J. Automation & Systems Engineering 9-3 (2015): 173-184

Regular paper

Naturally Based Optimisation Algorithm for Analogue Electronic Circuits: GA, PSO, ABC, BFO, and Firefly a Case Study

Ogri J. Ushie[1,*], Maysam Abbod[2], and Evans C. Ashigwuike[3]
[1, 2, 3] Department of Electronic and Computer Engineering, College of Engineering, Design and Physical Sciences, Brunel University London, Uxbridge, UB8 3PH, UK.
Ogri.Ushie@brunel.ac.uk

JASE
Journal of Automation & Systems Engineering

Abstract- This work presents various intelligence optimisation techniques. Namely: genetic algorithm (GA), particle swarm optimisation (PSO), artificial bee colony (ABC), bacterial foraging optimisation (BFO), and Firefly algorithm (FA). It attempts to optimise analogue electronic circuits using cascode amplifier design by applying all the swarm intelligence algorithms. Small signal model transformation of the cascode circuit is carried out while mesh analysis is used to convert the circuit to matrix form. Codes are developed in Matlab using GA, PSO, ABC, BFO, and (FA) to optimise the cascode amplifier circuit. The objective function is based on power consumption, lower frequency band, upper-frequency band, gain, input resistance and output resistance. Results are presented and compared to the original circuit simulated in PSPICE and an optimised circuit using Nelder-Mead constrained nonlinear minimisation algorithm. The PSO result has proven to be the best followed by GA in terms of frequency response and power consumption reduction. Three circuit elements are targeted (C, R, and transistors number) using the various optimisation methods then compared to the initial circuit.

Keywords: Electronic Amplifier, GA, PSO, ABC, BFO, FA and Small Signal Model.

1. INTRODUCTION

Power consumption in electronic circuits has been a source of concern for engineers because of its effect on the environment. Minimisation of electronic circuits helps to reduce component count, reduce power consumption, and increase system reliability. Hence, this approach is applied to cascode amplifier circuit.

Optimisation can be defined as the process of getting the best solution for a model [1]. The collection of individuals that make up the population is called a swarm. As long as the society continues to exist, the need to improve its standard of living also continues. Societal problems can come from a diverse field such as engineering, finance, manufacturing, music, computational art, medicine, physics, and chemistry. The quest to find the best solution is faced in everyday life and cannot be overemphasised. There are different swarm intelligence optimisations techniques such as: Particle Swarm Optimisation (PSO), artificial bee algorithm (ABC), bacterial foraging optimisation (BFO), and firefly algorithm (FA).

This paper surveys five intelligent methods mentioned in the title and use them to optimise cascode amplifier circuits and compare their performance in terms of frequency response, power consumption, gain, input and output resistance.

2. *Genetic Algorithm*

Genetic Algorithm (GA) is a population–based global search which applies the principle of survival of fittest to get a better solution [2]. During iteration, individuals are selected according to their outcome in the problem space. A group of individuals are created in the procedures that are better suited to the environment. The individuals are then encoded accordingly as strings; chromosome composed over some alphas, so that genotypes are mapped [3]. After the decoding, the fitness is calculated and provides criteria for the selection of individuals for future reproduction. GA operators are; crossover, selection, and mutation.

GA uses population called potential solutions composing between 30 to 100 individuals, even though micro GA applies small potential solutions (10 individuals) [3]. Chromosomes are represented in GA as grey coding, single-level binary string, and the real number representation as detailed in [3]. Application of genetic learning that has a memory of past solution to a problem that learnt to improve the quality of the result for related design problem is in [4]. Same algorithm is applied to parity checker, and the presented result has improvement.

Furthermore, GA algorithm extension for better performance is in [5], in their previous work, which allows parasitic effect in components to be contained within analogue circuit. The method is demonstrated in a filter circuit that is successful. Also, an automated algorithm application for optimisation of analogue circuit is found in [6]. An automated combinational circuit design applying a genetic algorithm is shown [7]. Besides, a related filter design circuits applying GA are [8-12]. GA uses in digital circuit are also in [4] [13]. Automated algorithm for CMOS amplifiers while using Genetic Programming (GP) and current-flow analysis is found in [14]. Electromagnetic GA optimisation (EGO) is in [15]. GA use in power electronic circuit minimisation is in [16].

3. Swarm Intelligent Optimisation Methods

Social behaviour of animals is used to develop an algorithm to solve societal problems. Individual among the group interacts so that the problem is better solved than just an individual contribution. In PSO, the individual specious is known as particle. A collection of the individual animal makes up the swarm. The problem-solving behaviour, which develops from the communications among swarm of specious, is called swarm intelligence (SI) [1].

3.1. ***Particle Swarm Optimisation***

PSO is a population-based optimisation technique developed by Kennedy and Eberhart 1995 [17] motivated by social behaviour of a flock of bird or school of fish. PSO has no genetic operators such as mutation, reproduction and crossover when compare to GA but dynamically adjust its velocity. Furthermore, PSO has fewer parameters, and it does not apply survival of the fittest. In PSO, particles are flown in search of the needed solution in the problem domain and each potential solution is renewed in the process. The collection of the particles as a whole is called a swarm. PSO algorithm that is suitable for parameter selection guide [18][19]. PSO algorithm applied to power electronics circuits (PECS) design is presented in [20].

Many studies on the use of particle swarm optimisation in analogue circuit designs are as follow: a work, which shows how analogue circuit performance can be enhanced using PSO algorithm [21] is being presented. Optimal design of analogue circuits applying PSO technique is in [22]. The emphasis is on how suitability of PSO to resolve both multi-objective and mono-objective discrete optimisation problem. In addition, the usages of PSO in microwave amplifier are in [23] [24]. Furthermore, PSO for design of analogue circuits is described in [25]. Analogue signal processing has many applications in an op amp based amplifiers, filters, mixers, and comparators. The quality of the result presented is compared to programme simulated with PSPICE. PSO approach provides accurate result and promising technique for device modelling in analogue circuit.

Photonic band gap and its application for low-pass filter design are in [26]. The firs-time use of Mini-max filter design using electromagnetic simulations is presented in [27]. Noise optimisation in operational trans-conductance amplifier filters is explored and compared as regard the filter parameters and topologies are in [28]. Despite all these studies, there is no research that tries to optimise an analogue circuit in terms of component count reduction, and power consumption reduction which this work intends to address.

*3.2. **Artificial Bee Colony Algorithm***

Honey bees forage, store honey in constructed colony and live in a colony. Honey bees communicate by 'waggle dance' and pheromone. Whenever the bees locate food source and bring nectar to the hive, the food source is communicated by waggle dance that vary from species to species [29]. Artificial bee colony (ABC) algorithm was developed by Karaboga in 2005 [29]. The bees in a colony are classified into three groups: onlooker bees (observer bees), employed bees (forager bees), and scouts. For a given food source, only one employed bee is involved. In other word the number of food sources are equal to the number of employed bees. Employed bee of the discarded food site becomes a scout to searching for new food sources. Employed bees and onlooker bees share information with each other to enable onlooker bees choose a food source to forage. It is a population-based, stochastic technique that carries out neighbourhood search. Foraging process includes: use ABC algorithm for scheduling in [30-32], for Structural optimisation [33]. ABC algorithms are used for electric load forecasting [34]. A modified version of a best-so-far for solution update in the ABC algorithm can be found in [35] where the authors claim that the improved algorithm perform better than ABC algorithm. The ABC algorithm is applied to optimise Nano-CMOS analogue present in [36]. Furthermore, ABC application to solve travelling salesman problems is found in [37]. Another use of ABC algorithm in electronic circuit design includes [38-40]. The use of ABC in the design of CMOS inverter is presented in [41]. Honey bee special features such as communication, decision-making, foraging, allocation of task, marriage, mating, reproduction and nest site selection are being translated into bee algorithm as in [42].

*3.3. **Bacterial Foraging Optimisation***

The survival of an organism as regard its mobility and its search for food depends upon their fitness criteria, and this is referred to bacterial foraging optimisation. It is motivated by the chemotaxis behaviours of bacteria that makes it move away or toward specific signals whenever it senses any chemical in the environment. The organism reshapes the weaker organism with poor search experience. The genes of those species that are fittest are used for development chain and have promised better organisms in next generations. The foraging behaviour in evolutionary species is being converted into algorithms that are applied to non-linear optimisation model. Detailed BFO Algorithm is presented in [43-44].

BFO has been applied to non-linear optimisation models such as control and distributed optimisation as shown in [43]. BFO is used for the optimisation of Yagi-Uda array design in [44]. Bacterial foraging optimisation is applied in distributed control application in a sensor network [45]. In addition, Genetic algorithm bacterial foraging algorithm (GABFA) for proportional-integral-derivative (PID) controller of an automatic voltage regulator (AVR) system is proposed [46-47]. When the results are compared to GA, PSO, and GA-PSO, it shows its potential for optimisation. Also, the application of BFO for optimum injection of voltage and position for the unified power flow controller (UPFC) and the quality taps illustrates the potential of BFO [48]. In addition, BFO is application to minimise coefficient of the plus integral controller [49]. In another paper, Bacterial foraging algorithm is used to minimise the real power loss and to improve the VSL of the system [50]. The better result is due BFO principle application in allocating UPFC and transformer taps.

In the area of power system stability, BFO is applied to resolve the problem of unstable cases in CPSS and GAPSS [51]. Also, an enhanced PSS stability by using BFO was in [52].

Also, an improved adaptive bacterial foraging algorithm (ABFA) using the method of adaptive delta modulation is presented in [53]. Multi-colony bacterial foraging optimisation (MC-BFO) for complex radio frequency identification (RFID) network planning problem was in [54]. BFA was used to training NN in [55] and result presented shows that BFO-NN technique yields quality result than GA-NN as regard accuracy and convergence. BFO in power system stability were in [56][57].

*3.4. **Firefly Algorithm***

Flashing light behaviour of fireflies which produce rhythmic and short flashes uniquely to a given species is used to develop an algorithm. The essence of the flashes is to serve as warning to potential predators, attract potential prey and attract mating partners. Firefly algorithm is developed based on;

a) Fireflies are attracted to one another regardless of their sex (unisex).
b) Attractiveness is directly proportional to brightness of the flash.
c) Brightness of the firefly light is determined or affected by landscape of the objective function.

A decentralised algorithm for synchronicity based on firefly features is presented [58]. The firefly algorithm (FA) is applied to sensor system and result shows that the algorithms can synchronise sensor network. The surprising feature of firefly is the flashing light. The flashing pattern defines the species that are about two thousand in number [59]. The purpose of the flash is to entice mating partner, entice the potential prey and to notify warning mechanism.

The flashing features inspired firefly algorithm used in optimisation. The algorithm is shown in [60]; the first reference provides a detailed firefly algorithm while the second reference shows development of the algorithm. Presented result shows how PSO outperforms firefly algorithm. Furthermore, the use of FA to construct codebook for vector quantisation is in [61]. Higher quality result is presented for reconstructed images better than those obtained from the LBG, PSO, and QPSO.

4. Cascode Amplifier Design

Cascode amplifier design is used as a case study. The original circuit is shown in Figure 1 while the minimised circuit used to optimise the initial circuit is shown in Figure 2. The objective function is specified according to PSPICE simulation result while Matlab programme is written to find equivalent components that satisfy the specifications. The circuit is transformed into its small signal module as shown in Figure 3; while mesh analysis is applied to convert the circuit into matrix form. BFO, Firefly and ABC, GA and PSO algorithms are applied to minimise the cascode amplifier design circuit. The best particle or individual for a specified iteration is taken as a solution. The same circuits are run in Nelder-Mead constrained nonlinear minimisation to test the effectiveness of the approach. The optimise section of the circuit is seen as a black box, and input and output impedance is optimised such that they were not affected.

Some of the circuit parameters such as Rs, RL are treated as constants and have fixed value while others are given range of values as illustrated in Table 1. It enables the programme to select their values at random in order to evolve a circuit that satisfy the objective function. The multi-objective optimisation function is based on Power, lower-frequency band, upper-frequency band, gain, input resistance, and output resistance. Figure 4 summarises the method in the form of a flowchart.

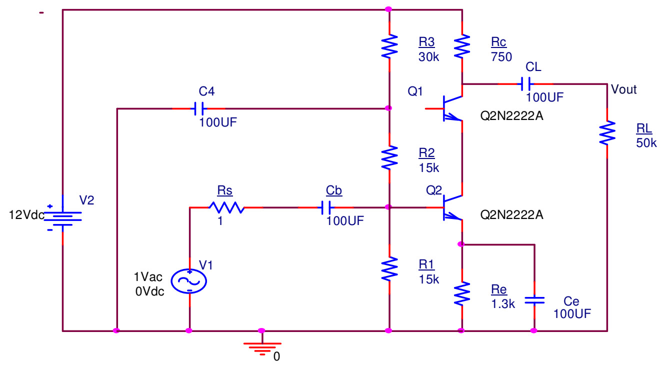

Figure 1 Cascode amplifier's initial circuit [62]

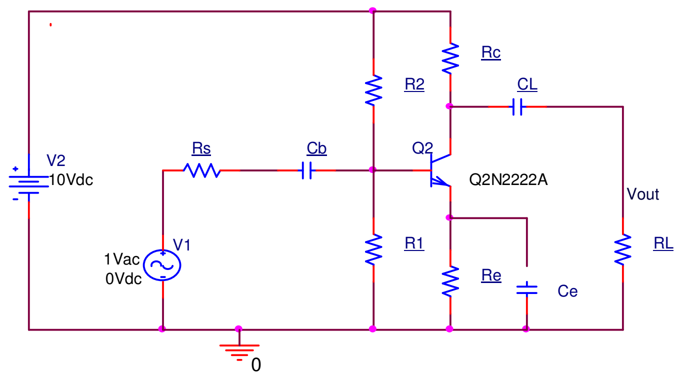

Figure 2 Cascode amplifier minimised circuit

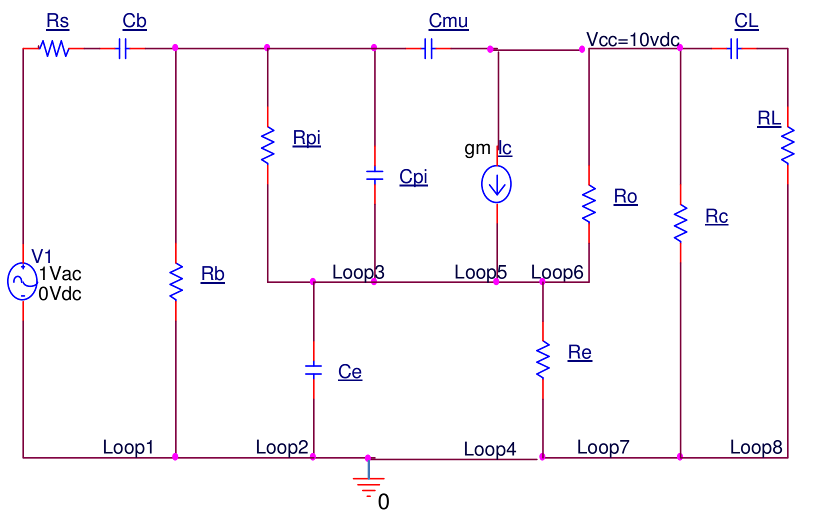

Figure 3 Small signal analysis of the minimised cascode amplifier circuit

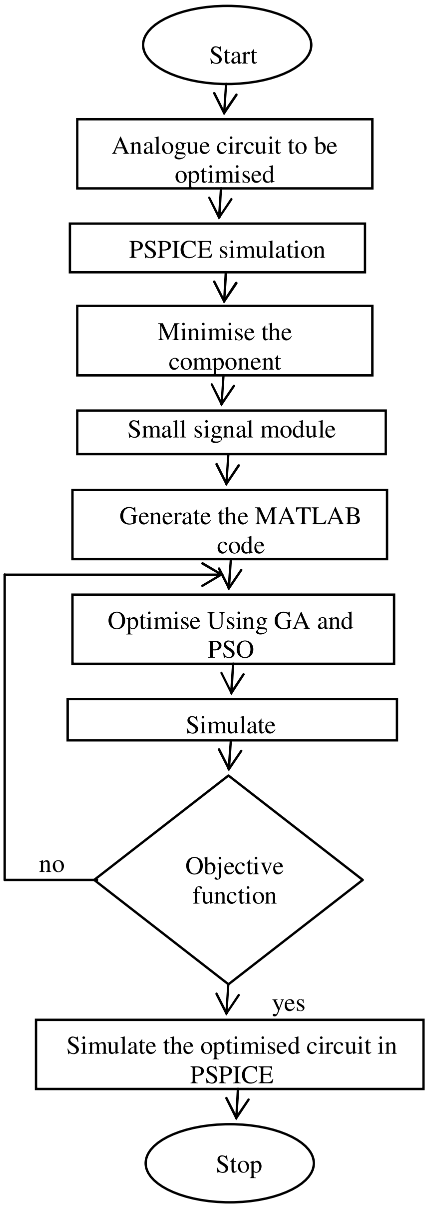

Figure 4 The proposed algorithm flow chart

The objective function is based on:

$$ff = cf_1 + cf_2 + \max power + A_v + R_{in} + R_{out} \quad (1)$$

where ff is the objective function, cf1 is the difference between the targeted and achieved lower-frequency band at -3dB, cf2 is the difference between the targeted and achieved upper-frequency band at -3dB, maximum power is the difference between the targeted and achieved maximum power required by the circuit, A_v is the range that the amplifier gain is optimised, while R_{in} and R_{out} are the range that the input and output resistances of the amplifier are optimised respectively. Their set values used in the minimisation are as follow:

cf_1 = targeted (cf_1) – 32.544 Hz
cf_2 = (targeted (cf_2) – 20.444E6)/E6 Hz
maxpower = 0.7236 mW
A_v = 20 to 46
R_{in} = 7 kΩ to 7.5 kΩ
R_{out} = 650 Ω to 800 Ω

The constants for this simulation are:
C_{pi}=10 pF, C_{mu}=2 pF, Rs=100 Ω, R_L=15 kΩ, R_o =100 kΩ, Vcc = 10 V, V_I=1 V, V_{be} = 0.648 V, I_C = 1 mA, h_{FE} = 250, q = 1.6E-19 C, k_b = 1.38e-23 J/K, Temperature = 300 K.

Table 1 Component ranges

Component name	Minimum value	Maximum value
CL (F)	0.01E-6	5E-6
Re (Ω)	200	400
Rc (Ω)	4.5e3	6e3
R_1 (Ω)	0.9e3	1.2e3
R_2 (Ω)	8.8e3	9.2e3
Cb (F)	0.1e-6	11e-6
Ce (F)	1e-6	20e-6

Formulas used in the simulation are:

$$V_T = k_b \times (Temp / q) \quad (2)$$

$$R_b = (R_1 \times R_2)/(R_1 + R_2) \quad (3)$$

$$\omega = 2 \times pi \times f \quad (4)$$

$$G_m = I_c / V_T \quad (5)$$

$$I_B = I_c / h_{FE} \quad (6)$$

$$R_{pi} = h_{FE} / G_m \quad (7)$$

$$R_{in} = (R_b \times R_{pi})/(R_b + R_{pi}) \quad (8)$$

$$R_{out} = (R_c \times R_o)/(R_c + R_o) \quad (9)$$

$$A_i = I_C / I_B \quad (10)$$

$$A_v = G_m \times R_c \times R_o /(R_c \times R_o)) \quad (11)$$

$$A^T = [10000000] \quad (12)$$

$$B = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 & 0 & 0 & 0 \\ 0 & a_{32} & a_{33} & 0 & a_{35} & 0 & 0 & 0 \\ 0 & a_{42} & 0 & a_{44} & 0 & 0 & a_{47} & 0 \\ 0 & 0 & a_{53} & 0 & a_{55} & a_{56} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{65} & a_{66} & a_{67} & 0 \\ 0 & 0 & 0 & a_{74} & 0 & a_{76} & a_{77} & a_{78} \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{87} & a_{88} \end{pmatrix} \quad (13)$$

Where $a_{11} = (R_S + R_b + (1/(j \times \omega \times C_b)))$

$a_{12} = -R_b$

$a_{21} = -R_b$

$a_{22} = (R_{pi} + R_b + (1/(j \times \omega \times C_e)))$

$a_{23} = -R_{pi}$

$a_{24} = -(1/(j \times \omega \times C_e))$

$a_{32} = -R_{pi}$

$a_{33} = (R_{pi} + (1/(j \times \omega \times C_{pi})))$
$a_{35} = -(1/(j \times \omega \times C_{pi}))$
$a_{42} = -(1/(j \times \omega \times C_{e}))$
$a_{44} = (R_{e} + (1/(j \times \omega \times C_{pi})))$
$a_{47} = -R_{e}$
$a_{53} = -(1/(j \times \omega \times C_{pi}))$
$a_{55} = -I_{C}$
$a_{56} = -(1/G_{m})$
$a_{65} = -(1/G_{m})$
$a_{66} = R_{o}$
$a_{67} = -R_{o}$
$a_{74} = -R_{e}$
$a_{76} = -R_{o}$
$a_{77} = (R_{o} + R_{c} + R_{e}$
$a_{78} = -R_{c}$
$a_{87} = -R_{c}$
$a_{88} = (R_{L} + R_{c} + (1/(j \times \omega \times C_{L})))$

$$C = B^{-1} \times A \qquad (14)$$

where *C* holds unidentified voltages on all the nodes to be determined. *C8* is the voltage across the load resistor and being analysed to get its frequency response within a certain range of frequency in this case 1 Hz to 1 GHz. The frequency response curves in Figs 5 are plot of gain in magnitude against frequency. Firefly parameter settings are: generation, N = 40, runtime = 100, N-iteration = 150, alpha = 0.5, betamin = 0.2, gama = 1. In addition, parameters setting for ABC are: size of colony, NP = 20, food number = NP/2, food source which must be improved (limit) = 100, number of cycle for foraging (maxcycle) = 2500, and runtime = 100.

Parameter setting in GA are; pop-size = 50, crossover (Pc) = 0.8, generation = 100 and mutation (Pm) = 0.05. While PSO parameters are; acceleration constants (c1 = c2 = 1.49618), initial weight (w = 0.7298), number of iteration = 30, number of particles = 20. Also, BFO parameter selections are: algorithm runtime = 100, number of bacteria in pop, S = 8, number of chemotactic step per bacteria lifetime, Nc = 5, Limits the length of a swim when it is on a gradient, Ns = 4, number of reproduction steps, Nre = 4, number of bacteria reproductions (splits) per generation, Sr = S/2, number of elimination-dispersal events, Ned = 2, and probability that each bacteria will be eliminated, Ped = 0.25.

5. Results and Discussion

The results obtained from the simulation are summarised in Table 2. The original circuit is simulated in SPICE to obtained frequency response, bandwidth and power. Figure 5 shows frequency response curves with different line style for the original circuit, Nelder-Mead, BFO, Firefly and ABC, GA and PSO optimised circuits. Results presented have shown that PSO and GA are useful minimisation tools for electronics but PSO is the best among all the five swarm algorithm methods in terms of frequency response and power reduction. Results presented also revealed that Nelder-Mead, BFO, Firefly and ABC are not good because their power consumption increases instead of power reduction and narrow bandwidth as regard their frequency response. It further revealed that PSO is better than GA in terms of higher power reduction and frequency response.

Table 2 Simulation optimisation results

Circuit element	Initial Circuit	Nelder-Mead	Firefly	ABC	BFO	GA Circuit	PSO Circuit
C_L (µF)	100	18.74	13.07	11.98	24.79	6.86	14.66
Re (kΩ)	1.3	1.91	1.92	0.99	2.70	1.43	2.76
Rc (Ω)	750	504	990	621	533	665	549
Cb (µF)	100	9.76	0.11	5.49	1.57	6.13	11
R1 (kΩ)	15	14.75	0.50	4.99	12.82	4.86	9.57
R2 (kΩ)	15	14.75	4.20	7.10	3.81	14.13	12.12
Ce (µF)	100	48.73	26.42	50	24.57	30.58	50
R_3 (kΩ)	30	-				-	-
C_4 (µF)	100	-				-	-
Q2N2222	2	1	1	1	1	1	1
Rs (Ω)	1	1	1	1	1	1	1
R_L (kΩ)	50	50	50	50	50	50	50
V_1ac (volt)	1	1	1	1	1	1	1
V_2dc (volt)	12	12	12	12	12	12	12
Earth	1	1	1	1	1	1	1
cf_1(Hz)	102	338.80	3.27E3	410.87	831.80	338.80	204.20
cf_2 (MHz)	48.95	36.31	47.70	26.02	23.24	45.69	47.75
Power (mW)	22.10	25.5	23.6	42.4	31.9	18.4	18.0
Percentage of power change		+15.4%	+6.79%	+91.86%	+44.3%	-16.7%	-18.6%
Number of component	16	13	13	13	13	13	13
Objective function error		6.74E3	53.44	7.47E3	6.98E3	6.83E3	6.70E3

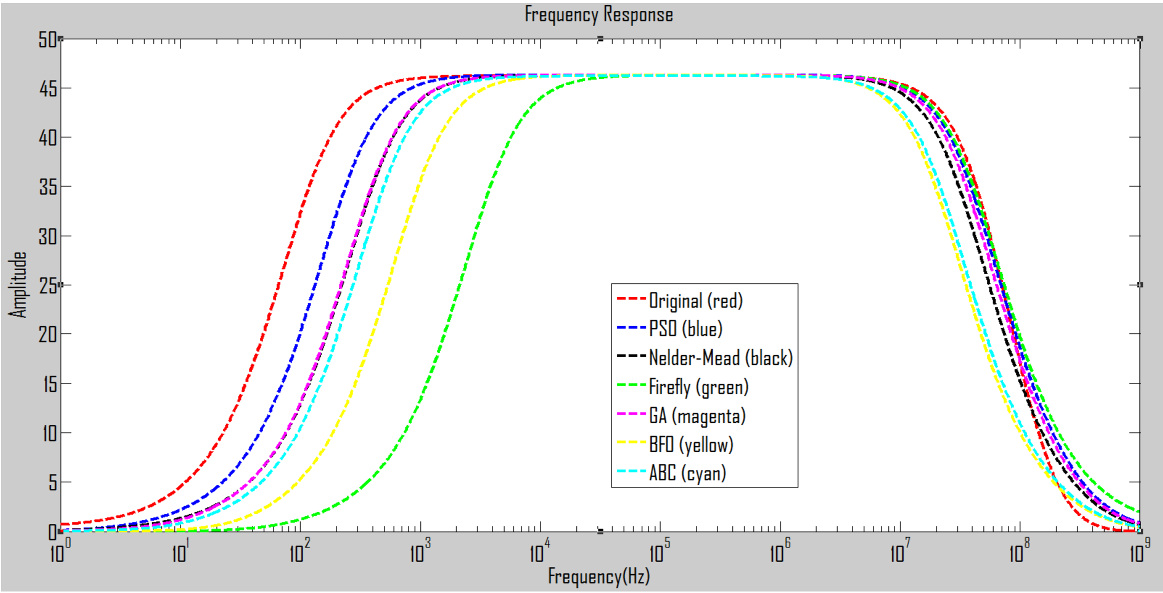

Figure 5 Frequency response curve for initial cascode circuit and all the optimised circuits

6. Conclusion

This work presents five different swarm optimisation algorithms for analogue circuit optimisation. In the illustrated example, primarily it shows how equivalent analogue circuit can be found. It further shows that component count reduction can also be achieved in analogue circuit just as it has been carried out in digital circuits. Original circuit has sixteen (16) components while the other optimised five methods (Nelder-Mead, BFO, Firefly, ABC, GA, and PSO) all obtained thirteen (13) components, but at different level of power consumption as illustrated in Table 2. In the approach, Nelder-Mead, BFO, Firefly, and ABC increased power consumption at different values. However, PSO and GA reduced power consumption at different levels.

ACKNOWLEDGMENT

The lead author acknowledges the financial support from Tertiary Education Trust Fund (TetFund) through University of Calabar, Calabar, Nigeria. The authors appreciate the comments provided by the anonymous reviewers and the editor, which has help the authors to improve the quality of the paper.

References

[1] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence.* Wiley Chichester, 2005.
[2] A. Chipperfield, P. Fleming, H. Pohlheim and C. Fonseca, "Genetic algorithm toolbox for use with MATLAB," 1994.
[3] M. O'Dare and T. Arslan, "Generating test patterns for VLSI circuits using a genetic algorithm," *Electron. Lett.,* vol. 30, pp. 778-779, 1994.
[4] D. H. Horrocks and Y. M. Khalifa, "Genetic algorithm design of electronic analogue circuits including parasitic effects," in *Proceedings of the First Online Workshop on Soft Computing (WSC1),* 1996, pp. 274-278.
[5] M. Wojcikowski, J. Glinianowicz and M. Bialko, "System for optimisation of electronic circuits using genetic algorithm," in *Electronics, Circuits, and Systems, 1996. ICECS'96., Proceedings of the Third IEEE International Conference on,* 1996, pp. 247-250.
[6] C. A. Coello, A. D. Christiansen and A. H. Aguirre, "Automated design of combinational logic circuits using genetic algorithms," in *Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms,* 1997, pp. 335-338.
[7] J. R. Koza, F. H. Bennett III, D. Andre, M. A. Keane and F. Dunlap, "Automated synthesis of analogue electrical circuits by means of genetic programming," *Evolutionary Computation, IEEE Transactions on,* vol. 1, pp. 109-128, 1997.

[8] J. D. Lohn and S. P. Colombano, "Automated analogue circuit synthesis using a linear representation," in *Evolvable Systems: From Biology to Hardware from nasa.gov* Springer, 1998, pp. 125-133.
[9] R. S. Zebulum, M. A. Pacheco and M. Vellasco, "Comparison of different evolutionary methodologies applied to electronic filter design," in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., the 1998 IEEE International Conference on,* 1998, pp. 434-439.
[10] J. R. Koza, F. H. Bennett III and O. Stiffelman, *Genetic Programming as a Darwinian Invention Machine.* Springer, 1999.
[11] J. D. Lohn and S. P. Colombano, "A circuit representation technique for automated circuit design," *Evolutionary Computation, IEEE Transactions on,* vol. 3, pp. 205-219, 1999.
[12] J. F. Miller, D. Job and V. K. Vassilev, "Principles in the evolutionary design of digital circuits—Part I," *Genetic Programming and Evolvable Machines,* vol. 1, pp. 7-35, 2000.
[13] T. Sripramong and C. Toumazou, "The invention of CMOS amplifiers using genetic programming and current-flow analysis," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on,* vol. 21, pp. 1237-1252, 2002.
[14] F. J. Villegas, T. Cwik, Y. Rahmat-Samii and M. Manteghi, "A parallel electromagnetic genetic-algorithm optimisation (EGO) application for patch antenna design," *Antennas and Propagation, IEEE Transactions on,* vol. 52, pp. 2424-2435, 2004.
[15] J. Zhang, H. S. Chung, W. Lo, S. Hui and A. Wu, "Implementation of a decoupled optimisation technique for design of switching regulators using genetic algorithms," *Power Electronics, IEEE Transactions on,* vol. 16, pp. 752-763, 2001.
[16] J. Zhang, H. S. Chung and W. Lo, "Pseudocoevolutionary genetic algorithms for power electronic circuits optimisation," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* vol. 36, pp. 590-598, 2006.
[17] J. F. Kennedy, J. Kennedy and R. C. Eberhart, *Swarm Intelligence.* Morgan Kaufmann, 2001.
[18] I. C. Trelea, "The particle swarm optimisation algorithm: convergence analysis and parameter selection," *Information Processing Letters,* 85, 6, 317-325, Elsevier, 2003.
[19] L. Zhou, Y. Shi, Y. Li and W. Zhang, "Parameter selection, analysis and evaluation of an improved particle swarm optimiser with leadership," *Artif. Intell. Rev.,* vol. 34, pp. 343-367, 2010.
[20] J. Zhang, Y. Shi and Z. Zhan, "Power electronic circuits design: A particle swarm optimisation approach," in *Simulated Evolution and Learning, 7th International Conference*, Melbourne, Australia, Springer, pp 605-614, 2008.
[21] M. Fakhfakh, Y. Cooren, M. Loulou and P. Siarry, "A Particle Swarm Optimisation technique used for the improvement of analogue circuit performances," *Particle Swarm Optimization,* pp. 169-182, 2009.
[22] M. Fakhfakh, Y. Cooren, A. Sallem, M. Loulou and P. Siarry, "Analog circuit design optimisation through the particle swarm optimisation technique," *Analogue Integr. Cir. Signal Proc.,* vol. 63, pp. 71-82, 2010.
[23] S. Ulker, "Design of Low Noise Microwave Amplifiers Using Particle Swarm Optimisation," *International Journal of Artificial Intelligence & Applications,* vol. 3, 2012.
[24] S. Ulker, "Broadband Microwave Amplifier Design Using Particle swarm optimisation," *Journal of Computers,* vol. 11, pp. 2272-2276, 2011.
[25] P. P. Kumar and K. Duraiswamy, "An Optimised Device Sizing of Analog Circuits using Particle Swarm Optimisation," *Journal of Computer Science,* vol. 8, pp. 930, 2012.
[26] J. Park, C. Kim, J. Kim, J. Park, Y. Qian, D. Ahn and T. Itoh, "Modeling of a photonic bandgap and its application for the low-pass filter design," in *Microwave Conference, 1999 Asia Pacific,* pp. 331-334, 1999.
[27] J. Bandler, S. Ye, R. Biernacki, S. Chen and D. Swanson Jr, "Minimax microstrip filter design using direct EM field simulation," in *Microwave Symposium Digest, 1993., IEEE MTT-S International,* pp. 889-892, 1993.
[28] A. Brambilla, G. Espinosa, F. Montecchi and E. Sánchez-Sinecio, "Noise optimization in operational transconductance amplifier filters," in *Circuits and Systems, 1989., IEEE International Symposium on,* pp. 118-1211, 989.
[29] X. Yang, Engineering Optimization: An Introduction with Metaheuristic Applications. John Wiley & Sons, 2010.
[30] C. S. Chong, M. Y. H. Low, A. I. Sivakumar and K. L. Gay, "A bee colony optimisation algorithm to job shop scheduling," in *Simulation Conference, 2006. WSC 06. Proceedings of the Winter,* 2006, pp. 1954-1961.
[31] K. Ziarati, R. Akbari and V. Zeighami, "On the performance of bee algorithms for resource-constrained project scheduling problem," *Applied Soft Computing,* vol. 11, pp. 3720-3733, 2011.
[32] Q. Pan, M. Fatih Tasgetiren, P. N. Suganthan and T. J. Chua, "A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem," *Inf. Sci.,* vol. 181, pp. 2455-2468, 2011.
[33] A. Hadidi, S. K. Azad and S. K. Azad, "Structural optimisation using artificial bee colony algorithm," in *2nd International Conference on Engineering Optimisation, Lisbon, Portugal,* 2010.
[34] W. Hong, "Electric load forecasting by seasonal recurrent SVR (support vector regression) with chaotic artificial bee colony algorithm," *Energy,* vol. 36, pp. 5568-5578, 2011.
[35] A. Banharnsakun, T. Achalakul and B. Sirinaovakul, "The best-so-far selection in artificial bee colony algorithm," *Applied Soft Computing,* vol. 11, pp. 2888-2901, 2011.

[36] O. Garitselov, S. P. Mohanty, E. Kougianos and P. Patra, "Bee colony inspired metamodeling based fast optimisation of a nano-CMOS PLL," in *Electronic System Design (ISED), 2011 International Symposium on,* 2011, pp. 6-11.
[37] D. Karaboga and B. Gorkemli, "A combinatorial artificial bee colony algorithm for traveling salesman problem," in *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on,* 2011, pp. 50-53.
[38] N. Karaboga, "A new design method based on artificial bee colony algorithm for digital IIR filters," *Journal of the Franklin Institute,* vol. 346, pp. 328-348, 2009.
[39] V. Manoj and E. Elias, "Artificial bee colony algorithm for the design of multiplier-less nonuniform filter bank transmultiplexer," *Inf. Sci.,* vol. 192, pp. 193-203, 2012.
[40] M. H. Kashan, N. Nahavandi and A. H. Kashan, " DisABC: A new artificial bee colony algorithm for binary optimisation," *Applied Soft Computing,* vol. 12, pp. 342-352, 2012.
[41] Y. Delican, R. Vural and T. Yildirim, "Artificial bee colony optimisation based cmos inverter design considering propagation delays," in *Symbolic and Numerical Methods, Modeling and Applications to Circuit Design (SM2ACD), 2010 XIth International Workshop on,* 2010, pp. 1-5.
[42] D. Karaboga, B. Gorkemli, C. Ozturk and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artif. Intell. Rev.,* pp. 1-37, 2012.
[43] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimisation and control," *Control Systems, IEEE,* vol. 22, pp. 52-67, 2002.
[44] B. Mangaraj, I. Misra and S. Sanyal, "Application of bacteria foraging algorithm for the design optimisation of multi-objective Yagi-Uda array," *International Journal of RF and Microwave Computer-Aided Engineering,* vol. 21, pp. 25-35, 2011.
[45] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert and S. S. Sastry, "Distributed control applications within sensor networks," *Proc IEEE,* vol. 91, pp. 1235-1246, 2003.
[46] D. H. Kim and J. H. Cho, "Intelligent control of AVR system using GA-BF," in *Knowledge-Based Intelligent Information and Engineering Systems,* 2005, pp. 854-859.
[47] D. H. Kim, A. Abraham and J. H. Cho, "A hybrid genetic algorithm and bacterial foraging approach for global optimisation," *Inf. Sci.,* vol. 177, pp. 3918-3937, 2007.
[48] M. Tripathy, S. Mishra, L. Lai and Q. Zhang, "Transmission loss reduction based on FACTS and bacteria foraging algorithm," in *Parallel Problem Solving from Nature-PPSN IX* Parallel Problem Solving from Nature - PPSN IX Lecture Notes in Computer ScienceVolume 4193, Springer, 2006, pp. 222-231.
[49] S. Mishra and C. Bhende, "Bacterial foraging technique-based optimised active power filter for load compensation," *Power Delivery, IEEE Transactions on,* vol. 22, pp. 457-465, 2007.
[50] M. Tripathy and S. Mishra, "Bacteria foraging-based solution to optimise both real power loss and voltage stability limit," *Power Systems, IEEE Transactions on,* vol. 22, pp. 240-248, 2007.
[51] S. Mishra, M. Tripathy and J. Nanda, "Multi-machine power system stabilizer design by rule based bacteria foraging," *Electr. Power Syst. Res.,* vol. 77, pp. 1595-1607, 2007.
[52] B. Sumanbabu, S. Mishra, B. Panigrahi and G. K. Venayagamoorthy, "Robust tuning of modern power system stabilizers using bacterial foraging algorithm," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on,* 2007, pp. 2317-2324.
[53] T. Datta, I. S. Misra, B. B. Mangaraj and S. Imtiaj, "Improved adaptive bacteria foraging algorithm in optimisation of antenna array for faster convergence," *Progress in Electromagnetic Research C,* vol. 1, pp. 143-157, 2008.
[54] H. Chen, Y. Zhu and K. Hu, "Multi-colony bacteria foraging optimisation with cell-to-cell communication for RFID network planning," *Applied Soft Computing,* vol. 10, pp. 539-547, 2010.
[55] Y. Zhang, L. Wu and S. Wang, "Bacterial foraging optimisation based neural network for short-term load forecasting," *Journal of Computational Information Systems,* vol. 6, pp. 2099-2105, 2010.
[56] D. Sumina, N. Bulić and I. Erceg, "Three-dimensional power system stabilizer," *Electr. Power Syst. Res.,* vol. 80, pp. 886-892, 2010.
[57] E. Ali and S. Abd-Elazim, "Coordinated design of PSSs and TCSC via bacterial swarm optimisation algorithm in a multimachine power system," *International Journal of Electrical Power & Energy Systems,* vol. 36, pp. 84-92, 2012.
[58] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh and R. Nagpal, "Firefly-inspired sensor network synchronicity with realistic radio effects," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems,* 2005, pp. 142-153.
[59] X. Yang, "Firefly algorithms for multimodal optimisation," in *Stochastic Algorithms: Foundations and Applications from arsxiv.org* Springer, 2009, pp. 169-178.
[60] S. Łukasik and S. Żak, "Firefly algorithm for continuous constrained optimisation tasks," in *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*, Lecture Notes in Computer ScienceVolume 5796, Springer, 2009, pp. 97-106.
[61] M. Horng, "Vector quantisation using the firefly algorithm for image compression," *Expert Syst. Appl.,* vol. 39, pp. 1078-1091, 2012.
[62] G. T. A. Kovacs, EE113 Course Notes Electronic Circuits, Stanford University, Department of Electrical Engineering, p. 161, 1997.