Application of EvoNorm in Fuzzy Optimization*

[Extended Abstract] †

Víctor Medrano Z.‡
Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica
San Nicolás de los Garza, N.L., México
vdejesusmedrano@gmail.com

Luis Torres-T.§
Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica
San Nicolás de los Garza, N.L., México
luis.torres.ciidit@gmail.com

ABSTRACT

This paper proposes the use of the EvoNorm algorithm to develop a Genetic Fuzzy System of a robot controlled by light. The rules of the system are changed constantly by the algorithm itself until a number of cycles is satisfied. The final result must be a set of rules that perform a good result for an evaluation function.

Keywords

Fuzzy Logic Controller; Evolutionary Algorithm; Genetic Fuzzy System; EvoNorm; Normal distribution function; Optimization

1. INTRODUCTION

There are two possible ways for integrating fuzzy logic and evolutionary algorithms. The first one involves the application of evolutionary algorithms for solving optimization and search problems related with fuzzy systems, obtaining genetic fuzzy systems. The second one concerns the use of fuzzy tools and fuzzy logic-based techniques for modelling different evolutionary algorithm components and adapting evolutionary algorithm control parameters, with the goal of improving performance. The evolutionary algorithms resulting from this integration are called fuzzy evolutionary algorithms [3].

In this case, we are concerned with genetic fuzzy systems. EvoNorm is the evolutionary algorithm in charge to obtain the optimal rules for the fuzzy system.

2. PRELIMINARY CONCEPTS

2.1 EVOlutionary Algorithm of Random Variables with NORMal Distributions (EvoNorm)

First versions of Evolutionary Algorithms contained the procedures of *crossover* and *mutation* to generate a new population of individuals. Then, appeared new trends suggesting the generation of a new population by means of a model that represents an estimation of distributions, where the model parameters are defined by the selected individuals. Examples of these algorithms comprehend: the Compact Genetic Algorithm, the Bayesian Optimization Algorithm and the Univariate Marginal Distribution Algorithm. These algorithms consider a weak interaction between variables (except for the Bayesian Optimization Algorithm). EvoNorm is inspired in the generation of a new population using estimation of distributions, and has been compared with Evolution Strategies to show that has a better performance. [6]

Being EvoNorm an evolutionary algorithm, it follows the steps of an evaluation process, a selection, and a variation procedure (replacing the *crossover* and *mutation* procedures for *the calculation of the parameters* and *the generation of a new population with the normal distribution function*). The general procedure for the EvoNorm algorithm is shown in the next frame:

General procedure for EvoNorm:

1) Generation of a uniform random population P of size I_p x D_r.
2) Evaluation of the I_p individuals.
3) Deterministic selection of the best I_s individuals ($I_s < I_p$)
4) Calculation of mean and standard deviation from I_s selected individuals.
5) Maintain an equilibrium between exploration and exploitation to get a chance to find a new solution.
6) Generation of a new population of size I_p from random variables with parameters calculated in (4) and (5).
7) If a criterion satisfied then end, else go to step (2).

A population P is a matrix of size I_p (total of individuals) and D_r (total of decision variables). A solution is a set of

*(Produces the permission block, and copyright information). For use with SIG-ALTERNATE.CLS. Supported by ACM.

†A full version of this paper is available as *Author's Guide to Preparing ACM SIG Proceedings Using LaTeX2ε and BibTeX* at `www.acm.org/eaddress.htm`

‡Ing. Victor M.

§The secretary disavows any knowledge of this author's actions.

GECCO'15, July 11-15, 2015, Madrid, Spain.

DOI: 10.475/123_4

decision variables and this set is represented as a real vector. Every row of the population P represents a set of decision variables. The population of selected individuals is a matrix P_s of size I_s (total of individuals selected) and D_r.

The I_p individuals are evaluated through the evaluation function. The total of individuals selected in step (3) is suggested to be in the range of 10-20% of the total population.

In EvoNorm, the new population is generated by a normal distribution function. This function has two parameters: the mean (a measure of the central tendency of the random variable) and the standard deviation (a measure of the dispersion of a variable around the mean). This parameters are calculated with the equations below:

$$\mu_{pr} = \sum_{k=1}^{I_s} \frac{P_{s_{pr,k}}}{I_s} \quad (1)$$

$$\sigma_{pr} = \sqrt{\sum_{k=1}^{I_s} \frac{(P_{s_{pr,k}} - \mu_{pr})^2}{I_s}} \quad (2)$$

where $pr = \{1, 2, ..., D_r\}$.

An heuristic is used to maintain an equilibrium between exploration and exploitation (5), so new solutions can be found not necessarily near of the mean calculated [7]. The best solution found Ix at the moment is involved in the generation so in the 50% percent of the times the mean is used in the calculations and in the other 50% percent of the time the best solution found Ix is used as a mean as is shown in the following equation:

$$P_{i,pr} = \begin{cases} N(\mu_{pr}, \sigma_{pr}) & U() > 0.5 \\ N(Ix_{pr}, \sigma_{pr}) & otherwise \end{cases} \quad (3)$$

2.2 Fuzzy Logic Controller (FLC)

Fuzzy logic was introduced by Zadeh in 1968 and is based on mathematical representation of human knowledge and experiences. FLCs can be considered as knowledge-based systems, incorporating human knowledge into their knowledge base through fuzzy rules and fuzzy membership functions. Fuzzy logic allows the manipulation of linguistic data (Large, Medium, and Small) and inaccurate data [4]. Figure 1 shows the general structure of a FLC.

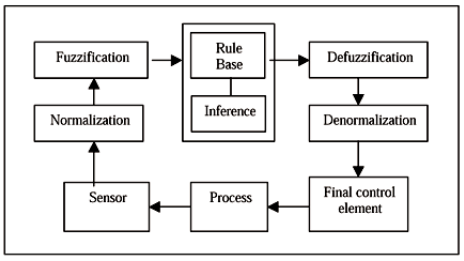

Figure 1: General Structure of a FLC.

The principal functions of a FLC include:

- **Fuzzifier:** It is responsible for assigning a membership value to each fuzzy set, through the characteristic functions associated to the fuzzy sets. The inputs must be normalized values with concrete values and the outputs are the membership values of each fuzzy set.
- **Rule Base:** FLCs use rules, which combine one or many input fuzzy sets (antecedents or premises) and then associate an output fuzzy set (consequent or result). The rules are IF-THEN statements. The input fuzzy sets are associated with fuzzy logic operators (AND,OR,NOT,etc.).
- **Fuzzy Inference:** It is the process of formulating the mapping from a given input to an output using fuzzy logic. The process of fuzzy inference involves membership functions, fuzzy logic operators, and IF-THEN rules.
- **Defuzzifier:** With the output fuzzy set obtained in the inference engine and through mathematical methods of defuzzification is obtained a concrete value as the output. This value is denormalized to be sent to the plant.

Fuzzy logic presents robust and flexible inference methods in problems subject to imprecision and uncertainty. The linguistic representation of knowledge permits a person to interact with a fuzzy system in an easy manner [5].

2.3 Genetic Fuzzy System (GFS)

GFSs have been widely employed to solve classification, regression and control problems. The main feature that highlights GFSs in respect to other mathematical, statistical and artificial intelligence models is its capability of extracting knowledge from datasets or industrial plants and state it in linguistic terms with reasonable accuracy[2].

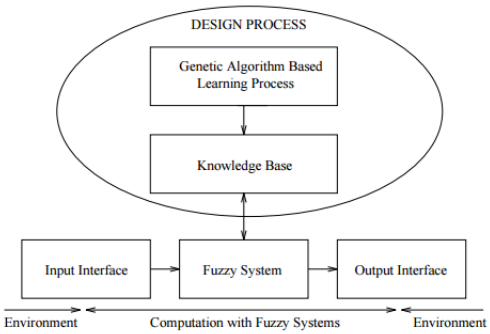

Figure 2: General structure of a GFS

The principles and operations of EvoNorm and FLCs have been briefly described in the previous sections. These two soft computing tools can be combined to form a GFS in which EvoNorm is used to evolve a fuzzy system by tuning the learning fuzzy rules.

3. EXPERIMENTAL RESULTS

3.1 Design of the plant

The hardware of the GFS (illustrated in figure 3) is composed of two inputs and one output. The inputs consist of two LDRs ($Input_1$ and $Input_2$), which have a variable resistance due to light change in the environment. Both LDRs

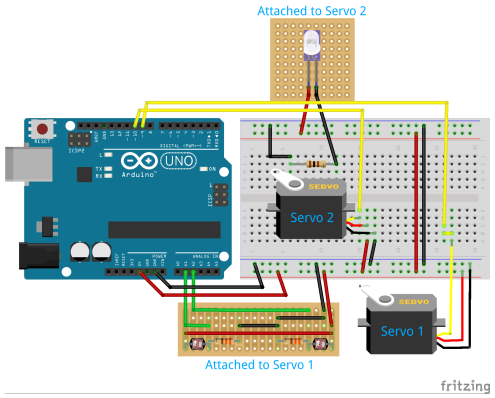

Figure 3: Illustration of the GFS.

are soldered to a board (end to end) attached to *Servo 1*. A board with an ultrabright LED is attached to a *Servo 2* to make this change of light in the environment more variable. Through a voltage divider the analog input signals are sent to the Arduino board. With the generation of a random population P with $I_p = 10$, the consequent of the rules for each combination of input fuzzy sets (D_r) are created. The output of the FLC is the angle of the *Servo 1* in the range of 45-135°, which will depend on the consequents created and the light sensed by the inputs. The consequents are evaluated by EvoNorm for each individual of the population P through the evaluation function. In this experiment, the evaluation function is $error = |Input_2 - Input_1|$. The best individual will be the one in which *error* is closest to zero. Being $I_s = 2$, EvoNorm selects the 2 best individuals and generates a new population through the normal function. The normal function used in the code is based in the Central Limit Theorem Method as stated by Matt Donadio in [1].The EvoNorm algorithm tries to find the optimal solution following the steps (2)-(7) of the frame in section 2.1. For this experiment, the steps (2)-(7) are repeated until 10 generations of new populations have been created.

The schematic diagram of the system is shown in figure 4.

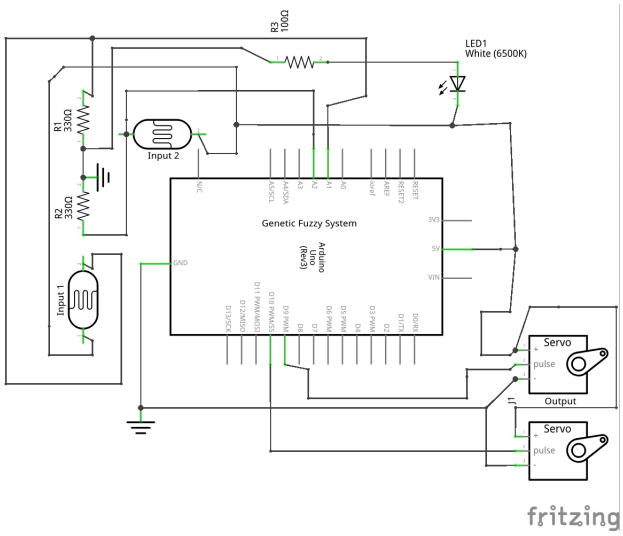

Figure 4: Schematic diagram of the GFS.

3.2 Results

The following results show the evolution of the error of each individual in P through 10 generations. For a better visualization of the results, the evolution of the error is splitted in 2 graphs (Figure 5 and 6).

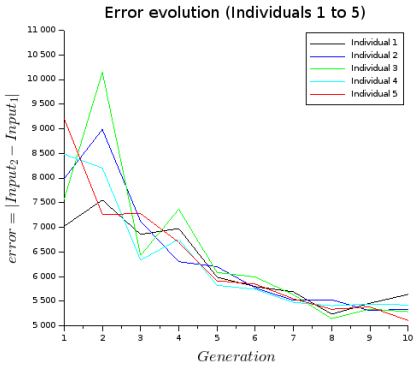

Figure 5: Error evolution (Individuals 1-5).

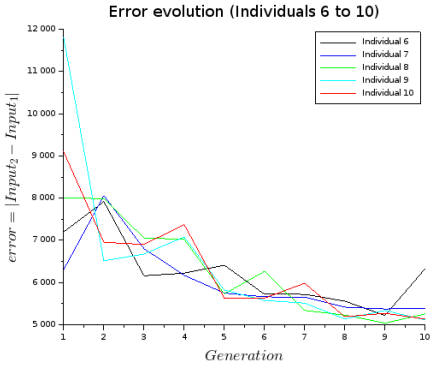

Figure 6: Error evolution (Individuals 6-10).

The next graph (figure 7) shows the mean error of the 10 individual. The mean error tends to decrease in each generation (except for the last generation).

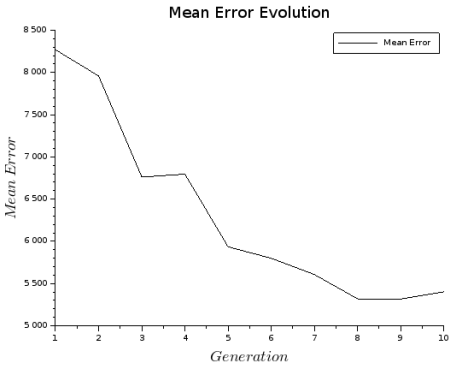

Figure 7: Mean Error Evolution.

4. CONCLUSIONS

The results show that is possible to obtain an optimal solution for a FLC with the help of an evolutionary algorithm as EvoNorm. Physically, the operation of the system has an uncertain behavior and the output varies in some cases abruptly; it is supposed that there could be very similar rules in which the consequent is completely different.The minimum mean error has a value of 5316.1, therefore the next step will be to modify the code embedded into the Arduino to obtain better possible solutions by decreasing error in a more substantial way.

5. REFERENCES

[1] M. Donadio. How to Generate White Gaussian Noise. http://dspguru.com/dsp/howtos/how-to-generate-white-gaussian-noise.

[2] A. Koshiyama et al. Gpfis-class: A genetic fuzzy system based on genetic programming for classification problems. *Applied Soft Computing*, 37:561–571, 2015.

[3] C. L. Mumford and L. C. Jain. *Computational Intelligence: Collaboration, Fusion and Emergence*. Springer, Berlin, 2009.

[4] S. M. Odeh et al. A hybrid fuzzy genetic algorithm for an adaptive traffic signal system. *Advances in Fuzzy Systems*, 2015:1–11, August 2015.

[5] P. M. Pawar and R. Ganguli. *Structural Health Monitoring Using Genetic Fuzzy Systems*. Springer, New York, 2011.

[6] L. Torres-T. Evonorm, a new evolutionary algorithm to continuous optimization. 2006.

[7] L. Torres-T. A mathematical model of a cold rolling mill by symbolic regression alpha-beta. November 2014.