An enhanced multi-objective artificial bee colony algorithm with non-dominated sorting strategy

Hamid Bouali[1], Bachir Benhala[2], Mohammed Guerbaoui[1]
[1]High School of Technology, Moulay Ismail University, Meknes, Morocco
[2]Faculty of Science Dhar El Mahrez, Sidi Mohamed Ben Abdellah University, Fez, Morocco

Article Info

Article history:

Received Oct 21, 2023
Revised Nov 27, 2023
Accepted Dec 25, 2023

Keywords:

Crowding distance
Greedy selection
Multi-objective algorithms
Multi-objective metrics
Nondominated sorting
ZDT benchmark

ABSTRACT

This paper presents an improved metaheuristic technique inspired by the foundational concepts of the artificial bee colony (ABC) algorithm adapted to deal with multi-objective optimization challenges. Our approach combines the main ideas of ABC with a non-dominated sorting strategy including aspects of Pareto dominance, crowding distance, and greedy selection method. Furthermore, the chosen non-dominated solutions are archived in a repository with a static size. The presented approach, multi-objective artificial bee colony (MOABC), is compared to other state-of-the-art algorithms including the non-dominated sorting genetic algorithm II (NSGA II) and the multi-objective particle swarm optimization (MOPSO). MOABC and selected algorithms from the literature are applied to five zitzler-deb-thiele (ZDT) Multi-objective benchmark functions. Then three key metrics are employed for performance evaluations: generational distance (GD), spread (SP), and hypervolume (HV). The simulation results suggest that the proposed method is competitive and presents an effective choice for tackling multi-objective optimization problems.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Hamid Bouali
High School of Technology, Moulay Ismail University
Meknes 50070, Morocco
Email: ham.bouali@edu.umi.ac.ma

1. INTRODUCTION

In various disciplines, such as engineering and computer science, professionals often encounter multi-objective optimization challenges that require balancing several goals simultaneously [1], [2]. These problems, rather than having a single solution, necessitate exploring a range of optimal outcomes known as a paret front. Over the last ten years, many metaheuristic algorithms have been developed to address difficult optimization problems using the ideas of Pareto-optimality and non-dominance [3], [4]. Notable contributions to this field include Karaboga's artificial bee colony (ABC), introduced in 2005 [5], [6], and Kennedy and Eberhart's particle swarm optimization (PSO),was developed in 1995 [7]. Other significant developments are price's differential evolution (DE) in 1996 and 1997 [8], [9], Holland's genetic algorithm (GA) from the 1960s and 1970s [10], and the ant colony optimization algorithm by Dorigo and Gambardella, first proposed in 1997 and further refined in 2006 [11]–[13]. The effectiveness of these algorithms is assessed using specific metrics that consider factors like solution diversity and the range they cover on the Pareto front [14].

The ABC algorithm draws inspiration from the foraging behavior of honeybees by which real bees discover food sources and relay this information to other bees. It stands out in single-objective optimization due to its simplicity, ease of implementation, minimal control parameters, and high speed of convergence. It is extensively utilized in many fields. In light of the advantageous qualities inherent in the ABC algorithm, we present a novel enhancement to this algorithm, tailored for multi-objective problems. Our MOABC algorithm combines a non-dominated sorting strategy with a dedicated archive for elite solutions, to improve the Pareto front exploration. To assess the effectiveness of the MOABC, we conducted an evaluation of the performance of the MOABC algorithm with that of MOPSO and non-dominated sorting genetic algorithm II (NSGA-II) on a set of five zitzler-deb-thiele (ZDT) Multi-objective benchmark functions.

The structure of the paper is organized as follows: section 2 provides an overview of the concepts related to multiobjective optimization problems (MOPs) and Pareto front metrics. Section 3 introduces the details of the MOABC algorithm. Section 4 outlines the experimental procedures. Section 5, the findings and discussion are provided. Finally, section 6 concludes the paper with our findings and future work.

2. AN OVERVIEW OF MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

Multi-objective optimization problems (MOOPs) are a class of optimization problems involving simultaneous optimization of two or more conflicting objectives. In contrast to optimization issues focused on a single objective, MOOPs possess a set of optimal solutions referred to Pareto front [15]. In multi-objective optimization, the concept of Pareto optimality is crucial: a solution is considered optimal if no other solution can advance one goal without worsening at least one other goal. Formally, a formulation of a multi-objective optimization problem can be expressed as follows:

– A collection of decision variables, $x = (x_1, x_2, \dots, x_n)$.
– A series of objective functions denoted as ,$\mathrm{F}(x) = (f_1(x), f_2(x), \dots, f_k(x))$, where each $f_i(x)$ is a function that maps the decision variable space to the real numbers.
– A group of constraints, $G(x) = (g_1(x), f_2(x), \dots, g_m(x))$,where each $g_j(x)$ represents a boundary condition within the space of decision variables.

The objective is to identify the collection of decision variables, denoted as x, such that:

– Satisfy the constraints: $g_i(\vec{x}) \leq 0$ *,i= 1,2... m.*
– Optimize the vector of objective functions $\mathrm{F}(x)$, typically by finding the optimal solutions.

The concept of Pareto optimality states that given vectors $\vec{\mathrm{u}} = (\mathrm{u}_1, \mathrm{u}_2, \dots, \mathrm{u}_\mathrm{n})$ and $\vec{\mathrm{v}} = (\mathrm{v}_1, \mathrm{v}_2, \dots, \mathrm{v}_\mathrm{n})$ in the multidimensional space R^n, we say that vector $\vec{\mathrm{u}}$ a is in a position of dominance over $\vec{\mathrm{v}}$ when each element of $\vec{\mathrm{u}}$ is less than or equal to the corresponding element of $\vec{\mathrm{v}}$, symbolically $\mathrm{u}_\mathrm{i} < \mathrm{v}_\mathrm{i}$ for every i from 1 to n, with at least one element being strictly less, ensuring $\vec{\mathrm{u}} \neq \vec{\mathrm{v}}$.

An element x^* from the feasible set X is considered to be in a state of Pareto optimality if there exists no other element x in X for which the function F(x) is dominant over F(x∗). Solutions satisfying Pareto optimality are alternatively known as efficient, non-dominated, or non-inferior [16]. The collection of all such Pareto-optimal solutions is referred to as the Pareto set, symbolized as PS. The corresponding outcomes of the objective function F, $\{F(x) x \in PS\}$, form what is known as the Pareto frontier, and this concept is visually exemplified in Figure 1.

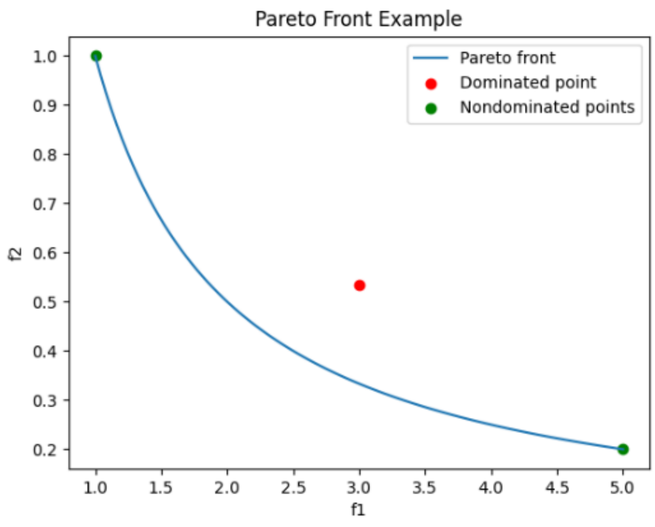

Figure 1. Pareto frontier illustration for multi-objective minimization analysis

3. MULTI-OBJECTIVE ARTIFICIAL BEE COLONY

In 2005, Karaboga proposed the fundamental artificial bee colony algorithm based on the foraging behavior of bees. This algorithm features three types of bees: employed, onlooker, and scout. Each food source position represents a solution to the problem being optimized and is found by employed bees. The onlooker bees assess the quality of the food source positions received, and the scout bee intervenes when a food source cannot be optimized by searching randomly for a new food source position [17]. The MOABC algorithm introduced in this paper begins with an initialization phase that randomly generates the initial positions of the food sources. An archive is utilized to store the non-dominated solutions identified during the selection process.

The refinement of existing solutions occurs during the employed bee phase. The process of generating new solutions can be described by (1):

$$\mathrm{sol}_i^{j^*} = \mathrm{food}_i^j + \mathrm{rand}[-1;1] * (\mathrm{food}_i^j - \mathrm{food}_k^j) \tag{1}$$

where,

$$\mathrm{i} \neq \mathrm{k}\,; \mathrm{i} \in (1,2,\dots,\mathrm{n}).$$

$food_i^j$ denotes the current solution, food_k^j represents the neighbor solution and $\mathrm{sol}_i^{j^*}$ becomes the potential new solution.
n is the number of employed bees.
j, k are selected randomly.
$\mathrm{rand}[-1;1]$ is a randomly chosen number within the [-1;1] interval.

The current solution and the candidate solution are compared, and the greedy selection method evaluates and determines which is better. The trial counter increases if the current solution maintains its position. The counter resets to zero if the candidate solution is the new one. After that, every solution is compared with every member in the archive. This is an important step since it updates the archive, making sure that it only has the best and most recent solutions.

In the onlooker bees phase, the archive members are involved in the process of improving other archive members. This improvement process is represented by (2):

$$\mathrm{sol}_i^{\mathrm{rp}^*} = \mathrm{food}_i^{\mathrm{rp}} + \mathrm{rand}[-1;1] * (\mathrm{AR}_k^{\mathrm{rp}} - \mathrm{food}_i^{\mathrm{rp}}) \tag{2}$$

where,

$\mathrm{i} \in (1,2,\dots,\mathrm{FN})$, FN: The quantity of food is equivalent to half of the colony size.
$\mathrm{k} \in (1,2,\dots,\mathrm{n})$, Is chosen randomly from the archive, where n represents the size of the archive.
rp is chosen at random from the archive.

During this phase, a candidate solution is generated using two archive members: the current one and a neighboring one. The selection of the neighbor involves considering the crowding distance values of all archive members, with the one having the lowest crowding distance value being chosen. In the scout bee phase, unlike employed bees that focus on improving known solutions, the scout bees are tasked with investigating unexplored regions of the problem space. This exploration is random and independent, meaning scout bees do not use information from other bees. Their job is to randomly identify new food sources, without prior knowledge from existing solutions. The random solution is generated by the following (3):

$$\mathrm{food}_i = \mathrm{lb} + \mathrm{rand}[1;\mathrm{d}] * (\mathrm{ub} - \mathrm{lb}) \tag{3}$$

where,

lb is the lower bound of parameters.
ub is the upper bound of the parameters.
d: the dimension of the problems.

During this phase, the trial counts for each food source are tracked. A new site is identified for any food source that exceeds a certain limit. Moreover, the procedure allows only a single scout bee in each cycle. The method is designed to stop after a set number of evaluations. Once this stopping criterion is met, the algorithm concludes its operations and returns the current archive as its output. The flowchart presented in Figure 2 and pseudocode of the MOABC algorithm presented in Pseucode 1, respectively, which is composed of five main parts: initialization, employed bees, onlooker bees, scout bees.

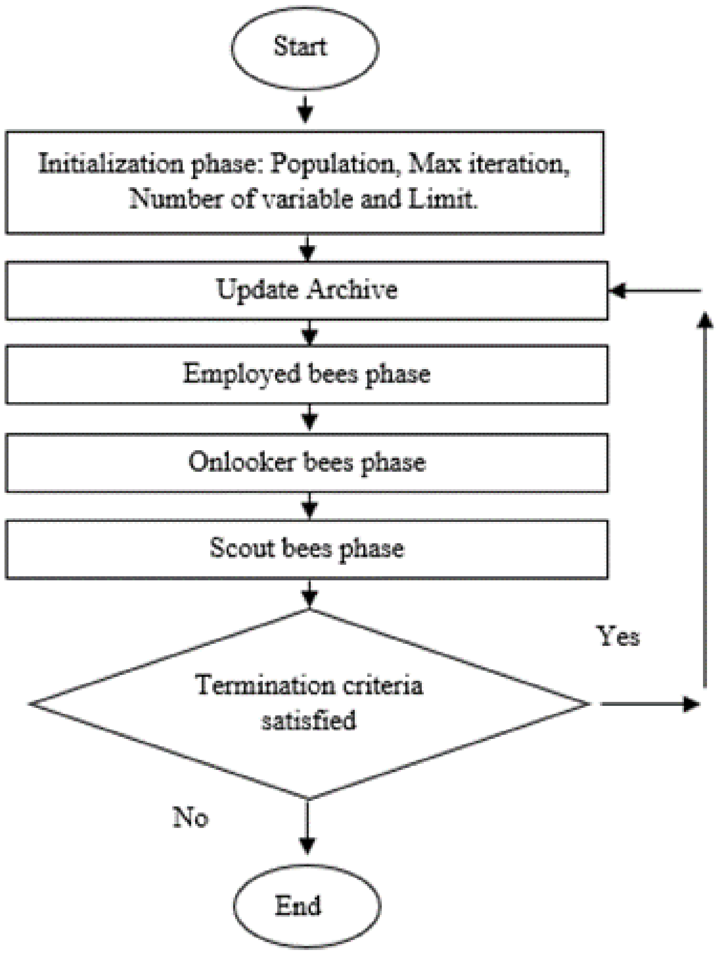
Start
Initialization phase: Population, Max iteration, Number of variable and Limit.
Update Archive
Employed bees phase
Onlooker bees phase
Scout bees phase
Termination criteria satisfied
Yes
No
End

Figure 2. Flowchart of the MOABC algorithm

Pseucode 1. MOABC algorithm

```
Begin
Initialization phase ()
Population, MaxIteration, Number of variables, and Limit.
Employed Bees phase ()
 For i=1 to FoodNumber
    Select a parameter k randomly.
    Select a parameter j randomly
    Generate the new solution by the equation (1)
% Evaluate the new solution using the greedy selection method
      If the new food source position dominates the old one
       Update the position.
      Else
       Increment the trial by one.
     End
     update the Archive.
End

Onlooker bee phase ()
  Evaluate the quality of food source positions found by the employed bee.
  For i=1 to FoodNumber
     Select a parameter k randomly.
     Select a parameter rp randomly from the archive
     Generate the new solution by equation (2)
% Evaluate the new solution using the greedy selection method.
```

```
     If the new food source position dominates the old one
        Update the position.
     Else
        Increment the trial by one.
     End
update the Archive.
  End

Scout Bees phase ()
  For i=1 to FoodNumber
        If I have the maximum the trial
          Reset setting the trial value to zero.
          Generate a random solution by equation (3)
        End
  End
End
```

4. EXPERIMENTAL PROCEDURES

This section presents a comparative analysis of the proposed MOABC algorithm's performance. For this study, two popular and advanced multi-objective algorithms were selected for comparison: NSGA and MOPSO. Additionally, this analysis includes the metrics utilized and highlights some key features of the benchmark test functions in the competition problem.

4.1. Overview of selected algorithms for comparison

4.1.1. Non-dominant sorting genetic algorithm II

The technique known as NSGA-I) was introduced by Deb *et al.* [18], [19] and integrates elite and diversity-preserving mechanisms. NSGA-II initiates by generating a parent population P0 through a random process. These solutions are then organized into different non-domination fronts, with each solution assigned a fitness level corresponding to its non-domination front.

The initial front comprises the best solutions, while subsequent fronts represent progressively inferior solutions. Binary tournament selection, utilizing the crowded comparison criterion, is employed, along with crossover and mutation operators, to form the initial child population Q0 of size N. For t>1, the NSGA-II approach proceeds as follows:

– Combining the parent population Pt with the child population Qt.
– The population Rt = Pt U Qt is sorted using the non-domination method, with a size of 2N.
– A new parent population Pt+1 is created by selecting solutions from the first front until the population size is greater than N.
– The crowding distance Fi (for all non-dominated fronts of Rt) is calculated.
– A new population Qt+1 is generated using selection, crossover, and mutation.

4.1.2. Multi-objective particle swarm optimization

In 1995, Kennedy and Eberhart introduced the PSO technique, drawing inspiration from the flock choreography of birds [20]. To address multi-objective optimization problems, Coello *et al.* [20] introduced the MOPSO technique, which applies the Pareto dominance theory to extract non-dominated solutions and an elitist selection by crowding distance factor. The algorithm also stores the found solutions in an archive, and an adaptive grid manages the archive and guides the update of particles [21]. In each iteration, the particle's velocity and position are determined using (4) and (5).

$$\mathrm{V}_{\mathrm{id}}^{\mathrm{n+1}} = \omega.\mathrm{v}_{\mathrm{id}}^{\mathrm{n}} + \mathrm{c}_1.\mathrm{rand}_1^{\mathrm{n}}(\mathrm{pbest}_{\mathrm{id}}^{\mathrm{n}} - \mathrm{X}_{\mathrm{id}}^{\mathrm{n}}) + \mathrm{c}_2.\mathrm{rand}_2^{\mathrm{n}}(\mathrm{gbest}_{\mathrm{id}}^{\mathrm{n}} - \mathrm{X}_{\mathrm{id}}^{\mathrm{n}}) \quad (4)$$

$$X_i^{n+1} = X_i^n + V_i^{n+1} \quad (5)$$

Where, i represents the particle index (1, 2, ..., N), N is the swarm size, d represents the search space dimension (1, 2, ..., D), and n is the iteration number. $\mathrm{V}_{\mathrm{id}}^{\mathrm{n}}$ and $\mathrm{V}_{\mathrm{id}}^{\mathrm{n+1}}$ represent the d-dimensional velocity of particle i in iterations n and n+1, respectively. $\mathrm{X}_{\mathrm{id}}^{\mathrm{n}}$ and $\mathrm{X}_{\mathrm{id}}^{\mathrm{n+1}}$ represent the d-dimensional positions of particle i in iterations n and n+1, respectively. $\mathrm{pbest}_{\mathrm{id}}^{\mathrm{n}}$ and $\mathrm{gbest}_{\mathrm{id}}^{\mathrm{n}}$ represent the personal best and global best of particle i in iteration n, respectively. rand1 and rand2 represent values selected randomly from the range between 0 and 1. c1 and c2 are the cognitive weight and social weight, respectively, and they are usually set to 2.

4.2. Performance metrics

To assess the performance of various multi-objective algorithms for a specific problem, each algorithm was executed for an equal number of generations. The resulting solutions, referred to as Pareto front approximations, were then compared using two quality measures: solution, which determines the similarity of the solution to the true Pareto front, and solution diversity, which evaluates the distribution of points in the solution. We utilized the three most used metrics identified in a recent literature survey. These metrics, ranked in order of popularity, include generational distance (GD), spacing (SP), and hypervolume (HV) [22], [23]. The definitions of these metrics are as follows:

4.2.1. Generational distance

The GD is a measure of how far the non-dominated solutions obtained by an algorithm are from these at the true Pareto front. This Metric is given by (6):

$$\mathrm{GD} = \frac{\left(\left(\sum_{i=1}^{|Q|} d_i^p\right)^{1/p}\right)}{|Q|} \tag{6}$$

For p=2, the parameter d_i is the Euclidean distance between each point in $PF^*$and the closest point in PF denoted as:

$$d_i = \min_{k=1}^{|P^*|}\left(\sum_{m=1}^{M}\left(PF_m^i - PF^{*i}_m\right)^2\right)^{1/2} \tag{7}$$

4.2.2. Spacing

The SP is the distance measure between the obtained solutions and describes the distribution of non-dominated solutions using a specific algorithm. This criterion can show how the solutions were distributed among each other and can be defined as:

$$\mathrm{Sp} = \sqrt{\frac{1}{|Q|}\sum_{i=1}^{|Q|}\left(d_i - \bar{d}\right)^2} \tag{8}$$

where,

d_i is the Euclidean distance and $\bar{d}$ is the mean value of all d_i:

$$\bar{d} = \sum_{i=1}^{|Q|}\frac{d_i}{|Q|} \tag{9}$$

4.2.3. Hypervolume

The HV metric, also referred to as the S-metric or Lebesgue measure is used in this research due to its ability to provide a single number that expresses convergence as well as variety. With the help of a reference point (r*), ($r^*= r_1^*. . ., r_m^*$) in Ω, this metric is used to calculate the normalized volume inside the space of objective Ω that is covered by the obtained Pareto front (P^*). A hypercube c_i is computed for every solution, i ∈ P^*given i and the reference point r. The hypervolume HV is calculated as (10).

$$HV = volume\left(\bigcap_{I=1}^{|P|} c_i\right) \tag{10}$$

4.3. Benchmark test problems

In this paper, we utilize a set of benchmark functions to assess and validate the performance of different algorithms through comparison. These benchmark functions are designed for multi-objective optimization problems and aim to simulate real-world scenarios. We specifically focus on five well-known problems from literature, known as the ZDT benchmark functions [24].

These problems are unconstrained and distinguished by well-defined mathematical models, including convex or concave characteristics. Furthermore, they have different shaped Pareto optimum borders. Table 1 provides mathematical models and characteristics associated with these test functions.

Table 1. Mathematical models and characteristics of five ZDT test functions

Function name	Mathematical model formula	Search domain/function characteristic
ZDT1	$f_1(x) = x_1$ $f_2(x) = g(x)h(f_1(x), g(x))$ $g(x) = 1 + \frac{9}{(n-1)}\sum_{i=2}^{n} x_i$ $h(f_1(x), g(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}}$	$0 \leq x_i^* \leq 1$; $1 \leq i \leq 30$. Has a convex Pareto optimal front.
ZDT2	$f_1(x) = x_1$ $f_2(x) = g(x)h(f_1(x), g(x))$ $g(x) = 1 + \frac{9}{(n-1)}\sum_{i=2}^{n} x_i$ $h(f_1(x), g(x)) = 1 - \left(\frac{f_1(x)}{g(x)}\right)^2$	$0 \leq x_i^* \leq 1$; $1 \leq i \leq 30$. Has a concave Pareto optimal front.
ZDT3	$f_1(x) = x_1$ $f_2(x) = g(x)h(f_1(x), g(x))$ $g(x) = 1 + \frac{9}{(n-1)}\sum_{i=2}^{n} x_i$ $h(f_1(x), g(x)) = 1 - \sqrt{\left(\frac{f_1(x)}{g(x)}\right)} - \left(\frac{f_1(x)}{g(x)}\right)\sin(10\pi f_1(x))$	$0 \leq x_i^* \leq 1$; $1 \leq i \leq 30$. Has several disconnected Pareto optimal front. .
ZDT4	$f_1(x) = x_1$ $f_2(x) = g(x)h(f_1(x), g(x))$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}(x_i{}^2 - 10cos(4\pi x_i))$ $h(f_1(x), g(x)) = 1 - \sqrt{\left(\frac{f_1(x)}{g(x)}\right)}$	$0 \leq x_i^* \leq 1$; $1 \leq i \leq 30$. Has many local fronts, single global convex front
ZDT6	$f_1(x) = 1 - e^{4x_1}\sin^6(6\pi x_1)$ $f_2(x) = g(x)h(f_1(x), g(x))$ $g(x) = 1 + \left(\frac{9}{(n-1)}\sum_{i=2}^{n} x_i\right)^{1/4}$ $h(f_1(x), g(x)) = 1 - \left(\frac{f_1(x)}{g(x)}\right)^2$	$0 \leq x_i^* \leq 1$; $1 \leq i \leq 30$. Non-uniform distribution, non-convex front

5. RESULTS AND DISCUSSION

In this section, we tested the suggested technique's effectiveness in solving unconstrained multi-objective mathematics ZDT problems, using performance metrics. We used MATLAB software to code the algorithm. For our work, we utilized a computer with an Intel Core i5 processor running at 1.6 GHz and 4 GB DDR3 RAM. The problems allowed us to assess how well multi-objective optimizers handle non-convex and non-linear problems.

We also conduct a comparison between MOABC, NSGA II, and MOPSO algorithms and present the best result from a set of Pareto optimal solutions. Table 2 provides the initial control parameters for each algorithm. Each experiment was carried out with a maximum of 100 iterations and 100 populations. The proposed algorithm is tested in five distinct case studies, as depicted in Table 1. As shown in Figure 3, the statistical results for various algorithms applied to multi-objective benchmark functions are presented. It was observed that all algorithms successfully converged towards the Pareto front, except for MOPSO and NSGA-II on certain ZDT problems.

Table 3 provides a comparison of the GD, SP value, and HV values among the three algorithms. The results indicate that the multi-objective artificial bee colony algorithm demonstrates exceptional convergence accuracy, outperforming the other algorithms in solving all ZDT benchmark functions, as shown in Table 3. In terms of the SP values, all algorithms consistently achieve values lower than 0.1 across all ZDT problems. Notably, the MOABC algorithm stands out as the top performer on all ZDT problems. Regarding the HV results, the MOABC algorithm consistently exhibits strong performance across all ZDT benchmark functions.

Table 2. Control parameters of all algorithms

Parameters	MOABC	MOPSO	NSGA II
Max iteration	100	100	100
Population size	100	100	100
Dimension of the solution space (D)	4	4	4
Number of employed bees (%)	50	-	-
Number of onlookers bees (%)	50	-	-
Number of scouts	1	-	-
Limit (=number of onlookers bees*D)	100	-	-
Size of the external archive (sizeAR)	25	25	-
Weight damping rate (Wdamp)	-	0.99	-
Weight factor (w)	-	0.5	-
Acceleration coefficient ($\mathbf{0.5 \leq C1 \leq 2.5}$)	-	1	-
Acceleration coefficient ($\mathbf{0.5 \leq C2 \leq 2.5}$)	-	2	-
Number of grids per dimension (NGrid)	-	7	-
Inflation rate ($\boldsymbol{\alpha}$)	-	0.1	-
Leader selection pressure ($\boldsymbol{\beta}$)	-	2	-
Deletion selection pressure ($\boldsymbol{\gamma}$)	-	2	-
Mutation rate ($\boldsymbol{\mu}$)	-	0.1	-
Crossover ratio (pc)	-	-	0.8
Mutation ratio (pm)	-	-	0.3

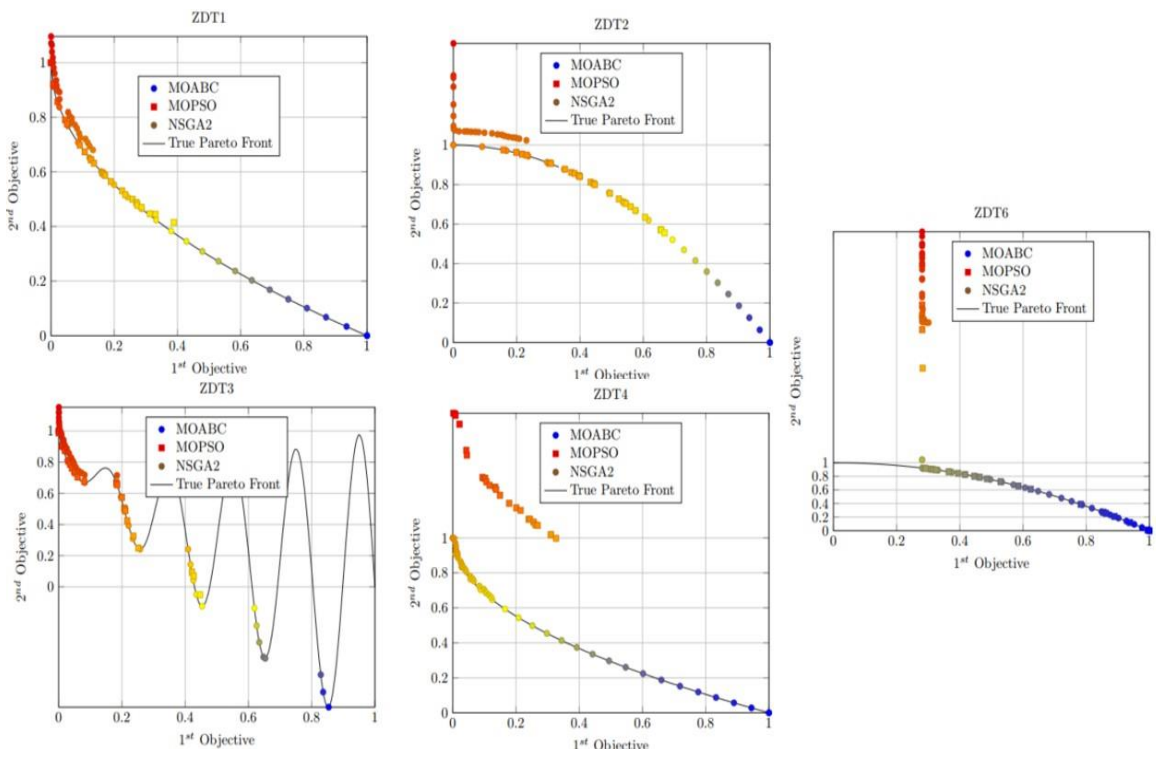

Figure 3. True and obtained Pareto fronts by MOABC, MOPSO, and NSGA-II algorithms on 5 ZDT test problems

As depicted in Figures 4 to 6, the box plots display the results of the GD, SP, and HV metrics, respectively, for the three algorithms. These box plots provide a visual representation of the data distribution. As can be seen from Figure 4, the results of the box plots for the GD metrics on the ZDT benchmarks indicate the performance of three algorithms: MOABC, MOPSO, and NSGA-II.

Overall, MOABC consistently performed better than MOPSO and NSGA-II in terms of their closeness to the true Pareto front, as indicated by the lower values of the GD metric. Specifically, MOABC achieved the lowest best values on most of the ZDT benchmarks, indicating its ability to generate solutions that are closer to the optimal Pareto front. While MOPSO and NSGA-II had higher best values, they still achieved reasonable performance in approaching the true Pareto front, especially on certain benchmarks such as ZDT3.

As shown in Figure 5, the box plots depict the results of the SP metrics on the ZDT benchmarks. The plots indicate that both MOABC and MOPSO consistently yield the smallest spacing outcomes, indicating a more diverse and evenly distributed set of solutions. On the other hand, NSGA-II tends to show higher spacing values, implying a less uniformly distributed set of solutions.

In Figure 6, the box plot analysis of the HV metrics provides insights into the algorithmic performance based on the best-estimated hypervolume values. Across the ZDT benchmarks, MOABC consistently exhibits the highest hypervolume values among the three algorithms. It outperforms both MOPSO and NSGA-II in terms of maximizing the hypervolume metric, indicating superior coverage of the Pareto front. This suggests that MOABC can generate diverse and well-distributed solutions, offering better trade-offs between multiple objectives.

Table 3. The statistical results of the performance metrics for 5 ZDT function of the algorithms

		Algorithms								
Test function		MOABC this work	MOPSO [25]	NSGA II [26]	MOABC this work	MOPSO [25]	NSGA II [26]	MOABC this work	MOPSO [25]	NSGA II [26]
		GD metric			SP metric			HV metric		
ZDT1	Best	8.52E-04	1.22E-03	2.45E-03	8.52E-04	2.65E-03	1.43E-02	7.20E-01	6.19E-01	
	Mean	8.52E-04	1.83E-03	2.59E-03	8.52E-04	5.56E-03	1.63E-02	7.17E-01	5.93E-0,1	
	Worst	1.06E-03	2.44E-03	2.72E-03	1.06E-03	8.48E-03	1.83E-02	7.14E-01	5.68E-01	
ZDT2	Best	8.54E-04	1.06E-03	2.21E-03	2.69E-03	2.57E-03	2.84E-03	4.56E-01	3.66E-01	
	Mean	8.91E-04	5.91E-03	4.47E-03	2.75E-03	2.73E-02	3.14E-02	4.52E-01	3.22E-01	
	Worst	9.29E-04	1.07E-02	6.72E-03	2.81E-03	5.20E-02	5.99E-02	4.49E-01	2.36E-01	
ZDT3	Best	4.93E-03	2.74E-03	3.72E-03	1.84E-02	7.91E-03	2.13E-02	8.33E-01	8.35E-01	
	Mean	5.02E-03	2.74E-03	3.72E-03	1.93E-02	7.91E-03	2.13E-02	8.32E-01	8.28E-01	
	Worst	5.11E-03	3.52E-03	5.97E-03	2.01E-02	1.16E-02	4.56E-02	8.29E-01	7.93E-01	
ZDT4	Best	1.08E-03	8.19E-02	5.84E-03	4.01E-03	1.36E-01	5.17E-02	7.20E-01	1.85E-01	
	Mean	1.08E-03	8.19E-02	5.84E-03	4.01E-03	1.36E-01	5.17E-02	7.17E-01	1.17E-01	
	Worst	1.17E-03	8.28E-02	8.15E-03	4.64E-03	1.45E-01	7.77E-02	7.15E-01	8.29E-02	
ZDT6	Best	1.03E-03	8.02E-02	2.32E-01	2.72E-03	3.92E-01	3.13E-01	4.26E-01	4.18E-01	
	Mean	1.03E-03	8.02E-02	2.38E-01	2.72E-03	3.92E-01	3.13E-01	4.23E-01	4.11E-01	
	Worst	1.16E-03	9.01E-02	2.45E-01	3.21E-03	4.40E-01	5.43E-01	4.19E-01	4.01E-01	

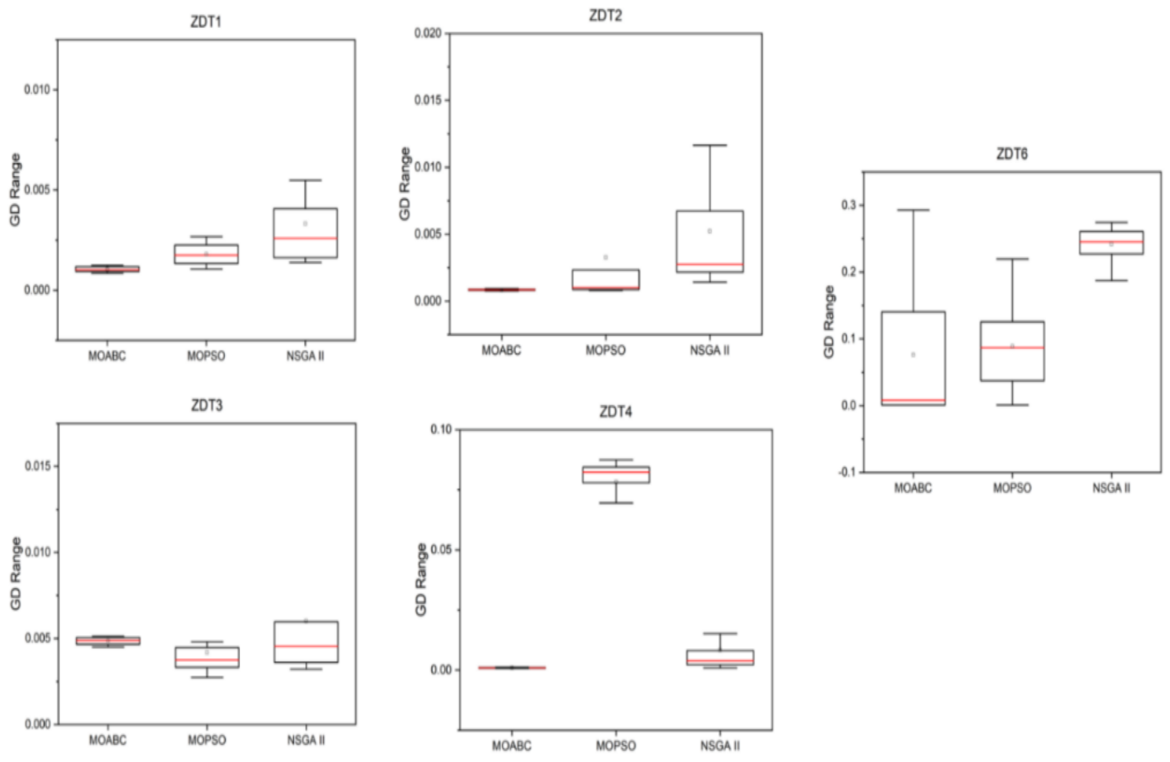

Figure 4. Box plot for the statistical results for GD on 5 ZDT test problems

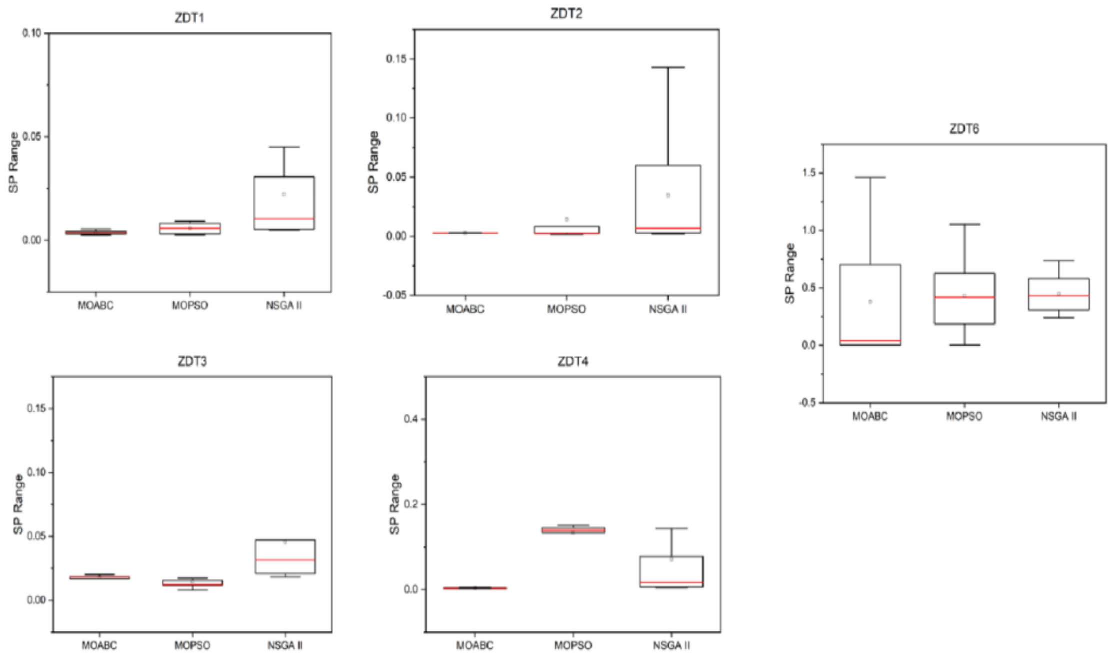

Figure 5. Box plot for the statistical results for SP on 5 ZDT test problems

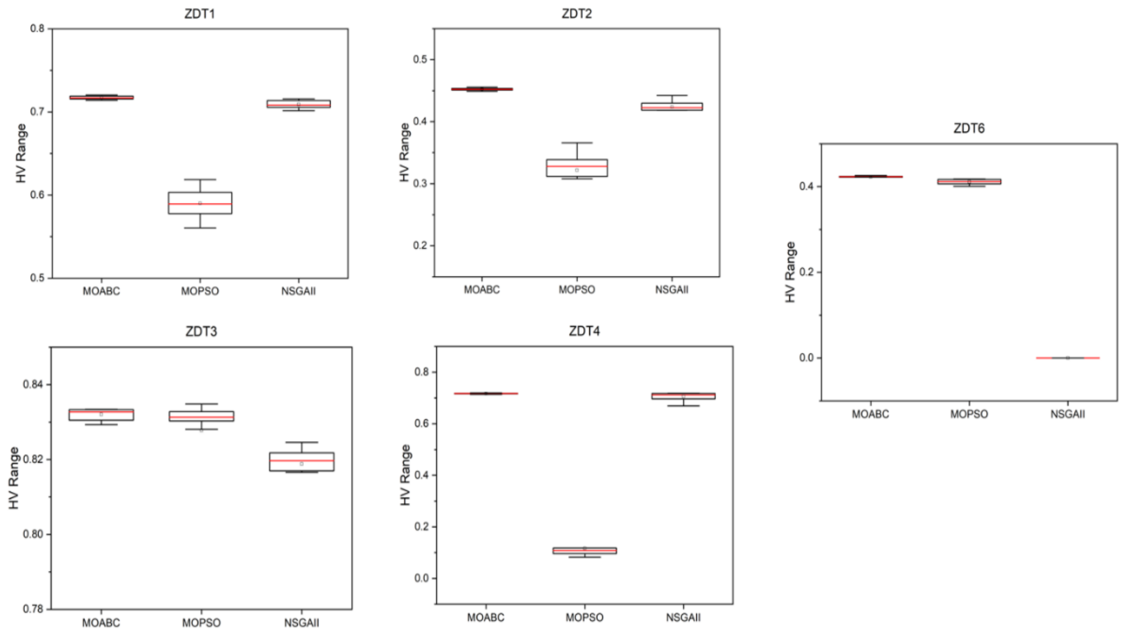

Figure 6. Box plot for the statistical results for HV on 5 ZDT test problems

6. CONCLUSION

This paper introduces an adapted version of the Artificial Bee Colony algorithm, tailored for addressing multi-objective optimization challenges. Our modified algorithm combines a non-dominated sorting strategy, including aspects of Pareto dominance, crowding distance, and a greedy selection method. This ensures a diverse population and effective convergence to the true Pareto optimal front. We evaluated the effectiveness of our MOABC algorithm using five distinct test functions, assessing its performance with three established metrics from the literature. The statistical analysis demonstrates that our algorithm achieves superior convergence towards the Pareto front and higher HV values compared to existing methods, establishing it as a top performer in this field. These results highlight the MOABC's efficiency and its potential as a preferred choice for tackling multi-objective optimization problems. Future research could focus on enhancing the MOABC framework and exploring its potential applications in the domain of analog circuit design.

REFERENCES

[1] A. Ambrisi, M. De Magistris, and R. Fresa, “Multi-objective optimization based design of high efficiency DC-DC switching converters,” *International Journal of Power Electronics and Drive Systems (IJPEDS)*, vol. 7, no. 2, p. 379, Jun. 2016, doi: 10.11591/ijpeds.v7.i2.pp379-386.

[2] H. Bouali, B. Benhala, and M. Guerbaoui, “Multi-objective optimization of CMOS low noise amplifier through nature-inspired swarm intelligence,” *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 12, no. 5, pp. 2824–2836, Oct. 2023, doi: 10.11591/eei.v12i5.5512.

[3] B. Benhala, P. Pereira, and A. Sallem, *Focus on swarm intelligence research and applications*. Nova Science Publishers, Inc., 2017.

[4] A. Sallem, B. Benhala, M. Kotti, M. Fakhfakh, A. Ahaitouf, and M. Loulou, “Simulation-based multi-objective optimization of current conveyors: Performance evaluations,” in *7th International Conference on Design & Technology of Integrated Systems in Nanoscale Era*, IEEE, May 2012, pp. 1–5. doi: 10.1109/DTIS.2012.6232948.

[5] B. Benhala, H. Bouyghf, A. Lachhab, and B. Bouchikhi, “Optimal design of second generation current conveyors by the artificial bee colony technique,” in *2015 Intelligent Systems and Computer Vision (ISCV)*, IEEE, Mar. 2015, pp. 1–5. doi: 10.1109/ISACV.2015.7106172.

[6] D. Karaboga and B. Akay, “A comparative study of artificial bee colony algorithm,” *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, Aug. 2009, doi: 10.1016/j.amc.2009.03.090.

[7] D. Wang, D. Tan, and L. Liu, “Particle swarm optimization algorithm: an overview,” *Soft Computing*, vol. 22, no. 2, pp. 387–408, Jan. 2018, doi: 10.1007/s00500-016-2474-6.

[8] Amritpal Singh and Sushil Kumar, “Differential evolution: an overview,” in *Proceedings of Fifth International Conference on Soft Computing for Problem Solving*, Springer, Singapore, 2016, pp. 209–217. doi: 10.1007/978-981-10-0448-3_17.

[9] S. Abi, B. Benhala, H. Bouyghf, and M. Fakhfakh, “A comparative study between ACO and DE techniques by numerical functions optimization,” in *2019 5th International Conference on Optimization and Applications (ICOA)*, IEEE, Apr. 2019, pp. 1–6. doi: 10.1109/ICOA.2019.8727614.

[10] D. A. Coley, *An Introduction to Genetic Algorithms for Scientists and Engineers*. World Scientific, 1999. doi: 10.1142/3904.

[11] V. Maniezzo and A. Carbonaro, “Ant colony optimization: an overview,” in *Essays and Surveys in Metaheuristics*, 2002, pp. 469–492. doi: 10.1007/978-1-4615-1507-4_21.

[12] B. Benhala *et al.*, “Application of the ACO technique to the optimization of analog circuit performances,” in *Analog Circuits: Applications, Design and Performance*, 2012, pp. 235–255.

[13] B. Benhala, A. Ahaitouf, A. Mechaqrane, and B. Benlahbib, “Multi-objective optimization of second generation current conveyors by the ACO technique,” in *2012 International Conference on Multimedia Computing and Systems*, IEEE, May 2012, pp. 1147–1151. doi: 10.1109/ICMCS.2012.6320142.

[14] C. Audet, J. Bigeon, D. Cartier, S. Le Digabel, and L. Salomon, “Performance indicators in multiobjective optimization,” *European Journal of Operational Research*, vol. 292, no. 2, pp. 397–422, Jul. 2021, doi: 10.1016/j.ejor.2020.11.016.

[15] K. Deb, “Multi-objective optimisation using evolutionary algorithms: an introduction (DOBLE),” in *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*, London: Springer London, 2011, pp. 3–34. doi: 10.1007/978-0-85729-652-8_1.

[16] C. Duddy and A. Piggins, “A foundation for Pareto optimality,” *Journal of Mathematical Economics*, vol. 88, pp. 25–30, May 2020, doi: 10.1016/j.jmateco.2020.02.005.

[17] H. Bouyghf, B. Benhala, and A. Raihani, “Analysis of the impact of metal thickness and geometric parameters on the quality factor-Q in integrated spiral inductors by means of artificial bee colony technique,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 4, p. 2918, Aug. 2019, doi: 10.11591/ijece.v9i4.pp2918-2931.

[18] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II,” 2000, pp. 849–858. doi: 10.1007/3-540-45356-3_83.

[19] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[20] C. A. Coello Coello and M. S. Lechuga, “MOPSO: a proposal for multiple objective particle swarm optimization,” in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC’02 (Cat. No.02TH8600)*, IEEE, pp. 1051–1056. doi: 10.1109/CEC.2002.1004388.

[21] H. Yu, Y. Gao, and J. Wang, “A multiobjective particle swarm optimization algorithm based on competition mechanism and gaussian variation,” *Complexity*, vol. 2020, pp. 1–23, Nov. 2020, doi: 10.1155/2020/5980504.

[22] K. Lwin, R. Qu, and G. Kendall, “A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization,” *Applied Soft Computing*, vol. 24, pp. 757–772, Nov. 2014, doi: 10.1016/j.asoc.2014.08.026.

[23] S. Garcia and C. T. Trinh, “Comparison of multi-objective evolutionary algorithms to solve the modular cell design problem for novel biocatalysis,” *Processes*, vol. 7, no. 6, p. 361, Jun. 2019, doi: 10.3390/pr7060361.

[24] K. Deb, A. Sinha, and S. Kukkonen, “Multi-objective test problems, linkages, and evolutionary methodologies,” in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, New York, NY, USA: ACM, Jul. 2006, pp. 1141–1148. doi: 10.1145/1143997.1144179.

[25] M. K. Heris, “Multi-Objective PSO (MOPSO) in MATLAB,” Yarpiz, [Online]. Available: https://github.com/smkalami/ypea121-mopso. (Accessess: Nov. 18, 2023).

[26] M. K. Heris, “NSGA-II in MATLAB,” Yarpiz, 2023. [Online]. Available: https://github.com/smkalami/ypea120-nsga2. (Accessess: Nov. 18, 2023).

BIOGRAPHIES OF AUTHORS

Hamid Bouali received a B.A. degree in electronics from the Faculty of Sciences of Moulay Ismail University, Meknes, Morocco and a master's degree in Microelectronics from Faculty of Science Dhar El Mahrez of Sidi Mohamed Ben Abdellah University, Fez, Morocco. He has more than 6 years experience in analog integrated circuits layout. He is currently a Ph.D. student in Department of Physics. His research interests are related to design and optimization of ICs area using swarm intelligence techniques. He can be contacted at email: ham.bouali@edu.umi.ac.ma.

Bachir Benhala received a master's degree in Telecommunication Systems and Micro-Electronic, a Ph.D. degree in Electronic and Micro-electronic, from the Faculty of Science and Technology of Fez, Morocco; He is full Professor at Sidi Mohamed Ben Abdellah University in Fez, Morocco. His research interests include analog design automation, digital/analog VLSI architecture, telecommunication and RF circuits, embedded systems, signal processing, applied optimization techniques. He can be contacted at email: bachir.benhala@usmba.ac.ma.

Mohammed Guerbaoui is a Professor of Electrical Engineering and member of Modelling, Materials and Control of Systems team at Laboratory of Computer Engineering and Intelligent Electric Systems in the Higher School of Technology, Moulay Ismail University, Meknes, Morocco. He received the Ph.D. degree in 2014, from Faculty of Sciences, Moulay Ismail University, Meknes. His research interests are mainly focussed on power quality, electrical drives, fuzzy logic control techniques for greenhouse drip irrigation. He can be contacted at email: m.guerbaoui@est.umi.ac.ma.