Optimizing an LNA Circuit by Combining Multi-Objective Genetic Algorithm and HSPICE

Hojat Jafari[1], Hossein Ghayoumi Zadeh[2], Majid Baghaei-Nejad[3], Javad Haddadnia[4]

[1.] M.Sc Student In Electrical Engineering, Electrical Engineering Department , Hakim Sabzevari University, Khorasan Razavi, Iran
[2.] Ph.D Student in Biomedical Engineering, Biomedical Engineering Department, Hakim Sabzevari University, Khorasan Razavi,Iran
[3.] Assistant Professor OF Electrical Engineering, Electrical Engineering Department ,Hakim Sabzevari University, Khorasan Razavi, Iran
[4.] Associate Professor of Biomedical Engineering, Biomedical Engineering Department ,Hakim Sabzevari University , Khorasan Razavi, Iran
Corresponding author: h.ghayoumizadeh@gmail.com, Tel +98-910-2911161

Abstract: In this paper, a multi-objective genetic algorithm is provided to help designing electronic circuits. The used optimization algorithm is an extended multi-objective genetic algorithm based on the distributed Pareto frontier, which optimizes circuit parameters in order to achieve low noise, low power and circuit stability. The circuit studied in this paper is a LNA circuit. The genetic algorithm is implemented in MATLAB and circuit simulations are performed using HSPICE and .18 um CMOS technology so that with the two linked software applications, the optimization process is begun. An important feature of this paper is the use of accurate models for the elements in simulation and obtaining results which are very close to reality. The performed simulations indicate that the proposed algorithm has better convergence and diversity in determining optimum solutions compared to multi-objective genetic algorithm NSGA-2. The proposed algorithm converges to the near optimum and optimum solutions with higher efficiency and speed and also enjoys appropriate diversity. Based on the results obtained, GA is shown to be capable in assisting circuit designs, solving the crucial circuit parameters for achieving the required specifications, preference and constraints .

[Hojat Jafari, Hossein Ghayoumi Zadeh, Majid Baghaei-Nejad, Javad Haddadnia. **Optimizing an LNA Circuit by Combining Multi-Objective Genetic Algorithm and HSPICE.** *Life Sci J* 2013;10(1):853-859]. (ISSN: 1097-8135). http://www.lifesciencesite.com.134

Keywords: LNA; Multi-Objective Genetic Algorithm; optimization.

1. Introduction

Even we are in the digital age and digital circuits directly benefit from advances in IC technologies, RF circuits do not as much. This issue is exacerbated by this fact that RF circuits often require external components - for example, inductors – where it is difficult bringing of them into the chip even in modern IC processes. In fact, computer aided analysis and synthesis tools for RF ICs are still in their infancy which it is forcing the designers to rely on experience, intuition, or inefficient simulation techniques to predict the performance. For example, nonlinearity, time variance, and noise in RF circuits usually require studying the spectrum of signals, but the standard ac analysis available in SPICE uses only linear, time invariant models [1].Therefore, developing reliable automatic tools in RF IC design seems very attractive. One solution to this problem is employing Evolutionary Computing and in particular Genetic Algorithms (GA). Genetic Algorithm is a global search algorithm, which it models the process of the natural evolution in order to optimize the parameters of a problem. Genetic algorithm utilizes a non-gradient-based random search and is used in the optimization of complex systems [2]. In this paper, an example for a LNA which was described in reference [3] is presented in 0.18μm process for evaluation of non-dominated sorting genetic algorithm (NSGA-II) as a method of multi objective genetic algorithm optimization. Simulation results confirm efficiency of the GA for determining of devices sizes and optimization of a RF circuit. In circuit design, power gain, power loss, noise figure, current, and circuit stability are the crucial specifications to be achieved. Conventionally, with the availability of circuit simulation system, many circuit design parameter tuning is carried out in a trial-and-error manner. Conventionally, with the availability of circuit simulation system, many circuit design parameter tuning is carried out in a trial-and-error manner. Although this is the most straight forward and simple approach, it is time consuming and not much convincing. This paper proposed the use of GA as a systematic approach to assist circuit simulation system for searching the best parameter setting in order to fulfil the circuit design specifications or objectives. Genetic algorithms (GAs) have been widely known for its robustness in solving tough and miscellaneous problems based on function optimization through evolutionary computation [4-6].GA is a stochastic optimization approach that mimics the biological evolution of human genetics. The GA search is guided by the probability of the survival of the chromosomes with better fitness. The robustness of GA in circuit design has been shown by a number of researchers. For instance, [7] applied GA for the structural cell-based VLSI circuit design and claimed the satisfaction in multiple output circuit criteria. Besides that, it is also commented in [8] that GA is usable for designing RF circuits. In addition, [9] which employed GA in LNA further convince the capability of GA in synthesizing the analogue circuit. In this research, the Agilent Advanced Design System (ADS) in combination with the GA optimizer is used to simulate the circuit performance and optimize the design parameters for a 5-GHz MMIC LNA and a 15.12-GHz SPDT switch respectively. In this paper, we link HSPICE and MATLAB and provide a multi-objective genetic algorithm in order to find the optimum parameters. The results are then compared to the results provided in other papers in this field.

2. Multi-objective optimization

The existence of different objective functions in a problem leads in fact to a set of, rather than one, optimal solution(s), which are called Pareto optimal solutions. If there is no constraint, none of these solutions will dominate the others. Because of this, the user looks for the largest possible Pareto set. Classic optimization methods suggest that in order to solve a multi-objective problem, the problem must be converted into a single objective problem with a specific Pareto solution each time. When such method is used to find multiple solutions, it must be run many times hoping that each time a different solution will be found [10].In the last decade, several multi-objective genetic algorithms were proposed [10-14]. The main reason for using this method is its ability to find multiple optimal solutions in a single run. NSGA2 is one of the fastest and latest multi-objective genetic algorithms based on non-dominated sorting that has an appropriate sorting time complexity of o(mn2), where m is the number of objective functions and n is the size of population. It also has a selection operator which creates a mating pool by integrating parent population and children population and then selects the best n solutions according to fitness and extension values. Stochastic algorithms such as genetic algorithm do not keep in the memory the visited and examined points. However, the Tabu search [15], as an exception, uses the most recent searches as a guide for the next step. But even in the Tabu search, all the examined points already visited are not kept in the memory. Therefore, in the existing methods, the revisit of points is inevitable [16].Re-evaluation of the fitness function for an element such as s, which is already evaluated, is called "revisit". Since in many real world applications, especially those requiring multi-objective optimization, the evaluation of the fitness function is one of the delicate computation processes in genetic algorithm, it is clear that revisits would waste computational resources. As examples of real world applications with high cost and time consuming fitness function estimation, 3-dimentional object recording in machine vision [17] and HVAC engineering can be mentioned [18].It is in the rules of NFL2 [16, 19] that if the problem is distributed uniformly, all the algorithms (stochastic or deterministic) will have the same average performance. An algorithm p with revisits will have the same sequence of points as an algorithm m with no revisits once its repeated points are left out, so since m has the same performance as p, it can be deduced that m is better than p. Therefore, it is useful to eliminate revisits at all times.

In this paper, we provide a multi-objective genetic algorithm that eliminates revisits. In order to eliminate revisits entirely, we use a BSP tree as the archive. The proposed algorithm converges to near optimal and optimal solutions with higher speed and efficiency and also has appropriate diversity.

3. Fast multi-objective genetic algorithm with revisit elimination

First, we introduce some of the concepts in multi-objective optimization.

Multi-objective optimization problem: an n-dimensional decision vector, $X = \{x_1, \dots, x_n\}$, is given in the solution space X. The problem is to find a vector X*, which optimizes m given objective functions , $F(x^*) = \{f_1(x^*), \dots, f_m(x^*)\}$. Usually the solution space X is constrained by a set of constraints $g_j(x^*) = p_j, j = 1, 2, \dots, k$.

Dominance: It is said that a vector u = (u1,..., un) is dominant over another vector v = (v1, …, v_n), u < v, if and only if u is partially less than v, i.e. ui < vi for each i = 1, …, n.

Pareto optimal solution: A solution Xu is Pareto optimal if and only if there is no Xv that F(Xv) = v = (v1, …, v_n) is dominant over $F(X_u)$ = u = (u_1, …, u_n).

Pareto optimal set and frontier: a non-dominated set in the solution space X is called the Pareto optimal set and is denoted by Xp, such that (fig.1)

$$X_p = \{x | x \text{ is non} - \text{dominated و} X_p\ X\} \quad (1)$$

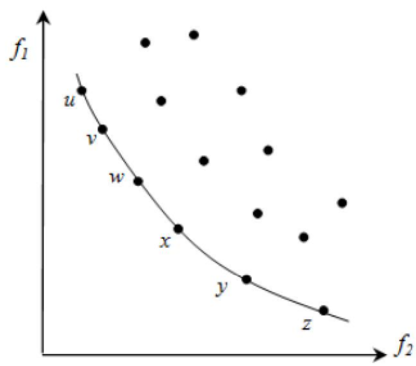

Figure 1- an illustration of the set of frontier points

Revisit: Re-evaluation of fitness function for the ith element in the sequence (x1, x2, …) is a revisit if there is a Xj that xi = xj, j < i.

Crowding distance: in order to have an estimate of the solution density around a solution in the population, we calculate the mean distance of two points in both sides of a specified point for different objective functions. Algorithm 1 (Table.1) shows this procedure.

f_y^{max} And f_y^{min} are respectively the maximum and minimum values that an objective function y can have. T[i].y is also the value of the objective function y for the ith member of the set T.

Table .1. assigning crowding distance to a set T.

1. m:= the number of elements of the set T
2. for each i: $T[i]_{distance} := 0$
3. for each objective function do
   3.1. sort T based on the values of function y (ascending)
   3.2. $T[m]_{\text{distance}} \coloneqq \infty$ and $T[1]_{\text{distance}} \coloneqq \infty$
   3.3. for i = 2 to m – 1 do:
   3.3.1. $T[i]_{\text{distance}} \coloneqq T[i]_{\text{distance}} + \frac{T[i+1].y - T[i-1].y}{f_y^{max} - f_y^{min}}$

Population comparator operator: between two population members, this operator selects the member with the smaller non-dominated rank (algorithm 3) and if both have the same rank, the member with the larger crowding distance is selected. This operator is used in the genetic algorithm.

4. Eliminating Solutions Revisit

In the proposed algorithm, first an initial population P0 is generated randomly. In order to avoid revisits of the elements, a BSP tree is used as an archive for keeping the members of the generated population. In fact, BSP tree is used as an appropriate data structure for performing fast queries - a query to check if the generated member is already visited or not. The algorithm that is used to generate such archive members is provided in algorithm 2(Table 2). In this algorithm, the whole search space is examined according to a specified resolution and the whole space is set as the space of root node, initially. Next, by generating the population members and children nodes, search space is partitioned so that the union of the children's subspace is equal to the space of the parent. Whenever a population member is generated and before it is inserted into the tree , an inquiry is performed to determine if the member is a duplicate or not. If it is a duplicate, in order to generate a non-duplicate member, the mutation operator is applied in the subspace of the node that is returned as the search result (adaptive mutation).

Table. 2. Elimination of Revisits

1. create root node
2. RF := 0 (RF is the revisit flag)
3. Flag(root):="open"
4. Create new element z using genetic algorithm
5. c_node:="root node"(C_node is the current node)
6. If flag(c_node):="open" then:
   6.1. if C_node has two children such as a and b then:
      6.1.1. if (z == a) or (z == b) then: RF := 1
      6.1.2. j := the dimension corresponding to the largest distance between a and b
      6.1.3. if z is nearer to a in dimension j then: C_node := a
      6.1.4. else: C_node := b
      6.1.5.go to step 6
   6.2. else (if flag(c_node)=="close"):
      6.2.1. if (RF == 0) then:
         6.2.1.1. insert z as a child node to C_node
         6.2.1.2. if the subspace of child node is not singleton then:
            6.2.1.2.1. flag(child_node):=open"
         6.2.1.3. else: flag(child_node):="close
         6.2.1.4. end
      6.2.2. else (if RF ≠ 0):
         6.2.2.1. use mutation operator to create z randomly in the not-visited subspace and add it to the child node
         6.2.2.2. if the subspace of child node is singleton then:
            6.2.2.2.1. Flag(child_node):"close"
         6.2.2.3. else: Flag(child_node):="open"
         6.2.2.4. end
7. else (Flag(c_node)=="close"):

7.1. C_node := "parent node"
7.2. If two children of the current node are "close" then:
7.2.1. Flag(child_node):="close"
7.2.2. prune the sub-tree under the current node
7.3. else: C_node := the open child
7.4. go to step 6

While searching, if we reach a node which is singleton, the algorithm returns to a higher level (parent node) and selects the other child direction. If the other child node is also singleton, then the sup-tree is pruned from the tree. That is because all the sub-tree values are visited and its pruning helps prevent the excessive growth of the tree. A node is singleton if it is not possible to select another member in its subspace (according to the specified resolution). (Table 3)

Table 3: non-dominated sorting algorithm for the population P

1. for each p϶P do:
1.1. S_P: =Ø; n_P: =0;
1.2. For each q϶P do:
1.2.1. If (p<q) then: Sp:=SpU{q}
1.2.2. Else if(q<p) then: np:=np+1
1.3. if (np == 0) then: F1:=F1U{p} Prank:=i+1;
2. i:=1
3. while (F_i≠Ø) do:
3.1. H:=Ø
3.2. For each p϶F_i do:
3.2.1. For each q϶S_p do:
3.2.1.1. np:=np-1
3.2.1.2. if (np == 0) then: H=H U {q} qrank:=i+1;
3.3. i:=i+1
3.4. Fi:=H

5. Multi-objective genetic algorithm without revisits

After examining the initial population P0 and eliminating revisits, the population is sorted by non-dominated sorting (algorithm 3). A fitness value (rank) is assigned to each one of the population members, which is equal to its non-dominated level - rank 1 to the best level, rank 2 to the second best level and so on. Assume that the goal of optimization is to minimize the objective functions. The new generation, H0 with size N, is created by using selection, crossover, revisit elimination and adaptive mutation operators on the initial population P0. Therefore, elitism is applied by comparing current generation and the best non-dominated members of the previous generation. Next, the created new generation and parent generation are merged and a new generation G0 with size 2N is formed. Then, G0 is sorted using the non-dominated sorting method and is placed in F as a number of sets, so that the members of each set has the same rank. The rest of the algorithm for the next generations are slightly more different and is provided in algorithm 4. (Table 4)

Table. 4. Multi-objective genetic algorithm with revisit elimination

1. Gi := HiU Pi
2. Sort Gi using non-dominated sorting and place it in F= {F1, F2,…}
3. k:1;p_{i+1}=Ø
4. While |Pi+1 U Fk|≤N do:
4.1. Compute crowding distance in Fk
4.2. Pi+1:=Pi+1U Fk
4.3. k:=k+1
5. if |Pi+1|<N then:
5.1. sort Fk using the population comparator operator
5.2. X:=N-|Pi+1|
5.3. Pi+1:=Pi+1U Fk[1:x]
6. Use cross over, revisit elimination and adaptive mutation operators to Generate a population based on Pi+1
7. Place the new population both in BSP tree and Hi+1
8. i:= i + 1 and go to step 1

The members of non-dominated set F1, the best members of the merged populations, are selected to create the next generation. If the size of F1 is less than N, the specified size of a generation, then all the members of F1 are selected to go to the next generation (pi+1). The rest of the new generation members are selected from the subsequent non-dominated sets according to their rank, i.e. members of F2 would be the next choice, and after that F3 and so on. This procedure will continue until the N members of new generation are selected, or no more non-dominated set, Fk, could be transferred to the new generation (i.e. |Pi+1|<N and |Pi+1U Fk|>N). If the second case occurs, i.e. the new generation of its members is not full, then the next non-dominated set is selected and its members are sorted using population comparator operator. Then, the remaining required members of the new generation pi+1 are selected by applying the selection operator. After creating the new generation pi+1, selection, cross over, revisit elimination, and adaptive mutation operators are used to create the generation Hi+1 (figure 2).

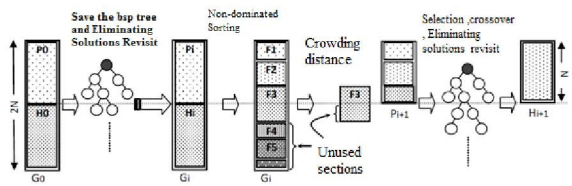

Figure 2: Multi-objective genetic algorithm procedure

6. Fitness Function Calculation

First step of simulation of mentioned algorithm using MATLAB and HSPICE RF, the net list of each parameters vector is created and HSPICE RF is called. Then, the output file of HSPICE RF is used for object evaluation. In fact, a LNA using GA as a search algorithm and HSPICE RF tool as the fitness evaluator is designed. (fig.3)

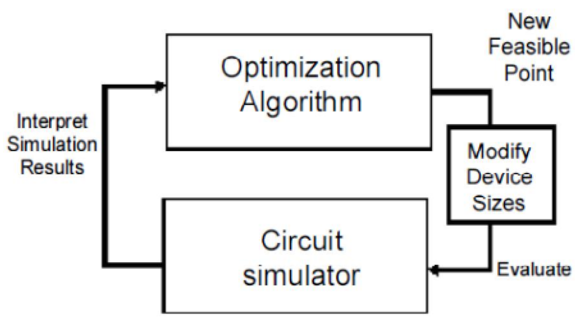

Fig. 3. Optimization Procedure [9]

7. The evaluated LNA circuit

Low noise amplifiers (LNA) are one of the key building blocks for RF receivers. They play a critical role for determining the overall system noise figure (NF) of the receiver [20]. The main function of an LNA is to provide sufficient gain to overcome the noise of subsequent stages (e.g. mixers) while adding as little noise as possible. For all kinds of receiver's architecture, LNA is the first block to interface the weak RF signal coming from the antenna and duplexer. The noise performance and gain of LNA have a significant impact on the overall system noise performance [21]. In this work, a LNA which described in reference (Fig. 4) is designed by using NSGA-II as a method of multi objective genetic algorithm optimization and HSPICE RF as evaluator tool [3, 22]. The reason we choose multi objective method for optimization is that RF circuits usually have several parameters. They are against together and designers need to trade off between these objectives such as gain, BW (band width), phase margin, power, and noise figure (NF) and so on. The reason for choosing of NSGA-II is low complexity and high computation time of its algorithm for optimization. Also, we chose a LNA which is described in reference [3], because was designed as one of the best LNAs with the best parameters till 2010.

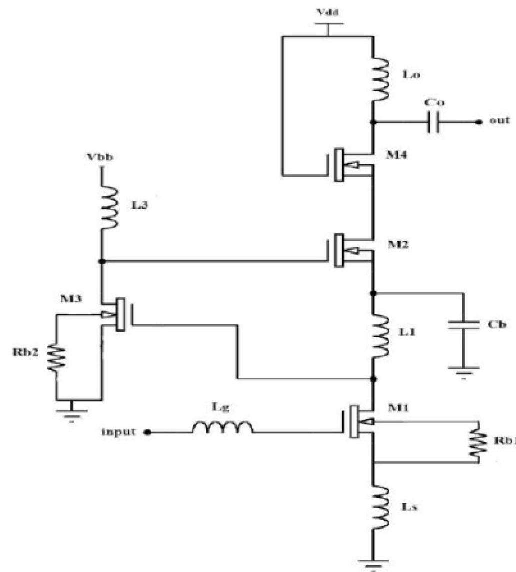

Fig. 4. Schematic and small signal equivalent of LNA [3].

8. Results

Figures illustrate that executed algorithm is converged to the optimized point after 100 generations with initial population which is equal to 50. Follow execution of GA program, the performance characteristics were obtained which were better than the desired objects in reference [1].The set of frontier points of power values in terms of noise figure are shown in figure 5.

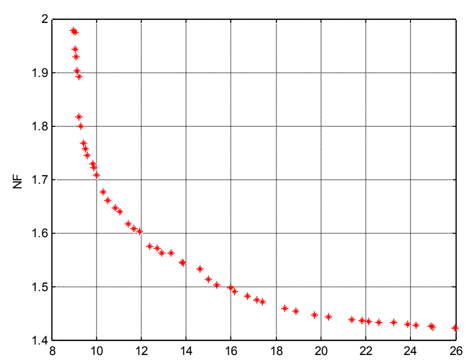

Figure 5. the set of frontier points of power in terms of noise figure

Figure 6 illustrates the frontier points of three parameters, s11, s22 and s21 in terms of one another.

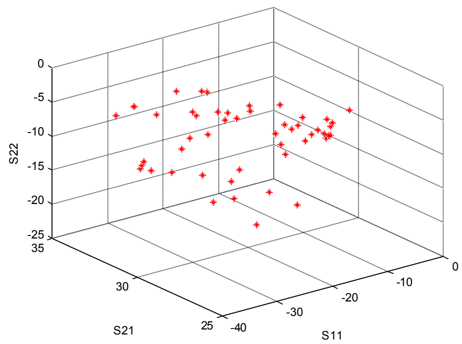

Figure 6: the set of frontier points of three parameters s11, s22 and s21in terms of one another

The circuit size vector and the performance characteristics and optimized value are shown in Table 5.

TABLE 5. Optimum Values Obtained Using NSGA- II

Parameter	Value1	Value2	Value3	Unit
(nr)m1	26.14	26.30	26.47	-
(nr)m2	29.78	29.78	29.73	-
(nr)m3	7.82	8	7.84	-
(nr)m4	19.48	18.80	19.57	-
(nr)L1	16.85	17.19	17.06	-
(nr)Ls	6.93	6.88	6.92	-
(nr)L3	13.62	13.07	13.17	-
(nr)Lo	19.7	18.96	18.92	-
(nr)Lg	1.41	1.41	1.4	-
(nr)L1	3.56	3.56	3.55	µm
(nr)Ls	3.1	3.09	3.08	µm
(nr)L3	5.57	5.59	5.57	µm
(nr)Lo	2.57	2.8	2.57	µm
(nr)Lg	5.03	4.96	5.03	µm
Cb	12.87	12.72	12.87	pf
Co	292.82	295.94	292.81	ff
Vbb	1.32	1.32	1.32	v
Vdd	0.612	0.612	0.612	v
power	12.62	12.78	12.71	µw
S21	27.97	29.94	29.41	dB
S11	-9.161	-9.73	-9.44	dB
S22	-2.86	-4	-3.77	dB
NF	1.67	1.656	1.66	-

Where (nr) ms is the number of finger of each transistor, (nr) Ls is the number of turns in the coil of each inductor and (rad) Ls is the Radius of coil of each inductor. Values 1, 2, 3 are diversity of results, on Pareto front while all of them are better than result in references [23, 24]. All of these values are usable for your work. The simulation results confirm the efficiency of the GA for determining the devices sizes in LNA. Also a comparison has been made between the results of proposed algorithm in this work and references [23, 25] in Table 6.

Table 6: Comparison with previous work

Ref	This work	Ref.[24]	Ref.[23]
F0(GHz)	5.7	5.7	5.7
NF(dB)	1.67	1.68	1.85
S21(dB)	27.97	39.36	32.5
S11(dB)	-9.161	-26.64	-14
S22(dB)	-2.86	-27.36	-17.5
power	12.62	14.62	16
Vdd(v)	0.61	0.61	1.8

9. Conclusion

In this paper Genetic Algorithm and simulation based optimization were combined to produce an accurate tool for LNA designing. Also we show that multi objective algorithms like NSGA-II are some of the best methods for designing of this kind of RF circuits where they can be even used for designing of other characteristics as distortion behaviour and so on. The proposed method which it used for designing of this circuit is a general method and it is usable for any other types of RF circuits.The run time of the algorithm depends on the number of HSPICE runs. As it is observed, solutions obtained by this method have higher quality and diversity and also in terms of the time required for design, it is much faster than manual design. That is because in manual calculation method, achieving appropriate results requires a lot of trial and errors which is very time-consuming. That is due to the fact that in each step the designer changes only one of the input parameters and after that repeats the simulations and evaluates the results. On the other hand, in manual method, the designer cannot be sure that the obtained solution is the best, while in this method; the chances are high that we get to the best solutions (global optimum solutions). This method is applicable to any other RF circuit such as LNA, mixer, oscillator, etc. and is preferred to current design techniques in terms of required time and accuracy of the results. Some of the other features of this method include taking the parasitic characteristics of elements into consideration and paying attention to their layout in circuit design.

References

1. B. Razavi, “RF Microelectronics”, Prentics-Hall,Upper Saddle River , 1997.
2. M. Taherzadeh, R. Lotfi, H. Zare and O. Shoaei, "Design optimization of analog integrated circuits using simulation-based genetic algorithm", Proc. IEEE International Symposium on signals and circuits.1: 73-76, July 2003 .

3. E. Kargarn, H. Khosrowjerdi, K.Ghaffarzadegan and H. Nabovati, "A 5.7 GHz low noise figure ultra high gain CMOS LNA with inter stage technique", IEICE Electronics Express. 7(23):1686-1693 .
4. D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Kluwer Academic Publishers, Boston, MA,1989.
5. M. Gen, and R. Cheng, Genetic algorithms and engineering optimization. John Wiley & Sons Inc., New York, 2000.
6. D. Fogel, Evolutionary Computation: Toward a new philosophy of machine intelligence. IEEE Press, Piscataway, 1995.
7. M.B. Fallahpour, K.D. Hemmati, A. Pourmohammad, and A.Golmakani, "Multi objective optimization of a LNA using genetic algorithm based on NSGA-II," Proceedings of the International Conference on Electrical Engineering and Informatics, 2011.
8. J. Yanfeng, J. Jiaxin, Z. Xiaobo, and Y. Bing, "Automated analog circuit design using genetic algorithms," The 3rd International Conference on Ant counterfeiting, Security, and Identification in Communication, ASID, 2009.
9. M. Sumathi and S. Malarvizhi, "Design and performance analysis of 5GHz CMOS RF front-end circuits for IEEE 802.11 a application," European Journal of Scientific Research vol. 54 (3), pp. 418-426, 2011.
10. Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T.Meyarivan, "A Fast and Elitist Multi objective Genetic Algorithm: NSGA-II", IEEE transactions on evolutionary computation, vol. 6, no. 2, April 2002.
11. Mitra, K., Deb, K., and Gupta, S. K. (1998). Multi objective dynamic optimization of an industrial Nylon 6 semibatch reactor using genetic algorithms. Journal of Applied Polymer Science, 69(1), 69–87.
12. Horn, J. and Nafploitis, N., and Goldberg, D. E. (1994) A niched Pareto genetic algorithm for multi-objective optimization, Proceedings of the First IEEE Conference on Evolutionary Computation, pages 82–87.
13. Srinivas, N. and Deb, K. (1995) Multi-Objective function optimization using non-dominated sorting genetic algorithms, Evolutionary Computation, 2(3):221–248.
14. Zitzler, E. and Thiele, L. 'Multiobjective optimization using evolutionary algorithms—A comparative case study". In Eiben,A. E., B¨ack, T., Schoenauer, M., and Schwefel, H.-P., editors, Parallel Problem Solving from Nature, V, pages 292–301.
15. F. Glover and M. Laguna, Tabu search. Norwell, MA: Kluwer, 1997.
16. Shiu Yin Yuen, Chi Kin Chow, "A Genetic Algorithm That Adaptively Mutates and Never Revisits", IEEE transactions on evolutionary computation, vol. 13(2): april 2009.
17. C. K. Chow, H. T. Tsui, and T. Lee, "Surface registration using a dynamic genetic algorithm," Pattern Recognition, 37(1):105–117, 2004.
18. K. F. Fong, V. I. Hanby, and T. T. Chow, "HVAC system optimization for energy management by evolutionary programming," Energy and Buildings, 38(3):220– 231, 2006.
19. D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," IEEE Trans. Evol. Comput. vol. 1, no. 1, pp. 67–82, Apr.1997.
20. G. Tulunay, S. Balkır, "A Compact Optimization Methodology For Single-Ended LNA", IEEE, ISCAS, pp273-276, 2004.
21. D.Feng, B .Shi, "Comprehensive Analysis and Optimization of CMOS LNA Noise Performance", IEEE. ASP.DAC.,pp . 1204-1207, 2005.
22. Mojtaba Behzad Fallahpour, Kamran Delfan Hemmati, Ali Pourmohammad and Abbas Golmakani."Multi Objective Optimization of a LNA Using Genetic Algorithm Based on NSGA-II". 2011 International Conference on Electrical Engineering and Informatics.
23. E. Kargarn, H. Khosrowjerdi, K.Ghaffarzadegan and H. Nabovati, "A 5.7 GHz low noise figure ultra high gain CMOS LNA with inter stage technique", IEICE Electronics Express, 7(23):1686-1693.
24. Mojtaba Behzad Fallahpour, Kamran Delfan Hemmati, Ali Pourmohammad. "Optimization of a LNA Using Genetic Algorithm". Electrical and Electronic Engineering 2012, 2(2): 38-42.
25. O.R.Seryasat, M.Aliyari shoorehdeli, M. Ghane, J. Haddadnia, M. Zeinali" Intelligent Fault Detection of Ball bearing Using FFT, STFT Energy Entropy and RMS" Life science journal, 9 (3) (2012) 1781-1786.

12/20/2012