# A Review of Techniques for On-line Control of Parameters in Swarm Intelligence and Evolutionary Computation Algorithms

**Rafael Stubs Parpinelli***

Graduate Program in Applied Computing
Santa Catarina State University
Joinville, SC, Brazil
E-mail: rafael.parpinelli@udesc.br
*Corresponding author

**Guilherme Felippe Plichoski**

Graduate Program in Applied Computing
Santa Catarina State University
Joinville, SC, Brazil
E-mail: guilherme.plichoski@edu.udesc.br

**Renan Samuel da Silva**

Graduate Program in Applied Computing
Santa Catarina State University
Joinville, SC, Brazil
E-mail: renan.samuel.da.silva@gmail.com

**Pedro Henrique Narloch**

Graduate Program in Applied Computing
Santa Catarina State University
Joinville, SC, Brazil
E-mail: pedro.narloch@gmail.com

**Abstract:** The two major groups representing biologically inspired algorithms are Swarm Intelligence (SI) and Evolutionary Computation (EC). Both SI and EC share common features such as the use of stochastic components during the optimization process and various parameters for configuration. The setup of parameters in swarm and in evolutionary algorithms has an important role in defining their behavior, guiding the search and biasing the quality of final solutions. In addition, an appropriate setting for the parameters may change during the optimization process making this task even harder. The present work brings an up-to-date discussion focusing on on-line parameter control strategies applied in SI and EC. Also, this review analyzes and points out the key techniques and algorithms used and suggests some directions for future research.

**Keywords:** Parameter Control; Bio-inspired Algorithms; Meta-heuristics; Natural Computing; Parameterless Algorithms

**Reference** to this paper should be made as follows: Parpinelli, R.S., Plichoski, G.F., da Silva, R.S. and Narloch, P.H. (2019) ‘A review of techniques for online control of parameters in swarm intelligence and evolutionary computation algorithms’, Int. J. Bio-Inspired Computation, Vol. 13, No. 1, pp.1–20.

## 1 Introduction

Nature constantly offers sources of inspiration for the development of optimization algorithms. Such algorithms are commonly referenced to biologically inspired (or bio-inspired) algorithms and are part of a research field called Natural Computing (Boussaid et al. 2013). Among all different bio-inspired algorithms it is possible to highlight two groups: the Swarm Intelligence and the Evolutionary Computation. Swarm Intelligence (SI) is characterized by algorithms inspired in the collective behavior of insects such as ants, bees, termites and also animals such as fish and birds. The intelligence of those systems are found in the collective and self-organized behavior that appears from local interactions and this phenomena is called emergent behavior (Parpinelli & Lopes 2011). Evolutionary Computation (EC) is another important group within the bio-inspired meta-heuristics being characterized by algorithms essentially inspired by Darwin's theory, where fittest and adapted individuals have greater chances of survival and evolve through natural selection. In this analogy, the individuals of a population represent candidate solutions to optimize a given problem and the environment represents the solution space. Evolutionary Computation simulates the evolution of individuals through processes of selection, reproduction, crossover, and mutation, aiming to probabilistically produce, generation after generation, a better pool of solutions (De Jong 2006).

In spite of the variety of optimization algorithms that both groups provide, they mostly share some common characteristics such as the use of stochastic components (i.e. that have some inherent randomness in which random variables are associated with inner decisions of such components) and parameters to be tuned (Yang 2010*a*). The amount of parameters can vary from a few like in the Firefly Algorithm (Yang 2010*a*), and the Artificial Bee Colony Optimization algorithm (Karaboga & Akay 2009) with 2 parameters each, to many parameters as in the Bacterial Foraging Optimization algorithm (Passino 2002), and in the Gene Expression Programming (Ferreira 2001) with 7 and 11 parameters, respectively.

In any meta-heuristic the adjustment of parameters plays an important role, influencing its behavior in the search for promising regions in the solution space of the problem. According to Eiben et al. (1999), the configuration of parameters directly affects the quality of the final solution, and it is necessary to know what the most promising configuration should be. At this point an optimization problem within the problem being optimized arises that is how to tune such parameters. Eiben et al. (1999) also points out that there are two ways to adjust parameters.

The first one known as off-line parameter control happens before the execution of the algorithm where several tests with different definition of values are performed. The aim is to find *a priori* the best set of values for the parameters. With this preliminary analysis standard values for the parameters can be recommended for future executions. However, such recommendations should not be generalized to all classes of problems. The off-line parameter control has demonstrated to be a complex combinatorial problem due to the high amount of values that each parameter can assume (Aleti & Moser 2013*a*). A prominent work done by Eiben & Smit (2011) reviews several methods for off-line parameter control and a taxonomy is proposed.

The second way to adjust parameters is known as on-line parameter control where values for the parameters change throughout the algorithm run. The on-line parameter control eliminates the previous parameter values analysis step so the adjustment occurs during the optimization process. Consequently, in the on-line parameter control the user or designer of the optimization algorithm abstains itself from the responsibility of adjusting such parameters. An important feature in on-line parameter control is that the value of a parameter can vary according to the search stage (Hinterding et al. 1997). To adjust parameter values during the optimization process can significantly improve the results obtained and this has motivated the research for methods to self-adjust parameters from different meta-heuristics (Kamrath et al. 2013, Kramer 2010, Tuson & Ross 1998, Aleti & Moser 2013*b*, Leung et al. 2012, Simons & Parmee 2010, Marques & Gomide 2011). This work focuses on the review of techniques for on-line parameter control.

In current literature there are some works devoted to the subject of control of parameters. In De Jong (2007) is presented a historical background concerning this topic. Angeline (1995) and Eiben et al. (1999) present different taxonomies for the adjustment of parameters. In Zhang et al. (2012) a review of adjustment methods and a another taxonomy is presented. Kramer (2010) presents a review of parameter control methods with emphasis on self-tuned control. Also, the work of Aleti & Moser (2013*b*) brings a review of feedback mechanisms for the use in on-line parameter control. A detailed review of methods for the adjustment of parameters can also be found in Rezaee Jordehi & Jasni (2013) with focus on SI, and in Karafotias et al. (2015) and Aleti & Moser (2016) with focus on EC. Despite several classification and reviewing initiatives of parameter control methods, these studies did not consider both areas of EC and SI. This article aims to identify and classify the most used techniques for on-line control of parameters, as well as to describe the main characteristics of each one.

This review points out for the importance of theoretical and convergence analysis of on-line control techniques. Also, the design of parameterless algorithms is ideal for the user point of view and on-line control of parameters contributes in this direction.

The research covers a large amount of papers that use different control techniques, domains of application, and bio-inspired algorithms, providing a broad overview of this topic. The search engines used to find related work were the well known IEEE XPlore™, ScienceDirect®, Scopus™, and SpringerLink.

The employed keywords were *parameter control*, *adjust* and *adaptive*, combined with the name of each optimization algorithm in the field of EC and SI. The appearance of a paper in this scope in any of the aforementioned search engines is used as inclusion criterion for the specific EC or SI algorithm to be described in this review. The research period for each optimization algorithm covers the year of its original publication until 20th March of 2018. Hence, this survey is composed by the review of 158 articles.

## 2 Bio-inspired Algorithms

It is possible to classify the bio-inspired algorithms into different groups based on their source of inspiration (Fister Jr. et al. 2013). Algorithms inspired by physiochemical systems were designed to mimic the behavior and characteristics of certain laws of physics or chemistry, including gravity, electric charges, and pluvial systems. Algorithms based on biological systems have its source of inspiration originated from Biology. In this category we have Artificial Neural Networks (McCulloch & Pitts 1943), Artificial Immune Systems (Dasgupta et al. 2011), Evolutionary Computation (De Jong 2006) and Swarm Intelligence (Parpinelli & Lopes 2011), being the last two the focus of this work due to its variety of optimization algorithms.

A common feature present in EC and SI algorithms is that they are population-based approaches. In a population-based algorithm the optimization process takes place in a set of candidate solutions at each iteration. This set of solutions can be called population, swarm, school, or hive, depending on the biological inspiration employed, and each corresponding candidate solution can be an individual, a particle, a bee, or an ant. Although all these algorithms are very similar, they differ in the mechanism or criterion for selecting solutions and in the way they modify (or build) a candidate solution by applying intensification and diversification procedures. Hence, a general pseudo-code (or framework) for population-based algorithms is shown in Algorithm 1 (Benitez et al. 2012, Fister et al. 2016). In line 1 of the algorithm, a population of candidate solutions is randomly initialized, and in line 2 this initial population is evaluated according to the problem being solved. The main loop (between lines 3 to 9) represents the generations (or iterations) of the algorithm, and line 4 defines the mechanism for selecting solutions for next step (i.e., survival of the fittest in Genetic Algorithms (GA), define which solutions will guide the search in Particle Swarm Optimization (PSO), or define which solutions will update pheromone in Ant Colony Optimization (ACO)). Two important characteristics of population-based algorithms, and also for meta-heuristics in general, are the intensification and diversification procedures used to produce new candidate solutions. In line 5 of the algorithm, intensification intends to search locally the search space (i.e., a crossover procedure in GA, the cognitive component in PSO, or the heuristic component in ACO), whereas diversification leads the algorithm to explore globally the search space (i.e., a mutation procedure in GA, the social component in PSO, or the pheromone component in ACO). In line 6 the new pool of candidate solutions is evaluated. In line 7 a replacement criterion can be applied and the main loop restarts until a termination condition is met. It is also common to these algorithms identify the current best solution at each iteration as pointed in line 8. Finally, line 10 outputs the best solution found in the optimization process.

**Algorithm 1:** General pseudocode of a population-based algorithm

```
1  Initialize the population with random candidate solutions;
2  Evaluate each candidate solution;
3  while termination condition is not satisfied do
4  |   Perform competitive selection;
5  |   Apply intensification and diversification procedures;
6  |   Evaluate the new pool of candidate solutions;
7  |   Apply replacement criterion to form the new
   |     population;
8  |   Find current best solution;
9  end
10 Output overall best solution;
```

The bio-inspired algorithms found in the reviewed papers are summarized in Table 1. The first column identifies the algorithm along with its main reference, the second column points out their main inspiration, and third column shows the parameters that need to be adjusted. From Table 1, it is possible to notice that all optimization algorithms have parameters to be tunned highlighting the importance for on-line parameter control. Also, the amount of parameters can vary from a few like in the Firefly Algorithm (Yang 2010*a*), and the Artificial Bee Colony Optimization algorithm (Karaboga & Akay 2009) with 2 parameters each, to many parameters as in the Bacterial Foraging Optimization algorithm (Passino 2002), and in the Gene Expression Programming (Ferreira 2001) with 7 and 11 parameters, respectively.

## 3 Taxonomy for Control of Parameters

In Angeline (1995) a taxonomy was proposed for parameter control based in the type of update rule and the level where it operates. According to this taxonomy, the update rule can be classified as absolute or empirical. Absolute rules are predetermined and specify how modifications occur. Empirical rules define a specific function for the control and allows the algorithm itself to determine whether the changes in the parameters are advantageous or not. This rule is called self-adapted (or aggregated) when the parameters are coded inside the solution and the algorithm evolves their respective values.

The adaptation can be accomplished at three different levels: population, individual or component. At the population level, there are techniques that dynamically adapts global parameters, e.g. the crossover probability in GA. The individual level contains the techniques which controls parameters that operates over a single individual, e.g. a mutation rate exclusive to a population member as in ES. At the component level, parameter values are associated for each component

| Algorithm | Inspiration | Parameters |
|---|---|---|
| Differential Evolution (DE) Storn & Price (1997) | Evolution Theory | POP,CR,F |
| Evolution Strategy (ES) Beyer & Schwefel (2002) | Evolution Theory | $\mu$, $\lambda$, $\delta$, $\rho$ |
| Gene Expression Programming (GEP) Ferreira (2001) | Evolution Theory | POP, H, G, MUT, IR, IS, RIS, OP, TP, GR, GT |
| Genetic Algorithms (GA) Goldberg (1989) | Evolution Theory | POP,CR,MUT,K |
| Genetic Programming (GP) Koza (1992) | Evolution Theory | POP, CR, MUT, K, R, S |
| Ant Colony Optimization (ACO) Dorigo & Di Caro (1999) | Ants foraging | POP, $\alpha$, $\beta$, $\rho$ |
| Artificial Bee Colony Algorithm (ABC) Karaboga & Akay (2009) | Bees foraging | POP,limit |
| Artificial Fish School Algorithm (AFSA) X. Li (2002) | Fish school | POP, Trynumber, Visual, $\delta$ |
| Bacterial Foraging Optimization (BFO) Passino (2002) | The bacteria *Escherichia coli* foraging | POP,$N_c$,$N_s$,$N_{re}$,$N_{ed}$,$p_{ed}$, C(i) |
| Bat Algorithm (BA) Yang (2010*b*) | Bats echo-localization | POP, $\alpha$, $\gamma$ |
| Cuckoo Search algorithm (CSA) Yang & Deb (2010) | Breeding behavior of cuckoos | POP, $\alpha$, $p_a$ |
| Firefly Algorithm (FA) Yang (2010*a*) | Fireflies luminosity model | POP, $\gamma$ |
| Gravitational Search Algorithm (GSA) Rashedi et al. (2009) | Gravitational and movement laws | POP, $G_0$, $\alpha$ |
| Particle Swarm Optimization (PSO) Kennedy & Eberhart (1995) | Coordinated movement of birds and fish school | POP,$\omega$, $C_1$, $C_2$ |

**Table 1** Bio-inspired Algorithms reviewed in this work: their inspiration and respective parameters.

during an individual's evolution, which determines how each component will be modified during reproduction process. This level of adaptation is distinguished by specifying how to manipulate a single component independent of the others. The Gaussian mutation variation is an example.

Eiben et al. (1999) proposed a taxonomy where the adjustment of parameters is divided in two main categories: off-line and on-line. Off-line parameter control is characterized by strategies that uses parameters with static values and previously adjusted. The off-line category consists of manual adjustment, adjustment by experiments planning, and adjustment through meta-evolution (Kramer 2010). Manual adjustment depends on the user experience, whom modifies the parameter values before each run. In the experiments planning, a group of parameters are tested previously and through an analysis of the results obtained it is defined the best configuration to be used. The meta-evolution consists in applying an algorithm to evolve the parameter values. On-line parameter control changes the parameter values during the algorithm run. This control is divided in three different categories: deterministic, adaptive, and aggregated. Deterministic control is accomplished by deterministic rules that change the parameter values while the algorithm is running, without the use of any kind of inherent information about the search process. Adaptive control uses inherent information about the search process to determine the direction or magnitude of the parameter value to be changed. Aggregated control performs a search in the space of parameter values, where the parameter itself is coded within the solution vector, modifying itself during the optimization process.

The taxonomy proposed by Zhang et al. (2012) contains three aspects: the adaptation object; the adaptation evidence; and the adaptation method. This taxonomy was proposed considering only the field of evolutionary algorithms. The adaptation object identifies which algorithm components will be adapted and is divided in four categories: the parameter control (which is the focus of this work), the evolutionary operators, the population structure and others. The adaptation evidence contains the information that the adaptation can use, e.g., deterministic factors, the adaptability value, the population distribution, and the combination of the adaptability value with the population distribution. The adaptation method can be based in different strategies such as simple rules, coevolution, fuzzy control, or entropy, for example. The taxonomy proposed by Angeline (1995) takes into the account component level where the parameter is adjusted. However, it does not present a way to classify the techniques by the adaptation mechanism, which is the most important characteristic. Eiben et al. (1999) taxonomy provides a global overview about the parameter adjustment in a simple way. With this taxonomy it is easy to identify different types of methods that are applied in on-line and off-line control for bio-inspired algorithms. However, this taxonomy does not present a way to classify the methods by their adaptation mechanisms. Zhang et al. (2012) proposed a taxonomy that allows classification by the adaptation mechanism, component, and adaptation evidence. This taxonomy can be seen as an extension of Eiben's taxonomy, where the on-line parameter control is divided in two categories, mainly by the adaptation mechanism.

Hence, Eiben's taxonomy was adapted adding some details in the adaptation mechanisms provided by Zhang's taxonomy. Therefore, it is possible to identify the parameter adjustment methods applied in all meta-heuristics correlated in this work. Figure 1 presents the complete taxonomy used in this work to classify different types of parameter adjustment mechanisms.

## 4 Techniques for Parameter Control

This section describes the main parameter control techniques used by the works found during the literature review. It is well known that the choice of correct

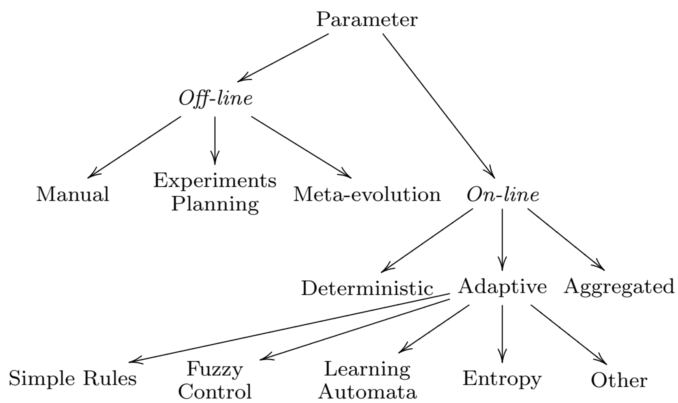


**Figure 1** Taxonomy for Parameter Control. Adapted from Eiben (1999) and Zhang (2012).

parameter control techniques has a direct consequence in the bio-inspired algorithm performance (Eiben et al. 1999). However, independently of the chosen control technique, such use is justified only if it leads to a better performance in comparison with the original algorithm. There is no rule that limits or determines the use of specific groups of techniques. The parameter control techniques found during the literature review range from simple deterministic rules to robust methods like clustering and learning automata, and are described next.

Deterministic parameter control occurs when a deterministic rule changes the parameter value. The rule modifies the parameter value without using any kind of information from the fitness function. Generally, a pre-defined function is used to change the parameter value over time, e.g. the parameter value starts with a big value that gradually decreases during the optimization process. It is common to use the number of generations (iterations) or the number of fitness evaluations to perform the control. Due the simplicity and low computational effort, a lot of applications use this type of control. This control technique hardly takes the bio-inspired algorithm behavior into account during the optimization (Zhang et al. 2012).

In aggregate control, the parameters are directly coded in the solution vector as extra dimensions and are optimized during the optimization process. The parameter optimization can be done through the same routines used to optimize the population or through specific routines (Zhang et al. 2012). In this approach, following the natural selection analogy used in Evolutionary Computation, the best coded parameter values that lead to better solutions will have more chance to propagate their values to the next generations (Eiben et al. 1999). Also, using the same methodology, Swarm Intelligence algorithms can optimize their parameter values.

In simple rules strategy, the control is obtained through simple rules that are defined by the observation of characteristics and behavior of the algorithm during its run (Zhang et al. 2012), e.g. the definition of a rule based on an exponential or linear function. This type of approach uses information from the optimization process to modify the parameter values. The feedback information can be the fitness or objective function itself, some population diversity measurement, and others.

Clustering can be used to find patterns among the parameter values and the feedback obtained in the optimization process. The clustering is one of the fundamental tasks in data mining and it is used to identify patterns in a non-supervised manner (Jain et al. 1999). The objective of clustering is to group objects that have some similarity according to a predefined metric (Rokach 2010). The similarity of an object in each group can be computed through some distance measurement, e.g the Euclidean Distance. There are different methods for clustering: Hierarchical; Partitioned; Density based; Model based; Grid based; and many others (Berkhin 2006).

A learning automata can be used to select the parameter values through a learning process according to the feedback obtained in the optimization. Learning Automata are mechanisms for decision-making that can act in stochastic environments. The objective is to progressively upgrade its performance through a learning process (Obaidat et al. 2002). Learning is defined as any permanent change in the behavior of the automata and it is resultant of its previous experiences. Hence, the main characteristic of a learning automata is its capacity to improve the performance over time.

The fuzzy control is a parameter control technique which modifies the parameter values using degrees of relevance (Zhang et al. 2012). The fuzzy logic was proposed by Zadeh in 1965 with the objective to insert degrees of uncertainty in the traditional Boolean logic, providing human inferences in concepts and knowledge that are not well-defined for machines. The basic components of a fuzzy controller are the base of knowledge, the fuzzy rules, and the inference mechanism (Jantzen 2007). In the context of parameter control, the inference mechanism is used to evaluate the actual state of the system. The rules are proposed according to the identification state and are used to control the parameter values.

Entropy-based strategies can be used for parameters control. The entropy term was proposed in Information Theory by (Shannon 2001). Entropy can be defined as a measure of expected information or uncertainty of a probability distribution. Also, it is defined as a disorder degree in a system or uncertainty of a partition (Okafor 2005). In bio-inspired algorithms, the generation of new possible solutions usually involves random and probabilistic factors. The entropy can be used to analyze the population state and characteristics in such a way that the parameters can be adjusted in accordance with this information (Zhang et al. 2012).

In probability theory the covariance is a measurement of statistical dependency between two random variables. If two variables are independent they have zero covariance. The covariance matrix is a symmetric matrix which summarizes the covariance among different variables. This technique can be used to learn the dependency among parameters and increase the probability to repeat successful steps (Hansen & Ostermeier 2001).

Another strategy for parameter control is the use of pheromone matrix. The pheromone matrix is inspired by the chemical substance released by ants (Dorigo & Di Caro 1999). This substance is deposited in the paths taken by ants, providing a positive reinforcement in it. The pheromone is attractive to other ants and a trail with more pheromone attracts more ants. The matrix is also used as a concept of negative reinforcement with an analogy with the pheromone evaporation. The pheromone matrix can be used to find better values for parameters. An important characteristic of this technique is the cooperation, thus allowing the usage of innumerable agents to construct the matrix.

## 5 On-line Control of Parameters in Bio-inspired Algorithms

This section presents a review of the works found in the literature using on-line parameter control techniques applied in bio-inspired algorithms. Although this work is not an exhaustive literature review, the research covers a large amount of papers that use different control techniques, domains of application, and bio-inspired algorithms. Hence, providing a broad overview of this topic in a total of 258 control parameters scattered in 158 reviewed articles.

The main objectives of this section are to identify which parameters were adjusted and to classify the strategies used according to the taxonomy presented in Section 3. In order to perform the classification, a specific nomenclature to discriminate the control techniques was defined: deterministic (CT-DT), aggregated (CT-AG), simple rules (CT-SR), learning automata (CT-LA), Fuzzy control (CT-FC), clustering (CT-CL), entropy-based (CT-ET), covariance matrix (CT-CM), and pheromone matrix (CT-PM). For example, the acronym CT-DT:2 represents a deterministic control technique that was applied in 2 different parameters. All acronyms different from CT-DT or CT-AG are considered to be adaptive in its specific category.

Applying the Differential Evolution algorithm, Huang. & Chen (2013) presents a method using sine and cosine functions to adapt the crossover probability and the mutation factor in a deterministic way (CT-DT:2). Zhang et al. (2013) presents a strategy to adapt the crossover probability using the number of iterations to tune the parameter value (CT-DT:1). Brest et al. (2006) presents a strategy that controls the crossover and mutation parameters encoded in the individual and they are tuned through a simple rule (CT-AG:2). Teo (2006) presents two models of aggregated control to tune the population size. Also, in these two models an aggregated control method for mutation and crossover parameters is applied. In the first model, the population size is encoded in the individual (CT-AG:3). In the second model, the population growth rate is encoded in the individual, where a positive value increases and a negative value decreases the population size (CT-AG:3). Dragoi et al. (2013) presents an aggregated control method to mutation and crossover parameters and are modified with the same operators applied in the individuals (CT-AG:2). Wang & Zhao (2013) presents a DE with population resizing defined by a simple rule based on the evolution of the best individual fitness. (CT-SR:1). Kovaevi et al. (2014) presents a control method for the mutation factor. A set of values is predefined to the parameter and a probability is set according to its success rate (CT-SR:1). Piotrowski (2013) presents a control method for the crossover parameter tuned according to its success in generating better solutions (CT-SR:1). Zhong et al. (2014) presents a method to adapt the mutation probability based on fitness (CT-SR:1). Ali & Törn (2004) proposed an adapted rule based on the maximum and minimum fitness of individuals to control the mutation factor (CT-SR:1). Meng & Pan (2016) proposed a control scheme for crossover and mutation routines. The crossover parameter is implemented through an automatically generated crossover matrix guided by a simple rule, while mutation is tuned using a simple rule based on the success of crossover and mutation probabilities (CT-SR:2). Chen et al. (2014) presents a strategy to tune the population size performed with simple rules based on the population fitness (CT-SR:1). Tanabe & Fukunaga (2013) presents a method to tune the crossover parameter and the mutation factor $F$ based on previous fitness values (CT-SR:2). Mallipeddi & Suganthan (2010) proposed a set of strategies for mutation and crossover associated to a set of values for each parameter. If this combination generates a better individual then it is stored, otherwise it is randomly restarted with a new strategy and parameter values or with a stored successful combination with the same probability (CT-SR:2). In Wang et al. (2018) an adaptation method for the parameters $F$ and $C_r$ are presented. The method consists in generating a set of random parameters and using them for a predefined number of generations, where the successful ones are recorded in a list (CT-SR:2). A simple method of parameter control is proposed in Shen et al. (2017), where the parameters for $F$ and $C_r$ are randomly generated for each individual and kept while they can improve the fitness. Once the fitness is not improved for an individual, $F$ is randomly generated in an interval using a uniform distribution and $C_r$ using a Gaussian distribution (CT-SR:2). In Dawar & Ludwig (2018) parameters $F$ and $C_r$ are updated using the successful mutation strategies. Also, a reset is performed for updating the memory of successful $F$ and $C_r$ values (CT-SR:2). Liu & Lampinen (2005) presents a DE algorithm with an adaptation control for the crossover parameter and mutation factor using fuzzy logic. The input is the difference between the values of the objective function and the population individuals (CT-FC:2). Kotinis (2014) presents a fuzzy control method for the crossover parameter and mutation factor. The control is performed based on the quantity of non-explored solutions and the difference of the population distribution between two iterations (CT-FC:2).

A well known strategy for parameter control in Evolution Strategy is the 1/5th rule (Rechenberg 1973). This rule establishes that the proportion of success mutations for all mutations should be 1/5. Therefore, if the proportion is superior than 1/5, the mutation step size should be increased and, if the proportion is inferior than 1/5, the mutation step size should be decreased (CT-SR:1). Beyer (1995) presents an aggregated strategy parameter control for mutation variance. The parameter is updated according to specific functions based on individual fitness (CT-AG:1). The fundamental mechanism of CMA-ES is the adaptation of the covariance matrix (CMA). This mechanism tunes the mutation step size. The CMA is based on the observation that the distribution model should be updated in a cumulative manner. The information of the evolution path is also used to adapt the covariance matrix in Zhang et al. (2012) (CT-CM:1).

Applying Gene Expression Programming, Bautu et al. (2007) adapts the number of genes ($G$) of an organism. The adaptation process takes place at chromosome level, allowing chromosomes in the population to evolve with different number of genes (CT-AG:1). Based on the comparison between the fitness of the parent organism and child organism in previous generations, Mwaura & Keedwell (2009) implements simple rules to regulate the probabilities of genetic operators. The parameters controlled are the mutation rate ($MUT$), one-point recombination rate ($OP$), two-point recombination rate ($TP$), gene recombination rate ($GR$), IS transposition rate ($IS$), RIS transposition rate ($RIS$), and the gene transposition rate ($GT$) (CT-SR:7).

Concerning Genetic Algorithms, Boeringer & Werner (2002) presented in his work a simple control method to mutation probability using an uniform distribution (CT-DT:1). Li & Chang (2006) presents a function to adapt the mutation and crossover probability that gradually decreases the values in relation to the number of generations (CT-DT:2). Ding & Wang (2008) presents an adaptive control method for crossover probability considering population maturity, measured through the population average fitness (CT-SR:1). Osaba et al. (2013) presents a GA based on population islands with a control method for crossover probability tuned using simple rules based on the individual with the best fitness value (CT-SR:1). Vafaee & Nelson (2009) presents a control model for mutation probability adapted according to the frequency of the best individual genes using statistics models of DNA sequence evolution (CT-SR:1). Yang et al. (2001) presents a control method for mutation probability tuned through a linear function based on the average population fitness (CT-SR:1). In Yang, Han, Li & Xu (2013) two simple rules are employed to control the crossover and mutation probabilities (CT-SR:2). The average fitness is used to determine the mutation and crossover rates. Yang, Zheng, Yang, Zhou & Liu (2013) presents a control method for the probability of mutation and crossover based on the individual fitness and a population dispersal degree (CT-SR:2). Rajkumar et al. (2016) uses two different simple rules to control the crossover and mutation probabilities, one for each (CT-SR:2) based on the average fitness of the population. In Huang et al. (2016) the mutation and crossover probabilities are adjusted based on a linear scaling function that compares the individual fitness against the population average fitness (CT-SR:2). Aleti & Moser (2011) proposed a method to predict the mutation and crossover parameters using the population average fitness along with the parameter value success rate to tune the probability for the next generations (CT-SR:2). In another work, Aleti et al. (2012) the limits of mutation and crossover values are tuned according to evolution progress (CT-SR:2). Aleti & Moser (2013*a*) presents another work where the control consists in the use of a k-means clustering algorithm based on the quality of mutation and crossover values. For each cluster the entropy is calculated, which is then used to adapt the parameter values (CT-ET:2). Boudjelaba et al. (2014) presents a hybrid GA where the mutation probability is tuned for each individual with a function based on their fitness, the population best, and average fitness (CT-SR:1). Ponz-Tienda et al. (2013) presents a control method for mutation and crossover probability. The values are tuned according to the quantity of invalid individuals that are generated by the operators (CT-SR:2). Rajakumar & George (2013) presents a population size control method based on the average fitness of the population and the number of generations (CT-SR:1). Linda & Nair (2013) presents a simple method for mutation probability control tuned according to the fitness of the generated individuals (CT-SR:1). Chen, Yan & Wang (2013) presents a function to tune the probability of mutation and crossover based on individuals' fitnesses (CT-SR:2). Aleti (2014) presents a method to tune the probability of mutation and crossover of a multi-objective elitist GA. The control method consists in the use of Bayesian probability to tune each parameter based on their respective performance (CT-SR:2). Pan & Chen (2013) presents a control method for population size and mutation probability. The population size is tuned with a simple function based on population fitness. The mutation probability is tuned according to population size (CT-SR:2). Rajappan & Rangasamy (2017) presents adaptive control techniques for parameters CR and MUT based on fitness information. The population is divided into sub-populations of high and low fitness values. To the sub-population with higher fitness local exploitation is applied, thus, only crossover routine is applied and tuned with simple rules. On the other hand, to the sub-population with lower fitness exploration is applied and only the mutation procedure is employed and tuned with simple rules (TC-SR:2) Bi et al. (2016) proposed to tune the crossover and mutation probabilities according to the individuals fitness (TC-SR:2). Algethami & Landa-Silva (2017) presents an adaptive parameter control method for the CR, MUT and POP parameters, based on a genotypic diversity measurement (CT-SR:3). Peng et al. (2014) presents a hybrid GA with Simulated Annealing. A fuzzy control method is used to tune the probability of mutation and crossover based on fitness of the individuals and population (CT-FC:2). Tarokh & Zhang (2014) presents a fuzzy control method for the probability of mutation and crossover parameters done for each generation according to the population fitness diversity (CT-FC:2). In Gudino-Penaloza et al. (2013) a fuzzy controller is applied over a GA to control crossover and mutation parameters (CT-FC:2). In the presented framework the GA is run several times while the Fuzzy Engine is operating. Pereira et al. (2013) proposed to tune mutation probability by a fuzzy controller using the number of iterations as the input and mutation probability as the output (CT-FC:1). Smetek & Trawinski (2011) performed a study with two control models. Each model encodes a different number of parameters. The first control model encodes the probability of mutation and crossover in the individual (CT-AG:2). In the second model, no parameter is encoded inside the individual. The adaptation consists in controlling the population size associating an integer value to each individual corresponding to its age, and through the relation of its fitness with the population this value is updated. When the individual age is less than or equal zero the individual is removed from the population (CT-SR:1). Fernandez-Prieto et al. (2011) presents different ways to adapt the mutation probability. The first way, proposed by Back & Schutz (1996), specifies a linear function that decreases the mutation value according to the number of generations (CT-DT:1). The second approach consists in the use of random values for the mutation parameter in each generation (CT-DT:1). The third one, encodes the parameter inside the individual, adding and extra dimension. A specific operator is applied to the parameter, generating a new value that is used to apply mutation on the individual (CT-AG:1). In the forth approach, proposed by Srinivas & Patnaik (1994), each individual has a value assigned, which is calculated according to its fitness (CT-SR:1). The fifth and last approach refers to fuzzy logic, performing the adaptation through the fitness convergence measure with a set of fuzzy language labels (low, medium and high) associated to different values of mutation probability (CT-FC:1). Several approaches to adapt the tournament size ($K$) is presented in Vajda et al. (2008). The parameter is tuned linearly based on the generation number (CT-DT:1). An aggregated control method is also presented where the tournament size is encoded inside the individual as an extra dimension and $K \in [0, 1]$ (CT-AG:1). A hybrid model of aggregated control is obtained through a function that tunes the value of the parameter $K$ if its fitness is better or worse than its parent (CT-SR:1). A fuzzy tournament model is also presented, where the adapted parameter is calculated through fuzzy rules using the genotypic and phenotypic diversities (CT-FC:1).

Applying parameter control in Genetic Programming, Al-Madi & Ludwig (2012) uses simple rules to tune the crossover and mutation parameters (CT-SR:2). Fitzgerald & Ryan (2013) proposed individualized genetic operators for controlling and setting crossover and mutation search parameters based on their fitness success rate, combined with adaptive tournament size (CT-SR:3). Yu et al. (2013) applies an adaptive strategy to the crossover and mutation parameters using simple rules composed by the diversity of the population and the standard deviation of fitness (CT-SR:2). Le Hai Nam (2015) adapts mutation and crossover rates based on the relative fitness of the individual weighted by population progress in the search process (CT-SR:2). Kalkreuth et al. (2015) uses a population diversity measure to adapt the crossover and mutation parameters using simple rules (CT-SR:2). Gregor & Spalek (2016) uses simple rules to tune the crossover and mutation parameters (CT-SR:2). Riley et al. (2016) uses a probability vector based on fitness to select sub-trees for mutation (CT-SR:1). The mutation and crossover probabilities are controlled based on the mean and minimum fitness of the population in Singh & Sharan (2017)(CT-SR:2). About the use of parameter control in Ant Colony Optimization, Merkle et al. (2002) proposed two deterministic control methods for the parameters $\beta$ and $\rho$, representing the attractiveness of the pheromone path and evaporation rate, respectively. For the parameter $\beta$, the value decreases over the iterations while the parameter $\rho$ increases according to the iterations (CT-DT:2). Meyer (2004) presents a deterministic control method for the parameter $\alpha$. The technique uses a function called $\alpha$-*annealing*, defined by the author (CT-DT:1). Cai & Huang (2008) proposed a method where each individual is assigned to a value for the parameter $\rho$. The parameter control is based on the solution quality (CT-SR:1). Chusanapiputt et al. (2006) proposed an ACO algorithm with two distinct populations, where the individuals migrate between populations. The parameters $\alpha$ and $\beta$ are tuned according to population size (CT-SR:2). The $\alpha$ and $\rho$ parameters are controlled in Yuan et al. (2017)(CT-SR:2) in which the parameter control routine is only employed if the optimization process does not improve the solution within a predetermined number of iterations. Li (2009) proposed a control method based on entropy. The method calculates the average entropy and, through a simple rule, the parameters $\alpha$ and $\beta$ are tuned (CT-ET:2). Li & Li (2013) follows the same idea from the previous work, however presents a different rule to adapt the parameters $\alpha$ and $\beta$ (CT-ET:2). Castillo et al. (2013) presents a fuzzy control method for the evaporation rate $\rho$. The control uses an error measure to tune the parameter values (CT-FC:2). Khichane et al. (2009) proposed a control method for the parameters $\alpha$ and $\beta$, based on pheromone matrix. The parameters adaptation occurs in iteration cycles (CT-PM:2). In Mavrovouniotis & Yang (2014) the $\rho$ parameter is controlled by predefining discrete intervals values for $\rho$ and using another pheromone map to adapt it (CT-PM:1).

Applying the Artificial Bee Colony Algorithm, Aydin et al. (2014) presents a method to tune the population size. The probability to add or remove individuals from the population is controlled by a variable that is tuned according to the algorithm success rate (CT-SR:1). Cui et al. (2017) also presents a method to tune population-size (POP). The smaller the success rate that an individual has, the larger is its probability of being removed. In contrast, if the bee colony successfully obtains better food source positions, the colony will be expanded by adding a new individual at random (CT-SR:1). In Yavuz et al. (2016) the population size is controlled using simple rules (TC-SR:1). The population size starts small and increases as the objective function gets better. The limit parameter is self-adapted in Jiang et al. (2016) based on fitness information. Three types of search are introduced, i.e. shallow search, medium search and deep search. However, two new parameters need to be adjusted: the degree of generations with no improvements ($af$) and the degree of improvement ($di$) (CT-SR:1).

Using parameter control techniques in the Artificial Fish School Algorithm, Hongrui et al. (2009) presents a control method for the $\delta$ parameter through a deterministic function (CT-DT:1). Yazdani, Nabizadeh, Kosari & Toosi (2011) presents a deterministic control method for the parameter *visual* using a random component as a function of the number of iterations (CT-DT:1). Yong (2011) presents a deterministic control method for the parameter *visual*. The parameter is controlled from a variable $\alpha$ that is tuned according to a exponential function which considers the number of iterations (CT-DT:1). He et al. (2009) presents two deterministic methods to tune the parameter *visual*. The first method is based on the number of iterations and it operates by gradually decreasing the value until reaches a predefined minimum (CT-DT:1). The second method is also based on the number of iterations with the addition of a constant $\beta$ to gradually decrease the parameter value (CT-DT:1). Jiye et al. (2013) proposed an adaptable method to control the parameter *visual*. The control is performed with a simple rule based on the individual fitness and the average population fitness (CT-SR:1). Tian & Tian (2009) presents a method that uses the fitness value to adapt the parameter *visual*. The function proposed decreases the parameter value over the iterations (CT-SR:1). Yazdani, Nadjaran Toosi & Meybodi (2011) proposed two fuzzy models to control the parameter *visual*. The first model, called Fuzzy Uniform Fish, uses as input the number of iterations and the ratio of individuals with the best current fitness (CT-FC:1). The second model, called Fuzzy Autonomous Fish, uses as input the distance of the best individual, the fitness ranking and the number of iterations (CT-FC:1).

Concerning the Bacterial Foraging Optimization algorithm, Dasgupta et al. (2009) presents a control method for the step size parameter $C$. The parameter values are adapted through a deterministic function (CT-DT:1). Chen et al. (2011) presents two control methods for the step size parameter $C$. In the first method, the parameter values are updated in each period, declining when the fitness of the best individual improves, otherwise it does not change. The parameter value is restarted at the end of a period of $t$ iterations (CT-SR:1). In the second model, an aggregated control method with a specific operator is used to control the parameter $C$ (CT-AG:1). In their study, the first model achieved better results. Farhat & El-Hawary (2010) presents a deterministic method for a linear decay of parameter $C$ (CT-DT:1). Panigrahi & Pandi (2009) proposed a simple function for deterministic control of the parameter $C$ (CT-DT:1). Rashtchi et al. (2009) presents a control method for the parameter $C$ tuned deterministically through exponential functions (CT-DT:1). Yan, Zhu, Zhang, Chen & Niu (2012) proposed a control method based on the number of objective function evaluations. The method is deterministic and generates a gradually decay of the parameter $C$ (CT-DT:1). Nasir et al. (2015) presents two parameter control methods for the bacteria step size ($C(i)$). The first uses a deterministic rule based on the number of elapsed generations (CT-DT:1). The second approach evolves a simple rule that uses the bacteria fitness and the number of elapsed generations (CT-SR:1). The authors report that both approaches outperformed the standard BFO, where the first approach was able to achieve better accuracy and the second approach was more consistent. Xu & Chen (2014) presents a function based on fitness values to tune the parameter $C$ (CT-SR:1). Majhi et al. (2009) presents a function to gradually decay the parameter $C$ based on the individual fitness (CT-SR:1). Tan et al. (2015) proposed a method to tune the step size ($C$) based on a non-linearly decreasing modulation model (CT-SR:1). Verma et al. (2016) presents a dynamic adaptive control for parameter $C$ based on fitness function. In this work, the algorithm modifies the parameter $C$ depending on how close it is to an optima (CT-SR:1). Datta et al. (2008) presents a control method to the parameter $C$ that is tuned to each individual based on their fitness (CT-SR:1). Sathya & Kayalvizhi (2011) proposed a method based on the individual fitness to control the parameter $C$ (CT-SR:1). Mezura Montes & López-Dávila (2012) proposed a method to control the parameter $C$ according to its success rate (CT-SR:1). In Zeng et al. (2017) a method to control the step size $C$ and the elimination probability $P_{ed}$ is presented. The method for controlling $C_i$ consists on a deterministic rule which decreases over time (CT-DT:1). The $P_{ed}$ parameter, on the other hand, is controlled with a simple rule based on the population fitness. Mishra (2005) proposed a fuzzy control method to adapt the parameter $C$ based on the fitness of the best individual of the population (CT-FC:1). Venkaiah & Kumar (2011) proposed a fuzzy control method to adjust the parameter $C$. The method uses as input the objective function error and the current value of the parameter $C$ (CT-FC:1).

Applying the Bat Algorithm, Yılmaz et al. (2014) presents a deterministic method to control the amplitude parameter ($\alpha$) and pulse emission ($\gamma$). For each individual dimensions a value is assigned for the amplitude and pulse emission. The values are tuned with simple functions based on the number of iterations (CT-DT:2). Chen, Zhou & LU (2013) proposed an adaptation method for amplitude ($\alpha$) and pulse emission ($\gamma$). The amplitude is tuned with a simple function based on the number of iterations (CT-DT:1) and the pulse emission is tuned with a simple function based on the amplitude value (CT-DT:1). A method using the Rechenberg $\frac{1}{5}$ rule is presented in Kabir et al. (2014) where the authors adapt the parameters $\alpha$ and $\gamma$ by controlling the pulse rate (CT-SR:2). In Fister Jr et al. (2014) a method to adapt the parameters $\alpha$ and $\gamma$ is employed (CT-SR:2). The method is based on making random perturbations with a predefined chance of occurrence. In Dhal & Das (2018) the authors utilized a simple rule to control when the parameters $\alpha$ and $\gamma$ are applied (CT-SR:2). While the proposed parameter control technique does not directly alter parameter values, it has the same effect of altering the parameters to a fixed point value (a value that does not change the state of the system), therefore it can be considered as a parameter control technique.

Concerning the Cuckoo Search Algorithm, a study on the effect of different deterministic controls for the discover probability ($p_a$) is presented in Mareli & Twala (2017) (CT-DT:1). Naik et al. (2015) presents a simple rule to control the step size ($\alpha$) of CSA (CT-SR:1). The rule works by measuring the difference between each nest fitness with the best and worst fitness. The number of elapsed iterations is then used to determine how much the rule can affect the parameter. In Naik & Panda (2016) the parameter $\alpha$ is controlled proportionally to the fitness of the individual in the current generation (CT-SR:1). Pauline et al. (2017) proposes an adaptive control method for parameter $\alpha$ based on global and local fitness values (CT-SR:1). In Chi et al. (2016) two simple rules are proposed, one for controlling the step size ($\alpha$) and the other for controlling the discovery probability ($p_a$) (CT-SR:2). The step-size $\alpha$ and discovery probability $p_a$ are controlled using simple rules and are presented in Jaballah & Meddeb (2017) (CT-SR:2). The parameter $\alpha$ is controlled in function of the number of iterations, the fitness of each solution and the average fitness of the population. For the parameter $p_a$, the ranking of the solutions and the distances are used. In Liu et al. (2018) is presented a simple rule for parameter $p_a$ and a deterministic control for parameter $\alpha$ (CT-DT:1) (CT-SR:1). The parameter $\alpha$ is controlled in function of the number of iterations and the parameter $p_a$ uses feedback from current solution and it is kept between two thresholds. The development on the modifications of the cuckoo search algorithm can be found in Chiroma et al. (2017) with a dedicated section on parameter control.

Applying the Firefly Algorithm, Abshouri et al. (2011) presents a hybrid FA with a learning automata. The automata is responsible for the absorption coefficient ($\gamma$) and uses only the number of iterations as input (CT-DT:1). Yan, Zhu, Wu & Chen (2012) presents a FA with an adapted absorption coefficient ($\gamma$). The parameter $\gamma$ is controlled deterministically decreasing its value based on the quantity of problem dimensions (CT-DT:1). Another use of deterministic control is employed in Wang et al. (2017), where the parameter $\alpha$ is decreased non linearly over time (CT-DT:1). Fister et al. (2013) presents a FA with aggregated parameter control. Each individual is encoded with its respective light absorption coefficient value ($\gamma$). For each parameter a specific strategy of mutation is defined (CT-AG:1). Niknam et al. (2012) proposed an aggregated model to control the parameter $\gamma$. The parameter value is encoded together with the individual (CT-AG:1). Selvarasu & Rajan (2013) presents an aggregated control method for the parameter $\gamma$. To tune the parameter it is used the same operators applied to the solutions (CT-AG:1). Roy et al. (2013) presents a strategy to control the parameter $\gamma$. A set of values is used and the respective probabilities of each parameter are tuned based on their performance in generate new individuals (CT-SR:1). In Cheung et al. (2014) the light absorption coefficient ($\gamma$) is controlled by using a simple rule based on the ratio of the best, the closest and the furthest fireflies (CT-SR:1). In Hassanzadeh et al. (2017) a method using a fuzzy technique is employed to control the $\alpha$ and $\gamma$ parameters (CT-FC:2). The proposal consists in adding an adaptive weight to update both parameters based on its previous value. This weight is controlled directly by the fuzzy engine.

Concerning the Gravitational Search Algorithm, Sombra et al. (2013) proposed a control method with fuzzy logic based on the quantity of iterations to control the parameter $\alpha$. This strategy does not use information derived from the search evolution (CT-DT:1). Precup et al. (2012) presents a deterministic method using an exponential function to scale the values for the gravitational constant $G$ (CT-DT:1). In another work, Precup et al. (2013) presents a fuzzy control method to adapt the gravitational constant $G$. The control uses only the number of iterations as input information to tune the parameter (CT-DT:1). Niknam et al. (2013) proposed an adaptable control method for the gravitational constant $G$. In each iteration the value of $G$ is generated randomly, respecting the maximum and minimum limits (CT-DT:1). Ji et al. (2017) proposed a method for controlling the parameters $G_0$ and $\alpha$ (CT-SR:2). To control $G_0$, a method based on the median distance between each particle is considered. For the parameter $\alpha$, each particle has its own value which is updated based on a rule which takes into account the mean value of the $\alpha$ parameter for all particles and the success rate of this parameter. Based on a calculated mean of $\alpha$, a new value is defined with a small Gaussian perturbation. Zahiri (2012) presents a fuzzy control method for the gravitational constant $G$. The control uses the best individual fitness, the number of iterations that the individual has not improved and population fitness variance to tune the parameter (CT-FC:1) Askari & Zahiri (2012) proposed a GSA with fuzzy control to tune the gravitational constant $G$. As input it is used the best fitness found in the iteration $T$, the number of iterations that the best fitness has not been improved and the fitness variance in the iteration $T$. With this information the parameter is tuned (CT-FC:1). Kumar et al. (2013) presents a fuzzy control method for the GSA algorithm. In their proposal is used the normalized fitness value of the individual and the current value of the gravitational constant $G$, and as output we have the adapted gravitational constant (CT-FC:1). Saeidi-Khabisi & Rashedi (2012) proposed a fuzzy control method for the parameter $\alpha$. The control is performed using the population diversity measure, a population progress measure and the number of iterations to increase or decrease the parameter values (CT-FC:1).

The application of on-line control techniques using the Particle Swarm Optimization algorithm is described next. Yasuda et al. (2010) presents a simple adaptation mechanism for the parameters $\omega$, $C1$ and $C2$ using linear functions (CT-DT:3). Khadhraoui et al. (2016) proposes a control method based on the number of iterations for parameters $\omega$, $C1$ and $C2$ (CT-DT:3). Niknam & Azad Farsani (2010) uses a linear function to adapt the parameter $\omega$ (CT-DT:1). Shi & Eberhart (1999) presents a deterministic method with a linear decrease of particles inertia weight based on the number of iterations (CT-DT:1). Jiao et al. (2008) presents a deterministic method with a non-linear function to adapt the inertia weight (CT-DT:1). Following the same idea of previous work by using non-linear functions, there are the works of Alfi (2012) and Chatterjee & Siarry (2006) (CT-DT:2). A random and simple function to adapt the inertia weight is proposed by Eberhart & Shi (2001) (CT-DT:1). Wang (2011) tunes the parameters $\omega$, $C1$ and $C2$. The parameter $\omega$ is controlled through a linear decrease in function of the number of iterations (CT-DT:1), $C1$ is tuned using the information of the best (*pbest*) and global (*gbest*) particles (CT-SR:1) and the parameter $C2$ is controlled increasing its value through a liner function based on the number of iterations (CT-DT:1). In Montalvo, Izquierdo, Prez-Garca & Herrera (2010) the parameter $\omega$ is tuned aggregated with the individual (CT-AG:1). Yang et al. (2007) presents an approach to tune the inertia weight parameter $\omega$. The parameter is tuned with a function based on the best individual fitness and the average population fitness (CT-SR:1). Vinay Kumar & Rao (2017) proposed a control technique to tune the parameter $w$ based on the particle's fitness (TC-SR:1). In Zhang & Ding (2011) a variation of PSO with sub-swarms is proposed using an adaptive dynamic strategy for inertia weight ($\omega$) based on the fitness values of all sub-swarms (CT-SR:1). In Tang et al. (2015) the inertia weight parameter ($\omega$) is controlled based on a diversity measurement (CT-SR:1). Zhang et al. (2018) proposes to control the parameter $w$ based on population's diversity. The adaptive adjustment is made in accordance with the aggregation degree of particles. (TC-SR:1). Zielinski & Laur (2007) presents a control method for the parameters $\omega$, $C1$ and $C2$. In each generation different combinations of values are used for the parameters and they are tuned according to their success in generating better solutions (CT-SR:3). Harrison et al. (2017) proposes a tuning technique for the parameters $w$, C1 and C2. If the particles personal best position has stagnated for a predefined number of iterations, new parameters values are randomly sampled from a region known to contain promising parameter configurations (CT-SR:3). In Montalvo, Izquierdo, Pérez-García & Herrera (2010) different parameter control techniques are employed. The inertia weight parameter ($\omega$) is controlled by a deterministic approach that decreases the parameter value over iterations (CT-DT:1). $C_1$ and $C_2$ parameters are also controlled using a simple rule (CT-SR:2). Setyawan et al. (2017) employed tuning techniques for parameters $w$, C1 e C2 based on the Gaussian function. An adaptive inertia weight approach is adopted based on a percentage of successful in every iteration (TC-SR:1), whilst the parameters C1 and C2 are tuned based on its maximum and minimum values and the number of iterations (CT-DT:2). Hashemi & Meybodi (2011) presents two approaches using learning automata to tune the parameters $\omega$, $C1$ and $C2$. In the first approach, called adventurer, the automata has the liberty to change drastically the parameters values. In the second, called conservative to value changes and gradual, is limited by a fixed value (CT-LA:3). Li & Yang (2009) presents a learning strategy to select the mutation operator. It consists in four different types of mutation that are initialized with the same probability. The probability of each operator is adapted through its success rate, respecting the minimum limit of the probability (CT-SR:1). Melin et al. (2013) presents a fuzzy control method for the parameters $C1$ and $C2$. This control uses the number of iterations, a diversity measure and an error measure based on fitness to tune the parameters values (CT-FC:2). **?** presents a fuzzy control method for the parameters $\omega$, $C1$ and $C2$. The control uses the best fitness value and the number of iterations that the fitness has not improved to tune the parameters (CT-FC:3). Olivas et al. (2013) presents a fuzzy control method for the parameters $C1$ and $C2$. The control uses a population diversity measure and the number of iterations to tune the parameters (CT-FC:2). In the same work is presented a modification for the fuzzy control where the input variables, the diversity measure and the number of generations are considered fuzzy (CT-FC:2). In their work, the second model achieved better results. Neshat (2013) presents a fuzzy control method for the parameters $C1$ and $C2$. The control uses the population fitness values to tune the parameters (CT-FC:2). Liang et al. (2014) presents a control method for the parameter $\omega$. The parameter control is performed through the difference between the population clusters and the best individual (CT-CL:1).

## 6 Discussions

The bio-inspired algorithms (Section 2) have parameters, from few to many, that need to be adjusted for a proper functioning and are problem dependent (i.e., there is not a set of parameter values that best fit to all classes of problems). Hence, doing such parameter control is not a simple task. With the objective to find the best ways to perform the parameter control, different control techniques was revised in Section 4. Their applications in bio-inspired algorithms are shown in Section 5.

In the literature review presented in this work, an amount of 158 scientific articles were analyzed in the context of parameter control techniques applied in bio-inspired algorithms. Through the review of all papers, it was noticed that there is a huge gap between the application of on-line parameter control techniques and its theoretical analysis. Therefore, there is a lack of theoretical convergence analysis in all proposals. In Harrison et al. (2016) it is discussed the importance of theoretical analysis for control of parameters, specifically in PSO, but it can be generalized to other algorithms. In Bonyadi (2018), assumptions for designing adaptive PSO algorithms are theoretically investigated. Their theoretical findings provided a beneficial guideline for successful adaptation of the parameters in the PSO algorithm.

Figure 2 presents the distribution of bio-inspired algorithms in relation to the quantity of tuned parameters. Based on the papers reviewed, the techniques for parameter control are more frequently applied in GA, representing 20.25% of all control methods analyzed. PSO is in second place with 16.46%, and DE in third place with 12.03%. This distribution can be explained by the fact that GA, PSO and DE are well-known algorithms and their efficiency are proved in different problem domains. Besides that, they are simple to implement favoring their choice. In this way, the opportunity to apply on-line parameter control techniques in other bio-inspired algorithms are clear.

Figure 3 shows the distribution of the parameter control methods (i.e., aggregated, deterministic, and adaptive) applied in the bio-inspired algorithms. The most used method was the adaptive control in 70.93% over all works reviewed. This shows the concern of the scientific community with the use of feedback information from the optimization process to adapt parameter values. Also, depending on the search stage the set of optimum parameter values may be different. Deterministic control comes next with 20.93%. The motivation of using deterministic control is in its simplicity of implementation. Even without using feedback information during the optimization process, the results obtained with deterministic control has shown to be better than using static values. Hence, despite its simplicity, deterministic control techniques favor a better diversification and intensification balance during the search compared with approaches using parameters with static values. Aggregated control appears in 8.14% of the analyzed works mainly by the low gain in efficiency when compared to the other two control strategies. It is important to highlight that for every work reviewed in this paper the results presented are better when using parameter control techniques.

Figure 4 shows the distribution of control methods employed for each bio-inspired algorithm. As a complement of Figure 3, for each algorithm there exists the application of at least one adaptive control method, being the most used technique. Based on the works reviewed, deterministic methods are used in several algorithms except for ES, ABC, GEP, and GP. Aggregated control methods are mainly found in DE and GA algorithms, followed by FA, BFO, PSO, GEP, and ES algorithms.

Figure 5 shows the adaptive control techniques distribution. The simple rule techniques (CT-SR), learning automata (CT-LA), fuzzy control (CT-FC), clustering (CT-CL), entropy (CT-ET), covariance matrix (CT-CM), and pheromone matrix (CT-PM), are representatives in this group. The most used

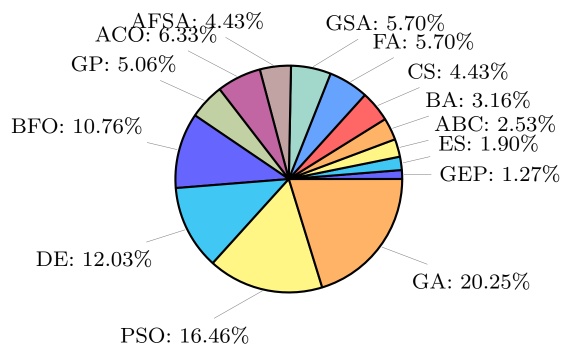


**Figure 2** Bio-inspired algorithms distribution in relation to the amount of parameters tuned

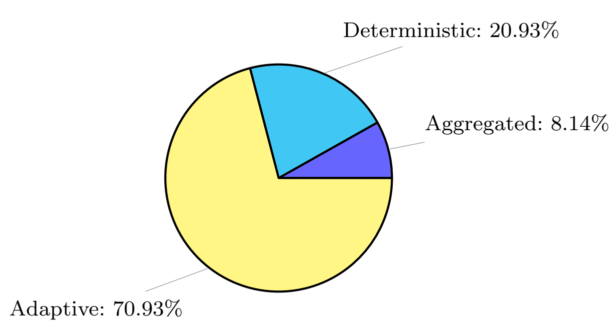


**Figure 3** Distribution of control mechanisms

technique is the simple rule due to its simplicity of implementation, representing 72.68%. The fuzzy control represents 19.67% showing promising results, however, the task of determining the fuzzy rule set and its respective thresholds can be a tough task, specially in this case of parameter control. The entropy technique has 3.28%, followed by the learning automata, pheromone matrix, clustering, and covariance matrix with 1.64%, 1.64%,0.55%, and 0.55%, respectively.

Figure 6 summarizes the use of adaptive control techniques applied in each bio-inspired algorithms. The ACO and PSO algorithms have the highest diversity of adaptive controls. All algorithms have the application of at least one simple rule technique. Fuzzy control can be found in the GA, PSO, DE, BFO, ACO, AFSA, FA, and GSA algorithms. The learning automata is found only in PSO, pheromone matrix is found only for the ACO algorithm and the covariance matrix is applied only in ES. The cluster technique was applied in the PSO algorithm.

## 7 Conclusions

Parameter control plays a crucial role in meta-heuristics directly influencing the process of finding promising regions in the search space where a proper adjustment of values can lead to better results. Also, parameter control can be seen as an upper-level optimization problem to be handled besides the optimization problem being solved. Hence, a literature review considering the on-line control of parameters in bio-inspired algorithms is presented in this work.

In the first moment, a review of the taxonomies for parameter control was made. Considering the differences among the taxonomies found, it was presented an extended version of the taxonomy based on Eiben et al. (1999) and Zhang et al. (2012). The proposed taxonomy has the objective to highlight the technique used for parameter control. With the proposed taxonomy in hand it was possible to classify the works identified during the review and summarize them properly.

The main objective of this review was to identify and classify the most used techniques for on-line parameter control. Three categories were identified: Deterministic, Aggregated, and Adaptive. Due to the quantity of different techniques for adaptive control it was necessary to divide this category in subcategories. Also, the most used techniques found in literature were the deterministic (mainly by its simplicity) and the adaptive (mainly by its robustness). Hence, the adaptive techniques are highlighted due to the use of search feedback information to guide the choice for parameters' values.

Based on the works reviewed, this survey has its significance in presenting a summary of the main on-line parameter control techniques utilized in the literature when applying EC and SI algorithms. With that it is possible to show the more active areas of research regarding the subject as well as presenting a cohesive discussion.

Working as a main hub for on-line parameter control using EC and SI, the structure of our work makes possible to find what are the main methods in use and which algorithms are using them. It is also highlighted what bio-inspired optimization algorithms are being

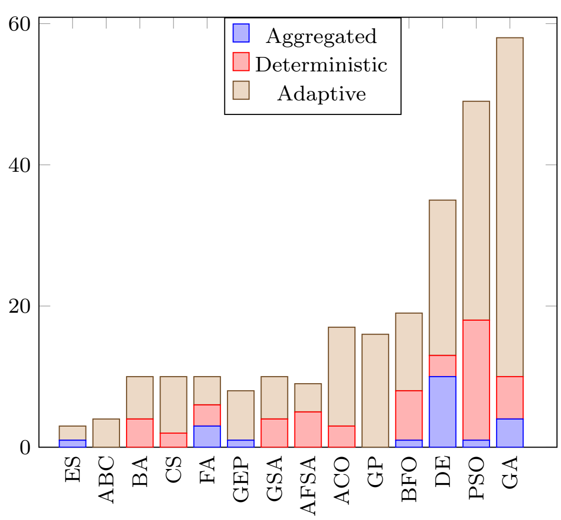


**Figure 4** Distribution of control mechanisms by algorithm

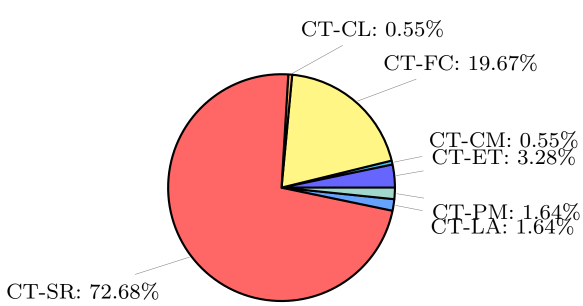


**Figure 5** Distribution of adaptive control mechanisms

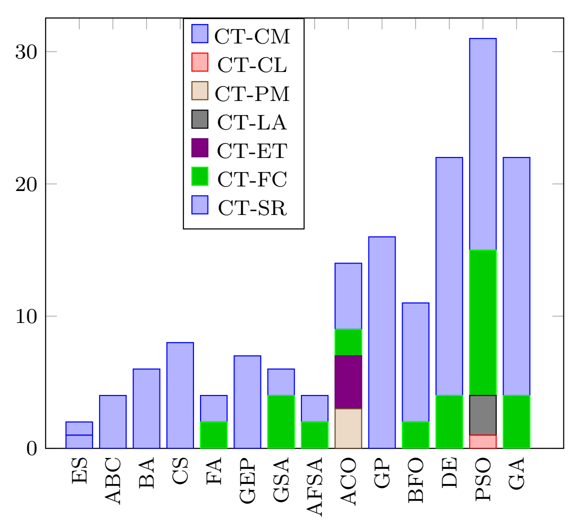


**Figure 6** Distribution of adaptive control mechanisms by algorithm

used and what parameter control techniques are being employed, working as a hands-on guide.

During the review it was found that there is a lot of opportunities to explore the application of on-line parameter control in other algorithms besides GA, PSO, and DE. Also, it is important to highlight that through all papers reviewed there is a huge gap between the application of on-line parameter control techniques and its theoretical analysis. Therefore, there is a lack of theoretical convergence analysis in all proposals. This indicates that new (and existing) optimizers that use on-line parameter control techniques must be examined in greater detail both analytically and empirically.

As future research directions, experiments could be done to evaluate the efficiency of parameter control techniques, pointing out its weaknesses and strengths when coupled with different optimization algorithms, applied in different problem domains. Another suggestion is the development of hybrid strategies combining different techniques. Understand the convergence behavior of the tuned parameters is another interesting direction.

This review points out for the importance of theoretical and convergence analysis of on-line control techniques. To improve general usage of bio-inspired algorithms in solving real life problems, the sensitivity of these algorithms to the parameter settings should be eliminated, while maintaining their performance. This can be achieved by using on-line parameter control techniques leading to parameter free (or parameterless) algorithms. Hence, the design of parameterless algorithms is ideal for the user point of view and on-line parameter control contributes in this direction.

## References


Abshouri, A. A., Meybodi, M. R. & Bakhtiary, A. (2011), New firefly algorithm based on multi swarm & learning automata in dynamic environments, *in* 'IEEE proceedings', Vol. 13, pp. 989–993.

Al-Madi, N. & Ludwig, S. A. (2012), Adaptive genetic programming applied to classification in data mining, *in* 'Fourth World Congress on Nature and Biologically Inspired Computing', pp. 79–85.

Aleti, A. (2014), 'Designing automotive embedded systems with adaptive genetic algorithms', *Automated Software Engineering* pp. 1–42.

Aleti, A. & Moser, I. (2011), Predictive parameter control, *in* 'Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation', GECCO '11, ACM, pp. 561–568.

Aleti, A. & Moser, I. (2013*a*), Entropy-based adaptive range parameter control for evolutionary algorithms, *in* 'Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference', GECCO '13, pp. 1501–1508.

Aleti, A. & Moser, I. (2013*b*), Studying feedback mechanisms for adaptive parameter control in evolutionary algorithms, *in* 'Evolutionary Computation (CEC), 2013 IEEE Congress on', pp. 3117–3124.

Aleti, A. & Moser, I. (2016), 'A systematic literature review of adaptive parameter control methods for evolutionary algorithms', *ACM Comput. Surv.* **49**(3), 56:1–56:35.

Aleti, A., Moser, I. & Mostaghim, S. (2012), Adaptive range parameter control, *in* 'Evolutionary Computation (CEC), 2012 IEEE Congress on', pp. 1–8.

Alfi, A. (2012), 'Particle swarm optimization algorithm with dynamic inertia weight for online parameter identification applied to lorenz chaotic system', *Int J Innov Comput Inf Control* **8**, 1191 – 1204.

Algethami, H. & Landa-Silva, D. (2017), Diversity-based adaptive genetic algorithm for a workforce scheduling and routing problem, *in* 'Evolutionary Computation (CEC), 2017 IEEE Congress on', IEEE, pp. 1771–1778.

Ali, M. M. & Törn, A. (2004), 'Population set-based global optimization algorithms: Some modifications and numerical studies', *Comput. Oper. Res.* **31**(10), 1703–1725.

Angeline, P. J. (1995), Adaptive and self-adaptive evolutionary computations, *in* 'Computational Intelligence: A Dynamic Systems Perspective', IEEE Press, pp. 152–163.

Askari, H. & Zahiri, S.-H. (2012), 'Decision function estimation using intelligent gravitational search algorithm', *International Journal of Machine Learning and Cybernetics* **3**(2), 163–172.

Aydin, D., Özyön, S., Yaşar, C. & Liao, T. (2014), 'Artificial bee colony algorithm with dynamic population size to combined economic and emission dispatch problem', *International Journal of Electrical Power & Energy Systems* **54**(0), 144–153.

Back, T. & Schutz, M. (1996), Intelligent mutation rate control in canonical genetic algorithms, *in* 'Foundation of Intelligent Systems 9th International Symposium, ISMIS '96', Springer, pp. 158–167.

Bautu, E., Bautu, A. & Luchian, H. (2007), AdaGEP - an adaptive gene expression programming algorithm, *in* 'Proceedings of the Ninth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing', pp. 403–406.

Benitez, C. M. V., Parpinelli, R. S. & Lopes, H. S. (2012), 'Parallelism, hybridism and coevolution in a multi-level ABC-GA approach for the protein structure prediction problem', *Concurrency and Computation: Practice and Experience* **24**(6), 635–646.

Berkhin, P. (2006), A survey of clustering data mining techniques, *in* J. Kogan, C. Nicholas & M. Teboulle, eds, 'Grouping Multidimensional Data', Springer Berlin Heidelberg, pp. 25–71.

Beyer, H.-G. (1995), 'Toward a theory of evolution strategies: Self-adaptation'.

Beyer, H.-G. & Schwefel, H.-P. (2002), 'Evolution strategies–a comprehensive introduction', *Natural computing* **1**(1), 3–52.

Bi, F., Wang, Y. & Chen, W. (2016), Adaptive genetic algorithm to optimize the parameters of evaluation function of dots-and-boxes, *in* 'International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness', Springer, pp. 416–425.

Boeringer, D. & Werner, D. (2002), 'Adaptive mutation parameter toggling genetic algorithm for phase-only array synthesis', *Electronics Letters* **38**(25), 1618–1619.

Bonyadi, M. R. (2018), 'A theoretical guideline for designing an effective adaptive particle swarm', *CoRR* **abs/1802.04855**.

Boudjelaba, K., Ros, F. & Chikouche, D. (2014), 'An efficient hybrid genetic algorithm to design finite impulse response filters', *Expert Systems with Applications* **41**(13), 5917 – 5937.

Boussaid, I., Lepagnot, J. & Siarry, P. (2013), 'A survey on optimization metaheuristics', *Information Sciences* **237**, 82 – 117.

Brest, J., Zumer, V. & Maucec, M. (2006), Self-adaptive differential evolution algorithm in constrained real-parameter optimization, *in* 'Evolutionary Computation, 2006. CEC 2006. IEEE Congress on', pp. 215–222.

Cai, Z. & Huang, H. (2008), Ant colony optimization algorithm based on adaptive weight and volatility parameters, *in* 'Intelligent Information Technology Application, 2008. IITA '08. Second International Symposium on', Vol. 2, pp. 75–79.

Castillo, O., Neyoy, H., Soria, J., García, M. & Valdez, F. (2013), 'Dynamic fuzzy logic parameter tuning for ACO and its application in the fuzzy logic control of an autonomous mobile robot.', *International Journal of Advanced Robotic Systems* **10**.

Chatterjee, A. & Siarry, P. (2006), 'Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization', *Computers and Operations Research* **33**(3), 859 – 871.

Chen, B., Yan, B. & Wang, Y. (2013), Research on multiple kill vehicles firepower distribution strategy based on adjust genetic algorithm, *in* 'Control and Decision Conference (CCDC), 2013 25th Chinese', pp. 3582–3586.

Chen, H., Zhu, Y. & Hu, K. (2011), Adaptive bacterial foraging optimization, *in* 'Abstract and Applied Analysis', Vol. 2011, Hindawi Publishing Corporation.

Chen, L., Zhao, S., Zhu, W., Liu, Y. & Zhang, W. (2014), 'A self-adaptive differential evolution algorithm for parameters identification of stochastic genetic regulatory networks with random delays', *Arabian Journal for Science and Engineering* **39**(2), 821–835.

Chen, Z., Zhou, Y. & LU, M. (2013), 'A simplified adaptive bat algorithm based on frequency', *Journal of Computational Information Systems* **9**(16), 6451– 6458.

Cheung, N. J., Ding, X.-M. & Shen, H.-B. (2014), 'Adaptive firefly algorithm: parameter analysis and its application', *PloS one* **9**(11), e112634.

Chi, R., Su, Y.-x., Zhang, D.-h. & Chi, X.-x. (2016), Adaptive cuckoo search algorithm for continuous function optimization problems, *in* 'Intelligent Control and Automation (WCICA), 2016 12th World Congress on', IEEE, pp. 672–676.

Chiroma, H., Herawan, T., Fister, I., Fister, I., Abdulkareem, S., Shuib, L., Hamza, M. F., Saadi, Y. & Abubakar, A. (2017), 'Bio-inspired computation: Recent development on the modifications of the cuckoo search algorithm', *Applied Soft Computing* **61**, 149 – 173.

Chusanapiputt, S., Nualhong, D., Jantarang, S. & Phoomvuthisarn, S. (2006), Selective self-adaptive approach to ant system for solving unit commitment problem, *in* 'Proceedings of the 8th annual conference on Genetic and evolutionary computation', ACM, pp. 1729–1736.

Cui, L., Li, G., Zhu, Z., Lin, Q., Wen, Z., Lu, N., Wong, K.-C. & Chen, J. (2017), 'A novel artificial bee colony algorithm with an adaptive population size for numerical function optimization', *Information Sciences* .

Dasgupta, D., Yu, S. & Nino, F. (2011), 'Recent advances in artificial immune systems: models and applications', *Applied Soft Computing* **11**(2), 1574–1587.

Dasgupta, S., Das, S., Abraham, A. & Biswas, A. (2009), 'Adaptive computational chemotaxis in bacterial foraging optimization: An analysis', *Evolutionary Computation, IEEE Transactions on* **13**(4), 919–941.

Datta, T., Misra, I. S., Mangaraj, B. B. & Imtiaj, S. (2008), 'Improved adaptive bacteria foraging algorithm in optimization of antenna array for faster convergence', *Progress In Electromagnetics Research C* **1**, 143–157.

Dawar, D. & Ludwig, S. A. (2018), 'Effect of strategy adaptation on differential evolution in presence and absence of parameter adaptation: An investigation', *Journal of Artificial Intelligence and Soft Computing Research* **8**, 211–235.

De Jong, K. (2006), *Evolutionary Computation: A Unified Approach*, Bradford Book, Mit Press.

De Jong, K. (2007), Parameter setting in eas: a 30 year perspective, Springer Berlin Heidelberg, pp. 1–18.

Dhal, K. G. & Das, S. (2018), 'A dynamically adapted and weighted bat algorithm in image enhancement domain', *Evolving Systems* pp. 1–19.

Ding, Y.-Y. & Wang, X.-Y. (2008), Real-coded adaptive genetic algorithm applied to pid parameter optimization on a 6r manipulators, *in* 'Natural Computation, 2008. ICNC 08. Fourth International Conference on', Vol. 1, pp. 635–639.

Dorigo, M. & Di Caro, G. (1999), Ant colony optimization: a new meta-heuristic, *in* 'Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on', Vol. 2, IEEE.

Dragoi, E.-N., Curteanu, S., Galaction, A.-I. & Cascaval, D. (2013), 'Optimization methodology based on neural networks and self-adaptive differential evolution algorithm applied to an aerobic fermentation process', *Applied Soft Computing* **13**(1), 222 – 238.

Eberhart, R. & Shi, Y. (2001), Tracking and optimizing dynamic systems with particle swarms, *in* 'Evolutionary Computation, 2001. Proceedings of the 2001 Congress on', Vol. 1, pp. 94–100 vol. 1.

Eiben, A. E. & Smit, S. K. (2011), 'Parameter tuning for configuring and analyzing evolutionary algorithms', *Swarm and Evolutionary Computation* .

Eiben, A., Hinterding, R. & Michalewicz, Z. (1999), 'Parameter control in evolutionary algorithms', *Evolutionary Computation, IEEE Transactions on* **3**(2), 124–141.

Farhat, I. & El-Hawary, M. (2010), 'Dynamic adaptive bacterial foraging algorithm for optimum economic dispatch with valve-point effects and wind power', *Generation, Transmission Distribution, IET* **4**(9), 989–999.

Fernandez-Prieto, J. A., Canada-Bago, J., Gadeo-Martos, M. A. & Velasco, J. R. (2011), 'Optimisation of control parameters for genetic algorithms to test computer networks under realistic traffic loads', *Appl. Soft Comput.* **11**(4), 3744–3752.

Ferreira, C. (2001), 'Gene expression programming: a new adaptive algorithm for solving problems', *Complex Systems* **13**, 87–129.

Fister, I., Brest, J. & Mlakar, U. (2016), Towards the universal framework of stochastic nature-inspired population-based algorithms, *in* 'Computational Intelligence (SSCI), 2016 IEEE Symposium Series on', IEEE, pp. 1–8.

Fister, I., Yang, X.-S., Brest, J. & Fister, I. J. (2013), 4 - memetic self-adaptive firefly algorithm, *in* 'Swarm Intelligence and Bio-Inspired Computation', Elsevier, Oxford, pp. 73 – 102.

Fister Jr, I., Fong, S., Brest, J. & Fister, I. (2014), Towards the self-adaptation of the bat algorithm, *in* 'The 13th IASTED International Conference on Artificial Intelligence and Applications (AIA 2014). Proceedings of the 13th IASTED International Conference on Artificial Intelligence and Applications (AIA 2014) Innsbruck. IASTED, Feb 2014, p. 400-406.'.

Fister Jr., I., Yang, X.-S., Fister, I., Brest, J. & Fister, D. (2013), 'A Brief Review of Nature-Inspired Algorithms for Optimization', *Elektrotehniski vestnik* **80**, 116–122.

Fitzgerald, J. & Ryan, C. (2013), Individualized self-adaptive genetic operators with adaptive selection in genetic programming, *in* 'World Congress on Nature and Biologically Inspired Computing', pp. 232–237.

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st edn, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Gregor, M. & Spalek, J. (2016), 'Fitness-based adaptive control of parameters in genetic programming: Adaptive value setting of mutation rate and flood mechanisms', *CoRR* .

Gudino-Penaloza, F., Gonzalez-Mendoza, M., Mora-Vargas, J. & Hernandez-Gress, N. (2013), Fuzzy hyperheuristic framework for ga parameters tuning, *in* 'Artificial Intelligence (MICAI), 2013 12th Mexican International Conference on', IEEE, pp. 53–58.

Hansen, N. & Ostermeier, A. (2001), 'Completely derandomized self-adaptation in evolution strategies', *Evolutionary Computation* **9**, 159–195.

Harrison, K. R., Engelbrecht, A. P. & Ombuki-Berman, B. M. (2016), The sad state of self-adaptive particle swarm optimizers, *in* 'IEEE Congress on Evolutionary Computation, CEC 2016, Vancouver, BC, Canada, July 24-29, 2016', pp. 431–439.

Harrison, K. R., Engelbrecht, A. P. & Ombuki-Berman, B. M. (2017), An adaptive particle swarm optimization algorithm based on optimal parameter regions, *in* 'Computational Intelligence (SSCI), 2017 IEEE Symposium Series on', IEEE, pp. 1–8.

Hashemi, A. & Meybodi, M. (2011), 'A note on the learning automata based algorithms for adaptive parameter selection in pso', *Applied Soft Computing* **11**(1), 689 – 705.

Hassanzadeh, T., Meybodi, M. R. & Shahramirad, M. (2017), 'A new fuzzy firefly algorithm with adaptive parameters', *International Journal of Computational Intelligence and Applications* **16**(03), 1750017.

He, S. et al. (2009), Fuzzy clustering with improved artificial fish swarm algorithm, *in* '2009 International Joint Conference on Computational Sciences and Optimization', Vol. 2, pp. 317–321.

Hinterding, R., Michalewicz, Z. & Eiben, A. E. (1997), Adaptation in evolutionary computation: A survey, *in* 'In Proceedings of the Fourth International Conference on Evolutionary Computation (ICEC 97', IEEE Press, pp. 65–69.

Hongrui, X., Ran, L., Jianli, G. & Hongru, W. (2009), An adaptive meta-cognitive artificial fish school algorithm, *in* 'Proceedings of the 2009 International Forum on Information Technology and Applications - Volume 01', IFITA '09, IEEE Computer Society, pp. 594–597.

Huang, M., Wang, L.-m. & Liang, X. (2016), An improved adaptive genetic algorithm in flexible job shop scheduling, *in* 'Cloud Computing and Internet of Things (CCIOT), 2016 2nd International Conference on', IEEE, pp. 177–184.

Huang., Z. & Chen, Y. (2013), 'An improved differential evolution algorithm based on adaptive parameter', *Journal of Control Science and Engineering* **2013**.

Jaballah, A. & Meddeb, A. (2017), Self adaptive cuckoo search algorithm for rfid network planning, *in* 'Internet Technologies and Applications (ITA), 2017', IEEE, pp. 122–127.

Jain, A. K., Murty, M. N. & Flynn, P. J. (1999), 'Data clustering: A review', *ACM Comput. Surv.* **31**(3), 264–323.

Jantzen, J. (2007), *Fuzzy Control*, John Wiley e Sons, Ltd, pp. 47–70.

Ji, J., Gao, S., Wang, S., Tang, Y., Yu, H. & Todo, Y. (2017), 'Self-adaptive gravitational search algorithm with a modified chaotic local search', *IEEE Access* **5**, 17881–17895.

Jiang, Y., Zhou, P., Zhan, R., Li, X. & Li, D. (2016), An artificial bee colony with self-adaptive operators and alterable search depth approach for intercell scheduling, *in* 'Evolutionary Computation (CEC), 2016 IEEE Congress on', IEEE, pp. 112–119.

Jiao, B., Lian, Z. & Gu, X. (2008), 'A dynamic inertia weight particle swarm optimization algorithm', *Chaos, Solitons and Fractals* **37**(3), 698 – 705.

Jiye, L., Xihong, C., Qiang, L. & Jizhe, S. (2013), Prediction of satellite clock errors using ls-svm optimized by improved artificial fish swarm algorithm, *in* 'Signal Processing, Communication and Computing (ICSPCC), 2013 IEEE International Conference on', pp. 1–5.

Kabir, M. W. U., Sakib, N., Chowdhury, S. M. R. & Alam, M. S. (2014), 'A novel adaptive bat algorithm to control explorations and exploitations for continuous optimization problems', *International Journal of Computer Applications* **94**(13).

Kalkreuth, R., Rudolph, G. & Krone, J. (2015), Improving convergence in cartesian genetic programming using adaptive crossover, mutation and selection, *in* 'IEEE Symposium Series on Computational Intelligence', pp. 1415–1422.

Kamrath, N. R., Goldman, B. W. & Tauritz, D. R. (2013), Using supportive coevolution to evolve self-configuring crossover, *in* 'Proceeding of the fifteenth annual conference companion on Genetic and evolutionary computation conference companion', GECCO '13 Companion, ACM.

Karaboga, D. & Akay, B. (2009), 'A comparative study of artificial bee colony algorithm', *Applied Mathematics and Computation* **214**(1), 108 – 132.

Karafotias, G., Hoogendoorn, M. & Eiben, Á. E. (2015), 'Parameter control in evolutionary algorithms: Trends and challenges', *IEEE Trans. Evolutionary Computation* **19**(2), 167–187.

Kennedy, J. & Eberhart, R. (1995), Particle swarm optimization, *in* 'Neural Networks, 1995. Proceedings., IEEE International Conference on', Vol. 4, pp. 1942–1948 vol.4.

Khadhraoui, T., Ktata, S., Benzarti, F. & Amiri, H. (2016), Features selection based on modified pso algorithm for 2d face recognition, *in* 'Computer Graphics, Imaging and Visualization (CGiV), 2016 13th International Conference on', IEEE, pp. 99–104.

Khichane, M., Albert, P. & Solnon, C. (2009), Learning and intelligent optimization, Springer-Verlag, Berlin, Heidelberg, chapter An ACO-Based Reactive Framework for Ant Colony Optimization: First Experiments on Constraint Satisfaction Problems, pp. 119–133.

Kotinis, M. (2014), 'Improving a multi-objective differential evolution optimizer using fuzzy adaptation and k-medoids clustering', *Soft Computing* **18**(4), 757–771.

Kovaevi, D., Mladenovi, N., Petrovi, B. & Miloevi, P. (2014), 'De-vns: Self-adaptive differential evolution with crossover neighborhood search for continuous global optimization', *Computers & Operations Research* (0), –.

Koza, J. R. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA.

Kramer, O. (2010), 'Evolutionary self-adaptation: a survey of operators and strategy parameters', *Evolutionary Intelligence* **3**(2), 51–65.

Kumar, J. V., Kumar, D. V. & Edukondalu, K. (2013), 'Strategic bidding using fuzzy adaptive gravitational search algorithm in a pool based electricity market', *Applied Soft Computing* **13**(5), 2445 – 2455.

Le Hai Nam, Hoang Tuan Hao, V. V. C. (2015), 'Self-adaptive crossover and mutation parameters in tree adjoining grammar guided genetic programming', *Le Quy Don Technical University* pp. 5–15.

Leung, S. W., Zhang, X. & Yuen, S.-Y. (2012), Multiobjective differential evolution algorithm with opposition-based parameter control, *in* 'Evolutionary Computation (CEC), 2012 IEEE Congress on', pp. 1–8.

Li, C. & Yang, S. (2009), An adaptive learning particle swarm optimizer for function optimization, *in* 'Evolutionary Computation, 2009. CEC09. IEEE Congress on', IEEE, pp. 381–388.

Li, H. & Li, P. (2013), 'Self-adaptive ant colony optimization for construction time-cost optimization', *Kybernetes* **42**(8), 1181–1194.

Li, R. & Chang, X. (2006), A modified genetic algorithm with multiple subpopulations and dynamic parameters applied in cvar model, *in* 'Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on', pp. 151–151.

Li, Y. (2009), 'A novel ACO with average entropy.', *Journal of Software Engineering & Applications* **2**(5).

Liang, X., Li, W., Zhang, Y. & Zhou, M. (2014), 'An adaptive particle swarm optimization method based on clustering', *Soft Computing* pp. 1–18.

Linda, M. M. & Nair, N. K. (2013), 'A new-fangled adaptive mutation breeder genetic optimization of global multi-machine power system stabilizer', *International Journal of Electrical Power and Energy Systems* **44**(1), 249 – 258.

Liu, J. & Lampinen, J. (2005), 'A fuzzy adaptive differential evolution algorithm', *Soft Computing* **9**(6), 448–462.

Liu, L., Liu, X., Wang, N. & Zou, P. (2018), 'Modified cuckoo search algorithm with variational parameters and logistic map', *Algorithms* **11**(3), 30.

Majhi, R., Panda, G., Majhi, B. & Sahoo, G. (2009), 'Efficient prediction of stock market indices using adaptive bacterial foraging optimization (abfo) and bfo based techniques', *Expert Systems with Applications* **36**(6), 10097 – 10104.

Mallipeddi, R. & Suganthan, P. (2010), Differential evolution algorithm with ensemble of parameters and mutation and crossover strategies, *in* 'Swarm, Evolutionary, and Memetic Computing', Vol. 6466 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 71–78.

Mareli, M. & Twala, B. (2017), 'An adaptive cuckoo search algorithm for optimisation', *Applied Computing and Informatics* .

Marques, V. & Gomide, F. (2011), 'Parameter control of metaheuristics with genetic fuzzy systems', *Evolutionary Intelligence* pp. 183–202.
**URL:** *http://dx.doi.org/10.1007/s12065-011-0059-y*

Mavrovouniotis, M. & Yang, S. (2014), Ant colony optimization with self-adaptive evaporation rate in dynamic environments, *in* 'Computational Intelligence in Dynamic and Uncertain Environments (CIDUE), 2014 IEEE Symposium on', IEEE, pp. 47–54.

McCulloch, W. S. & Pitts, W. (1943), 'A logical calculus of the ideas immanent in nervous activity', *Bulletin of Mathematical Biophysics* **5**, 115–133.

Melin, P., Olivas, F., Castillo, O., Valdez, F., Soria, J. & Valdez, M. (2013), 'Optimal design of fuzzy classification systems using {PSO} with dynamic parameter adaptation through fuzzy logic', *Expert Systems with Applications* **40**(8), 3196 – 3206.

Meng, Z. & Pan, J.-S. (2016), Quasi-affine transformation evolutionary (quatre) algorithm: a parameter-reduced differential evolution algorithm for optimization problems, *in* 'Evolutionary Computation (CEC), 2016 IEEE Congress on', IEEE, pp. 4082–4089.

Merkle, D., Middendorf, M. & Schmeck, H. (2002), 'Ant colony optimization for resource-constrained project scheduling', *Evolutionary Computation, IEEE Transactions on* **6**(4), 333–346.

Meyer, B. (2004), Convergence control in ACO, *in* 'Genetic and Evolutionary Computation Conference (GECCO), Seattle, WA, late-breaking paper available on CD'.

Mezura Montes, E. & López-Dávila, E. A. (2012), Adaptation and local search in the modified bacterial foraging algorithm for constrained optimization, *in* 'Evolutionary Computation (CEC), 2012 IEEE Congress on', IEEE, pp. 1–8.

Mishra, S. (2005), 'A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation', *Evolutionary Computation, IEEE Transactions on* **9**(1), 61–73.

Montalvo, I., Izquierdo, J., Pérez-García, R. & Herrera, M. (2010), 'Improved performance of pso with self-adaptive parameters for computing the optimal design of water supply systems', *Engineering applications of artificial intelligence* **23**(5), 727–735.

Montalvo, I., Izquierdo, J., Prez-Garca, R. & Herrera, M. (2010), 'Improved performance of pso with self-adaptive parameters for computing the optimal design of water supply systems', *Engineering Applications of Artificial Intelligence* **23**(5), 727 – 735.

Mwaura, J. & Keedwell, E. (2009), Adaptive gene expression programming using a simple feedback heuristic, *in* 'Proceedings of the Artificial Intelligence and Simulation of Behaviour(AISB2009)'.

Naik, M. K. & Panda, R. (2016), 'A novel adaptive cuckoo search algorithm for intrinsic discriminant analysis based face recognition', *Applied Soft Computing* **38**, 661–675.

Naik, M., Nath, M. R., Wunnava, A., Sahany, S. & Panda, R. (2015), A new adaptive cuckoo search algorithm, *in* 'Recent Trends in Information Systems (ReTIS), 2015 IEEE 2nd International Conference on', IEEE, pp. 1–5.

Nasir, A., Tokhi, M. O. & Ghani, N. A. (2015), 'Novel adaptive bacterial foraging algorithms for global optimisation with application to modelling of a trs', *Expert Systems with Applications* **42**(3), 1513–1530.

Neshat, M. (2013), 'FAIPSO: fuzzy adaptive informed particle swarm optimization', *Neural Computing and Applications* **23**(1), 95–116.

Niknam, T. & Azad Farsani, E. (2010), 'A hybrid self-adaptive particle swarm optimization and modified shuffled frog leaping algorithm for distribution feeder reconfiguration', *Eng. Appl. Artif. Intell.* **23**(8), 1340–1349.

Niknam, T., Azizipanah-Abarghooee, R. & Roosta, A. (2012), 'Reserve constrained dynamic economic dispatch: a new fast self-adaptive modified firefly algorithm', *Systems Journal, IEEE* **6**(4), 635–646.

Niknam, T., Narimani, M., Azizipanah-Abarghooee, R. & Bahmani-Firouzi, B. (2013), 'Multiobjective optimal reactive power dispatch and voltage control: A new opposition-based self-adaptive modified gravitational search algorithm', *Systems Journal, IEEE* **7**(4), 742–753.

Obaidat, M., Papadimitriou, G. & Pomportsis, A. (2002), 'Guest editorial learning automata: theory, paradigms, and applications', *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* **32**(6), 706–709.

Okafor, A. (2005), Entropy based techniques with applications in data mining, PhD thesis, University of Florida.

Olivas, F., Valdez, F. & Castillo, O. (2013), Particle swarm optimization with dynamic parameter adaptation using interval type-2 fuzzy logic for benchmark mathematical functions, *in* 'Nature and Biologically Inspired Computing (NaBIC), 2013 World Congress on', pp. 36–40.

Osaba, E., Onieva, E., Carballedo, R., Diaz, F., Perallos, A. & Zhang, X. (2013), 'A multi-crossover and adaptive island based population algorithm for solving routing problems', *Journal of Zhejiang University SCIENCE C* **14**(11), 815–821.

Pan, Z. & Chen, L. (2013), Minimization of binary decision diagrams by adaptive hierarchy genetic algorithm and its application in circuit test generation, *in* 'Advances in Mechanical and Electronic Engineering', Vol. 178, Springer Berlin Heidelberg, pp. 175–180.

Panigrahi, B. & Pandi, V. R. (2009), 'Congestion management using adaptive bacterial foraging algorithm', *Energy Conversion and Management* **50**(5), 1202 – 1209.

Parpinelli, R. S. & Lopes, H. S. (2011), 'New inspirations in swarm intelligence; a survey', *International Journal of Bio-Inspired Computation* **3**(1), 1–16.

Passino, K. (2002), 'Biomimicry of bacterial foraging for distributed optimization and control', *Control Systems, IEEE* **22**(3), 52–67.

Pauline, O., Sin, H. C., Sheng, D. D. C. V., Kiong, S. C. & Meng, O. K. (2017), Design optimization of structural engineering problems using adaptive cuckoo search algorithm, *in* 'Control, Automation and Robotics (ICCAR), 2017 3rd International Conference on', IEEE, pp. 745–748.

Peng, Y., Luo, X. & Wei, W. (2014), 'A new fuzzy adaptive simulated annealing genetic algorithm and its convergence analysis and convergence rate estimation', *International Journal of Control, Automation and Systems* **12**(3), 670–679.

Pereira, A. G., Roveda, J. A., Carlos, L. A., Campos, V. S. M. & Roveda, S. R. (2013), Convergence analysis of an elitist non-homogeneous genetic algorithm with mutation probability adjusted by a fuzzy controller, *in* 'IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint', IEEE, pp. 19–23.

Piotrowski, A. P. (2013), 'Adaptive memetic differential evolution with global and local neighborhood-based mutation operators', *Information Sciences* **241**(0), 164 – 194.

Ponz-Tienda, J. L., Yepes, V., Pellicer, E. & Moreno-Flores, J. (2013), 'The resource leveling problem with multiple resources using an adaptive genetic algorithm', *Automation in Construction* **29**(0), 161 – 172.

Precup, R., David, R., Petriu, E., Preitl, S. & Radac, M. (2012), 'Novel adaptive gravitational search algorithm for fuzzy controlled servo systems', *Industrial Informatics, IEEE Transactions on* **8**(4), 791–800.

Precup, R.-E., David, R.-C., Petriu, E., Preitl, S. & Radac, M.-B. (2013), 'Fuzzy logic-based adaptive gravitational search algorithm for optimal tuning of fuzzy-controlled servo systems', *Control Theory Applications, IET* **7**(1), 99–107.

Rajakumar, B. & George, A. (2013), 'Apoga: An adaptive population pool size based genetic algorithm', {*AASRI*} *Procedia* **4**(0), 288 – 296.

Rajappan, S. & Rangasamy, D. (2017), 'Adaptive genetic algorithm with exploration-exploitation tradeoff for preprocessing microarray datasets', *Current Bioinformatics* **12**(5), 441–451.

Rajkumar, P., Balamurgan, V. & Mishra, P. (2016), Adaptive genetic algorithm for a real time medical images, *in* 'Green Engineering and Technologies (IC-GET), 2016 Online International Conference on', IEEE, pp. 1–5.

Rashedi, E., Nezamabadi-pour, H. & Saryazdi, S. (2009), 'Gsa: A gravitational search algorithm', *Information Sciences* **179**(13), 2232 – 2248. Special Section on High Order Fuzzy Sets.

Rashtchi, V., Bayat, A. & Vahedi, H. (2009), Adaptive step length bacterial foraging algorithm, *in* 'Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on', Vol. 1, pp. 322–326.

Rechenberg, I. (1973), *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Problemata, 15, Frommann-Holzboog.

Rezaee Jordehi, A. & Jasni, J. (2013), 'Parameter selection in particle swarm optimisation: a survey', *Journal of Experimental e Theoretical Artificial Intelligence* .

Riley, M., Mei, Y. & Zhang, M. (2016), Improving job shop dispatching rules via terminal weighting and adaptive mutation in genetic programming, *in* 'IEEE Congress on Evolutionary Computation', pp. 3362–3369.

Rokach, L. (2010), A survey of clustering algorithms, *in* O. Maimon & L. Rokach, eds, 'Data Mining and Knowledge Discovery Handbook', Springer US, pp. 269–298.

Roy, A., Rakshit, P., Konar, A., Bhattacharya, S., Kim, E. & Nagar, A. (2013), Adaptive firefly algorithm for nonholonomic motion planning of car-like system, *in* 'Evolutionary Computation (CEC), 2013 IEEE Congress on', pp. 2162–2169.

Saeidi-Khabisi, F.-S. & Rashedi, E. (2012), Fuzzy gravitational search algorithm, *in* 'Computer and Knowledge Engineering (ICCKE), 2012 2nd International eConference on', pp. 156–160.

Sathya, P. & Kayalvizhi, R. (2011), 'Optimal segmentation of brain mri based on adaptive bacterial foraging algorithm', *Neurocomputing* **74**(1415), 2299 – 2313.

Selvarasu, R. & Rajan, C. (2013), Self adaptive firefly algorithm based transmission loss minimization using TCSC, *in* 'Advanced Nanomaterials and Emerging Engineering Technologies (ICANMEET), 2013 International Conference on', pp. 683–688.

Setyawan, N., Kadir, R. E. A. & Jazidie, A. (2017), Adaptive gaussian parameter particle swarm optimization and its implementation in mobile robot path planning, *in* 'Intelligent Technology and Its Applications (ISITIA), 2017 International Seminar on', IEEE, pp. 238–243.

Shannon, C. E. (2001), 'A mathematical theory of communication', *ACM SIGMOBILE Mobile Computing and Communications Review* **5**(1), 3–55.

Shen, X., Zou, D. & Zhang, X. (2017), A self-adaptive differential evolution with dynamic selecting mutation strategy, *in* 'Vision, Image and Signal Processing (ICVISP), 2017 International Conference on', IEEE, pp. 5–10.

Shi, Y. & Eberhart, R. (1999), Empirical study of particle swarm optimization, *in* 'Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on', Vol. 3, pp. –1950 Vol. 3.

Simons, C. & Parmee, I. (2010), Dynamic parameter control of interactive local search in uml software design, *in* 'Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on', pp. 3397–3404.

Singh, A. & Sharan, A. (2017), Adaptive genetic programming based linkage rule miner for entity linking in semantic web, *in* 'Computing, Communication and Automation (ICCCA), 2017 International Conference on', IEEE, pp. 373–378.

Smetek, M. & Trawinski, B. (2011), Investigation of self-adapting genetic algorithms using some multimodal benchmark functions, *in* 'Computational Collective Intelligence. Technologies and Applications', Vol. 6922 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 213–223.

Sombra, A., Valdez, F., Melin, P. & Castillo, O. (2013), A new gravitational search algorithm using fuzzy logic to parameter adaptation, *in* 'Evolutionary Computation (CEC), 2013 IEEE Congress on', pp. 1068–1074.

Srinivas, M. & Patnaik, L. (1994), 'Adaptive probabilities of crossover and mutation in genetic algorithms', *Systems, Man and Cybernetics, IEEE Transactions on* **24**(4), 656–667.

Storn, R. & Price, K. (1997), 'Differential evolution : A simple and efficient heuristic for global optimization over continuous spaces', *J. of Global Optimization* **11**(4), 341–359.

Tan, L., Lin, F. & Wang, H. (2015), 'Adaptive comprehensive learning bacterial foraging optimization and its application on vehicle routing problem with time windows', *Neurocomputing* **151**, 1208–1215.

Tanabe, R. & Fukunaga, A. (2013), Success-history based parameter adaptation for differential evolution, *in* 'Evolutionary Computation (CEC), 2013 IEEE Congress on', pp. 71–78.

Tang, K., Li, Z., Luo, L. & Liu, B. (2015), 'Multi-strategy adaptive particle swarm optimization for numerical optimization', *Engineering Applications of Artificial Intelligence* **37**, 9–19.

Tarokh, M. & Zhang, X. (2014), 'Real-time motion tracking of robot manipulators using adaptive genetic algorithms', *Journal of Intelligent & Robotic Systems* **74**(3-4), 697–708.

Teo, J. (2006), 'Exploring dynamic self-adaptive populations in differential evolution', *Soft Computing* **10**(8), 673–686.

Tian, W. & Tian, Y. (2009), An improved artificial fish swarm algorithm for resource leveling, *in* '2009 International Conference on Management and Service Science', pp. 1–4.

Tuson, A. & Ross, P. (1998), 'Adapting operator settings in genetic algorithms', *Evol. Comput.* pp. 161–184.

Vafaee, F. & Nelson, P. (2009), A genetic algorithm that incorporates an adaptive mutation based on an evolutionary model, *in* 'Machine Learning and Applications, 2009. ICMLA '09. International Conference on', pp. 101–107.

Vajda, P., Eiben, A. E. & Hordijk, W. (2008), Parameter control methods for selection operators in genetic algorithms, *in* 'Parallel Problem Solving from Nature–PPSN X', Springer, pp. 620–630.

Venkaiah, C. & Kumar, D. V. (2011), 'Fuzzy adaptive bacterial foraging congestion management using sensitivity based optimal active power re-scheduling of generators', *Applied Soft Computing* **11**(8), 4921 – 4930.

Verma, O. P., Chopra, R. R. & Gupta, A. (2016), An adaptive bacterial foraging algorithm for color image enhancement, *in* 'Information Science and Systems (CISS), 2016 Annual Conference on', IEEE, pp. 1–6.

Vinay Kumar, S. & Rao, P. (2017), 'Interactive self improvement based adaptive particle swarm optimization', *New Review of Information Networking* **22**(1), 13–33.

Wang, C., Liu, Y., Liang, X., Guo, H., Chen, Y. & Zhao, Y. (2018), 'Self-adaptive differential evolution algorithm with hybrid mutation operator for parameters identification of pmsm', *Soft Computing* **22**(4), 1263–1285.

Wang, H., Zhou, X., Sun, H., Yu, X., Zhao, J., Zhang, H. & Cui, L. (2017), 'Firefly algorithm with adaptive control parameters', *Soft computing* **21**(17), 5091–5102.

Wang, J. (2011), 'Particle swarm optimization with adaptive parameter control and opposition', *Journal of Computational Information Systems* **7**, 4463– 4470.

Wang, X. & Zhao, S. (2013), 'Differential evolution algorithm with self-adaptive population resizing mechanism', *Mathematical Problems in Engineering* **2013**.

X. Li, Z. Shao, J. Q. (2002), 'An optimizing method base on autonomous animates: fish- swarm algorithm', pp. 32–38.

Xu, X. & Chen, H.-l. (2014), 'Adaptive computational chemotaxis based on field in bacterial foraging optimization', *Soft Computing* **18**(4), 797–807.

Yan, X., Zhu, Y., Wu, J. & Chen, H. (2012), 'An improved firefly algorithm with adaptive strategies', *Advanced Science Letters* **16**(1), 249–254.

Yan, X., Zhu, Y., Zhang, H., Chen, H. & Niu, B. (2012), 'An adaptive bacterial foraging optimization algorithm with lifecycle and social learning', *Discrete Dynamics in Nature and Society* **2012**.

Yang, C., Han, J., Li, Y. & Xu, X. (2013), Self-adaptive genetic algorithm based mu-mimo scheduling scheme, *in* 'Communication Technology (ICCT), 2013 15th IEEE International Conference on', IEEE, pp. 180–185.

Yang, K., Zheng, J., Yang, M., Zhou, R. & Liu, G. (2013), 'Adaptive genetic algorithm for daily optimal operation of cascade reservoirs and its improvement strategy', *Water Resources Management* **27**(12), 4209–4235.

Yang, L., Widyantoro, D., Ioerger, T. & Yen, J. (2001), An entropy-based adaptive genetic algorithm for learning classification rules, *in* 'Evolutionary Computation, 2001. Proceedings of the 2001 Congress on', Vol. 2, pp. 790–796 vol. 2.

Yang, X.-S. (2010*a*), *Nature-inspired metaheuristic algorithms*, Luniver Press.

Yang, X.-S. (2010*b*), A new metaheuristic bat-inspired algorithm, *in* 'Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)', Vol. 284 of *Studies in Computational Intelligence*, Springer Berlin Heidelberg, pp. 65–74.

Yang, X.-S. & Deb, S. (2010), 'Engineering Optimisation by Cuckoo Search', *International Journal of Mathematical Modelling and Numerical Optimisation* **1**(4), 330–343.

Yang, X., Yuan, J., Yuan, J. & Mao, H. (2007), 'A modified particle swarm optimizer with dynamic adaptation', *Applied Mathematics and Computation* **189**(2), 1205 – 1213.

Yasuda, K., Yazawa, K. & Motoki, M. (2010), 'Particle swarm optimization with parameter self-adjusting mechanism', *IEEJ Transactions on Electrical and Electronic Engineering* **5**(2), 256–257.

Yavuz, G., Aydin, D. & Stützle, T. (2016), Self-adaptive search equation-based artificial bee colony algorithm on the CEC 2014 benchmark functions, *in* 'Evolutionary Computation (CEC), 2016 IEEE Congress on', pp. 1173–1180.

Yazdani, D., Nabizadeh, H., Kosari, E. M. & Toosi, A. N. (2011), Color quantization using modified artificial fish swarm algorithm, *in* 'AI 2011: Advances in Artificial Intelligence', Springer, pp. 382–391.

Yazdani, D., Nadjaran Toosi, A. & Meybodi, M. (2011), Fuzzy adaptive artificial fish swarm algorithm, *in* 'AI 2010: Advances in Artificial Intelligence', Vol. 6464 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 334–343.

Yılmaz, S., Ugur Kucuksille, E. & Cengiz, Y. (2014), 'Modified bat algorithm.', *Electronics & Electrical Engineering* **20**(2).

Yong, P. (2011), 'An improved artificial fish swarm algorithm for optimal operation of cascade reservoirs.', *Journal of computers* **6**(4).

Yu, Y., Ma, H. & Zhang, M. (2013), An adaptive genetic programming approach to QoS-aware web services composition, *in* 'Proceedings of the IEEE Congress on Evolutionary Computation', pp. 1740–1747.

Yuan, L., Yuan, C.-A. & Huang, D.-S. (2017), 'FAACOSE: A fast adaptive ant colony optimization algorithm for detecting SNP epistasis', *Complexity* **2017**.

Zahiri, S. (2012), 'Fuzzy gravitational search algorithm an approach for data mining', *Iranian Journal of Fuzzy Systems* .

Zeng, Z., Guan, L., Zhu, Y., Liu, Q. & He, J. (2017), 'A novel image segmentation method based on an improved bacterial foraging optimization algorithm', *Journal of information hiding and multimedia signal processing* **8**(2), 348–357.

Zhang, H., Zhou, J., Fang, N., Zhang, R. & Zhang, Y. (2013), 'An efficient multi-objective adaptive differential evolution with chaotic neuron network and its application on long-term hydropower operation with considering ecological environment problem', *International Journal of Electrical Power & Energy Systems* **45**(1), 60 – 70.

Zhang, J., Chen, W.-N., Zhan, Z.-H., Yu, W.-J., Li, Y.-L., Chen, N. & Zhou, Q. (2012), 'A survey on algorithm adaptation in evolutionary computation', *Frontiers of Electrical and Electronic Engineering* **7**(1), 16–31.

Zhang, J. & Ding, X. (2011), 'A multi-swarm self-adaptive and cooperative particle swarm optimization', *Engineering applications of artificial intelligence* **24**(6), 958–967.

Zhang, X., Zhang, R., Wang, J. & Wang, L. (2018), 'An adaptive particle swarm optimization algorithm based on aggregation degree', *Recent Advances in Electrical & Electronic Engineering* **11**, 1–6.

Zhong, Y., Zhao, L. & Zhang, L. (2014), 'An adaptive differential evolution endmember extraction algorithm for hyperspectral remote sensing imagery', *Geoscience and Remote Sensing Letters, IEEE* **11**(6), 1061–1065.

Zielinski, K. & Laur, R. (2007), Adaptive parameter setting for a multi-objective particle swarm optimization algorithm, *in* 'Evolutionary Computation, 2007. CEC 2007. IEEE Congress on', pp. 3019–3026.