SPEED: a Simple Parameterized Environment for Evolutionary Defuzzification

A. CINCOTTI, V. CUTELLO, G. SORACE
Department of Mathematics and Computer Science
University of Catania
V.le A. Doria 6, 95125 Catania
ITALY
{cincotti, cutello, sorace}@dmi.unict.it

Abstract: - In this paper, we present a parameterized defuzzification method (PDM) obtained as a linear combination of some of the most well known defuzzification methods. As a case study, we will test the behaviour of the proposed PDM on a simple fuzzy inference system where a good performance can only be obtained using a multi-criteria defuzzification process.

Key-Words: Minimum Hitting Set, Minimum Set Cover, Genetic Algoritms, Approximation Algorithms, NP-complete Problems.

1 Introduction

A fuzzy inference system is a computing framework based on fuzzy if-then rules and fuzzy reasoning [5]. It has found successful applications in a wide variety of fields, such as automatic control, decision analysis, expert system and robotics. The basic structure of a fuzzy inference system is built upon three major components:

- a rule base, which contains a selection of fuzzy rules;
- a database, which defines the membership functions used in the fuzzy rules;
- a reasoning mechanism, which performs the logical inferences based upon the rules and some given facts, and it derives a reasonable output or conclusion.

A fuzzy inference system can take either fuzzy input [11] or crisp inputs (which are viewed as fuzzy singletons), and it produces a fuzzy output. Whenever it is necessary to have a crisp output, especially in a situation where a fuzzy inference system is used as a controller, we need a defuzzification method to extract a crisp value that best represents the fuzzy output. The choice of the defuzzification strategy, therefore, can directly affect the success of the overall fuzzy inference system.

In the next section, we will describe five common defuzzification methods and define a new one, called PDM (Parameterized Defuzzification Method), obtained as a linear combination of the previous methods.

2 Defuzzification

Let A be a fuzzy set defined over the universe $\mathcal{U}$. Let μ be A's fuzzy membership function. The following defuzzification methods are very well known in literature

- Centroid of area

$$COA = \frac{\int_{\mathcal{U}} \mu(z) z dz}{\int_{\mathcal{U}} \mu(z) dz}$$

- Bisector of area

$$\int_{\alpha}^{BOA} \mu(z) dz = \int_{BOA}^{\beta} \mu(z) dz$$

where $\alpha = \min\{z|z \in Z\}$ and $\beta = \max\{z|z \in Z\}$. That is, the vertical line $z = BOA$ partitions the regions between $z = \alpha$, $z = \beta$, $y = 0$, $y = \mu(z)$ into two regions with the same area.

- Mean of maximum is the average of the maximizing z at which μ reacheas a maximum value y^*. Formally,

$$MOM = \frac{\int_{\mathcal{V}} z dz}{\int_{\mathcal{V}} dz} \tag{1}$$

  where $V = \{z|\mu(z) = y^*\}$.

  In particular, if μ has a single maximum at $z = z^*$, then $MOM = z^*$.

- Smallest of maximum (SOM) if the minimum of the maximizing z.

- Largest of maximum (LOM) is the maximum of the maximizing z.

Among the existing strategies neither the COA nor the others methods emerge as the best defuzzification strategy. We will try to come up now with a strategy that combines the five methods above defined.

The parameterized defuzzification method (PDM) is a simple convex linear combination of the previous above and it is defined as:

$$\begin{aligned} PDM &= a_1 COA + a_2 BOA + a_3 MOM + \\ &\quad + a_4 SOM + a_5 LOM \end{aligned}$$

where $\sum_{i=1}^{5} a_i = 1$. As for any other linear aggregation operator, the main problem is to decide the values that the parameters a_i must have (given a specific problem at hand).

Alternative approaches can be found in [6, 10].

3 Genetic algorithms

Choosing the best values for the parameters is very much like an optimization problem. We think, therefore, that it might be worthwhile exploring an evolutionary approach [3, 4, 8]. In the following we will define the basic elements of the genetic algorithm we used.

3.1 What is a chromosome

A chromosome is an ordered list of 5 elements. The first element represent the weight for the COA, the second element the weight for the BOA and so on. How many bits to use for this representation, i.e. what level of precision we want depends upon the fitness function we will be using.

3.2 Fitness

The fitness value of a chromosome depends upon the particularly fuzzy system we are using. In general, a chromosome represent a specific defuzzification method, so its fitness value is proportional to the performance of the fuzzy system when that specific defuzzification strategy is used.

4 Case study: SPEED

As a simple example, let us describe SPEED. In this example, the fuzzy system controls a vehicle running over a circular speedway where there are several types of obstacles that the vehicle must avoid if it does not want to crash. Three different difficulty levels are defined:

- **Beginner level.** The speedway is without obstacles and the vehicle control system must only be careful enough to stay within the speedway borders.

- **Intermediate level.** In this case we have a certain number of fixed obstacles that the vehicle control system must obviously avoid.

- **Expert level.** In this level we have a certain number of obstacle that appear instantaneously in front of the vehicle. This way we want to simulate the concept of unforeseeable events typical of the real life.

To measure the performance of our fuzzy system we make a simple test. Our vehicle must go through the speedway for a fixed number of times and during the test we also consider the concept of stability that now we define.

At every instant, the vehicle control system must decide whether to go straight, turn left or turn right so we define stability as a numeric value that we increase every time the vehicle goes straight and decrease every time it turns left or right. We observe that the introduced concept of stability is in contrast with the concept of safe driving, because a driver that turns at the last moment has a good *stability* but it is more likely that it will have a crash. On the other hand, a careful driver will never have a crash but the defined stability value will not be as good.

We will describe now some of the implementation decisions we made for our case study.

4.1 Implementation details

To simplify the implementation and concentrate on testing the evolutionary approach, we used (semi)triangular membership functions. So, for instance, the fuzzy set OBSTACLE_FAR is defined as:

$$OBSTACLE\_FAR(z) = \frac{z - \min}{\max - \min}. \quad (2)$$

where $\min \leq z \leq \max$ and min and max represent respectively the minimum and maximum values for an obstacle distance. We are also assuming that the speedway road is (ideally) divided into three lanes: internal, middle and external lane. Here we have the five IF_THEN rules of our fuzzy system:

- IF (INTERNAL_LANE)
  THEN TURN_RIGHT
- IF (EXTERNAL_LANE)
  THEN TURN_LEFT
- IF (CENTRAL_LANE AND OBSTACLE_FAR)
  THEN GO_STRAIGHT
- IF (OBSTACLE_TO_THE_RIGHT AND OBSTACLE_NEAR)
  THEN TURN_LEFT
- IF (OBSTACLE_TO_THE_LEFT AND OBSTACLE_NEAR)
  THEN TURN_RIGHT

We use Mamdani fuzzy inference system to infer the output fuzzy set ([7, 12]).

4.2 Some genetic algorithm implementation details

In our evolutionary approach, we used the following genetic parameters:

Number of generation	:	25
Population size	:	40
Crossover	:	dynamic
Mutation	:	0.02
Elitism	:	2
Length of a chromosome	:	$40(5 * 8)$

	COA	BOA	MOM	SOM	LOM
Beg.	2383	-976	-976	-976	-976
Int.	1053*	-794	536	-845	-853
Exp.	1065*	405*	676*	-610	-737

Table 1: Performace of the five defuzzification methods

	COA	BOA	MOM	SOM	LOM
Beg.	0.62	0.07	0.07	0.04	0.20
Int.	0.46	0.12	0.30	0.01	0.11
Exp.	0.32	0.08	0.07	0.07	0.46

Table 2: Weights obtained from evolution

The length of the chromosome is obtained by discretizing with 8 bits the weights. The concept of dynamic crossover was introduced in [2], and it basically means that high fitness chromosomes will use crossover with few points, and low fitness chromosomes will use crossover with many points.

5 Results

In table 1 we show the performance of the five methods of defuzzification. The * denotes a crash. We can observe that the COA is the best in the beginner level but it is unusable in the other levels. In the intermediate level the best defuzzification is MOM but we cannot use it in the expert level. In the last level SOM and LOM are the defuzzification methods that pass the test but their performances are not good.

Table 2 shows the weights obtained from evolution. We can observe that defuzzification methods that have a good performance give a large contribution in the linear combination for PDM. It is interesting to note that the weight for COA decreases as the difficulty level increases.

The performance of our fuzzy system when we use PDM as defuzzification strategy is the following

	PDM
Beginner	2231
Intermediate	1650
Expert	1344

For the beginner level we do not have a better result than COA but for the intermediate and expert levels the result are very good.

6 Conclusion and future works

In this paper we applied the idea of evolutionary defuzzification in the framework of a simple fuzzy inference system. We plan to extend the simple model introduced here in many directions, let us mention a few.

- We intend to change the nature of the membership functions using for example trapezoidal or gaussian and the inference system using for example the Sugeno inference system (see for instance [9]).
- We plan to extend the model by introducing a pre-processing step to learn good membership functions (see for instance [1]).
- We also plan to consider time and energy with stability to define the performance of the system and extend the simulation to a three-dimensional space.
- Finally, we would like to study the performance of PDM when it is defined as a non-linear combination.

References

[1] F. Bergadano and V. Cutello Probably approximately correct learning in fuzzy classification systems IEEE Transaction on Fuzzy System, 3(4):473-478, 1995.

[2] A. Cincotti, V. Cutello, G. Sorace Combining Fuzzy Sets and Genetic Algorithms for improving strategies in Game Playing, in N. Mastorakis Ed., *Problems in Applied Mathematics and Computational Intelligence*, WSES Press, 2001, pp. 149-152.

[3] D.E. Goldberg *Genetic Algorithms in Search Optimization and Machine Learning* Addison Wesley, 1989.

[4] J. H. Holland. Adaption in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor, 1975.

[5] J.-S.R. Jang, C.-T. Sun and E. Mizutani *Neuro-Fuzzy and Soft Computing. A computational approach to learning and Machine Intelligence* Prentice Hall, 1997.

[6] Jiang Tao, Li Yao. Generalized defuzzification strategies and their parameter learning procedures. *IEEE Transactions on fuzzy system.* Vol. 4, no. 1, pp. 64-71 feb 1996.

[7] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1):1-13, 1975.

[8] M. Mitchell *An introduction to Genetic Algotithms* MIT press, 1996.

[9] M. Sugeno. Fuzzy measures and fuzzy integrals: a survey. In M. M. Gupta, G. N. Saridis, and B. R. Gaines, editors, *Fuzzy automata and decision processes*, pages 89-102. North-Holland, New York, 1977.

[10] R. R. Yager, D. P. Filev. SLIDE: A simple adaptive defuzzification method. *IEEE Transactions on fuzzy system.* Vol. 1, no. 1, pp. 69-78 feb 1993.

[11] L. A. Zadeh. Fuzzy sets. *Information and control*, 8:338-353, 1965.

[12] L. A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(1):28-44, January 1973.