

An optimal design of current conveyors using a hybrid-based metaheuristic algorithm

Soufiane Abi, Bachir Benhala

Faculty of Sciences, Moulay Ismail University, Meknes, Morocco

Article Info

Article history:

Received Aug 26, 2021

Revised Jun 21, 2022

Accepted Jul 18, 2022

Keywords:

Ant colony optimization

Current conveyors

Differential evolution

Hybrid metaheuristic

Optimization

ABSTRACT

This paper focuses on the optimal sizing of a positive second-generation current conveyor (CCII+), employing a hybrid algorithm named DE-ACO, which is derived from the combination of differential evolution (DE) and ant colony optimization (ACO) algorithms. The basic idea of this hybridization is to apply the DE algorithm for the ACO algorithm's initialization stage. Benchmark test functions were used to evaluate the proposed algorithm's performance regarding the quality of the optimal solution, robustness, and computation time. Furthermore, the DE-ACO has been applied to optimize the CCII+ performances. SPICE simulation is utilized to validate the achieved results, and a comparison with the standard DE and ACO algorithms is reported. The results highlight that DE-ACO outperforms both ACO and DE.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Soufiane Abi

Faculty of Sciences, Moulay Ismail University

BP. 11201 Zitoune, Meknes, Morocco

Email: sou.abi@edu.umi.ac.ma

1. INTRODUCTION

Analog circuit optimal design techniques are increasingly gaining attention from analog designers because of the continuous need for a high-performance electronic circuit, with properties such as low power consumption and small area [1]. Analog circuit optimization problems involve various types of objective functions, variables, and constraints. Hence, these problems can be formulated as optimization problems, which optimization algorithms can fix. Metaheuristic methods have been suggested in the literature to tackle challenging problems in diverse domains [2], such as genetic algorithm (GA) [3], differential evolution (DE) [4], [5], particle swarm optimization (PSO) [6], ant colony optimization (ACO) [7]–[9], and artificial bee colony (ABC) [10]–[12]. Nevertheless, many researchers use hybrid metaheuristic methods to overcome many optimization issues.

The hybridization of metaheuristic algorithms is not simply an association of various algorithms, but their association is based on a specific framework [13]. In general, the performance of a unique algorithm is not efficient compared to a hybrid algorithm. Numerous studies have suggested hybridizing metaheuristic techniques belonging to the swarm intelligence domain. For instance, in [14], the authors proposed two-hybrid methods: a genetic algorithm with ant colony optimization (GA-ACO) and simulated annealing with ant colony optimization (SA-ACO) to optimize an operational amplifier and a complementary metal-oxide semiconductor (CMOS) second-generation current conveyor. In [15], a hybrid method based on whale optimization algorithm and particle swarm optimization was proposed to size a CMOS differential amplifier, two-stage operational amplifier, and radio frequency micro-electro-mechanical systems (RF MEMS) shunt switch. A hybrid algorithm between the whale optimization algorithm and modified grey wolf optimization (WOA-mGWO) was proposed in [16] and applied to optimize the performance of a two-stage operational

amplifier. In [17], the authors suggest hybridization of PSO and cuckoo search (CS) for sizing a low-voltage CMOS second-generation current conveyor.

The ACO algorithm has emerged as a valuable and efficient method for seeking optimal solutions to an optimization problem. ACO drawback requires a longer computation time than DE. Therefore, a novel hybrid algorithm is suggested by combining ACO with DE to enhance the execution speed of ACO. The hybrid algorithm developed in this work is named DE-ACO. In this hybridization, DE provides an enhanced initialization to ACO to limit the search space, focus on the exploitation process, and quickly converge to optimal solutions. First, DE-ACO's performances regarding solution quality, robustness, and computation time were evaluated using four benchmark test functions. Second, DE-ACO was applied to optimize the CCII+ performances.

The rest of this paper is organized as outlined below: ACO, DE, and DE-ACO are highlighted in section 2. Performance evaluation of DE-ACO using test functions is shown in section 3. In section 4, the CCII+ circuit is described. The obtained results are summarized in section 5. Finally, section 6 recaps the study.

2. METAHEURISTIC ALGORITHMS

2.1. Ant colony optimization algorithm

ACO is a population-based metaheuristic method illustrated by the real ants' foraging behavior. Dorigo *et al.* [18] proposed ACO in 1996, and it has been employed to deal with combinatorial optimization issues like the travelling salesman problem (TSP) [19], [20]. The transition probability of an ant k located at city i to an adjacent city j is given by (1):

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{j \in N_i^k} [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta} & \text{if } j \in N_i^k \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

where N_i^k is the set of neighbors of vertex i of the k^{th} ant. $\tau_{ij}(t)$ is pheromone trail quantity on edge (i, j) . α and β are two factors that illustrate the relative effects of pheromone trail and heuristic function. The expression of η_{ij} is given by the (2):

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (2)$$

where d_{ij} represent the distance from vertex i to j . Once all ant constructs their solutions, the pheromone value τ_{ij} is updated as shown in (3):

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (3)$$

where m is the number of ants, and ρ is the pheromone evaporation rate. $\Delta\tau_{ij}^k$ is the quantity of pheromone laid on edge (i, j) by ant k :

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{if ant } k \text{ uses edges}(i, j) \text{ in its tour} \\ 0 & \text{Otherwise} \end{cases} \quad (4)$$

here, L^k denoted the length of the tour constructed by ant k , and Q is a predefined constant. The ACO pseudocode is presented as shown in algorithm 1.

Algorithm 1. Pseudo code of ACO

```

Randomly initialize the values of pheromones
While stop criterion is not met do
  For each ant do
    Generate a candidate solution using the equation 1
    Assess the solution fitness
  End
Determine the best solutions

```

```

Perform the update of pheromone using equations 3 and 4
Save the optimal solutions
END

```

2.2. Differential evolution algorithm

DE is an evolutionary metaheuristic algorithm suggested by Storn and Price in 1997 [21]. The DE comprises the following primary operations: mutation, crossover, and selection. In the mutation, each individual in the population known as the target vector X_i is utilized to produce a mutant vector, V_i . The DE/rand/1 mutation strategy is used in this work as (5):

$$V_i = X_{r1} + F \cdot (X_{r2} - X_{r3}) \quad (5)$$

where $i = 1, 2, \dots, NP$; NP is the population size; $r1, r2, r3$ ($r1 \neq r2 \neq r3 \neq i$) are integers randomly selected in the interval $[1, NP]$. F is a scale factor chosen in $[0, 2]$. After the mutation operation finish, the DE utilizes mutation to produce a trial vector U_i by the (6):

$$U_{ij} = \begin{cases} V_{ij} & \text{if } (\text{rand}j \leq CR) \text{ or } (j = jrand) \\ X_{ij} & \text{otherwise} \end{cases} \quad (6)$$

here, $\text{rand}j \in [0, 1]$, $jrand$ is chosen randomly in $[1, D]$, and D is design variable numbers. CR is the crossover parameter $\in [0, 1]$. Trial vector U_i is compared to target vector X_i in the selection operation. For next-generation, the individual with the best fitness will be selected as (7). The pseudocode of the DE is as shown in algorithm 2.

$$X_i = \begin{cases} U_i & \text{if } f(U_i) \leq f(X_i) \\ X_i & \text{otherwise} \end{cases} \quad (7)$$

Algorithm 2. Pseudo code of DE

```

Randomly initialize the individuals of the population
Assess the individual fitness
While stop condition is not met do
  For each individual do
    Generate the vector Vi using the equation 5
    Create the vector Ui using the equation 6
    Evaluate the individual fitness
    Choose the best individuals using equation 7
  End
Save the optimal solutions
End

```

2.3. Proposed hybrid DE-ACO algorithm

The consumption of more computational time is the most significant shortcoming of ACO. Therefore, the DE-ACO has been designed using the advantage of DE to enhance ACO performances. In this hybridization strategy, DE is run for a predefined number of iterations. The best-achieved results by DE are employed to find a better launching point for the initialization of ACO. After DE completes the initialization phase, the search process is transferred to ACO to determine optimal solutions. DE-ACO structure is shown as:

Algorithm 3. Pseudo code of DE-ACO

```

Initialize the pheromone values randomly
Initialize the individuals of the population randomly
Assess the fitness of each individual
Calculate the distances values using solutions found by DE
While stop criterion is not met do
  For each ant do
    Generate a candidate solution using the equation 1
    Assess the solution fitness
  End
  Determine the best solutions
  Perform the update of pheromone using equations 3 and 4
  Save the optimal solutions
END

```

3. APPLICATION TO BENCHMARK TEST FUNCTIONS

In this section, four test functions [22], [23] are utilized to evaluate DE-ACO performances, and a comparison against ACO and DE is made. The functions description is provided in Table 1. Fmin is the optimum function value, and Range is the boundary of search space. The environment for the experimental simulation is an Intel 2.00 GHz processor with 4 GB of RAM and a Windows 10 operating system. Table 2 shows the parameter values of the algorithms. Dimension size is 2, population size is 50, and the number of iterations is 500.

Table 1. Test functions details

Name	Functions expression	Range	Fmin
Sphere	$F_1(x) = \sum_{i=1}^D x_i^2$	[-5.12, 5.12]	0
Alpine	$F_2(x) = \sum_{i=1}^D x_i \sin(x_i) + 0.1x_i $	[-10, 10]	0
Ackley	$F_3(x) = -20e^{\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^D x_i^2}\right)} - e^{\left(\frac{1}{D}\sum_{i=1}^D \cos(2\pi x_i)\right)} + 20 + e$	[-5, 5]	0
Griewank	$F_4(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-5, 5]	0

Table 2. Parameter settings

Parameter description	DE	ACO
Scale factor (F)	0.5	----
Crossover constant (CR)	0.9	----
Pheromone Factor (α)	----	1
Heuristics Factor (β)	----	1
Evaporation Rate (ρ)	----	0.1
Quantity of Deposit Pheromone (Q)	----	0.2

The four test functions are repeatedly evaluated using the DE-ACO, ACO, and DE algorithms to assess the proposed approach's performance. The algorithms are developed in MATLAB and run 30 times independently. The mean (mean) and standard deviation (SD) of the obtained results are reported and listed in Table 3.

Table 3. Statistical results of the used algorithms

		DE-ACO	ACO	DE
F1	Mean	0	0	3.65E-136
	SD	0	0	1.98E-135
F2	Mean	0	0	5.32E-59
	SD	0	0	2.58E-58
F3	Mean	8.88E-16	8.88E-16	8.88E-16
	SD	0	0	0
F4	Mean	0	0	0
	SD	0	0	0

From Table 3, DE-ACO and ACO perform better than DE. The mean CPU execution time of the used algorithms is investigated in Table 4. Figure 1 presents a boxplot of the best results from running these algorithms 30 times. As shown in Table 4, DE-ACO decreases the computational time of ACO. Figure 1 shows that the three algorithms are robust for the four functions. It can be deduced from the boxplots of the four test functions that the proposed DE-ACO generally converges toward the identical best solution.

Table 4. Average computing time (in seconds)

Function name	DE-ACO	ACO	DE
F1	4.03	4.84	0.20
F2	1.89	1.97	0.20
F3	4.00	4.23	0.25
F4	4.11	4.25	0.32

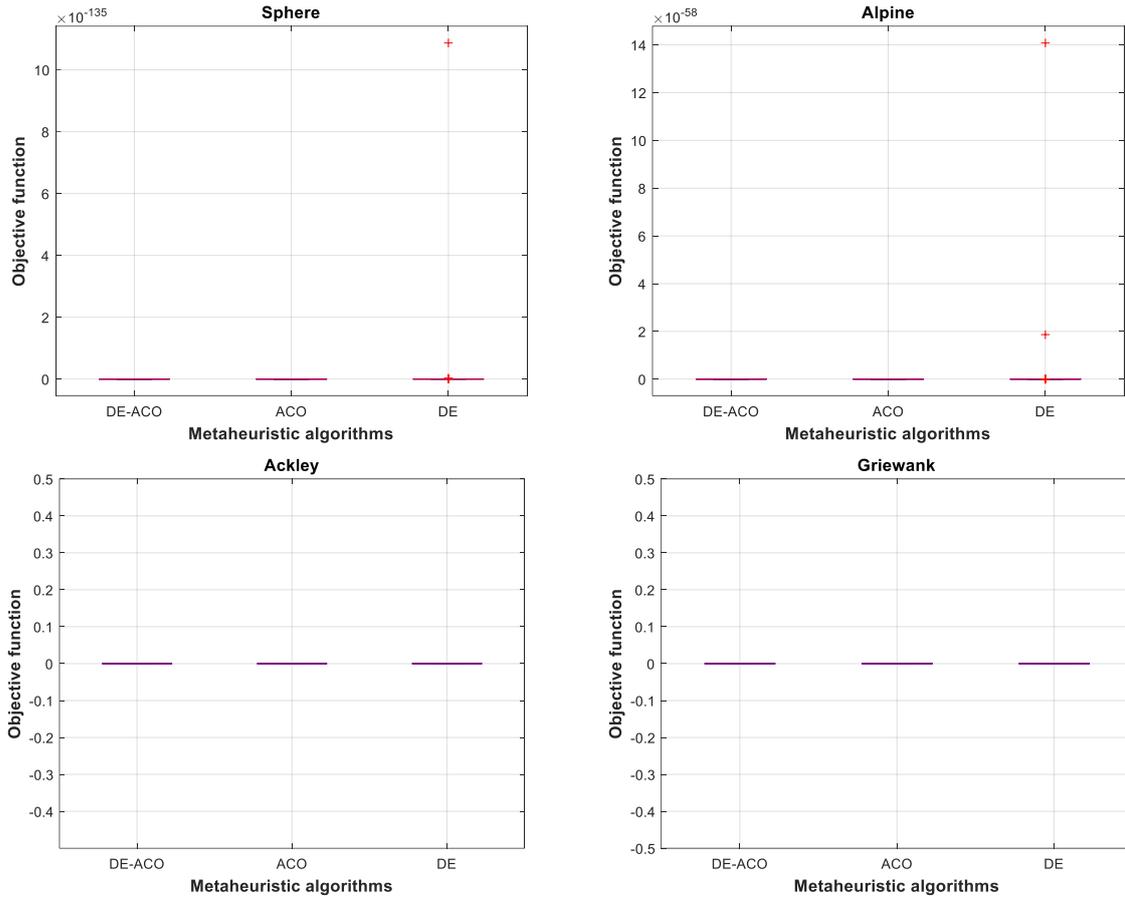


Figure 1. Boxplot representation for the test functions

4. CMOS CURRENT CONVEYORS

The current conveyors are analog current mode circuits widely applied in oscillators, amplifiers, and filters [24]. The current conveyor circuit used in this work is a positive second-generation current conveyor (CCII+), as depicted in Figure 2. It contains three active ports, X, Y, and Z. The trans linear loop made by transistors M1-M4 ensures the function of the current follower between ports X and Z. The current mirrors provided by transistors M5-M6 and M7-M8 allow the operating of the voltage follower between ports X and Y.

The optimization problem will be tackled through two distinct strategies. The first one will handle each objective function independently as a mono-objective optimization. The main objective is to find the optimal design variables that minimize the input X-port resistance (R_x) and maximize the high cut-off frequency (fci). The input X-port resistance (R_x) is given by (8) [25].

$$R_x \approx \frac{1}{g_{m2} + g_{m4}} \tag{8}$$

Here, gm represents transconductance for the MOS transistor. The expression of (fci) is not provided here because of its many terms. Furthermore, it can be derived from the expression of the current transfer function between X and Z ports offered by (9) [26].

$$\frac{I_Z}{I_X} = \frac{\left(1 + \frac{g_{mN}}{g_{0N}} + \frac{g_{mN}g_{mP}}{g_{0N}g_{0P}} + \frac{g_{mP}}{g_{0P}}\right)g_{0N}}{\left(\frac{C_{gsP}C_{gdP} + 2C_{gsP}C_{gsN}}{g_{0P}}\right)s^2 + \left(C_{gsN} + C_{gsP} + C_{gdP} + \frac{g_{mP}C_{gdP}}{g_{0P}} + \frac{g_{mP}C_{gsN}}{g_{0P}} + C_{gsN} + \frac{g_{mN}C_{gsP}}{g_{0P}}\right)s + \left(g_{mN} + \frac{g_{mN}g_{mP}}{g_{0P}}\right)} \tag{9}$$

Where g_0 represents the conductance of the MOS transistor, C_{gs} and C_{gd} are the parasitic grid to source capacitance, and the parasitic grid to drain the MOS transistor. s is Laplace coefficient.

According to the results below, the proposed approach highlights the best results for Rx and fci, and the simulation results match well with those of the optimization. In addition, the DE-ACO gives the lowest error than other algorithms. Due to the stochastic effect of the algorithms, these algorithms are executed 30 times. Figure 4 shows the boxplot representation of the best performances for Rx in Figure 4(a) and fci in Figure 4(b). Table 7 presents the average computing times of the methods used.

Table 5. Optimal results of CCII+ for Rx

	LN (μm)	LP (μm)	WN (μm)	WP (μm)	Rx (Ω)		Error (%)
					Optimization	Simulation	
DE-ACO	0.55	0.35	18.82	30	456	467	2.36
ACO	0.59	0.35	18.80	30	465	480	3.13
DE	0.59	0.35	18.88	30	464	479	3.13

Table 6. Optimal results of CCII+ for fci

	LN (μm)	LP (μm)	WN (μm)	WP (μm)	fci (Ghz)		Error (%)
					Optimization	Simulation	
DE-ACO	0.55	0.35	4.48	7.69	1.913	2.024	5.48
ACO	0.55	0.35	4.67	7.96	1.874	2.002	6.39
DE	0.55	0.35	4.86	8.34	1.838	1.980	7.17

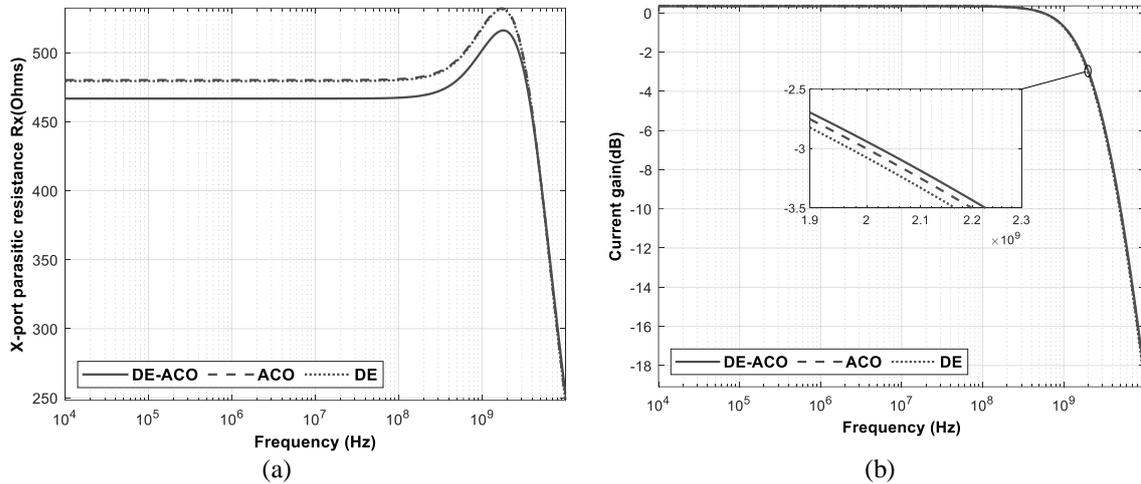


Figure 3. Simulation results for (a) Rx vs frequency and (b) current gain vs frequency

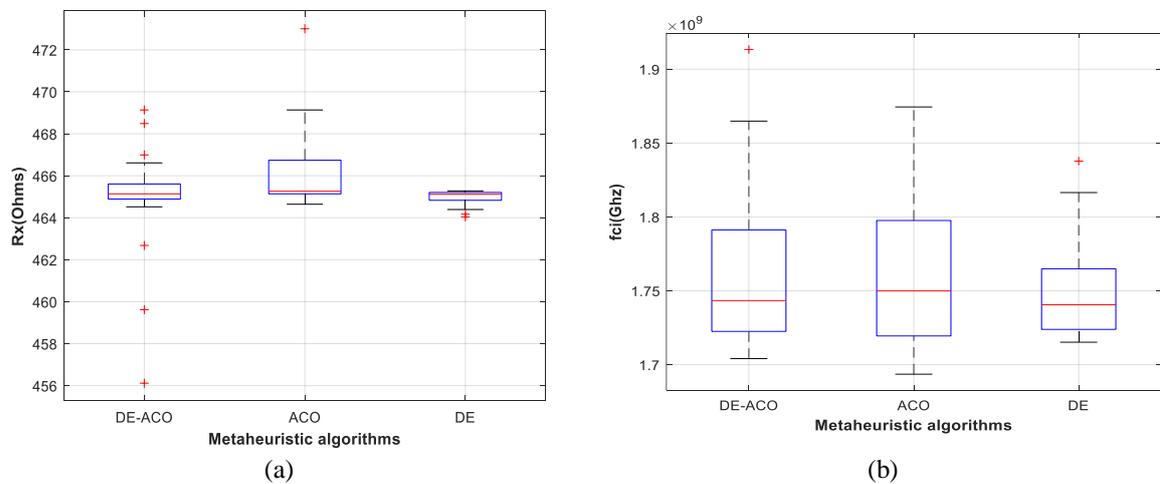


Figure 4. Boxplot representation for (a) Rx and (b) fci

As illustrated in Figure 4 and Table 7, DE-ACO enhances the robustness and reduces the computing times of the ACO algorithm. The second case, where the channel length of the N-MOS transistors is equal to that of the P-MOS transistors, is treated. The results of this case are provided in Tables 8 and 9.

The above results indicate that the hybrid algorithm provides the best performances of Rx and fci for CCII+ compared to ACO and DE algorithms. In addition, DE-ACO and ACO give a lower error for fci and Rx, respectively. Figure 5 presents the simulation results for Rx in Figure 5(a) and fci in Figure 5(b).

Table 7. Average computing time (in seconds)

	DE-ACO	ACO	DE
Rx	45.08	45.38	17.12
fci	46.06	46.61	17.19

Table 8. Optimal results of CCII+ for Rx

	LN=LP (μm)	WN (μm)	WP (μm)	Rx (Ω)		Error (%)
				Optimization	Simulation	
DE-ACO	0.35	11.36	29.93	519	461	12.58
ACO	0.35	11.29	29.64	521	464	12.28
DE	0.35	11.27	30.00	521	462	12.77

Table 9. Optimal results of CCII+ for fci

	LN=LP (μm)	WN (μm)	WP (μm)	fci (Ghz)		Error (%)
				Optimization	Simulation	
DE-ACO	0.35	4.82	5.37	2.254	2.522	10.63
ACO	0.35	4.86	5.41	2.245	2.523	11.02
DE	0.35	4.95	5.55	2.224	2.495	10.86

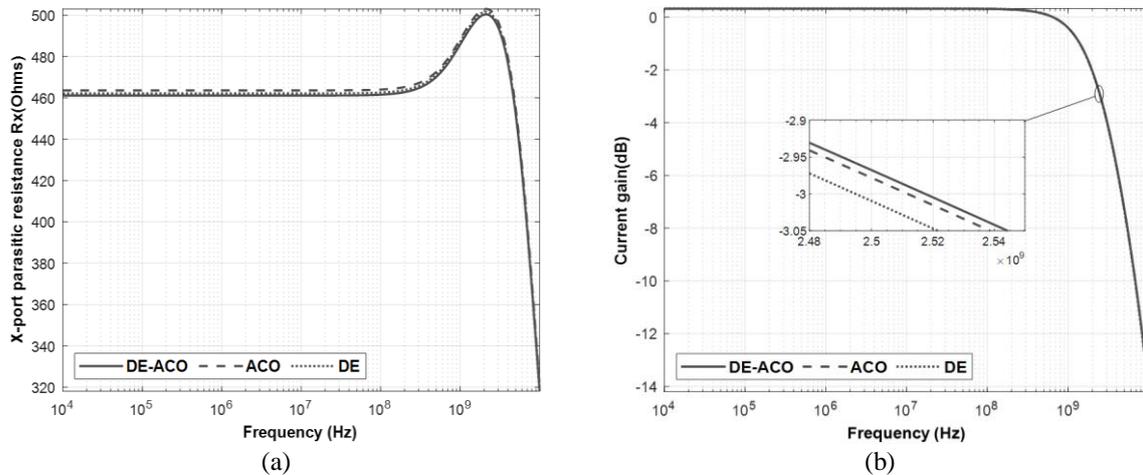


Figure 5. Simulation results for (a) Rx vs frequency and (b) current gain vs frequency

All algorithms are run 30 times independently. Figure 6 presents the boxplot representation for Rx in Figure 6(a) and fci in Figure 6(b). Table 10 highlights the average computing times of DE-ACO, ACO, and DE. The optimization results are in good agreement with the simulation results. DE-ACO and ACO are more robust than DE for Rx than DE, and the reverse for fci. In addition, DE-ACO minimizes the computing time of the ACO algorithm.

5.2. Bi-objective optimization results

This subsection discusses the second approach, where Rx and fci are optimized simultaneously. The channel length of N-MOS transistors is different from that of P-MOS transistors. Table 11 summarizes the optimization and simulation results. Table 11 shows that the DE-ACO achieved better results simultaneously for Rx and fci. Moreover, DE-ACO provides a lower Rx value than the ACO and DE algorithms. Figure 7 shows the SPICE simulation results for Rx in Figure 7(a) and fci in Figure 7(b). The used algorithms were run

30 times independently. Figure 8 presents the boxplot representation for Rx in Figure 8(a) and fci in Figure 8(b). Table 12 shows the average execution times of the three algorithms.

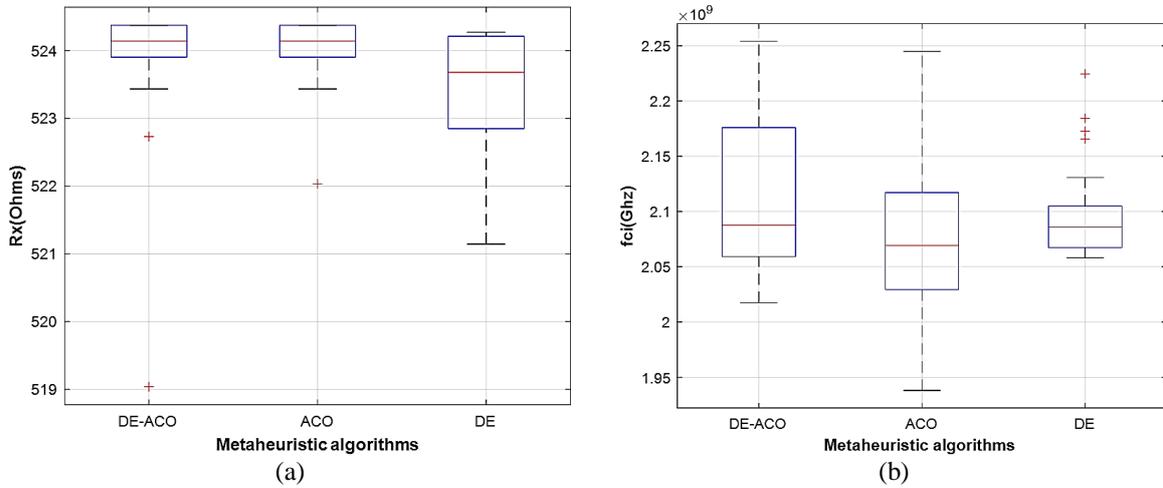


Figure 6. Boxplot representation for (a) Rx and (b) fci

Table 10. Average computing time (in seconds)

	DE-ACO	ACO	DE
Rx	40.06	41.75	16.78
fci	38.74	39.02	17.63

Table 11. Optimal results of CCII+ for Rx and fci

	LN	LP	WN	WP	Rx (Ω)		Error	fci (Ghz)		Error
	(μm)	(μm)	(μm)	(μm)	Optimization	Simulation	(%)	Optimization	Simulation	(%)
DE-ACO	0.58	0.35	9.60	14.88	647	743	12.92	1.298	1.554	16.47
ACO	0.59	0.35	9.56	14.97	652	747	12.72	1.337	1.537	13.01
DE	0.60	0.35	9.62	15.00	653	750	12.93	1.282	1.520	15.66

As seen in Figure 7, the simulation results match the optimization results. From Figure 8, DE-ACO is more robust than ACO for Rx and almost identical to ACO for fci. Furthermore, as shown in Table 12, the hybrid algorithm gives the lowest value in computing times than the ACO algorithm.

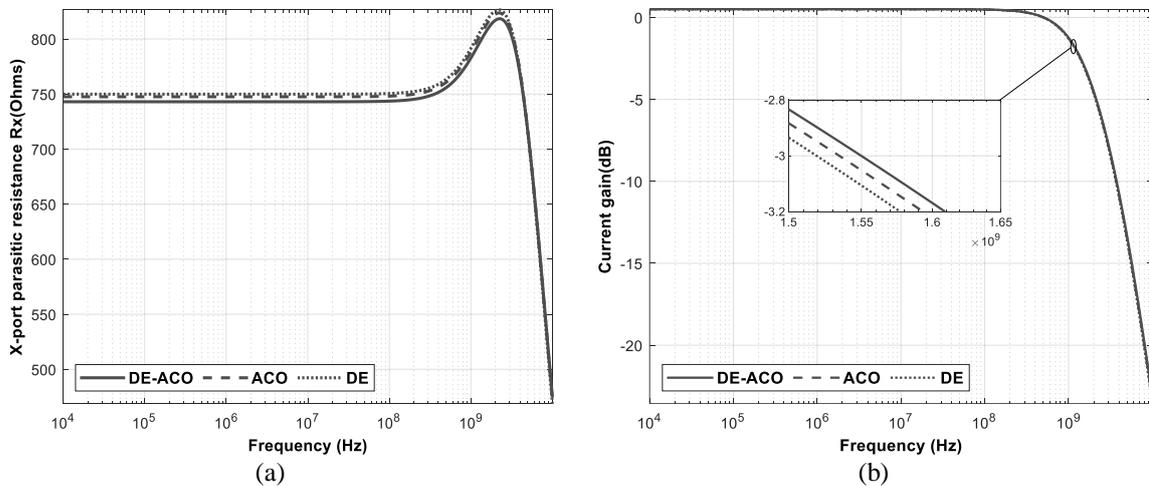


Figure 7. Simulation results for (a) Rx vs frequency and (b) current gain vs frequency

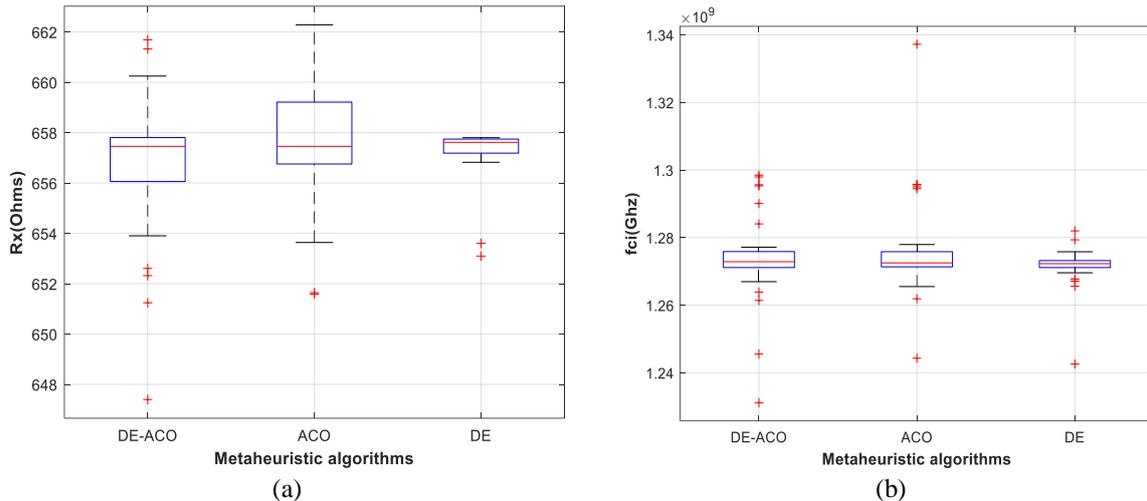


Figure 8. Boxplot representation for (a) Rx and (b) fci

Table 12. Average computing time (in seconds)

	DE-ACO	ACO	DE
f_{obj}	32.82	39.36	16.52

6. CONCLUSION

A hybrid algorithm referred to as DE-ACO was proposed in this work, which uses ACO and DE's advantages to obtain a robust algorithm with better quality solutions. The suggested hybridization uses the DE to provide a better initialization of ACO. Four test functions are used to evaluate the hybrid algorithm's performances regarding solutions quality, robustness, and running times. As an analog circuit optimization application, DE-ACO was employed to optimize the Rx and fci of CCII+. The results highlight that DE-ACO performs better for the test functions and the CCII+ circuit. We conclude that the DE-ACO technique is successfully applied for analog circuit optimization.

REFERENCES

- [1] M. Fakhfakh, *Analog/RF and mixed-signal circuit systematic design*, vol. 233. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
- [2] E.-G. Talbi and A. Nakib, *Bioinspired heuristics for optimization*. Springer; 1st ed. 2019 edition, 2019.
- [3] A. El Beqal, B. Benhala, and I. Zorkani, "A Genetic algorithm for the optimal design of a multistage amplifier," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, pp. 129–138, Feb. 2020, doi: 10.11591/ijece.v10i1.pp129-138.
- [4] S. Abi, B. Benhala, and H. Bouyghf, "A hybrid DE-ACO algorithm for the global optimization," in *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*, Dec. 2020, pp. 1–6, doi: 10.1109/ICECOCS50124.2020.9314533.
- [5] B. Durmuş, "Optimal components selection for active filter design with average differential evolution algorithm," *AEU-International Journal of Electronics and Communications*, vol. 94, pp. 293–302, Sep. 2018, doi: 10.1016/j.aeu.2018.07.021.
- [6] E. Tlelo-Cuautle, M. A. Valencia-Ponce, and L. G. de la Fraga, "Sizing CMOS amplifiers by PSO and MOL to improve DC operating point conditions," *Electronics*, vol. 9, no. 6, pp. 1027–1046, Jun. 2020, doi: 10.3390/electronics9061027.
- [7] K. Loubna, B. Bachir, and Z. Izeddine, "Ant colony optimization for optimal low-pass state variable filter sizing," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 1, pp. 227–235, Feb. 2018, doi: 10.11591/ijece.v8i1.pp227-235.
- [8] M. Akbari, M. Shokouhifar, O. Hashemipour, A. Jalali, and A. Hassanzadeh, "Systematic design of analog integrated circuits using ant colony algorithm based on noise optimization," *Analog Integrated Circuits and Signal Processing*, vol. 86, no. 2, pp. 327–339, Feb. 2016, doi: 10.1007/s10470-015-0682-0.
- [9] B. Benhala, "Sizing of an inverted current conveyors by an enhanced ant colony optimization technique," in *2016 Conference on Design of Circuits and Integrated Systems (DCIS)*, Nov. 2016, pp. 1–5, doi: 10.1109/DCIS.2016.7845271.
- [10] A. K. Dwivedi, S. Ghosh, and N. D. Londhe, "Low power FIR filter design using modified multi-objective artificial bee colony algorithm," *Engineering Applications of Artificial Intelligence*, vol. 55, pp. 58–69, Oct. 2016, doi: 10.1016/j.engappai.2016.06.006.
- [11] D. Bose, S. Biswas, A. V. Vasilakos, and S. Laha, "Optimal filter design using an improved artificial bee colony algorithm," *Information Sciences*, vol. 281, pp. 443–461, Oct. 2014, doi: 10.1016/j.ins.2014.05.033.
- [12] S. Patel and R. A. Thakker, "An efficient artificial bee colony algorithm and analog circuit design environment," *WSEAS Transactions on Circuits and Systems*, vol. 16, pp. 108–122, 2017.

- [13] E.-G. Talbi, "A taxonomy of hybrid metaheuristics," *Journal of Heuristics*, vol. 8, no. 5, pp. 541–564, 2002, doi: 10.1023/A:1016540724870.
- [14] B. Benhala, "Hybridization approaches of metaheuristics for optimal analog circuit design," *International Journal Of Microwave And Optical Technology*, vol. 9, no. 6, pp. 421–428, 2014.
- [15] N. M. Laskar, K. Guha, I. Chatterjee, S. Chanda, K. L. Baishnab, and P. K. Paul, "HWPSO: A new hybrid whale-particle swarm optimization algorithm and its application in electronic design optimization problems," *Applied Intelligence*, vol. 49, no. 1, pp. 265–291, Jan. 2019, doi: 10.1007/s10489-018-1247-6.
- [16] M. A. M. Majeed and S. R. Patri, "A hybrid of WOA and mGWO algorithms for global optimization and analog circuit design automation," *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, vol. 38, no. 1, pp. 452–476, Jan. 2019, doi: 10.1108/COMPEL-04-2018-0175.
- [17] K. B. Maji, R. Kar, D. Mandal, and S. P. Ghoshal, "Optimal design of low voltage CMOS second generation current conveyor using hybrid cuckoo search and particle swarm optimization algorithm," in *2017 International Conference on Information Technology (ICIT)*, Dec. 2017, pp. 246–251, doi: 10.1109/ICIT.2017.51.
- [18] M. Dorigo, V. Maniezzo, and A. Colomi, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, Feb. 1996, doi: 10.1109/3477.484436.
- [19] M. Dorigo and T. Stützle, "The ant colony optimization metaheuristic: Algorithms, applications, and advances," in *Handbook of Metaheuristics*, 2003, pp. 250–285.
- [20] J. Yang, X. Shi, M. Marchese, and Y. Liang, "An ant colony optimization method for generalized TSP problem," *Progress in Natural Science*, vol. 18, no. 11, pp. 1417–1422, Nov. 2008, doi: 10.1016/j.pnsc.2008.03.028.
- [21] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *Rainer Storn and Kenneth Price*, vol. 11, no. 4, pp. 341–359, 1997, doi: 10.1023/A:1008202821328.
- [22] M. Jamil, X.-S. Yang, and H.-J. Zepernick, "Test functions for global optimization," in *Swarm Intelligence and Bio-Inspired Computation*, Elsevier, 2013, pp. 193–222.
- [23] I. Ismail and A. H. Halim, "Comparative study of meta-heuristics optimization algorithm using benchmark function," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 3, pp. 1643–1650, Jun. 2017, doi: 10.11591/ijece.v7i3.pp1643-1650.
- [24] A. Sedra and K. Smith, "A second-generation current conveyor and its applications," *IEEE Transactions on Circuit Theory*, vol. 17, no. 1, pp. 132–134, Feb. 1970, doi: 10.1109/TCT.1970.1083067.
- [25] A. Chatterjee, M. Fakhfakh, and P. Siarry, "Design of second-generation current conveyors employing bacterial foraging optimization," *Microelectronics Journal*, vol. 41, no. 10, pp. 616–626, Oct. 2010, doi: 10.1016/j.mejo.2010.06.013.
- [26] M. Kotti, A. Sallem, M. Fakhfakh, and M. Loulou, "A novel multi-objective algorithm: application to the optimal sizing of current conveyors," in *2010 Xlth International Workshop on Symbolic and Numerical Methods, Modeling and Applications to Circuit Design (SM2ACD)*, Oct. 2010, pp. 1–5, doi: 10.1109/SM2ACD.2010.5672353.
- [27] A. El Dor, M. Fakhfakh, and P. Siarry, "Performance optimization of CMOS second generation current conveyors using a multi-swarm algorithm," *AEU-International Journal of Electronics and Communications*, vol. 68, no. 6, pp. 496–503, Jun. 2014, doi: 10.1016/j.aeue.2013.12.007.

BIOGRAPHIES OF AUTHORS



Soufiane Abi    received a Master's degree from the National School and Applied Sciences (ENSA) of Fez. Currently, he is a Ph.D. student at the Faculty of Sciences, Moulay Ismail University in Meknes, Morocco. His research interests include metaheuristic-based techniques, hybrid metaheuristics, and analog circuit design automation. He can be contacted at email: sou.abi@edu.umi.ac.ma.



Bachir Benhala    received a Master's degree from the Faculty of Science and Technology (FST) of Fez and his Ph.D. degree in 2013 in Electronics and Microelectronics. He is currently a Professor at the Faculty of Sciences, Moulay Ismail University in Meknes, Morocco. His research interests include metaheuristic optimization techniques, analog design automation, and digital/analog VLSI architectures. He can be contacted at email: b.benhala@umi.ac.ma.