STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **13**, February 2025, pp 857–876.
Published online in International Academic Press (www.IAPress.org)

Multi-Objective Design Optimization of Planar Spiral Inductors Using Enhanced Metaheuristic Techniques

Hamid Bouali[1,*], Soufiane Abi[1], Bachir Benhala[2], Mohammed Guerbaoui[1]

[1]*High School of Technology, Moulay Ismail University, Meknes, Morocco*
[2]*Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez, Morocco*

Abstract The study presented in this paper improves the Multi-Objective Artificial Bee Colony (MOABC) method. It evaluates its performance using Generational Distance (GD), Spread (SP), and Hypervolume (HV) metrics on the Zitzler-Deb-Thiele (ZDT) benchmark functions. Subsequently, the improved MOABC method, along with Multi-Objective Particle Swarm Optimization (MOPSO) and the Non-Dominated Sorting Genetic Algorithm II (NSGA-II), is applied to optimize the design of a square planar spiral inductor. The objectives are to maximize the quality factor (Q) and minimize the inductor area (A) simultaneously while maintaining a necessary inductance of $4\,\text{nH}$ at a $2.4\,\text{GHz}$ operating frequency, utilizing $0.13\,\mu\text{m}$ CMOS technology. The optimization findings are verified and confirmed using Advanced Design System (ADS) Momentum, demonstrating the feasibility of multi-objective optimization for integrated inductor design.

Keywords Multi-objective algorithms, Multi-objective metrics, ZDT Benchmark, Integrated inductors

DOI: 10.19139/soic-2310-5070-1873

1. Introduction

Multi-objective problems are a common type of problem that involves optimizing multiple objectives simultaneously [1]. Instead of generating a single solution, these problems aim to provide a set of solutions, known as the Pareto front. Each solution on the Pareto front represents a trade-off between two or more objectives optimized using the same set of variables. Various algorithms have been proposed to solve these problems, and their effectiveness is measured using several metrics. These metrics are real values that assess specific characteristics of a given Pareto front, such as the diversity of solutions or the volume of the area generated by the Pareto front [2]. Using multi-objective metrics, researchers can compare the quality of an observed Pareto solution set obtained by a multi-objective optimization method and select the algorithm with the desired performance. Over the past three decades, metaheuristics have been increasingly used to solve complex and challenging problems in different engineering fields [3]. They have gained popularity rapidly as they have proven to be more ingenious and efficient, providing good approximate solutions within a reasonable time [4], [5], [6].

Recently, multi-objective optimization approaches have been successfully employed in numerous studies to optimize analog and radio-frequency (RF) integrated circuits. In [7], a study was conducted to evaluate the performance of two multi-objective techniques, namely, Multi-Objective Artificial Bee Colony (MOABC) and Multi-Objective Particle Swarm Optimization (MOPSO), for the optimal design of a CMOS Low Noise Amplifier (LNA) cascode with inductive source degeneration. In [8], the Non-Dominated Sorting Genetic Algorithm (NSGA-II) was applied to size a biomedical instrumentation amplifier. A comparative study presented in [9] evaluated the

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

ISSN 2310-5070 (online) ISSN 2311-004X (print)

performance of two evolutionary-based algorithms, NSGA-II and the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D), for automating the analog/RF circuit design process.

In this work, the performance of three well-known metaheuristic-based multi-objective algorithms is compared: Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [10], [11], Multi-Objective Particle Swarm Optimization (MOPSO) [12], [13], and Multi-Objective Artificial Bee Colony (MOABC) [14], [15]. The evaluation uses a set of metrics: Generational Distance (GD), Spread (SP), and Hypervolume (HV) [16], [17]. This study contributes to the field by comparing the performance of these algorithms on a standardized set of problems using various metrics. The findings can help researchers and practitioners select the most suitable algorithm for a specific multi-objective problem, potentially saving time and resources while improving optimization outcomes. Additionally, the paper focuses on optimizing the dimensions of a square planar spiral inductor by applying the MOABC, MOPSO, and NSGA-II algorithms to create a Pareto front that simultaneously maximizes the quality factor (Q) and minimizes the area of the inductor. Simulations performed with Keysight's ADS Momentum simulator provide results that align with the optimization outcomes, confirming the validity of this approach.

The paper is organized as follows: Section 2 provides an overview of multi-objective metaheuristic algorithms and the metrics used to evaluate the Pareto front. Section 3 details the experimental setup, including the methodology employed, the obtained results, and their analysis and discussion. Section 4 describes the planar spiral inductor and presents the experimental results of its optimization. Finally, Section 5 concludes the paper by summarizing the findings, highlighting limitations, and suggesting avenues for future research.

2. Background

This section outlines the multi-objective algorithms chosen for the study, the metrics used, and some key characteristics of the benchmark test functions.

2.1. Multi-objective meta-heuristic algorithms

Multi-objective optimization algorithms are commonly used to optimize multi-objective problems because they can avoid local optima and are versatile enough to handle various issues. They can easily incorporate linear and non-linear inequality and equality constraints. These features make them a preferred choice for solving multi-objective problems [18].

2.1.1. Multi-Objective Artificial Bee Colony

The original Artificial Bee Colony (ABC) algorithm, developed by Karaboga in 2005, is inspired by the intricate foraging behaviors of bee colonies [19]. This algorithm models three types of bees: employed bees, onlooker bees, and scout bees. Employed bees search for potential solutions, represented as food sources within the optimization problem's domain. Onlooker bees assess the quality of these solutions, while scout bees are dispatched to explore new solutions through a random search when current solutions are insufficient. The Multi-Objective Artificial Bee Colony (MOABC) algorithm is structured into five main phases: Initialization, Employed Bee Phase, Onlooker Bee Phase, Scout Bee Phase, and Archive Updating. Detailed explanations of these phases can be found in the referenced literature [20], [21]. The MOABC pseudocode and flowchart are shown in Figures 1 and 2, respectively.

Begin
Initialization phase ()
Number of variables, Population, MaxIteration, and Limit.
Employed Bees phase ()
for *i=1 to FoodNumber* **do**
  Randomly choose a parameter k.
  Randomly choose a parameter j.
  Use equation (1) to obtain a new solution:

$$sol_i^j = food_i^j + \text{rand}[-1,1] \cdot (food_i^j - food_k^j) \tag{1}$$

  if *the new food source position dominates the previous one* **then**
    Update the position.
  end
  else
    Increment the trial by one.
  end
end
Onlooker bee phase ()
Assess the quality of the food source positions discovered by the employed bee.
for *i=1 to FoodNumber* **do**
  Randomly choose the parameter k.
  Randomly choose the parameter rp from the archive.
  Use equation (2) to obtain the new solution.

$$sol_i^{rp} = food_i^{rp} + \text{rand}[-1,1] \cdot (AR_k^{rp} - food_i^{rp}) \tag{2}$$

  if *the new food source position dominates the previous one* **then**
    Update the position.
  end
  else
    Increment the trial by one.
  end
end
Scout Bees phase ()
for *i=1 to FoodNumber* **do**
  if *I have the maximum trial* **then**
    Reset the trial value to zero.
    Generate a random solution by equation (3):

$$food_i = lb + \text{rand}[1,d] \cdot (ub - lb) \tag{3}$$

  end
end
End

Figure 1. Pseudocode of MOABC algorithm.

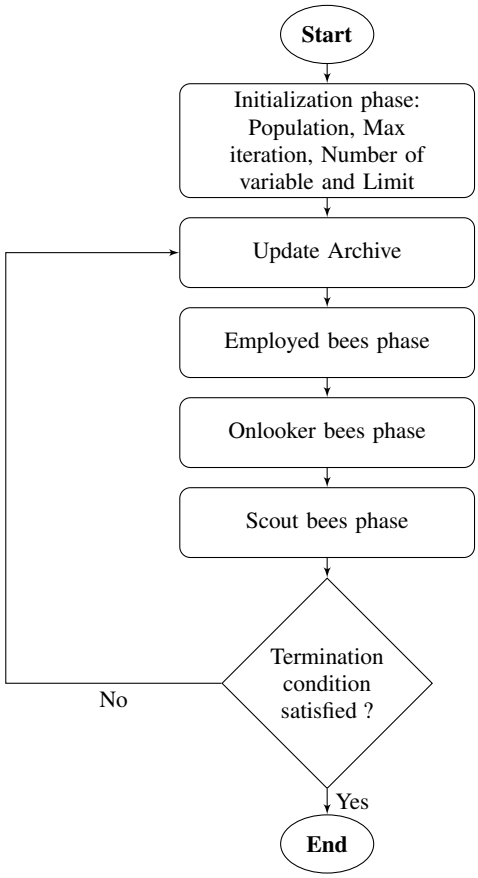


Figure 2. Flowchart of the MOABC algorithm.

The MOABC adjustment process involved adjusting the population size and limit parameters. The population size was optimized to balance the computational demands with the diversity of solutions. The limit parameter, determining how quickly bees abandon exhausted sources, was calibrated through testing to balance exploration and exploitation, which is crucial for preventing premature convergence and ensuring efficient search behavior.

2.1.2. Multi-Objective Particle Swarm Optimization

The particle swarm optimization technique, introduced by Eberhart and Kennedy in 1995 [22], was derived from the coordinated movement of birds in a flock. Coello and Pulido [23] proposed the MOPSO algorithm to tackle multi-objective optimization problems. This technique utilizes the principle of Pareto dominance to identify non-dominated solutions and employs an elitist selection based on the crowding distance factor. The algorithm also saves the discovered solutions in an external archive, and a flexible grid oversees the archive and directs the updating of particles [24]. The position and velocity of each particle [25] are computed in every iteration using the following equations :

$$V_{id}^{(n+1)} = \omega \cdot V_{id}^{n} + c_1 \cdot \text{rand}_1^n(pbest_{id}^n - X_{id}^n) + c_2 \cdot \text{rand}_2^n(gbest_{id}^n - X_{id}^n) \tag{4}$$

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

Where i denotes the index of the particle $(1, 2, \ldots, N)$; N represents the size of the swarm; d denotes the dimension of the search space $(1, 2, \ldots, D)$; and n denotes the iteration number. V_{id}^n and $V_{id}^{(n+1)}$ represent the d-dimensional velocity of particle i in iterations n and $n+1$, respectively. X_{id}^n and $X_{id}^{(n+1)}$ represent the d-dimensional positions of particle i in iterations n and $n+1$, respectively. $pbest_{id}^n$ and $gbest_{id}^n$ represent the personal

best and global best of particle i in iteration n, respectively. $rand_1$ and $rand_2$ are random values between 0 and 1. The cognitive weight denoted as c_1, and the social weight, denoted as c_2, are often assigned a value of 2. The MOPSO pseudocode and flowchart are shown in Figures 3 and 4, respectively.

Begin
Initialization stage
Particle Initialization: Position, Velocity, and Archive.
while *Iteration* $<$ *MaxIteration* **do**
  for *Each Particle* **do**
    Select leader.
    Update Position and Velocity.
    Mutation.
    Evaluation.
    Update pbest.
  end
  Update gbest.
  Update Archive.
  Selection by crowding distance factor.
end
Update Archive.
End

Figure 3. Pseudocode of MOPSO algorithm.

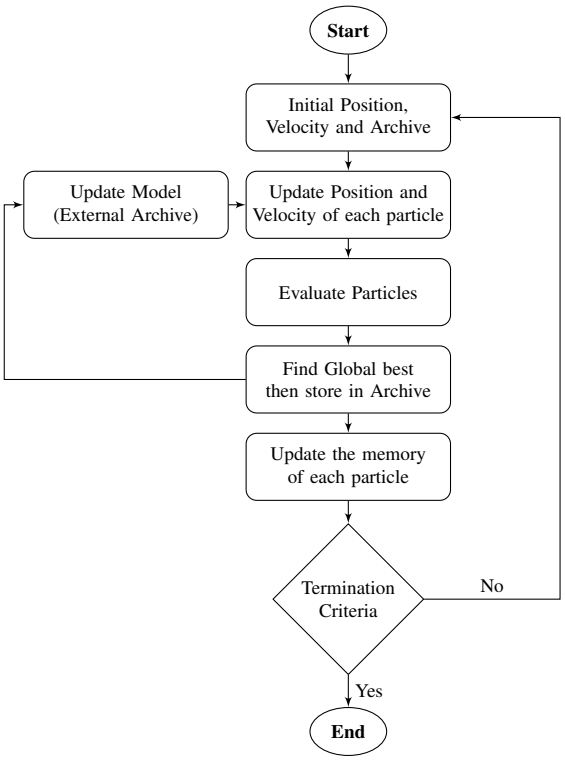


Figure 4. Flowchart of the MOPSO algorithm.

The MOPSO adjustment process involved careful tuning of key parameters such as inertia weight, cognitive coefficients, and social coefficients. The inertia weight used a decreasing strategy to facilitate a smooth transition from the exploration to the exploitation phases. The cognitive and social coefficients were calibrated to balance the individual and collective intelligence of the swarm, impacting both the speed of convergence and the quality of the solution set.

2.1.3. Non-Dominant Sorting Genetic Algorithm II

The Non-Dominant Sorting Genetic Algorithm II technique was developed by Deb et al. [26], [27] as an approach that incorporates both elite and diversity-preserving mechanisms. NSGA-II starts by generating a parent population P_0 through a random process. These solutions are then sorted into various non-domination fronts, where each solution is assigned a fitness level corresponding to its non-domination front. The first front represents the best solutions, while subsequent fronts represent progressively worse solutions. Based on the crowded comparison criterion and crossover and mutation operators, binary tournament selection is used to create the initial child population Q_0 of size N.

For $t > 1$, the NSGA-II method works as follows:

- The populations of parent P_t and child Q_t are combined.
- The non-domination approach is used to sort the population $R_t = P_t \cup Q_t$, which has a size of $2N$.
- Selected solutions from the first front are used to build a new parent population P_{t+1} until the population size exceeds N.
- For each non-dominated front of R_t, the crowding distance F_i is computed.
- Selection, crossover, and mutation operators are used to create the new population Q_{t+1}.

The NSGA-II pseudocode and flowchart are shown in Figures 5 and 6, respectively.

Begin
Initialize Population.
Generate Population randomly with size N.
Evaluate Objectives Values.
Assign Rank (front) based on Pareto - sort.
Generate children population.
Binary tournament selection.
Crossover and mutation.
for $i = 1$ *to* g **do**
  for *each Parent and child in population* **do**
    Assign Rank (front) based on Pareto - sort.
    Generate sets of non-dominated solutions.
  end
  Determine Crowding distance.
  Adding solutions to the next generation starting from the 1st front until individuals.
end
Select points on the lower front with high crowding distance.
Create the next generation.
Binary tournament selection.
Crossover and mutation.
End

Figure 5. Pseudocode of NSGA-II algorithm.

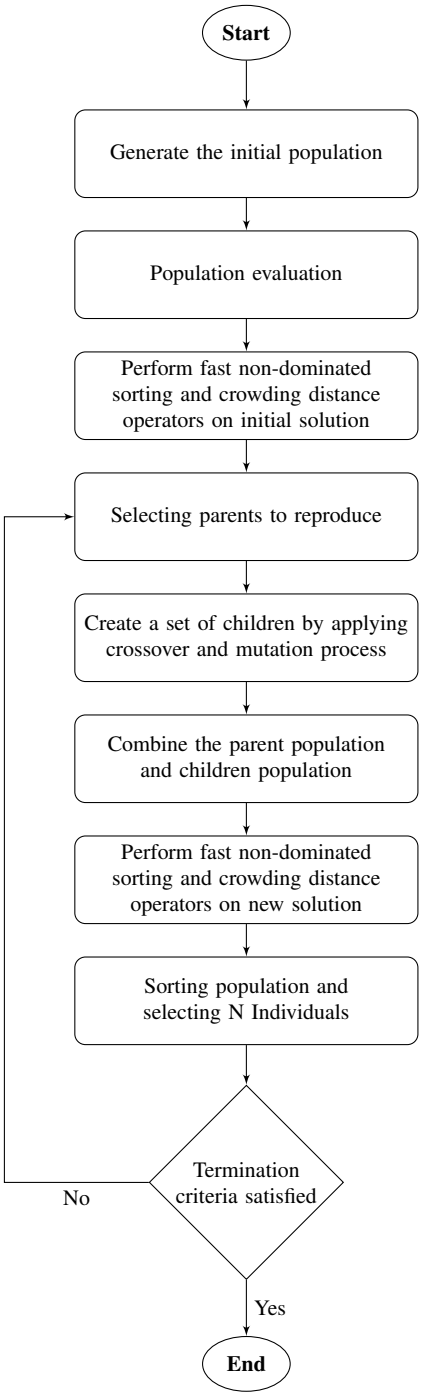


Figure 6. Flowchart of the NSGA-II algorithm.

The NSGA-II adjustment process involved carefully fine-tuning the probabilities of crossover and mutation to optimize search robustness and maintain genetic diversity within the population. We set the crossover probability to promote an effective combination of genetic attributes, ensuring a balanced exchange. Simultaneously, the mutation probability was kept sufficiently low to prevent significant changes while being high enough to encourage the exploration of new solution spaces.

2.2. Performance metrics

The performance of several Multi-Objective Evolutionary Algorithms (MOEAs) for a specific problem was evaluated by running each algorithm for the same number of iterations. The obtained solutions, also known as Pareto front approximations, were assessed using two quality metrics: solution diversity, which analyzes the distribution of points within the solution set, and solution accuracy, which measures how close the solutions are to the true Pareto front. Three metrics frequently employed in recent literature were used [28]. These metrics, listed in order of popularity, are Generational Distance (GD), Spacing (SP), and Hypervolume (HV). The definitions of these metrics are as follows:

2.2.1. **Generational distance**

The Generational Distance (GD) is widely recognized as a key indicator of convergence quality in multi-objective optimization [29], [30]. It measures the average distance from the Pareto front produced by an optimization algorithm to the true Pareto front, as depicted in Figure 7. The GD is calculated using Equation 6, where p represents the number of non-dominated solutions on the true Pareto front and a lower GD value indicates better algorithm performance. Here, n represents the number of non-dominated solutions the algorithm generates for a problem with M objective functions. The term d_i refers to the shortest distance from each non-dominated solution to the true Pareto front, illustrated in Figure 7, and is computed according to Equation 7.

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

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

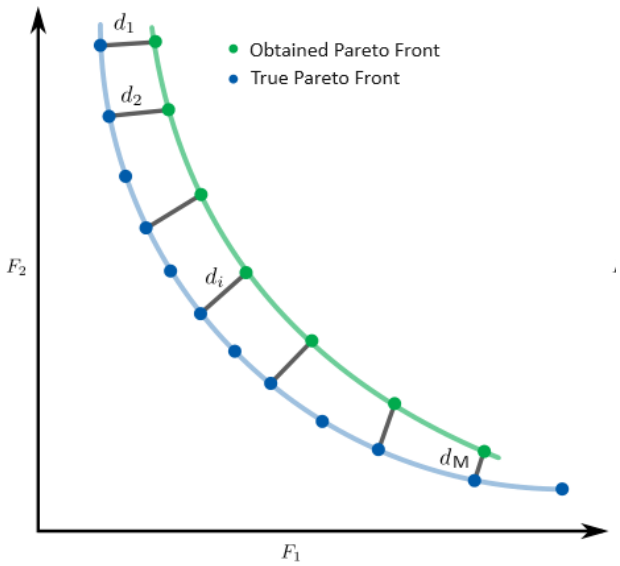


Figure 7. Schematic representation of GD metric for MOPs.

2.2.2. **Spacing**

The Spacing (SP) quantifies the uniformity of distribution among non-dominated solutions on a Pareto front. It is defined mathematically as:

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

Where:

- d_i represents the shortest distance (Euclidean distance) between any given point s_i from the set Q and the nearest point on the Pareto front approximation generated by the same algorithm.
- $\bar{d}$ is the mean of these distances d_i, calculated as $\bar{d} = \frac{1}{|Q|}\sum_{i=1}^{|Q|} d_i$.

A lower value of S_p indicates a more uniform distribution of solutions, meaning each point is approximately equidistant from its neighbors. This metric is crucial for evaluating the distribution and uniformity of the solutions, which is vital for detecting clustering or irregular spacing along the curve. Figure 8 provides a schematic representation of the S_p metric.

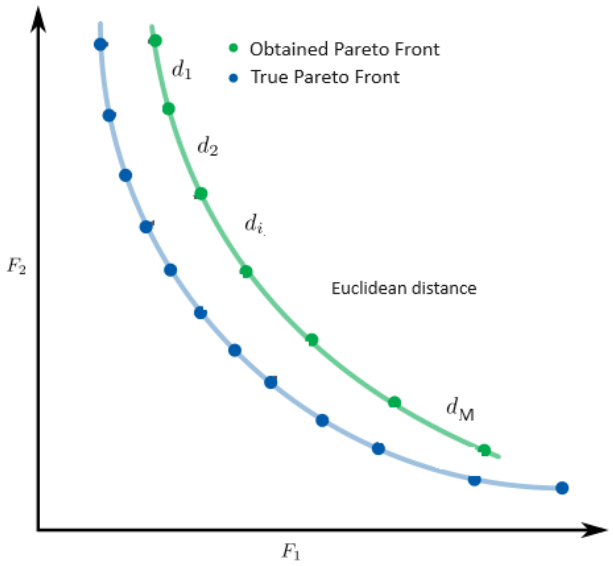


Figure 8. Schematic representation of Sp metric for MOPs.

2.2.3. **Hypervolume**

The hypervolume indicator (HV) was first introduced by Zitzler and Thiele [31]. It is used to assess either the area (in multi-objective problems) or the volume between a chosen reference point (r^*), ($r^* = r_1^*, \ldots, r_m^*$) in Ω, and the Pareto front achieved by the algorithm's non-dominated solutions. This concept is illustrated in Figure 9, which depicts the total area of the HV. Effective algorithms should ensure that non-dominated solutions not only extend away from the reference point but are also evenly spread between the boundary solutions and the actual Pareto front. Ideally, a large HV value indicates a favorable Pareto front generated by the algorithm. The HV is mathematically defined as follows:

$$HV = \sum_{i=1}^{|P|} v_i \tag{9}$$

Where (r^*) is a vector defined by the worst objective values of the actual Pareto front, and v_i represents the hypervolume of the $i-th$ hypercube defined by (r^*) and the diagonal corner of each non-dominated solution, assuming there are n such solutions.

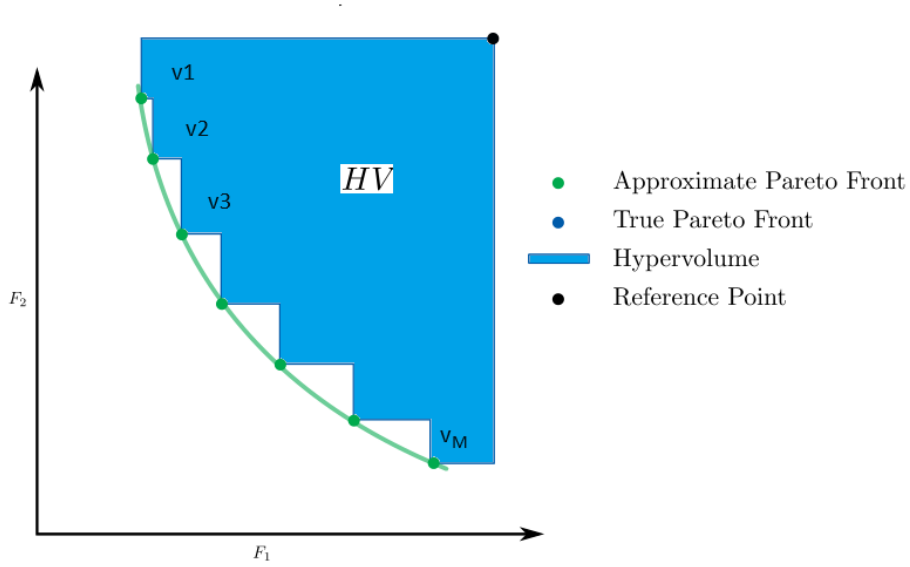


Figure 9. Schematic representation of HV metric for MOPs.

3. Tests and validation

In this paper, the performance of several algorithms is evaluated and validated by comparing them using a set of benchmark functions specifically created for multi-objective optimization problems. The focus is on five well-known ZDT benchmark functions [32], recognized for their unconstrained environments and distinct Pareto optimal frontier shapes. These functions are mathematically modeled and detailed in Table 1. The evaluation uses performance metrics obtained through MATLAB on a system with an Intel Core i5 CPU @ 1.6 GHz and 4 GB of DDR3 RAM. This setup examines how effectively multi-objective optimization techniques handle convex and non-linear issues.

Comparative analyses were conducted among the Multi-Objective Artificial Bee Colony (MOABC), Non-dominated Sorting Genetic Algorithm II (NSGA-II), and Multi-Objective Particle Swarm Optimization (MOPSO) algorithms. Each algorithm was tested across five case studies, with 100 iterations and populations, as depicted in Table 2. Figure 10 displays the statistical results of several algorithms applied to multi-objective benchmark functions. Except for MOPSO and NSGA-II on certain ZDT problems, it was found that all three methods effectively converged towards the Pareto front in most cases.

According to Table 3, the MOABC algorithm showed superior convergence accuracy in all ZDT benchmark tests compared to the others. For the spread (SP) metrics, all algorithms consistently achieved values below 0.1 across all ZDT challenges. Particularly noteworthy is that the MOABC algorithm outperformed others in all ZDT benchmarks, consistently demonstrating robust performance across all measured metrics.

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

Function Name	Mathematical Model Formula	Search Domain	Function Characteristic
ZDT1	$f_1(x) = x_1$ $f_2(x) = g(x) \cdot h(f_1(x), g(x))$ $g(x) = 1 + \frac{9}{(n-1)} \sum_{i=2}^{n} x_i$ $h(f_1(x), g(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}}$	$0 \leq x_i^* \leq 1$ for $1 \leq i \leq 30$	Convex Pareto optimal front
ZDT2	$f_1(x) = x_1$ $f_2(x) = g(x) \cdot h(f_1(x), g(x))$ $g(x) = 1 + \frac{9}{(n-1)} \sum_{i=2}^{n} x_i$ $h(f_1(x), g(x)) = 1 - \left(\frac{f_1(x)}{g(x)}\right)^2$	$0 \leq x_i^* \leq 1$ for $1 \leq i \leq 30$	Concave Pareto optimal front
ZDT3	$f_1(x) = x_1$ $f_2(x) = g(x) \cdot h(f_1(x), g(x))$ $g(x) = 1 + \frac{9}{(n-1)} \sum_{i=2}^{n} x_i$ $h(f_1(x), g(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}} - \frac{f_1(x)}{g(x)} \sin(10\pi f_1(x))$	$0 \leq x_i^* \leq 1$ for $1 \leq i \leq 30$	Disconnected Pareto optimal fronts
ZDT4	$f_1(x) = x_1$ $f_2(x) = g(x) \cdot h(f_1(x), g(x))$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n} \left(x_i^2 - 10\cos(4\pi x_i)\right)$ $h(f_1(x), g(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}}$	$0 \leq x_i^* \leq 1$ for $1 \leq i \leq 30$	Many local fronts, single global convex front
ZDT6	$f_1(x) = 1 - e^{-4x_1} \sin^6(6\pi x_1)$ $f_2(x) = g(x) \cdot h(f_1(x), g(x))$ $g(x) = 1 + \left(\frac{9}{(n-1)} \sum_{i=2}^{n} x_i\right)^{\frac{1}{4}}$ $h(f_1(x), g(x)) = 1 - \left(\frac{f_1(x)}{g(x)}\right)^2$	$0 \leq x_i^* \leq 1$ for $1 \leq i \leq 30$	Non-uniform distribution, non-convex front

.

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% of NCS)	50	-	-
Number of onlooker bees (50% of NCS)	50	-	-
Number of scouts (1)	1	-	-
Limit (=Number of onlooker bees*D)	100	-	-
Size of the external archive (sizeAR)	25	25	-
Weight Damping Rate (Wdamp)	-	0.99	-
Weight factor (w)	-	0.5	-
Acceleration coefficient ($0.5 \leq$ C1 ≤ 2.5)	-	1	-
Acceleration coefficient ($0.5 \leq$ C2 ≤ 2.5)	-	2	-
Number of Grids per Dimension (NGrid)	-	7	-
Inflation Rate (α)	-	0.1	-
Leader Selection Pressure (β)	-	2	-
Deletion Selection Pressure (γ)	-	2	-
Mutation Rate (μ)	-	0.1	-
Crossover ratio (pc)	-	-	0.8
Mutation ratio (pm)	-	-	0.3

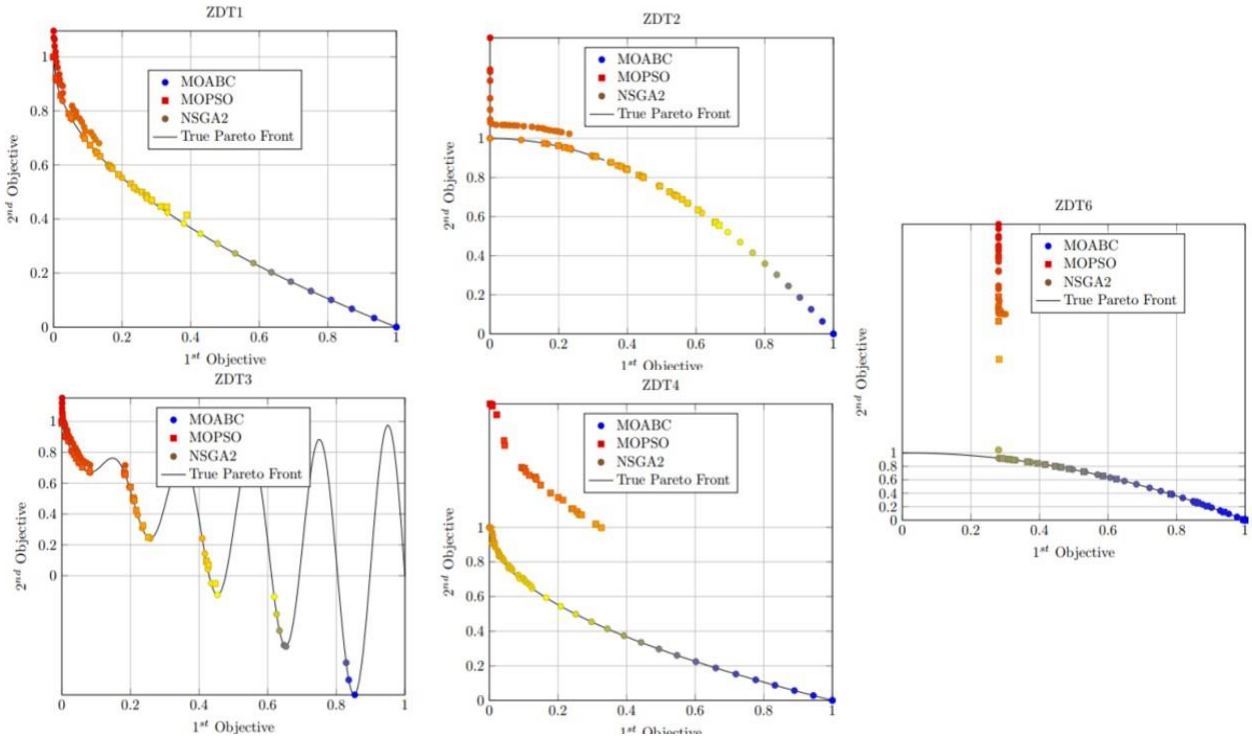


Figure 10. True and obtained Pareto fronts by MOABC, MOPSO, and NSGA-II algorithms on 5 ZDT test problems.

Table 3. Statistical results of GD, SP, and HV metrics on ZDT benchmark functions

Test Function		Algorithm/GD metric			Algorithm/SP metric			Algorithm/HV metric		
		MOABC	MOPSO	NSGA-II	MOABC	MOPSO	NSGA-II	MOABC	MOPSO	NSGA-II
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	7.16E-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-01	7.09E-01
	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	7.02E-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	4.42E-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	4.24E-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	4.00E-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	8.25E-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	8.19E-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	8.08E-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	7.19E-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	7.06E-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	6.70E-01
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	0.00E+00
	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	0.00E+00
	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	0.00E+00

4. Application example: planar spiral inductor

This section evaluates the performance of the planar spiral inductor using the MOABC, MOPSO, and NSGA-II algorithms. The square shape of the planar spiral inductor, as depicted in Figure 11, was employed in this study. The geometry of this inductor is determined by the following parameters: the outer diameter (d_{out}), the inner diameter (d_{in}), the turn width (w), the spacing between turns (s), and the number of turns (n). Figure 12 depicts the π-model of the spiral inductor, which comprises the following components: series inductance (L_s), series resistance (R_s), capacitance of the parallel plates between the spiral and the center-tap underpass (C_s), capacitance between the spiral and the oxide capacitance (C_{ox}), substrate capacitance (C_{si}), and substrate resistance (R_{si}).

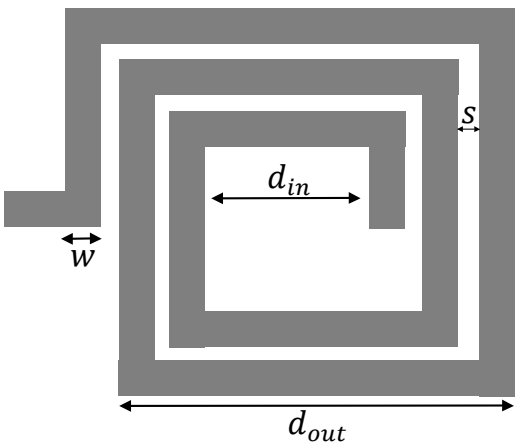


Figure 11. A square shape of spiral inductor.

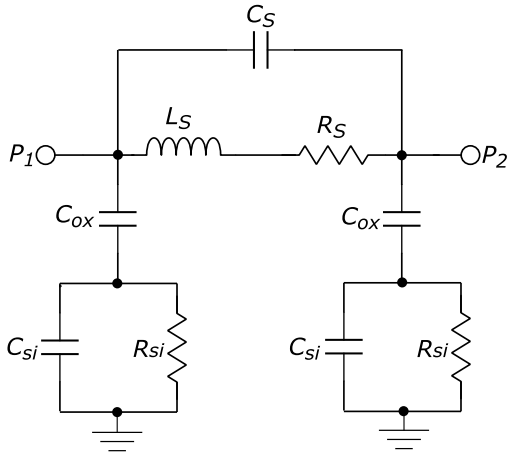


Figure 12. π-model of a spiral inductor.

The expressions for the parasitic elements R_s, C_s, C_{ox}, C_{si}, and R_{si} are presented below [33]:

$$R_s = \frac{l}{\sigma w \delta \left(1 - e^{\left(\frac{-t}{\delta}\right)}\right)} \tag{10}$$

$$C_s = \frac{\epsilon_{ox}}{t_{ox,M_1-M_2}} nw^2 \tag{11}$$

$$C_{ox} = \frac{1}{2}\frac{\epsilon_{ox}}{t_{ox}} lw \tag{12}$$

$$C_{si} = \frac{1}{2} C_{sub} lw \tag{13}$$

$$R_{si} = \frac{2}{G_{sub} lw} \tag{14}$$

where σ represents the conductivity of the metal, t the thickness of the metal trace, δ the skin depth, and ϵ_{ox} the permittivity of the oxide. t_{ox,M_1-M_2} represents the oxide thickness located between the spiral and the center-tap underpass, t_{ox} represents the oxide thickness between the spiral and the substrate, l is the total length of the spiral, C_{sub} is the substrate capacitance per unit area, and G_{sub} is the substrate conductance per unit area.

This study focuses on two objectives: $f_1(x) = -Q(x)$ and $f_2(x) = A(x)$. Here, Q denotes the quality factor at the operating frequency, and A represents the inductor's layout area, equal to d_{out}^2.

The expression for the quality factor (Q) is as follows [33], [34]:

$$Q = \frac{\omega L_s}{R_s} \cdot \frac{R_p}{R_p + \left(\left(\frac{\omega L_s}{R_s}\right)^2 + 1\right) R_s} \cdot \left(1 - \frac{R_s^2(C_s + C_p)}{L_s} - \omega^2 L_s (C_s + C_p)\right) \tag{15}$$

Where,

$$R_p = \frac{1}{\omega^2 C_{ox}^2 R_{si}} + \frac{R_{si}(C_{ox} + C_{si})^2}{C_{ox}^2} \tag{16}$$

$$C_p = C_{ox} \cdot \frac{1 + \omega^2 (C_{ox} + C_{si}) C_{si} R_{si}^2}{1 + \omega^2 (C_{ox} + C_{si})^2 R_{si}^2} \tag{17}$$

In addition, the inductance (L_s) is given by the following expression [35]:

$$L_s = \beta d_{out}^{\alpha_1} w^{\alpha_2} d_{avg}^{\alpha_3} n^{\alpha_4} s^{\alpha_5} \tag{18}$$

Where,

$$d_{avg} = \frac{1}{2}\left(d_{out} + d_{in}\right) \tag{19}$$

$$d_{out} = d_{in} + 2nw + 2(n-1)s \tag{20}$$

The coefficients β and $\alpha_i (i = 1, \cdots, 5)$ depend on the inductor's shape, and their values can be found in [35]. The optimization problem is formulated as follows:

$$\begin{array}{ll} \text{minimize} & (f_1(x), f_2(x)) \\ \text{subject to} & L_s = L_{spec}, \\ & g_1(x) = 2n(w+s) - 2s - d_{out} \leq 0, \\ & g_2(x) = 0.2 - \frac{d_{in}}{d_{out}} \leq 0, \\ & g_3(x) = \frac{d_{in}}{d_{out}} - 0.8 \leq 0, \\ & g_4(x) = 5w - d_{in} \leq 0, \end{array} \tag{21}$$

4.1. Results and discussion

This study aims to design a $L_{spec} = 4\,nH$ inductor operating at $2.4\,GHz$ using UMC 130 nm CMOS technology. In this section, the primary objective is to find the optimal values of the variables d_{out}, w, s, and n to optimize two conflicting performances: maximizing the quality factor Q while minimizing the inductor area A, achieved by generating a Pareto-optimal front. Tables 4 and 5 present the technological parameters and constraints of the design variables, respectively.

Table 4. Technology parameters

Parameters	value
Metal thickness (μm)	2.8
Metal conductivity (S/m)	$3.7736 \cdot 10^7$
Oxide thickness (μm)	5.42
Oxide thickness between spiral and underpass (μm)	0.4
Permeability of vacuum $\mu_0(H/m)$	$4\pi \cdot 10^-7$
Vacuum permittivity $\epsilon_0(F/m)$	$8.85 \cdot 10^-12$
Permittivity relative of oxide ϵ_r	4
Permittivity relative of substrate ϵ_r	11.9
Substrate thickness (μm)	100
Substrate resistivity $(\Omega.m)$	28

Table 5. Design variables bounds

Variable	upper bound	lower bound
d_{out} (μm)	140	350
w (μm)	1	12
s (μm)	2.5	7.5
n	1.5	7.5

4.1.1. Optimization results

The results of the inductor optimization using MOABC, MOPSO, and NSGA-II are listed in Table 6, which presents three solutions for each algorithm. Additionally, the Pareto front generated by MOABC, MOPSO, and NSGA-II is shown in Figure 13. The results in Table 6 reveal that MOABC yields the highest quality factor value, while NSGA-II achieves the lowest area value. The solutions illustrated in Figure 13 demonstrate that MOABC provides superior results compared to MOPSO and NSGA-II, showing a better distribution of solutions on the Pareto front. Consequently, MOABC offers a more suitable trade-off between the two conflicting objectives: the quality factor and the area of the inductor.

Table 6. Optimization results

MOABC			
Solutions	**Solution 1**	**Solution 2**	**Solution 3**
d_{out} (μm)	206.27	192.97	186.05
w (μm)	11.850	10.812	9.9643
s (μm)	2.5224	2.5013	2.5052
n	4.8191	5.1579	5.2496
A (μm^2)	4.2545E+4	3.7239E+4	3.4613E+4
Q	12.0747	11.3786	10.9212
L_s (nH)	3.9995	3.9435	4.0079
MOPSO			
Solutions	**Solution 1**	**Solution 2**	**Solution 3**
d_{out} (μm)	208.38	186.78	173.51
w (μm)	11.422	9.8928	9.2295
s (μm)	2.5	2.5	2.5
n	4.3387	5.2384	5.2384
A (μm^2)	4.3424E+4	3.4888E+4	3.0106E+4
Q	12.0297	10.9313	10.2070
L_s (nH)	3.9510	4.0775	3.8028
NSGA-II			
Solutions	**Solution 1**	**Solution 2**	**Solution 3**
d_{out} (μm)	195.85	182.98	165.66
w (μm)	10.853	9.8414	7.8502
s (μm)	2.6856	2.7353	2.7217
n	5.0576	5.4507	5.3848
A (μm^2)	3.8357E+4	3.3482E+4	2.7444E+4
Q	11.4102	10.6171	9.3999
L_s (nH)	3.9645	3.8807	3.9283

4.1.2. Simulation results

The optimization results obtained using MOABC, MOPSO, and NSGA-II algorithms are compared against electromagnetic (EM) simulations. Table 7 shows relative errors in quality factor Q and inductance L_s. Additionally, Figure 14 displays the simulation results of the quality factor and inductance as functions of frequency for each algorithm. As shown in Table 7, MOABC provides the lowest relative error for the quality factor compared to the other algorithms. Figure 14 reveals that for all three algorithms, the inductance L_s remains constant up to around 5 GHz, while the quality factor reaches its peak value near the operating frequency. Therefore, the optimization results are consistent with the simulation results.

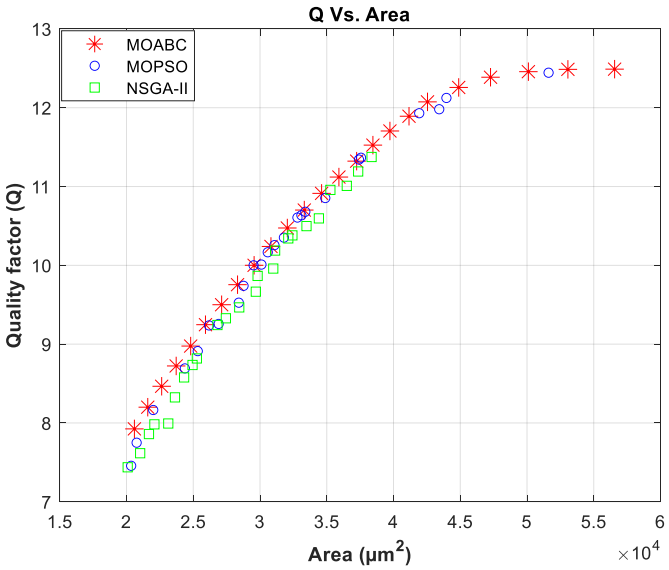
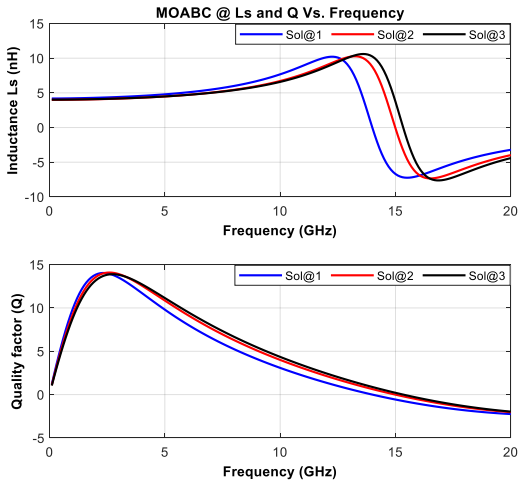


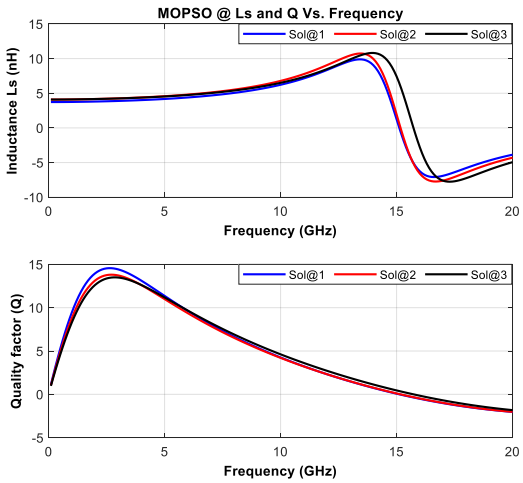
Figure 13. Inductor Pareto front with MOABC, MOPSO, and NSGA-II.

Table 7. Optimization and simulation results

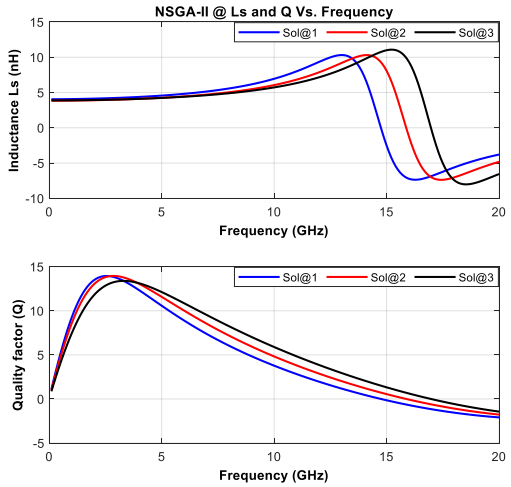
MOABC			
Solutions	**Solution 1**	**Solution 2**	**Solution 3**
$Q_{\_opt}$	12.0747	11.3786	10.9212
$Q_{\_sim}$	14.029	14.035	13.758
$Error(\%)$	13.9304	18.9270	20.6193
$L_{s\_opt}(nH)$	3.9995	3.9435	4.0079
$L_{s\_sim}(nH)$	4.332	4.109	4.129
$Error(\%)$	7.6754	4.0277	2.9329
MOPSO			
Solutions	**Solution 1**	**Solution 2**	**Solution 3**
$Q_{\_opt}$	12.0297	10.9313	10.2070
$Q_{\_sim}$	14.492	13.698	13.282
$Error(\%)$	16.9908	20.1978	23.1516
$L_{s\_opt}(nH)$	3.9510	4.0775	3.8028
$L_{s\_sim}(nH)$	3.833	4.202	4.195
$Error(\%)$	3.0785	2.9629	9.3492
NSGA-II			
Solutions	**Solution 1**	**Solution 2**	**Solution 3**
$Q_{\_opt}$	11.4102	10.6171	9.3999
$Q_{\_sim}$	13.919	13.719	12.691
$Error(\%)$	18.0243	22.6102	25.9326
$L_{s\_opt}(nH)$	3.9645	3.8807	3.9283
$L_{s\_sim}(nH)$	4.174	3.938	3.962
$Error(\%)$	5.0192	1.4551	0.8506



(a)



(b)



(c)

Figure 14. Inductance and quality factor vs. frequency for (a) MOABC, (b) MOPSO, and (c) NSGA-II.

5. Conclusion

The paper presents a comparative analysis of three well-known multi-objective algorithms applied to the ZDT benchmark functions. Three Pareto front metrics were used to evaluate their performance. Among these algorithms, the MOABC algorithm consistently demonstrates superior convergence accuracy and HV values, making it the top performer. The MOPSO algorithm struggles to find high-quality solutions, while the NSGA-II algorithm shows mixed performance. Additionally, efforts focus on applying these three algorithms to achieve optimal sizing of the inductor, aiming to maximize the quality factor (Q) while minimizing its area simultaneously. In summary, the results demonstrate the efficacy of the MOABC algorithm and provide significant guidance for algorithm selection in multi-objective optimization situations.

Limitations and Future Work:

This study identifies key findings but acknowledges several limitations. ZDT benchmark functions may not fully capture the complexity of real-world problems, and the focus on analog circuit problems with specific parameters may limit the generalizability of the results. The computational cost can also be significant, limiting practicality for real-time applications or large-scale problems.

Future research should include more complex benchmark functions, apply the MOABC algorithm to other engineering areas, and develop techniques for hybrid optimization systems that combine the strengths of MOABC with different algorithms.

REFERENCES

1. J. Wu, and S. Azarm, *Metrics for Quality Assessment of a Multiobjective Design Optimization Solution Set*, Journal of Mechanical Design, vol. 123, no. 1, pp. 18–25, 2000.
2. S. Sharma, and V. Kumar, *A Comprehensive Review on Multi-objective Optimization Techniques: Past, Present and Future*, Archives of Computational Methods in Engineering, vol. 29, pp. 5605–5633, 2022.
3. B. Benhala, P. Pereira, and A. Sallem, *Focus on swarm intelligence research and applications*, NOVA Science Publishers, 2017.
4. B. Benhala, A. Ahaitouf, A. Mechaqrane, and B. Benlahbib, *Multiobjective optimization of second generation current conveyors by the ACO technique*, International Conference on Multimedia Computing and Systems (ICMCS'12), Tangier, Morocco:IEEE Press, pp. 1147–1151, 2012.
5. I. Boussaïd, J. Lepagnot, and P. Siarry, *A survey on optimization metaheuristics*, Information Sciences, vol. 237, pp. 82–117, 2013.
6. B. Benhala, A. Ahaitouf, M. Kotti, M. Fakhfakh, B. Benlahbib, A. Mecheqrane, M. Loulou, F. Abdi, and E. Abarkane, *Application of the ACO Technique to the Optimization of Analog Circuit Performances*, Chapter 9, Book: Analog Circuits: Applications, Design and Performance, Ed., Dr. Tlelo-Cuautle, NOVA Science Publishers, 2011.
7. 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, vol. 12, no. 5, pp. 2824–2836, 2023.
8. R. Dekimpe, and D. Bol, *A Configurable ULP Instrumentation Amplifier With Pareto-Optimal Power-Noise Trade-Off Achieving 1.93 NEF in 65nm CMOS*, IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 68, no. 7, pp. 2272–2276, 2021.
9. E. Sağlican, and E. Afacan, *MOEA/D vs. NSGA-II: A Comprehensive Comparison for Multi/Many Objective Analog/RF Circuit Optimization through a Generic Benchmark*, ACM Transactions on Design Automation of Electronic Systems, vol. 29, pp. 1–23, 2023.
10. K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, *A Fast Elitist Nondominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II*, in Parallel Problem Solving from Nature PPSN VI, vol. 1917, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E.Lutton, J. J. Merelo, and H.-P. Schwefel, Eds. Berlin, Heidelberg:Springer Berlin Heidelberg, 2000.
11. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Trans. Evol.Comput, vol. 6, pp. 182–197, 2002.
12. Y. Cui, X. Meng, and J. Qiao, *A multi-objective particle swarm optimization algorithm based on two-archive mechanism*, Applied Soft Computing, vol. 119, pp. 108532, 2022.
13. A. Sallem, B. Benhala, M. Kotti, M. Fakhfakh, A. Ahaitouf, and M. Loulou, *Simulation-based multi-objective optimization of current conveyors: Performance evaluations*, 7th International Conference on Design & Technology of Integrated Systems in Nanoscale Era, Tunis, Tunisia, pp. 1–5, 2012.
14. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, *Deep Learning Approach for Intelligent Intrusion Detection System*, IEEE Access, vol. 7, pp. 41525–41550, 2019.
15. H. Bouali, B. Benhala, and M. Guerbaoui, *A comparative Analysis of Two Multiobjective Metaheuristic Methods using Performance Metrics*, 3rd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Mohammedia, Morocco, pp. 1–4, 2023.
16. K. H. Ang, and Y. Li, *An overview of benchmarking techniques for multi-objective evolutionary algorithms*, Soft Computing and Industry: Recent Applications, pp. 337–348, 2002.

17. S. Rostami, and F. Neri, *A fast hypervolume driven selection mechanism for many-objective optimisation problems*, Swarm and Evolutionary Computation, vol. 34, no. 12, pp. 50–67, 2017.
18. I. Giagkiozis, R.C. Purshouse, and P.J. Fleming, *An overview of population-based algorithms for multi-objective optimization*, International Journal of Systems Science, vol. 46, pp. 1572–1599, 2013.
19. D. Karaboga, *An Idea Based On Honey Bee Swarm for Numerical Optimization*, Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
20. H. Bouali, B. Benhala, and H. Bouyghf, *Performance study of Multi-Objective Artificial Bee Colony (MOABC) Algorithm by Numerical Problems Benchmark*, in 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), pp. 1–6, 2020.
21. R. Akbari, R. Hedayatzadeh, K. Ziarati, and B. Hassanizadeh, *A multi-objective artificial bee colony algorithm*, Swarm Evol. Comput., vol. 2, pp. 39–52, 2012.
22. C. A. C. 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), vol. 2, pp. 1051–1056, 2002.
23. C.A.C. Coello, G.T. Pulido, and M.S. Lechuga, *Handling multiple objectives with particle swarm optimization*, IEEE Transactions on Evolutionary Computation, vol. 8, no. 3, pp. 256–279, 2004.
24. H. Yu, Y. Gao, and J. Wang, *A Multiobjective Particle Swarm Optimization Algorithm Based on Competition Mechanism and Gaussian Variation*, Complexity, pp. 1–23, 2020.
25. M. Z. bin Mohd Zain, J. Kanesan, J. H. Chuah, S. Dhanapal, and G. Kendall, *A multi-objective particle swarm optimization algorithm based on dynamic boundary search for constrained optimization*, Appl. Soft Comput, vol. 70, pp. 680–700, 2018.
26. K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, *A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II*, in Parallel Problem Solving from Nature PPSN VI, vol. 1917, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 849–858, 2000.
27. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Trans. Evol. Comput, vol. 6, pp. 182–197, 2002.
28. N. Riquelme, C. Von Lücken, and B. Baran, *Performance metrics in multi-objective optimization*, In Proceedings of the 2015 Latin American Computing Conference (CLEI), Arequipa, Peru, pp. 1–11, 2015.
29. K. Lwin, R. Qu, and G. Kendall, *A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization*, Appl. Soft Comput., vol. 24, pp. 757–772, 2014.
30. 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, pp. 1-13, 2019.
31. 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, 2014.
32. 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, Seattle, WD, USA, vol. 2, pp. 1141–11, 2006.
33. C. P. Yue, C. Ryu, J. Lau, T. H. Lee, and S. S. Wong, *A physical model for planar spiral inductors on silicon*, in International Electron Devices Meeting. Technical Digest, pp. 155–158, 1996.
34. C. P. Yue, and S. S. Wong, *On-chip spiral inductors with patterned ground shields for Si-based RF ICs*, IEEE Journal of Solid-State Circuits, vol. 33, no. 5, pp. 743–752, 1998.
35. S. S. Mohan, M. del Mar Hershenson, S. P. Boyd, and T. H. Lee, *Simple accurate expressions for planar spiral inductances*, IEEE Journal Solid-State Circuits, vol. 34, no. 10, pp. 1419–1424, 1999.