Simulated Annealing Algorithm with Multi-Molecule: an Approach to Analog Synthesis *

H.Z. Yang, C.Z. Fan, H. Wang, R.S. Liu

Dept. of Electronic Engineering,
Tsinghua Univ., Beijing, P.R. China, 100084

Abstract

The research presented in this paper is concerned with the design automation of analog integrated circuits, simply called analog synthesis. A new algorithm, namely a simulated annealing algorithm with multi-molecule (SAMM), is proposed to solve analog synthesis problems. Besides inheriting the global convergence of the traditional simulated annealing algorithm (SA), SAMM can avoid the excessively time-consuming final iterations of SA, which makes it more efficient and suitable for analog synthesis. Several analog synthesis examples are also given in this paper to demonstrate the efficiency and validity of SAMM.

1. Introduction

Economic and other factors favor the incorporation of analog circuits (interface or primary signal processing circuits) and digital circuits onto the same silicon die. The design of application specific integrated circuits (ASIC) moves therefore steadily towards the integration of complete systems into a single chip. However, this trend is hindered by a bottleneck in the design process for these chips: the continuing lack of mature analog EDA tools.

The development of analog synthesis tools has lagged far behind that of digital tools, partly because analog circuit design has often been considered as a creative and highly unstructured synthesis problem. Experts rely on years of experience to guide the selection of circuit topologies and parameters. Generally speaking, analog synthesis is to solve the following mixed-integer nonlinear programming (MINLP) problem,

$$\begin{cases} \min\limits_{\vec{X},\vec{Y},\vec{V}} \ \mathit{Area}(\vec{Y},\vec{X}) \big/ \mathit{Yield}(\vec{Y},\vec{X},\vec{V},\vec{\sigma}^{\,2}_{\vec{X}}) \\ s.t. \quad \vec{P}(\vec{Y},\vec{X},\vec{V},\vec{\sigma}_{\vec{X}}) \geq \vec{P}_E \end{cases} \tag{1}$$

Where, binary vector $\vec{Y}$ represents the circuit topology . In another word, if y_i (the *i*-th component of vector $\vec{Y}$) is one, then block B_i is included in the circuit, and B_i is taken away while y_i is zero[1,2]. Continuous vector $\vec{V}$ denotes the bias voltages, $\vec{X}$ denotes device sizes and $\vec{\sigma}^{\,2}_{\vec{X}}$ is the variance of $\vec{X}$. Function $\mathit{Yield}(\bullet)$ and $\mathit{Area}(\bullet)$ represent the yield and the area of the circuit respectively. Function $\vec{P}(\bullet)$ is the performance of the circuits, which is composed of voltage gain, power, slew rate, band width, etc. Vector $\vec{P}_E$, the expected circuit performance, expresses the designer's desire.

The foundation of (1) is very simple[1-3]. The kernel aspect of analog synthesis lies in how to get the global minimum of the MINLP problem (1). There are some embryonic systems for analog synthesis[4-8], which can be classified into ad hoc rule-based[4,5], compiler-based[6], and symbolic[7,8] methods. All of them include two steps: topology selection and parameter selection (also called device sizing). In another word, these analog synthesis systems determine $\vec{Y}$ at first, then $\vec{X}$ and $\vec{V}$.

* Sponsored by China Ph.D. Foundation

ED&TC '96
0-89791-821/96 $5.00 © 1996 IEEE

P.C. Maulik[1,2] employ **NPSOL**[9] and **branch and bound algorithm**[10] to do topology selection and device sizing simultaneously, however it can not realize a real simultaneous topology selection and device sizing[3]. The Maulik's method also falls short in finding a global minimum of the MINLP problem (1). In this paper, we will introduce a simulated annealing algorithm with multi-molecule (SAMM), which is superior to the traditional simulated annealing algorithm (SA)[11] and the Maulick's method.

2. SAMM: Simulated Annealing Algorithm with Multi-Molecule

Some of the VLSI design problems can be regarded as a minimization problem in multidimensional continuous space with an objective function that has plural local minima. Well-established minimization procedures for convex functions, for example, the Levenberg-Marquardt method[12], can be easily trapped in one of the local minima and thus can not find a global minimum. Recently a method called simulated diffusion (SD) has been proposed[13,14] to find the global minimum of a multiminimal function on continuous space. Although it has been demonstrated theoretically that under certain conditions SD will find the global minimum with a probability of unity[15], little is known about the practical aspects of the SD as an optimization procedure. The SD is very slow[13] although it can find a global minimum, and can not be directly applied to this field because the MINLP problem (1) includes binary vector $\vec{Y}$.

As you know, which topology is optimal or feasible will depend on the device sizes obtained from the topology, and this in turn will depend on the specification. So, it is very hard to draw conclusions about a particular topology without having determined the sizes of the devices of that topology. In the worst case, two-step mode analog synthesizer might have to size all the topologies to determine which is the most suitable one for a particular set of specifications. Therefore, in order to get the global minimum of the MINLP problem (1), determining $\vec{Y}, \vec{X}$ and $\vec{V}$ simultaneously is very important. Our SAMM has four principal components as SA does: the **representation** of the synthesis problem manipulated by the annealer, the **moves** used to transform one circuit configuration (with device sizes) into another, the **cost function** or **energy function** that evaluates the quality of each visited circuit configuration, and the **control mechanisms** that direct the overall cooling process.

The problem **representation** is straightforward. The annealer manipulates the values of the independent variables composed by $\vec{Y}, \vec{X}$ and $\vec{V}$. We employ vector $\vec{Z} = \left(\vec{Y}^T, \vec{X}^T, \vec{V}^T\right)^T$ to denote them. At first, SAMM transfroms $\vec{X}$ and $\vec{V}$ into discrete variables. The device sizes $\vec{X}$ can be reasonably regarded as discrete quantities. The underlying limitations comes from either lithography or the need to restrict designs to a known set of carefully qualified device sizes for which accurate models are available. For example, device sizes are discretized in $0.5\mu\, m$ if the synthesized circuit will be manufactured in $0.5\mu\, m$ CMOS process. Each component of $\vec{V}$ is further represented by 7 current or voltage sources[16], which will be discretized in $1A$ or $1V$, $0.1A$ or $0.1V$, $10mA$ or $10mV$, $1mA$ or $1mV$, $0.1mA$ or $0.1mV$, $10\,\mu A$ or $10\,\mu V$, and $1\,\mu A$ or $1\,\mu V$ respectively. In SA, the annealer manipulates only one vector $\vec{Z}$ at annealing temperature T. But the annealer in SAMM manipulates M (> 1) vectors: $\vec{Z}^{(T,j)}$, $j = 1, 2, \cdots, M$, each represents "a molecule of a real system". In each perturbation experiment, only one of them is randomly selected to play Brownian movement.

Obviously, a more interesting problem is the design of the **move-set**. Since all the components of $\vec{Z}$ are discrete variables, there is always a smallest allowable move, an atomic perturbation, and the choice is what larger moves should be included in the move-set for efficiency. Suppose $\Delta\vec{Z}$ is the movement randomly generated at annealing temperature T, and the vector or molecule $\vec{Z}^{(T,k)}$, $1 \le k \le M$, is selected to play the role of Brownian movement, then the following iteration will transform one circuit configuration into another,

$$\vec{Z}^{(T,k)} \Leftarrow \vec{Z}^{(T,k)} + \lambda_0 \sqrt{T/T_0} \cdot \Delta\vec{Z} \qquad (2)$$

Where, T_0 and λ_0 are the initial annealing temperature and the initial iteration step respectively. The determination of T_0 and λ_0 in SAMM is the same as that in SA.

Simultaneous topology selection and device sizing is really achieved because the movement of $\vec{Y}, \vec{X}$ and $\vec{V}$ is generated simultaneously. It makes SAMM better than Maulik's method and those two-step synthesis systems.

The heart of the annealer is the energy function used to evaluate each visited circuit configuration $\vec{Z}$. The **energy function** in SAMM, the same as that in SA, has the form:

$$E(\vec{Z}) = cost(\vec{Z}) \equiv \underbrace{w_o \sum_{\text{device } d} \delta_d W_d L_d}_{\text{objective}} + \underbrace{\sum_i w_{p,i}(P_i - P_{E,i})}_{\text{penalty terms}} \quad (3)$$

Where, P_i and $P_{E,i}$ are the *i*-th component of vector $\vec{P}(\vec{Y}, \vec{X}, \vec{V}, \vec{\sigma}_{\vec{X}})$ and $\vec{P}_E$ respectively. w_o and $w_{p,i}$ are weights chosen appropriately. Obviously, w_o depends on the yield. W_d and L_d are the channel width and length of the *d*-th MOSFET respectively. δ_d is set to unity if and only if the *d*-th MOSFET is included in the circuit according to vector $\vec{Y}$, otherwise, δ_d is set to zero.

The **control mechanisms** in SAMM differ from that in SA. When annealing temperature T is above the critical temperature T_C, SAMM preforms in the same mode as SA. In this period, the mean energy of the M molecule determines the cooling schedule. Suppose the mean energy at temperature T is $\overline{E} = \frac{1}{M}\sum_{j=1}^{M} E(\vec{Z}^{(T,j)})$, and $\overline{E}$ changes to $\overline{E}'$ after m random moves, then SAMM reduces T to εT. The cooling parameter ε is defined as,

$$\varepsilon = \max\left(0.5,\ \min\left(0.95,\ \frac{\overline{E}'}{\overline{E}}\right)\right) \quad (4)$$

When $T < T_C$, SAMM performs like a definite optimization, such as the steepest descendant method, the uniform design method (UDM)[17]. SAMM prefers UDM because a) no gradient information is needed and b) the computational cost of UDM is in order $O(\dim(\vec{Z}))$. The search space of uniform design method, denoted by Ω, is bounded in a ρ_C -neighborhood at the center of the M molecule at temperature T_C,

$$\Omega = \left\{\vec{Z} : \left\|\vec{Z} - \vec{Z}^{(C)}\right\| < \rho_C\right\} \quad (5)$$

Where, $\vec{Z}^{(C)} = \frac{1}{M}\sum_{j=1}^{M} \vec{Z}^{(T,j)}\Big|_{T=T_C}$, $\rho_C \propto \lambda_0\sqrt{T_C/T_0}$, and $T_C = 10^{-5} \times T_0$.

It is well known that the probability of a move with positive energy increment damps exponentially as the annealing procedure cooling down, and $\vec{Z}$, the molecule, can not jump out of the valley of the energy function at lower temperatures. In another word, the global convergence of SA is reached at higher annealing temperatures and has little relationship with final iterations at lower annealing temperatures[18]. If and only if the critical temperature T_C is much smaller than the initial temperature T_0, most of the M molecule, $\vec{Z}^{(T,j)}$ ($T = T_C$, $j = 1, 2, \cdots, M$) must be in the deepest valley of the energy function. Therefore, Ω includes the global minimum of the analog synthesis problem (1). The intervention of uniform design method in SAMM can avoid the excessively time-consuming final iterations of SA but does no harm to the global convergence of SA. It makes SAMM very efficient and suitable for solving the analog synthesis problem (1). SAMM also performs better than the Chavez's SA-based algorithm[18] because SAMM needs no gradient information.

3. Experimental Results and Analysis

We have implemented an analog synthesizer based on SAMM. In this section, we will give several experiments to demonstrate the efficiency and validity of SAMM. All our results were run on a SUN SPARC-10 workstation.

Experiment 1: Find the global minimum of the 3 humpback camel function[19],

$$\min_{\vec{Z}} \ E(\vec{Z}) = 2z_1^2 - z_1 z_2 + z_2^2 - 1.05z_1^4 + \frac{z_1^6}{6}, \quad (6)$$

The above function has two saddle points and three local minima. The global minimum locates at $E(0,0) = 0$. We discretize variable z_1 and z_2 in 0.0001 unit. It costs SA 2016 iterations to get the global minimum.

SAMM sets parameter M to 4. The annealing temperature drops below $T_C = 10^{-5} \times T_0 = 1.0$ after 934 iterations. At this moment, the center of the 4 molecule is $\vec{Z}^{(C)} = (0.0084, -0.0043)^T$, and ρ_C is 0.040. Then it costs UDM only 80 iterations to get the final solution $E(0.0001, 0.0002) = 4.0 \times 10^{-8}$, which can be considered as the global minimum. So SAMM spends 1014 iterations, less than half of iterations of SA, to reach the global minimum. On average, the computational cost of SAMM is only about 60 percent of that of SA. SAMM is more efficient than SA, because it employs UDM below the critical temperatures and can avoid the excessively time-consuming final iterations of SA.

Experiment 2: Fig.-1 is the supercircuit of simple bipolar amplifiers. There are four binary variables, y_1, y_2, y_3 and y_4. Devices in the box indicated by y_1 is taken away when y_1 is 0, and those devices are included when y_1 is 1. Other binary variables has the same meaning. Constraints must be applied to these binary variables to make the synthesized circuit a reasonable one. In this example, the following constraints must be satisfied:

$$\begin{cases} y_3 \text{ .and. } y_4 = 0 \\ \text{If } \quad y_2 = 1, \quad \text{Then } y_1 = 1 \end{cases} \tag{7}$$

Therefore, the supercircuit can be simplified into 9 bipolar amplifiers. As a synthesis example, we use $2\mu\, m$ bipolar process. Specifications, predicted performances(PP) and SPICE simulated results are listed in Tab.-1. It takes about 87seconds to run each experiment.

Experiment 3: Fig.-2 is the supercircuit of two-stage CMOS operational amplifiers composed by P.C. Maulik[1]. There are 7 binary variables. For simplicity, the constraints for these binary variables are omitted. Tab.-2 is a comparison of Maulik's synthesis results to

Tab.-1 Synthesis examples of bipolar amplifiers

No.	Gain (dB)		UGF(MHz)		Yield
	Spec.	PP/Sim.	Spec.	PP/Sim.	(PP/Sim.)
1	40	44/45	5.0	10/8.0	95%/88%
2	40	43/41	50.0	70/65	90%/91%
3	20	22/22	0.5	0.7/0.7	93%/95%

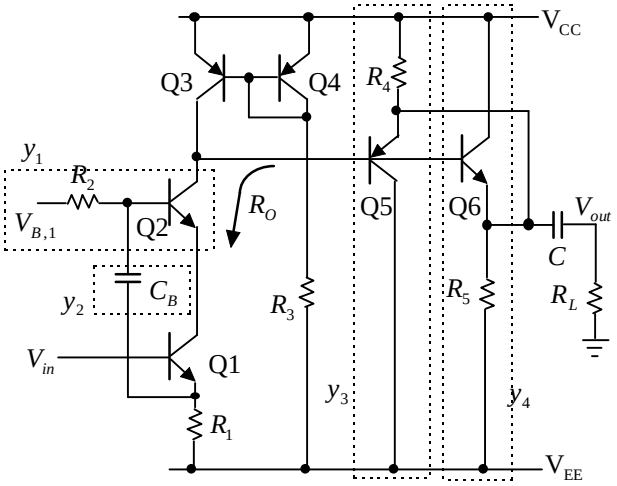

Fig.-1 The supercircuit of simple bipolar amplifiers.

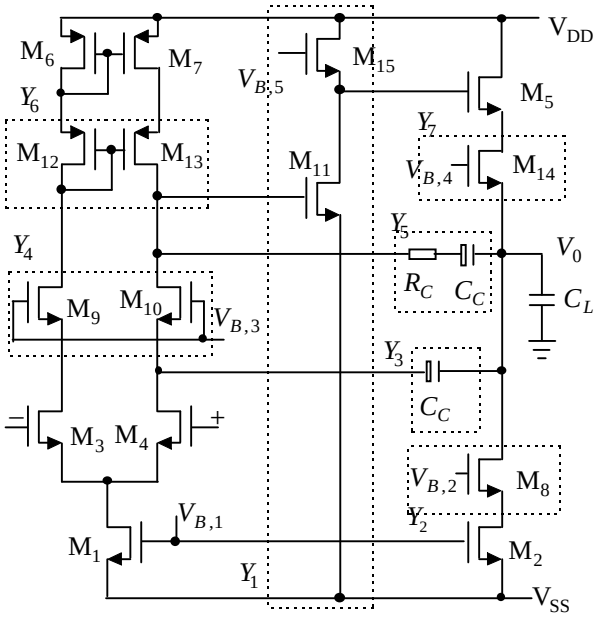

Fig.-2 The Supercircuit of 2-Stage CMOS Op Amps

Tab.-2 Comparison vs. Maulik's synthsis results

Performance	Spec.	Maulik's Results	Ours PP/Sim.
A_0 (dB)	≥ 66	63	68/66
UGF (MHz)	≥ 5	10	10/7.4
SR ($V/\mu s$)	≥ 10	13	10/11
PM (deg)	≥ 45	58	48/50
$PSRR$ (dB)	≤ -60	-69	-62 / -63
$CMRR$ (dB)	≤ -65	---	-69 / -68
Area ($\mu\, m^2$)	---	4357	3920
Yield	---	---	96% / 90%

ours. It costs SAMM about 9.5 minutes to finish the synthesis. Our results is better because a) the chip area is smaller and b) all performances are up to standard.

4. Conclusions

The heart of analog synthesis is to find the global minimum of MINLP problem (1). Compared with most previous optimization techniques, SAMM has three obvious distinctions: a real simultaneous topology selection and device sizing, the Brownian movement of multi-molecule in the perturbation period, and the intervention of uniform design method below the critical temperature. SAMM has global convergence and can avoid the final excessively time-consuming iterations of SA. In the previous section, SAMM is compared with SA and the Maulik's method respectively. These comparisons prove that SAMM is more efficient than previous methods and very suitable for analog synthesis. The hardest aspect of our approach is the composition of the supercircuit, and our future work is to develop an automatic supercircuit composer for analog cells.

References

[1] Maulik P.C., Carley L.R., Rutenbar R.A., "Integer Programming Based Topology Selection of Cell-Level Analog Circuits," IEEE Trans. Computer-Aided Design, Vol.CAD-14(4), Apr. 1995, pp.401-412.

[2] Maulik P.C., "Formulations for Optimization-Based Synthesis of Analog Cells," Research Report No. CMUCAD-92-50, Carnegie Mellon Univ., Oct. 1992.

[3] Yang H.Z., Wang H., Liu R.S., Fan C.Z., "Simultaneous Topology Selection and Sizing for Synthesis of Analog Cells," IEEE TENCON'95, Hong Kong, Nov. 1995, pp.159-162.

[4] El-Turky F.M., Perry E.E., "BLADES: An Artificial Intelligence Approach to Analog Circuit Design," IEEE Trans. Computer-Aided Design, Vol.CAD-8(3), June 1989, pp.680-692.

[5] Klinke R., Fiedler H.--L., Hosticka B.J., Kokozinshi R., Munster I., "Rule-Based Analog Circuit Design," in Proc. European Design Automation Conf. (EDAC), Sept. 1992, pp.480-484.

[6] Koh H.Y., Sequin C.H., Gray P.R., "OPASYN: A Compiler for CMOS Operational Amplifiers," IEEE Trans. Computer-Aided Design, Vol.CAD-9(1), Feb. 1990, pp.113-125.

[7] Gielen G.G.E., Walscharts H.C.C., Sansen W.M.C., "ISAAC: A Symbolic Simulator for Analog Integrated Circuits," IEEE J. Solid-State Circuits, Vol.SC-24(6), Dec. 1989, pp.1587-1597.

[8] Gielen G.G.E., Walscharts H.C.C., Sansen W.M.C., "Analog Circuit Design Optimization Based on Symbolic Simulation and Simulated Annealing," IEEE J. Solid-State Circuits, Vol.SC-25(3), June 1990, pp.707-713.

[9] Gill P., Murray W., Saunders M., Wright M., "User's Guide for NPSOL (Version 4.0)," Tech. Report SOL 86-2, Stanford Univ., Jan. 1986.

[10] Nemhauser G.L., Wolsey L.A., Integer and Combinatorial Optimization, New York: Wiley, 1988.

[11] Kirkpatrick S., Gelatt C.D., Vecchi M.P., "Optimization by Simulated Annealing," Science, Vol.220(4958), May 13, 1983, pp.671-680.

[12] Doganis K., Scharfetter D.L., "General Optimization and Extraction of IC Device Model Parameters," IEEE Trans., Electron Devices, Vol.ED-30, Sept. 1983, pp.1219-1228.

[13] Aluffi-Pentini F., Parisi V., Zirilli F., "Global Optimization and Stochastic Differential Equations," J. Optimiz., Vol.47(1), Sept. 1985, pp.1-16.

[14] Sakurai T., Lin B., Newton A. R., "Fast Simulated Diffusion: An Optimization Algorithm for Multiminimum Problems and Its Application to MOSFET Model Parameter Extraction," IEEE Trans., Computer-Aided Design, Vol.CAD-11(2), Feb. 1992, pp.228-234.

[15] Chiang T.-S., Hwang C.-R., Sheu S.-J., "Diffusion for Global Optimization in R_n," SIAM J. Contr. Optimiz., Vol.25(3), May 1987, pp.737-753.

[16] Ochotta E.S., Rutenbar R.A., Carley L.R., "Equation-Free Synthesis of High-Performance Linear Analog Circuits," Proc. of the 1992 Brown/MIT Conf. on Advanced Research in VLSI and Parallel Systems, 1992, pp.129-143.

[17] Fang K.T., Zhang J.T., "Criteria for Uniform Design," Technical Report Math-002, Hong Kong Baptist College.

[18] Chavez J., Aguirre M.A., Torralba A., "Analog Design Optimization: A Case Study," in Proc. IEEE Intern. Sym. Circ. Syst. (ISCAS), May 1993, pp.2083-2085.

[19] Dixon L.C.W., Szego G.P., (ed.), Towards Global Optimization, North-Holland Pub., 1975.