

# A comparison of bio-inspired optimization methodologies applied to the tuning of industrial controllers

Macarena Céspedes, Mónica Contreras, Joaquín Cordero,  
Gustavo Montoya, Karen Valverde and José David Rojas  
Escuela de Ingeniería Eléctrica  
Universidad de Costa Rica  
San José, Costa Rica  
email: jdrojas@eie.ucr.ac.cr

**Abstract**—A comparison of different bio-inspired methods is presented for the problem of tuning the parameters of a PID industrial controller for second order plus time delay plants. The compared algorithms are representative of three different kind of methodologies (evolutionary, swarm-based and ecology-based). All methods are briefly explained and are fully implemented to solve the control problem at hand. It was found that all methods are well suited to solve the problem, but they differ in its computational cost. However in all cases, the different methodologies tested were able to find similar minimum values for different families of plants.

## I. INTRODUCTION

Human have always found inspiration on nature. Since 1970, with the introduction of neural networks, several different bio-inspired computing methods have been appearing that found its roots in the study of the behavior of animals and plants [1]. Since then, different bio-inspired methods of optimization have been presented and applied to a large variety of problem. In particular, the optimal tuning of Proportional-Integral-Derivative (PID) controllers is an important problem that still is not closed.

PID control is the most important algorithm used to regulate industrial processes [2]. The problem of finding the optimal tuning of industrial controller is considered to be one of the many optimization problems that can benefit from bio-inspired algorithms [3], and have been subject of several different studies and applications (See [4]–[7] for a small set of examples).

PID controllers have three main parameters, the proportional gain the integral time and the derivative time. Even the fact that the number of decision variables are low, the problem is far from trivial [8]. The different aspects of the controlled system performance has derived in a plethora of different tuning rules and methodologies to find the “best” possible tuning [9], [10].

The contribution of this paper is precisely to compare how different bio-inspired optimization methods perform in solving the problem of industrial PID tuning for second order plus time delay (SOPTD) processes. It is common

to model industrial plants as first order plus time delay systems, however it is advantageous to model these plants as SOPTD because the dynamics of real higher order plants can be well approximated using only second order models.

According to [11], bio-inspired optimization can be divided into three general branches: Evolutive algorithms, swarm-based and ecology-based. Five different methodologies were chosen in order to test how well they adapt to solve the problem: for the evolutive branch, the original Genetic Algorithm [12] method was chosen. For the swarm branch, Particle Swarm Optimization [13] and Ant Colony Optimization [14] were selected while for the ecological branch, Linear Biogeography-based Optimization [15] and Invasive Weed Optimization [16] were chosen.

Other authors have also made a similar comparison [17]. However, the approach in this paper takes into account other considerations. First, the model chosen in this paper is more representative of industrial processes, since it takes into account the time delay, which is an important characteristic of this kind of plants. Also, the time delay adds an extra layer of complexity to the problem. Second, the cost function selected to optimize in this paper is the Integral of the Absolute value of the Error (IAE) of the disturbance rejection response, while in [17] the overshoot and settling time of the set-point response are selected. It is known that the minimization of IAE is an important factor for the tuning of industrial controllers and it has been widely accepted as good measure of closed-loop performance, because disturbance rejection is of primary importance over set-point tracking in industrial processes [18].

The rest of the paper is divided as follows. In Section II, the bio-inspired methods are briefly presented along with the control framework and the problem formulation. Then, in Section III, the methodologies are employed to solve the control problem and are compared from the IAE and computational point of view. Conclusions are presented at the end of the paper in Section IV.

## II. METHODS

### A. Control framework

Consider a feedback controlled system as shown in Fig. 1. The objective is to maintain the process output  $y(s)$  as close as possible to the reference input  $r(s)$  despite the presence of disturbances  $d_i(s)$ . For this, it uses the difference between these signals, to compute the required control signal  $u(s)$ , in such a way that the closed system is able to reach zero steady state error as fast as possible. The elements of Figure 1 are:

- Controller  $C(s, \theta)$ : In industrial processes, the most employed controller is the Proportional-Integral-Derivative (PID) algorithm. It is common to define  $\theta$  as the vector formed by the controller parameters  $\theta = [K_p \ T_i \ T_d]^T$ , where:
  - $K_p$  is the proportional gain,
  - $T_i$  is the integral time constant,
  - $T_d$  is the derivative time constant,
- $P(s)$  represents the controlled process. A Second-Order Plus Time Delay model (SOPTD) is considered as a good representation of most process in industry [19]. Its transfer function has the form:

$$P(s) = \frac{K e^{-Ls}}{(T_1 s + 1)(T_2 s + 1)}, \quad (1)$$

where  $K$ ,  $L$ ,  $T_1$  and  $T_2$ , correspond to the static gain, the time delay and time constants respectively. It is common to define  $T_2$  as the less dominant lag time of the system and therefore  $T_2 = aT_1$ .

The control signal (the output of the controller) can be computed as:

$$u(s) = K_p \left( (r(s) - y(s)) + \frac{1}{T_i s} (r(s) - y(s)) - \frac{T_d s}{\alpha T_d s + 1} y(s) \right) \quad (2)$$

with  $\alpha$  the constant for the derivative filter.

### B. Problem formulation

To measure the performance of a given control strategy, it is common to use the Integral of the Absolute value of the Error (IAE) given by:

$$J(\theta) = \int_0^{\infty} |r(t) - y(t)| dt, \quad (3)$$

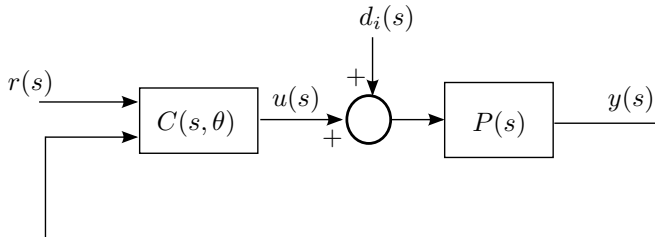


Fig. 1. Feedback control loop.

if  $J(\theta)$  is computed when only a change in the reference signal is performed,  $J(\theta)$  is defined as  $J_r(\theta)$ , whereas if  $J(\theta)$  is measured only when a change in the disturbance input is applied is called  $J_{di}(\theta)$ .  $J_r(\theta)$  is a cost function of the performance of the controller to work as a servo-control while  $J_{di}$  is a measure of the performance as a regulator. In the process industry, a change in the reference is less common than the need to reject disturbances. Therefore, for this paper, the optimization problem is defined as the minimization of  $J_{di}(\theta)$ :

$$\min_{\theta} J_{di}(\theta) \quad (4)$$

This problem formulation is very common for PID tuning rules, however, according to the knowledge of the authors, there is not a paper that compares the performance of different bio-inspired optimization methods, to solve the tuning of the parameters with an industrial PID with a SOPTD plant.

### C. Bio-inspired optimization methodologies

In the following, the bio-inspired optimization methodologies compared in this paper are briefly described. Due to space constraints, the reader is encourage to follow the references for more detailed explanation of each method.

1) *Ant Colony Optimization*: One of the characteristics that has been observed on ants is that they are able to find the shortest path between the nest and the food source. In order to accomplish this, the ants communicate with each other by producing changes in the environment, as pointed by [20]: “these ants deposit pheromone on the ground in order to mark some favorable path that should be followed by other members of the colony. Ant colony optimization (ACO) exploits a similar mechanism for solving optimization problems”

The idea is that, in the beginning, a number of artificial ants randomly select one path. This path is represented by the possible values of the decision variables, that can be chosen as a solution to the problem. Certain quantity of artificial pheromones is added to this particular choice, according to the fitness of the solution. The larger the quantity of pheromones, the better the chances that the ants select that particular path, because a path with bigger quantity of pheromones represents a shorter path to the source of food [21].

2) *Invasive Weed Colony Optimization*: Invasive Weed Optimization (IWO) is a numerical stochastic search algorithm proposed by Mehrabian and Lucas in 2006. They were inspired by the ecological process of weed colonization and distribution [11]. The method is capable of solving general multidimensional, linear and nonlinear optimization problems efficiently by mimicking the robustness, adaptation and randomness of colonizing weeds in a simple but effective optimizing algorithm [16].

The steps of this algorithm are described as follows:

- 1) Initialize a population: The method begins with a population ( $nPob0$ ) of initial solutions widespread

over the  $d$  dimensional problem space with random positions [16].

- 2) **Reproduction:** Each member of the population is allowed to produce seeds depending on its own, as well as the colony's, lowest and highest fitness.
- 3) **Spatial dispersal:** The generated seeds are randomly distributed over the search space by normally distributed random numbers with a mean equal to zero, but with a varying variance. This ensures that seeds will be randomly distributed in such a way that they abide near to the parent plant.
- 4) **Competitive exclusion:** If a plant leaves no offspring then it would go extinct. After passing some iteration, the number of weeds in a colony will reach its maximum. At this moment, each weed is allowed to produce seeds. The produced seeds are then allowed to spread over the search area. When all seeds have found their position in the search area, they are ranked together with their parents (as a colony of weeds). Next, weeds with lower fitness are eliminated to reach the maximum allowable population in a colony. In this way, weeds and seeds are ranked together and the ones with better fitness survive and are allowed to replicate. The population control mechanism also is applied to their offspring to the end of a given run, realizing competitive exclusion [16].

3) *Linear Biogeography-based optimization:* Biogeography is the branch of biology that studies the geographical distribution of plants and animal and the mathematical models associated with the extinction and migration of species [22].

Biogeography-based optimization (BBO) is an evolutionary algorithm that has been employed to solve problems in different areas as engineering, economics, medicine, etc. [23]. Each possible solution to the optimization problem is considered to be a different habitat (or island).

The values of the components of the solutions are analogous to the characteristics of the habitat. A solution with a good fitness is considered to be equivalent to a habitat with good characteristics for the thrive of different species. Good habitats are considered to have high rate of emigration, because its good features allows an increase in the number of species. This increase may lead to a saturation in its capacity to house more species. The solutions with lower fitness has a high rate of immigration, because animals and plants search for less concentrated habitats to grow and reproduce.

The idea behind BBO is to interchange the characteristics of each habitat according to their immigration and emigration rates, in order to improve all habitats characteristics. The linear BBO (LBBO) variant is equivalent to the BBO, except that, instead of interchange the characteristics of each habitat, the new features are computed using a linear transformation between the features of each habitat.

4) *Genetic Algorithms:* The Genetic Algorithms (GA) methodology is based on the ideas of evolution, genetics and natural selection [24]. According to [25] it is possible to consider the methodology of genetic algorithms as a way to simulate natural selection in order to solve optimization problems. The main characteristics of GA can be summarized as [25]:

- It is based on the simulation of a biological system, which includes a population of individuals with the ability to reproduce.
- The individuals have a finite life span.
- There exist variations in the characteristics of the population.
- There is a positive correlation between the ability to survive and the ability to reproduce.

The idea behind GA is that the individuals with the best fitted are able to pass their characteristics to the next generation. To accomplish this, the characteristics of the best are merged to create a new generation and random mutations are incorporated into the genetic pool [26].

From the optimization point of view, each individual represent a possible solutions to the problem. In that sense, the individuals are real numbers coded in binary chains, in order to represent the genetic information. Each individual bit represent a gene while the complete chain represent a chromosome. The cost function of the optimization problem represent the degree of adaptation of the individuals. According to [27], this fitness represents the probability of survival and reproduction of an individual. The selection algorithm within the GA decides which individuals of each generation are able to have offspring.

5) *Particle swarm optimization:* Particle swarm optimization (PSO) is a search algorithm based on a population whose individuals are called particles. Each of these particles is a possible solution to the problem. The algorithm is initialized with a population of random solutions. Unlike other techniques of evolutionary computing, every particle in PSO is associated with a speed. The particles fly through the search space with speeds that are adjusted according to their historical behavior, i.e., they depend on their own experience and the experience of neighboring particles [28].

This algorithm has its roots in the study on how flocks of birds and fish banks move as a unit, but without an apparent leader (using what has been called swarm intelligence) [13]. The particles travel through the search space, trying to find the optimal solution. Each particle keeps track of its coordinates in the search space, which are associated with the best solution achieved so far. The value of this best solution is also stored. At each iteration, the acceleration of each particle is changed to move them to the best solutions found.

### III. RESULTS AND DISCUSSION

In this section, the results of the application of each bio-inspired method to the control problem presented in

Section II-B are presented. All the optimization were performed in a computer with an Intel Core© i5-3470 CPU at 3.20GHz and 8 GB of RAM using MATLAB© 2015b.

The results of each method are presented in Table I and Table II. For comparison purposes, the result of the optimization using the interior-point (IP) algorithm is presented. For the case of the bio-inspired methods, a minimum of 125 iteration where performed to ensure that the algorithms have opportunity to explore the complete solution space. In all cases, the initial value of the decision variables were set to be near the point given by the uSORT method with a maximum sensitivity of two [29]. The search space was restricted around  $\pm 20\%$  of the uSORT point to avoid an unstable response during each iteration of the methods.

In Table I and Table II, the presented values correspond to  $K_p$ ,  $T_i$ ,  $T_d$  and  $J_{di}$ . The methods were tested with 100 different experiments applied to 9 different plants (i.e. each method was executed 900 times in total). The characteristics of the plants are presented in Table III.  $K = 1$  and  $T = 1$  where chosen, because it is common to optimize a normalized version of the plants to then applied a de-normalization factor. That way, one is able to tune the controller for a complete family of plants, rather than a single case.

As it can be seen all the methods are able to either match the result from IP or get near to it. IWO and PSO yield almost the same results and both find the minimum value, follow by GA, LBBO and ACO. The reason why this may be the case is that ACO requires to set all the possible values that the decision variables can achieve beforehand. The method then tries to find the best combination of values to minimize the cost function. However the other methods are able to find new results that were not contemplated from the beginning. However, when comparing the disturbance rejection response, as in Fig. 2, all methods produce an acceptable result. The results on Table I and Table II are the average values obtained after the 100 experiments and the simulation in Fig. 2 is the application of the average parameter of the controller with plant P5. IWO, GA and PSO are able to produce the same overshoot which is also lower than the overshoot from ACO and LBBO.

The computational cost of each method is presented in Table IV. It can be seen that the method with higher number of iteration, in average, is LBBO, followed by PSO. IWO and GA has the lowest number of mean iterations.

It is clear that the function count for the bio-inspired methods are much higher than in the IP case. The reason behind this is that all bio-inspired methods have a high number of agents (particles, ants, genes, habitats, seeds, etc). For this paper, 100 agents were used for each method. PSO has a particularly low number of function counts compared with the other methods.

With respect to the time spend in each iteration, the

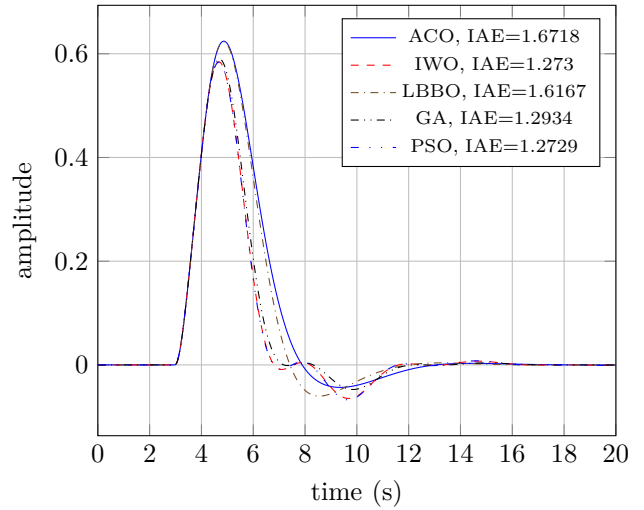


Fig. 2. Simulation of the different bio-inspired optimization methods to an unitary step change in the disturbance for plant P5.

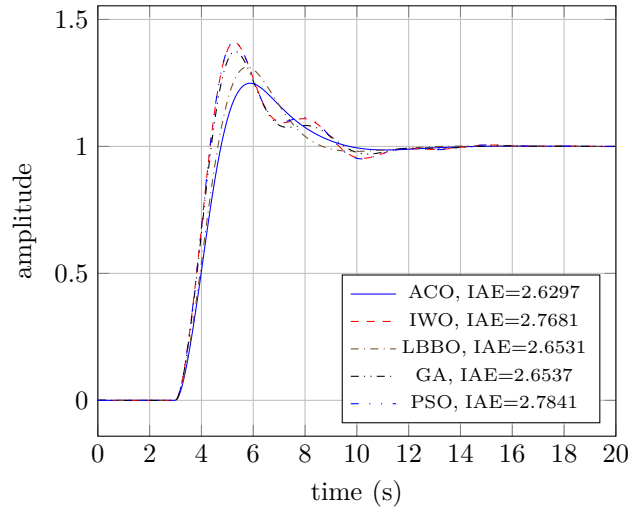


Fig. 3. Simulation of the different bio-inspired optimization methods to a unitary step change in reference for plant P5.

method that needs the longest time is ACO, while LBBO is the fastest algorithm. It can be concluded from the mean value of the standard deviation that, in general, all the methods spend the same time at each iteration. Only IWO has more variability than the other methods.

Despite the fact that the controllers are tuned for disturbance rejection, they can also be employed for set-point tracking. The response is presented in Fig. 3. It can be seen from the response that the ACO method has lower  $J_r$  while the higher IAE value correspond to the IWO and PSO methods. This is the expected result, because, as it is well known in the control engineering field, there exists a compromise between the servo and regulator responses.

This compromise can be solved with a more complex topology, for example, using a two degrees of freedom controller. However, from the optimization point of view, this new topology implies a completely different problem to solve. For instance, it would be necessary to minimize

TABLE I  
OBTAINED AVERAGE RESULTS BY PLANT (1 OF 2).

Plant	IP	ACO	IWO
	$\{K_p, T_i, T_d, J_{di}\}$		
P1	{8.83, 0.21, 0.05, 0.03}	{8.69,0.21,0.05,0.03}	{8.83, 0.21, 0.05, 0.03}
P2	{12.01, 0.47, 0.22, 0.05}	{8.29,0.69,0.20,0.11}	{12.02, 0.47, 0.22, 0.05}
P3	{17.35, 0.60, 0.25, 0.04}	{11.81,0.88,0.26,0.11}	{17.35, 0.60, 0.25, 0.04}
P4	{1.29, 1.05, 0.42, 0.96}	{1.04,1.09,0.32,1.20}	{1.29, 1.06, 0.42, 0.96}
P5	{1.25, 1.28, 0.71, 1.28}	{0.95,1.35,0.51,1.67}	{1.25, 1.28, 0.71, 1.28}
P6	{1.50, 1.43, 0.95, 1.29}	{1.15,1.73,0.64,1.76}	{1.50, 1.43, 0.95, 1.29}
P7	{0.81, 1.60, 0.71, 2.28}	{0.62, 1.51, 0.58, 2.72}	{0.81, 1.60, 0.71, 2.28}
P8	{0.82, 1.89, 0.89, 2.68}	{0.61, 1.77, 0.73, 3.25}	{0.82, 1.89, 0.89, 2.68}
P9	{0.96, 2.21, 1.07, 2.72}	{0.69, 2.11, 0.87, 3.44}	{0.96, 2.21, 1.06, 2.72}

TABLE II  
OBTAINED AVERAGE RESULTS BY PLANT (2 OF 2).

Plant	LBBO	GA	PSO
	$\{K_p, T_i, T_d, J_{di}\}$		
P1	{7.315, 0.224, 0.047, 0.041}	{8.82, 0.21, 0.05, 0.03}	{8.83, 0.21, 0.05, 0.03}
P2	{9.790, 0.541, 0.183, 0.072}	{11.97, 0.49, 0.22, 0.05}	{12.02, 0.47, 0.22, 0.05}
P3	{14.150, 0.684, 0.235, 0.058}	{17.24, 0.62, 0.25, 0.04}	{17.35, 0.60, 0.25, 0.04}
P4	{1.042, 1.048, 0.283, 1.219}	{1.27, 1.09, 0.40, 0.98}	{1.29, 1.05, 0.42, 0.96}
P5	{1.011, 1.342, 0.462, 1.623}	{1.23, 1.36, 0.65, 1.30}	{1.25, 1.27, 0.72, 1.28}
P6	{1.209, 1.602, 0.583, 1.703}	{1.49, 1.64, 0.79, 1.32}	{1.50, 1.44, 0.94, 1.29}
P7	{0.660, 1.550, 0.504, 2.635}	{0.80, 1.61, 0.69, 2.29}	{0.81, 1.61, 0.71, 2.28}
P8	{0.682, 1.837, 0.649, 3.042}	{0.82, 1.90, 0.88, 2.69}	{0.82, 1.89, 0.89, 2.68}
P9	{0.775, 2.156, 0.774, 3.198}	{0.94, 2.21, 1.05, 2.74}	{0.96, 2.21, 1.06, 2.72}

TABLE III  
PARAMETER OF THE TEST PLANTS.

Plant	Parameters $\{K, T, L, a\}$
P1	{1, 1, 0.1, 0.0}
P2	{1, 1, 0.1, 0.5}
P3	{1, 1, 0.1, 1.0}
P4	{1, 1, 1.0, 0.0}
P5	{1, 1, 1.0, 0.5}
P6	{1, 1, 1.0, 1.0}
P7	{1, 1, 2.0, 0.0}
P8	{1, 1, 2.0, 0.5}
P9	{1, 1, 2.0, 1.0}

two functions at the same time, on a larger search space. The comparison made in this paper can be extended to the multi-objective case and the results will be presented elsewhere.

#### IV. CONCLUSIONS

In this paper, a comparison of different bio-inspired optimization methodologies was performed to solve the tuning of PID industrial controllers for disturbance rejection. The algorithms that were tested were representative examples of the three main branches of bio-inspired optimization methodologies: Evolutive algorithms, swarm-based and ecology-based.

It was found that all the methodologies are well suited to solve this particular control problem. IWO and PSO where the methods that find the lower  $J_{di}$  value. LBBO was found to be the method with higher function count but also was the faster method. PSO was the method

that needed less function count. GA and IWO were the methods that needed less quantity of iteration.

This comparison of methodologies, can be augmented to take into account also  $J_r$  for a two degrees of freedom controller. This new results are currently a work-in-progress and will be presented elsewhere.

The implementation of each method can be given upon request to the corresponding author.

#### ACKNOWLEDGEMENTS

This work was done with grant 322-B4-218 by Vicerrectoría de Investigación de la Universidad de Costa Rica

#### REFERENCES

- [1] A. K. Kar, "Bio inspired computing - a review of algorithms and scope of applications," *Expert Systems with Applications*, vol. 59, pp. 20 – 32, 2016.
- [2] V. M. Alfaro and R. Vilanova, *Model-Reference Robust Tuning of PID Controllers*, 1st ed., ser. Advances in Industrial Control. Springer International Publishing, 2016.
- [3] G. Reynoso-Meza, J. Sanchis, X. Blasco, and J. M. Herrero, "Multiobjective evolutionary algorithms for multivariable PI controller design," *Expert Systems with Applications*, vol. 39, no. 9, pp. 7895 – 7907, 2012.
- [4] H. Boubertakh, M. Tadjine, P. Y. Glorennec, and S. Labiod, "Tuning fuzzy PID controllers using ant colony optimization," in *Control and Automation, 2009. MED '09. 17th Mediterranean Conference on*, June 2009, pp. 13–18.
- [5] Z. Chen, S. Wang, Z. Deng, and X. Zhang, "Tuning of auto-disturbance rejection controller based on the invasive weed optimization," in *Bio-Inspired Computing: Theories and Applications (BIC-TA), 2011 Sixth International Conference on*, Sept 2011, pp. 314–318.

TABLE IV  
COMPUTATIONAL PERFORMANCE FOR DIFFERENT OPTIMIZATION METHODS AND DIFFERENT PLANTS.

Method	Number of iterations		Function count	Iteration time		
	mean	max		mean	max	average std dev
IP	51	90	284	0.011	0.036	0.007
ACO	135	135	6750	0.118	0.125	0.001
IWO	125	125	6106	0.059	0.127	0.021
LBBO	337	500	7706	0.032	0.052	0.008
GA	125	125	6300	0.073	0.078	0.001
PSO	148	253	2976	0.060	0.072	0.002

- [6] D. Nangru, D. K. Bairwa, K. Singh, S. Nema, and P. K. Padhy, "Modified PSO based PID controller for stable processes," in *Control, Automation, Robotics and Embedded Systems (CARE), 2013 International Conference on*, Dec 2013, pp. 1–5.
- [7] R. J. Rajesh and C. M. Ananda, "PSOtuned PID controller for controlling camera position in UAV using 2-axis gimbal," in *Power and Advanced Control Engineering (ICPACE), 2015 International Conference on*, Aug 2015, pp. 128–133.
- [8] G. Reynoso-Meza, J. Sanchis, X. Blasco, and M. Martínez, "Algoritmos evolutivos y su empleo en el ajuste de controladores del tipo PID: Estado actual y perspectivas," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 10, no. 3, pp. 251 – 268, 2013.
- [9] A. O'Dwyer, *Handbook of PI and PID Controller Tuning Rules*, 3rd ed. Imperial College Press, 2009.
- [10] M. Contreras Leiva and J. Rojas, "New tuning method for PI controllers based on Pareto-optimal criterion with robustness constraint," *Latin America Transactions, IEEE (Revista IEEE America Latina)*, vol. 13, no. 2, pp. 434–440, Feb 2015.
- [11] S. Binitha and S. Sivasathya, "A survey of bio inspired optimization algorithms," *International Journal of Soft Computing and Engineering*, vol. 2, no. 2, pp. 137–151, 2012.
- [12] P. J. Fleming and C. M. Fonseca, "Genetic algorithms in control systems engineering: a brief introduction," in *Genetic Algorithms for Control Systems Engineering, IEE Colloquium on*, May 1993, pp. 1/1–1/5.
- [13] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.
- [14] C. Blum, "Ant colony optimization: Introduction and recent trends," *Physics of Life Reviews*, vol. 2, no. 4, pp. 353 – 373, 2005.
- [15] D. Simon, M. G. Omran, and M. Clerc, "Linearized biogeography-based optimization with re-initialization and local search," *Information Sciences*, vol. 267, pp. 140 – 157, 2014.
- [16] A. Mehrabian and C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization," *Ecological Informatics*, vol. 1, no.4, pp:355-366., 2006.
- [17] B. Nagaraj and N. Muruganath, "A comparative study of PID controller tuning using GA, EP, PSO and ACO," in *Communication Control and Computing Technologies (ICCCCT), 2010 IEEE International Conference on*, Oct 2010, pp. 305–313.
- [18] F. G. Shinsky, "Process control: As taught vs as practiced," *Industrial & Engineering Chemistry Research*, vol. 41, no. 16, pp. 3745–3750, 2002.
- [19] V. M. Alfaro and R. Villanova, "Set-point weight selection for robustly tuned PI/PID regulators for over damped processes," *17th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2012)*, 2012.
- [20] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, Nov 2006.
- [21] S. Goss, S. Aron, J. L. Deneubourg, and J. M. Pasteels, "Self-organized shortcuts in the argentine ant," *Naturwissenschaften*, 1989.
- [22] R. MacArthur and E. Wilson, *The theory of Island Biogeography*. Princeton New Jersey, 1967.
- [23] D. Simon, "Biogeography-based optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 12, no. 6, pp. 702–713, Dec 2008.
- [24] L. Araujo and C. Cervigón, *Algoritmos Evolutivos: Un Enfoque Práctico*. Alfaomega, 2009.
- [25] D. Simon, *Evolutionary Optimization Algorithms*. John Wiley and Sons, 2013.
- [26] M. Mitchell, "Genetic algorithms: An overview," *Complexity*, 1995.
- [27] M. Gestal, D. Rivero, J. R. Rabuñal, J. Dorado, and A. Pazos, *Introducción a los Algoritmos Genéticos y a la Programación Genética*. Universidade da Coruña, 2010.
- [28] Y. Shi, "Particle swarm optimization," *IEEE Connections*, vol. 2, no. 1, pp. 8–13, 2004.
- [29] V. M. Alfaro and R. Vilanova, "Optimal robust tuning for 1DoF PI/PID control unifying FOPDT/SOPDT models," in *Advances in PID Control*, University of Brescia, Brescia, Italy, 3 2012, pp. 572–577.