



## Design of a CMOS Gilbert Cell Mixer Using Differential Evolutionary Algorithm

Hamidreza Rezaee-D. , Amin Ibrahimy-K. , Mohammad Danaie

Department of Electrical Engineering,

Ferdowsi University of Mashhad, Iran

E-Mails: {hr.rezaee@gmail.com, amin.i.k.8@gmail.com, mdanaie@gmail.com}

**Abstract:** This paper introduces a simulation-based differential evolutionary optimization algorithm for design of analog integrated circuits. This algorithm happens to be much faster than other none derivative-based algorithms mentioned in literature. A 2.4GHz CMOS Gilbert Mixer has been chosen in this case. In order to evaluate the time of convergence, the algorithm is compared with a GA optimization procedure, using automated iterations of MATLAB-APLAC in a 0.18 $\mu$  process. The differential evolutionary algorithm proves to be several times faster, even for smaller population sizes.

**Keywords:** Automatic analog circuit design, Optimization, Evolutionary algorithms, Gilbert Cell mixer, DE, GA.

### 1 Introduction

With the fast growth of communication systems, design of efficient high speed low power RF circuits is becoming very important. Because of the complexity of models in high frequencies and reduction of supply voltages, the design of such circuits is very time-consuming for the designer, thereby computer aided design automation of such circuits seems to be very helpful. Given a specified topology, this tool should be able to find the optimum values of circuit elements which satisfy the needs.

There have been several analog tools developed for specific applications such as operational amplifiers and filters [1-8], but most of them suffer from low precision and rough modeling due to use of simple macro models [9, 10].

The methods which could be used for circuit optimizations are divided to simulation-based and

equation-based methods. It is obvious that because of high order effects and high frequency phenomena in MOS transistors equation-based methods are not reliable. Evolutionary algorithms are among the best methods used for simulation-based optimization. The reason is their adaptability with discrete functions, higher speed compared to random approaches, and not becoming trapped in local optima.

Genetic algorithms (GA) are usually used in this case. These algorithms are proven to find the global optima, but as the complexity of circuit increases their convergence time grows rapidly. Thereby GA becomes useless for circuits of high complexity. In order to solve this problem an algorithm called Differential Evolutionary Algorithm (DE) is proposed. This algorithm has a high convergence rate and is times faster than GA. The DE algorithm is discussed in Section 2, in Section 3 GA is compared with DE, Section 4 concerns the circuit optimization and Section 5 deals with simulation results.

### 2 Differential evolutionary algorithm

The DE algorithm is a population based algorithm like genetic algorithms using the similar operators; crossover, mutation and selection. The main difference in constructing better solutions is that genetic algorithms rely on crossover while DE relies on mutation operation. This main operation is based on the differences of randomly sampled pairs of solutions in the population. The algorithm uses mutation operation as a search mechanism and selection operation to direct the search toward the prospective regions in the search space. The DE algorithm also uses a non-uniform crossover

that can take child vector parameters from one parent more often than it does from others. By using the components of the existing population members to construct trial vectors, the recombination (crossover) operator efficiently shuffles information about successful combinations, enabling the search for a better solution space [11].

DE is a parallel direct search method which utilizes NP D-dimensional parameter vectors:

$$X_{i,G}, i = 0, 1, 2, \dots, NP - 1 \quad (1)$$

NP doesn't change during the minimization process. The initial population is chosen randomly and should try to cover the entire parameter space uniformly.

The crucial idea behind DE is a scheme for generating trial parameter vectors. Basically, DE generates new parameter vectors by adding the weighted difference between two population vectors to a third vector (Figure 1). If the resulting vector yields a lower objective function value than a predetermined population member, the newly generated vector will replace the vector with which it was compared in the following generation; otherwise, the old vector is retained. In most cases, it is also worthwhile to mix the parameters of the old vector with those of the perturbed one. The performance of the resulting vector is then compared to that of the old vector [11, 12, 13, and 14].

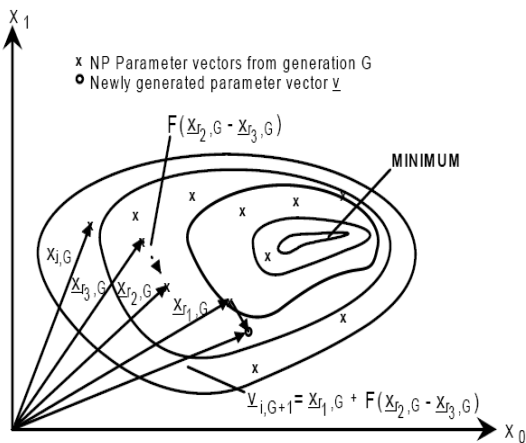


Figure 1: An example of vector selection for a two dimensional function

$$V_{i,G+1} = X_{r1,G} + F \times (X_{r2,G} - X_{r3,G}) \quad (2)$$

With  $r_1, r_2, r_3 \in [0, NP-1]$ , integer and mutually different, and assuming that  $F \in [0, 2]$ . Factor F

determines the weight of  $(X_{r2,G} - X_{r3,G})$ . This kind of producing new vectors is called mutation, while crossover (or recombination) is defined as follows:

$$U_{i,G+1} = (U_{0i,G+1}, U_{1i,G+1}, \dots, U_{(D-1)i,G+1}) \quad (3)$$

When:

$$U_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{for } j = \langle n \rangle_D, \langle n+1 \rangle_D, \dots, \langle n+L-1 \rangle_D \\ x_{ji,G} & \text{for all other } j \in [0, D-1] \end{cases} \quad (4)$$

The acute brackets  $\langle \rangle_D$  denote modulo function with modulus D. The starting index, n, in (4) is a randomly chosen integer from the interval  $[0, D-1]$ . The integer L, which denotes the number of parameters that are going to be exchanged, is drawn from the interval  $[1, D]$ .

### 3 DE versus GA

Since GA and DE are both evolutionary algorithms they are similar to each other. They both have crossover, mutation, and selection. Their main difference is due to order of their sections and the way each procedure is carried out.

The first section of both algorithms is Initialization. In this section random or typical values are chosen for the circuit parameters. These parameters could be transistors length and width, resistor or capacitor values, bias voltages or currents or etc.

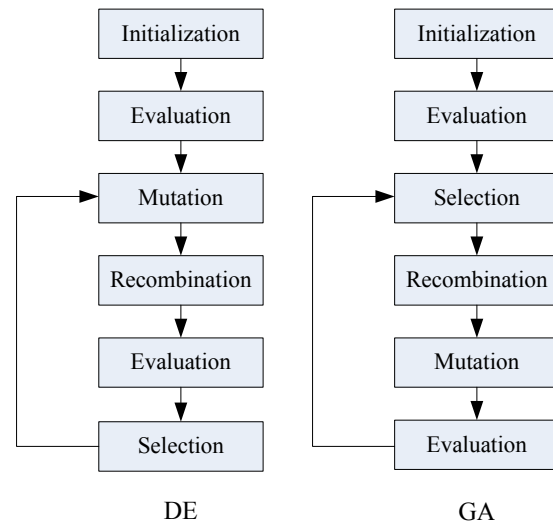


Figure 2: DE and GA optimization procedures

Next stage is the evaluation stage. This section analyzes the circuit and gathers output results. An evaluation function allocates a number to each of the individuals whose magnitude shows how good

the obtained result is. The selection section chooses the individuals for the next generation with a probability according to their fitness. As it could be seen in Figure 2 the selection section is the last stage in DE algorithms this causes the next generation to be most likely to comprise of the best individuals of the current generation, while in GA the mutation and recombination which is done after the selection might lead to poor output results (see Figure 2) and hence decrease the rate of convergence. Furthermore, the way recombination and mutation are performed in DE is different from those of GA this may also help DE in its better performance [15, 16].

#### 4 Circuit optimization

The optimization flowchart is shown in Figure 3. For a given topology MATLAB creates a NETLIST file with random initial values for the circuit parameters. A circuit simulator then analyses the input NETLIST file and creates an output file. MATLAB opens this file and analyzes the results, and then it changes the parameter values it had assumed in the first stage using one of the specified algorithms. This loop continues until a vector of objectives is satisfied [17].

Among the common circuit simulators, APLAC was selected to analyze the NETLIST file. This was mainly because of its ability to perform Harmonic Balance analysis.

In order to compare DE with GA, a double balanced Gilbert mixer was chosen. The needed requirements were: conversion gain >13dB,  $IIP_3 > 5\text{dBm}$ , Noise Figure <9dB, Power <10mW, LO-IF Isolation >50dB

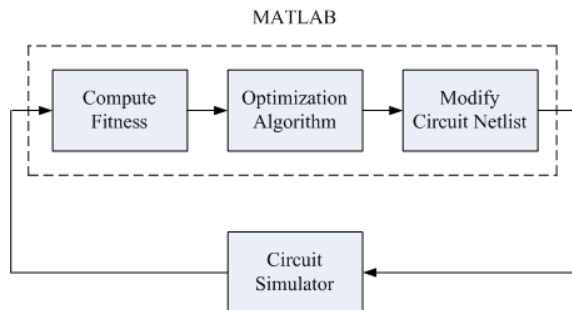


Figure 3: Optimization Procedure

Table 1: Circuit specifications

$V_{DD}$	Process	$F_{LO}$	$F_{RF}$	$R_{LOAD}$
1.8 v	0.18 $\mu\text{m}$	2.25 GHz	2.5 GHz	500 $\Omega$

The circuit specifications and requirements are shown in Table 1 and Table 2 accordingly.

Table 2: Problem requirements

Conversion Gain	$IIP_3$	Noise Figure	Power	LO-IF Isolation
> 13 dB	>5dBm	< 9 dB	<10mW	> 50 dB

Assuming that  $(error)_k$  is defined as desired values of Table 2 minus the actual values when the results are poorer than the desired value and zero if the actual value is better, fitness will be defined as:

$$fitness = \sum_{k=1}^5 (error)_k \quad (5)$$

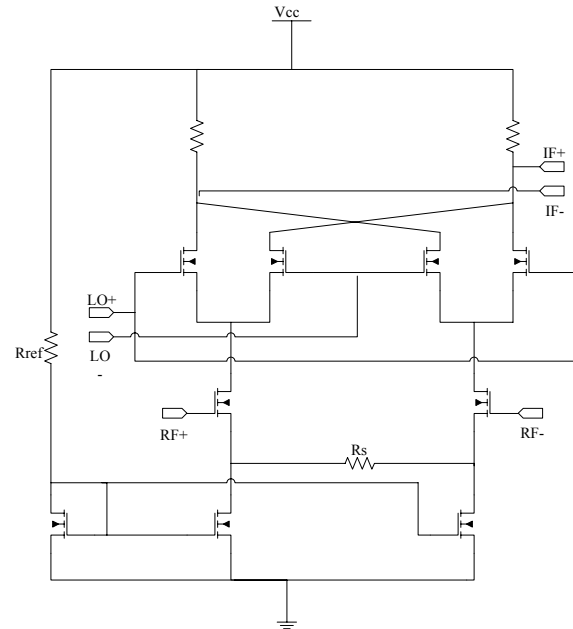


Figure 4: Circuit topology

Figure 4 shows the circuit topology to be optimized; the variables whose optimum values are to be found are the MOS transistors aspect ratios, biases plus the resistor values; however it is obvious that for the circuit to become symmetric some transistors should have identical aspect ratios. Therefore the circuit parameters were defined as follows:

- $W1$  = RF MOSFETs aspect ratio
- $W2$  = LO MOSFETs aspect ratio
- $W3$  = Current mirror MOSFETs aspect ratio

- $W_4$  = Current source MOSFET aspect ratio
- $V_{RF}$  = DC voltage of RF port
- $V_{LO}$  = DC voltage of LO port

The GA parameters and its bit allocation are shown in Table 3 and Table 4 accordingly.

Table 3: GA parameters

Population size	Generation	Crossover	Mutation	Elitism
80	50	1	0.05	0.025

Table 4: GA bit allocation

Variable	$V_{LO}$	$V_{RF}$	$W_1$	$W_2$	$W_3$	$W_4$	$R_S$	$R_{REF}$
Bit	6	6	11	11	11	11	7	8

DE parameters are shown in Table 5, the iteration and population size were chosen equal to those of GA to provide equal circumstances for both circuits

Table 5: DE parameters

Method	Population size	Iteration	$F$	$CR$
DE/rand/1	80	50	0.3	0.8

## 5 Simulation results

A comparison has been made between the results of GA and DE in Table 6.

Table 6: Optimization Results

Algorithm	Conversion Gain	IIP3	Noise Figure	Power	LO-IF Isolation
GA	12.47 dB	5.01 dBm	8.5 dB	15mW	60.18 dB
DE	13.33 dB	5.26 dBm	7.95dB	9mW	52.3 dB

Table 7 shows the optimum values obtained using DE method.

Table 7: Optimum values obtained using DE

$V_{LO}$	$V_{RF}$	$W_1$	$W_2$	$W_3$	$W_4$	$R_S$	$R_{REF}$
1.18	0.86	4.0E-5	5.2E-5	3.4E-6	3.28E-5	2.37	420.5

Figure 5 compares maximum fitness between GA and DE. As it could be seen DE's convergence time is much smaller and it converges much faster. One of the methods for showing the superiority of one algorithm to another is comparing their fitness

average per generation. Figure 6 compares fitness average between the two algorithms.

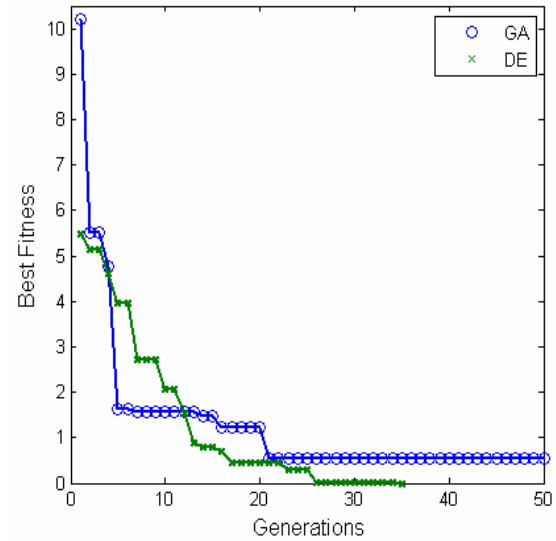


Figure 5: Best fitness per generation

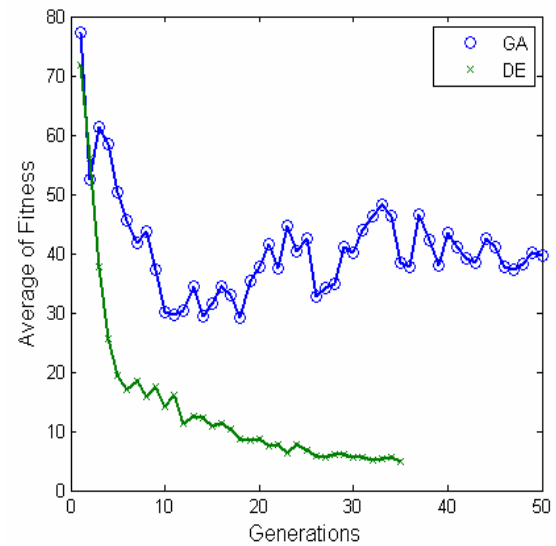


Figure 6: Fitness average per generation

## 6 Conclusion

In this paper a simulation-based optimization procedure was proposed for design of a Gilbert mixer; however this is a general method and can be used for any other types of analog circuit. If the circuit configuration is predetermined the software can optimize the device sizes in order to meet a vector of objectives. This method also needs a much shorter convergence time compared to Genetic Algorithms and therefore can help to reduce the cost and marketing time.

## 6 Acknowledgment

The Authors would like to thank professors: Khalil Mofinejad, Reza Lotfi, Sasan Naseh, and Hooman Nabovati greatly for their advices.

## References

- [1] W. Nye et al. "DELIGHT.SPICE: An optimization based System for the design of integrated circuits" *IEEE Transactions on Computer-Aided Design*, 1988.
- [2] Ricardo S. Zebulum, Marco A.C. Pacheco e Marley M.B.R. Vellasco, "Synthesis of CMOS Operational Amplifiers through Genetic Algorithms.", *Brazilian Symposium on Integrated Circuits*, 1998.
- [3] W. Kruiskamp and D. Leenaerts, "DARWIN: CMOS op amp synthesis by means of a genetic algorithm," in *32nd Annual Design Automation conference*, 1995.
- [4] M. G. R. Degrauwe et al ; "Towards an Analog System Design Environment," *IEEE Journal of Solid-State circuits*, 1989.
- [5] R. Harjani, R. A. Rutenbar, and L. R. Carley, "OASYS: A framework for analog circuit synthesis," in *IEEE Transactions on Computer-Aided Design*, 1989.
- [6] Maria Del Mar Hershenson, Stephen P. Boyd, Thomas H. Lee, "GPCAD: A Tool for CMOS Op-Amp Synthesis," in *IEEE Transactions on Computer-Aided Design*. 1998.
- [7] D. E., Goldenberg, "Genetic algorithms in search, optimization and machine learning," *Addison Wesley, Reading MA*, 1989.
- [8] M. Taherzadeh-Sani, R. Lotfi, H. Z. Hoseini, and O. Shoaie, "Design Optimization of Analog Integrated Circuits Using Simulation-Based Genetic Algorithm," in *proceedings of Signal Circuit and Systems Conference(SCS)*, Romania, pp. 73-76, 2003.
- [9] Kenneth V. Noren and John E. Ross, "Analog Circuit Design Using Genetic Algorithms", *Second Online Symposium for Electronics Engineers*, Summer 2001.
- [10] Horta, N.H.; M. Barros ; J. Silva; G. Neves ; "Enhanced Genetic Algorithm Kernel Applied to a Circuit-Level Optimization E-Design Environment ", *Proc. of the IEEE Conf. on Electronics, Circuits and System* , Sharjah , United Arab Emirates , December , 2003.
- [11] Karaboga D. and Okdem S. (2004) , "A Simple and Global Optimization Algorithm for Engineering Problems: Differential Evolution Algorithm", *Turk J Elec Engin*, VOL.12, NO.1 2004.
- [12] R. Storn .(1996), "On the Usage of Differential Evolution for Function Optimization", *NAFIPS* 1996, Berkeley, pp. 519 – 523.
- [13] R. Storn and K. Price (1995)., "Differential Evolution - a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces" , *Technical Report TR-95-012, ICSI*, March 1995.
- [14] R. Storn and K. Price .(1996), "Minimizing the real functions of the ICEC'96 contest by Differential Evolution", *IEEE Conference on Evolutionary Computation*, Nagoya, 1996, pp. 842 – 844.
- [15] Nagesh Kumar, K. Srinivasa Raju, B.Ashok, "Optimal Reservoir Operation for Irrigation of Multiple Crops using Genetic Algorithms", *Journal of Irrigation and Drainage Engineering, American Society of Civil Engineers*.
- [16] Vesterstrm, J. and R. Thomsen (2004). "A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems." *In Proceedings of the 2004 Congress on Evolutionary Computation*, vol. 2, pp. 1980-1987.
- [17] McEachern, L. (1999), "Constrained Circuit Optimization Via Library Table Genetic Algorithms", 0-7803-5471 -0/99 *IEEE*.