

Parameter adaptation of the Bat Algorithm, using type-1, interval type-2 fuzzy logic and intuitionistic fuzzy logic

Jonathan Pérez¹, Fevrier Valdez¹, Olympia Roeva²
and Oscar Castillo¹

¹ Tijuana Institute of Technology, Calzada Tecnológico s/n, Tijuana, Mexico
e-mails: tecjonathan@gmail.com, {fevrier, ocastillo}@tectijuana.mx

² Institute of Biophysics and Biomedical Engineering, BAS, Sofia, Bulgaria
e-mails: olympia@biomed.bas.bg

Received: 7 February 2016

Revised: 4 March 2016

Accepted: 12 March 2016

Abstract: We describe in this paper the Bat Algorithm (BA) and a proposed enhancement using fuzzy and intuitionistic fuzzy systems to dynamically adapt BA parameters. BA is a metaheuristic algorithm inspired by the behavior of micro bats, which has been applied to different optimization problems obtaining good results. We propose a new method for dynamic parameter adaptation in the BA using Type-1, interval Type-2 fuzzy logic and intuitionistic fuzzy logic. The goal is improving the performance of the BA.

Keywords: Dynamic parameter adaptation, Bat algorithm, Type-1 fuzzy logic, Type-2 fuzzy logic, Intuitionistic fuzzy logic.

AMS Classification: 03E72.

1 Introduction

One of the main challenges of the field of metaheuristic techniques is appropriately varying parameter values during an metaheuristic algorithm run (parameter control). In order to increase the performance of the regarded algorithms it is necessary to provide the adjustments of their parameters depending on the considered problem.

Finding parameters setting is not a trivial task, since their interaction with algorithm performance is a complex relationship and the optimal one are problem-dependent [15]. An optimal or near-optimal set of control parameters for one metaheuristic algorithm does not

generalize to all cases. This stresses the need for efficient techniques that help finding good parameter settings for a given problem, i.e. the need for good parameter adapting methods.

Fuzzy logic has been successfully applied for improving many metaheuristic optimization algorithms [1, 4, 11, 14]. Intuitionistic fuzzy logic (IFL) and Intuitionistic fuzzy sets (IFS) [5-10] have gained recognition as a useful tool for control uncertain phenomena. There are few application of IFL in control of metaheuristic algorithms parameters [15, 16].

This paper is concerned with the application of the Type-1, Interval Type-2 fuzzy logic and IFL to dynamic parameter adaptation of Bat Algorithm (BA). The main idea is changing the parameters of the BA over time in order to improve the algorithm performance.

The BA is a metaheuristic optimization method proposed by Yang in 2010 [18]. This algorithm is based on the behavior of micro bats, which use echolocation pulses with different emission and sound. The BA has the characteristic of being one of the best to face problems of nonlinear global optimization [3, 13, 17, 19]. In this paper the fuzzy and intuitionistic fuzzy systems are presented with the aim of dynamically setting some of the parameters in the BA.

2 Bat algorithm

This section describes the basic concepts of the original BA. The BA is a metaheuristic algorithm that was proposed by Xin-She Yang in 2010 [18]. It is based on the echolocation capability of micro bats guiding them on their foraging behavior.

2.1 Rules of bats

The BA is a novel metaheuristic swarm intelligence optimization method developed for global numerical optimization, in which the search algorithm is inspired by the social behavior of bats and the phenomenon of echolocation to sense distance [3].

If we idealize some of the echolocation characteristics of microbats, we can develop various bat-inspired algorithms or bat algorithms. For simplicity, we now use the following approximate or idealized rules [3, 18]:

1. All bats use echolocation to sense distance, and they also know the difference between food/prey and background barriers in some unknown way.
2. Bats fly randomly with velocity v_i at position x_i with a fixed frequency f_{min} , varying the wavelength λ and the loudness A_0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission $r \in [0, 1]$, depending on the proximity of their target.
3. Although loudness can vary in many ways, we assume that the loudness varies from a large (positive) A_0 to a minimum constant value A_{min} .

For simplicity, the frequency $f \in [0, f_{max}]$, and the new solutions x_i^t and velocity v_i^t at a specific time step t are represented by a random vector drawn from a uniform distribution [13, 19].

2.2 Movements in the BA

Each bat is associated with a velocity v_i^t and location x_i^t , at iteration t , in a dimensional search or solution space. Among all the bats, there exist a current best solution x^* . Therefore, the above three rules can be translated into the updating equations for x_i^t and velocities v_i^t :

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (1)$$

$$v_i^{t+1} = v_i^t + (x_i^t - x^*)f_i, \quad (2)$$

$$x_i^t = x_i^{t-1} + v_i^t, \quad (3)$$

where $\beta \in [0, 1]$ is a random vector selected from a uniform distribution [4], x^* is the current global best solution which is located after comparing all the solutions among all the n bats, f_i is used to adjust the velocity change.

As mentioned earlier, we can either use wavelengths or frequencies for the implementation, and we will use $f_{min} = 0$ and $f_{max} = 1$, depending on the domain size of the problem of interest. Initially, each bat is randomly assigned a frequency which is drawn uniformly from $[f_{min} - f_{max}]$. The loudness and pulse emission rates essentially provide a mechanism for automatic control and auto zooming into the region with promising solutions [4].

For the local search part, once a solution is selected among the current best solutions, a new solution for each bat is generated locally using a random walk

$$X_{new} = X_{old} + \varepsilon A^t, \quad (4)$$

where $\varepsilon \in [-1, 1]$ is a random number, while $A^t = \langle A_i^t \rangle$ is the average loudness of all the bats at this time step.

2.3 Pseudo code for the BA

The basic steps of the bat algorithm can be summarized as the pseudo code shown in Fig. 1 [4].

```

Initialize the bat population  $x_i$  ( $i = 1, 2, \dots, n$ ) and  $v_i$ 
Initialize frequency  $f_i$ , pulse rates  $r_i$  and the loudness  $A_i$ 
while ( $t < \text{Max numbers of iterations}$ )
    Generate new solutions by adjusting frequency
    and updating velocities and locations/solutions [Eqs. (1) to (3)]
    if ( $\text{rand} > r_i$ )
        Select a solution among the best solutions
        Generate a local solution around the selected best solution
    end if
    Generate a new solutions by flying randomly
    if ( $\text{rand} < A_i \ \& \ f(x_i) < f(x^*)$ )
        Accept the new solutions
        Increase  $r_i$  and decrease  $A_i$ 
    end if
    Rank the bats and find the current best  $x^*$ 
end while

```

Figure 1. Pseudocode of Bat Algorithm

2.4 Loudness and pulse rates

In order to provide an effective mechanism to control the exploration and exploitation and switch to the exploitation stage when necessary, we have to vary the loudness A_i and the rate r_i of pulse emission during the iterations. Since the loudness usually decreases once a bat has found its prey, while the rate of pulse emission increases, the loudness can be chosen as any value of convenience, between A_{min} and A_{max} , assuming $A_{min} = 0$ means that a bat has just found the prey and temporarily stop emitting any sound. With these assumptions, we have

$$A_i^{t+1} = \alpha A_i^t, r_i^{t+1} = r_i^0 [1 - \exp(-\gamma^t)], \quad (5)$$

where α and γ are constants. In essence, here α is similar to the cooling factor of a cooling schedule in simulated annealing. For any $0 < \alpha < 1$ and $\gamma > 0$, we have

$$A_i^t \rightarrow 0, r_i^t \rightarrow r_i^0, \text{ as } t \rightarrow \infty. \quad (6)$$

In the simplest case, we can use $\alpha = \gamma$. We have used $\alpha = \gamma = 0.9$ to 0.98 in our simulations [3].

3 Fuzzy logic and Intuitionistic Fuzzy Logic

3.1 Type-1 fuzzy logic systems

A fuzzy logic system (FLS) that is defined entirely in terms of Type-1 fuzzy sets, is known as Type-1 Fuzzy Logic System (Type-1 FLS). Its elements are defined in the following Fig. 2 [12].

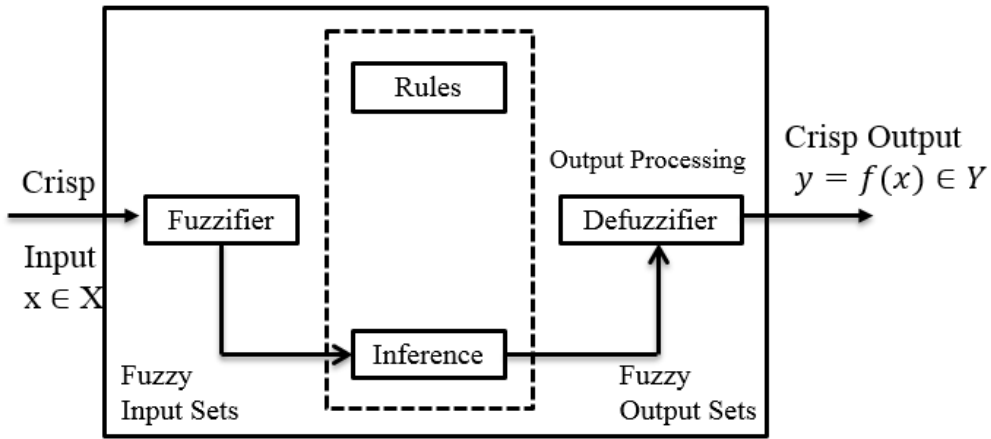


Figure 2. Architecture of a Type-1 Fuzzy Logic System

A fuzzy set in the universe U is characterized by a membership function $u_A(x)$ taking values on the interval $[0, 1]$ and can be represented as a set of ordered pairs of an element and the membership value of the set:

$$A = \{(X, u_A(X)) | X \in U\}. \quad (7)$$

3.2 Type-2 fuzzy logic systems

A variety of types of membership functions exist, but one of them is typically known as the bell of Gauss (Gaussian). The mathematical function is defined with the following equation [18, 19].

$$f(x) = \exp\left(\frac{-0.5(x-c)^2}{\sigma^2}\right). \quad (8)$$

In Type-2 FLS the membership functions can now return a range of values instead of only one number, which vary depending on the uncertainty involved in not only the inputs, but also in the same membership function for a specific value of x' , the membership function (u'), assumes different values, which do not have the same weight, so that it can assign a wide distribution of values at all points and this is called the footprint of uncertainty (FOU) [1, 11].

If we do this for all $x \in X$, we can create a three-dimensional membership function, a Type-2 membership function that characterizes a Type-2 fuzzy set [1, 11].

A Type-2 fuzzy set, \tilde{A} , is characterized by its membership function.

$$\tilde{A} = \{(x, u), u_{\tilde{A}}(x, u) | x \in X, u \in J_x \subseteq [0, 1]\}, \quad (9)$$

where $0 \leq u_{\tilde{A}}(x, u) \leq 1$.

3.3 Footprint uncertainty

Uncertainty affects decisions in a number of different ways. The concept of information is fully connected to the concept of uncertainty. The most fundamental part of this connection is that the uncertainty involved in any solution of a problem is the result of poor information, which may be incomplete, imprecise, fragmentary, not fully reliable, vague, contradictory, or deficient in some way or another [12]. Fig. 3 shows the architecture of a Type-2 FLS.

The output processor includes a type-reducer and defuzzifier. It generates a Type-1 FLS output (from the type-reducer) or a crisp number (from the defuzzifier) [14].

A Type-2 FLS is also characterized by IF-THEN rules, but their fuzzy sets are now Type-2 FLS. The FLS can be used when circumstances are too uncertain to determine exact membership degrees. as is the case when the membership functions in a fuzzy controller can take different values and we want to find the distribution of membership functions to show better results in the stability of fuzzy control [11].

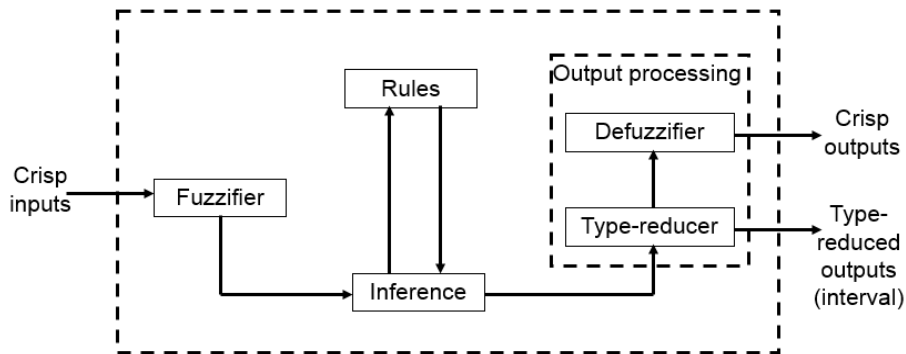


Figure 3. Architecture of a Type-2 fuzzy logic system

3.4 Intuitionistic fuzzy logic

According to Atanassov [5–10], an IFS on the universum $X \neq \emptyset$ is an expression A given by:

$$A = \{\langle x, \mu_A(x), \nu_A(x) \rangle \mid x \in X\}, \quad (10)$$

where the functions

$$\mu_A, \nu_A : X \rightarrow [0, 1] \quad (11)$$

satisfy the condition

$$0 \leq \mu_A(x) + \nu_A(x) \leq 1 \quad (12)$$

and describe, respectively, the degree of the membership $\mu_A(x)$ and the non-membership $\nu_A(x)$ of an element x to A . Let

$$\pi_A(x) = 1 - \mu_A(x) - \nu_A(x), \quad (13)$$

therefore, function π_A determines the degree of uncertainty.

4 Proposed fuzzy adaptation of parameters

In the BA the selected parameters integrated into the Type-1 FLS, Interval Type-2 FLS and IFL system are "Iteration", "Beta" (β) and "Pulse Rate" (r_i). The "Iteration" variable is defined by the Eq. (14), and has a range from 0 to 1. This variable can be seen as the percentage of the current iteration.

$$Iteration = \frac{Current\ Iteration}{Maximum\ of\ Iterations}. \quad (14)$$

The β variable is located between $[0, 1]$ which is increasing with the step iterations and the variable r_i value is between $[0, 1]$, which is decreasing with the step iterations.

4.1 Type-1 and Type-2 FLS

The main difference between a Type-1 FLS and an Interval Type-2 FLS, is that the degree of membership is also fuzzy, and is represented by the FOU. If we shift from Type-1 FLS to Type-2 FLS, theoretically we need a degree of footprint uncertainty, so that this degree was manually modified until the best possible FOU is obtained. The Type-1 FLS for BA parameter adaptation is shown in Fig. 4 and the Type-2 FLS – in Fig. 8.

Fig. 5 shows the rule set from the original Type-1 FLS for parameter adaptation. These rules stay the same in the change from Type-1 to interval Type-2. The set of IF-THEN fuzzy rules are granulated into 3 rules in order to cover all iterations and the search space. To start in low iterations the parameter β is low and the r_i is high. Going to the middle iterations iterations the parameter β is middle and r_i is middle start to cover much of the search space. In the high iterations the same procedure is repeated the parameter β is high and r_i is low. In this way the exploitation of the search space is achieved.

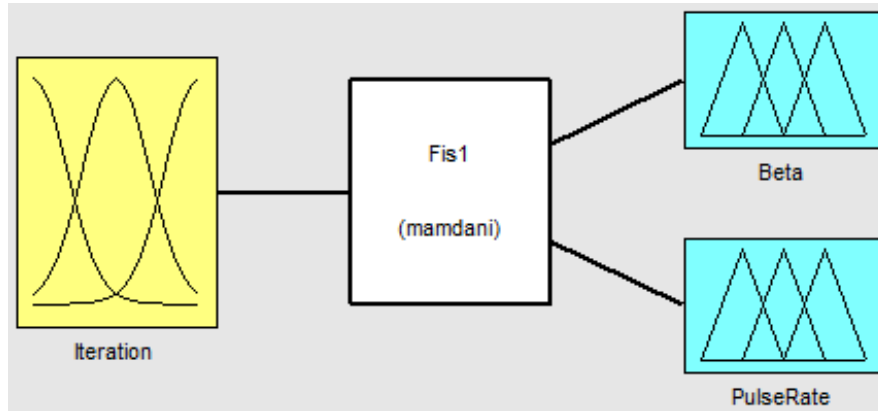


Figure 4. Type-1 FLS for parameter adaptation

The proposed fuzzy *system* is of Mamdani type because it is more commonly used in this type of fuzzy control and the defuzzification method is the centroid. The membership functions are of triangular form in the inputs and the outputs because of their simple definition for this problem.

In the “input1” variable “*Iteration*” the membership functions are of triangular form (Fig. 5).

On the input variable (as mentioned above) triangular membership functions granulated into three fuzzy sets are used, which is performed in this manner considering the reviewed literature and by analyzing the results by others that have been successful [2].

In the output variables, β and the r_i , the literature values between the range of 0 to 1 are recommended for each of the output variables by which the same are designed using this range of values. Each output is granulated into three triangular membership functions (Low, Middle, High), and the design of the output variables can be found in Fig. 6 for β , and in Fig. 7 for r_i .

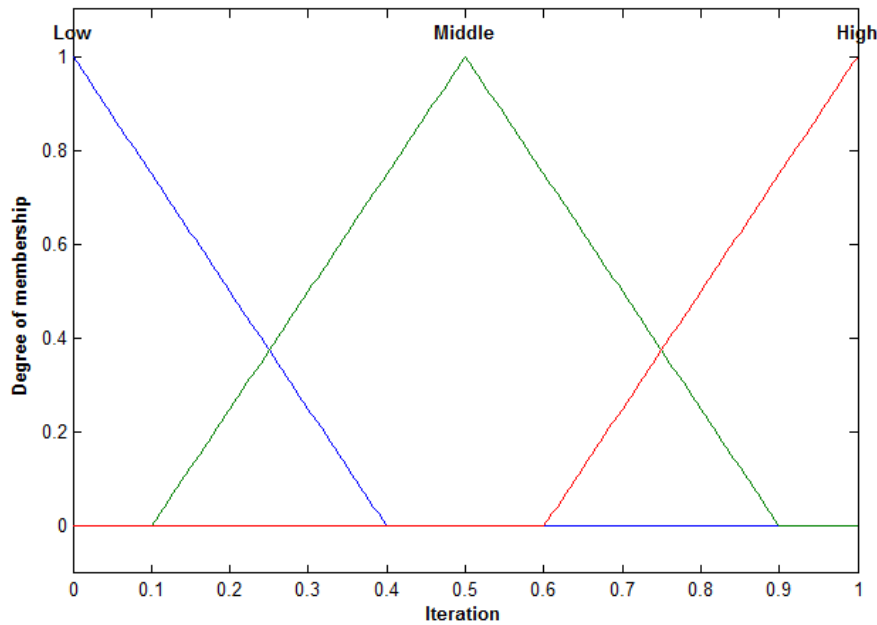


Figure 5. Type-1 FLS for the variable *Iteration*

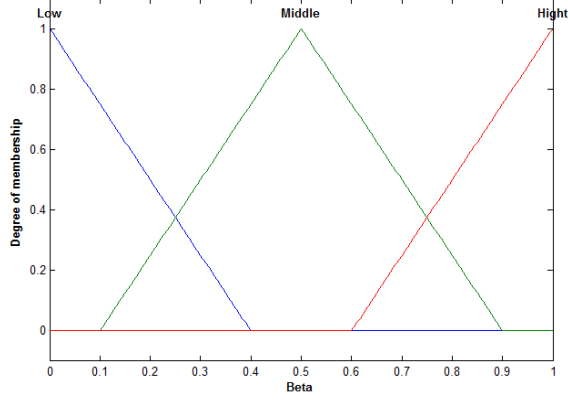


Figure 6. Type-1 FLS for the β parameter

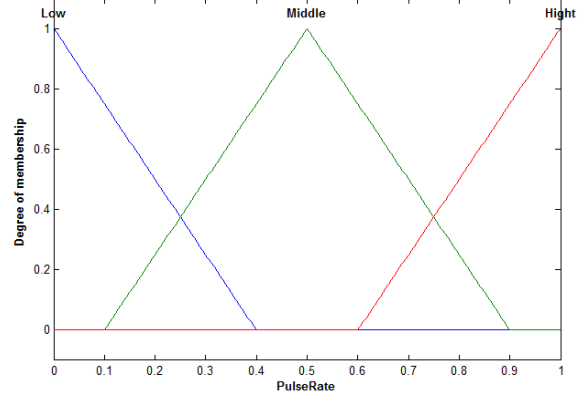


Figure 7. Type-1 FLS for the parameter r_i

The interval Type-2 FLS was designed similarly to the one in [2], for the parameter adaptation and is shown in Fig. 8. We develop this system manually, this is, we change the levels of FOU of each point of each membership function, but each point has the same level of FOU, also the input and output variables have only interval Type-2 triangular membership functions.

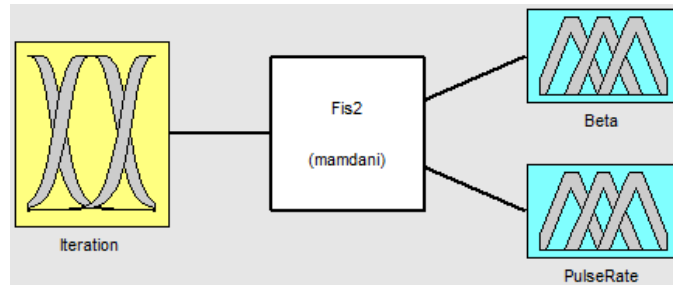


Figure 8. Interval Type-2 FLS for parameter adaptation

Again the membership functions are of trapezoidal form in the inputs and the outputs because of their simple definition for this problem. In the “input1” for the “Iteration” variable the membership functions are of triangular form as shown in Fig. 9.

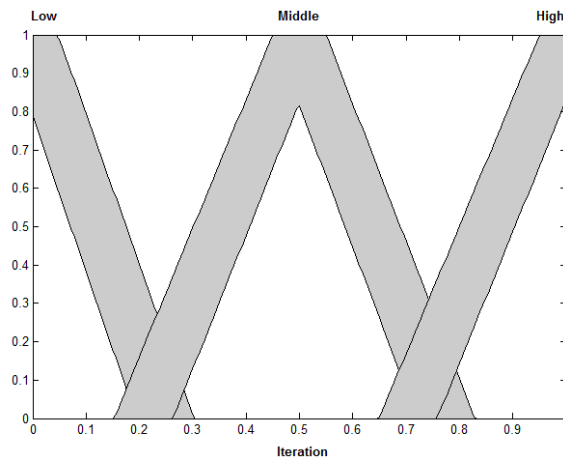


Figure 9. Interval Type-2 FLS of the variable *Iteration*

In the output variables, β and the r_i granulated into three triangular membership functions (Low, Middle, High), and the design of the output variables can be found in Fig. 10 for β , and in Fig. 11 for r_i .

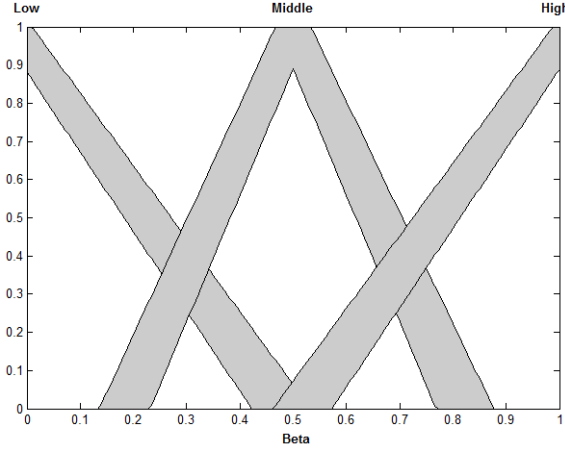


Figure 10. Interval Type-2 FLS of the β parameter

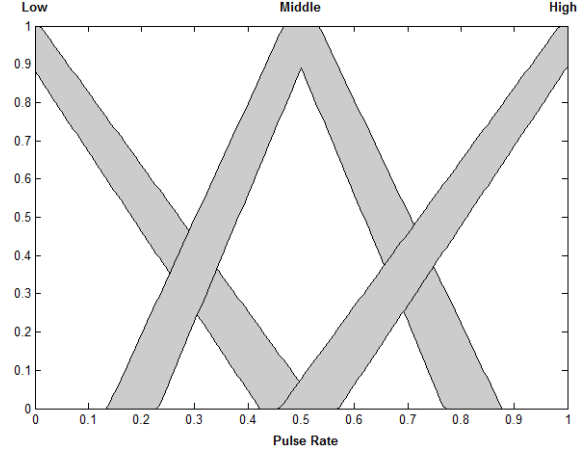


Figure 11. Interval Type-2 FLS of the r_i parameter

The fuzzy rule set used for parameter adaptation in the Bat Algorithm is shown in Fig. 12.

-
- | | | |
|-----------------------------|---------------------------|--------------------|
| 1. If (Iteration is Low) | then (β is Low) | (r_i is High) |
| 2. If (Iteration is Middle) | then (β is Middle) | (r_i is Middle) |
| 3. If (Iteration is High) | then (β is High) | (r_i is Low) |
-

Figure 12. Rule set from original Type-1 FLS for parameter adaptation

Membership functions in the Type-1 FLS and interval Type-2 FLS are for the one input and the two outputs of the fuzzy system. The system has one input – the parameter "Iteration", which has three membership functions with Low, Middle and High linguistic values and the two outputs – parameters "Beta" (β) and "Pulse Rate" (r_i) which has three membership functions with Low, Middle and High linguistic values.

4.2 Intuitionistic fuzzy logic system

Considering the variable *Iteration* (Eq. (14)) it can be assigned intuitionistic maximum and minimum values ($Iteration_{max}$ and $Iteration_{min}$). So, if the current *Iteration* falls outside the intuitionistic limits can be unambiguously assigned to rules 1) and 3) presented in Fig. 12. Conversely, values between intuitionistic limits ($Iteration_{max}$ and $Iteration_{min}$) cannot be determined unambiguously. In this case the rule 2), Fig. 12. In the case of IFL system the following membership functions are defined:

$$\begin{aligned}\mu_A &: (Iteration \leq Iteration_{min}) \\ \pi_A &: \text{if } (Iteration_{min} < Iteration < Iteration_{max}) \\ \nu_A &: \text{if } (Iteration \geq Iteration_{max})\end{aligned}$$

The intuitionistic fuzzy rule set used for parameter adaptation in the BA is shown in Fig. 13.

1. If ($Iteration \leq Iteration_{\min}$)	then (β is Low)	(r_i is High)
2. If ($Iteration_{\min} < Iteration < Iteration_{\max}$)	then (β is Middle)	(r_i is Middle)
3. If ($Iteration \geq Iteration_{\max}$)	then (β is High)	(r_i is Low)

Figure 13. Rule set from intuitionistic fuzzy rule system for parameter adaptation

Considering IFL system the design of the output variables, β and r_i , is analogical to the presented one in Fig. 10 and in Fig. 11. According to [9] the geometrical forms of the intuitionistic fuzzy numbers can be generalized as follows:

For the first case functions μ_A and ν_A satisfied the conditions [9]:

$$\sup_{y \in E} \mu_A(y) = \mu_A(x) = a, \quad \inf_{y \in E} \nu_A(y) = \nu_A(x) = b,$$

for each $x \in [x_1, x_2]$, and for the second case [9]:

$$\sup_{y \in E} \mu_A(y) = \mu_A(x_0) = a, \quad \inf_{y \in E} \nu_A(y) = \nu_A(x_0) = b.$$

For the first case we have:

- μ_A is increasing function from $-\infty$ to x_1 ;
- μ_A is decreasing function from x_2 to $+\infty$;
- ν_A is decreasing function from $-\infty$ to x_1 ;
- ν_A is increasing function from x_2 to $+\infty$.

For the second case we have:

- μ_A is increasing function from $-\infty$ to x_0 ;
- μ_A is decreasing function from x_0 to $+\infty$;
- ν_A is decreasing function from $-\infty$ to x_0 ;
- ν_A is increasing function from x_0 to $+\infty$.

Obviously, in both cases the functions μ_A and ν_A can be represented in the form

$$\mu_A = \mu_A^{\text{left}} \cup \mu_A^{\text{right}}, \quad \nu_A = \nu_A^{\text{left}} \cup \nu_A^{\text{right}},$$

where μ_A^{left} and ν_A^{left} are the left, while μ_A^{right} and ν_A^{right} are the right sides of these functions.

Therefore, the above conditions can be re-written in the (joint) form [9]:

$$\sup_{y \in E} \mu_A(y) = \mu_A(x) = a, \quad \inf_{y \in E} \nu_A(y) = \nu_A(x) = b,$$

for each $x \in [x_1, x_2]$ and in the particular case, when $x_1 = x_0 = x_2$, μ_A^{left} is increasing function; μ_A^{right} is decreasing function; ν_A^{left} is decreasing function and ν_A^{right} is increasing function.

Following [9], we will consider, ordered by generality, the definitions:

1. In the graphical representation in both cases above $a = 1, b = 0$.
2. $\sup_{y \in E} \mu_A(y) = \mu_A(x_0) > 0.5 > \nu_A(x_0) = \inf_{y \in E} \nu_A(y)$.
3. $\sup_{y \in E} \mu_A(y) = \mu_A(x_0) \geq 0.5 \geq \nu_A(x_0) = \inf_{y \in E} \nu_A(y)$.
4. $\sup_{y \in E} \mu_A(y) = \mu_A(x_0) > \nu_A(x_0) = \inf_{y \in E} \nu_A(y)$.
5. $\sup_{y \in E} \mu_A(y) = \mu_A(x_0) \geq \nu_A(x_0) = \inf_{y \in E} \nu_A(y)$.
6. $\sup_{y \in E} \mu_A(y) = \mu_A(x_0) > 0$.
7. $\inf_{y \in E} \nu_A(y) = \nu_A(x_0) < 1$.

In considered here BA parameter adaptation, for simplicity, it can be used the presented in Fig. 10 and Fig. 11 design of the output variables applying the intuitionistic fuzzy rule set.

5 Conclusions

One of the main challenges of the field of metaheuristic algorithms is the parameter control. In order to increase the performance of the algorithms it is necessary to adapt the algorithm parameters during the computation. Such a procedure is not a trivial. In this paper, Type-1, interval Type-2 fuzzy logic and intuitionistic fuzzy logic systems to dynamically adaptation of BA parameters are proposed. The presented methods perform dynamic adaptation of considered parameters, during the algorithm run, trying to improve the BA performance.

Acknowledgment

We would like to express our gratitude to the CONACYT and Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

References

- [1] Olivas, F., Valdez, F. & Castillo, O. (2013) Particle swarm optimization with dynamic parameter adaptation using interval type-2 fuzzy logic for benchmark mathematical functions, *2013 World Congress on Nature and Biologically Inspired Computing (NaBIC)*, 12-14 Aug 2013, Fargo, ND.
- [2] Olivas, F., Valdez, F. & Castillo, O. (2015) Dynamic parameter adaptation in Ant Colony Optimization using a fuzzy system for TSP problems, *IFSA-EUSFLAT 2015*, Gijón, Asturias, Spain, 765–770.
- [3] Wang, G. & Guo, L. (2013) A Novel Hybrid Bat Algorithm with Harmony Search for Global Numerical Optimization, *Journal of Applied Mathematics*, Volume 2013, 21 pages.

- [4] Perez, J., Valdez, F. & Castillo, O. (2015) Modification of the Bat Algorithm using fuzzy logic for dynamic parameter adaptation, *CEC2015 IEEE Congress on Evolutionary Computation*, Sendai, Japan, May 26, 2015.
- [5] Atanassov, K. (1986) Intuitionistic Fuzzy Sets, *Fuzzy Sets and Systems*, 20(1), 87–96.
- [6] Atanassov, K. (1983) Intuitionistic Fuzzy Sets, *VII ITKR Session*, Sofia, 20–23 June 1983, Reprinted: *Int. J. Bioautomation*, 2016, 20(S1), S1–S6.
- [7] Atanassov, K. (1999) *Intuitionistic Fuzzy Sets: Theory and Applications*, Springer, Heidelberg.
- [8] Atanassov, K. (2012) *On Intuitionistic Fuzzy Sets Theory*, Springer Physica-Verlag, Berlin.
- [9] Atanassov, K. Vassilev, P., Tsvetkov, R. (2013) *Intuitionistic Fuzzy Sets, Measures and Integrals*. Academic Publishing House "Prof. Marin Drinov", Sofia.
- [10] Atanassov, K. (1988) Review and New Results on Intuitionistic Fuzzy Sets, Mathematical Foundations of Artificial Intelligence Seminar, Sofia, 1988, *Preprint IM-MFAIS-1-88*. Reprinted: *Int. J. Bioautomation*, 2016, 20(S1), S7–S16.
- [11] L Amador-Angulo, L. & Castillo, O. (2014) Optimization of the Type-1 and Type-2 Fuzzy Controller Design for the Water Tank using the Bee Colony Optimization, *Norbert Wiener in the 21st Century, (21CW), 2014 IEEE Conference*, Boston, MA, 8 pages.
- [12] Amador-Angulo, L. & Castillo, O. (2015) Statistical Analysis of Type-1 and Interval Type-2 Fuzzy Logic in dynamic parameter adaptation of the BCO, *IFSA-EUSFLAT'2015*, 776–783.
- [13] Goel, N., Gupta, D. & Goel, S. (2013) Performance of Firefly and Bat Algorithm for Unconstrained Optimization Problems, *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(5), 1405–1409.
- [14] Castillo, O. & Melin, P. (2012) A review on the design and optimization of interval Type-2 fuzzy controllers, *Appl. Soft Computing*, 12(4), 1267–1278.
- [15] Roeva, O. & Michalíková, A. (2013) Generalized net model of intuitionistic fuzzy logic control of genetic algorithm parameters, *Notes on Intuitionistic Fuzzy Sets*, 19(2), 71–76.
- [16] Roeva, O. & Michalíková, A. (2014) Intuitionistic fuzzy logic control of metaheuristic algorithms parameters via a Generalized net, *Notes on Intuitionistic Fuzzy Sets*, 20(4), 53–58.
- [17] Yilmaz, S. & Küçüksille, E. (2015) A new modification approach on bat algorithm for solving optimization problems, *Applied Soft Computing*, 28, 259–275.
- [18] Yang, X. S. (2010) *A New Metaheuristic Bat-Inspired Algorithm*, Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK.
- [19] Yang, X. S., Fister I., Rauter, S., Ljubic, K. & Fister I., Jr. (2015) Planning the sports training sessions with the bat algorithm, *Neurocomputing*, 149, 993–1002.