\tau_{max} = 2.0$ sec)	9.2	25.3	0
S7 ($\tau_{max} = 2.0$ sec)	14.2	41.0	0
PID and fuzzy (variant delay)			
S1 ($\tau_{max} = 2.0$ sec)	10.8	16.5	0
S2 ($\tau_{max} = 2.0$ sec)	9.8	9.0	0
S3 ($\tau_{max} = 2.0$ sec)	9.4	7.4	0
S4 ($\tau_{max} = 2.0$ sec)	9.2	12.3	0
S5 ($\tau_{max} = 2.0$ sec)	11.2	24.7	0
S6 ($\tau_{max} = 2.0$ sec)	6.8	2.5	0
S7 ($\tau_{max} = 2.0$ sec)	9.4	9.9	0

Table 2: Experimental study analysis.

	Settling time(s)	Overshoot (%)	Steady state error (%)
	Oy	*Oy*	*Oy*
PID (fixed delay)			
$\tau = 0.0$ sec (E)	3.8	0.0	0
$\tau = 0.4$ sec (E)	3.2	3.9	0
$\tau = 0.6$ sec (E)	5.0	10.2	0
$\tau = 1.0$ sec (E)	11.6	31.5	0
$\tau = 1.4$ sec (E)	44.0	49.3	0
$\tau = 1.8$ sec (E)		unstable	
$\tau = 2.0$ sec (E)		unstable	
PID and fuzzy (fixed delay)			
$\tau = 0.0$ sec (E)	4.4	0.0	0
$\tau = 0.4$ sec (E)	4.0	0.0	0
$\tau = 0.6$ sec (E)	4.0	0.1	0
$\tau = 1.0$ sec (E)	6.4	6.2	0
$\tau = 1.4$ sec (E)	7.8	7.5	0
$\tau = 1.8$ sec (E)	6.8	0.0	0
$\tau = 2.0$ sec (E)	7.0	0.0	0
PID (variant delay)			
E1 ($\tau_{max} = 2.0$ sec)	3.2	3.1	0
E2 ($\tau_{max} = 2.0$ sec)	16.8	30.9	0
E3 ($\tau_{max} = 2.0$ sec)	13.2	34.5	0
E4 ($\tau_{max} = 2.0$ sec)	14.8	27.5	0
E5 ($\tau_{max} = 2.0$ sec)	9.6	24.6	0
E6 ($\tau_{max} = 2.0$ sec)	9.0	36.3	0
E7 ($\tau_{max} = 2.0$ sec)	11.4	42.8	0
PID and fuzzy (variant delay)			
E1 ($\tau_{max} = 2.0$ sec)	5.4	1.9	0
E2 ($\tau_{max} = 2.0$ sec)	8.4	14.2	0
E3 ($\tau_{max} = 2.0$ sec)	8.2	17.6	0
E4 ($\tau_{max} = 2.0$ sec)	8.2	8.6	0
E5 ($\tau_{max} = 2.0$ sec)	4.6	2.1	0
E6 ($\tau_{max} = 2.0$ sec)	6.2	0.0	0
E7 ($\tau_{max} = 2.0$ sec)	4.2	2.3	0

The process computer used has the following hardware configuration: Intel(R) Core(TM)2 Duo CPU E7200 @ 2.53 GHz 2.53 GHz, 2.00 GB of RAM. As an operating system, it was used Microsoft Windows XP Professional Version 2002 Service Pack 3. Also, we have used the Matlab.

In Tables 1 and 2 are presented the cases analyzed in this paper.

(i) PID (fixed delay) S, E represent the performances of the system obtained in simulation and experimentally after the tuning of the PID controller was achieved and when were considered different communication delays.

(ii) PID and fuzzy (fixed delay) S, E represent the performances of the system obtained in simulation and experimentally when is used the proposed control strategy and when were considered different communication delays.

(iii) PID (variant delay) S, E represent the performances of the system obtained in simulation and experimentally after the tuning of the PID controller was achieved and when the communication delay has variant values between 0 and 2 seconds.

(iv) PID and Fuzzy (variant delay) S, E represent the performances of the system obtained in simulation and experimentally when is used the proposed control strategy and the communication delay has variant values between 0 and 2 seconds.

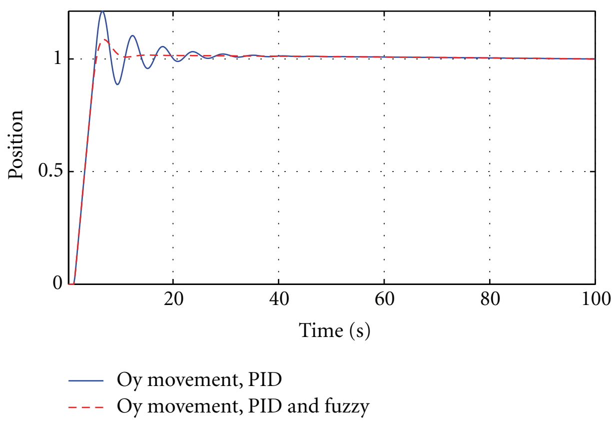


FIGURE 6: Simulated responses for both situations when the delay is fixed (τ = 1.0 sec).

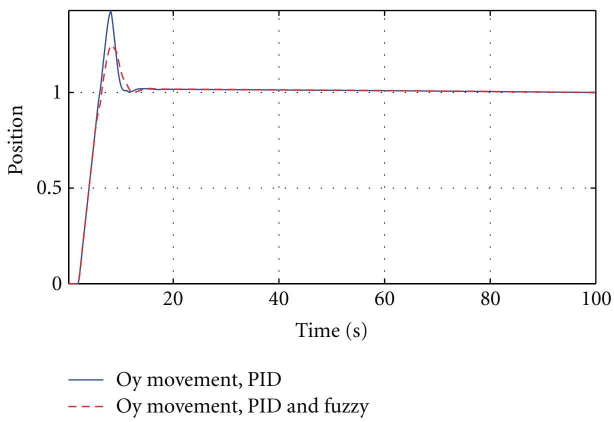


FIGURE 7: Simulated responses for both situations with variant delay (case S5).

In Figures 6 and 7 are described the simulated responses of the system for the case when is used only the PID controller, and for the case when is used the fuzzy logic adjustment. The responses from Figure 6 were obtained for fixed delay ($\tau = 1.0$ sec), and those from Figure 7 for variant delay. The distribution of the variant delay is presented in Figure 8 (case S5).

For the experimental study we have considered the Ethernet/IP (TCP/IP) communication protocol. As in simulation, we have considered the communication delay $\tau = 0.6$ seconds and $\tau = 1.0$ seconds. In Figure 9, are presented the experimental responses of the system using the same PID controller and different communication delays.

As in simulation, also in experimental environment, the PID controller designed for the case when there is no communication delay is not able to perform as requested and the closed-loop system's performances are getting worse as the communication delay is getting higher values. The experimental results are presented in Figure 9.

In Figures 10 and 11 are presented the experimental responses of the system for the case when is used only the PID controller, and for the case when is used the fuzzy logic adjustment. The responses from Figure 10 were obtained for fixed delay ($\tau = 1.0$ sec), and those from Figure 11 for variant

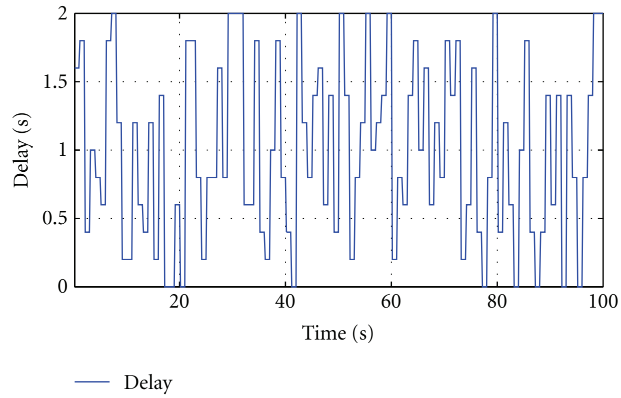


FIGURE 8: The distribution of the delay (case S5).

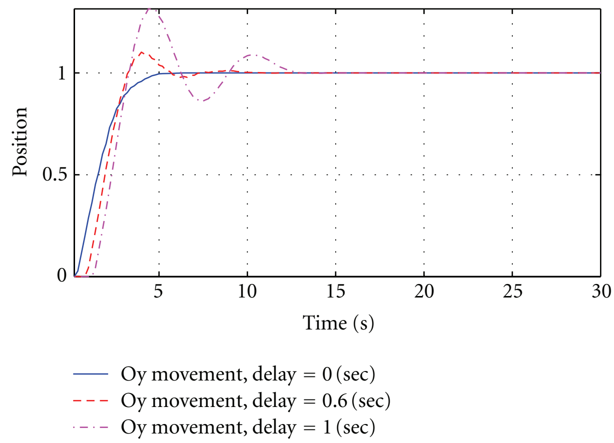


FIGURE 9: Experimental responses using the same PID controller and different communication delays.

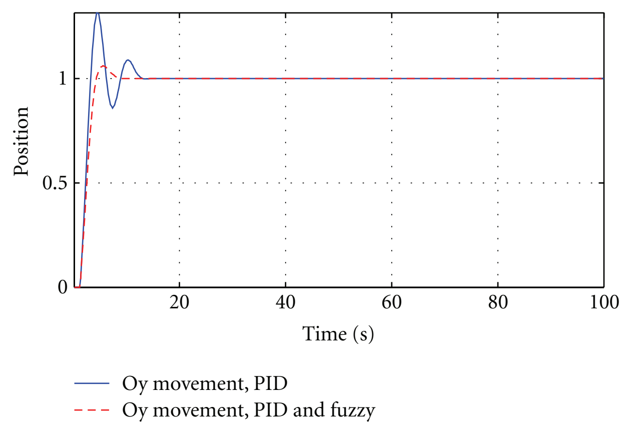


FIGURE 10: Experimental responses for both situations when the delay is fixed (τ = 1.0 sec).

delay. The distribution of the variant delay is presented in Figure 12 (case E7).

By analyzing the responses presented in Figures 4, 5, and 6 and in Figures 9, 10, and 11 we can see that the performances of the closed-loop system are improved when is used our proposed control strategy, in terms of overshoot

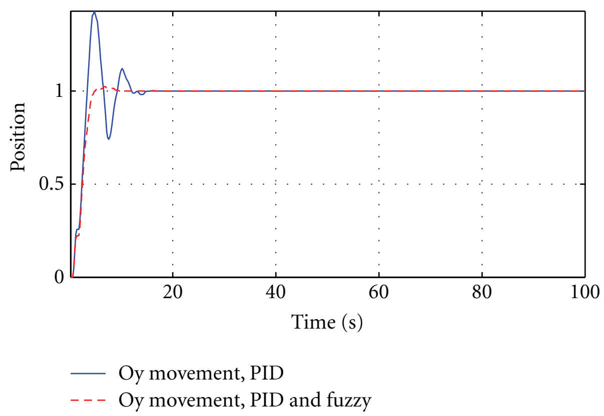


Figure 11: Experimental responses for both situations with variant delay (case E7).

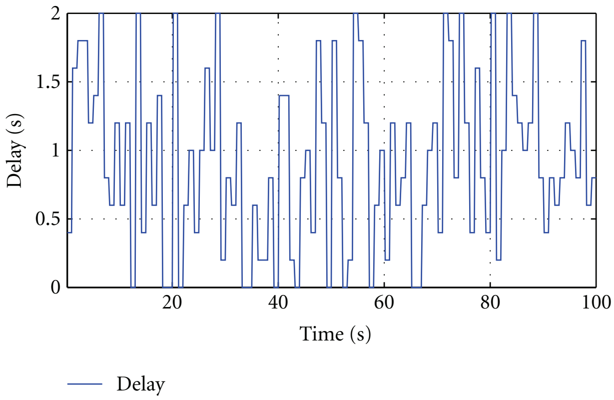


Figure 12: The distribution of the delay (case E7).

and settling time, while there is no difference in terms of steady state error.

4. Conclusions

In this paper, we proposed a networked control strategy for linear SISO systems affected by variant communication delays. By using a fuzzy logic controller, it is adjusted the command provided by the PID controller. As input for the fuzzy logic controller are used the delay and the change in delay. By using the output of the FLC, the PID controller's command is adjusted accordingly to the current value of the communication delay.

By analyzing the simulation and experimental results, we can see that the performances of the closed-loop system are improved when is used our proposed control strategy. For experimental analysis it was used an axis of a 3D Crane and the Ethernet/IP (TCP/IP) communication protocol.

In future work, we will use another controller for nonlinear processes.

Acknowledgments

The work has been funded by the Sectoral Operational Programme Human Resources Development 2007–2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/88/1.5/S/61178. Figure 3 and the nonlinear crane model are presented with the courtesy of INTECO.

References

[1] C. Aubrun, D. Simon, and Y. Song, *Co-design Approaches for Dependable Networked Control Systems*, John Wiley & Sons, New York, NY, USA, 2010.

[2] D. Lehmann and J. Lunze, "Event-based control with communication delays," in *Proceedings of the 18th International Federation of Automatic Control World Congress (IFAC '11)*, pp. 3262–3267, Milano, Italy, 2011.

[3] M. Mazo and P. Tabuada, "Decentralized event-triggered control over wireless sensor/actuator networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2456–2461, 2011.

[4] F. Xia, "QoS challenges and opportunities in wireless sensor/actuator networks," *Sensors*, vol. 8, no. 2, pp. 1099–1110, 2008.

[5] M. Athans, S. Fekri, and A. Pascoal, "Issues on robust adaptive feedback control," in *Proceedings of the 16th Triennial World Congress of International Federation of Automatic Control (IFAC '05)*, pp. 547–577, Prague, Czech Republic, July 2005.

[6] I. Dumitrache, *Automatica. Volumul 1*, Editura Academiei Romane, Bucuresti, Romania, 2010.

[7] M. Patrascu and A. B. Hanchevici, *Sisteme Avansate de Conducere. Indrumar de laborator*, Editura Politehnica Press, Bucuresti, Romania, 2011.

[8] R. Toscano, "Robust synthesis of a PID controller by uncertain multimodel approach," *Information Sciences*, vol. 177, no. 6, pp. 1441–1451, 2007.

[9] K. J. Aström and B. Wittenmark, *Adaptive Control*, Addison-Wesley, Reading, Mass, USA, 1989.

[10] I. Dumitrache, *Ingineria Reglării Automate. Volumul 1*, Editura Politehnica Press, Bucuresti, Romania, 2010.

[11] I. Dumitrache, *Ingineria Reglării Automate. Volumul 2*, Editura Politehnica Press, Bucuresti, Romania, 2010.

[12] G. Tao, *Adaptive Control Design and Analysis*, Wiley-Interscience, John Wiley & Sons, 2003.

[13] R. Iserman, *Digital Control Systems. Volume 1I*, Springer, Berli, Germany, 1991.

[14] J. J. E. Slotine, *Nonlinear Control Systems*, Prentice Hall, 1993.

[15] L. V. Thomas and J. W. Grantham, *Nonlinear and Optimal Control Systems*, John Wiley & Sons, 2000.

[16] A. B. Hanchevici and I. Dumitrache, "Online tuning of PID controller for linear SISO system with random communication delay by using genetic algorithms," in *Proceedings of the IFAC Conference on Advances in PID Control*, Brescia, Italy, 2012.

[17] A.B. Hanchevici and I. Dumitrache, "Intelligent PID control for linear SISO system with random communication delay by using online genetic algorithms," in *Proceedings of the 11th IFAC/IEEE International Conference on Programmable Devices and Embedded Systems*, Brno, Czech Republic, 2012.

[18] S. Fekri, M. Athans, and A. Pascoal, "Issues, progress and new results in robust adaptive control," *International Journal of Adaptive Control and Signal Processing*, vol. 20, no. 10, pp. 519–579, 2006.

[19] S. Fekri, M. Athans, and A. Pascoal, "Robust multiple model adaptive control (RMMAC): a case study," *International Journal of Adaptive Control and Signal Processing*, vol. 21, no. 1, pp. 1–30, 2007.

[20] InTeCo, *3D Crane*, InTeCo, Krakow, Poland, 2000.
[21] S. F. Barrett and D. J. Pack, *Atmel AVR Microcontroller Primer: Programming and Interfacing*, Morgan & Claypool Publishers, Seattle, Wash, USA, 2008.