

# PAGE: Parallel Agile Genetic Exploration towards Utmost Performance for Analog Circuit Design

Po-Cheng Pan\*, Hung-Ming Chen\* and Chien-Chih Lin†

\*Institute of Electronics, National Chiao Tung University, Hsinchu, Taiwan

†Macronix International Co., Ltd., Hsinchu, Taiwan

Email: saxdebreeze@gmail.com, hmchen@mail.nctu.edu.tw and ecool357@gmail.com

**Abstract**—This paper presents an agile hierarchical synthesis framework for analog circuit. To acknowledge the limitation for a given topology analog circuit, this hierarchical synthesis work proposes a performance exploration technique and a non-uniform-step simulation process. Apart from spec targeted designs, this proposed approach can help to search the solutions better than designers' expectation. A parallel genetic algorithm (PAGE) method is employed for performance exploration. Unlike other evolution-based topology explorations, this is the first method that regards performance constraints as input genome for evolution and resolves the multiple-objective problem with the multiple-population feature. Populations of selected performance are transferred to device variables by re-targeting technique. Based on a normalization of device variable distribution, a probabilistic stochastic simulation significantly reduces the convergence time to find the global optima of circuit performance. Experimental results show that our approach on radio-frequency distributed amplifier (RFDA) and folded cascode operational amplifier (Op-Amp) in different technologies can obtain better runtime and higher quality in analog synthesis.

## I. INTRODUCTION

Nowadays, analog circuit block in a SoC design is often critical. Analog components are heavily influenced by nonlinear physical effects, which create a great barrier for analog automation. Overall, the analog synthesis process consists of topology generation [1], [2], [3], circuit sizing [4] and layout synthesis. As topology generation are fully illustrated in [1], [2], [3], the circuit sizing methodologies also take the indispensable role. Comparing to digital system, the development for analog design automation is still in development. Either topology selection or circuit sizing methodologies are considered in time complexity and the accuracy caused by process variation, parasitics effect and operation conditions.

There are plenty of works which are already well-developed on how to find optimal design parameters for a prior selected topology. However, as many performance specifications need to be considered, finding the optima solution for multi-objective performance at sizing stage as a priori problem is still uncertain. It seems that deterministic optimization for circuit sizing still has space to improve. Since deterministic optimization keeps the efficiency and full-circuit SPICE-based simulation maintains the accuracy, selecting methodology for analog circuit sizing is beyond trade-off.

It is therefore essential to have an agile multi-objective synthesizer which explores the limitation towards required technology. Moreover, it is capable of searching the performance space and re-targeting to design parameters with accuracy and efficiency for analog circuit design perspectives.

### A. Previous Works

State-of-the-art analog synthesis methodologies were formulated as a numerical problem which relies on macro-modeling. Usually a complete circuit schematic and the circuit's performance spec are given, then the sizes and biasing value of all devices have to be determined. As a result, the optimal values meet the specs of the required circuit. This kind of optimization engine determines these

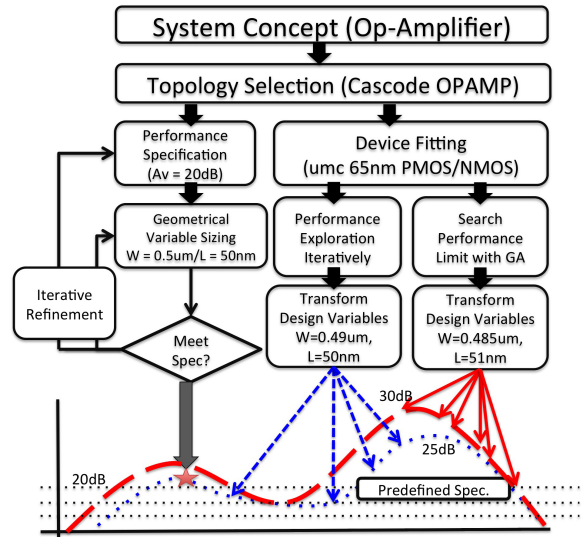


Fig. 1. While traditional sizing strategy iteratively search the solution beyond the fixed spec, a performance constraints exploration approach unfolds the utmost of performance space. Moreover, an evolutionary methodology application elaborates exploration faster and more precise.

optimal values and there exists an evaluation engine to assess the performance. It is very likely that the initial sizing result in a near-optimal design, therefore, a further fine-tuning follows for improving yield and design robustness [5].

In algorithm of [6], a numerical simulation has been employed successfully, while deterministic optimization are developed in [4], [7]. At the same time, methods to acquire accurate yet interpretable analytic formulation as design equations in posynomial forms have been proposed intensively in [8], [9]. Analog/RF design problems can be described as posynomial form in order to be fed as standard input for convex optimization problem. Thus, the convex problem is certainly solved efficiently with accurate design equation [10]. These approaches attempt to generate the mapping from device models to performance metrics, and then utilize the numerical solvers or optimization engines to re-target the corresponding device-level parameters. Although the process of such limited-scaled device modeling is efficient and accurate, the circuit design equations are imprecise and hardly express the global nonlinearity. Another kind of approach [11], [12] shows that the performance space can be modeled in posynomial form for system-level design and trade-off analysis. It directly maps the device-level design variables to performance metrics. Such approach can improve the precision but requires iterations on time-consuming full-sized circuit simulation for optimality. However, it restricts to small-scaled circuit since the execution time depends on the circuit complexity, and usually grows exponentially.

In all, due to the requirement for more probability on performance metrics, we believe that it is important to have mechanism at the

This work was partially supported by the National Science Council of Taiwan ROC under grant No. NSC 101-2220-E-009-040 978-3-9815370-0-0/DATE13/©2013 EDAA

early design stage in exploring the performance limitation before optimization. Meng et al.[13] provide a hierarchical performance Pareto-front mapping methodology to acquire performance metrics before circuit optimization stage. [13] first traverse the performance specs as constraints in a set of convex problem. Therefore, a combination of performance space is generated. Moreover, the corresponding design variables can be obtained according to the design equation. Labrak et al. [14] perform a hybrid optimizer with a multi-objective optimization problem(MOOP) for a CMOS Op-Amp. Our method integrates the hierarchical synthesis strategy with a performance mapping methodology and a non-uniform circuit simulation approach to meet the multi-objective performance requirement wisely.

### B. Our Contributions

According to the trade-off among performance specs on analog designs, it is rarely possible to find an optimal solution for all metrics. Therefore, it is practical to search the performance limitation with agile and accurate synthesis procedure. Referring to [13], Meng et al. attempt to search the performance space by re-targeting back to the corresponding design variables, in iterative manner. Nevertheless, the strategy which costs time complexity up to  $O(S^K)$ (S stands for the performance range in discrete number and K represents the number of performance specification), which is time-consuming and will lose the effective prediction of performance metrics. To our best knowledge, genetic algorithm is well developed for many analog synthesis topics in [15], [3], [16], [17]. On the contrary, the proposed work employs a parallel genetic algorithm (PAGE) for traversing performance space as optimization constraints. A parallel genetic algorithm like [18], [19] can divide the target population into several sub-populations with particular performance specs and reunion after evolution. The obtained populations resulted from PAGE can be re-targeted to the non-uniform stochastic circuit simulation. Fig. 1 illustrates the comparison among the traditional optimization on performance specification, the iterative performance exploration by [13] and our methodology with novel parallel genetic algorithm engine. The proposed work achieves three principle contributions as follows:

- **Hierarchical synthesis architecture.** This synthesis demonstrates a bi-direction search approach for device model fitting to performance metrics through circuit-level domain. After obtaining the corresponding performance space, it re-targets to feasible device level design parameters with a good prediction.
- **Parallel genetic algorithm based multi-objective performance exploration.** A multi-objective evolutionary methodology like parallel genetic algorithm (PGA) not only performs evolution in parallel for diverse performance metrics, but also migrates chromosome by interleaving among populations. PAGE brings out a population of potential solution for selected performance among metrics. Such population is also projected to the feasible design parameters so that we affirm the global optimal solution is located nearby this optimization result by exploration.
- **Non-uniform stochastic circuit simulation.** This work integrates each design variables to analyze the possibility distribution. Other than uniformly swapping the values of design variables in stochastic searching, a interval with higher possibility earns more searching resources. A non-uniform step searching for circuit SPICE simulation is proposed.

The rest of this paper is organized as follows, Section II first define the identity of circuit performance, and exploration objective in this paper. Section III elaborates the flow of our hierarchical synthesis with PAGE. In Section IV, we apply our approach to a radio-frequency distributed amplifier (RFDA) and another operational amplifier (Op-Amp) with different technologies for demonstration. Finally, we draw a conclusion in Section V.

## II. PROBLEM DESCRIPTION

State-of-the-art acknowledges a collection of Pareto-fronts which sketch the performance space, later an optima point is selected

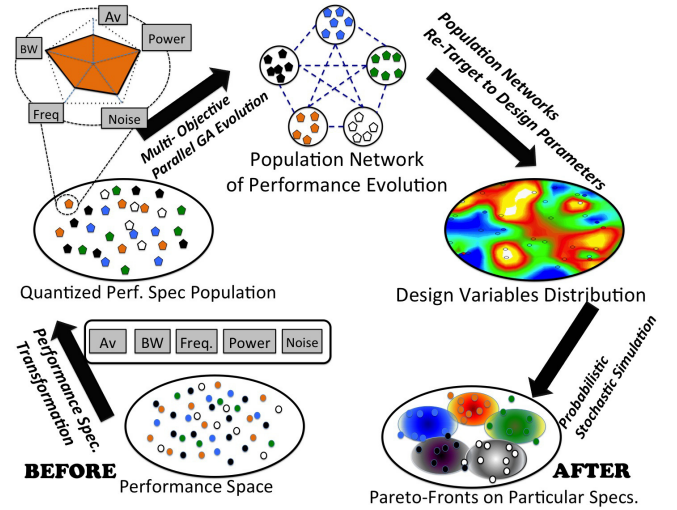


Fig. 2. Originally, a performance space for analog circuit is chaotic without regulation. After transformation to quantized values and performing an evolutionary process, the correlation among performance metrics is explored and converges to a set of populations, which provides guideline for stochastic simulation.

for local search problem which didn't mention that how to define optima point among the space. Instead of collecting the performance space information, this paper aggressively define performance limit as exploration main objective:

**Definition 1: Circuit Performance:** A circuit performance is consisted of multiple values, such as DC voltage gain, 3dB gain bandwidth and power consumption. Different circuit has different circuit performance target.

**Definition 2: Performance limit exploration for global search problem:** Given a circuit design equation in posynomial forms with a set of feasible circuit-level design variables and a set of circuit performance constraints, perform convex optimization with different performance value to traverse the utmost performance space of the given circuit.<sup>1</sup>

**Definition 3: Stochastic simulation for local search:** A set of feasible performance is re-targeted to corresponding design variables. The local search practices a stochastic SPICE simulation w.r.t. these selected design variables.

## III. OVERALL HIERARCHICAL PERFORMANCE EXPLORATION FLOW

This section proposes a framework for analog circuit sizing via resolving multi-objective optimization problem, which can be divided into two major search problems: a global search and a local search process. Fig. 2 shows the overview of our hierarchical methodology.

### A. PAGE Methodology Overview

In order to elaborate the advantages of both deterministic optimization and circuit simulation, this framework tends to integrate parallel genetic algorithm and convex optimization for such global search problem. The framework of PAGE is summarized in Fig. 3. The global search first collect technology information by *Device Fitting to Circuit-level Variables*. As soon as collecting technology information is done, the device variables are mapped into corresponding circuit-level variables, and the non-feasible values of variables are exclusive.

<sup>1</sup>Here, the maximum and minimum performance values are investigated whether feasible or not. Therefore, it can tell that the global search process generates a space of feasible performances and each represents a set of optimal design variables. Although the global search obtains a space of performance, it is not the exact optimal solution. The global search is a preparation for later local search. This work proposes a flexible non-uniform stochastic simulation as local search for optimal sizing solution.

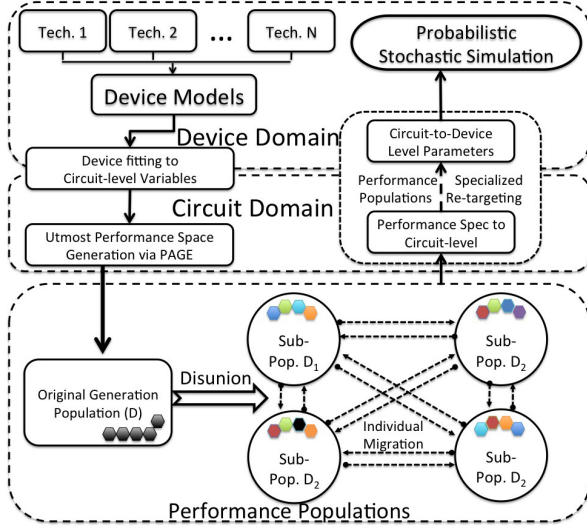


Fig. 3. The Complete Performance Exploration Flow.

Later, *Utmost Performance Space Generation via PAGE* implements parallel genetic algorithm to obtain performance limit as population networks. These populations are evolved by particular fitness function according to each performance spec. Therefore, each population has better performance on particular spec respectively. The local search procedure follows the populations to re-target back to a set of corresponding design variables. According to the distributions, the *Probabilistic Stochastic Simulation* stage implements an efficient simulation procedure. A probabilistic swap strategy is employed on an SA-based simulation, which earns runtime in convergence for optimal solution. In the end, each optima of required specs is found. Moreover, we can tell that the global optimal is close to such optimization result.

### B. Device fitting to Circuit-level Variables

At the begin of synthesis framework, we have an abstraction from device model to circuit-level variables is performed. Given the required device of the target circuit design, the foundry device models provide such device characteristics with SPICE modeling. Inspired by [13], a set of analytical design equations are capable to map device-level variables into circuit-level design variables by modeling techniques such as symbolic analysis or curve fitting like [8], [9], [20]. Two steps of technique accomplish the abstraction for design variables. First of all, it is necessary to discover feasible variable values for each attendant device. In other words, all design variable values which cause device failed should be exclusive at this stage. By SPICE simulator, a matrix of accessible device level variable value are generated. Secondly, such matrix of device-level variables are further mapped into circuit-level variables by curve fitting. An vivid example for design variable mapping is shown in Fig.4.

Given a set of attendant devices  $M = \{m_k | 1 \leq k \leq |M|\}$ , a set of device-level variables  $V^D = \{v_k^D | 1 \leq k \leq |V^D|\}$  and a set of circuit-level variables  $V^C = \{v_k^C | 1 \leq k \leq |V^C|\}$ . A set of feasible device-level variables are constructed in a table T, which  $T = \{t_{a,b} | 1 \leq a \leq |V^D|, 1 \leq b \leq |V_1^D| \times |V_2^D| \dots |V_1^D|\}$ . Each  $V_D$  consists of a range of variable value. All training pairs of the extracted circuit-level variables from previous transformation and device-level variables are then formulated as least-square error problem in analytical posynomial form design equations to acquire fitting parameters.

### C. Utmost Performance Space Generation via PAGE

From previous step, the design equations of the devices, along with the parasitic effects and mapping parameters are integrated into circuit-level design equations for the circuit. Note that the parasitic

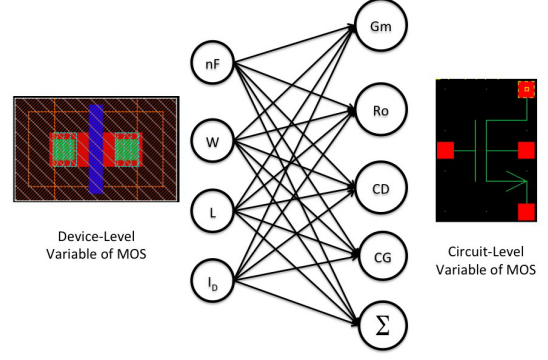


Fig. 4. Mapping device-level variables of one NMOS(number of fingers, device channel width/length and current) to circuit-level variables ( $G_m$ ,  $R_o$ ,  $C_D$ ,  $C_G$  and  $\Sigma$ )

effects of the devices are included so as to explore the trade-off between each aspect of performance combination. An optimization problem in Eq.(1) describes a unit performance optimization step.

$$\begin{aligned}
 \text{Variables : } & \quad v_i^C | 1 \leq i \leq |V^C| & V^C : \text{circuit variables} \\
 & \quad p_k | 1 \leq k \leq |P| & P : \text{parasitics} \\
 & \quad f_k | 1 \leq k \leq |F| & F : \text{fitting parameters} \\
 & \quad r_k | 1 \leq k \leq |R| & R : \text{performance result} \\
 \text{minimize } & \quad f_{OBJ}(v_i^C, p_k, f_k) \\
 \text{subject to } & \quad r_1 = Perf_1(V^C, P, F) \geq z_1 \\
 & \quad r_2 = Perf_2(V^C, P, F) \geq z_2 \\
 & \quad \vdots \\
 & \quad r_k = Perf_k(V^C, P, F) \geq z_k
 \end{aligned} \tag{1}$$

Each optimization result in a set of performance value ( $r_1, \dots, r_k$ ) corresponding to the given specification of performance ( $z_1, \dots, z_k$ ). Therefore, according to the same design equation for optimization, it is a one-by-one mapping relationship from spec to result of performance and the corresponding circuit-level design variables. As a result, it is similar to a set of chromosome. An evolutionary computing with genetic algorithm for traversing solution space is employed. A set of performance boundaries,  $\{[Z_{min}, Z_{MAX}] = \{z_{kmin}, z_{kmax} | 1 \leq k \leq K\}$ , on the performance are swept as the constraints for an optimization problem in [13]. Here we expand the performance space as an  $S \times N$  matrix in Eq.(2). Moreover, each performance spec from constraints is encoded as chromosome G from maximal to minimal in a set of  $G = \{g_k | 1 \leq k \leq |G|\}$ . For example,  $g_i \in G$  randomly obtains value of the  $k^{th}$  spec from  $z_{k,1}$  to  $z_{k,S}$ .

$$Z_{K \times S} = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1S} \\ z_{21} & z_{22} & \dots & z_{2S} \\ \vdots & \vdots & \vdots & \vdots \\ z_{K1} & \dots & \dots & z_{KS} \end{bmatrix} \tag{2}$$

where

- K: the number of performance specification types. (eg: Av, BW,...,etc.)
- $[Z_{min}, Z_{MAX}]_k$ ,  $k = 1, \dots, K$  is the  $k^{th}$  type specification range of the performance space.
- S is the sampling number for each  $Z_S$  between  $Z_{kmin}$  and  $Z_{kMAX}$ .
- $\forall z_{ki} \in Z_{K \times S}$ , if  $i = 1$ ,  $z_{ki} = Z_{Smin}$  and if  $i = S$ ,  $z_{ki} = Z_{SMAX}$

Our PAGE approach is summarized as shown in Algorithm1. As described in *Input*, a set of performance space matrix  $Z_{K \times S}$  is given. At the beginning of PAGE, a major population is constructed



---

**Algorithm 1** PAGE( $Z_{K \times S}, N_P, N_S, S, k, T, C$ )

$Z_{K \times S}$ : Performance Space Matrix,  
 $N_P$ : Number of Individuals in major population.  
 $N_S$ : Number of sub-populations.  
**Input:**  $S$ : Sampling number for each performance spec  $z_k$   
 $k$ : Number of Performance spec type  
 $T$ : Technology type.  
 $C$ : Circuit type.  
**Output:**  $R_{K \times S}$ : The result performance space  
 $P$ : The population of performance specs after PAGE  
1: **for**  $i = 1 \rightarrow N_P$  **do**  
2:   **for**  $j = 1 \rightarrow k$  **do**  
3:      $G_i \leftarrow \text{RandGetSpec}(Z_{j \times S})$  {Randomly generate spec value from Z}  
4:   **end for**  
5:    $P \leftarrow G_i$   
6: **end for**  
7: **for**  $i = 1 \rightarrow N_S$  **do**  
8:    $P_i \leftarrow \text{Partition}(P, i)$   
9: **end for**  
10: **while** Convergence criterion satisfied **do**  
11:   **for all**  $i = 1 \rightarrow N_S$  **do in parallel**  
12:      $R_i \leftarrow \text{Evaluate}(P_i, T, C)$   
13:      $\text{Pool}_i \leftarrow \text{Reproduction}(P_i, \text{Fitness}(T, C, k, R_i))$   
14:      $P_i \leftarrow \text{Crossover}(P_i, \text{Pool}_i)$   
15:      $P_i \leftarrow \text{Mutation}(P_i)$   
16:   **end for**  
17:   **for all**  $i = 1 \rightarrow N_S$  **do in parallel**  
18:      $\text{Migration}(P_i, \text{exclusive}(P, P_i))$   
19:   **end for**  
20: **end while**  
21:  $P \leftarrow \text{Merge}(P_1, P_2, \dots, P_{N_S})$   
22: **return**  $P, R_{K \times S}$

---

according to  $Z_{K \times S}$ . Line 1 to Line 6 illustrates the process to assign performance specs as chromosome for each individuals of the major population iteratively. Therefore, a major population  $P$  is generated. According to the size of sub-population  $N_S$ , master processor allocates individuals to each slave processor uniformly as sub population  $P_1 \dots P_{N_S}$ . A evolution is executed between Line 10 and Line 20. Because the *Evaluate*, *Reproduction*, *Crossover* and *Mutation* part are independent, the parallel parts start from Line 11 to Line 16. Hence, in each sub-population, each location of gene in one individual is fed into  $\text{Evaluate}(P, T, C)$  as target constraints for performance to the design equation (shown in Eq.(1)) and a corresponding result  $R_i$  is obtained. However, if one combination of performance metrics sketch out the space which is not convex, such individual would be failed by *Evaluate*. Then, a random generated individual replaces and redo *Evaluate* again until each chromosome  $G$  has its corresponding result  $R = \{r_k | 1 \leq k \leq |R|\}$ .

Fig. 5 shows the flow of the parallel genetic evolution from random performance space matrix ( $Z_{K \times S}$ ) to convergence. Each sub-population experience a evolution with *Evaluate*, *Reproduction*, *Crossover* and *Mutation*. Between Line 12 to Line 15, PAGE utilizes a fitness function to determine the suitability for each  $G_i$ . According to our requirement, we tend to specialize the particular spec, such as voltage gain ( $A_v$ ). A set for fitness function is given,  $\text{Fitness} = \{\text{fitness}_i | 1 \leq i \leq N_S\}$  as kind of objective function to determine how important does each individual is with the *Evaluate* result. Each sub-population  $P_i$  applies one particular fitness function which is related to the required performance result  $R_k$  of *Evaluate* value. Therefore, the fitness function determines the qualified individuals to be preserved to the crossover pool in a weighted ratio, and the others should be extinct. In *Crossover* of Line 14, the *Crossover* step selects each two genes  $G_i$  and  $G_j$ , where  $\{G_i, G_j \in \text{Pool}; 1 < i < j < N\}$  for copulation. Since each individual has  $K$  types of spec, these two individuals exchange  $c$  specs and reserve  $K - c$  specs with each other. In the end of parallel sub-level evolution, the *Mutation* step picks up one individual with mutation rate and then replaces one gene value by one slot of  $Z_K$ .

For each sub-population  $P_i$ , we perform *Mutation* and obtain an updated  $P_i$ . *Migration* in Line 18 of Algorithm 1 aims to exchange individuals in the population network shown in the bottom of Fig. 3. According to [19], our approach selects the coarse-grained parallel

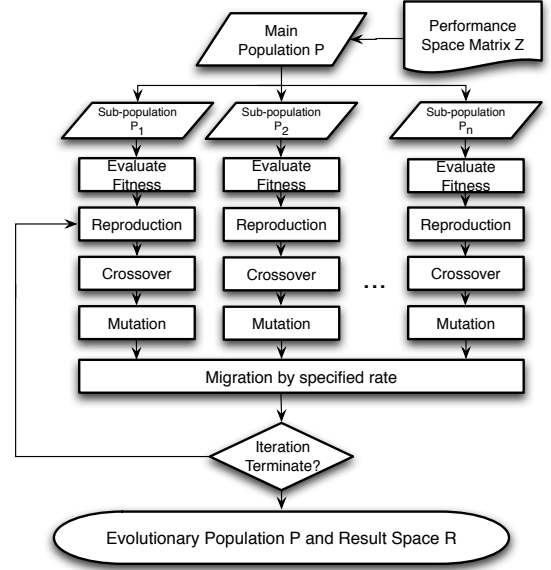


Fig. 5. The Coarse-grained parallel genetic approach from major population  $P$  to partitioned population  $P_i$  for parallel evolution. The migration step benefits each sub-population on diversity every iteration.

genetic algorithm (CoPGA) in order to increase the diversity for each sub-population. We use the ring-shape population network illustrated in Fig.3. Therefore, each  $P_i$  should operate *Migration* with the others. In each *Migration* between  $P_i$  and  $P_j$ ,  $P_i$  exchanges its best individual with respect to  $\text{fitness}_j$ , and vice versa. After all the *Migration* executed, the composition of each  $P_i$  is updated with higher diversity. After all, while the termination condition meets, all the sub-populations are merged for next step re-targeting.

To consider the complexity of PAGE, it is considerable to check the dimension of  $Z_{K \times S}$ . According to line 12 in Algorithm 1,  $\text{Evaluate}(P_i, T, C)$  is the most critical. Each *Evaluate* for  $P_i$  need to resolve  $\frac{N_P}{N_S}$  convex optimization problems in serial, which is  $O(\frac{N_P}{N_S})$ . Comparing to the exhaustive approach, the complexity is restricted to  $K$  and  $S$ . That is, we need to traverse every combination from specification space in  $Z_{K \times S}$  for  $O(S^K)$  complexity. For example, if a circuit require two performance specs ( $A_v$  and  $BW$ ) with 4 sampling steps ( $A_v = \{5, 10, 15, 20\}$ ,  $BW = \{3MHz, 5MHz, 8MHz, 10MHz\}$ ). Therefore, the overall combination of specs need to be checked is  $4^2 = 16$ . Obviously, we can observe the gap with more specs ( $K \uparrow$ ) and greater precision in sampling ( $S \uparrow$ ). To sum up, a genetic-based approach can reduce the complexity by controlling population number and a parallel enhancement further reduce the timing complexity vi parallel number. However, the accuracy is also sacrificed as trade-off.

#### D. Performance Metrics to Design Variables via Re-targeting

After generating specialized populations by parallel genetic algorithm evolution, this step is a reverse interpolation from a series of performance specifications through circuit-level design variables to device-level design variables. Hence,  $K$  groups of optimal-potential performance spaces of the circuit under chosen technology are traversed. Since a set of performance metrics directly represents the limitation of specifications, such group of optimal performance specification is locked from optimal performance. Ideally, the optimization engine should be capable of directly finding the geometry-biasing-level design variables.

Next, we want to find the optimal candidates of device-level design variables. From previous stage, the circuit-level design variables are obtained. Here the distribution of the device-level design variables

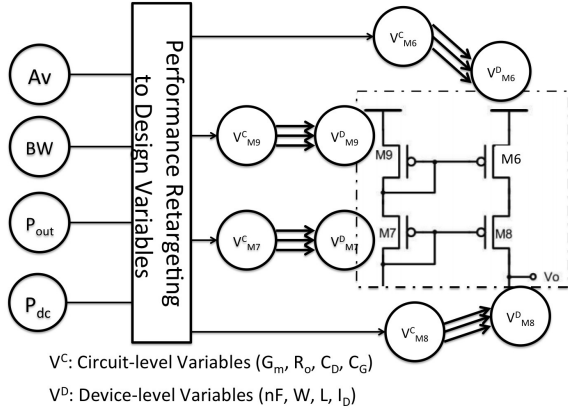


Fig. 6. Given merged populations of performance specs, a reversed process is to retrieve the corresponding design variables of each device (PMOS) in the circuit for optimal sizing values.

are also obtained through this design re-targeting stage. As Eq.(3) illustrated, since each device has its particular device variables, all of them should be considered for optimization.  $|M|$  stands for a set of attendant devices in the circuit. Thus,  $N_{V^D}$  collects the number of all device variables type. A set of overall device level variable values  $\Psi$  are collected from populations as shown in Fig.6.

$$\begin{aligned}
 M &= \{m_k | 1 \leq k \leq |M|\} \\
 D &= \{d_k | 1 \leq k \leq |D|\} \\
 N_{V^D} &= \sum_{i=1}^{|M|} |V_i^D| \\
 \Psi &= \{V_{i,j}^D | 1 \leq i \leq |D|, 1 \leq j \leq N_{V^D}\}
 \end{aligned} \tag{3}$$

#### E. Probabilistic Stochastic Circuit Simulation

The final step performs a full-circuit stochastic simulation. Instead of analyzing all possible value of each device's variables, a probabilistic approach for circuit simulation is proposed. Since these design variables are transformed from performance populations by PAGE, each population of performance metrics directly projects to a set of variable distribution. Given a set of value of variables, a normalized distribution for such design variables is obtained by calculating the mean value and standard deviation. In other words, it represents the probability distribution function (PDF) for that device-level variable. Thus, a set of PDF for each device-level variable is generated.  $PDF = \{pdf_k | 1 \leq k \leq |\Psi|\}$

We assign a maximum and minimum values for each  $V^D$  from technology design rules, and apply stochastic circuit simulation among these variable boundaries with an SA-based search. Each  $V^D$  is uniformly divided into step values from  $V_{max}^D$  to  $V_{min}^D$ . While performing stochastic simulation, each swap is determined according to the  $pdf$  of such  $V^D$ . In other words,  $pdf_k(V_K^D)$  effects the possibility to simulate such variable values.

A hierarchical synthesis for circuit sizing strategy is accomplished after the optimal solution is converged in that stochastic simulator or the termination requirement is met.

## IV. EXPERIMENTAL RESULTS

As demonstration of our methodology, we obtain a radio-frequency distributed amplifier(RFDA) in [13] and another folded cascode operational amplifier to be synthesized automatically by our framework through three technology processes: umc 65nm, umc 90nm and tsmc 90nm. Table I shows the statistics of the two circuits, including the components of each and the original performance specifications. Our methodology is implemented by GCC version 4.3.4, Matlab R2011a,

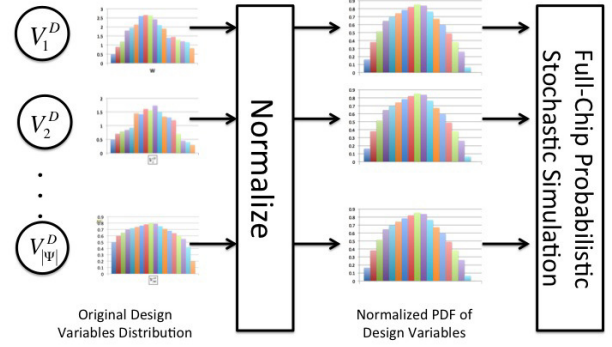


Fig. 7. Given a set of device variables distribution  $\Psi$ , each variable  $V^D$  transforms to a normalized probability distribution function (PDF), and then a probabilistic full-chip stochastic simulation performs swapping strategy according to these PDFs

TABLE I. DEVICE STATISTICS OF RFDA AND OPAMP

circuits	Device Number				
	MOS	Capacitor	Resistor	Inductor	Total
RFDA	12	30	0	30	72
Spec	$A_v(\frac{V}{V})$	$P_{dc}(\mu W)$	$P_{out}(mW)$	$F_{cent}(GHz)$	$BW(GHz)$
	$\geq 5$	$\leq 0.5$	$\geq 2$	$\geq 5$	$\geq 10$
Op-Amp	18	1	0	0	19
Spec	$A_v(\frac{V}{V})$	$P_{dc}(\mu W)$	$P_{out}(\mu W)$	$BW(MHz)$	Phase Margin
	$\geq 40$	$\leq 100$	$\geq 0.1$	$\geq 60$	$\geq 50$

and the cvx optimization package. Since we rewrite the Matlab cvx package with our design equations of RFDA and Op-Amp into our c++ based PAGE, the interpretation is accomplished by Matlab Runtime Compiler 4.15 on Intel Xeon E5620 2.4GHz with 72GB memory. The parallel computation threads we utilize here are 4 to 8 threads for evaluation in this PGA-based exploration.

Here we implement frameworks for performance exploration as follows:

- 1) **Meng et al. in [13]**, an exhaustive performance exploration method with a basic stochastic circuit simulation.
- 2) **AGE**, a genetic algorithm based performance exploration method with basic stochastic circuit simulation framework.
- 3) **PAGE**, a parallel genetic algorithm based performance exploration method with probabilistic stochastic circuit simulation framework.

Table II shows the comparison among above three frameworks for analog circuit synthesis. The first column denotes 2 circuits, RFDA and Op-Amp. In RFDA case, we only apply umc 65nm technology, which is the same as previous work. In addition, three technologies, umc 65nm, umc and tsmc 90nm are applied for Op-amp circuit. The left third and 4th column show the overall runtime and improve percentage, and the right five represent the performance values of one optimal point in each methodology.

By definition, PAGE can search the limitation of performance metrics and also find the performance Pareto-fronts with particular populations. Since genetic algorithm has ability to traverse different combination with crossover and mutation wisely, our approach can collect a bunch of potential spec combinations as a population, and transfer them to re-target back to the desired design variables. Not only obtaining a good initial performance metric population is important, also a design equation which can precisely sketch out the circuit characteristic is necessary. However, we tend to keep the stochastic simulation for a final search, but also using an evolutionary methodology to reduce the convergent resource. As we can see, AGE already earns runtime improvement at umc65-RFDA, umc65-OPamp, umc90-OpAmp and tsmc90-Opamp with 398% , 227%, 174% and 224% than the exhaustive way respectively. Moreover, the PAGE earns even better quality by simultaneously explore the performance

TABLE II. THE PERFORMANCE RESULTS FOR RFDA AND OP-AMP CIRCUIT WITH UMC 65NM, UMC 90NM AND TSMC 90NM TECHNOLOGIES ON [13], AGE AND PAGE FRAMEWORK

RFDA	Algorithm	Runtime(s)	Improv.(%)	$A_v (\frac{V}{V})$	$P_{dc}(\mu W)$	$P_{out}(mW)$	$F_{cent}(GHz)$	$BW(GHz)$
umc 65nm	[13]	38880	-	6.4322	0.23	2.65	9.85	17.48
	AGE	9756.02	398%	7.38	0.175	23.3	20.9	40.2
	PAGE	6300.15	617%	8.505	0.183	18.8	22.5	40.4
Op-Amp	Algorithm	Runtime(s)	Improv.(%)	$A_v (\frac{V}{V})$	$P_{dc}(\mu W)$	$P_{out}(\mu W)$	$BW(MHz)$	Phase Margin
umc 65nm	[13]	19432	-	45.73	102	0.21	144	45.7
	AGE	8568	227%	44.88	93.76	0.428	102	67.84
	PAGE	3424	568%	44.17	93.94	0.527	102	67.7
umc 90nm	[13]	15285	-	33.16	95.6	1.72	78.58	65.872
	AGE	8797	174%	44.247	96.36	0.38	111	74.45
	PAGE	3583.6	427%	44.981	95.11	0.763	110	74.4
tsmc 90nm	[13]	19488	-	38.42	111	1.2	284	50
	AGE	8703	224%	40.46	100.1	0.27	100	82
	PAGE	4651	419%	41.36	99.62	0.22	87	82.1

space with 617%, 568%, 427% and 419% than [13] respectively. The runtime report from each GA-based experiment shows the good quality in efficiency.

For the target performance result, in umc 65nm RFDA, all performance requirements have better quality than the exhaustive way. AGE explores the unprecedented results on  $BW$  and  $P_{out}$ , while PAGE has obvious improvement on  $A_v$  and  $F_{cent}$ . On first umc65 folded cascode Op-Amp, although [13] has good quality on  $A_v$  and  $BW$ , the  $PM$  and  $P_{out}$  are sacrificed. On the other hands, AGE and PAGE come to the quality on each performance target. We can tell that genetic-algorithm based approach can balance the multi-objective optimization via evolution. For umc90-Opamp case, [13] generated the results far from optimal region with  $A_v$  and  $BW$  to earn the better output power  $P_{out}$ . As well as the tsmc90-Opamp case,  $A_v$ ,  $P_{dc}$  and  $PM$  are poor while such approach searched into optimal region of  $P_{out}$  and  $BW$ . The quality for exhaustive methodology is uncertain and time-consuming. On the contrary, as the lower part of Table II, AGE successfully searches the optimal region of all target performances. Moreover, PAGE reaches the better quality on  $A_v$ ,  $P_{dc}$  and  $P_{out}$  for umc90-Opamp case respectively. Likewise, PAGE also explores new limitation for  $A_v$ ,  $P_{dc}$  and  $PM$  at tsmc90-OpAmp as different technology implementation. As a result, the exhaustive approach for performance exploration needs more timing resource to explore the Pareto-fronts but the uncertainty is indispensable. In contrast, the parallel genetic-algorithm based approach for performance exploration with the probabilistic stochastic simulation resolves the problem efficiently and effectively.

## V. CONCLUSION

In this paper, we have proposed a performance utmost exploration framework for analog synthesis framework with a parallel genetic algorithm based approach to efficiently explore a potential performance space for optimal solution. Unlike exhaustive search the performance space which is time-consuming, this work first transforms the problem set as chromosome and then implements a parallel evolutionary algorithm to resolve multi-objective performance optimization. After a re-targeting transformation between performance and design variables, we also implement a probabilistic stochastic simulation with respect to the design variable distribution. Our methodology also minimizes time to converge the global optima with accuracy. As demonstration for our methodology, a RFDA circuit and an Op-Amp are practiced via 3 different technologies to show that our proposed performance exploration approach and probabilistic stochastic simulation are effective and efficient for analog circuit synthesis.

## REFERENCES

- [1] T. McConaghy, P. Palmers, G. Gielen, and M. Steyaert, "Automated extraction of expert knowledge in analog topology selection and sizing," in *International Conference on Computer-Aided Design*, pp. 392–395, Nov. 2008.
- [2] P. Maulik, L. Carley, and R. Rutenbar, "Integer programming based topology selection of cell-level analog circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, pp. 401–412, Apr. 1995.
- [3] T. McConaghy, T. Eeckelaert, and G. Gielen, "Caffeine: template-free symbolic model generation of analog circuits via canonical form functions and genetic programming," in *Design, Automation and Test in Europe*, vol. 2, pp. 1082–1087, Mar. 2005.
- [4] M. Eick and H. E. Graeb, "Mars: Matching-driven analog sizing," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 31, pp. 1145–1158, Aug. 2012.
- [5] R. Rutenbar, G. Gielen, and J. Roychowdhury, "Hierarchical modeling, optimization, and synthesis for system-level analog and rf designs," *Proceedings of the IEEE*, vol. 95, pp. 640–669, Mar. 2007.
- [6] R. Schwencker, F. Schenkel, H. Graeb, and K. Antreich, "The generalized boundary curve—a common method for automatic nominal design and design centering of analog circuits," in *Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings*, pp. 42–47, 2000.
- [7] H. Habal and H. Graeb, "Constraint-based layout-driven sizing of analog circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, pp. 1089–1102, Aug. 2011.
- [8] J. Kim, J. Lee, L. Vandenberghe, and C.-K. K. Yang, "Techniques for improving the accuracy of geometric-programming based analog circuit design optimization," in *International Conference on Computer-Aided Design*, pp. 863–870, Nov. 2004.
- [9] T. Eeckelaert, W. Daems, G. Gielen, and W. Sansen, "Generalized posynomial performance modeling," in *Design, Automation and Test in Europe Conference and Exhibition*, pp. 250–255, June 2003.
- [10] J. Kim, R. Jhaveri, J. Woo, and C.-K. K. Yang, "Device-circuit co-optimization for mixed-mode circuit design via geometric programming," in *International Conference on Computer-Aided Design*, pp. 470–475, Nov. 2007.
- [11] X. Li, P. Gopalakrishnan, Y. Xu, and T. Pileggi, "Robust analog/rf circuit design with projection-based posynomial modeling," in *International Conference on Computer Aided Design*, pp. 855–862, Nov. 2004.
- [12] X. Li, J. Wang, L. Pileggi, T.-S. Chen, and W. Chiang, "Performance-centering optimization for system-level analog design exploration," in *International Conference on Computer-Aided Design*, pp. 422–429, Nov. 2005.
- [13] K.-H. Meng, P.-C. Pan, and H.-M. Chen, "Integrated hierarchical synthesis of analog/rf circuits with accurate performance mapping," in *International Symposium on Quality Electronic Design*, pp. 1–8, Mar. 2011.
- [14] L. Labrak, T. Tixier, Y. Fellah, and N. Abouchi, "A hybrid approach for analog design optimisation," in *Circuits and Systems, 2007. MWSCAS 2007. 50th Midwest Symposium on*, pp. 718–721, Aug. 2007.
- [15] D. L. Wim Kruiskamp, "Darwin: Cmos opamp synthesis by means of a genetic algorithm," in *Design Automation Conference*, pp. 433–438, 1995.
- [16] E. Martens and G. Gielen, "Top-down heterogeneous synthesis of analog and mixed-signal systems," in *Proceedings of Design, Automation and Test in Europe*, pp. 1–6, Mar. 2006.
- [17] P. Conca, G. Nicosia, G. Stracquadanio, and J. Timmis, "Nominal-yield-area tradeoff in automatic synthesis of analog circuits: A genetic programming approach using immune-inspired operators," in *Adaptive Hardware and Systems, 2009. AHS 2009. NASA/ESA Conference on*, pp. 399–406, Aug. 2009.
- [18] E. Cant-Paz, "A survey of parallel genetic algorithms," May 1997.
- [19] E. Alba and J. M. Troya, "A survey of parallel distributed genetic algorithms," *Complexity*, vol. 4, p. 3152, May 1999.
- [20] W. Daems, G. Gielen, and W. Sansen, "An efficient optimization-based technique to generate posynomial performance models for analog integrated circuits," in *Design Automatic Conference*, pp. 431–436, June 2002.