Layout-Aware Sizing Methodology for Analog Integrated Circuits

by
©Tuotian Liao

A dissertation submitted to the School of Graduate Studies
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

**Faculty of Engineering & Applied Science
Memorial University of Newfoundland**

Supervisory Committee
Dr. Lihong Zhang (Supervisor)
Dr. Cheng Li
Dr. Vlastimil Masek

May 2021

St. John's, Newfoundland

Abstract

The traditional iterative design flows for analog integrated circuit synthesis, which can help meet circuit performance requirements in the conventional technology processes, often experience longer runtime. The nonnegligible impact of layout parasitics and layout dependent effects (LDEs) on electrical performance has posed increasingly greater challenges to determining circuit parameters (i.e., circuit sizing), which makes it harder for designers to close the synthesis loop especially in the advanced nanometer technologies. This dissertation is focused on parasitic-aware and LDE-aware circuit sizing solutions in the early schematic design stage of the circuit synthesis process. A number of techniques, which include analytical modeling for devices and circuits, mathematical programming, sensitivity analysis, curve fitting, and heuristic optimization as well as machine learning, are utilized to construct the proposed circuit sizing methodologies. In this regard, we combine geometric programming and differential evolution as well as a many-objective evolutionary algorithm to construct a novel two-phase hybrid sizing methodology for dealing with parasitics. In addition, we propose to use g_m/I_D-based mixed-integer nonlinear programming to improve the accuracy of the first-phase sizing, and adapt it to address the layout-dependent effects with the aid of sensitivity analysis. Furthermore, we develop a machine-learning based approach called Bayesian optimization featuring high-dimensionality and many objectives to tackle parasitics and LDEs for analog circuit sizing. The ultimate objective of this research is to develop efficient methodologies and algorithms to include the consideration of parasitics and LDEs from layout design into schematic design stage as an early action to reduce the analog IC design iterations. The experimental results show the efficacy of our proposed sizing methodologies over other similar works for the layout-aware analog circuit sizing.

Acknowledgments

I would like to express my sincere gratitude to my supervisor Dr. Lihong Zhang for his continuous support in my PhD study, research, motivation, enthusiasm, immense knowledge and his assistance in writing papers and dissertation. I also intend to thank the other members of my supervisor committee, Dr. Li and Dr. Masek, for their guidance and suggestions.

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, in part by Canada Foundation for Innovation, in part by the Research and Development Corporation of Newfoundland and Labrador through its Industrial Research and Innovation Fund and ArcticTECH R&D Award, and in part by the Memorial University of Newfoundland.

To my adorable parents

Table of Contents

Table of Contents

List of Tables

List of Figures

List of Algorithms

List of Abbreviations

Complementary Metal-Oxide-Semiconductor	**CMOS**
Differential Evolution	**DE**
Electronic Design Automation	**EDA**
Figure of Merit	**FOM**
Gaussian-Process-Based Bayesian Optimization	**GP-BO**
Geometric Programming	**GeoP**
High-Dimensional Many-Objective GP-BO	**HMBO**
Integrated Circuits	**IC**
Layout-Dependent Effect	**LDE**
Length of Oxide Diffusion	**LOD**
Lookup Table	**LUT**
Low-Noise Amplifier	**LNA**
Many-Objective Evolutionary Algorithm	**Many-OEA**
Multi-Objective Evolutionary Algorithm	**MOEA**
Operational Amplifier	**Op-Amp**
Radio Frequency	**RF**
Pareto Front	**PF**
Simulated Annealing	**SA**
Sing-Objective Evolutionary Algorithm	**SOEA**
Well Proximity Effect	**WPE**

Chapter 1 Introduction

The semiconductor industry aims at developing more compact electronic products while maintaining higher speed and increasing functionality at lower cost. Moore's Law provides sound prediction to the scalability of MOSFETs in industry that facilitates the achievement of this objective. However, along with the continuous advancement of complementary metal oxide semiconductor (CMOS) technology, some known drawbacks, such as strong impact of parasitics, short channel effect, interconnection problems, and layout-dependent effects (LDEs), have become more prominent in the advanced technologies.

From the old technology processes to the contemporary 20nm and below technology nodes, analog integrated circuit synthesis flow is never an obsolete topic as it is the key from the designers' perspective to provide a stable, malfunction-free, and low-cost design regarding power, chip area and redesign effort, and further a successful tape-out with sufficient design-for-manufacturability included. In the course of pursuing a high quality tape-out design, the LDEs, which are not prominent at old technology nodes, become increasingly influential with respect to circuit performance in the advanced technologies. Electrical parameter variations have been widely observed due to the stress incurred effects. In addition, the space among devices and interconnecting wires becomes closer as the technology node advances, which retains the importance of considering parasitics in the design of integrated circuits. Especially for the analog and radio-frequency (RF) integrated circuits, circuit electrical performance can be very sensitive to parasitics and/or parasitic mismatches.

A macro standpoint of analog and RF circuit synthesis depicted in Fig. 1 includes topology selection, circuit sizing and layout generation (i.e., placement, routing, and extraction for parasitics as well as LDEs). In the microscopic at the circuit level, the circuit synthesis only comprises topology formation and circuit sizing, while layout synthesis referring to the stage of layout generation resulting in a post-layout netlist is separated from the circuit synthesis. A post-layout simulation is needed to verify the design before fabrication. According to Kruiskamp and Leenaerts [1], circuit topology selection is to select device set out of hundreds of combinations. Each set behaves as one stage of the whole design at the schematic level. For instance, an operational amplifier (Op-Amp) is composed of an input stage, a gain stage, and an optional output buffer. Some detailed classification of design automation techniques for topology synthesis can be found in [2].

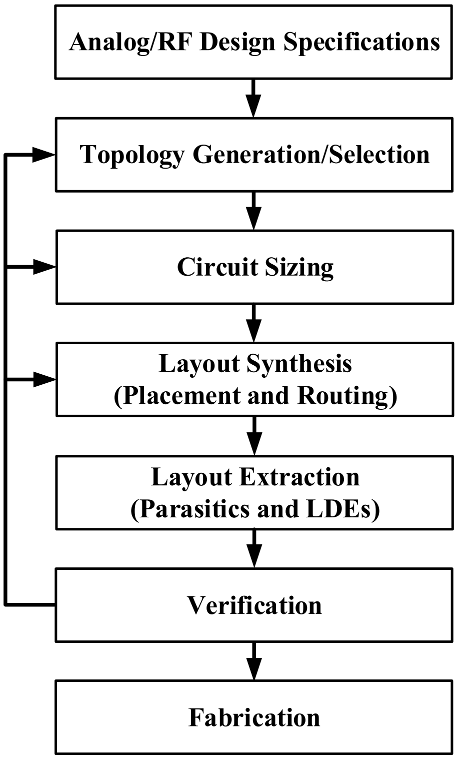


Fig. 1. Analog/RF circuit synthesis flow

With respect to circuit sizing, this design stage is aimed at determining various device geometries and electrical biases, which are essential in early part of the design flow. Device geometry, specifically in the CMOS technology, mainly refers to transistor width (*W*) and length (*L*) among others, and resistor/capacitor/inductor nominal values. The electrical bias may include circuit biasing voltage or current information among the sizing variables. Until now, circuit sizing is still mostly done manually or semi-automatically by experienced analog designers and therefore is a time-consuming and error-prone task [3]. Automated sizing tools are normally very application (i.e., circuitry) specific and problem (i.e., specification) specific.

Layout generation following the completion of the sizing stage is a critical process, which can significantly affect the performance of fabricated chips. It is common that a well-designed circuit at the schematic level but omitting layout consideration is not able to function after fabrication in the advanced technologies. Layout information refers to physical placement and interconnection with the implication of parasitics and performance-related effects caused by neighboring devices and common underneath substrate. Those effects, which are found to be prominent, can cause performance degradation when an ideally symmetric structure (e.g., current mirror and differential pair) appears in a mismatch manner physically. As the technology scales down towards even finer grid, LDEs become more significant. However, the parasitics and LDEs cannot be fully detected until a schematic is converted to its corresponding layout in the traditional analog IC design flow (i.e., Fig. 1). Thus, analog designers may have to go back to the schematic stage to pursue another design solution if the performance degradation due to parasitics and LDEs cannot be alleviated by any subsequent layout refinement/modification. In such cases, plenty of tweaking effort including re-sizing, re-placement, and re-routing is expected to close the synthesis loop.

Therefore, it is no longer sensible for the circuit designers to stop at designing a sized circuit topology and toss the consideration of parasitics and LDEs over the wall to the layout designers. The optimization of those elements in the modern high-performance analog design calls upon either a more intensive cooperation between the two groups of designers or an advanced coordination mechanism that can help pass guidelines of optimizing parasitics and LDEs (e.g., the reference values of the related parameters) to the layout implementation [4]. As a reaction, some so-called layout-aware synthesis approaches, which are reviewed in [5], have come into being. In this dissertation, we are motivated to explore better analog/RF circuit synthesis flows, methodologies, and algorithms to consider the layout parasitics and LDEs in the schematic synthesis stage (i.e., circuit sizing stage) as an early action to reduce the whole circuit synthesis runtime while attaining satisfactory circuit performance.

The rest of the dissertation is organized as follows. Chapter 2 reviews analog/RF electronic design automation (EDA) challenges with regard to layout effects, the definition of analog circuit sizing problem, and the previous related works in the area of analog/RF circuit sizing. Chapter 3 demonstrates the geometric programing (GeoP) and evolutionary algorithm (EA) based hybrid methodology for parasitic-aware circuit sizing. In Chapter 4, a g_m/I_D- and EA-based hybrid methodology for parasitic-aware circuit sizing is detailed. Chapter 5 illustrates an LDE-aware hybrid sizing methodology by employing the techniques of g_m/I_D, sensitivity analysis, and EA. In Chapter 6, a machine-learning-based method using Gaussian-process-based Bayesian optimization (GP-BO) is presented for the LDE-aware circuit sizing. Chapter 7 concludes this dissertation and discusses the future work. Our contributions in this dissertation are summarized in the introduction sub-section at the beginning of each of Chapters 3-6.

Chapter 2 Analog Design Automation, Challenges and Solutions

Electronic design automation (EDA) tools are computer-aided design (CAD) software specific to the electronics industry. They aim at reducing development effort and cost by allowing circuit/system designs to be simulated and analyzed before manufacturing. With the assistance of the EDA tools, the development period of an electronic system has been shortened a large extent. Usually the portion containing analog circuitry is smaller than the digital one in terms of silicon area in the modern System-on-Chip solutions. However, due to high complexity of analog circuits, the design of the analog/RF part stays as a bottleneck of the whole system design. Thus far, analog/RF circuitry has not largely benefited from the mature hardware description language synthesis flow as much as its digital counterpart. As a matter of fact, the analog/RF circuitry design is a creative and intuitive process that requires a clear understanding of circuit components and their matching requirements. Thus, it is knowledge intensive and complex in nature. It is often difficult to find a single solution that can satisfy all the analog constraints.

In addition, the newer technologies are associated with some drawbacks, such as strong impact of parasitics, short channel effects, interconnection issues, layout-dependent effects (LDEs), etc. Thinner interconnects may produce unwanted larger parasitic resistance, and closely spaced interconnects can cause an increase of parasitic coupling capacitance. Especially for the nanometer technologies, parasitic resistance and capacitance may drastically affect circuit performance [6]. Furthermore, the surrounding layout around a device might change the behavior of its fine-grained model constructed originally for an isolated state, which are referred to as LDEs [7]. Thus, the analog layout designers, although being aware, are often heavily burdened due to either lack of knowledge passed along from the schematic-level circuit designers or intricacy of handling parasitic and LDE constraints. In turn, a prolonged re-design cycle is typically expected since the incurred problems due to those effects may unfortunately not emerge until the final signoff check in the worst scenarios. In this chapter, Section 2.1 discusses the recent challenges of analog design automation including layout parasitics and LDEs. Then in Section 2.2, we will provide a literature survey of the solutions for addressing the challenges.

2.1. Challenges in Analog Design Automation

2.1.1. Parasitics

In electrical networks, a parasitic element is a circuit element (resistance, capacitance or inductance) that is a real existence although usually undesirable when an electrical component is laid out on the substrate. In relatively older technologies (i.e., CMOS 90nm and above), parasitic resistance is nearly a local effect where a conducting wire or via does not rely on the presence or absence of neighboring wires and vias. The resistance value is a function of the geometry and resistivity of the conducting material. For newer technologies (i.e., CMOS 65nm and below), because the resistivity depends on neighboring wires and vias, which is not constant by nature, the parasitic resistance is not a local effect any more. Parasitic capacitance exists when two closely placed conductors (e.g., neighboring electrical nets) conducting different signals, and the electric field between them leads to electric charge to be stored on them. For parasitic inductance, when a wire conducting current exerts magnetic field, this field is coupled to the current in another (or the same) wire, which produces voltage when the magnetic flux is changed by the current change. The first two parasitic elements are the main focus in this dissertation. Parasitics are normally either extracted by accurate but slow off-the-shelf layout extraction tools such as Mentor Graphics PEX

[8] or approximated by using parasitic models [9] [10]. Circuit performance degradation due to the layout parasitics has been a major issue as a result of shrinking feature size in the advanced CMOS technologies.

2.1.2. Layout-Dependent Effects (LDEs)

The behavior of a MOSFET is not only reflected by its performance model built in the isolated state, but also affected by its surrounding devices in the physical layout, which is known as LDEs. Typical LDE-incurred impacts include variations of MOSFET characteristics, such as mobility and threshold voltage (V_{th}) drifting, which might further degrade circuit performance. In this dissertation, two dominant LDEs are studied, which includes shallow trench isolation (STI) and well proximity effect (WPE). For STI, the shallow trench is formed during the process of transistor isolation by etching onto the wafer and filling with undoped polysilicon or silicon oxide (SiO_2) as isolation between active areas. This exerts mechanical force, which is a compressive stress applying to the vicinities, i.e., diffusion areas. This stress is commonly referred as STI stress, also called *Length of Diffusion* (LOD) effect, which improves the mobility of PMOS but decreases it for NMOS. As a result, STI can cause variations of mobility, saturation velocity, V_{th}, body effect, and drain-induced barrier lowering effect.

SA/*SB*, *STIW* and other STI-related parameters are illustrated in Fig. 2. *SA*/*SB* is a pair of distance parameters measured from the edges of each poly finger to its corresponding diffusion edges. For a layout netlist, each finger has its individual *SA*/*SB* pair. The width of STI (i.e., *STIW*) is measured from the edge of a device to its adjacent active area. A linear relationship between stress and layout information, *SA* and *SB*, is modelled by BSIM [11] as below,

$$stress = 1\,/(SA + 0.5 * L\,) + 1/(SB + 0.5 * L)\ . \tag{1}$$

The effect of this *stress* on mobility is modeled by BSIM [11] via (2)(3) below,

$$\frac{\mu_{eff}}{\mu_{eff0}} = 1 + \rho_{\mu_{eff}}\ , \tag{2}$$

$$\rho_{\mu_{eff}} = \frac{KU0}{Kstress\_\mu 0} * stress\ , \tag{3}$$

where μ_{eff} is the effective mobility after considering STI effect and μ_{eff0} is the one before that. $Kstress\_\mu 0$ is a function of many parameters from numerical models like $KU0$, and some of them are not completely disclosed. Therefore, such numerical STI models are hardly employed by designers in a symbolic form, which calls for the need of direct involvement of numerical simulation. The effect of STI *stress* on other device characteristics like saturation velocity can be found in [11] in a similar way.

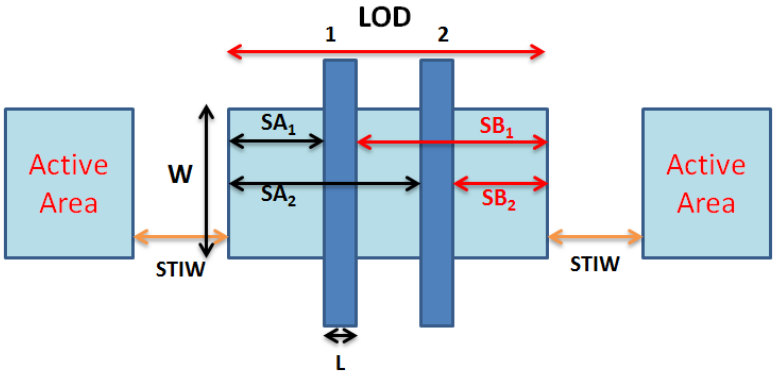


Fig. 2. Illustration of STI factors [5]

During the implantation process, some of the ions scattered from the edge of photoresist are implanted in the silicon surface near the mask edge, changing the threshold voltage of these devices by upwards of 100mV [12]. This effect is known as well proximity effect (WPE). The result of WPE is the formation of a graded channel due to a MOSFET placed too close to a well edge. This graded channel can cause the shift of electrical characteristics of the MOSFET. The WPE is a strong function of the distance of a MOSFET from mask edges (or well edges). The electrical parameters of the MOSFET due to WPE show larger variation if it has shorter distance from the edge of well mask. In short, WPE can cause variations of mobility, V_{th}, and body effect. As exposed in the BSIM model [11], they can be analytically expressed in the following,

$$\mu_{eff} = \mu_{eff_{org}} * (1 + KU0WE * (SCA + WEB * SCB + WEC \ * SCC) \ , \tag{4}$$

$$Vth0 = Vth0_{org} + KVTH0WE * (SCA + WEB * SCB + WEC \ * SCC) \ , \tag{5}$$

$$K2 = K2_{org} + K2WE * (SCA + WEB * SCB + WEC \ * SCC) \ , \tag{6}$$

where *SCA*, *SCB*, and *SCC* are instance parameters that represent the integral of the first/second/third distribution functions for scattered well dopants. They are functions of MOSFET geometric parameters. In most cases, the first order distribution parameter *SCA* dominates as it can already exhibit a reasonable level of accuracy. *SCB* and *SCC* are used when a fine tuning for the model is needed in order to match observed data for a wide variety of processes. $KU0WE$, $KVTH0WE$, and $K2WE$ are mobility degradation factor, threshold shift factor, and $K2$ shift factor respectively for WPE. *WEB* and *WEC* are just coefficients for *SCB* and *SCC* [11]. The modeling of *SCA*/*SCB*/*SCC* for WPE as well as *SA*/*SB* for STI will be further investigated in Chapter 5 for LDE-aware circuit sizing.

2.2. State-of-the-Art Analog Circuit Sizing Methods

The state-of-the-art circuit sizing works can be categorized into two groups: analytical (or symbolic-analysis) based and stochastic based techniques. Some may employ both techniques as a hybrid solution in their work. The analytic-based methods often require nontrivial modeling efforts on performance objectives and constraints, while the modeling accuracy may be controversial. Mathematical programming is often resorted to for this group. The solving efficiency for a modeled circuit sizing problem is normally pretty good, and the high reusability is one important advantage for the analytical-based methods. For the stochastic-based methods, usually a series of semi-random trial solutions are composed to compete with the existing solutions via a heuristic or statistical-based selection mechanism. The numerical simulation is often involved for this kind of method. In the rest of this section, the analog circuit sizing problem is firstly defined, and its layout awareness is also emphasized in Section 2.2.1. Then four main state-of-the-art sizing methods that are highly relevant to this dissertation will be discussed from Section 2.2.2 to Section 2.2.5. For the purpose of comprehensiveness, other layout-aware circuit sizing methods are also reviewed in Section 2.2.6.

2.2.1. Definition of the Analog Circuit Sizing Problem

Analog circuit sizing, usually referred to at the schematic level, is to determine device geometrical and electrical parameters, such as transistor width (W) and length (L), settings for resistors, capacitors and inductors as well as bias conditions. In the overall circuit synthesis, the sizing task takes place after topology generation/selection and is followed by layout design, which is mainly comprised of floorplanning/placement [13] and routing [14]. In the traditional process, once a circuit is sized by the schematic designers, the layout designers take over the ideal design towards the physical/layout design domain. However, they might suffer from repeatedly adjusting the layout due to complex layout effects. Therefore, the so-called layout-aware circuit sizing methods came into being to take into account layout parasitics and/or layout-dependent effects. These methods may adjust or even generate new device and circuit parameters, which make the circuit performance less vulnerable to layout effects. Some may also generate useful layout information including layout floorplans, wire length and width for various electrical nets, and other geometrical parameters regarding LDEs as guidance to layout designers.

2.2.2. Geometric Programming (GeoP)

The geometric programming (GeoP) based methods originate from an observation that a wide variety of design objectives and constraints are in the posynomial or monomial form versus design variables [15]. The geometric program is an optimization problem in the following form [15],

$$\begin{gathered} minimize\ obj(x)\ , \\ \text{subject to } f_i(x) \leq 1, \quad i = 1, \dots, p \\ g_i(x) = 1, \quad i = 1, \dots, m \\ x_i > 0, \quad i = 1, \dots, n \end{gathered} \tag{7}$$

where x_1,…, x_n are n real, positive variables, and the vector (x_1,…, x_n) is denoted as x, the objective function $obj(x)$ and constraints of f_1,…, f_p are posynomial functions, and equality constraints of g_1,…, g_m are monomial functions. The GeoP problem can be reformulated as a convex optimization problem and solved by a GeoP solver that uses standard interior point algorithm [16]. The GeoP-based methods are able to efficiently solve large convex optimization problems. If not solvable, certain constraints need to be loosened for reaching a resolution. Otherwise, a global-view solution is obtained.

The GeoP is originally applied to circuit sizing for a two-stage operational amplifier (Op-Amp) to optimize power and die area [15]. The device characteristics and circuit electrical constraints as well as other geometrical constraints are all modeled in the GeoP form, and solved by such a mathematical GeoP solver in order to finally solve for the sizing variables. Reference [17] exhibited a fast parasitic-aware synthesis approach, which considers the performance constraints and layout induced parasitics simultaneously within a concurrent phase of circuit synthesis. The GeoP-based sizing algorithm can include both device intrinsic parasitics and interconnect parasitics induced from layout floorplan. The analytic expressions of interconnect parasitic substrate and coupling capacitance can be further improved with the aid of the work in [18].

Another single-GeoP-process-based optimization [19] divides the design space into sub-problems by using piecewise-linear fitting instead of genetic-algorithm-based modeling in order to achieve accuracy improvement without compromising complexity. Given specific performance requirement and circuit topology, only a limited number of sub-spaces are needed and calculated rather than a costly blind search for all the sub-spaces. Without involving multiple GeoP execution for fine tuning, the optimization efficiency can be improved. However, a sound balance ought to be made between the GeoP process execution iteration and the knowledge-involved design effort regarding sub-space simplification. Zhang *et al.* [20] conducted an LDE-aware optimization in the schematic-level synthesis stage. Due to the employed square-law current equation involved in the GeoP formulation, its modeling accuracy is rather questionable.

2.2.3. g_m/I_D-Based Circuit Sizing

The g_m/I_D-based methods are built upon the theory that transconductance over drain current (i.e., g_m/I_D) is solely dependent on node voltages (e.g., V_{GS}) regardless of transistor sizes [21]. They have been recently promoted in the analog circuit retargeting and sizing domain [22]. The g_m/I_D amount can not only imply the electrical performance of analog devices, but also be used to derive transistor dimensions given the performance requirements. Jespers applied the g_m/I_D methodology to low-voltage analog CMOS circuits as a sizing tool [23].

Most of the early g_m/I_D works in the literature [21] [23] tackle analog circuit design as a manual sizing problem by firstly determining the slope factor and Early voltage. Then the designers' knowledge is involved in order to determine bias conditions, transistor operating regions, and g_m/I_D values or ranges. Based on a g_m/I_D table derived from numerical simulations, the transistor sizes can be finally obtained through a mapping process as per the g_m/I_D theory [24]. In contrast, Girardi *et al.* [25] applied a g_m/I_D method to automate the circuit synthesis problem through a simulated annealing (SA) based heuristic scheme, which replaces the manual input of designers' knowledge for the g_m/I_D estimation. Tlelo-Cuautle and Sanabria-Boron [26] combined the g_m/I_D and EA optimization to link g_m/I_D and transistor width (W) by using a lookup table (LUT) obtained via sweeping V_{GS} at a preselected transistor length (L). However, this work is short of identical *L's* for all the MOSFETs, and the decision of L's requires the involvement of designer's knowledge by plotting curves of g_m/I_D versus other device characteristics with different L's.

A normalized measure of I_D, called inversion coefficient (IC), was defined in the EKV model [27] to reflect MOSFET's inversion level, which is of importance in the g_m/I_D-based approaches. Binkley *et al.* [28] applied three independent degrees of analog CMOS design freedom (i.e., I_D, IC, and L), where IC links to W, DC biases and small-signal parameters including g_m/I_D and g_{ds}/I_D, as well as device intrinsic gain and bandwidth. So a performance tradeoff of single devices can be reached by exploring the combination of the above three parameters for the sizing of the whole circuit. However, due to the nonlinear relationship between device and circuit performance, designers' intervention and optimization iterations are still expected besides uncertain layout parasitic effects.

Aside from the abovementioned methods that use g_m/I_D concept as a sizing inference, there is a bias driven approach for g_m/I_D-based circuit sizing as another main stream group. Lin *et al.* [29] developed such a g_m/I_D-based sizing automation approach by utilizing a bias-driven LUT. An SA engine is used to try different bias conditions within a range restricted by a group of constraints in the device operating regions. Once a trial bias condition is generated, a small-scale LUT is built by sweeping MOSFET width for a reference device. Later this idea was further extended in [22] [30], where the transistor operating points are treated as variables. Analytic performance equations are formed for linear programming (LP) problem solving, which replaces the real simulation for improving the sizing efficiency. These works have largely extended the scope compared to the previous g_m/I_D research by including parasitic handling in the analog circuit sizing process. Nevertheless, only simplified linear parasitic models are utilized yet without the layout-effect consideration of the MOSFET multi-finger structure and floorplan constraints. Moreover, they are not able to fully operate on g_m/I_D, g_{ds}/I_D, and Meyer capacitance C over I_D (altogether called g_m/I_D-parameters thereafter in this dissertation), which are actually strong functions of V_{DS} and L in addition to V_{GS} for sub-100nm technologies [31]. However, this dependence is closely considered in our work, which is detailed in Section 4.3.

2.2.4. Evolutionary Algorithm (EA)

Evolutionary algorithm (EA) is a subset of evolutionary computation that is a generic population-based metaheuristic optimization algorithm. It is inspired by biological evolution and involves four main steps: initialization, genetic operation, selection, and termination. The middle two steps execute in an iterative manner before the termination criteria are met. A semi-random trial solution is recombined and evaluated to make the whole population (i.e., candidate solutions, EA chromosomes or individuals) evolve through smart selections in a heuristic manner. EA is widely applied to solve problems that cannot be easily solved in polynomial time, such as classically NP-Hard problems. Besides the universal application to the other fields, EAs of all kinds of variants have been applied to the circuit sizing domain. Even though the convergence for this type of stochastic-based algorithms (including EAs, genetic algorithms (GAs), simulated annealing (SA), particle swarm optimization (PSO), and other) is hardly proved, optimal solutions can be empirically found via a balanced exploration and exploitation during the course of solution space searching. Typical control parameters (i.e., genetic operators) including mutation and crossover rates determine the searching quality by balancing the weight between overall exploration and local refinement (i.e., exploitation).

Many works in the last decade of the 20th century dealt with topology selection and sizing together. Thus, overhead was inevitable when useless topology was generated. Authors in [1] claimed their CMOS OPAMP synthesis tool called DARWIN, using genetic algorithm (a subclass of EA), can simultaneously deal with topology selection and sizing. They translated circuit specification and constraints into certain representations used in their genetic algorithm in order to require less expert knowledge for circuit optimization. Their tool can cover different topologies in an efficient way. However nowadays, as the CMOS technology is scaling down, this tool may not be applicable due to lack in addressing many LDE issues (e.g., WPE and STI), which were not found prominent in the old days.

In evolutionary computation, differential evolution (DE) [32], which originated from Ken Price's attempts to solve the Chebychev polynomial fitting problem, optimizes a problem through iteratively trying to improve a candidate solution with respect to a given measure of quality. It makes few or no assumptions about the problem being optimized and can search a very large solution space. As one of the promising heuristic methods, DE is capable of evolving multidimensional real-valued variable vector for a function. It does not utilize the gradient of the problem which makes it superior to those classical optimization methods, such as gradient descent and quasi-newton methods, in terms of less dependency of the problem being differential [32]. DE is also extended to be used in the area of discrete, noisy and time-variant problem optimization.

There are two mutation schemes [32] that highlight the spirit of DE in the following. In the first scheme, for each candidate solution denoted by vector $\boldsymbol{X}_i$, $i = 0,1,2,\ldots,NP\text{-}1$, where NP is the number of population, a trial solution $\boldsymbol{T}$ is generated based on (8),

$$\boldsymbol{T} = \boldsymbol{X}_{s_1} + M * (\boldsymbol{X}_{s_2} - \boldsymbol{X}_{s_3}), M > 0 \, , \tag{8}$$

where $s_1, s_2, s_3 \in [0, NP\text{-}1]$ are randomly and mutually different integers and the mutation rate (M) controls the amplification of the difference between $\boldsymbol{X}_{s_2}$ and $\boldsymbol{X}_{s_3}$.The generation of $\boldsymbol{T}$ depends on $\boldsymbol{X}_{s_1}$,$\boldsymbol{X}_{s_2}$, and $\boldsymbol{X}_{s_3}$ instead of the current individual $\boldsymbol{X}_i$. In the second scheme, $\boldsymbol{T}$ is generated via (9),

$$\boldsymbol{T} = \boldsymbol{X}_i + \gamma * (\boldsymbol{X}_{best} - \boldsymbol{X}_i) + M * (\boldsymbol{X}_{s_2} - \boldsymbol{X}_{s_3}), M > 0, \gamma \in [0,1] \, . \tag{9}$$

The additionally introduced control parameter γ helps enhance the greediness by taking the current best solution $\boldsymbol{X}_{best}$ into account, which is especially effective for handling non-critical objective functions. The greatest dependence on the global best solution $\boldsymbol{X}_{best}$ is achieved when $\gamma = 1$.

Vancorenland *et al.* [33] extended the idea of analog circuit design from [1] to a new one involving tasks of circuit sizing and layout generation, in addition to parasitic estimation. The coupling of sizing and layout generation was made possible in the proposed layout-aware synthesis method, which contained DE-based optimization, cost function formulation, numerical simulation, and layout generation by using layout templates. The adopted Hooke algorithm in fitting the cost function was non-stochastic and thus contributed to faster convergence. The evaluation of the fitted cost function utilizes a mechanism, which combines few steps of model approximation with one simulation, in order to refine the model. According to the authors, this combined evaluation mechanism could largely increase the accuracy. However, this improvement is ensured at the cost of actual layout generation and detailed parasitic extraction.

Multi-objective (i.e., ≤3 objectives) evolutionary algorithms (MOEAs) are well-known for solving complex multi-objective problems (MOPs), which mean to include two or three objectives conflicting from each other. Pareto set or Pareto front (PF), is a set of nondominated solutions, being chosen as optimal, where no objective can be improved without sacrificing another objective. Among the first in the field of analog EDA, Aggarwal and O'Reily [34] brought forth the concept of spatial locality and dimensional locality, with which an analog/RF sizing problem is usually equipped. They built up an adapted sizing engine based on NSGA-II [35] and proposed a correlation sensitive mutation operator (COSMO). Moreover, they exploited the locality concept to enhance variable exploration capability with the aid of the circuit knowledge extracted from the first-order circuit performance equations or circuit sensitivity study. Nevertheless, no layout-related information was considered in that work.

Optimization problems with more than three objectives are commonly called many-objective problems (many-OPs), which most of the analog/RF sizing problems actually fall into. NSGA-II, as a typical implementation of MOEAs, is weak in handling many-OPs since a large number of solutions would be trapped in the first nondominated front, which leads to rich diversity but less exploitation capability. To address many-OPs, advanced many-objective EA (many-OEA) strategies have recently emerged. NSGA-III [36] stresses diversity more than convergence due to its less capability of attracting solutions towards Pareto Front (PF) in high-dimensional solution space, whereas MOEA/D [37] is able to approach PF quite well with its aggregation-function-based selection operator. However, without a smart control on the aggregation function, it might lose some valuable search regions. Therefore, by combining the merits of prevalent NSGA-III and MOEA/D, Yuan *et al.* proposed θ-DEA [38], which is able to outperform its peers in handling many-OPs. θ-DEA can not only preserve the diversity by maintaining the structural strength of NSGA-III, but also promote the convergence by employing the fitness evaluation scheme borrowed from MOEA/D.

2.2.5. Gaussian-Process-Based Bayesian Optimization (GP-BO)

With a clear emphasis on utilizing statistics to manage probabilistic models and uncertainty, machine-learning-based methods have been emerging as an important stream under the stochastic-based class. Most of the simulation-based stochastic approaches, which need little circuit knowledge compared to the symbolic-analysis-based methods, are usually called black-box optimizers. They typically suffer from longer runtime due to slowness of SPICE simulation. Recently, Bayesian optimization (BO) has appeared as a prevalent scheme for handling expensive black-box derivative-free functions. BO comprises two key components: probabilistic surrogate model and acquisition function. The surrogate model, which can play a role of any expensive objective function, is normally established by first using some random data observations. It would be trained and improved with new promising data points (called query points) along the BO iterations. Interactively, the acquisition function serves as a query-point generator by integrating the statistical characteristics (e.g., mean and variance) of the surrogate model. Specifically, a good query point with balanced force of exploration (i.e., with high uncertainty) and exploitation (i.e., with high confidence) is generated by minimizing or maximizing the acquisition function.

Gaussian-process-based Bayesian optimization (GP-BO) has been applied to the automated analog circuit sizing research [39]. However, multi-objective Bayesian optimization is only achieved by recovering the Pareto Front (PF) of objectives through weighted Tchebysheff formulation [39] or random scalarizations [40] instead of directly confronting the multi-objective problem. In [41], a direct multi-objective based GP-BO is proposed. However, it merely simultaneously evaluates multiple popular acquisition functions when selecting query points for the next BO iteration. The objective value is calculated by using a user-defined figure of merit (FOM) expressed in a summation form with weighting factors for various circuit performance aspects. In this regard, we deem this as a "pseudo multi-objective BO", because the lumped FOM cannot necessarily loyally reflect the multi-objective nature.

Besides that, the GP-BO scalability versus problem input space creates another barrier to the applicability for a variety of problems. Successful applications of GP-BO are typically found for the problems with low input dimension, i.e., less than 20 optimization variables [42]. References

[39] [41], which have not taken into account any layout effects (i.e., *nf*, parasitics, and LDEs) in the advanced technology nodes, might end up with impassable difficulties when optimizing a large number of sizing variables (e.g., involving LDE parameters).

The framework of Gaussian-process-based Bayesian optimization (GP-BO) will be illustrated in Section 6.2 with more details. And our solution to overcoming the two difficulties when applying GP-BO to the optimization of analog circuit sizing is described in Chapter 6 as well.

2.2.6. Other Circuit Sizing Tools

In addition to the works reviewed in the above sections, other circuit sizing tools in the literature are developed for distinct purposes, which can be categorized into two big groups, i.e., using stochastic-based and non-stochastic-based techniques. Statistical and/or heuristic processes are often involved in the first group. The second group can be further categorized into two sub-groups including pure symbolic-analysis-based (often involving mathematical modeling and programming) technique and gradient-based error-minimization-directed optimization technique. These works could suffer from using inaccurate device and parasitic models, insufficient or no layout considerations (parasitics and LDEs), or overwhelming involvement of placement and layout (e.g., massive layout numeration using costly off-the-shelf layout extraction tools), and/or over-simplified strategies for dealing with high-dimensional variable space and multiple circuit performances as well as neglecting the importance of taking into account domain knowledge from analog circuit.

For the stochastic-based approaches, according to Rutenbar [43], simulated annealing (SA), which is a statistical and heuristic process, uses either some numerical cost functions or circuit-level simulation for verification. Design knowledge based optimization is usually integrated with such a heuristic technique to improve exploration efficiency. For instance, De Ranter *et al.* [44] presented a specification-driven layout-aware CMOS RF design tool called CYCLONE. They used adaptive simulated annealing (ASA) package as their search engine. A thought similar to [33] is that the circuit sizing and layout generation are combined for the optimization of oscillators. This tool includes three major components, the optimization startup, the optimization loop using electromagnetic simulation, and the layout generation. The design configuration file and technology layout file are inputs of the layout tool to form leaf cell branches, which are used as the building blocks to the final layout. The use of parameterized leaf-cell-based design method facilitates parasitic estimation in each layout generation step. The use of technology-independent template-based layout generation decreases the effort of generating redundant physical layout as that in [33].

Agarwal *et al.* [45] illustrated the importance of including layout information in circuit sizing by comparing the deviation in performance with and without parasitic consideration. Their core engine to size the circuit is SA. An integrated circuit sizing method with floorplan variation plus simulation for performance evaluation was introduced in [46]. At each step a floorplan is generated and parasitics are estimated using the floorplan and transistor sizes. Several floorplans are considered for performance evaluation. Once a floorplan is selected, a layout is then generated, extracted and verified. If the specification is not met, the loop would be executed again. Therefore, this simulation-based method tends to be CPU-time costly.

For the non-stochastic methods, Ranjan *et al.* [47] proposed a slightly different approach, which uses a symbolic performance models (SPM) generated by using equations from small-signal models. The SPM is used as the evaluation method instead of real simulation in the sizing process.

Due to the integration of intelligence into performance evaluation (by using symbolic cost function), this work can be grouped into the symbolic-analysis-based techniques. Another work from Agarwal and Vemuri [48] used a similar sizing engine, but put more emphasis on the estimation of layout parasitics in RF circuit synthesis considering worst-case corners.

Schwencker *et al.* [49] proposed an automatic sizing method for analog integrated circuit. They introduced structural constraints as circuit knowledge in the sizing process. Their sizing algorithm is based on linearization with sensitivity coefficient and gradient-based method for better convergence. The authors claimed that considering structural constraints could reduce design parameters and cut down simulation time, as well as being insensitive to process variation. Dessouky *et al.* [50] proposed a trial-and-error based method by using a tool called COMDIAC, which applies the equations already defined from the detailed knowledge of a circuit. At each step, a layout tool is called multiple times to generate parasitic estimation and a circuit sizing tool responds to the estimated parasitics by changing transistor sizes. However, this method may have problems on loop termination in newer CMOS technologies since the parasitics may deviate a lot even for a small change in sizing.

Habal and Graeb [51] proposed an automatic layout-driven synthesis flow. Their sizing steps include partitioning the problem into sub-problems by using linearized approximation of constraints and specification with respect to design parameters in a manner identical to [52], and then solving the sub-problems by using a modified trust-region algorithm. The whole work even includes SPICE simulation for evaluation, parasitic capacitance extraction by an integral equation field solver, and placement optimization with B*-tree representation. In some optimization approaches, the designers' knowledge is imperative to continue the sizing process, which is based on a deterministic algorithm introduced in [53]. The entire synthesis process was arranged at the cost of additional effort in layout exploration and extraction. Overall this work is quite comprehensive in the circuit synthesis domain, although it tends to experience costly layout generation for every sized alternative.

Another deterministic algorithm in [54] was proposed to consider process variation in the automated design of analog circuits that include mismatch-sensitive components. With respect to the consideration of manufacturing and operating variation, Schwencker *et al.* [53] proposed a generalized boundary curve (GBC) to decide the step length within an iterative trust-region optimization algorithm. Applying the nonlinear cost function on the linearized objectives can largely cut down the iteration number during the optimization.

A sizing approach by using combined techniques was proposed in [55]. In this work, a transistor-level simulator (HSPICE) is used with simulated annealing technique for the first phase of sizing. In the second phase a deterministic method is used. Template-based layout generation, which takes a few seconds to generate layout, is deployed along with Cadence PCELL and SKILL programming language [56]. At first the sizing engine selects a set of random design values within a range. This set of values is used by the geometric constraint module (GCM) to generate a number of candidate layout styles. One candidate layout is selected as per some constraints. Then the parasitics are extracted from the selected layout and the performance is evaluated with some layout awareness. If the specification is not met, the loop is executed again. Therefore, the computation cost from layout enumeration under the various constraints can still be moderate or high.

2.3. Summary

In this chapter, we have first discussed the main challenges (i.e., layout parasitics and LDEs) of analog/RF circuit synthesis in the advanced technology era. Since those issues are time-consuming to be fixed in the later layout design stage with respect to the whole synthesis flow, it is the responsibility of the EDA tools to consider them in an earlier stage (i.e., schematic design stage) so as to make the designed layout less subject to layout-effects-induced performance degradation, and therefore the whole synthesis process can speed up. In addition, the previous works relevant to the abovementioned issues have been reviewed with their advantages and limitations pointed out.

In the next chapter, our proposed efficient parasitic-aware hybrid sizing methodology will be detailed. It will not only address the efficiency challenge which the simple stochastic-based methods would often suffer from, but also resolve the accuracy issue that the plain symbolic-analysis based methods may normally experience, meanwhile, considering the performance-related parasitics for the analog/RF circuit synthesis.

Chapter 3 Efficient Parasitic-Aware Hybrid Sizing Methodology for Analog and RF Integrated Circuits

3.1. Introduction

In order to generate a quality-guaranteed tape-out as the objective of CMOS design, diverse analog integrated circuit synthesis flows have been proposed to address the drawbacks of the traditional iterative design flow that may meet performance requirements but are super time-consuming. As the primary second-order effect, parasitic issues have to be seriously addressed when synthesizing high-performance analog and RF integrated circuits. For a new circuit structural design in the advanced technology, estimated pre-layout parasitics (i.e., before the actual circuit layout is available) by using a stereotypical method may have large deviations from the real or extracted ones by using off-the-shelf parasitic extraction tools. This is especially true for the synthesis process of high-performance analog and RF ICs where circuit sizing needs to be first identified before the formation of the corresponding layout.

By extending our preliminary work [57], in this chapter, we have proposed a complete parasitic-aware GeoP-EA (geometric programming plus evolutionary algorithm) hybrid sizing method. We not only include floorplan optimization, GeoP modeling, and theoretical investigation on floorplan and interconnect parasitic modeling in a GeoP compatible way, but also explore performance enhancement by integrating single-objective evolutionary algorithm (SOEA) and many-objective evolutionary algorithm (many-OEA) together for the sizing problems. Compared to the existing schemes aforementioned, our proposed parasitic-aware GeoP-EA hybrid sizing method in this piece of work has the following notable advantages:

- It is an effective combination between GeoP and EA sizing optimizations. The GeoP-phase sizing process is fast and ensured to output a global optimum, if feasible [15]. This gives the simulation-based EA-phase sizing process an elite starting point with implied circuit knowledge to help exploration convergence.
- By using device and interconnect parasitic models as well as floorplan symbolic constraints backed by our proven theorem, the proposed method provides more holistic parasitic estimation much faster than any actual layout generation or procedural layout generators typically used in the conventional nested sizing-layout loop.
- Rather than by providing a pre-defined fixed floorplan template, the integral floorplan selection is conducted by an SA-driven engine with B*-tree representation [58] according to the constraints and objectives of a specific circuit.
- To the best of our knowledge, this is the first work that applies many-objective EA in the analog circuit sizing domain, where the single- and many-objective EAs can switch as an optimization refiner. We also propose a scheme on experimental design and analysis of single- and many-objective EAs for optimizing engineering problems.

The research conducted on this topic has been mainly published in Integration, the VLSI Journal [J4], and presented in IEEE/ACM 22nd Asia and South Pacific Design Automation Conference (ASP-DAC) [C4] among others [C5].

3.2. Proposed Parasitic-Aware Hybrid GeoP-EA Circuit Sizing Flow

Our proposed method within the sizing flow as shown in Fig. 3 is a two-phase hybrid sizing optimization process. A convex optimization formulation called *GeoP* is used in the first phase to incorporate a set of performance constraints formed by the given technology parameters and required specifications, as well as a set of symbolic parasitic expressions modeled by geometric requirements and floorplanning constraints. A GeoP solver is deployed to provide a solution considering layout-induced parasitic effects [15].

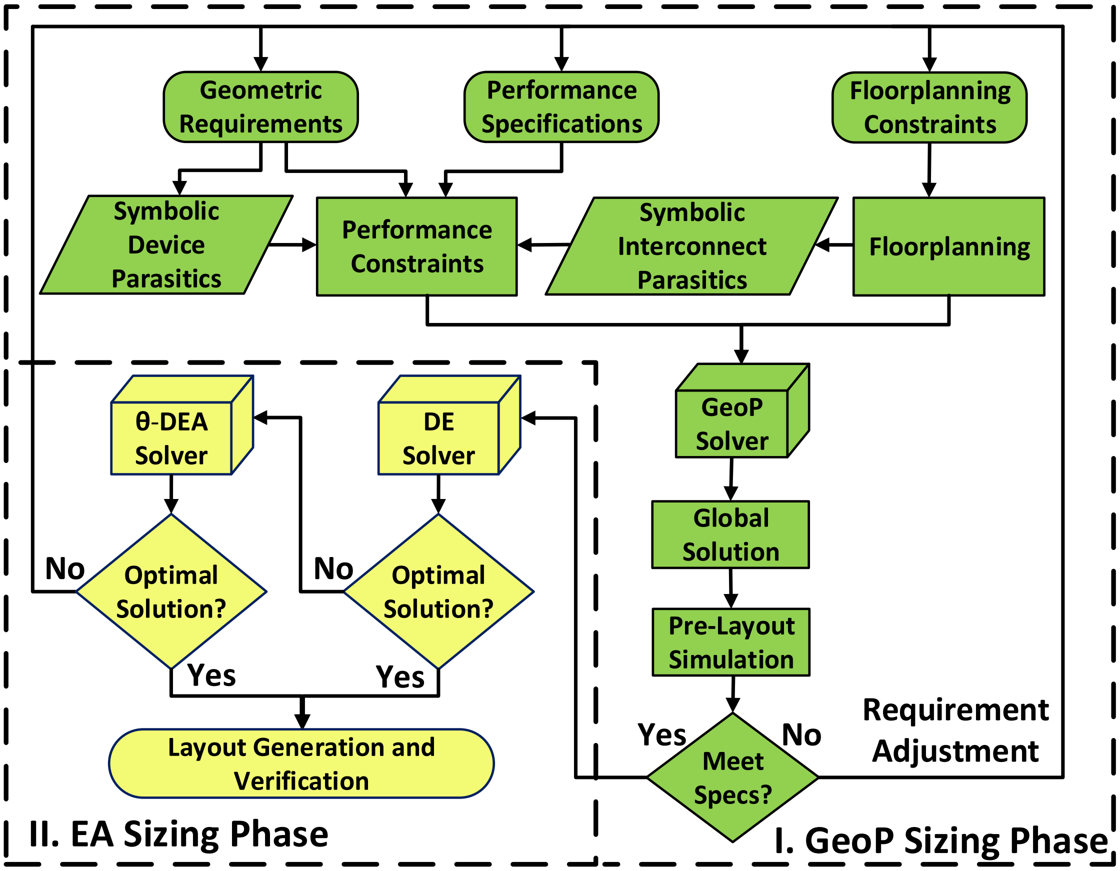


Fig. 3. The proposed GeoP-EA two-phase hybrid sizing flow

After that, this GeoP solution provides an initial point with implied variable ranges as circuit knowledge to the second phase that involves EA solvers along with proper parasitic estimation. EA can optimize to derive a solution by iteratively improving the candidates with respect to a given measure of quality. Any commercial simulator (e.g., Spectre or HSPICE) can be involved in the EA process by returning simulation performance. Such numerical simulations would definitely help ensure the accuracy of the solutions that might be a controversy for the GeoP modeling. The second sizing phase (EA-based) remains parasitic-aware because it follows the parasitic modeling used in the first sizing phase (GeoP-based) and reflects such estimated parasitics through a circuit netlist.

As shown in Fig. 3, two sets of constraints are formed for GeoP solving: geometric floorplan constraints and performance constraints, which rely on the required specifications and internal device parasitics as well as interconnect parasitics, both in a symbolic form. Since a floorplan optimizer normally requires a set of device sizes as input, in this work we deploy a standalone parasitic-free GeoP sizing process [15], which does not include any parasitic consideration, to generate a sizing solution as initial input to the subsequent floorplan operation. Then we run an SA-driven placement algorithm with B*-tree representation [58] to generate multiple sound floorplan candidates, among which the best floorplan (called *floorplan template* hereafter in this dissertation) is selected. The selection criteria include reasonable signal flows, resemblance to circuit schematic, and implementation constraints and objectives (e.g., matching and area). Afterwards, this floorplan template would be loyally preserved in the subsequent parasitic-aware sizing optimization. Compared to the work in [59], our method can effectively resolve the convergence problem caused by inconsistent parasitics from varying floorplans as exhibited in our experimental results of Section 3.5.

After a successful run with the GeoP solver, a derived sized circuit (called *Global Solution* as shown in Fig. 3) can be verified in any circuit design environment (e.g., Cadence) for pre-layout simulation. If the GeoP solver is not able to find a global solution or the pre-layout simulation fails to meet the due specifications as promised by the GeoP solver, the requirements of the GeoP formulation need to be adjusted by modifying the applied constraints and parameters. This first turning-back design can largely save the design cycles by avoiding passing invalid solutions on. Once verified, the sized circuit associated with the optimized floorplan is then passed as an elite solution to the EA solvers for further refinement. In the worst case where the possible trials including constraint relaxation and parameter tweaking for GeoP resolving have been exhausted, the second EA sizing phase should take over. Because it might be difficult to seek a global GeoP solution even with the most relaxed constraints, especially for sizing a circuit with strong nonlinearity or concave configuration space, EA-based sizing should take over in time rather than wasting CPU resources for model tweaking inside the GeoP sizing phase.

In the second phase, the EA solvers involve repeated pre-layout simulations along with dynamic estimation of interconnect parasitics by using the same parasitic modeling as the first sizing phase (GeoP-based). The device parasitics, which do not need to be symbolic, can be included from the foundry technology model files into the simulation netlist. Firstly, a fast DE search based on the GeoP elite output is executed. If the best solution obtained from the DE solver fails to meet the specifications, the optimization automatically switches to the many-OEA solver. The many-OEA solving would be integrated with the GeoP-derived circuit knowledge, which includes one definite GeoP elite solution and implied information on shrinkable variable ranges. If even the sophisticated many-OEA solver fails the sizing task, a reiteration from the very beginning of the flow is expected.

The GeoP-based and EA-based hybrid sizing will be further elaborated on in Section 3.3. Below several EA terms are introduced for clarity purpose. The maximum number of generations (G_{max}) indicates the depth of evolution. Inside each generation, all the individuals constitute a *population* with its size denoted by *NP*. The chromosome of an individual is represented by a variable vector, which is called *chromosome-vector* throughout this chapter. Each element within the chromosome-vector is named as *chromosome-variable*.

3.3. Sizing with Geometric Programming and Evolutionary Algorithms

3.3.1. Geometric-Programming-Based Sizing

Even though a form of statistical or deterministic algorithms is adopted for sizing in most of the layout-inclusive synthesis works surveyed in Section 2.2, in practice this optimization process is very time-consuming due to large search solution space especially in the context of lack of sound domain knowledge for the starting point. A wide variety of design objectives and constraints have a special form, which is called posynomial functions of design variables [15]. This has motivated us to apply GeoP to take both performance constraints and floorplanning constraints simultaneously to quickly determine a first-level global optimal solution. It would effectively help reduce the intensive calling of computationally expensive commercial simulators or layout generators. Aggarwal and O'Reilly [34] developed an algorithm to automatically generate posynomial models for MOSFET parameters by utilizing pre-layout simulations based on genetic algorithm and quadratic programming. Thanks to the alleviated modeling efforts, GeoP can equip a highly efficient circuit sizing optimization method for quickly outputting a global-view solution.

For a target circuit, the performance constraints like open loop gain, unity gain bandwidth, symmetry, matching, etc. can be modeled either in a monomial or posynomial form, which is applied within the block "Performance Constraints" in our proposed sizing flow in Fig. 3. For common analog structures, these expressions are normally available in the literature. More complicated analog structures can be partitioned into several smaller blocks, for which expressions can be derived by using symbolic analysis [60]. In addition, a floorplan can be constructed by using device sizes, minimum allowable distances between geometries, and matching constraints among different components. All these constraints are represented in the form of equations/inequalities used by the GeoP solver in the GeoP-based first sizing phase. If the performance or parasitic constraints are found to be in a complex form, which cannot be expressed as a monomial or posynomial form, such as

$$f_i(x) = \frac{ax^b + cx^d + \cdots}{pq^r + lm^n + \cdots}\,, \tag{10}$$

they can be formulated by using temporary variables as follows:

$$\begin{gathered} ax^b + cx^d + \cdots \leq temp_1\,, pq^r + lm^n + \cdots \leq temp_2\,, \\ \frac{temp_1}{temp_2} = f_i(x)\,. \end{gathered} \tag{11}$$

3.3.2. Differential-Evolution-Algorithm-Based Sizing

In this work, the compound EA solvers are composed of single-objective DE and many-objective θ-DEA. As a population-based stochastic function minimizer, DE is capable of evolving multi-dimensional real-valued chromosome-vectors by using a fitness function for evaluation purpose, as shown in (12) utilized in our work:

$$F(X) = \sum_{i=1}^{t} \alpha_i \cdot \frac{1}{R_i} \cdot S_i + \sum_{i=t+1}^{n} \alpha_i \cdot \frac{1}{S_i} \cdot R_i + \beta_1 \cdot A_{norm} + \sum_{i=2}^{m} \beta_i \cdot GR_{i(norm)} \text{ ,} \tag{12}$$

where α_i's and β_i's are the user-defined weighting factors for different electrical specifications and geometric requirements, respectively. R_i is the resultant value returned from numerical simulations and S_i is the corresponding defined specification. The first two terms on the right side of (12) show reciprocal division between R_i and S_i. In this way, if minimizing $F(X)$, the resultant value, R_i (i = 1 to t, such as open-loop gain), can be maximized, whereas R_i (i = t+1 to n, such as noise figure) can be minimized. Thus, both maximization and minimization of multiple objectives are integrated into one single-objective minimization problem of $F(X)$, where n is the total number of electrical specifications, and m is the total number of geometrical requirements. A is the normalized layout total area and GR_i's are the normalized other geometric requirements, which can be weighted by β_i's (i = 1 to m).

In this work, the uniform initialization is adopted to help the inclusion of any potential optimal solutions right from the first generation. The mutation is implemented by a currently-best-individual-based greedy scheme to favor a strong desire for local exploration. After evolutionary recombination operations, some chromosome-variables may not stay inside their originally defined bounds. A naive way to solve this is to set all cross-border data to either the upper bound or the lower bound. However, this may experience a weakness of big n-to-1 correspondence that those out-of-bounds solutions are all mapped to the boundary values, and so it may degrade the efficiency of genetic operations. Therefore, we propose a special clamping scheme as shown in Algorithm 1 based on modulo operation, which can also support floating-point chromosome-variables. Line 2 converts all floating-point parameters to integer values where N is the smallest integer to achieve this purpose. Line 3 derives a positive interval starting from 0. In Line 5, out-of-bound infeasible solutions are mapped back to valid solution space since they may also contain useful genetic information from parental generation. Furthermore, another technique [61] adopted in our DE scheme is its self-adaption capability of the control parameters, mutation ratio δ and crossover probability σ in (13),

$$\delta = \delta_0 * e^{-2(sumWeight/bestF)} \text{ , } \sigma = \sigma_0 * e^{-2(sumWeight/bestF)} \text{ , } \tag{13}$$

where δ_0 and σ_0 denote the initial values of δ and σ respectively, $sumWeight = \sum_{i=1}^{n} \alpha_i + \sum_{i=1}^{m} \beta_i$ whose symbols are defined in (12), and *bestF* is the currently best fitness amount. If the specifications are just met while evaluating the best chromosome-vector, namely $R_i/S_i = 1$ or $S_i/R_i = 1$, as well as $A_{norm} =1$ and $GR_{i(norm)} = 1$ in (12), *bestF* should be the same as *sumWeight*. The exponential terms would be valued smaller if better solutions appear with smaller *bestF*. So the new δ and σ would be much shrunk compared to δ_0 and σ_0. That is to say, the better the performance turns, the more δ and σ shrink and thus favor local refinement to help the evolution converge.

Algorithm 1. Variable clamping

Input: floating-point interval [b_1, b_2] and the varaible a that might be out of the interval

Output: clamped a within [b_1, b_2]

```
if ($a$ is outside of the interval [$b_1$, $b_2$])
    Convert $a$, $b_1$ & $b_2$ to integers by $a' = 10^N * a$, $b_1' = 10^N * b_1$, and $b_2' = 10^N * b_2$;
    Shift [$b_1'$, $b_2'$] to [0, $b_2'$ - $b_1'$];
    $r = a'$ mod ($b_2'$ - $b_1'$);
    return ($b_1'$+$r$) / $10^N$;
end if
return $a$;
```

SOEAs might perform well if the settings are properly configured for solving a group of similar problems. This configuration requires knowledge on the problem itself, e.g., variable boundaries, algorithmic control parameters, or evolutionary selection schemes. Such knowledge demands uncertain trials for achieving a good understanding in the landscape of solution space, which can help drive the evolution smoother. However, it has less value if the problem changes. In addition, in order to deal with multi- or many-objective tasks, SOEAs need a balanced weighting consideration to lump various objectives together in the fitness function (12). If multiple solutions with different emphases on distinct objectives are demanded, repeated executions with different settings have to be resorted to. Each successful run may require certain knowledge learned from initial trials. Therefore, an SOEA without special handling may not be able to complete the multi- or many-objective optimization tasks especially in the context of computation effectiveness.

3.3.3. Theta Dominance-Based Evolutionary Algorithm

θ-DEA [38] inherits the framework of NSGA-III, including the most important idea of cluster or niche. Firstly, N clusters are composed according to N systematically-distributed reference points that are dependent on the number of objectives and division parameters. After the initialization of the first generation, the simulated binary crossover operator is performed to generate candidate offspring (i.e., next-generation) solutions from the recombination of their parental (i.e., current-generation's) solutions. The objectives (i.e., circuit performances in our context) are evaluated from the function evaluation (i.e., circuit numerical simulation in our case). Nondominated sorting across each generation can then be performed to filter out inferior solutions.

In order to utilize the reference points, the m-dimensional objective space has to be normalized by using the objective values, nadir points and the best points found so far all from the current children population. The objective values from function evaluations and the best points from records are already known, while the current nadir point is solved via the following steps. Firstly, the extreme points along each objective axis are solved by minimizing an achievement scalarizing function that takes into account the current best points and the last nadir points. Then the obtained m extreme points are used to construct an m-dimensional linear hyperplane. Lastly, the intercepts that are the target current nadir points are obtained by having the linear hyperplane intercepted with each objective axis. Thus, the m-dimensional solution space of a complicated real problem is mapped to a normalized unit solution space with the same dimension, where the systematically pre-composed reference points can take effect.

Within the normalized objective space, N reference lines can be constructed between the origin and each reference point. Then each normalized objective vector can be projected to the reference lines, and two distance values are obtained, $d_{i,1}$ being the distance to the corresponding reference line and $d_{i,2}$ being the one between the projected vector and the origin, where $i \in \{1,2,\ldots,N\}$. The best diversity is achieved when $d_{i,1} = 0$ thanks to its perfect alignment with the reference line, whereas better convergence is achieved with a smaller $d_{i,2}$ under $d_{i,1} = 0$. Then the balance of exploitation and exploration can be controlled by $d_{i,1}$ and $d_{i,2}$ included in F_i (x),

$$F_i(x) = \theta * d_{i,1} + \ d_{i,2}(x) \ , \tag{14}$$

where θ is a user-defined factor further controlling the balance. The regular nondominated sorting with respect to objective values will be modified to regard F_i (x) (so-called θ-dominance), which is the main difference from NSGA-III. With F_i (x) for θ-dominance based selection, the next population can be prepared by selecting members from currently available clusters, and the evolution carries on until the termination condition is satisfied.

To improve the θ-DEA performance for resolving the analog/RF sizing problem in this dissertation, we have proposed the following modifications to the original algorithm of [38]. First, the uniform initialization and clamping scheme are adopted as discussed in Section 3.3.2. Then regarding our selection scheme for the new generation, when the number of the remaining slots for composing the next generation is smaller than that of the clusters available, we opt to give selection priority to those with the smallest cluster members in order to preserve diversity all the way down to the end, rather than randomly selecting members from the available clusters.

For many-OPs, there are two metrics when comparing performance: set coverage (called C-metric), and inverse generational distance (IGD) for representing the distance from representatives in the PF (called D-metric) [37] by using (15),

$$\mathrm{IGD}(Q, P^*) = \frac{1}{|P^*|} \sum_{q_i \in Q, o \in P^*} min_{i=1}^{|Q|} d(q_i, o) \ , \tag{15}$$

where Q is the nondominated points in the solution space as an approximation to the ideal Pareto Front (PF), P^*. IGD expresses the convergence of set Q by calculating the average Euclidean distance $d(q_i,o)$ between each point q_i in Q and each member o in P^*. However, in the analog/RF sizing problem, discovering all the optimal solutions and determining such an ideal PF is even harder than solving the sizing engineering problem itself. Therefore, a highly nondominated optimal set has to be formed as the known best representative for the ideal PF. Such a representative pseudo PF, P_{pseu}^*, would replace P^* in (15) for this work. A solution pool, which is composed by all the specification-passing non-duplicate solutions from various optimization methods running on the same problem, is maintained in our experiments. By applying a nondominated sorting to the solution pool with a predefined size n, the solution pool would be always refined to an n-size nondominated set as the updated P_{pseu}^{*}, which is a resource-free side product when we study different optimization methods in our experiments. Naturally, it is the currently best solution set for the analog/RF sizing problems themselves versus any others derived from each single optimization method. In this regard, our challenge is actually different from the conventional many-OPs research in the area of computer science whose target problems are normally certain well-defined mathematical problems with definite PF.

The IGD metric is supposed to be used for comparison among different MOEAs if the ideal P^{*} is known in advance, e.g., the benchmark test cases used in computer science [36] [38]. However, in the engineering applications, it might provide misleading clue if the IGD is employed as the sole metric for comparison among different methods because P_{pseu}^{*}, only an empirical subset of P^{*}, may not include all the PF segments. In any practical experiments, usually limited runs are conducted for collecting statistics data. Since it is not easy to judge which solution should be used as a representative for reporting data, IGD can help serve as a criterion for evaluating optimization quality for one scheme. Therefore in our experiments, we choose the run with the median IGD to represent the scheme for comparison as reported in Section 3.5.

3.3.4. Sizing with Hybrid Evolutionary Algorithms

It is beneficial to exploit the merits of each adopted EA method in a unified optimization flow. A regular implementation of DE seems incompetent in resolving certain hard problems due to the weakness of SOEA. Nevertheless, DE features several advantages, including its implementation simplicity and good time efficiency compared to the others [62]. These features have contributed to its continuous popularity in the most recent applications even for handling MOPs [63] [64].

As discussed in Section 3.3.1, GeoP features fast access to global optimum along with the convenience of easy integration with device parasitics, interconnect parasitics, geometrical and performance constraints. The first-level optimum solutions from the GeoP phase may be relaxed in terms of accuracy requirement in our analog/RF sizing problems. As long as the GeoP output can facilitate the evolution process in the subsequent EA optimization phase, we believe that the GeoP phase is helpful to be integrated due to its merit of a tiny footprint. Through exploration around the GeoP elite output, the solution space in the following DE optimization can be largely trimmed due to its inherent nature of strong focus but less diversity. Our experiments exhibit that this GeoP-DE combination cooperates very well for the problems with less complex solution space.

The major challenge, which our *θ*-DEA scheme is aimed to address, is to harvest multiple clusters where diverse optimal solutions are located for the multi-dimensional complex problems, although at the cost of computation time. Its population size, *NP*, has to be set to be no less than the number of reference points, which relies on division parameter. It might be fine to select a smaller division parameter to still enjoy a good coverage of solution space for a less complex problem. However, this might lead to a loss of many optimal regions for a hard problem featuring widely distributed optima. On the other hand, a larger division parameter may lead to time-consuming evolution that is actually unnecessary for a less complex problem. So it is not always worthwhile to solve the analog/RF sizing problems by just using many-OEAs. Instead, we can first apply GeoP-DE and then continue with many-OEA if really needed. In this way, GeoP can help shrink the solution space by providing trimmed chromosome-variable boundary more reliably than any aimless random reduction. Therefore, at least one GeoP elite solution is preserved and exploited in the DE evolutionary process, and/or one cluster around the GeoP elite solutions would be discovered in the θ-DEA optimization process.

In our proposed optimization flow, DE just functions as a trial toolbox that may quickly yield a solution to meet the specifications if feasible especially for relatively less complex problems, whereas θ-DEA would take over to further deal with harder problems by exploring multi-dimensional clusters. To sum up, our proposed three blocks, GeoP, DE, and θ-DEA, can play their distinct roles complementarily in the hybrid optimization flow with increasing accuracy at the cost of CPU time. Although none of them is perfect, our proposed method can choose the right combination to deliver efficient search based on the complexity of the actual sizing problems.

3.4. Parasitic-Aware Sizing Methodology

3.4.1. Floorplan Generation

An important feature of our proposed sizing methodology is the inclusion of layout effects to be considered among the sizing constraints. To incorporate sensible calculation of layout-induced capacitance and resistance in both sizing phases, the circuit is floorplanned by using an SA-based engine with the B*-tree representation (as illustrated in Section 3.2). The output floorplan template, which meets the specific constraints, e.g., symmetry, matching, signal flow, and user-defined topology requirements, would be sustained in the following sizing optimization. Moreover, our global routing scheme starts with recording interconnect pins and collecting obstacle regions (e.g., transistor devices) based on the generated floorplan. Then a fast lookup-table-based rectilinear Steiner minimum tree algorithm, FLUTE [65], is employed to generate a minimum-Steiner-tree (MST) path for each electrical net. Based on the formed MST routing paths, the interconnect segments can be symbolically expressed in terms of the device geometric parameters and technology design rule factors. As a result of deploying the tractable floorplan template, the vulnerable parasitic impact can be effectively made up by tuning device sizes or bias conditions in the sizing process. So without compromising the optimization resolution, the search solution space can be significantly reduced.

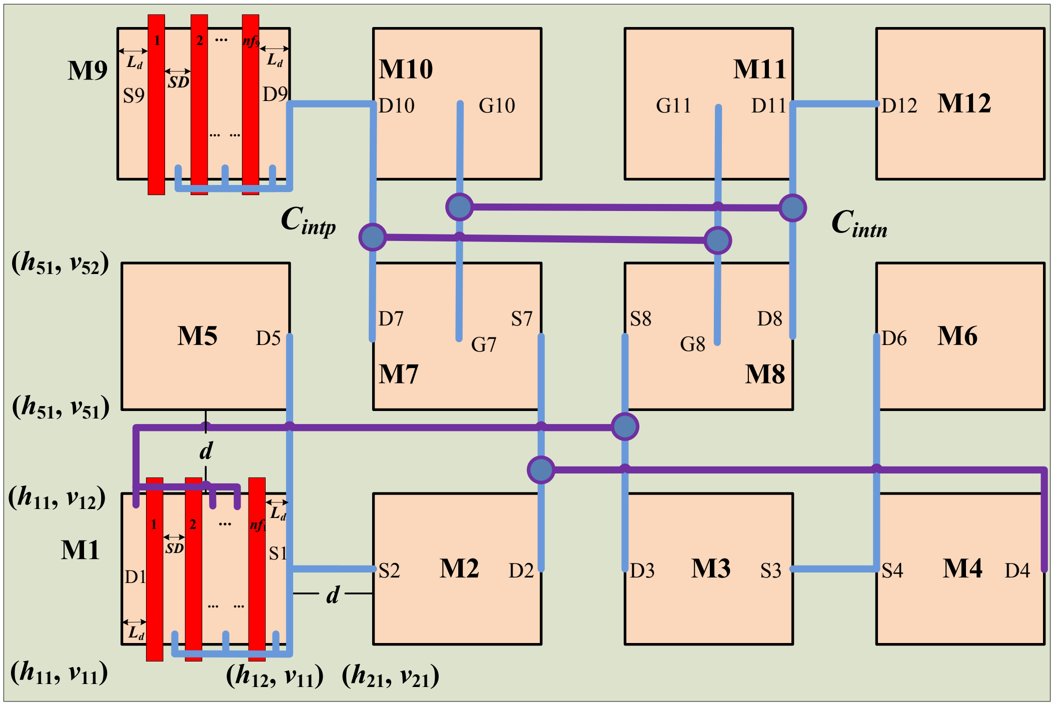


Fig. 4. One floorplan of the differential-pair comparator

Based on the floorplan template, each device capacitance and interconnect capacitance & resistance are modeled as a set of symbolic layout constraints. To mitigate the layout-induced mismatch, the symmetry requirements are put inside the floorplanning constraints. For example, one floorplan template of the differential-pair comparator in Fig. 5(b), is shown in Fig. 4 with the presence of interconnects. The transistors M1 & M2, M3 & M4, M7 & M8 and M9 - M12 are placed symmetrically to avoid parasitic mismatch, which is also visible from the floorplan. Floorplanning constraints are formulated to minimize the total area. Cartesian coordinates are used to denote the position of devices. Special separation requirements between two adjacent devices can be reflected by user-defined spacing constraints or simply from technology-dependent design rules. Relative geometric positioning constraints are added to avoid overlap of devices. For instance, the following inequalities can be formulated for the transistors in Fig. 4,

$$\begin{gathered} w_{mi} + 2 * polyExt + v_{i1} \leq v_{i2} \,, \quad d + v_{i2} \leq v_{j1} \,, \\ l_{mi} * nf_i + (nf_i - 1) * SD + 2 * L_d + h_{i1} \leq h_{i2} \,, \\ d + h_{i2} \leq h_{j1} \,, \end{gathered} \tag{16}$$

where w_{mi} is the single transistor finger width, l_{mi} is the transistor length, nf_i is the total number of transistor fingers, *polyExt* is the polysilicon extension over active diffusion area, d is the user-specified minimum distance between modules, *SD* is the distance between transistor fingers, L_d is the side lateral diffusion length of the source & drain region in the multi-finger structure, v_{i1} and v_{i2} are the vertical coordinates of the ith transistor, and h_{i1} and h_{i2} are the horizontal coordinates of the ith transistor. The multi-finger structure is implemented and demonstrated by M1 and M9 in Fig. 4, and the rest of devices are depicted by simplified blocks. Special constraints can be also included so that the total interconnect parasitics of sensitive nodes can be well restricted, e.g., C_{intp} and C_{intn} of the two output nodes in Fig. 4 staying equal in the minimum-sized floorplan for reducing capacitive mismatch.

3.4.2. Parasitics Consideration in Both GeoP and EA Sizing Phases

Once a floorplan template is derived, two categories of parasitics, i.e., device parasitics and interconnect parasitics, can be modeled and integrated into a set of symbolic layout constraints as follows. Firstly, for the CMOS technology, sensible capacitance and resistance models for sub-circuit device parasitics, like C_{ds}, C_{gs}, and C_{db}, are available from the foundry technology model files. These models provide acceptable device intrinsic capacitance, intra-device local interconnect capacitance, and resistance in terms of transistor width, length, number of fingers, and technology-dependent coefficients. By using these parasitic models, intrinsic device capacitance, local interconnect capacitance, and resistance constraints are all formulated in a symbolic form and passed to the GeoP modeling and later be reused during the EA optimization phase.

Secondly, by using interconnect geometric size, unit resistivity of interconnect layer, unit capacitance from interconnect layer to substrate and active region, as well as unit interconnect-interconnect coupling capacitance, the symbolic expressions of interconnect parasitics can be obtained (as detailed by (17)-(20) in Section 3.4.3). We adopt the scheme in [10], which claimed to have estimation errors lower than 10% to accurately estimate the interconnect substrate parasitic capacitance and coupling parasitic capacitance between interconnects on the same or different layers. Such an analytic capacitance model, developed on the basis of electric field approximation and curve-fitting technique, and the inter-device parasitic resistance model [9] are used in our work to generate symbolic (for the GeoP-based sizing process) and numerical (for the EA-based sizing process) parasitic constraints for circuit interconnections. To be specific, these models incorporated in the EA phase can dynamically calculate the interconnect parasitics, which result from the fluctuating device geometries during the evolution. The updated interconnect parasitics will be back annotated to netlist for numerical simulation. Furthermore, thanks to the availability of the device intrinsic parasitic models that come from the foundry and the interconnect parasitic models that come from the literature [9] [10], the only effort for formulating complete symbolic parasitic expressions relies on the derivation of interconnect segments, which are generated by the floorplan optimization and global routing.

These parasitic expressions, which are present inside the analytic performance constraints in the GeoP sizing phase and through netlist for the numerical simulation in the EA sizing phase, would influence the sizing results. Furthermore, as a result of deploying the tractable floorplan template, the vulnerable parasitic impact can be effectively made up by tuning device sizes or bias conditions in the sizing process. So without compromising the optimization resolution, the search solution space can be significantly decreased. Below we demonstrate such a contrast quantitatively by using a circuit with only four MOSFETs. We assume only the following limited discrete values of MOSFET width, length and multiplier are included as the sizing solutions: the MOSFET width varies from 100nm to 1μm, the MOSFET length varies from 100nm to 500nm, the MOSFET multiplier varies from 1 to 5, and the minimum metric step size is 100nm for the width and length while the minimum multiplier step size is 1. Therefore, the number of sizing solutions is $(9*4*5)^4$ = 1,049,760,000. We also assume that the four transistors in this case study can only be located in certain places so that they form a regular matrix (i.e., just four options including 1×4 matrix, 2×2 matrix, and their transposed counterparts). Thus, the total number of possible floorplans is at least 4!*4=96 and accordingly the entire search solution space of this sizing problem contains as much as 96*1,049,760,000 = 100,776,960,000 distinct sizing solutions. Within such huge search solution space, by using our proposed scheme with the aid of a traceable floorplan template, we can effectively shrink the search scope by (96-1)/96 = 98.95%.

Moreover, for the high-performance analog circuits or RF circuits, the performance or geometric constraints should be properly managed within our proposed parasitic-aware hybrid sizing methodology. Sensitivity analysis can be conducted for these circuits to gain comprehensive domain knowledge. To account for the parasitic estimation difference between our proposed parasitic modeling and the off-the-shelf layout extraction tools, the designers may opt to apply conservative bounds for the sensitive nets in order to leave certain room in case the actual parasitics may drive the performance off track.

3.4.3. GeoP Compatibility for Interconnect Parasitics

In the following, we prove the intact GeoP-compatibility nature of an analog/RF circuitry design if adding floorplan and parasitic constraints.

Theorem: For a known floorplan, integration of floorplan and interconnect parasitic constraints has no impact on the GeoP-compatibility of an analog/RF circuitry design.

Proof: The position of each rectangular device can be represented by symbolic coordinates of its four corners. The relative device positions in a given floorplan can be expressed by linear inequalities with device corner coordinates and distance symbols like Eq. (16). Without loss of generality, since all the expressions are non-negative if following the constraint construction scheme above, the formulated posynomial floorplan inequality constraints would not alter the GeoP-compatibility of the original analog/RF circuitry design.

By using coordinates and segment symbols, interconnect length can be expressed to enclose several non-negative component lines based on a given floorplan while interconnect width can be represented by a single variable to be optimized. The interconnect overlap capacitance can be calculated by [10],

$$C_{ov} = intLength * intWidth * C_{ov\_unit} \ , \tag{17}$$

where *intLength* and *intWidth* are the interconnect length and width to be optimized, and C_{ov_unit} is a technology dependent constant. The interconnect fringe capacitance depends on the perimeter of the interconnect geometry,

$$C_{fringe} = 2 * \frac{\varepsilon_0 * \varepsilon_r}{\pi} * ln\left(1 + \frac{2*t}{dist}\right) * C_{fringe_{unit}} * (intWidth + intLength) \ , \tag{18}$$

where ε_0 is the vacuum permittivity, ε_r is the relative dielectric coefficient, *t* is the thickness of a given interconnect layer, *dist* is the vertical distance between this layer and substrate. C_{fringe_unit} is a technology-dependent constant, and so do *t* and *dist*. Therefore, the total interconnect capacitance is,

$$C_{int} = C_{ov} + C_{fringe} \ . \tag{19}$$

The interconnect parasitic resistance can be achieved by,

$$R_{int} = \frac{intLength * \rho}{intWidth * intThick} \ , \tag{20}$$

where *intLength* is a posynomial, *intWdith* is a monomial, ρ is the sheet resistivity and *intThick* is the thickness of the interconnect layer (both as technology-dependent constants). Since all of the parasitic equations above are posynomials, adding these interconnect parasitic constraints would not affect the original nature of the formulation in terms of GeoP-compatibility. Therefore, the theorem holds for the inclusion of both floorplan and parasitic constraints. ■

3.4.4. The Implication of GeoP Elite Output

The implied knowledge from the GeoP output can help eliminate unknowledgeable random exploration. As the standalone EA sizing method has no knowledge of the target circuits, their chromosome-variable range is normally set much wider than that of the GeoP-EA scheme equipped by the implied knowledge from the GeoP elite output. Moreover, some fundamental knowledge of the circuit sizing design rules are required. Usually the device width and length (i.e., equivalently silicon area) are encouraged to be smaller. The minimal value of device width or length is limited by the technology node. Then the optimization job is to attempt to gain better performance by consuming less silicon area (i.e., selecting smaller width and length if possible).

We have integrated the following tactics for generating design variables used within the EA optimization. A user-defined percentage (e.g., 50% by default) is added to each MOSFET length variable value obtained from the GeoP elite solution as its upper bound, whereas another user-defined percentage (e.g., 100% by default) is appended to the other variables (including the MOSFET width) on top of the GeoP elite solution as their upper bounds. Based on these extended variable ranges, the EA optimization would select the optimal variable values and their combination to reach the best performance. The value selection of these user-defined percentages can be determined by the designers according to their understanding of the complexity level for the circuit to be sized.

Then a step size is determined for enumerating possible discrete variable values within the applied knowledge-implied variable range. The selection of the step size should consider the tolerance constrained by the target technology and circuit simulator. In our experiments, 10nm and 5nm were used by default as the step sizes of MOSFET width and length, respectively. Special care should be given to the inductor since the relationship between its device properties (e.g., *Q* factor and inductance) and parameters (e.g., radius, width, and turns) are highly discontinuous and nonlinear. In contrast, without any clue of the GeoP elite solution, any EA method may have to be obligated to provide wider variable ranges to avoid missing any potential optimal solutions. This would naturally result in hardship in the evolutionary search and optimization.

3.5. Experimental Results

This section is divided into four sub-sections. Sub-section 3.5.1 briefly introduces the three experimental circuits followed by an elaboration of the GeoP modeling. Sub-section 3.5.2 studies performance difference between the parasitic-inclusive GeoP sizing and the one without consideration of parasitics. Following the introduction of experimental setup, Sub-section 3.5.3 highlights the merits of our GeoP-EA hybrid method by providing experimental results compared to the other alternative schemes. Finally Sub-section 3.5.4 discusses the post-layout verification.

The flow of our experiment is as follows. In the GeoP-based first sizing phase, the device sizes and bias are treated as the GeoP sizing variables, which are involved in the circuit modeling built in Matlab. Then the solved GeoP elite is input to the EA-based second sizing phase as one evolutionary individual introduced into the initial evolutionary generation. During the iterative evolution, every recombined trial solution and its contained sizing variables will be written into the circuit netlist. It will then be simulated by using the numerical simulator (e.g., Cadence Spectre [56]). Then the performance is extracted and associated with its corresponding trial solution, which is used for competitive survival test for composing the next generation. The EA engine is implemented in C++. The experiment setup for a variety of experimental schemes are detailed in Section 3.5.3.

3.5.1. Parasitic-Aware GeoP Modeling

In this chapter, we use the following three circuits as demonstrative test examples. A widely used two-stage P-channel input operational amplifier (Op-Amp) made of a single-ended differential amplifier stage followed by a common-source stage is shown in Fig. 5(a). R1 and R2 represent interconnect parasitic resistances between differential pair (M1 and M2) and tail transistor M5. A differential-pair comparator as depicted in Fig. 5(b) is explained in more detail to show the parasitic-inclusive GeoP modeling in compliance with the selected floorplan template in Fig. 4. The third circuit, a Cascode common-source low noise amplifier (LNA), is shown in Fig. 5 (c). In our experiments, these circuits were designed in the different technologies: the two-stage Op-Amp in a CMOS 0.18um technology, the differential comparator in a regular CMOS 90nm technology, and the LNA in a CMOS 90nm low power technology. Moreover, we used BSIM4 level-14 model for all the circuit numerical simulations.

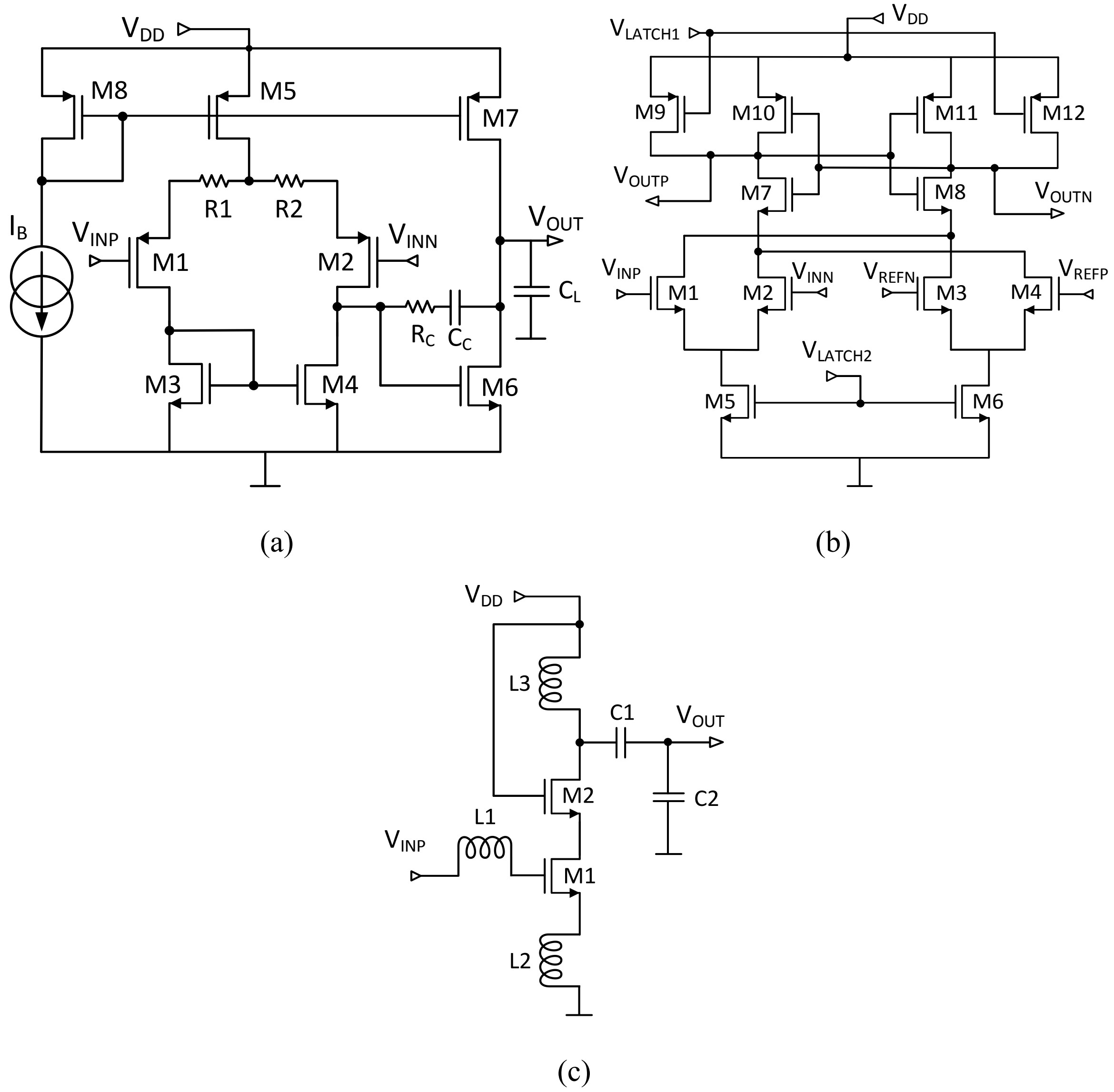


Fig. 5. Circuit diagrams for a) two-stage Op-Amp, b) differential-pair comparator, and c) cascode common source LNA with source degeneration

The differential-pair comparator, belonging to the category of dynamic comparators, is faster than any gain-based comparators yet still with minimum power consumption because it is driven by clock signals. As the comparator is a regenerative one based on latch, the latch time constant can be expressed as [66],

$$\tau_l = \frac{C_{out}}{g_m}\,. \tag{21}$$

where C_{out} is the total capacitance at the positive ($C_{out,p}$) and negative ($C_{out,n}$) output nodes. These capacitances can be written as,

$$C_{out,p/n} = C_{db7/8} + C_{db10/11} + C_{gs11/10} + C_{gs8/7} + C_{intp/n}\,, \tag{22}$$

where the slash symbol represents the meaning of "OR", and $C_{intp/\mathrm{n}}$ is the interconnect capacitance that can be modeled in a symbolic form by using the minimum-size floorplan as depicted in Fig. 4. The propagation delay of the latch, t_{prop}, as a target specification, can be written in terms of the final high and low output voltages (V_{oh} and V_{ol}),

$$t_{prop} = \tau_l ln\left(\frac{V_{oh}-V_{ol}}{2\Delta V_{in}}\right), \tag{23}$$

where ΔV_{in}, which is always less than $V_{oh} – V_{ol}$, is the difference between the two latch output voltages before the latch is enabled. The target maximum allowable propagation delay may be specified as a user-defined value (e.g., 1ns).

When the two clocks are in the evaluative phase, the latch is enabled. And depending on the resistance of the two branches, the latch decides which output will stay high and which one will go low. As the transistors M1, M2, M3, and M4 work in triode region, the MOSFET on-resistance of the two branches can be written as,

$$\frac{1}{R_{1,3/2,4}} = k_n\left[\frac{W_{1/2}}{L_{1/2}}\left(v_{in}^{+/-} - v_t\right) - \frac{W_{3/4}}{L_{3/4}}\left(v_{ref}^{-/+} - v_t\right)\right]. \tag{24}$$

As W_2 & W_4 (L_2 & L_4) and W_1 & W_3 (L_1 & L_3) are considered equal, respectively, the conductance of the two branches can be written as,

$$G_{1,3} = G_{2,4} = k_n \frac{W}{L}\left(V_{in} - V_{ref} - 2V_t\right) . \tag{25}$$

The resistances, which must be equal for the same applied inputs voltages and reference voltages in order to ensure proper matching between two branches, are found to be the inverse of the conductances above. Each resistance is taken under a certain specified value R_{max} to ensure sufficient speed. Moreover, a capacitive mismatch between the two output nodes can readily cause the comparator to malfunction. So another constraint (26) has to be added in order that the difference between the interconnect capacitances, modeled from the floorplan, is smaller than a certain specified value (C_{diff}),

$$C_{out,p} - C_{out,n} \le C_{diff} . \tag{26}$$

With the floorplan template as shown in Fig. 4, the geometric constraint in terms of overall silicon area can be formulated as,

$$\begin{gathered}
\max(Dist_{v(9,5,1)}, Dist_{v(10,7,2)}, Dist_{v(11,8,3)}, Dist_{v(12,6,4)}) * \\
\max(Dist_{h(9,10,11,12)}, Dist_{h(5,7,8,6)}, Dist_{h(1,2,3,4)}) \le Spec._{area} , \\
Dist_{h(9,10,11,12)} = Dist_{h9} + Dist_{h10} + Dist_{h11} + Dist_{h12} + 3d , \\
Dist_{v(9,5,1)} = Dist_{v9} + Dist_{v5} + Dist_{v1} + 2d , \\
Dist_{hi} = l_{mi} * nf_i + (nf_i - 1) * SD + 2 * L_d , \\
Dist_{vi} = w_{mi} + 2 * polyExt ,
\end{gathered} \tag{27}$$

where $Dist_{hi}$ and $Dist_{vi}$ are the horizontal and vertical edge-to-edge distances (i.e., module size) for module i, $Spec_{area}$ is the specification of the consumed silicon area, and all the other parameters are defined by following (16).

With this floorplan template, the interconnect length for the positive output node (C_{intp} as shown in Fig. 4) is calculated by,

$$\begin{aligned} intLength_{out,p} = Dist_{h7} + 0.5Dist_{h8} + intLocal_{9,10,7,8,11} + \\ 0.5(Dist_{v9} + Dist_{v10} + Dist_{v7} + Dist_{v8} + Dist_{v11}) + 4d \ , \end{aligned} \tag{28}$$

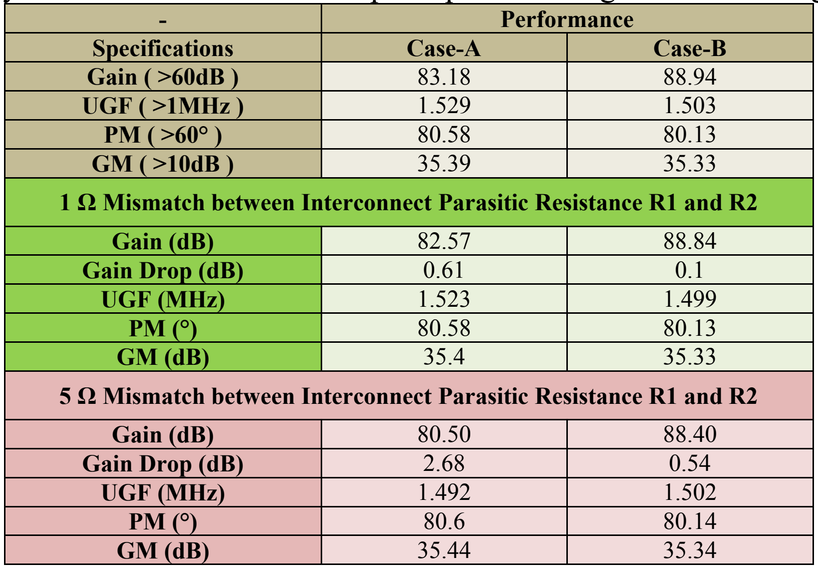
where $intLocal_{9,10,7,8,11}$ is the total length of the intra-module interconnect segments for modules 9, 10, 7, 8, and 11 (assuming that the length of each intra-module interconnect segment is constant). Then $C_{out,p}$ can be calculated by applying $intLength_{out,p}$ to (17)-(19).

As exemplified above, the performance equations (e.g., (23)), electrical constraints (e.g., (24)-(26)), and geometric constraints (e.g., (27)) are formulated in the GeoP-compatible form for sizing a circuit. Moreover, the intrinsic parasitics (e.g., C_{db} and C_{gs}) and interconnect parasitics (e.g., C_{intp}) as described above are integrated together into (22) in order to control the circuit performance t_{prop} via (21) and (23). The primary GeoP variables include MOSFET width, length, multiplier, and interconnect width.

3.5.2. Parasitic Consideration in the GeoP Sizing Phase

The highly efficient GeoP solver as the first-phase sizing engine can achieve an initial solution within a few seconds. It is much faster than any computationally intensive simulators or statistical/deterministic algorithms running from scratch. The experimental performance of the two-stage Op-Amp from the GeoP sizing phase is listed in Table 1. Case A uses the performance model along with MOSFET intrinsic parasitic model, and Case B takes into account interconnect parasitics on top of the models used in Case A. Without considering the floorplan-induced interconnect parasitics in the sizing process, Case-A in Table 1 shows that the gain can only reach 83.18dB. In contrast, if the interconnect parasitics are considered in the sizing process as proposed in Section 3.2, the device sizes from the GeoP optimization reflect such a change and derive a set of different sizing results accordingly. Thus the simulation results with interconnect parasitics consideration has exhibited a gain boost up to 88.94dB.

Table 1. Pre-layout simulation results for Op-Amp when using the GeoP sizing methodology



-	Performance	
Specifications	**Case-A**	**Case-B**
Gain (>60dB)	83.18	88.94
UGF (>1MHz)	1.529	1.503
PM (>60°)	80.58	80.13
GM (>10dB)	35.39	35.33
1 Ω Mismatch between Interconnect Parasitic Resistance R1 and R2		
Gain (dB)	82.57	88.84
Gain Drop (dB)	0.61	0.1
UGF (MHz)	1.523	1.499
PM (°)	80.58	80.13
GM (dB)	35.4	35.33
5 Ω Mismatch between Interconnect Parasitic Resistance R1 and R2		
Gain (dB)	80.50	88.40
Gain Drop (dB)	2.68	0.54
UGF (MHz)	1.492	1.502
PM (°)	80.6	80.14
GM (dB)	35.44	35.34

Although the size and layout structure (e.g., finger number) of matching-constrained devices are relatively easy to be considered in the sizing process, it is normally difficult to ensure a parasitic-matched design without floorplan information. This is particularly true for the matched MOSFETs that may be affected by distinct neighboring devices, different routing circumstances, layers and vias in between, etc. Table 1 also shows the effects of interconnect parasitics mismatch on performance, mainly on gain due to parasitic resistance mismatch. As shown in Fig. 5(a), R1 and R2 reflect the interconnect parasitic resistances between M5 and M1 and between M5 and M2, respectively. Two sets of resistance mismatch (1 Ω and 5 Ω), which can be readily attained from regular analog layouts, were applied to both Case-A and Case-B. In both mismatch cases, the gain drops of Case-A (i.e., 0.61dB and 2.68dB) are significantly greater than those of Case-B (i.e., 0.1dB and 0.54dB). This experiment shows that the sizing results with interconnect parasitic consideration can be more immune to the parasitic mismatch, which might appear due to imperfect layout in practice.

3.5.3. GeoP-EA Hybrid Sizing

After the GeoP solver solves the modelled sizing problem exemplified in Section 3.5.1, a GeoP elite solution is obtained. Since the GeoP modeling is an approximation approach, modeling errors do exist. And if they can be compensated by a refinement optimization like EA, it is possible to further discover higher-accuracy solutions. Another reason of passing the GeoP results into the EA (i.e., DE or θ-DEA) optimization is to help form a condensed search space. Thus, by eliminating unknowledgeable random exploration, the time consumed by EA optimization can be significantly decreased without compromising the search accuracy. Based on the GeoP elite solution, the shrunk search space in the EA sizing phase can be established by properly arranging the variable bounds and setting the step size for each chromosome-variable in the chromosome-vector as described in Section 3.4.4.

In our experiments, each method was run for 10 times iteratively and some statistics data were extracted to reflect the performance of the method under study. The IGD metric for multi- or many-objective methods was defined in (15). The reported data in Table 2, Table 3, and Table 4 were extracted from a selected run with the median fitness for single-objective EAs or with the median IGD for many-objective EAs. For each test circuit, eight schemes are compared with one another. Scheme-0 is the standalone parasitic-aware GeoP-based sizing method as discussed in Section 3.3.1 and Section 3.5.1. Scheme-1 follows the Synthesis Flow for fast Parasitic Closure (called *SFPC* for short) originally proposed in [59], which encloses placement and global routing inside a refined-sizing loop.

The rest of parasitic-aware GeoP-EA hybrid sizing methods are implemented with different evolutionary configurations. Scheme-2 is for the single-objective DE sizing method (as discussed in Section 3.3.2) without GeoP phase (called *NoGeoP-DE*) whose NP and G_{max} are set as 30 and 50 respectively [67], while Scheme-3 is similar but integrated with the GeoP result (called *GeoP-DE*) whose NP and G_{max} are set as 15 and 8 respectively. Schemes 4-7 are for many-objective θ-DEA (as discussed in Sec. 4.3), where Schemes 4-5 have no GeoP involvement and Schemes 6-7 include the GeoP elite output in their evolution process. For each of the two categories above, large-scale and small-scale θ-DEA configurations with NP * G_{max} of 56 * 40 (i.e., in Scheme-4 and Scheme-6) and 32 * 20 (i.e., in Scheme-5 and Scheme-7) respectively are studied. In addition, Schemes 1-7 are based on simulation, and there is no actual layout generation procedure for any schemes in Table 2, Table 3, and Table 4. Thus, the reported run time only reflects the schematic-level sizing optimization process. Moreover, the silicon area is estimated by the device sizes and the power consumption is reported from the simulation for the representative solutions.

For many-objective optimization problems, to encourage the generation of optimal clusters, the clusters have to be distributed across the entire solution space even in certain infeasible regions by managing a systematic construction of reference points. Therefore, only the solutions that pass specifications should be included in the statistics calculation for the many-objective methods. Otherwise, the collected statistical data, which are dominated by infeasible solutions, reflect little sense for evaluating the performance of any individual method. In addition, rather than using standard deviation in the final nondominated set, we employ success-rate to depict how many solutions within one population can meet the due specifications. On the other hand, for the single-objective methods, the fitness function in (12) is used for the statistics calculation, while the standard deviation is still extracted to exhibit the status of convergence and diversity for the evolutionary optimization. If single-objective and many-objective methods have to be compared with each other, we suggest simply unifying the many-objective solutions by using (12) (as a post processing for each obtained solution from the many-objective methods) to treat the calculated fitness amount as the ultimate benchmark (as shown in the row of "Median IGD/Fitness Run: Best-Fitness" in Tables 2-4), which itself should be viewed impartially due to lack of dominance concept in the operation above.

As shown in Table 2, the single-objective methods generally perform worse than the many-objective methods in terms of the best-fitness. But it is noticeable that Scheme-3 (*GeoP-DE*) can base its evolution on top of the GeoP elite output and deliver acceptable solutions with less runtime compared to the others. One can also observe that there is a very small difference of the fitness amounts among Schemes 4-7, which can hardly tell an overridden preference regarding the optimization quality. To the benefit of computation efficiency, the small-scale configuration is certainly appealing.

Table 2. Algorithmic settings and performance of the Two-stage Op-Amp

Schemes	GeoP	Single-objective Methods			Many-objective θ-DEA			
	Sch-0	**Sch-1 (SFPC)** [59]	**Sch-2 (NoGeoP-DE)** [67]	**Sch-3 (GeoP-DE) [This work]**	**Sch-4 (NoGeoP-θ-Large)**	**Sch-5 (NoGeoP-θ-Small)**	**Sch-6 (GeoP-θ-Large)**	**Sch-7 (GeoP-θ-Small) [This work]**
Median IGD/Fitness Run: Best-Fitness	0.593	0.554	0.621	0.544	0.450	0.478	0.458	0.460
Median IGD Run: Average-Fitness	-	-	-	-	0.519	0.552	0.528	0.569
Median IGD Run: Success-Rate	-	-	-	-	30.36%	25.00%	53.6%	21.88%
Median Fitness Run: Average	-	0.734	0.731	0.711	-	-	-	-
Median Fitness Run: Standard-Deviation	-	0.098	0.114	0.114	-	-	-	-
Run Time (hours)	1.31 sec.	1.74	9.64	0.34	8.04	2.37	7.38	2.02
Obj. & Spec.	**Performance (from the Representative Solution with the Smallest Fitness Value)**							
Est. Area (μm^2)	556.11	8243.05	5059.54	542.34	1749.16	4758.87	799.11	467.40
DC Power (μW)	20.92	85.38	59.19	28.56	34.65	95.78	210.10	113.2
Gain > 60dB	88.94	93.53	91.96	87.12	85.97	72.46	73.26	80.75
UGF > 1M	1.50	2.53	1.35	2.54	4.06	4.32	20.83	5.52
PM > 60°	80.13	78.39	81.86	76.58	124.41	131.35	91.32	87.59
GM > 10dB	35.33	24.15	27.93	32.06	26.82	25.34	32.26	43.48

With reference to Scheme-0 (i.e., standalone *GeoP*), the size variations of the eight transistors (i.e., M1/M2, M3/M4, M5, M6, M7, and M8) in the two-stage Op-Amp example are 50%, 0%, 2.30%, 0.47%, 2.44%, and 2.28% for Scheme-3 (i.e., *GeoP-DE*), and 98.53%, 84.51%, 65.22%, 34.32%, 86.57%, and 53.52% for Scheme-7 (*GeoP-θ-Small*), respectively. It is obvious that a close resemblance of device sizes can be observed between Scheme-0 and Scheme-3, whereas less similarity can be found between Scheme-0 and Scheme-7. This is because the many-objective θ-DEA method treats the GeoP elite solution as a candidate only in one of its many clusters, which are explored simultaneously. So the reported representative solution might be derived from another cluster different from the one holding the GeoP elite solution. In contrast, the DE method explores the neighboring regions based on the current best-fitness solution. Depending on the configured problem space and evolutionary operators, the DE method may not be able to conduct an adventurous search with big step size. Therefore, it normally accounts for less aggressive size variation with reference to Scheme-0.

Table 3 provides the sizing results for the differential comparator. Propagation delay is one of the most important characteristics for the comparator circuit, and the positive and negative overshoots are given with the absolute values in the table. For the single-objective methods, *SFPC* (i.e., Scheme-1) and *NoGeoP-DE* (i.e., Scheme-2) exhibit the results of poor best-fitness and average-fitness but with good standard-deviation (i.e., 0.7, 0.837, and 0.111 respectively), or good best-fitness but with poor average-fitness and standard-deviation (i.e., 0.4, 0.957, and 0.804 respectively). It is obvious that *SFPC* can only generate a less favorable solution compared to the standalone GeoP method (i.e., Scheme-0) with a fitness of 0.584. In contrast, *GeoP-DE* (i.e., Scheme-3) performs reasonably well with the acceptable performance as well as the least runtime. For the many-objective methods, both *NoGeoP-θ-Large* (i.e., Scheme-4) and *NoGeoP-θ-Small* (i.e., Scheme-5) perform relatively worse than *GeoP-θ-Large* (i.e., Scheme-6) and *GeoP-θ-Small* (i.e., Scheme-7) as per the data from the first two rows, which tend to exhibit a different trend from the two-stage Op-Amp. We believe this is highly related to the nature of the comparator circuit, whose switching operation in the current path may be reversed quickly when the logic balance is broken by sufficient change of device sizes or incurred parasitics. That is to say, the solutions space of the comparator is more complex than that of the two-stage Op-Amp. In this situation, the circuit knowledge from the GeoP elite seems more helpful for such complex sizing problems as it can lead to confined solution space for an enhanced depth of exploitation.

Table 3. Algorithmic settings and performance of the Differential Comparator

Schemes	**GeoP**	**Single-objective Methods**			**Many-objective *θ*-DEA**			
	Sch-0	**Sch-1 (SFPC)** [59]	**Sch-2 (NoGeoP-DE)** [67]	**Sch-3 (GeoP-DE) [This work]**	**Sch-4 (NoGeoP -θ- Large)**	**Sch-5 (NoGeoP -θ- Small)**	**Sch-6 (GeoP-θ- Large)**	**Sch-7 (GeoP-θ- Small) [This work]**
Median IGD/Fitness Run: Best-Fitness	0.584	0.700	0.400	0.455	0.458	0.549	0.378	0.416
Median IGD Run: Average-Fitness	-	-	-	-	0.502	0.542	0.508	0.507
Median IGD Run: Success-Rate	-	-	-	-	3.85%	3.57%	9.62%	17.86%
Median Fitness Run: Average	-	0.837	0.957	0.778	-	-	-	-
Median Fitness Run: Standard-Deviation	-	0.111	0.804	0.297	-	-	-	-
Run Time (hours)	1.24 sec.	1.93	11.80	0.45	7.44	1.95	7.76	1.88
Obj. & Spec.	**Performance (from the Representative Solution with the Smallest Fitness)**							
Est. Area (μm^2)	358.33	517.00	460.15	239.51	357.48	470.71	189.39	178.43
DC Power (μW)	4.97	23.24	9.07	8.45	7.80	20.18	5.49	5.78
Propagation Delay < 600ps	594	245	311	409	472	540	332	512
+Overshoot < 450mV	208	411	160	176	165	216	144	100
-Overshoot < 150mV	45	117	49	44	33	40	39	26

Similar experiments were conducted on the LNA circuit with the results listed in Table 4. On the single-objective side, none of the schemes can reach the specifications, which partially exhibits the hardship of the LNA sizing optimization. *SFPC* (i.e., Scheme-1) cannot deliver a solution with sound fitness due to its frequent floorplan variation. *NoGeoP-DE* (i.e., Scheme-2) fails on S22 after running for 8.07 hours. Although highly efficient, *GeoP-DE* (i.e., Scheme-3) attempts to minimize the best-fitness to 0.759, but with a failure in satisfying S11 specification. In contrast, the many-objective EAs perform much better than the single-objective methods. Within the many-objective EA group, one can observe that both *NoGeoP-θ-Large* (i.e., Scheme-4) and *NoGeoP-θ-Small* (i.e., Scheme-5) perform worse than *GeoP-θ-Large* (i.e., Scheme-6) and *GeoP-θ-Small* (i.e., Scheme-7) with larger best-fitness and around five times lower success-rate (i.e., 3.13% and 1.79% versus 15.63% and 8.93%, respectively). This exhibits the positive effect of the circuit knowledge integration from the GeoP elite output with the many-objective θ-DEA optimization. We believe this phenomenon is highly correlated to the complex nature of the LNA design, where the relationship between inductor parameters and properties does not follow one-to-one correspondence. As a result, the formed super complex solution space demands specific circuit knowledge assistance from the GeoP elite output to facilitate the localization of optimal regions provided that the same amount of distributed reference points is accessible.

Table 4. Algorithmic settings and performance of the Low Noise Amplifier

Schemes	**GeoP**	**Single-objective Methods**			**Many-objective θ-DEA**			
	Sch-0	**Sch-1 (SFPC)** [59]	**Sch-2 (NoGeoP-DE)** [67]	**Sch-3 (GeoP-DE) [This work]**	**Sch-4 (NoGeoP -θ- Large)**	**Sch-5 (NoGeoP -θ- Small)**	**Sch-6 (GeoP-θ-Large)**	**Sch-7 (GeoP-θ-Small) [This work]**
Median IGD/Fitness Run: Best-Fitness	0.767	2.983	0.885	0.759	0.691	0.702	0.615	0.646
Median IGD Run: Average-Fitness	-	-	-	-	0.691	0.702	0.657	0.697
Median IGD Run: Success-Rate	-	-	-	-	1.79%	3.13%	8.93%	15.63%
Median Fitness Run: Average	-	3.119	3.719	2.516	-	-	-	-
Median Fitness Run: Standard-Deviation	-	0.129	1.972	1.300	-	-	-	-
Run Time (hours)	1.76 sec.	1.61	8.07	0.38	8.14	2.74	7.84	2.46
Obj. & Spec.	**Performance (from the Representative Solution with the Smallest Fitness Value)**							
Est. Area (mm^2)	0.267	0.308	0.292	0.255	0.281	0.306	0.288	0.260
DC Power (mW)	42.96	19.97	18.73	47.05	18.59	25.97	16.72	14.70
Gain > 15dB	20.32	3.01	20.21	19.63	21.00	16.47	19.55	19.34
NF < 2.5dB	1.87	5.08	2.19	1.71	2.12	2.20	1.98	2.01
S11 < -12dB	-15.16	-8.23	-19.82	-11.71	-22.17	-29.13	-20.69	-19.41
S22 < -12dB	-15.16	-3.47	-9.12	-21.28	-18.14	-19.79	-37.47	-30.94

Based on the analysis of the experimental data above, one can learn that the efficiency of GeoP (i.e., Scheme-0) is the highest thanks to its pure symbolic nature. The sizing result of GeoP is generally better than *SFPC* (i.e., Scheme-1) and *NoGeoP-DE* (i.e., Scheme-2). With the aid of the

GeoP elite information, Schemes 3, 6, and 7 can improve the performance in terms of the best-fitness. In addition, even without the GeoP involvement as the first sizing phase, the powerful many-objective θ-DEA method in Schemes 4 and 5 can still achieve better performance than GeoP.

Furthermore, it is observed that single-objective methods generally derive solutions inferior to those from the many-objective schemes. The *SFPC* scheme is not as suitable as our proposed hybrid GeoP-EA method for addressing the analog/RF parasitic-aware sizing problems along with layout effects. We believe this is strongly related to the fact that the evolving parasitics in *SFPC* are quite inconsistent along the optimization path. At each iteration, the sizing process may derive various module geometries, which can lead to very different floorplans due to free control in the subsequent placement and global routing. These frequently changed floorplans would, in turn, bring forth oscillating parasitics, which hardly provide a priori informative guidance to the next refined-sizing iteration. Our experimental results expose that the intractable parasitics fail to cooperate well with the optimization engine and thus increase the difficulty of localizing optimal solutions in practice. Moreover, *NoGeoP-DE* (i.e., Scheme-2) based on DE with large-scale setup fails to ensure a clear improvement despite high execution time. And *GeoP-DE* (i.e., Scheme-3) can perform well for simple and moderately complex problems but may fail in super complex problems.

On the other hand, based on the observation of *NoGeoP-θ-Large* (i.e., Scheme-4), one can infer that the large-scale standalone θ-DEA scheme might work fine for a simple sizing problem like the Op-Amp example even without the GeoP-implied circuit knowledge, but may not necessarily be sufficient to any moderately or super complex sizing problems such as the comparator or LNA circuit. The slight improvement of average-fitness and success-rate only in the simple Op-Amp circuit can hardly justify over three times CPU hours in practical usage, whereas the integrated GeoP running in Scheme-7 only takes a couple of seconds to generate a GeoP-elite solution for further improvement. By comparing *GeoP-θ-Large* (i.e., Scheme-6) and *GeoP-θ-Small* (i.e., Scheme-7), the largest improvement of the average-fitness (i.e., 7.2% in the Op-Amp example circuit) and the best-fitness (i.e., 9.1% in the comparator example circuit) has little support to the large-scale configuration, not to mention the fact that the representative solutions from both schemes are equally good, which are nondominated from each other in the three example circuits. It tends to be true that the more the reference points are assigned in this case, the more they are wasted as exposed from the low success-rate in *GeoP-θ-Large* (i.e., Scheme-6) for the comparator and LNA circuits. Therefore, our proposed *GeoP-θ-Small* (i.e., Scheme-7) with maximum execution time of up to 2.5 hours can provide a reasonable configuration of many-OEA after inheriting the knowledge from the GeoP elite output.

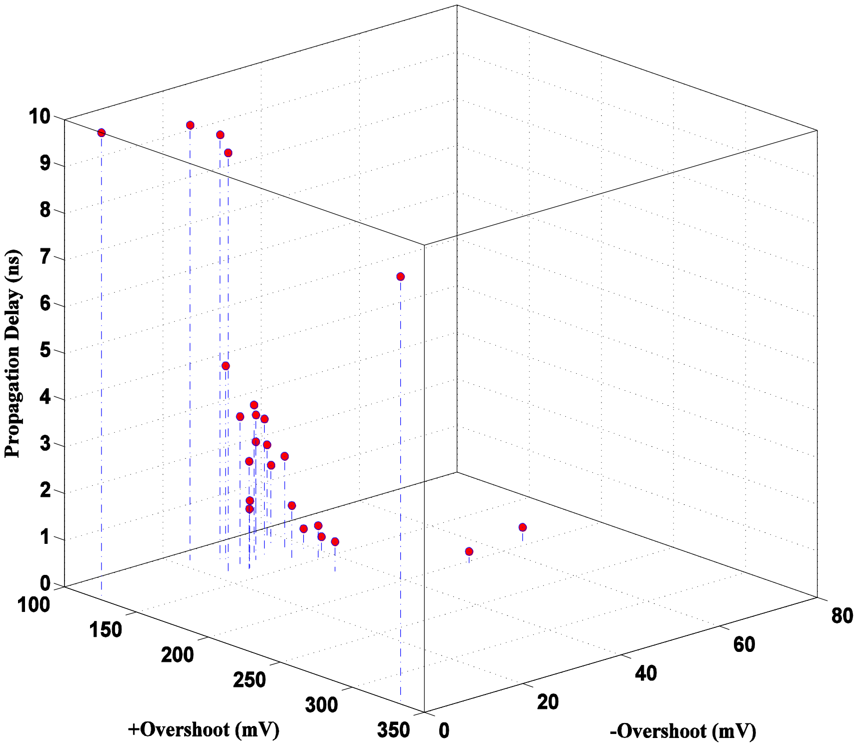


Fig. 6. Plot of the resultant solution set from the many-objective θ-DEA method for the comparator test circuit

Fig. 6 shows one 3D plot of the resultant optimal solution set from the many-objective θ-DEA method for the comparator test circuit. Each axis is defined by its corresponding objective marked in the plot. The red solid dots represent the optimal solutions, while the blue dash-dot lines exhibit their projections towards the X-Y plane. To maintain a compact solution performance space, any solution with propagation delay of over 10ns is considered to saturate at the maximum amount of 10ns in the plot. In such a minimization problem, the ideal optimum solution is supposed to be the origin (0, 0, 0) in the plot.

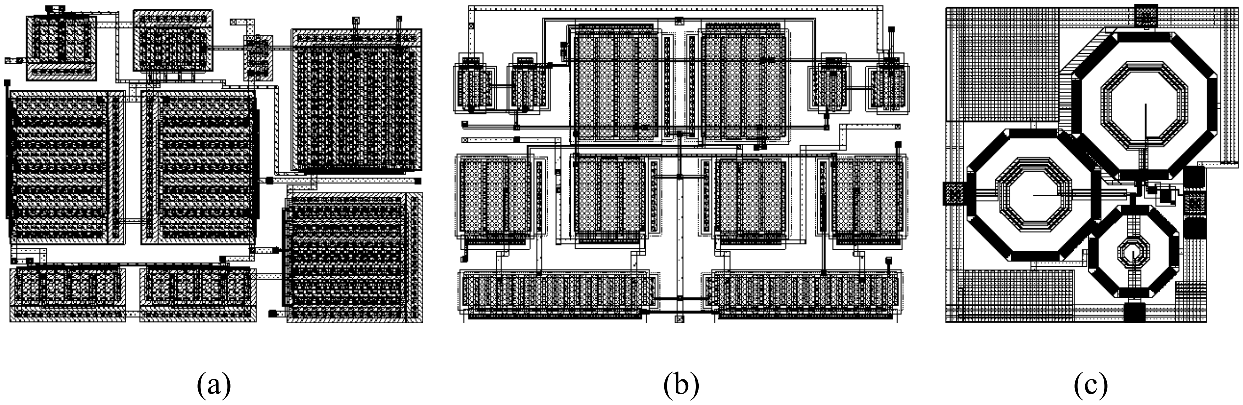
In summary, the circuit knowledge information induced by the GeoP elite output tends to effectively facilitate the optimization process, especially for the sizing problems with complex solution space. The constrained search ranges can help skip the solution space that is full of inferior solutions. Thus, sound solutions can be more efficiently approached and explored. Even though some potential regions may be lost due to the elimination, in practical engineering tasks with resource limitation, the constrained space with the elite solution centered is more worthwhile or already sufficient in the exploitation especially for the complex problems.

3.5.4. Post-Layout Verification

By following the sizing results from the EA sizing phase as well as the floorplan template used in both GeoP and EA sizing phases, in our experiment we used Cadence Virtuoso Layout-XL Suite [68] to automatically place and route the auto-generated modules to obtain the final layouts. Then we used Cadence Diva tool at the CMOS 0.18um technology node and Mentor Graphics Calibre xRC [8] at the CMOS 90nm technology node for automatic parasitics extraction. By comparing with the pre-layout simulation results for the *GeoP-DE* method (i.e., Scheme-3) and the hybrid *GeoP-θ-Small* method (i.e., Scheme-7) from Tables 2-4, the similar post-layout simulation results as reported in Table 5 further confirm the suitability of the modeled intrinsic and interconnect parasitics along with the applicability of the deployed optimization scheme by using our proposed GeoP-EA hybrid sizing methodology. As a demonstration, the layouts of the three example circuits for our promoted *GeoP-θ-Small* method (i.e., Scheme-7) are depicted in Fig. 7 (a), (b) and (c).

Table 5. The post-layout simulation results of the three example circuits

Circuits	**Performance**	**Post-Layout Simulation Results, Scheme-3 [This work]**	**Post-Layout Simulation Results, Scheme-7 [This work]**
CMOS 0.18um Two-stage Op-Amp	**Actual Area (µm^2)**	793.54	635.18
	DC Power (µW)	28.69	114.6
	Gain (dB)	87.14	80.67
	UGF (MHz)	2.54	5.36
	PM (°)	77.02	87.96
	GM (dB)	31.92	42.53
CMOS 90nm Differential Comparator	**Actual Area (µm^2)**	434.61	273.54
	Average Power (µW)	9.44	5.88
	Delay (ps)	403.6	517.1
	+ Overshoot (mV)	181.5	147
	- Overshoot (mV)	132.5	56.62
CMOS 90nm(LP) LNA	**Actual Area (mm^2)**	0.398	0.434
	DC Power (mW)	41.03	14.21
	Gain (dB)	17.74	17.95
	NF (dB)	2.07	1.96
	S11 (dB)	-21.27	-14.06
	S22 (dB)	-10.93	-12.09



(a) (b) (c)

Fig. 7. GeoP-θ-Small (Scheme-7) final layouts for a): two-stage Op-Amp, b): differential-pair comparator, and c): Cascode common source LNA with source degeneration

In summary, our proposed parasitic-aware sizing methodology is featured by its holistic concatenation of several optimization schemes. The GeoP-based first sizing phase can quickly attempt a global-view solution, which would facilitate the EA-based second sizing phase. An automatically generated floorplan template can be consistently utilized for symbolic and numerical parasitic representation in the GeoP and EA sizing process. Moreover, it is found that the single-objective DE is generally inferior to the many-objective θ-DEA especially on the complex sizing problems although the latter one requires more run time.

3.6. Summary

In this chapter, we have presented a highly efficient parasitic-aware hybrid sizing methodology. The proposed method firstly utilizes a GeoP formulation by modeling circuit performance constraints and parasitic contribution for seeking a global solution in the first phase.

Then in the second phase, it firstly employs a fast DE and then switches to a many-objective θ-DEA (if needed) both with GeoP elite output as a guidance for more focused and refined search. Compared to the other approaches that use a pre-generated look-up table or interpolation/extrapolation for parasitic estimation, our proposed method includes intrinsic device parasitics and layout interconnect parasitics in the symbolic modeling with the aid of layout floorplan information. The experimental results demonstrate the efficacy of our proposed methodology as well as its reliability over the other similar works.

In the next chapter, we will firstly propose another symbolic-analysis-based circuit modeling approach called g_m/I_D-based modeling, which is more accurate than the GeoP-based modeling due to the involvement of accurate numerical simulations. Then we will propose the g_m/I_D-EA two-phase optimization for the analog/RF circuit synthesis with parasitic awareness.

Chapter 4 Efficient Parasitic-Aware g_m/I_D-Based Hybrid Sizing Methodology for Analog and RF Integrated Circuits

4.1. Introduction

In this chapter, we firstly emphasize the importance of considering the layout parasitics, preferably in the early design stage for analog/RF circuits, by using the following example. A Cascode common-source low noise amplifier (LNA) as shown in Fig. 5 (c) features several key factors, such as an input impedance match with 50Ω resistance and a sufficient gain to overpower the noise at a pre-defined resonant frequency of 5.6GHz. According to our experiments, input reflection coefficient S11 and output reflection coefficient S22 from a parasitic-free sizing process are initially -11.09dB and -12.61dB respectively verified in an ideal pre-layout simulation (i.e., no parasitics), which are perfectly good for their due specifications of less than -10dB. However, they deteriorate to -7.51dB and -9.1dB respectively once the estimated parasitics are back annotated to corresponding electrical nets in the further pre-layout verification. Moreover, after the design is actually laid out, S11 and S22 keep deteriorating to -6.68dB and -3.4dB respectively in the post-layout verification, which definitely renders malfunctioning in reference to the specifications. Therefore, a solid and trustable parasitic-aware technique for analog and RF circuit sizing in the advanced technologies, preferably in an automated fashion, is highly demanded.

In this chapter, we propose a new g_m/I_D-based parasitic-aware analog/RF circuit sizing methodology, which includes the modeling of technology-independent circuit structure and technology-dependent device characteristics as well as parasitics. Our developed sizing method can integrate accurate intrinsic parasitics modeling (by using a piecewise curve fitting technique) and interconnect parasitics modeling (by considering layout floorplan and device geometry) into a mixed-integer nonlinear programming (MINLP) problem. The proposed g_m/I_D-based circuit models is more accurate than the geometric programming (GeoP) based circuit models introduced in Chapter 3 due to the involvement of accurate numerical simulations. In addition, we advocate a two-phase parasitic-aware sizing flow, which is comprised of a g_m/I_D-based nonlinear programming (aiming for a fast solution) and a theta dominance-based evolutionary algorithm (θ-DEA) [38] sizing refiner (for fixing any modeling shortcomings in the previous phase).

The research conducted in this chapter has been published in ACM Transactions on Design Automation of Electronic Systems (TODAES) [J3], and presented in 2018 IEEE International Symposium on Quality Electronic Design (ISQED) [C2] and 2018 IEEE International Symposium on Circuits & Systems (ISCAS) [C3].

4.2. Proposed Parasitic-Aware Hybrid Synthesis Flow

In this chapter, we are motivated to develop a hybrid parasitic-aware analog/RF circuit sizing methodology by concatenating the best ingredients from the various categories discussed above. We intend to take advantage of the accuracy of numerical simulations embedded inside the stochastic-based methods, while we prefer to improve the optimization efficiency by offering advanced global insights through promising candidates derived from the earlier phase. We opt to preserve the global view of the analytic-based methods, while we strive to eliminate the concerns of losing the modeling accuracy by considering technology-dependent factors. In this regard, we propose to use g_m/I_D-based MINLP and curve-fitting technique for the first-phase optimization, and a many-objective θ-DEA with the aid of numerical simulations for the second-phase optimization.

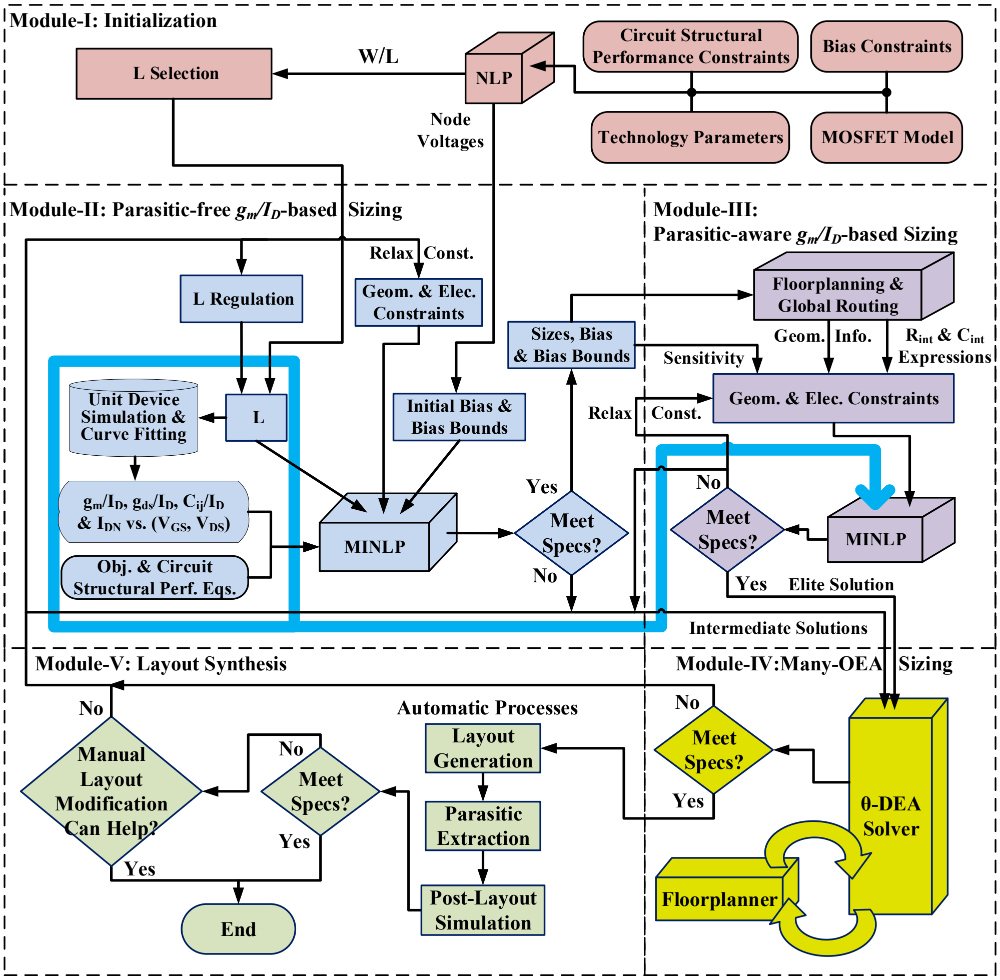

Module-I: Initialization
L Selection
W/L
NLP
Circuit Structural Performance Constraints
Bias Constraints
Technology Parameters
MOSFET Model
Node Voltages
Module-II: Parasitic-free g_m/I_D-based Sizing
Relax
Const.
L Regulation
Geom. & Elec. Constraints
Unit Device Simulation & Curve Fitting
L
g_m/I_D, g_{ds}/I_D, C_{ij}/I_D & I_{DN} vs. (V_{GS}, V_{DS})
Obj. & Circuit Structural Perf. Eqs.
Initial Bias & Bias Bounds
MINLP
Yes
Meet Specs?
No
Sizes, Bias & Bias Bounds
Module-III: Parasitic-aware g_m/I_D-based Sizing
Floorplanning & Global Routing
Geom. Info.
R_{int} & C_{int} Expressions
Sensitivity
Relax
Const.
Geom. & Elec. Constraints
No
Meet Specs?
MINLP
Yes
Elite Solution
Intermediate Solutions
Module-V: Layout Synthesis
No
Manual Layout Modification Can Help?
Yes
No
Meet Specs?
Yes
Automatic Processes
Layout Generation
Parasitic Extraction
Post-Layout Simulation
End
Module-IV:Many-OEA Sizing
No
Meet Specs?
Yes
θ-DEA Solver
Floorplanner

Fig. 8. The g_m/I_D-EA two-phase hybrid synthesis flow

As shown in Fig. 8, our proposed analog/RF circuit synthesis flow consists of five modules, each of which is composed of several operational blocks. The initialization module is to determine an initial bias condition that includes the variables in the g_m/I_D phase for configuring the problem space. Since MOSFET length (L) has an impact on device characterization, we have proposed an L-selection mechanism in order to avoid selecting improper L values that might account for repeated sizing failures in the following g_m/I_D-based modules. Both initial bias conditions (i.e., node voltages) and initial L are obtained by solving a nonlinear programming (NLP) problem formulated with topology-dependent circuit performance equations & specifications, bias constraints, and MOSFET model in addition to some technology parameters. The details of using W/L of each MOSFET as a result of the abovementioned NLP solving to determine the initial L will be further elaborated on in Section 4.3.2.

The parasitic-free and parasitic-aware g_m/I_D-based sizing modules form the core of the first-phase sizing process, where one optimal floorplan is generated in Module-III. Once L is identified, a group of numerical simulations on reference MOSFET will be performed so that the output data could be curve-fitted into analytic functions between device characteristics (e.g., g_m, g_{ds}, and intrinsic capacitances) over I_D and node voltages. In the parasitic-free g_m/I_D-based sizing module, the curve-fitted device characteristics (i.e., g_m/I_D-parameters), L, and initial bias conditions associated with the technology-independent circuit performance equations are used to formulate an MINLP problem. The sizing output from Module-II would be used as the input to the *Floorplanning & Global Routing* block in Module-III. Once an optimal floorplan is obtained and further globally routed, the symbolic interconnect relationship in terms of W and other geometrical parameters can be derived. With the aid of the nonlinear parasitic models [18], interconnect parasitic capacitances and resistances can be symbolically expressed. Then the symbolic interconnect parasitics plus their sensitivity constraints will be incorporated into the previously established MINLP problem for deriving a parasitic-aware solution via the MINLP solver in Module-III.

The fourth module is to further improve the parasitic-aware solution by using the many-objective θ-DEA sizing refiner that involves numerical simulations interacting with the floorplanner. The last module reflects the conventional layout synthesis flow, which includes layout generation, parasitic extraction, and post-layout verification by using the off-the-shelf design tools. In case the MINLP solvers in Modules II-III fail to derive a feasible solution, the synthesis would redo the optimization by relaxing the constraints inside these modules. If the solutions from Modules II-V fail to pass any specification verified by the simulations, another set of L will be attempted from module II, which is selected via the proposed L-regulation scheme discussed in Section 4.3.5. Moreover, those solutions especially from Modules II and III, which were derived by successfully satisfying the symbolic constraints, might be able to provide some optimization insights and especially enrich the population diversity if being integrated into the second-phase (i.e., Module-IV) EA-based sizing optimization.

4.3. Parasitic-Aware g_m/I_D-Based Sizing

4.3.1. Preliminaries

The g_m/I_D ratio, known as transconductance generation efficiency, is defined as follows,

$$\frac{g_m}{I_D} = \frac{1}{I_D}\frac{\partial I_D}{\partial V_{GS}} = \frac{\partial(\ln I_D)}{\partial V_{GS}} = \frac{\partial[\ln\left(I_D/(\frac{W}{L})\right)]}{\partial V_{GS}} \tag{29}$$

where g_m is the MOSFET transconductance, I_D is the MOSFET drain current, V_{GS} is the MOSFET gate-source voltage, W and L are the MOSFET width and length, respectively. As the aspect-ratio, W/L, does not depend on V_{GS}, its introduction to the natural logarithm has no impact on the partial differential operation. This derivation also shows that the g_m/I_D ratio is independent of W/L for the fixed bias voltages in the long-channel transistors [69]. In addition, the analytical expression of the large signal current I_D always includes W/L as a multiplier irrespective of its operating region. Therefore, $I_D/(W/L)$, the normalized I_D commonly referred to as I_{DN}, is also independent of W/L for the fixed bias voltages in the long-channel transistors [69]. Moreover, g_m/I_D, I_{DN}, and a set of node voltages for a MOSFET have one-to-one correspondence. For the short-channel transistors at advanced technology nodes, both g_m/I_D and g_{ds}/I_D show certain dependence on W and L. Since L is always optimized via our proposed L-initialization or L-regulation before performing the following curve fitting and circuit modeling, we only need to consider their dependence on W, which will be handled via the current density factor illustrated in Section 4.3.3 and the multiple reference W's scheme in Section 4.3.4.

Therefore, once I_D is available, the device aspect ratio (i.e., W/L) can be unambiguously determined via,

$$\frac{W}{L} = \frac{I_D}{I_{DN}} = \frac{I_D}{I_{D\_REF}/(W_{\_REF}/L_{\_REF})} \tag{30}$$

where any parameter with "_REF" is acquired from simulations on the reference MOSFET by sweeping node voltages. Moreover, the drain-to-source conductance to current ratio, g_{ds}/I_D, can be inversely measured by the Early voltage, $(V_{EA})^{-1}$, which is proportional to L for long channel transistors. For short channel devices, we have proposed to use the current density factor detailed in Section 4.3.3 to address accuracy challenges. Finally, since the intrinsic parasitic capacitances C_{ij} are mostly dependent on $W * L$ where i and j are any of the drain, source, gate or bulk nodes, we can have

$$\frac{C_{ij}}{I_D} = L^2 f_{others} \,, \tag{31}$$

where f_{others} expresses other effects largely from oxide capacitance C_{ox} and gate overlap capacitance C_{ov}. Therefore, C_{ij}/I_D is also independent of transistor sizes if L and bias voltages are fixed. According to [31], the intrinsic capacitances rely on the inversion level (i.e., mostly V_{GS}), V_{DS} and L. In this work, we have used curve fitting technique to derive C_{ij}/I_D as a function of V_{GS} and V_{DS} as detailed in Section 4.3.4.

4.3.2. Bias and *L* Initialization

As depicted in Fig. 8, the synthesis flow starts with the initialization (i.e., Module-I), which helps configure manageable variable search space by providing reasonable bias and L values for the following sizing process. MOSFET performance is characterized by I_D, g_m, and g_{ds}, which are attributed to node voltages, W/L, and technology parameters. For the first NLP modeling in the initialization module, the node voltages as well as the bias current (i.e., V_{GS}, V_{DS}, and I_D) for each MOSFET are set as variables. Given the operating region of each MOSFET, its current can be expressed in terms of W/L and node voltages. The MOSFET bias constraints and the circuit topology-dependent performance equations with respect to specifications can be also involved in the NLP modeling similar to [15]. After the NLP solving, the resultant bias conditions and W/L will be used as the initial state for the subsequent g_m/I_D sizing process. This W/L will be referred to when selecting promising L values by the L-initialization scheme. The final sizing of W will be determined by our proposed subsequent g_m/I_D sizing approach.

In comparison with the previous works on optimizing L, our proposed performance-driven L-initialization scheme is more general as it needs no deep insight into MOSFET characteristics. Firstly, a sensitivity analysis can be conducted via numerical simulations by sweeping L while maintaining W/L. As for the sweeping boundary of L (i.e., L-bound), the lower bound is defined by the technology design rules, while the upper bound is specified by the users. The minimal L-interval denoted by L_λ (e.g., 5nm by default) is determined by the technology design rules or users. To reduce the optimization complexity for better efficiency, we simplify such a sweeping process by assuming all the MOSFETs in the circuit to share an identical L initially. Then we apply our proposed performance-driven L-initialization scheme as shown in Algorithm 2 to find the best L (L_B) while minimizing the number of simulations required. Here we define the cost as a performance metric, the smaller the better, which is computed via the summation of all the normalized circuit performances (12) (e.g., DC gain) after a simulation is conducted by using an attempted L.

Then we start the L-initialization process by firstly conducting a rough-sampling operation with a large step size (e.g., running simulations with an interval of $10L_\lambda$ = 50nm by default) within the L-bound (i.e., $[L_{Min}, L_{Max}]$) to return N sampling costs. In Lines 3-7 of Algorithm 2, we perform segmentation by dividing the whole L-bound into multiple segments. Firstly, a smoothness factor (S_{th}) is derived by the average of all the cost displacements ($d_{i,i+1}$, i = 1,2,…,N–1), each of which is the absolute difference between two costs of any neighboring pair from the N sampling points. Reflecting the flatness of data distribution, S_{th} is used as a threshold to control segmentation operation in order to generate one relatively smooth segment (i.e., ψ_m, m = 1,2,…) enclosing one (i.e., L_i) or multiple rough-sampling points.

Then within each smooth segment, if the data trend is not monotonic, we further divide ψ_m into several sub-segments (ψ_m^n's) with their corresponding bounds (say, $[Lb_m^n, Ub_m^n]$) defined according to the locations of trough and peak points during the division. As a result, each sub-segment is in an ascending or descending shape. Due to the minimization purpose for later fine-sampling operation, the ascending shape is preferred where the cost becomes larger (i.e., worse) so that more points can be skipped when searching along the ascending direction. Therefore, the start point Ls_m^n in each ψ_m^n is selected at either Lb_m^n or Ub_m^n, whichever gives a smaller cost. The search direction is implicitly determined (e.g., from Lb_m^n to Ub_m^n if Ls_m^n = Lb_m^n, and vice versa).

In the next step, a fine-sampling operation is performed inside each ψ_m^n with finer sampling yet varying step size (via k) in Lines 8-21. Along the search direction defined in Line-7, index j increases and l_j denotes the L value mapped by j within each ψ_m^n, while $cost_j$ is its corresponding cost obtained from simulation. This sampling process would skip a number of L values by enlarging k, which in turn can save simulation time. The update of k is controlled by the performance change as reflected by the simulation cost. As long as the sampling points with poorer performance (i.e., larger cost) are detected, k gets larger for the next sampling operation. Or k stays intact if the cost of the current sampling point bears no change. Otherwise, a backtrack operation (i.e., go back to examine the previously skipped region with a reset of k = 1) is required if a smaller cost is identified for the current sampling point. In Line-13, the update of k is dynamically controlled by the current sampling point, the last sampling point, and the global minimal point discovered so far (i.e., $cost_B$). The best L, i.e., L_B, keeps being updated whenever a new sampling point provides a smaller cost that is even less than $cost_B$. This process is repeated until the next scheduled sampling point is out of the range of ψ_m^n where the last element needs to be examined exactly once.

Algorithm 2. L-initialization

Input: *W*/*L*'s of all the MOSFETs, **Output:** L_B

```
1. Do a rough-sampling operation based on $L$-bound (i.e., $[L_{Min}, L_{Max}]$) to return $N$ sampling costs;
2. Initialize $L_B$ (and $cost_B$) with the $L$ (and its corresponding cost) that has the best cost among the $N$ points;
3. Calculate the smoothness factor $S_{th}$ based on the cost displacements ($d_{i,i+1}$) between any two neighboring points;
4. $L_1 = L_{Min}$, $L_N = L_{Max}$, initialize $\psi_1 = \emptyset \cup L_1$, $m = 1$, $i = 1$;    // $\psi_1$ is a set with only one element $L_1$
5. while (1)    // handling one segment $\psi_m$
6.     Form a smooth segment $\psi_m$ by adding more rough sampling points ($L_{i+1}$) if $d_{i,i+1} \leq S_{th}$ through $i$++;
7.     Divide $\psi_m$ into several monotonic sub-segments $\psi_m^n$, each of which will be processed as an ascending sub-
       segment as per the determined start point and search direction;
8.     for each $\psi_m^n$    // Start the fine-sampling operation within each $\psi_m^n$
9.       $j = 1$, $k = 1$;    // $j$ as the index used for the fine-sampling inside $\psi_m^n$, $k$ for the step size
10.      $j = j + k$, and run simulation for $l_j$ inside the current $\psi_m^n$ if $cost_j$ is unknown;
11.      while ($l_j$ is within the range of the current sub-segment)
12.          if ($cost_{j-k} < cost_j$)    // if the previous cost is smaller
13.              $k = k * e^{(cost_j - cost_{j-k})/(cost_j - cost_B)}$;    // increase the step size with control
14.          else if ($cost_{j-k} == cost_j$)    $k$ keeps intact;    // maintain the step size
15.          else {    use $l_j$ & $cost_j$ to update $L_B$ & $cost_B$ if $cost_j < cost_B$;    // better cost detected
16.              if ($k$ != 1)    $j = j - k$, $k = 1$; }    // backtrack
17.          if ($l_{j+k}$ is not within the current sub-segment)
18.              examine the very last element in the sub-segment and then leave this inner while loop;
19.          else $j = j + k$, and run simulation for $l_j$ if $cost_j$ is unknown;    // examine $l_j$
20.      endwhile
21.    endfor
22.    Calculate $S_{th}^x$ based on the average of the cost displacements from all the sampled points thus far;
23.    if ($S_{th} \geq S_{th}^x$)    $S_{th} = 2 * S_{th} - S_{th}^x$;    // enlarge $S_{th}$
24.    else    $S_{th} = S_{th} * S_{th} / S_{th}^x$;    // shrink $S_{th}$
25.    if ($L_N$ is reached)    examine $L_N$ and terminate to output $L_B$;
26.    else {  $m$++, $\psi_m = \emptyset \cup L_i$; }
27. endwhile
```

In Lines 22-24, we propose a self-adaptive S_{th} updating scheme by dynamically tracking the smoothness reflected from all the sampled points thus far. The dynamic update of S_{th} is used to reasonably control the next segmentation process for the remaining unvisited L-range in the L-

bound. In Line-22, S_{th}^{x}, as a new smoothness factor, is calculated similarly to S_{th} described in Line-3 but based on all the already sampled points (i.e., including all the elements in ψ_1 to ψ_m and the rest of the N rough sampling points). S_{th} is updated based on the relationship between the current S_{th} and S_{th}^{x}. If S_{th}^{x} is less than S_{th}, such an implied smoother data trend encourages us to try a larger segment in the next round. Thus, we opt to enlarge S_{th} by $S_{th} = S_{th} + (S_{th} - S_{th}^{x})$. Otherwise, we shrink S_{th} (i.e., $S_{th} = S_{th} * S_{th} / S_{th}^{x}$) to establish a smaller segment for conducting fine sampling if the already sampled points pose a bumpier data trend. Once S_{th} is updated, the process would reiterate through Lines 5-27 until the last rough-sampling point, i.e., L_N, is examined.

4.3.3. Parasitic-Aware Circuit Sizing Mechanism

Device sizes can be determined if I_D and any of g_m/I_D, I_{DN}, and node voltages (i.e., V_{GS} and V_{DS}) are available. This has enlightened us to develop an analytic-based sizing methodology that takes node voltages and bias currents as free variables to calculate the device sizes by solving an optimization problem modeled with g_m/I_D and g_{ds}/I_D in a symbolic form. First of all, the initialization module in Fig. 8 provides the initial bias conditions and derives a list of node voltage ranges for all the devices. Alternatively, these conditions as well as the initial L value can be loosely specified by designers in order to speed up the process.

When a MOSFET works in weak inversion, its drain current is given by [70],

$$I_D = 2n\mu C_{ox}' U_T^2 ({}^{W}/_{L})(e^{\frac{V_{GS}-V_{TH}}{nU_T}}) , \tag{32}$$

where n is the substrate factor equal to 1.4 in the weak inversion region, U_T is the thermal voltage equal to 25.9mV, μ is the carrier mobility, and C_{ox}' is the gate oxide capacitance per unit area. I_{DN}

has a dependence on W especially for short-channel devices. This dependence quickly changes with W starting from one small value (e.g., $120nm$) and finally becomes stable. The onset point of the stable region for W (called applicable region hereafter) depends on L. This I_{DN}-versus-W dependence might result in sizing errors if (30) is followed. That is to say, it is not accurate enough to obtain a scaled W by just scaling I_D for the short-channel devices. In this regard, we propose to utilize current density factor to overcome the accuracy problem of the conventional g_m/I_D approaches when dealing with short-channel devices as follows.

The current density for a specific L is defined as (I_D/W). To reflect the sizing error, we define ($I_D/W - I_{D\_REF}/W_{\_REF}$) / ($I_D/W$) as current density error, where W is within the applicable region. According to our experiments, this error may reach 25% for short-channel devices working in the weak inversion region, mainly due to $\mu C'_{ox}$ and then $\frac{V_{GS}-V_{TH}}{nU_T}$ in (32). Moreover, $\mu C'_{ox}$ has a strong dependence on inversion level (i.e., V_{GS}) and then V_{DS}, while V_{TH} is also affected by V_{DS} and V_{GS}. As a consequence, the error that depends on the bias condition always varies. Therefore, we have proposed to use a fitting-based factor as shown in (33) to improve the accuracy of (30),

$$CDF = \frac{{}^{I_D}/_{W}}{{}^{I_{D\_REF}}/_{W_{\_REF}}} = f(V_{GS}, V_{DS})_{CDF}\Big|L\,, \tag{33}$$

where CDF is the current density factor under a specified L. Generally, CDF is greater than 1, and the task of sizing W in (30) should then be modified to

$$W = L * (I_D/CDF)/I_{DN}\,. \tag{34}$$

The onset point of the applicable region (i.e., the smallest applicable W) can be found by observing (I_D/W)/($I_{D\_REF}/W_{\_REF}$) in the weak and moderate inversion regions because the current density error is very stable and almost ignorable in the strong inversion region even for short-channel devices (e.g., 60nm) in our experiments. Pollissard et al. [31] suggested that the short-channel devices with small *W* are not applicable to the conventional g_m/I_D-based approaches due to accuracy concern. However, MOSFETs with very small *W* sometimes may be useful in low-power applications. To overcome this difficulty, we have proposed a scheme by using multiple reference MOSFET widths as further described in Section 4.3.4. Once the abovementioned limitations are resolved by our proposed schemes for *CDF* correction and multiple *W* references, we can apply the g_m/I_D idea into our nonlinear design problem formulation below. Firstly, the minimization-based objective function is defined by,

$$obj = \alpha \sum_{i=1}^{m} \frac{\left(\frac{g_{m_i}}{I_{D_i}}\right) I_{D_i}}{\left(V_{GS_i} - V_{TH_i}\right)} + \beta \sum_{j=1}^{n} intLen_j + \gamma I_{ss} V_{DD} \ , \tag{35}$$

where α, β, and γ are the weighting factors for overall silicon dimension, interconnect length, and power consumption, respectively. Variable *m* is the number of MOSFET devices, *n* is the number of interconnect sections between any two devices, and $intLen_j$ is the length for each interconnect, which will be further explained in Section 4.5.2.

Linear voltage and current inequalities, which are reflected by the relationship between the free variables (i.e., V_{DS} and I_D) and power components (i.e., V_{DD} and I_{SS}) based on circuit structure, can be built up. The operating region constraints for each MOSFET are reflected by a group of relationships among node voltages, threshold voltage, and thermal voltage if the subthreshold region is also considered for pursuing higher g_m/I_D ratio and in turn higher gain. All the performance equations, which are dependent on circuit structure, can be expressed as functions of

g_m/I_D, g_{ds}/I_D, C_{ij}/I_D as well as node voltages of specific MOSFETs, which can be further transformed to inequalities with respect to specifications,

$$f_a(V_{GS}, V_{DS}, \frac{g_m}{I_D}, \frac{g_{ds}}{I_D}, \frac{C_{ij}}{I_D}, I_D) \leq \text{or} \geq \ Spec._a\ , \tag{36}$$

where C_{ij} typically refers to C_{gs}, C_{ds}, and C_{db}. Ratios g_m/I_D, g_{ds}/I_D, and C_{ij}/I_D can be expressed in terms of free variables inclusive of node voltages, while I_{DN} is a function of node voltages as well to help form geometrical constraints with the assistance of Eq. (34).

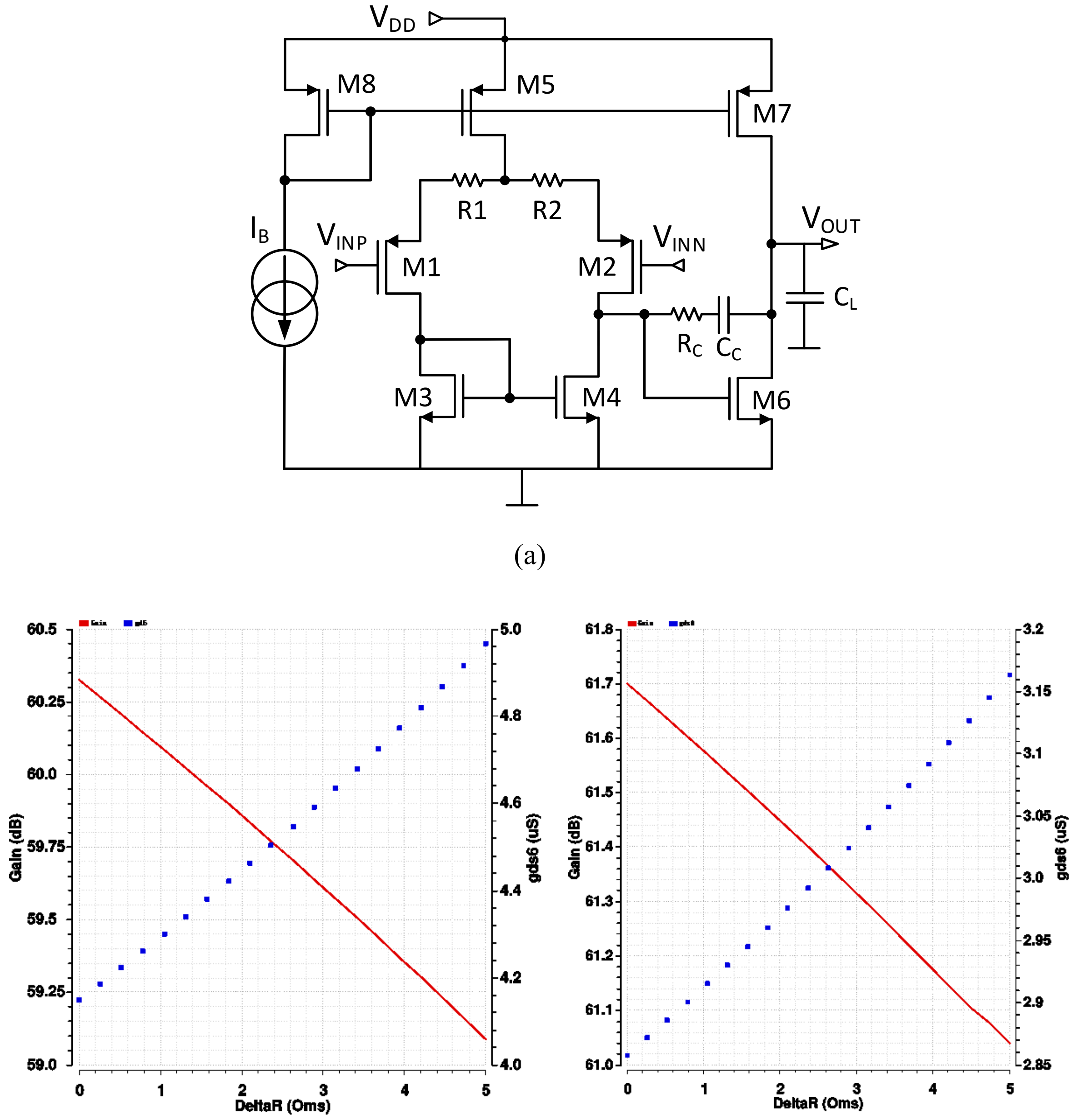
According to (34), W/L of each transistor can be derived once the corresponding node voltages and I_D are solved. That is to say, W can be determined if L is provided. Moreover, instead of the conventional LUT search mechanism used in the previous g_m/I_D-based sizing works, we use curve fitting technique in this research to transform the single-transistor simulation data to nonlinear equations in order to build up a systematic modeling platform that facilitates the inclusion of any special constraints, such as parasitics [57] or other layout-dependent effects [71]. Then the modeled MINLP problem can be solved within one single invocation of a nonlinear programming solver, which is more versatile and efficient than any LUT-based g_m/I_D approaches.

Sizing high-performance analog/RF circuits with any second-order effects (e.g., parasitics) at one time would complicate the MINLP solver. Therefore as exhibited in Fig. 8, we decompose this sizing task by firstly solving the modeled problem without consideration of any interconnect parasitics, a process called *parasitic-free optimization* in this chapter. Then by using the parasitic-free optimization outcome as an initial point, a parasitic-aware optimization process is followed along with an update of resistance and capacitance modeling as (37),

$$R_{total} = ((\frac{g_{ds}}{I_D})I_D)^{-1}\ op\ R_{int}\ ,\quad C_{total} = ((\frac{C_{ij}}{I_D})I_D)\ op\ C_{int}\ , \tag{37}$$

where R_{total} and C_{total} are the total resistance and capacitance, R_{int} and C_{int} are the interconnect parasitic resistance and parasitic capacitance for one electrical net, and *op* is either parallel or serial operator determined by the detailed connection configuration.

In addition, we also include another type of parasitic constraint based on sensitivity analysis in our proposed parasitic-aware optimization, which would prevent certain influential parameters (e.g., g_m or g_{ds} of some MOSFETs) from causing interconnect-parasitic-induced performance degradation. For instance, Fig. 9(a) depicts a widely used two-stage operational amplifier (Op-Amp), while Fig. 9(b) and (c) exhibit the simulation results by using the sizing solutions from the parasitic-free and parasitic-aware optimizations. The resistance mismatch portion, *DeltaR* = R_1 – R_2, is swept for analysis purpose. Parameter g_{ds6} is the drain-source conductance of NMOS transistor M6. In Fig. 9(b) and (c), the red solid curve and the blue dot curve indicate the output of voltage gain and g_{ds6}, respectively, with reference to *DeltaR*. For the parasitic-free sizing result, one can observe that the resistance mismatch (e.g., *DeltaR* increment from 0 to 1.4Ω) due to layout parasitics would lead to a bias voltage change on the operating point and thus an increase of g_{ds6} (e.g., from 4.15μS to 4.35μS), which in turn contributes to a decline of gain output (e.g., from 60.33dB to 60dB). With the help of the sensitivity analysis, we can add a new constraint in the parasitic-aware optimization by restricting g_{ds6} from increasing to an absolute threshold or a variation percentage based on the value solved from the parasitic-free optimization. Thus, in the parasitic-aware sizing result as shown in Fig. 9(c), we can see that g_{ds6} would significantly decrease and the gain output would stay above 61dB even though a 5Ω of *DeltaR* is imposed.



(a)

(b)

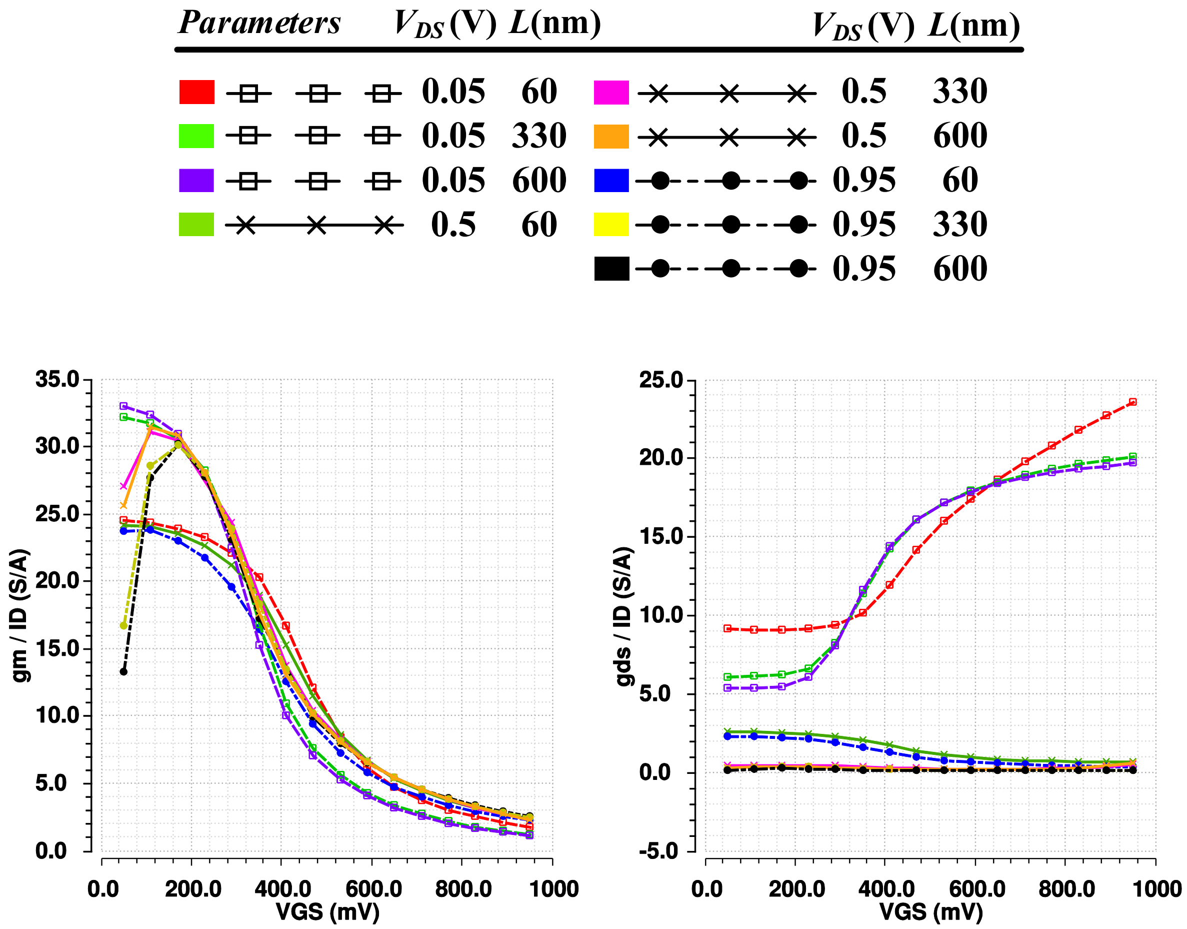
(c)

Fig. 9. (a) Schematic of a two-stage Op-Amp, (b) gain and g_{ds6} output versus *DeltaR* by using the parasitic-free sizing result, and (c) the parasitic-aware one from sensitivity analysis

In [72], semi-empirical models are presented in order to characterize on-chip passive components at a target frequency with consideration of technology variation. For the inductors, these semi-empirical models are generated in the form of look-up-table by sweeping inductor width, turn, and radius via numerical simulations. The resultant relationships are between the inductance and three important inductor characteristics including quality factor (Q_{ind}), parallel resistance ($R_{p,ind}$), and series resistance ($R_{s,ind}$). In our work, we reduce the modeling complexity by selecting discrete amounts of inductor turn & width (considering geometrical and electrical constraints in the applied technology) and sweeping inductor radius. By using the curve fitting technique, multiple symbolic expressions in between inductance and inductor radius can be derived for various combinations of inductor turn and width. Then any semi-empirical models of Q_{ind}, $R_{p,ind}$, and $R_{s,ind}$ can be established as one-to-one correspondence to inductance. They are finally integrated into the mixed-integer nonlinear programming (MINLP) modeling for the subsequent problem solving.

4.3.4. Refined Curve Fitting with V_{GS}, V_{DS}, and L

For sub-100nm technologies, the transistor characteristics, g_m/I_D, g_{ds}/I_D, $C_{ij}/I_{D,}$ and I_{DN}, are strongly affected by V_{DS} and L in addition to V_{GS} [31]. In this work, the relationship between these characteristics and V_{GS} under the effect of V_{DS} and L are fitted into nonlinear expressions and included in our problem formulation. The reference MOSFETs with fixed transistor width W and varying length L under various bias voltages applied among four terminals are used to conduct the simulations. For instance, in the CMOS 65nm technology, the transistor width W is set as 1μm with L ranging from 60nm to 600nm, while both V_{GS} and V_{DS} are swept from 50mV to 950mV. In the g_m/I_D and g_{ds}/I_D curves shown in Fig. 10, the dashed line with squares, solid line with crosses, and dot-dashed line with solid dots represent the configuration of V_{DS} equal to 50mV, 500mV, and 950mV, respectively.



(a) (b)

Fig. 10. (a) g_m/I_D and (b) g_{ds}/I_D versus V_{GS}: 0.05V - 0.95V for regular NMOS devices in the CMOS 65nm technology under the conditions of W=1μm, L: 60nm - 600nm, and V_{DS}: 0.05V - 0.95V

In Fig. 10(a), when V_{GS} is around 180mV as the breakpoint for the subthreshold region, some curves can reach g_m/I_D of around 30.5 S/A. When L is equal to 60nm, g_m/I_D has a monotonic relationship with V_{GS}, while the impact of V_{DS} is not significant. In contrast, when L gets larger,

g_m/I_D presents more reliance on V_{DS} before the subthreshold breakpoint where a smaller V_{DS} leads to a larger g_m/I_D. After this breakpoint, g_m/I_D becomes moderately dependent on V_{DS} and L. As shown in Fig. 10(b), lower V_{DS} leads to a significant increase of g_{ds}/I_D, which would result in lower output impedance. Moreover, smaller L could lead to lower output impedance as a secondary impact. The influence of both L and V_{DS} on I_{DN} and C_{ij}/I_D is also observed but not detailed here.

The observations above imply that V_{DS} and L should be included in the curve-fitting equation (38), which is based on polynomial expressions in this work,

$$y_{cf} = f(V_{GS}, V_{DS})_{cf}\big|L = \sum_{i=0}^{m}\sum_{j=0}^{n} a_{i,j} V_{GS}{}^{i} V_{DS}{}^{j}\ , \quad m, n \geq 1 \tag{38}$$

where y_{cf} is any of g_m/I_D, g_{ds}/I_D, I_{DN}, or C_{ij}/I_D for a given L, $a_{i,j}$ is the constant weighting factor, and m and n are the highest order for V_{GS} and V_{DS}, respectively. It offers a significant improvement compared to the large-signal square-law current equation that is commonly believed inaccurate for advanced technologies. Since these symbolic expressions through the curve fitting process stem from accurate numerical simulations, the accuracy of the whole modeling process in our proposed parasitic-aware g_m/I_D-based sizing methodology merely depends on (36), which is a group of inequalities containing relatively accurate circuit topology-dependent performance equations.

In addition, from Fig. 10, it is also easy to understand that the L parameter is very important in characterizing the g_m/I_D-parameters. For example, a MOSFET with L = 60nm would not be able to produce high g_m/I_D (e.g., over 25 S/A) and low g_{ds}/I_D (e.g., lower than 1 S/A) for pursuing high intrinsic gain regardless of any node voltages. So our proposed performance-driven L-initialization algorithm in the pre-optimization Module-I plays an indispensable role in generating a promising L that would influence the curve fitting and the subsequent MINLP. Furthermore, the solution of node voltages solved from the NLP in Module-I offers an initial bias solution, which can not only facilitate the following MINLP search but also simplify the curve fitting process in (38). For instance, for a MOSFET with high intrinsic gain expectation, an output solution of V_{GS} = 300mV and V_{DS} = 600mV from Module-I could help form smaller voltage ranges of $V_{GS} \in$ [200mV, 400mV] and $V_{DS} \in$ [400mV, 800mV] rather than the primitive full supply voltage range. This can contribute to fewer simulations, simpler and more accurate fitted equations.

For an effective yet practical curve fitting operation, we need to determine proper sampling range and step size of the free fitting variables (i.e., V_{GS} and V_{DS} in (38)). In principle, when V_{GS} and V_{DS} vary from 0 to the power supply voltage V_{DD} in simulations, all biasing inversion levels can be covered. Such curve-fitted equations can cover various inversion regions. The initial bias conditions obtained from Module-I in Fig. 8 would help decrease the bias range for a more focused fitting. Therefore, by using our proposed approach, it is not necessary to specify the inversion level of each MOSFET in a circuit. Yet it is beneficial if having certain knowledge of inversion level and circuit DC biasing constraints especially related to short-channel devices for the sake of improving fitting accuracy and efficiency.

When W is too small to be included in the applicable region, nearly all g_m/I_D-parameters in y_{cf} have dependence on W. Therefore, in order to extend the g_m/I_D-based sizing scheme to the non-applicable region, we have selected several reference W values in the non-applicable region with the same preselected L based on a user-defined tolerance rate (10% by default) for all the g_m/I_D-parameters. Firstly, since the non-applicable region depends on L and device type of MOSFET, each unique combination of L and device type is recorded into a list. Then we perform single-device simulations with default reference width (i.e., 1μm by default) for each item on the list. For every single-device simulation, three bias conditions, which offer the weak, moderate, and strong inversion levels as reflected by inversion coefficient (IC) as per V_{GS} and V_{DS}, are provided.

Secondly, to identify the devices on the list with non-applicable region and specified bounds, for each single device at various *IC* levels, we sweep W from W_{min} and keep monitoring the g_m/I_D-parameters until a large width W_y (e.g., 10μm by default) is reached. We define W_x as a stable width if the following conditions are satisfied: (1) the difference of each g_m/I_D-parameter between W_x and W_y is less than the user-defined tolerance rate in the three *IC* cases; (2) the following several (e.g., 5 by default) W sampling values after W_x still meet condition-1. If such a W_x (not equal to W_{min}) is discovered, then we say this device has a non-applicable region with its tight bound of [W_{min}, W_x]. Lastly, we divide the non-applicable region [W_{min}, W_x] into multiple smaller segments, each of which features relatively stable g_m/I_D-parameters (i.e., within the user-defined tolerance rate) at three *IC* levels. Then for each of the smaller segments, a reference W is used for the curve fitting operation. A bunch of curve fitting expressions, y_{cf_i} in the similar form of (38), are obtained and then connected via (39),

$$y'_{cf} = \sum_{i=1}^{n} B_i\, y_{cf\_i} + B_{n+1} y_{cf} \ , \qquad \sum_{i=1}^{n+1} B_i{=}1 \ , \quad B_i \in \{0{,}1\} \ , \tag{39}$$

where y'_{cf} is any of the g_m/I_D-parameters for W in both applicable and non-applicable regions, y_{cf} is only for the applicable region, and B_i is the binary coefficient (i = 1…n). So by using the piecewise curve fitting technique, Eq. (39) helps extend the g_m/I_D-based sizing method to the conventionally non-applicable W regions, which used to have large sizing error due to strong W dependence. In the worst scenario, inaccurate fitted equations may render the MINLP problems unsolvable. However, the introduced piecewise fitting in (39) normally clear such a concern at the cost of extra simulations. For the entire curve fitting process, the number of the required simulations is expected to be $\sum_{i=1}^{N_L} N_{W\_i} * N_{VGS\_i} * N_{VDS\_i}$, where N_L is the number of the preselected *L's*, N_{W_i} is the number of the selected reference *W's*, N_{VGS_i} is the number of the V_{GS} sampling points, and N_{VDS_i} is the number of the V_{DS} sampling points.

4.3.5. Performance-Driven *L*-Regulation Scheme

To the best of our knowledge, in the literature many existing g_m/I_D-based sizing automation works (e.g., [22] [29] [30]) assumed that *L* is a pre-defined constant although reference [31] acknowledged that *L* is influential to the device performance. References [26] [28] [31] selected *L* with the aid of designers' intervention. In this chapter, we have proposed a new *L*-regulation scheme as follows. Firstly, the sizes of all the MOSFETs are available from the most recent progressive solution. Since the most recently conducted performance verification can report the failure constraints that are symbolically expressed in the corresponding MINLP modeling, this would help identify the influential MOSFETs whose *L*'s need to be regulated. For instance, if the second-stage amplifier fails to meet the specification, only M6 and M7 are involved. Then the sizes of all the MOSFETs are used as the starting point in the *L*-regulation as shown in Module-II of Fig. 8, while only the sizes of the influential MOSFETs are set as variables.

Similar to the sensitivity study conducted in Module-I of Fig. 8, the *L*-bound and step size are provided. For each influential MOSFET, two sub-regions (i.e., [*MinL*, *lastL*) and (*lastL*, *MaxL*]) need to be examined if the last *L* value (i.e., *lastL*) causing a failure is neither *MinL* nor *MaxL*. Otherwise, only one sub-region that is just the *L*-bound but eliminating *lastL* will be obtained. The *L*-regulation is a three-step iterative process. Firstly, for each influential MOSFET, Algorithm 2 is executed in each sub-region respectively, and the *L* with the best cost is found after checking the two sub-regions (or only one sub-region if applicable). Secondly, the array of the best costs for all the influential MOSEFTs is sorted in order to identify the most influential MOSFET and its L. Thirdly, the influential MOSFET found in the previous step is removed from the array by fixing its identified L value for the next iteration. This process iterates until the array turns to empty, where all influential MOSFETs are regulated successively. Our proposed L-regulation scheme makes the effort to tune the length of each influential MOSFET in order to recover the failed constraints with the aid of sensitivity analysis. Therefore, after the L-regulation iterative process, in principle, the L values of all the MOSFETs in the circuit might be different from one another.

4.4. Second-Phase EA Sizing

Even though our proposed g_m/I_D-based approach above includes numerical simulations to perform device characterization that can well consider the applied technology, the topology-dependent circuit performance equations are still an approximation to the real circuit performance that can be accurately obtained through numerical simulation. Therefore, we employ a simulation-involved many-objective evolutionary algorithm (many-OEA) sizing method as the second-phase optimization to further improve the sizing solution by addressing any modeling inaccuracy issues.

As reflected by our proposed flow in Fig. 8, we strive to take advantages of the first symbolic-based sizing phase to benefit the second heuristic-based sizing phase by introducing promising initial solutions into the first population of EA. They include both the output elite solution (ε) from Module-III and any intermediate solutions (φ) that can only satisfy symbolic constraints but fail in the subsequent simulation verification. In case ε cannot be found, a strategy as discussed in Section 4.5.3 can help configure the first EA population fully with φ. We incorporate three pieces of information (i.e., ε derived from the g_m/I_D-based sizing phase (i.e., Module-III), the variable boundaries implied by the locality around ε (called knowledge implication hereafter in this chapter), and φ) into the second-phase EA-based sizing optimization.

Then the initial EA population can be configured by using the following scheme, $NP = N_{\varepsilon} + N_{\varphi} + N_{\xi}$, where *NP* is the user-defined EA population size, N_{ε} is the number of elite solution (1 if existing), and N_{φ} is the number of intermediate solutions. Here N_{ξ} is the number of candidate solutions (ξ) generated within implied variable boundaries by using the following tactics. Firstly, a user-defined percentage (100% by default) is appended to each variable value obtained from ε in order to determine its upper and lower bounds respectively. Then a step size is determined for enumerating possible discrete variable values within the applied knowledge-implied variable range. In our experiments, 10nm and 5nm were used by default as the step sizes of MOSFET width and length, respectively. Next, N_{ξ} candidates representing the locality of ε are randomly generated within the selected variable boundaries. In the special case, where ε does not exist, the variable bounds need to be set to include all intermediate solutions provided from the iterative g_m/I_D-based symbolic sizing phase. And this could lead to a larger search space that might increase the hardship for the convergence of optimal solutions during the EA-phase sizing optimization.

The implied knowledge from the g_m/I_D elite solution ε, which provides a rich resource of locality information as reflected by ξ, can help eliminate unknowledgeable random exploration. As the traditional standalone EA-based sizing methods have no knowledge of the target circuits, their chromosome-variable range is normally set much wider than that of our proposed g_m/I_D-EA scheme equipped with the g_m/I_D elite knowledge in order to avoid missing any potential optimal solutions. This would naturally result in hardship in the subsequent search and optimization. Next, we apply the improved version of θ-DEA discussed in Section 3.3.3 to the circuit sizing problem as the second-phase sizing refinement.

4.5. Parasitic-Awareness in g_m/I_D and EA Sizing

The intrinsic parasitics have been included in our proposed g_m/I_D-based sizing via (36) in Section 4.3.3. In this section, we mainly discuss how layout information is considered for interconnect parasitics in both g_m/I_D-based and EA-based sizing phases.

4.5.1. Floorplan Optimization

To extract interconnect parasitics, the geometric interconnection relationship among all the circuit modules has to be clearly identified. Although a carefully pre-designed floorplan can provide such interconnect information, it usually demands significant expertise and effort from designers. Therefore, we utilize a floorplan optimization method aimed for analog layouts [13], via a B*-tree representation driven by an SA-based engine, to derive a compact floorplan in each iteration shown in Module-III.

We use the parasitic-free sizing solution, which provides definite device geometric information in order to initialize the input modules for the floorplanner. Once a floorplan is obtained, it will be used for estimating interconnect parasitics for the parasitic-aware sizing in Module-III, and input to the following EA sizing phase. The aspects, such as sensible signal flows, resemblance to circuit schematic, electrical and geometric constraints, are the necessary metrics for deriving robust floorplans. An adaptive floorplan variation scheme in the EA phase will be discussed in Section 4.5.3.

For a derived floorplan, the Manhattan distance between two electrical terminals is used to express the length of the shortest path in a symbolic form. As an example, one floorplan of the differential-pair comparator in Fig. 5(b) is shown in Fig. 4 with the presence of interconnects. The transistors M1 & M2, M3 & M4, M7 & M8, and M9-M12 are placed symmetrically to avoid parasitic mismatch [73]. Cartesian coordinates are used to denote the position of devices [74].

The following floorplan constraints are formulated to minimize the total area. The transistors in Fig. 4 are constrained via,

$$w_{mi} + 2 * polyExt + v_{i1} \leq v_{i2}\ , \quad d + v_{i2} \leq v_{j1}\ ,$$
$$l_{mi} * nf_i + (nf_i - 1) * SD + 2 * L_d + h_{i1} \leq h_{i2}\ , \quad d + h_{i2} \leq h_{j1}\ , \tag{40}$$

where w_{mi} is the single transistor finger width, l_{mi} is the transistor length, nf_i is the total number of transistor fingers, *polyExt* is the polysilicon extension over active diffusion area, *d* is the distance between devices, *SD* is the distance between transistor fingers, L_d is the side lateral diffusion length of the source & drain region in the multi-finger structure, v_{i1} and v_{i2} are the vertical coordinates of the *i*th transistor, and h_{i1} and h_{i2} are the horizontal coordinates of the *i*th transistor. Additional constraints can also be included so that the total interconnect parasitics of sensitive nodes can be well restricted. For instance, C_{intp} and C_{intn} of the two output nodes in Fig. 4 should stay equal in the floorplan in order to reduce capacitive mismatch.

4.5.2. Integration of Interconnect Parasitics

Once a floorplan is generated, the interconnect relative location is definite and the interconnect length between any two transistors can be calculated as a function of MOSFET geometry including length, width, finger number, technology parameters, and other user-specified values (e.g., *d*). If the interconnect length is available, the derivation of interconnect parasitic capacitance involving the coupling and fringe components can be done by following the modeling approach in [18]. The interconnect parasitic resistance, R_{int}, can be derived by (*intLen* * ρ) / (*intWid* * *intThick*), where ρ is the sheet resistivity and *intThick* is the thickness of the interconnect layer, both as technology-dependent constants. Those parasitic equations above are of the simple nonlinear form, which can be readily integrated into our proposed g_m/I_D-based sizing framework through (37).

In the subsequent EA optimization phase, the intrinsic parasitics are already considered in the numerical simulations through technology-dependent device models. The interconnect parasitics are calculated in the abovementioned way when *W*, *L*, and *nf* of all the transistors are definite from an evolutionary trial-solution along with its compatible floorplan. Finally, the values of interconnect parasitics are present as resistance and capacitance of electrical nets defined in a netlist called during circuit numerical simulations. Therefore, the second-phase EA sizing discussed in Section 4.4 remains to be parasitic aware.

4.5.3. Compatibility-Aided Adaptive Floorplan Variation

By offering geometric relationship among circuit devices, a floorplan helps induce estimation of parasitics. It tends to be improper to preserve a floorplan template while varying device sizes since the resultant layout might be substantially suboptimal. However, it is costly to enumerate device layout styles and extensively try the size combinations [51]. Therefore, we advocate applying a scheme as reflected in Algorithm 3 for adaptive floorplan variation embedded in the EA sizing phase. We define a metric called *floorplan compatibility* between a given floorplan and a new set of device sizes, which helps determine whether the given floorplan is still good to be reused for the new device sizes.

```
Algorithm 3. The first population configuration in EA with compatibility-aided adaptive floorplan variation
Input: EA population size ($NP$), elite solution and floorplan, $N_\varphi$ intermediate solutions ($\varphi$) and their floorplans
Output: Configured EA population associated with their updated floorplans in the first generation
1. if ($N_\varphi < NP$) {    // the elite solution is available
2.         Form ($NP$-$N_\varphi$-1) chromosomes to introduce elite locality by using the tactics discussed in Section 4.4;
3.         Use $W$, $L$, and $nf$ of each device defined in the elite solution and other technology-defined parameters (e.g.,
           finger distance $SD$) to calculate the geometrical width and height for all the devices;
4.         Calculate the summation of device area, $Area_{dev}$, from all the devices in the elite solution;
5.         Calculate the estimated chip area, $Area_{chip}$, from the bounding-box of the elite floorplan;
6.         Calculate floorplan compatibility $FC$ for the elite solution (along with its elite floorplan) as a reference;
7.         Calculate $Area_{dev}$ for the other ($NP$-$N_\varphi$-1) chromosomes as initialized in Line-2;
8.         Follow the elite floorplan to derive $Area_{chip}$ and $FC$ for the other ($NP$-$N_\varphi$-1) chromosomes;
9.         Decide if the elite floorplan is reusable or not for each of the other ($NP$-$N_\varphi$-1) chromosomes by calculating
           its floorplan compatibility difference. If not, derive a suitable floorplan and calculate $FC$; }
10. Calculate $FC$ for all the intermediate solutions $\varphi$ with known device sizes and floorplans from the input;
```

Given the user-defined EA population size *NP*, if the number of iterations, due to failure in the simulation verification from the preceding g_m/I_D-sizing modules, has reached *NP*, we consider there is no need for further effort seeking an elite solution under the limited resources since sufficient intermediate solutions have already been collected. N_φ denotes the number of the intermediate solutions (φ) that are input to Algorithm 3 along with the elite solution from Module-III if $N_\varphi < NP$. Besides those, the corresponding floorplan is needed for each solution included in the input. The algorithm starts by focusing on the elite solution and generating $N_\xi = (NP - N_\varphi - 1)$ chromosomes for introducing the locality of the elite solution. Then it calculates the geometrical *width* and *height* of each device by using its schematic parameters and technology-dependent parameters (in Line-3). In Line-4, the total device area, $Area_{dev}$, is calculated by the summation of *width*height* for all the devices from the elite solution. In Line-5 the estimated chip area, $Area_{chip}$, can be calculated for the elite solution by using the bounding-box of the elite placement. In Line-

6 the *floorplan compatibility*, defined as $FC = Area_{dev} / Area_{chip}$, is calculated for the elite solution. For each of the rest ($NP - N_{\varphi} - 1$) chromosomes initialized in Line-2, its $Area_{dev}$ can be readily obtained in Line-7, while a *packing* operation is needed to calculate $Area_{chip}$ and FC for each of the other ($NP - N_{\varphi} - 1$) chromosomes by using the elite floorplan in Line-8.

We employ a new term called *floorplan compatibility difference* between chromosomes a and b both attempting a's floorplan, as define below: $FCDiff(a, b) = (FC_a - FC_b) / FC_a$, where FC_a and FC_b refer to the floorplan compatibility amounts of a and b by using a's floorplan. A smaller $FCDiff(a, b)$ represents a better compatibility when chromosome b is reusing a's floorplan. For the first EA generation, if the floorplan compatibility difference between the elite solution and any one from the other ($NP - N_{\varphi} - 1$) chromosomes is less than a user-defined threshold value, $FCDiff_{ref}$ (15% by default), we consider the elite floorplan is still good to be reused. Otherwise, a new floorplan has to be derived for the current chromosome by using our B*-tree-based floorplanner. After that, the bounding-box and compatibility of the current chromosome are updated in Line-9, while in Line-10 the floorplan compatibility FC of each solution in φ is calculated as per the corresponding input floorplan. Algorithm 3 ends by providing the configuration of the first evolutionary generation including the chromosomes (i.e., definite device sizes) and appropriate corresponding floorplans. From the second generation to the end, we always keep tracking the status of the parental floorplans and reuse them if possible. In detail, each chromosome, a, in each generation attempts to reuse its parental floorplan if any good one can be discovered (i.e., if $FCDiff(a\text{'s one } parent, a) < FCDiff_{ref}$). If the floorplans from both parents are all good, the one that offers larger compatibility will be selected for reuse. Otherwise, a new floorplan has to be derived by invoking the B*-tree-based floorplanner with FC updated for the evolution in the next generation.

4.6. Experimental Results

This part is divided into three subsections. The experimental circuits of the two-stage Op-Amp in Fig. 5(a), the comparator in Fig. 5(b), and the Cascode common source LNA in Fig. 5(c) are employed. Subsection 4.6.1 briefly conducts a performance analysis between the parasitic-free and parasitic-aware g_m/I_D-based sizing methods using the Op-Amp as an exemplary circuit. Following the introduction of the experimental setup, subsection 4.6.2 highlights the merits of our proposed g_m/I_D-EA hybrid approach with the adaptive floorplan variation scheme by providing the experimental results compared to some alternative methods. Subsection 4.6.3 illustrates the robustness of our proposed g_m/I_D-EA hybrid sizing approach reflected by satisfactory post-layout simulation results from the real extracted designs. All the experiments in this chapter were conducted in the TSMC CMOS 65nm technology.

4.6.1. Verification of the First-Phase Parasitic-Aware g_m/I_D-Based Sizing

To formulate the MINLP problem in the g_m/I_D form, each g_m, g_{ds} and C_{ij} in the given circuit performance equations will be replaced by g_m/I_D, g_{ds}/I_D, and C_{ij}/I_D, respectively. Only the topology-dependent circuit equations are employed, which eliminates the accuracy concerns caused by any MOSFET square-law-based equations. For integrating the interconnect parasitics, as one example, the total capacitance at the output net in the two-stage Op-Amp circuit is $C_L + ({}^{C_{db}}/_{I_D})_6 I_{D6} + ({}^{C_{db}}/_{I_D})_7 I_{D7} + ({}^{C_{gd}}/_{I_D})_6 I_{D6} + ({}^{C_{gd}}/_{I_D})_7 I_{D7} + C_{int}$, where C_{int} is the interconnect capacitance estimated at the output net. Another example of circuit modeling and integration of interconnect parasitics in the g_m/I_D form for the differential-pair comparator circuit as shown in Fig. 5(b) can be found in [75].

To demonstrate the significance of the pre-optimization module in our proposed synthesis flow, a case study was conducted for the two-stage Op-Amp by using the standalone parasitic-free sizing Module-II along with some user-defined input of node voltages and L's (denoted by "User-Defined" here). We granted L = 120nm, two times of the CMOS65nm technology minimal L, to all the MOSFETs in the circuit. We also doubled the voltage bounds and respected the constraints of all the MOSFET operating regions according to our reference design by using the proposed sizing method (i.e., Module-I + Module-II). The initial node voltages were set by the medians of the corresponding bounds. For testing such a method, in our experiment we had to loose some specifications in order to make the MINLP problem solvable. The experimental results of the User-Defined method include DC gain of 50.24dB, unity gain bandwidth (UGB) of 19.65MHz, phase margin (PM) of 55.69 degrees, and gain margin (GM) of 32.71dB. They are obviously inferior to the performances of our reference design, including DC gain of 60.33dB, UGB of 12.02MHz, PM of 61.02 degrees, and GM of 24.77dB. Thus, one can understand the significance and effectiveness of our proposed pre-optimization Module-I towards the subsequent optimization modules in the synthesis flow as illustrated in Fig. 8.

Table 6. g_m/I_D sizing result verification under mismatch condition for the two-stage Op-Amp

	Spec.	**Ideal**	**With 1.5Ω (5Ω) Mismatch**
Parasitic-free g_m/I_D-based method	**Gain > 60dB**	60.33	59.98 (59.10)
	UGB > 1M	12.02	12.01 (12.07)
	PM > 60°	61.02	61.08 (61.22)
	GM > 10dB	24.77	24.76 (24.72)
Parasitic-aware g_m/I_D-based method	**Gain > 60dB**	61.70	61.51 (61.04)
	UGB > 1M	9.63	9.65 (9.66)
	PM > 60°	61.72	61.75 (61.84)
	GM > 10dB	23.11	23.12 (22.13)

Table 6 presents the pre-layout simulation verification of the sizing results from the standalone parasitic-free and parasitic-aware g_m/I_D-based sizing methods for the two-stage Op-Amp. The column with the title of "Ideal" shows the simulation results with no R_{int} and C_{int} involved. The column with the title of "Mismatch" shows the simulation results when a 1.5Ω or 5Ω resistive mismatch exists between R_1 and R_2 caused by imperfect layout design. Although the sizing result from the parasitic-free g_m/I_D-based method could satisfy the specifications if no layout parasitics are included in the pre-layout simulation, the gain (with 0.35dB drop) failed to meet the 60dB specification if a 1.5Ω mismatch is involved. In contrast, our proposed parasitic-aware g_m/I_D-based method was able to achieve 61.51dB of gain (with only 0.19dB drop that is 45.7% less) in the same situation. Moreover, in the case of the 5Ω mismatch, the parasitic-free g_m/I_D-based method degraded the performance of gain by 1.23dB, while our parasitic-aware g_m/I_D-based method could reduce the degradation to 0.66dB (i.e., 46.3% less). Thus, we can conclude that our proposed parasitic-aware g_m/I_D-based sizing phase is able to derive a preliminary sizing result that can not only reserve some performance margin for absorbing parasitic disturbance but also be more immune from any unexpected parasitic effect caused by subsequent imperfect layout.

4.6.2. g_m/I_D-EA Hybrid Sizing Verification

To have a comprehensive comparison with the previous works, we have implemented the following eight alternative methods. Scheme-0 is the standalone parasitic-aware g_m/I_D-based sizing method as discussed in Section 4.3. Among the single-objective methods, Scheme-1 follows the Sizing Flow for fast Parasitic Closure (called *SFPC* for short) originally proposed in [59], which encloses placement and global routing inside a refined-sizing loop. Scheme-2 implements the idea in [67] that uses a conventional evolutionary algorithm on analog circuit sizing. In order to be comparable with the other parasitic-aware schemes, the parasitic handling scheme proposed in this paper was also applied to Scheme-2. Due to no pre-optimization to generate any knowledge for evolutionary variables in [67], the variable ranges in Scheme-2 had to be set wide for covering large search configuration space, which can be also reflected by a large population size (i.e., $NP = 56$) and a large maximum generation number (i.e., $G_{max} = 40$). Scheme-3 mimics the idea from a layout-aware sizing work [33] by using the differential evolution (DE) algorithm. We grant it with the knowledge from the g_m/I_D phase, like that in our proposed method (i.e., Scheme-5), in order to fairly compare the performance between the single-objective EA and the many-objective θ-DEA. The NP and G_{max} are set as 32 and 20, a smaller configuration for these two schemes, respectively. Scheme-4 is provided as a comparison set in the many-OEA group to show whether a sophisticated many-objective yet standalone EA can derive a good output under the same condition of large evolutionary configuration as in Scheme-2 if no g_m/I_D-involved knowledge is integrated. Scheme-7 is our proposed parasitic-aware hybrid g_m/I_D-EA sizing method, which is integrated with our adaptive floorplan variation scheme. In order to demonstrate its efficiency, we have also introduced two more schemes with the same configuration of Scheme-7 but with fixed floorplan (for Scheme-5) and with full floorplan variation all the time (for Scheme-6).

For each of Schemes 1-7 in our experiment, 10 runs were conducted for each test circuit. Statistical data (e.g., average and standard deviation) were then calculated for our comparison and analysis purpose. To enable a direct comparison between the single-objective and many-objective EAs, in the analysis part we evaluate the many-objective solutions with a unified metric called *fitness* that is calculated from a fitness function defined in (41), the smaller the better, as the ultimate benchmark (as shown in the row of "DE/θ: Best-Fitness" in Tables 7-8).

$$F(\overline{X}) = \sum_{i=1}^{G} u_i \frac{S_i}{P_i} + \sum_{i=G+1}^{H} u_i \frac{P_i}{S_i} + v_1 A_{norm} + \sum_{j=2}^{K} v_j \, T_{j(norm)} \; , \tag{41}$$

where u_i's and v_i's are the weighting factors for different electrical specifications and geometric requirements, respectively. P_i is the circuit performance returned from numerical simulations for solution $\overline{X}$, and S_i is its required specification accordingly. The first two terms on the right side of (41) show reciprocal division between P_i and S_i. In this way, if minimizing $F(\overline{X})$, the performance value, P_i (i = 1 to G, such as open-loop gain), can be maximized, whereas P_i (i = G+1 to H, such as noise figure) can be minimized. Thus, both maximization and minimization of multiple objectives are integrated into one single-objective minimization problem of $F(\overline{X})$, where H is the total number of electrical specifications, and K is the total number of geometrical requirements. A_{norm} is the normalized layout total area and $T_{j(norm)}$'s are the normalized other geometric requirements, which can be weighted by v_j's (i = 1 to K). From our experiments, the selection of the user-defined weighting factors in (41) may slightly alter the sizing performance as a local effect. Whereas the configuration of variable bounds, initial solution point, and the constraints would contribute more to the performance variation. The numbers of SPICE invocations for Schemes 1-7 are reported in Tables 7 and 8. The run time only covers the simulation and optimization process without including any subsequent layout synthesis operations (i.e., layout generation and extraction).

For a complete resultant solution set S_s from any scheme, the specification-satisfied solutions form a subset denoted as $S_{s\text{-}s}$, leaving a complementary subset with failure solutions. Since the nature of the single-objective EA is to converge to a best-fitness solution, the complete set should be used to reflect the evolution status, and the average fitness and standard deviation are calculated inside S_s. However for the many-objective θ-DEA, in order to promote the generation of optimal clusters with the exploration emphasis on multiple objective aspects, the clusters have to be distributed across the entire solution space even in certain infeasible regions directed by the systematic construction of reference points. Therefore, the average fitness is calculated inside $S_{s\text{-}s}$ only, which refrains from presenting unaccountable solutions from any infeasible regions. Moreover, we employ a success-rate (i.e., $S_{s\text{-}s}/S_s$) to exhibit the diversity of the final solution set for the many-objective methods.

The experimental results of the differential-pair comparator are listed in Table 7. Propagation delay is one of the most important characteristics for this comparator circuit, and the positive and negative overshoots are given with the absolute values. Without the help of the elite or intermediate solutions from the g_m/I_D phase, the single-objective SFPC (i.e., Scheme-1) and single-objective EA method (i.e., Scheme-2) demonstrate poor best-fitness and average-fitness (i.e., 0.828 & 0.840 and 0.707 & 0.712, respectively). As an alternative, although Scheme-3 slightly improves the best-fitness by integrating the informative first-phase solutions from Scheme-0, it still fails for the delay specification. In contrast, the performance from the many-objective methods is definitely superior to that of all the single-objective methods. Since the output logic of the comparator frequently flips when the parasitic capacitance fluctuates during the solution exploration to break the balance between two output paths, the search configuration space might be highly bumpy. Therefore, the knowledge from the g_m/I_D phase comprising the g_m/I_D elite and other intermediate solutions, which is integrated into Scheme-7, can effectively help shrink the search configuration space with a better focus on the promising regions. As a result, our proposed Scheme-7 despite fewer resources involved not only ran faster but also performed better than Scheme-4 especially in terms of the success rate (i.e., 78.57% vs. 30.77%).

Table 7. Settings and performance of the various schemes for the differential-pair comparator

Schemes	***g_m/I_D***	**Single-objective Methods**			**Many-objective *θ*-DEA Methods**			
	Sch-0	**Sch-1 (SFPC)** [59]	**Sch-2** [67]	**Sch-3** [33]	**Sch-4 Larger setting**	**Sch-5 Fixed Fp.**	**Sch-6 Var. Fp.**	**Sch-7 [This work]**
EA/*θ*: Best-Fitness	0.496	0.828	0.707	0.483	0.267	0.224	0.255	0.171
***θ*: Average-Fitness**	-	-	-	-	0.320	0.284	0.309	0.183
***θ*: Success-Rate**	-	-	-	-	30.77 %	82.14 %	78.57 %	78.57%
EA: Average-Fitness	-	0.840	0.712	0.513	-	-		-
EA: Standard-Deviation	-	0.013	0.005	0.021	-	-		-
#SPICE Invocations	-	560	2240	640	2240	640	640	640
Run Time(mins)	3.16 Sec.	13.27	28.27	12.15	37.72	12.52	17.18	14.90
Specification	**Performance (from the Representative Solution with the Smallest Fitness)**							
Propagation Delay < 250ps	152.5	493	279	320	175	108	95.88	83
+Overshot < 350mV	183.0	400	220	20	11	37	61.39	23
-Overshot < 150mV	53.3	89	57	13	10	20	30.73	18
Area(μm^2)	153.29	244.17	288.35	252.04	275.36	154.06	136.30	180.67

By maintaining a floorplan template, the sizing result from Scheme-5 tends to be suboptimal compared to that from Scheme-7 (i.e., best-fitness of 0.224 vs. 0.171) since the fixed floorplan template might be erroneous to follow for non-scaled device sizes. In addition, for Scheme-6, an offspring solution may not be able to readily continue the floorplan style used by its parents due to dramatic floorplan variation between individuals. In particular, any offspring solutions with scaled device sizes compared to their parents may be equipped with very different floorplans, which would unnecessarily add hardship to the evolution process due to non-scaled parasitics. However, our proposed Scheme-7 endeavors to focus more on solutions with cooperative floorplans by introducing the concept of floorplan compatibility. It would not only effectively reduce the search configuration space, but also avoid awkward mismatch situations between device sizing and floorplan. As reflected from the experimental data, under limited evolution resources (i.e., small configuration of evolutionary population size and maximum generation), our proposed Scheme-7 could still improve the best-fitness to 0.171 on top of 0.255 obtained from Scheme-6.

For the experimental results of LNA in Table 8, the SFPC scheme could not work well with a lot of specification failures due to the following reasons. Firstly, the single-objective EA might not have enough strength in handling hard problems like LNA (partially due to the nonlinearity of inductors). Secondly, the frequently changed floorplans would bring forth oscillating parasitics, which can hardly provide a priori informative guidance to the next refined-sizing iteration. Similarly, the single-objective EA method Scheme-2 could not deliver a good result (i.e., best-fitness of 1.058) even though it is equipped with extra evolutionary resources. In addition, the high average-fitness and standard-deviation (i.e., 2.649 and 1.654, respectively) from Scheme-3 shows that it requires a longer generation to converge. In contrast, the best-fitness and average-fitness attributes are much better for Schemes 5-7 with the same level of CPU time consumed in comparison to the others. Under the two big categories of many-objective schemes, Scheme-4 produced a relatively premium best-fitness solution of 0.787 with a little over 0.2 mm^2 area cost yet at the cost of almost triple execution time compared to Schemes 5-7, which indicates that the capability of θ-DEA is not fully exploited when there is no specific circuit knowledge offered. In addition, since the size of an inductor is much bigger than the other types of devices, the attempted floorplans for all the chromosomes may only have some local variations but with a major global resemblance. This partially explains why the percentage difference of best-fitness could only reach 7.8% between Scheme-6 and Scheme-7, but it was improved by 32.9% between these two schemes for the comparator test circuit even though both circuits are sensitive to parasitics. Therefore, our proposed adaptive floorplan variation scheme (i.e., Scheme-7) could excel beyond the other two (i.e., Scheme-5 and Scheme-6) in terms of circuit performance by maintaining a good tradeoff between floorplan consistency and suitability.

Table 8. Settings and performance of the various schemes for the LNA circuit

Schemes	**g_m/I_D**	**Single-objective Methods**			**Many-objective θ-DEA Methods**			
	Sch-0	**Sch-1 (SFPC)** [59]	**Sch-2** [67]	**Sch-3** [33]	**Sch-4 Larger setting**	**Sch-5 Fixed Fp.**	**Sch-6 Var. Fp.**	**Sch-7 [This work]**
EA/θ: Best-Fitness	0.814	1.158	1.058	0.813	0.787	0.764	0.793	0.731
θ: Average-Fitness	-	-	-	-	0.838	0.803	0.828	0.807
θ: Success-Rate	-	-	-	-	5.36%	6.25%	6.25%	9.38%
EA: Average-Fitness	-	1.340	1.058	2.649	-	-	-	-
EA: Standard-Deviation	-	0.347	0.0003	1.654	-	-	-	-
#SPICE Invocations	-	560	2240	640	2240	640	640	640
Run Time(mins)	3.17 Sec.	11.31	23.85	9.94	32.26	11.09	12.67	11.71
Specification	**Performance (from the Representative Solution with the Smallest Fitness)**							
Gain > 15dB	21.21	15.62	18.75	20.05	21.45	18.15	18.17	20.47
NF < 2.5dB	2.04	2.71	2.15	2.08	2.06	2.29	2.20	1.90
S11 < -10dB	-13.45	-7.33	-5.26	-13.52	-15.56	-12.76	-12.77	-17.08
S22 < -10dB	-10.10	-8.17	-14.88	-10.74	-10.20	-18.85	-14.60	-11.80
Area(mm^2)	0.189	0.216	0.228	0.214	0.209	0.158	0.181	0.154

These experimental results demonstrate that our proposed many-objective θ-DEA optimization search equipped with the knowledge from the g_m/I_D phase is the best choice out of all the alternatives not only in terms of performance but also in the view of computational efficiency. The experimental results of the two-stage Op-Amp, which exhibit similar comparison effect among all the alternative methods as Tables 7-8, are not detailed in this chapter. Finally, by following the g_m/I_D-EA two-phase hybrid sizing results and their final floorplans, we used Cadence Layout-XL tool [56] to place and route the auto-generated modules in order to obtain the final layouts. Then we used Mentor Graphics Calibre tool [76] for parasitic extraction and Cadence Spectre circuit simulator for post-layout performance verification.

4.6.3. Post-Layout Verification for the Proposed g_m/I_D-EA Hybrid Sizing Solutions

In this section, the optimized designs from our proposed parasitic-aware hybrid sizing method are laid out and verified in post-layout simulations. For fair comparison, some intermediate optimization results are also laid out and verified in the same manner. Here we mainly use two examples (i.e., two-stage Op-Amp and LNA) to demonstrate the effectiveness of our proposed g_m/I_D-EA hybrid sizing methodology.

Table 9 exhibits both pre-layout simulation verification (i.e., Columns 3 and 4) and post-layout verification (i.e., Column-5) for the two-stage Op-Amp sizing results from the standalone parasitic-free sizing method (i.e., Row-(A)) and the two-phase hybrid sizing method with parasitic-awareness (i.e., Row-(B)). As for the verification settings, there is no interconnect parasitics present in the "No Parasitic" case (i.e., Column-3), while there are floorplan-based estimated parasitics back annotated (denoted by "Estimated Parasitics") into the corresponding electrical nets in Column-4. After the designs have been laid out along with the extraction of parasitics, the simulation setting would include the extracted parasitics (denoted by "Extracted Parasitics") in Column-5. Moreover, the Bode plots that exhibit the frequency response for the three sizing solutions as reported in

Table 10 are given in Fig. 11. Since both the g_m/I_D-based sizing Module-III and the EA-based sizing Module-IV feature parasitic awareness, the estimated parasitics are included in the frequency response as shown in Fig. 11(b) and (c), while no parasitic is included when obtaining Fig. 11(a) for the parasitic-free g_m/I_D-based sizing Module-II.

Table 9. Verification of sizing results for the parasitic-free sizing method and the proposed two-phase hybrid parasitic-aware sizing method with no parasitics, estimated parasitics, and extracted parasitics for the two-stage Op-Amp circuit

Op-Amp g_m/I_D Schemes & Performance	**Spec.**	**No Parasitics**	**Estimated Parasitics**	**Extracted Parasitics**
(A). Parasitic-free Symbolic g_m/I_D (Module-II)	**Gain > 60dB**	60.33	60.35	59.78
	UGB > 1M	12.02	12.02	11.30
	PM > 60°	61.02	60.64	60.28
	GM > 10dB	24.77	24.43	24.17
(B). Parasitic-aware Symbolic g_m/I_D + Heuristic Many-OEA (Module-IV)	**Gain > 60dB**	63.71	63.69	63.42
	UGB > 1M	27.27	27.22	26.51
	PM > 60°	106.1	105.1	100.0
	GM > 10dB	16.84	16.76	15.69

Table 10. Sizing solutions from various sizing methods without and with parasitic awareness for the two-stage Op-Amp circuit

Sizes / Methods (Modules)		**A. Parasitic-free (Module-II)**	**B. Parasitic-aware (Module-III)**	**C. Parasitic-aware (Module-IV)**
M1/M2	**L (μ)**	0.3	0.3	0.98
	W (μ)	82.18	70.89	67.9
M3/M4	**L (μ)**	0.6	0.6	0.505
	W (μ)	5.3	5.14	5.54
M5	**L (μ)**	0.3	0.3	0.705
	W (μ)	16.38	13.38	47.25
M6	**L (μ)**	0.6	0.6	0.34
	W (μ)	17.3	17.89	24.96
M7	**L (μ)**	0.3	0.3	0.605
	W (μ)	1.36	1.13	128.08
M8	**L (μ)**	0.3	0.3	1.23
	W (μ)	1	1	11.48

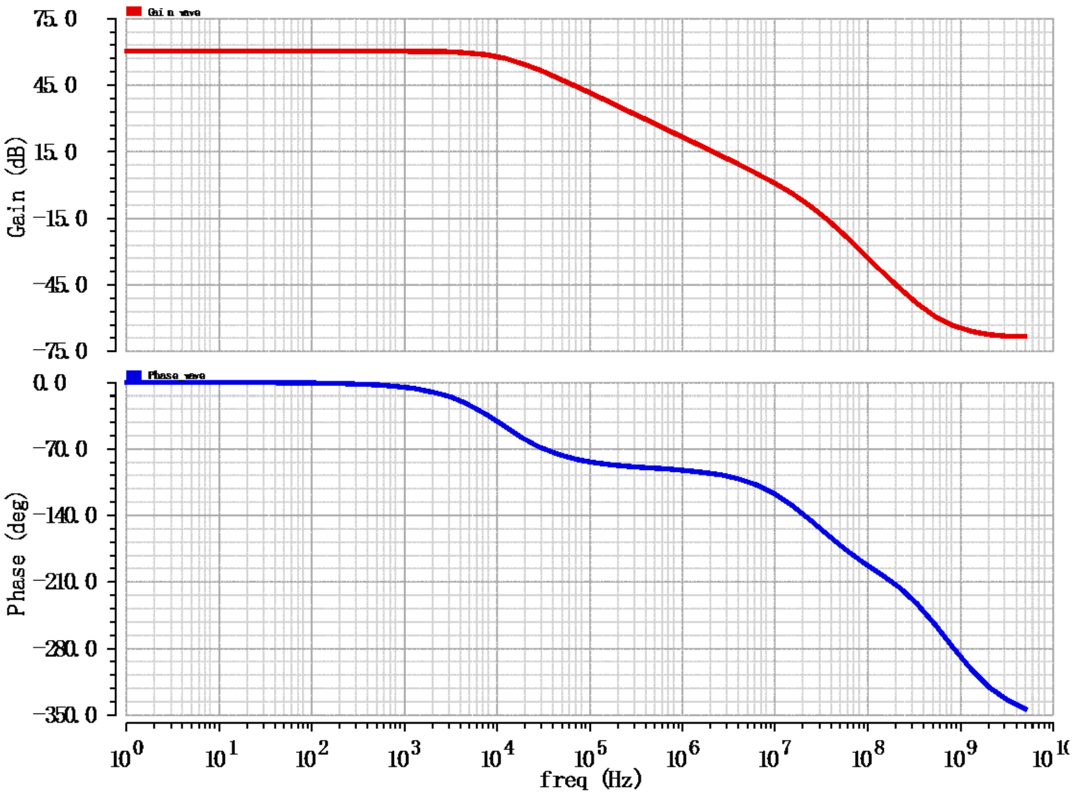
As reported in

Table 10 for the optimized device sizes before and after the parasitics consideration in the two-stage Op-Amp, the parasitic-aware standalone g_m/I_D-based sizing approach yields the solution in Column-B based on the solution (as listed in Column-A) from the parasitic-free sizing method. Aside from 13.7% and 18.3% differences for M1/M2's W and M5's W respectively, there is a high resemblance of device sizes between the two solutions. By referring to the reported performance under the “Ideal”-Column in Table 6, one can see the gain increase from 60.33dB to 61.70dB, which can be also seen through Fig. 11(a) and (b).

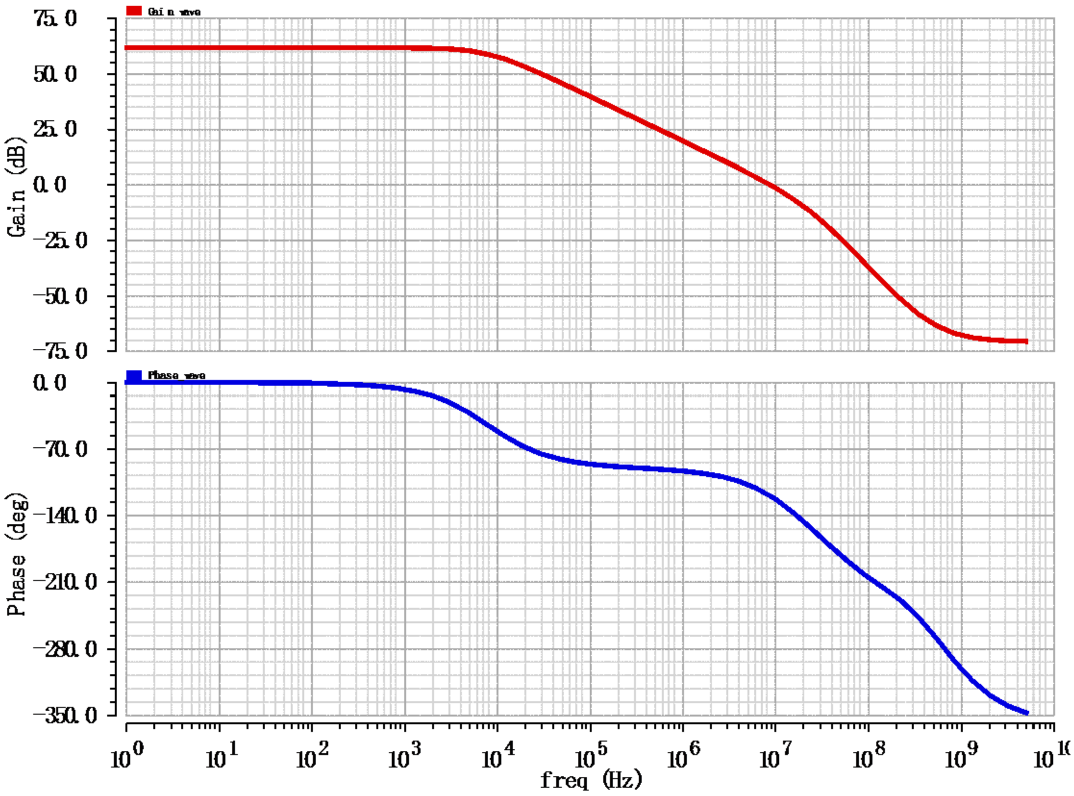
After the elite solution from Module-III being refined via the parasitic-aware EA-phase sizing, a different solution is obtained in Column-C of

Table 10. As reported in Tables 6 and 9, the solution from the EA sizing phase could improve the performances of Gain from 61.70dB to 63.69dB, UGB from 9.63MHz to 27.22MHz and PM from 61.72° to 105.1° at the cost of GM dropping from 23.11dB to 16.76dB. This improvement should be credited to the EA heuristics via the size adjustment of M5-M8. Moreover as observed from Fig. 11(c), the phase curve does not start to drop significantly before the Gain curve reaches the unit gain, which leads to ample PM (i.e., 105.1°) in comparison to those (i.e., 61.02° and 61.36°) in Fig. 11(a) and (b) from both g_m/I_D-based symbolic sizing approaches.

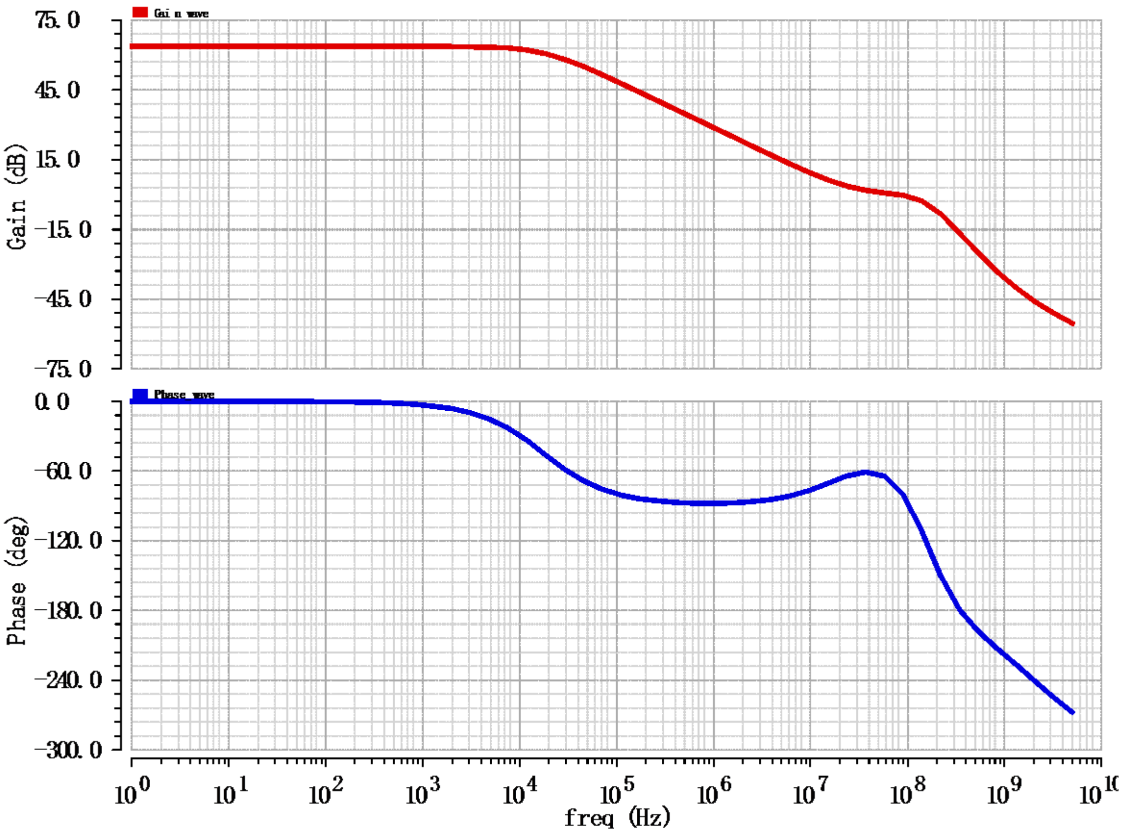
Fig. 12(a) and (b) depict the final layouts from the two sizing methods reported in Table 9. Obviously, the extracted parasitics included in the post-layout simulation make the DC gain of Fig. 12(a) fail in its specification of 60dB, whereas the layout of Fig. 12(b) from our proposed two-phase hybrid parasitic-aware sizing method performs much better with sufficient margins in its post-layout simulation.


Gain wave
75.0
45.0
15.0
-15.0
-45.0
-75.0
Gain (dB)
Phase wave
0.0
-70.0
-140.0
-210.0
-280.0
-350.0
Phase (deg)
10^0
10^1
10^2
10^3
10^4
10^5
10^6
10^7
10^8
10^9
10^{10}
freq (Hz)

(a)

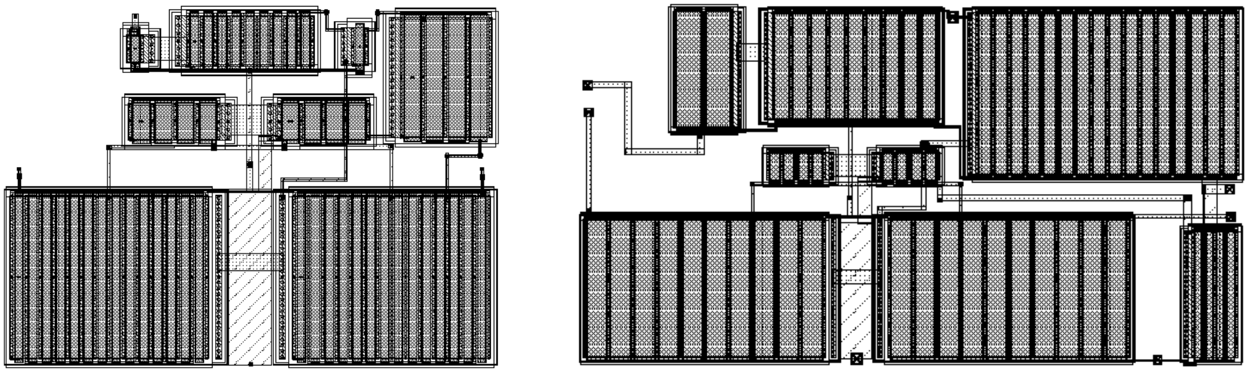

Gain wave
75.0
50.0
25.0
0.0
-25.0
-50.0
-75.0
Gain (dB)
Phase wave
0.0
-70.0
-140.0
-210.0
-280.0
-350.0
Phase (deg)
10^0
10^1
10^2
10^3
10^4
10^5
10^6
10^7
10^8
10^9
10^{10}
freq (Hz)

(b)



(c)

Fig. 11. Frequency response Bode plots of the two-stage Op-Amp for (a) parasitic-free g_m/I_D-based sizing method with no parasitics, (b) parasitic-aware g_m/I_D-based sizing method with estimated parasitics, (c) parasitic-aware g_m/I_D-EA hybrid sizing method with estimated parasitics



(a)

(b)

Fig. 12. Layouts of sizing solutions from (a) parasitic-free g_m/I_D-based sizing method and (b) our proposed two-phase hybrid parasitic-aware sizing method (i.e., g_m/I_D-based plus EA-based) for the two-stage Op-Amp circuit

Table 11. Verification of various sizing results with no parasitics, estimated parasitics, and extracted parasitics for the LNA circuit

LNA g_m/I_D Schemes & Performance	**Spec.**	**No Parasitics**	**Estimated Parasitics**	**Extracted Parasitics**
(A). Parasitic-free Symbolic g_m/I_D (Module-II)	**Gain > 15dB**	22.35	21.11	16.17
	NF < 2.5dB	2.02	2.06	1.938
	S11 < -10dB	-11.09	-7.51	-6.68
	S22 < -10dB	-12.61	-9.1	-3.40
(B). Parasitic-aware Symbolic g_m/I_D (Module-III)	**Gain > 15dB**	22.13	21.21	17.29
	NF < 2.5dB	1.99	2.04	1.87
	S11 < -10dB	-16.56	-13.45	-9.20
	S22 < -10dB	-24.08	-10.10	-4.04
(C). Parasitic-aware Symbolic g_m/I_D + Heuristic Many-OEA (Module-IV)	**Gain > 15dB**	20.04	20.47	18.91
	NF < 2.5dB	1.91	1.90	1.91
	S11 < -10dB	-19.00	-17.08	-10.29
	S22 < -10dB	-11.68	-11.80	-10.23

To further demonstrate the effectiveness of our proposed hybrid sizing method on RF circuits, the post-layout analysis is conducted on the LNA circuit. Table 11 reports both pre-layout simulation verification (i.e., Columns 3 and 4) and post-layout verification (i.e., Column-5) for the LNA sizing results from the standalone parasitic-free sizing method (i.e., Row-(A)), parasitic-aware g_m/I_D-based sizing method (i.e., Row-(B)), and the continued method after the concatenation of heuristic many-OEA sizing (i.e., Row-(C)). The columns are arranged in the same way as those in Table 9. All the three sizing results can meet the specifications initially as shown in Column-3, while only the sizing solutions with parasitic awareness (i.e., Row-(B) and Row-(C)) can pass the specifications if involving estimated parasitics. A certain amount of performance degradation can be observed between the "No Parasitic" and "Estimated Parasitics" cases for both the symbolic-based parasitic-free and parasitic-aware methods, for instance, -11.09dB deteriorates to -7.51dB for S11 in Row-(A) while -24.08dB deteriorates to -10.10dB for S22 in Row-(B). One can also observe more performance degradation between the estimated (i.e., Column-4) and extracted (i.e., Column-5) parasitics for both Row-(A) and Row-(B), none of which can pass the post-layout verification. However, the sizing result from our proposed parasitic-aware two-phase hybrid sizing method (i.e., Row-(C)) can achieve better performance convergence and meet all the specifications in the post-layout verification without fail.

4.7. Summary

In this chapter, an efficient parasitic-aware two-phase g_m/I_D-EA hybrid circuit sizing methodology for high-performance analog and RF circuits was presented. It utilizes the g_m/I_D and curve fitting design techniques to represent the parasitic-aware sizing problem in the mixed-integer nonlinear programming form by enclosing the technology-independent circuit structure models, technology-dependent device intrinsic parasitic characterization, and interconnect parasitic models in order to seek a global solution. Then in the second optimization phase, a many-objective θ-dominance-based evolutionary algorithm is adopted for a more focused and refined search under an informative guide implied from the knowledge generated by the g_m/I_D sizing phase. With the adaptive floorplan variation scheme, the sizes and parasitics are harmoniously optimized in the EA sizing phase. The experiments on several analog and RF circuits demonstrate the efficacy of our proposed methodology compared to several well-known published alternatives.

In the next chapter, aside from the parasitics, LDE that is identified as an emerging challenge for analog circuit synthesis will be firstly discussed. Then our proposed g_m/I_D-based LDE-aware sizing methodology will be presented in detail.

Chapter 5 An LDE-Aware g_m/I_D-Based Hybrid Sizing Method for Analog Integrated Circuits

5.1. Introduction

As the complementary metal oxide semiconductor (CMOS) technology advances, Layout-Dependent Effects (LDEs) have become increasingly more influential to performance of custom and analog integrated circuit design [71]. This is because the surrounding layout around a device might change the behavior of its fine-grained model constructed originally for an isolated state. This is especially prominent in the sophisticated nanometer technologies. Thus, the analog layout designers, although being aware, are often heavily burdened due to either lack of knowledge passed along from the schematic-level circuit designers or intricacy of handling LDE constraints. In turn, a prolonged re-design cycle is typically expected since the LDE-incurred problems may unfortunately not emerge until the final signoff check in the worst scenarios.

LDEs that have been identified as a prominent second-order effect might readily lead to circuit malfunction if not being properly taken care of. For instance, in terms of MOSFET finger number (*nf*), certain transistor key parameters (e.g., carrier mobility and threshold voltage (V_{th})), which are affected by the *nf*-incurred LDEs, may in turn ruin the circuit performance. Therefore, consideration of the LDEs preferably in the early design stage becomes indispensable in the advanced technologies.

In this chapter, we propose a two-phase hybrid sizing method for high-performance analog circuits. It consists of g_m/I_D-based device characterization, circuit modeling, sensitivity-based constraints for LDEs, and mixed-integer nonlinear programming in the first phase, and a refined θ-DEA [38] in the second phase. The main contributions of this chapter are summarized as follows,

- This is the first work that can optimize LDE parameters at the fast schematic level in the synthesis stage of analog design by using our g_m/I_D-based sizing framework.
- The accuracy of the whole methodology is enhanced by adopting technology-variant curve-fitted device characterization from SPICE simulations and sensitivity-study-aided curve-fitting models for the LDE parameters. Thanks to the involved simulation, it is very friendly to target newer technologies, which is not for those that use traditional device models like GeoP.
- To the best of our knowledge, this is the first work that applies many-OEAs in the subject of LDE optimization. Moreover, our proposed model in the second EA sizing phase can offer more accurate estimation of device geometrical parameters at the schematic level.

The research work conducted on this topic has been mainly published in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD) [J2] and Journal of Microelectronics and Solid State Electronics [J5], and presented in 2018 IEEE International Symposium on Circuits & Systems (ISCAS) [C1] among others [C6].

5.2. Proposed LDE-Aware Hybrid Synthesis Flow

5.2.1. Preliminary of Layout-Dependent Effects

The impact of LDEs on MOSFET characterization includes mobility, velocity saturation, V_{th}, body effect, drain-induced barrier lowering effect for STI, and V_{th}, mobility, and body effect for WPE [5]. However, it is not necessarily true that minimizing the LDEs would surely yield the best circuit performance [77]. As a solution at the circuit schematic level, we have proposed a sensitivity-analysis-based approach that can study device features through circuit simulation to constrain the search space $\mathcal{X}$ for the sizing variables (W, nf) and (LR_{ext}, SC_t), denoted by $\mathcal{X}(W, nf)$ and $\mathcal{X}(LR_{ext}, SC_t)$ respectively, for the LDE matters during the sizing optimization.

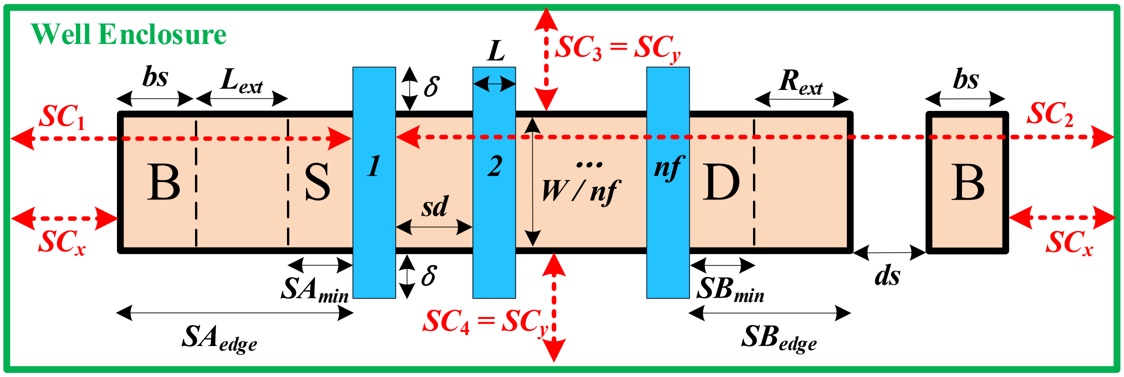


Fig. 13. Illustration of STI and WPE parameters for a multi-finger structure MOSFET with integrated bulk style (left) and detached bulk style (right)

Fig. 13 depicts those geometrical parameters of STI and WPE for a multi-finger MOSFET that covers both integrated bulk connection style and detached style. Coefficients SCA (first-order), SCB, and SCC for each MOSFET, which are included in a simulation netlist, can reflect WPE in advanced CMOS technologies according to the BSIM model. They are functions of W, L, nf, and SC_t (i.e., distances from well edges to polysilicon gate edges from various directions) [5]. For instance,

$$SCA_i = \frac{1}{W_{drawn}L_{drawn}} * \left[SC_{Ref}^2 \sum_{t=1}^{N} \left(W_t \left(\frac{1}{SC_t} - \frac{1}{SC_t + L_{drawn}}\right)\right) + SC_{Ref}^2 \sum_{t=N+1}^{N+M} \left(L_t \left(\frac{1}{SC_t} - \frac{1}{SC_t + W_{drawn}}\right)\right)\right]_i , \quad (42)$$

$$SCA_{eff} = \sum_i^{nf} SCA_i \,/nf ,$$

where SC_{Ref} is a technology-dependent constant (e.g., 1µm for CMOS 65nm). W_{drawn} (i.e., W/nf) and L_{drawn} (i.e., L) are the channel width per finger and channel length, respectively. Depending on the shape of well enclosure, the perimeter (i.e., 2*W_{drawn}+2*L_{drawn}) of the MOSFET channel can be decomposed into segmental widths' (W_t) and lengths' (L_t) such that 2*$W_{drawn} = \sum_{t=1}^{N} W_t$ and 2*$L_{drawn} = \sum_{t=N+1}^{N+M} L_t$, where N and M are the numbers of well edge segments from all directions projected onto W_{drawn} and L_{drawn}, respectively. For the regular rectangular well enclosure (i.e., $N = M = 2$), $W_t = W_{drawn}$ and $L_t = L_{drawn}$. And there are four distance values of SC_t, t = 1,…,4, in the four directions (i.e., left, right, top, and bottom) of the rectangular device as depicted in Fig. 13. In the case of irregular well shape, SC_t, which is perpendicular to the corresponding part of MOSFET body (i.e., W_t or L_t), is the distance between one well edge segment and its corresponding MOSFET channel edge in certain direction. SC_x and SC_y are set as the optimization variables solely accounting for WPE, which will be detailed in Section 5.3. SCA_{eff} is the effective *SCA* for the entire multi-finger transistor. The consideration of *SCB* and *SCC* can be similarly managed if higher computation resolution is required.

For the STI effect, the stress distribution and the incurred effects can be expressed by functions of two symmetrical parameters, *SA* and *SB*, which are the distances from the polysilicon gate edges to the device isolation edges on both sides. *SAB* uniformly expresses *SA* or *SB* since a device is usually self-symmetric. For a multi-finger MOSFET, SAB_{eff} denotes the effective *SAB* which is calculated by averaging *SAB* from all fingers, and specifically, SAB_{edge} denotes the *SAB* only for the edge finger that is placed closest to the MOSFET isolation edge given as follows,

$$SAB_{edge} = SAB_{min} + LR_{ext} + k_i * bs \ ,$$

$$SAB_{eff} = \frac{\sum_{i=1}^{nf}[SAB_{edge} + (L+sd)*(i-1)]}{nf} \ , \tag{43}$$

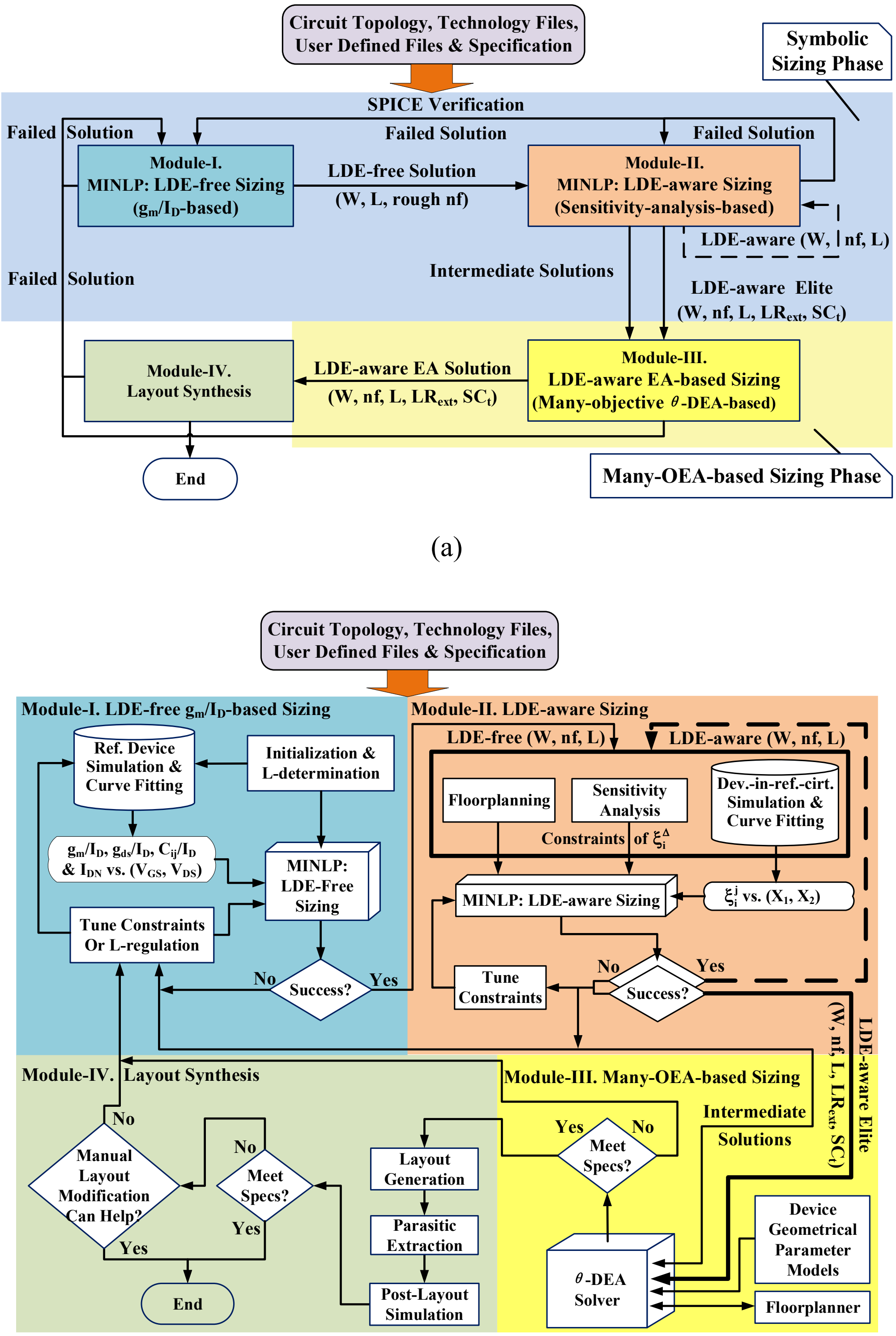
where SAB_{min} is the allowable minimum distance for *SAB*, LR_{ext} uniformly expresses the left (i.e., L_{ext}) or right lateral diffusion extension (i.e., R_{ext}), k_i is 1 if the bulk area abuts the source area (i.e., the integrated bulk style), and 0 otherwise. In Fig. 13, *ds* is the space left between the diffusion and the detached bulk in the *L*'s direction, and *bs* is the size of the bulk in the *L*'s direction. Parameter *sd* defines the finger space and *nf* is the number of fingers. The remaining parameters presented in Fig. 13 which are in linear relationships with the abovementioned ones will be introduced accordingly in the following sections.

5.2.2. LDE-Aware Two-Phase Circuit Synthesis Flow

Our proposed two-phase synthesis flow shown in Fig. 14 is comprised of four main modules. In the symbolic sizing phase as depicted in Fig. 14(a), Module-I and Module-II, which model the circuit sizing problem by using mixed integer nonlinear programming (MINLP), generate the LDE-free and LDE-aware solutions, respectively. In the many-OEA-based sizing phase as shown in Fig. 14(a), the heuristic many-OEA-based sizing Module-III takes the initial solutions from the previous module and seeks refined LDE-aware solutions. The device parameters and other layout factors optimized with LDE-awareness, which are included in Module-III's output solutions, will be used to perform the layout synthesis through placement and routing inside Module-IV. The flow iterates if failures emerge, and a more detailed diagram is depicted in Fig. 14(b).

The first module, LDE-free g_m/I_D-based sizing, starts with the initialization and L-determination (previously discussed in Section 4.3.2), which can generate initial biases and bounds for the following MINLP problem, and a uniform L for all MOSFETs. This initial L is used for the reference MOSFET (with 1µm width by default and varying node voltages) in the subsequent simulations and curve fitting process, where device characterization as reflected by g_m/I_D, g_{ds}/I_D, C_{ij}/I_D, and normalized drain current $I_{DN} = I_D/(W/L)$ (altogether called g_m/I_D-parameters hereafter) with respect to V_{GS} and V_{DS} is extracted and curve-fitted into the nonlinear form. These fitted characterization terms will be used in the objective function and constraints built for any MINLP problems, whose solving is called LDE-free sizing that employs the g_m/I_D-based circuit sizing approach discussed in Section 4.3.

Performance of a sized circuit without LDE considerations might significantly change after they are enabled in the numerical simulation environment. So Module-II is to perform an LDE-aware optimization process based on the LDE-free sizing solution obtained from Module-I. We firstly replace the independent design variables (i.e., node voltages) and the intermediate variables (i.e., g_m/I_D-parameters) in Module-I with symbolically expressed (X_1, X_2) and ξ_i (formally defined in Section 5.3) in Module-II. Since the MOSFET device parameters of W (i.e., X_1) and nf (i.e., X_2) play an important role not only in affecting LDEs but also in device characterization, they are optimized in the first round of Module-II. It is followed by the second round (indicated by the dash arrows in Fig. 14) to further optimize the other LDE-related geometrical parameters of LR_{ext} and SC_t by symbolically involving them into SA/SB (i.e., X_1) and SCA (i.e., X_2), which are present in the simulation netlist for reflecting LDEs.


Circuit Topology, Technology Files, User Defined Files & Specification
Symbolic Sizing Phase
SPICE Verification
Failed Solution
Failed Solution
Failed Solution
Module-I. MINLP: LDE-free Sizing (gm/ID-based)
LDE-free Solution (W, L, rough nf)
Module-II. MINLP: LDE-aware Sizing (Sensitivity-analysis-based)
LDE-aware (W, nf, L)
Failed Solution
Intermediate Solutions
LDE-aware Elite (W, nf, L, LRext, SCt)
Module-IV. Layout Synthesis
LDE-aware EA Solution (W, nf, L, LRext, SCt)
Module-III. LDE-aware EA-based Sizing (Many-objective θ-DEA-based)
End
Many-OEA-based Sizing Phase
Circuit Topology, Technology Files, User Defined Files & Specification
Module-I. LDE-free gm/ID-based Sizing
Ref. Device Simulation & Curve Fitting
Initialization & L-determination
gm/ID, gds/ID, Cij/ID & IDN vs. (VGS, VDS)
MINLP: LDE-Free Sizing
Tune Constraints Or L-regulation
No
Success?
Yes
Module-II. LDE-aware Sizing
LDE-free (W, nf, L)
LDE-aware (W, nf, L)
Floorplanning
Sensitivity Analysis
Dev.-in-ref.-cirt. Simulation & Curve Fitting
Constraints of ξiΔ
MINLP: LDE-aware Sizing
ξij vs. (X1, X2)
Tune Constraints
No
Success?
Yes
LDE-aware Elite (W, nf, L, LRext, SCt)
Module-IV. Layout Synthesis
Module-III. Many-OEA-based Sizing
No
Manual Layout Modification Can Help?
Yes
No
Meet Specs?
Yes
Layout Generation
Parasitic Extraction
Post-Layout Simulation
End
Yes
No
Meet Specs?
Intermediate Solutions
θ-DEA Solver
Device Geometrical Parameter Models
Floorplanner

(a)

(b)

Fig. 14. (a) Module-level and (b) detailed diagrams of the LDE-aware g_m/I_D-EA two-phase synthesis flow

In each inner optimization round, we keep observing circuit performance and DC current in all current branches by exploring the search space $\mathcal{X}(X_1, X_2)$ in a simulation-based sensitivity analysis for identifying sensitive current branches. Variable ξ_i, a current ratio from the identified sensitive branch i over its reference value, is curve-fitted in terms of X_1 and X_2 of the related MOSFETs and is constrained by a user-defined bound. For example for the two-stage Op-Amp, from all random sampling points, if the observed DC gain varies between -100% and +20% based on the reference solved from the previous sizing module, we aim at the region between 0% and +18% (so-called user-defined) for attaining a larger gain. The top 20%-18% = +2% of maximum percentage in performance might be too difficult to attain. In general, 90% of the maximum variation is used (e.g., 90%*20% = 18%) as the user-defined percentage for building the bounds. After identifying the promising sampling points by gain variations, their current variations (based on the same reference) are collect to construct the bounds that cover the variation ranges.

So the optimization of X_1 and X_2 can be conducted through such bound-based constraints introduced in the LDE-aware MINLP. A simulated annealing driven B*-tree based floorplanner [78] is employed in Module-II. Once an LDE-free sizing solution from Module-I is generated, *W*, *L*, and initial *nf* of each device are determined. Such geometrical information is used to transform device sizes into rectangular blocks as the input to the floorplanner, which is to generate an optimum floorplan as output as per the defined constraints and objectives [74].The output floorplan offers the most compact estimated chip layout area and definite interrelationships among devices, which can be used to compute the shortest wire paths for estimation of interconnect parasitics. Due to the symbolic parasitic equations discussed in Chapters 3 and 4, there is no problem for considering the parasitics throughout our proposed LDE-aware symbolic g_m/I_D-based plus EA-based hybrid sizing method. The focus in this Chapter is mainly on addressing LDEs.

In the second inner-loop inside Module-II, the floorplanner is invoked again to take the refined *W* and *nf* as input and output an updated optimum floorplan towards the following optimization of LR_{ext} and SC_t. Based on the previous work [79], we have improved our floorplanner to include the LDE consideration. On top of the only optimization of *nf* and LR_{ext} in [79], in this work we can further deal with the optimization of *W* and *nf* as the first-round parameters over LR_{ext} and SC_t, which are tackled in the second round of the LDE-aware optimization.

There are two outlets after trying to solve the formulated MINLP problem. If it cannot be solved, constraints have to be tuned for a resolution. If it is solved but the solution does not satisfy the specification by being verified in numerical simulations, a process called *L*-regulation would be resorted to for generating new *L*'s in the next round of iteration. Since the *L*-regulation may generate different *L* values for the MOSFETs in the circuit, multiple standalone reference MOSFETs (i.e., default width with various *L* values) might be needed for conducting simulations and curve fittings. The details of *L*-determination as well as the *L*-regulation have been discussed in Section 4.3.2.

The solution generated from Module-II is called LDE-aware elite solution. The solutions, which were successfully solved by the MINLP solver but fail in the following simulation verification, are referred to as intermediate solutions. Both of them will be imported to the LDE-aware heuristic-based sizing module (i.e., Module-III). After a sufficient number of intermediate solutions are collected according to the configuration of the EA (e.g., evolutionary population size) in Module-III, the iteration in Module-II breaks from further pursuing a qualified elite solution. This means the standalone optimization within Module-II is too challenging to attain a good solution. Thus, this work should be taken over by Module-III.

Inside the EA-based sizing Module-III, a many-objective EA called θ-DEA is employed to refine the sizing optimization, which involves numerical simulations yet with our proposed device geometrical parameter models to maintain the LDE-awareness. We still adopt the adaptive floorplan variation scheme previously introduced in Section 4.5.3 to help the convergence of solutions and reduce the complexity of floorplan optimization. This LDE-aware EA-based sizing phase aims at improving the intermediate solutions and the elite solution by seeking better solutions in the vicinity, which might be masked by any inaccuracy of curve-fitting and circuit modeling adopted in the previous modules. The last module (i.e., Module-IV) reflects the conventional layout synthesis that includes layout generation, parasitic extraction, and post-layout verification by using off-the-shelf design tools.

5.3. LDE-Aware Symbolic-Based Circuit Sizing

A LDE-free sizing solution can be obtained from Module-I in Fig. 14, which utilizes the g_m/I_D-based parasitic-aware sizing method introduced in Section 4.3. It is the input to the following LDE-aware sizing process (i.e., Module-II in Fig. 14). In the case of geometric expansion or constriction of device body resulted from variation of *W*, *L*, *SA*/*SB* or *nf*, complex LDEs would be incurred. For example, assuming the well enclosure follows device shape variation by considering the minimal enclosure design rule, a lateral expansion on *SA*/*SB* will increase the effective SC_t and ultimately decrease the corresponding *SCA*/*SCB*/*SCC*. That is to say, the variation of STI parameters (i.e., *SA*/*SB*) does impact on the calculation of WPE coefficients. In another example where a well encloses multiple MOSFETs, if one device increases its *nf*, it becomes thicker in the channel length direction, which leads to different SC_t. In addition, the distances from the device isolation edges on both sides to the edges of the center-located MOSFET fingers will increase due to the *nf* increment. This means the average STI effect of such a multi-finger structure device will diminish. In other words, the *nf* parameter that changes WPE by varying the effective SC_t also impacts on the STI effect. Therefore, the WPE and STI effects have to be optimized simultaneously.

Algorithm 4. LDE-aware circuit sizing

Input: LDE-free sizing solution, floorplan, and the intermediate solution set φ

Output: W's, nf's, LR_{ext}'s and SC_t's of all sensitive MOSFETs

```
1. Configure the $W$ space around the initial reference value $W_0$;
2. Initialize $nf_0$ by assuming a square-shape device via (44), and configure the $nf$ space by constraining the device shape;
3. Do a sensitivity study by sampling $N$ points within $\mathcal{X}(W, nf)$;
4. Identify sensitive current branches, and extract bounds [$\xi_{LBi}$, $\xi_{UBi}$] for the normalized sensitive current $\xi_i$;
5. Sweep ($W$, $nf$) for each MOSFET $j$ via simulation, and curve fit $\xi_i^j$, the $j$th contributor to $\xi_i$ via (45) for all $i$;
6. Link all $\xi_i^j$ to $\xi_i^\Delta$ as per their contributions via (46)-(47);
7. Solve the MINLP problem with objective (49) and constraints (48);
8. Configure the $LR_{ext}$ space for the second-round optimization;
9. By assuming a rectangular-shape device and applying the symmetry constraint, configure $SC_X$ and $SC_Y$ for the $SC_t$ space;
10. Conduct another sensitivity study by firstly sampling new $N$ points within the space of $\mathcal{X}(LR_{ext}, SC_t)$;
11. After identifying the sensitive current branches, similar to Line-4, extract bounds [$\bar{\xi}_{LBi}$, $\bar{\xi}_{UBi}$] for $\bar{\xi}_i$;
12. Sweeping ($LR_{ext}$, $SC_t$) for each MOSFET, calculate ($SAB_{eff}$, $SCA_{eff}$) via (43) and (42), and curve fit $\bar{\xi}_i^j$ with ($SAB_{eff}$, $SCA_{eff}$).
13. Link all $\bar{\xi}_i^j$ to $\bar{\xi}_i^\Delta$ as per their contributions similar to (46)-(47);
14. Solve another MINLP problem with (51) and $\bar{\xi}_i^\Delta$ being constrained with respect to $\bar{\xi}_{LBi}$ and $\bar{\xi}_{UBi}$ similar to (48);
15. Tune constraints if the MINLP solving fails, until being successful;
16. if (the sizing result cannot pass the simulation verification){
17.     Include this failed sizing result into $\varphi$;
18.     Reiterate the flow via the $L$-regulation; }
19. else { Output the verified LDE-aware sizing result; } //elite
```

A solution derived from the LDE-free optimization stage may degrade its performance after the LDE option is enabled in the numerical simulation. Provided L and bias variables from the previous optimization stage, there are still four types of geometric parameters, *W*, *nf*, *SA*/*SB*, and SC_t, to be optimized for LDEs. Since both *W* and *nf* are not only related to LDEs but also MOSFET characteristics, we propose to optimize them within search space $\mathcal{X}(W, nf)$ in the first inner-iteration of Module-II and leave *SA*/*SB* and SC_t to be optimized within search space $\mathcal{X}(SA/SB, SC_t)$ in the second inner-iteration. Our LDE-aware circuit sizing Algorithm 4 is thus proposed for them.

As indicated in Line-1 of Algorithm 4, *W*'s search space is first created by allowing a user-defined percentage variation (by default, 25%) based on the initial width, W_0, from the LDE-free solution. Then in Line-2, we define the initial finger number, nf_0, by assuming a starting point constraint that each MOSFET appears with a square shape as the following,

$$dist_{len} = (nf - 1) * (L + sd) + L + 2[SAB_{edge} + k_d * (ds + bs)] \;,$$

$$dist_{wid} = {}^{W}\!/_{nf} + 2\delta \;, \quad \mathit{Constraint}\!: dist_{len} = \; dist_{wid} \;, \tag{44}$$

where $dist_{wid}$ and $dist_{len}$ represent the distances along MOSFET *W*'s direction and *L*'s direction respectively, and δ is the short extension over the active region along the *W*'s direction. The minimum values of *sd* and δ from the design rules are used, and k_d is 1 if the bulk area and the source area are separate (called detached bulk style) and 0 otherwise. Then we use W_0 for *W* and the pre-optimized *L* from the LDE-free sizing solution to solve for nf_0 in (44). Next, *nf* is bound by [max(1, design-rule minimum), min(nf_{UB}, design-rule maximum)], where nf_{UB} can be supplied by the larger value of two *nf*'s solved via $dist_{len}$ / $dist_{wid}$ = C_{Ro} and $dist_{wid}$ / $dist_{len}$ = C_{Ro}. Here C_{Ro} is a user-defined ratio, 5 by default, for constraining the device shape. The selection of C_{Ro} should base on designers' understanding of device/chip geometry as per targeted circuits. A general rule is to try to avoid very small *nf* (e.g., 1 or 2) if possible, which can be very sensitive to layout effects.

In the LDE-free sizing stage, we have constrained the independent variables of bias node voltages and curve-fitted g_m/I_D-parameters, which are linked to circuit performance. When attempting various *W* and *nf* values in order to further optimize them for LDEs, the circuit bias condition keeps changing, and it would be hard to track node voltages that are not independent variables any more. To address this issue, we propose to track and constrain the DC bias current that flows through all branches in the given circuit topology. In the literature, Binkley *et al.* [28] and Enz *et al.* [27] suggested that the inversion coefficient, which is actually the drain current expressed in a normalized way, can reflect MOSFET's all levels of inversion and other related device characteristics. Thus, it was used to explore the trade-offs among circuit performances. This has motivated us to leverage the correspondence between normalized branch current and circuit performance by using the sensitivity analysis shown in Lines 3-6. In the study, we explore the space of *W* and *nf* with SPICE simulations, identify sensitive current branches, and gain the knowledge of critical bounds for identified branch currents. The constraints for the identified sensitive branch currents are formed according to the sensitivity analysis and incorporated into the LDE-aware MINLP. Finally, *W* and *nf*, which are linked to the constrained branch currents via the curve-fitting technique, can be optimized via MINLP in the LDE-aware sizing (in Line-7).

In this chapter, we call an LDE-free sizing result, which is to be verified in the LDE-on environment, as one reference solution. In our sensitivity study, we sample *N* points ($N = 200$ by default if using a pure random scheme) within $\mathcal{X}(W, nf)$. Less sampling points can be utilized if using a systematic reference approach [38]. After running SPICE simulations for the samples, the circuit performance and DC current (i.e., I_{ds} of each MOSFET in the worst case) of all branches are recorded. The sample data are filtered first by eliminating those, whose performance fails any specification. Then we examine the current variation of each branch individually based on the remaining data. Any branch, whose current stays over a certain amount of variation, 25% by default, is identified as a sensitive branch. We normalize each sensitive current I_i with respect to its reference value I_{ref_i} (received from the reference solution) by using $\xi_i = I_i / I_{ref\_i}$. Thus, we can obtain $[\xi_{LBi}, \xi_{UBi}]$ of ξ_i as the bound for the ith sensitive branch. Being constrained within such bounds, the circuit performance can be improved during the optimization of W and nf with MINLP. ξ_i can also be defined to specify the relationships among multiple sensitive branch currents. For example, in the comparator circuit shown in Fig. 5(b), due to the matching requirement, we can define $\xi_1 = (I_{ds1} / I_{ds4}) / (I_{ds1\_ref} / I_{ds4\_ref})$ and $\xi_2 = (I_{ds2} / I_{ds3}) / (I_{ds2\_ref} / I_{ds4\_ref})$. Moreover, it is possible to include multiple disjoint bounds so that a set of related requirements can be connected in a lumped expression as a single constraint for such ξ_i in the mixed-integer fashion.

Assume optimizing W and nf of one MOSFET in the circuit is to explore the circuit performance space dominated by a two-dimensional variable space. For m MOSFETs, we have m such two-dimensional spaces that impact on the performance. We can link these m two-dimensional spaces to the performance via branch currents, especially in the sensitive branches. In this regard, we link each two-dimensional space to the normalized currents ξ_i with the following tactics. Inside $\mathcal{X}(W, nf)$, we take turns to sweep (W, nf) for each MOSFET through LDE-on simulations while keeping the other MOSFETs the same as the reference solution. After the DC analysis from the sweeping simulations, the relationship between ξ_i^j (i.e., impact of MOSFET j on ξ_i) and (W_j, nf_j) pair is curve fitted as such,

$$\xi_i^j = f(W_j, nf_j)_{cf}\big|_L = \sum_{p=0}^{N_p} \sum_{q=0}^{N_q} a_{p,q} W_j^p \, nf_j^q \ , \tag{45}$$

where $a_{p,q}$ is the constant weighting factor, N_p and N_q are the highest order (≥1) for W_j and nf_j. Higher order polynomial contributes to higher fitting accuracy at the cost of more complex form of the resultant fitting equation. The selection of orders and their combinations for involved variables depend on the quality of the fitting including the sum of squares due to errors (SSE), R-square, adjusted R-square, root mean squared error (RMSE). Those statistics are available from the Matlab curve fitting tool box. In addition, higher order should be assigned to the variable (e.g., W) that has a more dominant influence in the fitting. The weighting factors are automatically adjusted (instead of user defined) by the fitting once the orders are decided by the users.

For each identified sensitive branch i, $S_{i,j}^{W}|_{nf}$ and $S_{i,j}^{nf}|_{W}$ are used to denote the sensitivity of ξ_i^j with respect to W and nf,

$$S_{i,j}^{W}|_{nf} = \frac{\partial \xi_i^j}{\partial W_j}|_{nf_j} ,$$

$$S_{i,j}^{nf}|_{W} = \frac{\partial \xi_i^j}{\partial nf_j}|_{W_j} . \quad (46)$$

They are used in the ξ_i variation that is defined as ξ_i^{Δ},

$$\xi_i^{\Delta} = \sum_{j=1}^{N_j} (S_{i,j}^{W}|_{nf} * \Delta_W + S_{i,j}^{nf}|_{W} * \Delta_{nf}) , \quad (47)$$

where N_j is the number of MOSFETs that have been identified influential to ξ_i, Δ_W and Δ_{nf} are the minimal variations (10nm and 1 by default for W and nf, respectively). To prevent the performance from detriment, we constrain ξ_i^{Δ} within $[\xi_{LBi}, \xi_{UBi}]$ by using the following constraint set that considers the sign of ξ_i^{Δ},

$$\begin{cases} \xi_i^{\Delta} \leq (\xi_{UBi} - 1), \\ -\xi_i^{\Delta} \leq (1 - \xi_{LBi}). \end{cases} \tag{48}$$

Next, the MINLP's objective function is defined as follows,

$$obj_1 = \alpha_1 \sum_{k=1}^{m} (dist_{wid} * dist_{len})_k + \beta_1 Power \ , \tag{49}$$

where α_1 and β_1 are the weighting factors for the device area and power consumption. The power consumption is calculated through multiplication of V_{DD} and current that is a linear relationship of I_i ($I_i = I_{ref\_i} * \xi_i = I_{ref\_i} * (1 + \xi_i^{\Delta})$) as per topology.

After *W* and *nf* are optimized from the MINLP solving, we continue to optimize the rest two parameters of SA/SB (i.e., SAB), and SC_t. We optimize LR_{ext} instead of SAB for STI thanks to their linear relationship in (43). In our implementation, LR_{ext} ranges from 0 to 500nm and its initial value is set by the median because the STI effect normally quickly diminishes when the distance is greater than 500nm in our experiment technology. Once a topology is selected, the bulk style is determined and the corresponding distance values like *ds* and *bs* in (44) can be found from the technology-dependent design rules.

In terms of the WPE parameters, by referring to Fig. 13, SA_{edge}/SB_{edge} is inevitably involved in calculating both STI and WPE parameters, and the fixed length of $k_d * (ds + bs)$ is also a component of SC_2 for WPE. So we introduce free variables SC_X and SC_Y that are only related to WPE as follows: $SC_X = SC_1 - SA_{edge} = SC_2 - SB_{edge} - k_d * (ds + bs)$ along the *L*'s direction and $SC_Y = SC_3 = SC_4$ along the *W*'s direction. The symmetry constraints are imposed to simplify the layout implication in order to decrease the complexity of the variable search space denoted by $\mathcal{X}(LR_{ext}, SC_t)$ [80]. In our implementation, we set the bounds as [design-rule minimum, 0.5μm] and [design-rule minimum, 1μm] for SC_X and SC_Y, respectively, with their medians as the initial values.

After the configuration of $\mathcal{X}$(*LRext*, SC_t) in Lines 8-9, we can optimize these variables similar to the previous operation on *W* and *nf*. Firstly in Lines 10-11, we similarly conduct the sensitivity study in order to extract the knowledge bounds, [$\overline{\xi}_{LBi}, \overline{\xi}_{UBi}$], for the normalized sensitive currents denoted by $\overline{\xi_i}$ and obtain the circuit performance by sampling points from $\mathcal{X}$(*LRext*, SC_t). With the optimized *W*, *L*, and *nf* as well as the LDE-related variables of *LRext* and SC_t, we can calculate SAB_{eff} and SCA_{eff} (i.e., the most important WPE coefficients for multi-finger MOSFETs) in Line-12 by using (43) and (42) respectively. Then in Line-13, we link SAB_{eff} and SCA_{eff} of each MOSFET *j* to $\overline{\xi}_i^j$ via the curve fitting (50) just like (45) and eventually to the $\overline{\xi_i}$ variation (i.e., $\overline{\xi}_i^{\Delta}$) as well as the constraints similar to (46)-(48):

$$\overline{\xi}_i^j = \overline{f}(SAB_{eff\_j}, SCA_{eff\_j})_{cf}\Big|_{W,\ L,\ nf}\ . \tag{50}$$

Thus, by mapping $\mathcal{X}$(*LRext*, SC_t) to $\mathcal{X}$(SAB_{eff}, SCA_{eff}), we are able to handle any irregular shape of well enclosure in the layout because there is no limit on the number of SC_t to be considered.

The $\overline{\xi}_i^{\Delta}$ can be constrained in the second MINLP formulation with the following objective function including the user-defined weighting factors α_2 and β_2 for chip area and power respectively,

$$obj_2 = \alpha_2 ChipArea + \beta_2 Power\ . \tag{51}$$

Compared to (49) where *W* and *nf* are not determined yet, the chip area in (51) can be well optimized by using a trustable floorplan output from a floorplanner that takes a list of devices as input, since the device shape is only subject to minor changes of LR_{ext} and SC_t given the optimized *W*, *L*, and *nf*. In addition to the floorplanning criteria adopted in [78], in this work we have added one more LDE feature as follows. If two devices close to each other have the same type (i.e., PMOS or NMOS), a bonus score is added to such a trial floorplan solution during the floorplanning process. Such solutions are favored since they provide flexibility for applying a unified larger well that encloses a bunch of devices with the same type well compared to multiple isolated wells each enclosing an individual MOSFET.

5.4. LDE-Aware EA-Based Circuit Sizing

Even though the solution from the LDE-aware optimization, called elite solution, may pass all the given specifications, it may still have room to improve due to imperfect selection of initial points and variable bounds for MINLP. Approximation error might exist in the curve fitting and nonlinear modeling operations. Therefore, we propose to refine the elite solution by exploring its locality via a many-objective evolutionary algorithm with the aid of numerical simulations. In this regard, we have adopted the θ-DEA method discussed in Section 3.3.3 thanks to its advantages in balancing diversity and convergence.

In the EA sizing phase, all sizing variables are represented as chromosomes to be evolved in a population along the evolutionary generations. In order to enhance the LDE-awareness in the EA sizing phase, all the device geometric parameters included in the simulation netlist have to dynamically follow the size variation along the EA process. These parameters include the finger space (i.e., *sd*), STI parameter (i.e., *SA*/*SB*), areas of source and drain (i.e., *AS* and *AD* or *ASD* if symmetric), perimeters of source and drain (i.e., *PS* and *PD* or *PSD* if symmetric), and number of equivalent diffusion squares for the source and drain regions (i.e., *NRS* and *NRD* or *NRSD* if symmetric).

Given *W*, *L*, *nf*, LR_{ext}, and bulk style, we propose to use the following analytical models for calculating these device geometric parameters. As shown in Fig. 13, the minimum *sd* and SAB_{min} are constants from the design rule, while the diffusion length is $Len_{df} = SAB_{edge} - k_i * bs$, which accounts for the lateral size of the diffusion region for an edge finger. The *AS*/*AD* is expressed by,

$$\begin{cases} AS = 2 * Len_{df} * \frac{W}{nf} + \left(\frac{nf}{2} - 1\right) * sd * \frac{W}{nf},\ nf\ is\ even \\ AD = \frac{nf}{2} * sd * \frac{W}{nf},\ nf\ is\ even\ , \\ ASD = Len_{df} * \frac{W}{nf} + \frac{nf-1}{2} * sd * \frac{W}{nf},\ nf\ is\ odd \end{cases} \tag{52}$$

where we assume the source terminal is assigned to both outermost active regions if *nf* is even. The *PS*/*PD* is given by,

$$\begin{cases} PS = 4\left(Len_{df} + \frac{W}{nf}\right) + 2\left(\frac{nf}{2} - 1\right)\left(sd + \frac{W}{nf}\right), nf\ is\ even \\ PD = 2\frac{nf}{2}\left(sd + \frac{W}{nf}\right),\ nf\ is\ even\ . \\ PSD = 2\left(Len_{df} + \frac{W}{nf}\right) + 2\frac{nf-1}{2}\left(sd + \frac{W}{nf}\right), nf\ is\ odd \end{cases} \tag{53}$$

The *NRS*/*NRD* is computed through (54),

$$\begin{cases} NRS = \left(\frac{Len_{df}}{2\frac{W}{nf}}\right) || \left(\frac{sd}{2\frac{W}{nf}}\right)_1 || \left(\frac{sd}{2\frac{W}{nf}}\right)_2 || \dots || \left(\frac{sd}{2\frac{W}{nf}}\right)_{nf-2},\ nf\ is\ even \\ NRD = \left(\frac{sd}{2\frac{W}{nf}}\right)_1 || \left(\frac{sd}{2\frac{W}{nf}}\right)_2 || \dots || \left(\frac{sd}{2\frac{W}{nf}}\right)_{nf},\ nf\ is\ even\ , \\ NRSD = \left(\frac{Len_{df}}{\frac{W}{nf}}\right) || \left(\frac{sd}{2\frac{W}{nf}}\right)_1 || \left(\frac{sd}{2\frac{W}{nf}}\right)_2 || \dots || \left(\frac{sd}{2\frac{W}{nf}}\right)_{nf-1},\ nf\ is\ odd \end{cases} \tag{54}$$

where “||” is the parallel operator like the one used for shunt resistance calculation.

During the EA optimization, the device sizes keep changing among the trial solutions (i.e., evolutionary chromosome individuals). On the one hand, a fixed floorplan template obtained from Module-II is no doubt suboptimal. On the other hand, it is extremely time-consuming to rerun the floorplanner all the time as in [59] whenever a trial solution is constructed by evolutionary operators. This would inevitably increase the convergence difficulty since the exploitation of floorplans may not be in line with that of device sizes in the course of EA evolution. Therefore, we still employ the adaptive floorplan variation scheme discussed in Section 4.5.3 within the EA sizing phase to maintain potentially good floorplans and only rerun the floorplanner when the old floorplans’ compatibility is not acceptable for the updated device sizes.

5.5. Experimental Results

In Section 5.5.1, a case study is conducted to demonstrate high accuracy of our proposed symbolic modeling scheme for calculating device geometric parameters. Section 5.5.2 introduces another case study and a performance analysis of the first-phase sizing results. Section 5.5.3 highlights the merits of our proposed LDE-aware hybrid sizing method by providing our experimental results in comparison with other layout-aware circuit sizing approaches. All the case studies and experiments in this chapter were conducted in a CMOS 65nm technology process with 1V power supply.

5.5.1. Verification of Modeling for Device Geometric Parameters

The following case study was conducted on a single NMOS transistor under the same bias voltages for different test cases. It demonstrates that our proposed modeling scheme can offer more accurate computation of the device geometric parameters, which lead to closer device electrical characteristics than the schematic-level estimation performed by the commercial Cadence tool [56] with reference to the corresponding layouts.

The W and L of the test MOSFET device are 10μm and 100nm, respectively. To save silicon area, the integrated bulk type is adopted if one of the drain or source terminal is supposed to connect to the bulk terminal. Table 12 includes two groups of data for even (i.e., $nf = 4$) and odd (i.e., $nf = 5$) finger numbers. In the $nf = 4$ group, there are two sub-groups where the bulk connection style is either symmetrically integrated (denoted as I.-I.) or detached (denoted as D.-D.) on both sides. In the $nf = 5$ group, one side uses the integrated bulk style while the other side utilizes the detached bulk style (denoted as I.-D.), as shown in Fig. 13. For estimation of geometric parameters, the Cadence Virtuoso schematic tool using the BSIM4 model considers effect of neither nf variation nor diffusion expansion, which is strongly dependent on the specific bulk connection style when calculating SA_{eff} and SB_{eff}. This would lead to large difference in the calculation of SCA_{eff} compared to the post-layout references. In contrast, by using our proposed modeling scheme, a satisfactory convergence of the geometric parameters between this work and the post-layout reference can be reached.

In terms of MOSFET electrical performance, the effective beta ($Beta_{eff}$), which can reflect the effective mobility (μ_{eff}) of a device for a given geometry in a specific technology, is,

$$Beta_{eff} = \mu_{eff} c_{oex} W_{eff}/L_{eff} \ , \tag{55}$$

where C_{oex} is a technology-dependent constant. $Beta_{eff}$ is closely related to I_D in all work regions of MOSFET. In our case study, the largest difference of I_D, which can be found in the case of D.-D. between the reference (i.e., 539.46) and Cadence estimation (i.e., 509.91), gives 5.48% error, whereas the estimation error is reduced to only 2.74% by using our proposed model (i.e., 554.64). Thus, a 2.73% (i.e., 5.47%-2.74%) reduction regarding the I_D estimation error is achieved in our method. In general more than half of the estimation error can be reduced in other MOSFET electrical characteristics between Cadence and our proposed LDE-aware device characterization model with respect to the same reference.

Table 12. Device parameter measurement and performance: A case study

Settings, Measurement & Performance	**nf =4**						**nf =5**		
	Post-layout I.-I.	**Cadence**	**This work**	**Post-layout D.-D.**	**Cadence**	**This work**	**Post-layout I.-D.**	**Cadence**	**This work**
Device Geometrical Parameter Measurement									
SA_{eff} (μm)	0.98	0.175	0.98	0.625	0.175	0.625	1.13	0.175	1.13
SB_{eff} (μm)	0.98	0.175	0.98	0.625	0.175	0.625	0.775	0.175	0.775
AS (μm^2)	2.1	1.375	2.1	1.375	1.375	1.375	1.44	1.15	1.44
AD (μm^2)	1	1	1	1	1	1	1.15	1.15	1.15
PS (μm)	16.68	16.1	16.68	16.1	16.1	16.1	13.44	13.15	13.44
PD (μm)	10.8	10.8	10.8	10.8	10.8	10.8	13.15	13.15	13.15
NRS	0.015	0.010	0.015	0.013	0.010	0.013	0.012	0.010	0.012
NRD	0.010	0.010	0.010	0.010	0.010	0.010	0.011	0.010	0.011
SCA_{eff}	4.969	7.540	4.969	4.651	7.540	4.651	5.364	7.601	5.364
Performance of Device Electrical Characteristics from Simulation									
$Beta_{eff}$ (m)	13.68	13.30	13.77	13.70	13.30	13.81	13.79	13.49	13.88
I_D (μ)	537.54	509.91	550.14	539.46	509.91	554.64	538.02	517.03	550.30
V_{th} (m)	362.28	369.30	360.46	361.81	369.30	359.49	363.25	368.77	361.54
g_m (m)	3.334	3.228	3.371	3.340	3.228	3.385	3.347	3.267	3.384
g_{ds} (m)	0.275	0.252	0.284	0.276	0.252	0.287	0.274	0.257	0.283

5.5.2. Verification of LDE-Aware g_m/I_D-Based Sizing

We have used the two stage Op-Amp in Fig. 5(a) and the differential comparator in Fig. 5(b) for our experiments. The bias condition of this comparator circuit includes: V_{INP} = 0.8V, V_{INN} = 0.4V, V_{REFN} = 0.6V, and V_{REFP} = 0.8V. In Table 13, a case study is used to compare the performance of the Op-Amp between the traditional methods and our proposed scheme for optimizing *W* and *nf*.

The second column records the performance of the g_m/I_D-based LDE-free sizing solution that passes the specification verification but with the LDE option inactivated. When simulating this solution under the LDE-on simulation environment, the performance drops significantly especially for Gain and UGB (unity gain bandwidth) as shown in Columns 3-4 of Table 13. Here PM and GM stand for phase margin and gain margin, respectively. In Columns 2-3, the finger numbers (i.e., *nf*) of all the MOSFETs are set by the minimum values as per the design rule. Since the minimal finger number is usually not desirable in terms of MOSFET geometry, in Column-4 we consider each device in a square shape by calculating *nf*'s according to (44). This change happens to show performance improvement but very limited due to lack of electrical consideration.

By following our proposed framework for optimizing *W* (i.e., transistor width) and *nf* but using the traditional square-law MOSFET current equations instead of our proposed curve-fitting ones for linking *W* and *nf* to ξ_i, the solution shown in Column-5 of Table 13 can exhibit further performance improvement but still fail in the Gain due to inaccuracy of the applied square-law equations as well as the approximate technology parameters. However, a much better performance shown in the last column of Table 13 demonstrates the merit of our proposed optimization method by using the accurate simulation-based and fitted current expressions (45).

Table 13. Performance comparison between using the traditional methods and our fitting model: A case study for the two-stage Op-Amp

Performance	**LDE-free Min. nf (W & L)**	**LDE-on**			
		Traditional Methods			**Fitting Model**
		Min. nf	**Fair (nf)**	**Opt. Fp. (W & nf)**	**Opt. Fp. (W & nf)**
Gain > 60 (dB)	61.71	56.77	57.54	59.66	62.52
UGB > 4 (MHz)	9.62	5.63	8.49	9.02	8.30
PM > 60 (°)	61.18	68.88	71.18	70.45	64.57
GM > 15 (dB)	22.53	25.86	33.15	35.91	28.07

Tables 14-15 present our sizing results in the standalone first sizing phase under different settings in comparison with [20] and [78], which use geometric programming with and without LDE considerations, respectively. The method used in [78] is actually detailed in Section 3. Setting-1 (i.e., Set-1), which is verified under the LDE-off simulation environment, presents the LDE-free sizing performance where *nf*'s are set by the minimum allowable values. Setting-2 (i.e., Set-2) is verified under the LDE-on simulation environment with the same sizing results used in Set-1. Without any implication of *nf*'s, repeated regulation might take place in practice during the transformation from the sizes associated with a floorplan to a real layout.

To optimize *nf*'s for better LDEs, Setting-3 (i.e., Set-3) that represents our proposed first-round LDE optimization tunes *nf*'s and *W*'s based on the g_m/I_D-based LDE-free reference solution. In the second-round optimization as represented by Setting-4 (i.e., Set-4), we optimize LR_{ext} and SC_t that are the key parameters reflecting STI and WPE. Since Zhang *et al.* [20] optimizes all three LDE parameters (i.e., *nf*, *SA*/*SB*, and only one *SC*) in one stage, it has only Set-4 but no Set-3.

For the Op-Amp in Table 14, the reported data in Set-2 reveal that after the LDEs are activated in the simulation environment, the performances drop dramatically for all three works with reference to Set-1. In addition, after the LDE parameters are optimized in [20], the Gain in Set-4, which is improved in comparison to that of Set-2, becomes close to the one in the ideal LDE-free case (i.e., Set-1). While this demonstrates the effectiveness of the LDE-aware sizing method proposed in [20], the performances are still inferior to the given specifications. However in Set-3 of our work, the performances of Gain, PM, and GM even excel the ones from Set-1, and the performances of Set-4 can be further improved in all aspects compared to Set-3.

Table 14. Two-stage Op-Amp: g_m/I_D-based LDE-aware sizing results

Settings	**Settings**	**Gain > 60 (dB)**	**UGB > 4 (MHz)**	**PM > 60 (°)**	**GM > 15 (dB)**
T. Liao [78]	**Set-1**	51.00	7.91	65.85	30.72
	Set-2	46.08	5.49	76.26	37.15
Y. Zhang [20]	**Set-1**	50.73	10.99	63.31	46.07
	Set-2	39.69	9.59	67.98	46.78
	Set-3	-	-	-	-
	Set-4	49.95	11.94	64.71	46.70
This Work	**Set-1**	61.71	9.62	61.18	22.53
	Set-2	56.77	5.63	68.88	25.86
	Set-3	62.52	8.30	64.57	28.07
	Set-4	62.97	8.53	68.03	34.55

Table 15. Comparator: g_m/I_D-based LDE-aware sizing results

Settings	**Settings**	**Delay < 250 (ps)**	**+Overshoot < 300 (mV)**	**-Overshoot < 150 (mV)**
T. Liao [78]	**Set-1**	249.9	200.6	46.0
	Set-2	310.8	200.3	50.2
Y. Zhang [20]	**Set-1**	299.9	266.6	74.0
	Set-2	337.5	263.3	72.2
	Set-3	-	-	-
	Set-4	318.6	296.2	77.9
This Work	**Set-1**	152.5	183.0	53.3
	Set-2	296.7	184.0	53.9
	Set-3	164.6	202.2	60.6
	Set-4	150.4	219.9	65.2

In Table 15, propagation delay (i.e., Delay) is considered as an important aspect besides positive and negate overshoots for the comparator circuit. A similar trend of the Op-Amp in Table 14 can be generally observed from the comparator case in Table 15. Moreover, it is interesting to see that the Delay from our Set-3 (i.e., 164.6ps) fails to reach the one in Set-1 (i.e., 152.5ps). But by further optimizing LR_{ext} and SC_t, Set-4 can achieve the delay of 150.4ps, which outperforms 152.5ps gained by Set-1. This can help demonstrate the necessity of optimizing LR_{ext} and SC_t in our proposed LDE-aware sizing method.

By comparing all three methods, one can see [78] is obviously inferior due to lack of LDE consideration in the optimization. The performances were significantly degraded in the LDE-on simulation environment. On the other hand, in [20] the standalone GeoP-based approach fails to provide a sufficient level of accuracy, in comparison with our proposed g_m/I_D sensitivity-analysis based approach that can utilize numerical simulation and curve fitting technique in a more general MINLP modeling. This can not only be reflected from the inaccurate posynomial fitting model for WPE, but also the μ_{eff}/μ_0 ratio (the mobility after considering WPE over the intrinsic one before the consideration), which is supposed to be always greater than 1 but different from our verification regarding the model in [20]. Moreover in that work, the basic circuit modeling just utilizes the traditional square-law equations (e.g., I_D and g_m), which are not accurate and hard to represent technology variations.

5.5.3. Verification of LDE-Aware g_m/I_D-EA Hybrid Sizing

In the many-OEA-based sizing phase, device sizes in addition to circuit bias inputs are included in the chromosome variable vector. By using the proposed device parameter models (42)(43)(52)-(54), all the layout-dependent parameters present in the simulation netlist can dynamically follow the device geometric variables whenever different chromosomes as a result of evolutionary recombination are attempted. This makes the many-OEA-based sizing process LDE-aware. The SPICE simulation invoked in the EA process is conducted with the setting of LDE-on. For each test circuit, six schemes are compared with one another. Scheme-0 is the LDE-aware standalone symbolic-phase sizing method whose performance data are copied from Set-4 in Table 14 and Table 15. Scheme-1 follows the Synthesis Flow for fast Parasitic Closure (called *SFPC* for short) originally proposed in [59], which encloses placement and global routing inside a refined-sizing loop.

Scheme-2 reflects the idea in [67] that uses a traditional evolutionary algorithm on analog circuit sizing. The implementation of Scheme-3 imitates one layout-aware sizing work [33] by employing the differential evolution (DE) algorithm. As for the schemes implemented by using the many-objective θ-DEA method, Scheme-4 that is the standalone many-OEA-based sizing includes neither pre-optimized elite knowledge nor information from the intermediate solutions, as configured in the single-objective Scheme-2. By following the proposed methodology in this chapter, Scheme-6 takes advantage of the LDE-aware elite solution as well as the intermediate solutions, which is similarly configured in Scheme-3 too in order to fully test the capability of such a single-objective DE sizing method with our granted symbolic-sizing-phase results. In order to justify the necessity of conducting the g_m/I_D-based LDE-aware sizing in Module-II, Scheme-5 only includes Module-I and Module-III by solely taking the initial solutions from the LDE-free sizing (Module-I) and then running the subsequent LDE-aware many-OEA-based sizing process (Module-III). Furthermore, in order to fairly compare the performance among Schemes 1-6, features of parasitic-awareness and LDE-awareness are reasonably implemented for all of them by following [75] and reusing the advocated device parameter models (42)(43)(52)-(54) to compose a LDE-aware simulation netlist when attempting a new chromosome resulted from evolutionary recombination. Since our work in this chapter is focused on early awareness of LDEs for fast circuit sizing optimization, detailed time-consuming layout generation and extraction are not included during the EA sizing process. Instead, layout parasitics and LDEs, which are estimated by our sizing tool, are derived and embedded into the circuit netlist for SPICE simulation.

The initial solution from the LDE-free sizing Module-I and the elite solution from the LDE-aware sizing Module-II as well as the intermediate solution set φ can be used to restrict the search space in the EA sizing phase for exploring the locality. Therefore, we choose the size of population, SP=32, and the maximum number of generation, Gen_{max}=20, as a small configuration in Scheme-3, Scheme-5, and Scheme-6, while they are 56 and 40 in Scheme-2 and Scheme-4 respectively as a large configuration in the context of lack of initial solution, elite and information from φ. In this work, we adopt a minimization-based fitness function [78], which is a summation of terms with weighting factors accordingly from different performance aspects. A smaller fitness indicates a better circuit performance. For each scheme, we ran 10 times and evaluated the quality in terms of the best-fitness [81] for single-objective schemes and inverted generational distance (IGD, a metric (the smaller the better) to assess the quality of a solution set among the others [78]) for many-objective schemes. Similar to [78], we collected all the specification-satisfied nondominated solutions from multiple runs of various schemes on the same problem to generate a pseudo Pareto Front for IGD calculation. The statistics of 10 runs for best-fitness and IGD are reported in Table 16 and Table 17 for the two-stage Op-Amp, and Table 19 and Table 20 for the comparator circuit, respectively. For each run, if there is at least one solution satisfying the given specifications, we consider it successful.

We select the best run in terms of best best-fitness[1] for the single-objective schemes and best IGD for the many-objective schemes for reporting the detailed performance in Table 18 (for the two-stage Op-Amp) and Table 21(for the comparator circuit). In these tables, we also provide the best-fitness for all the schemes in the row of "Single/θ: Best-Fitness" for comparison purpose. The total run time is comprised of three parts (i.e., Modules I-III). In addition, the total run time of MINLP taking place in Modules I-II is also reported. All the experiments were conducted on a server with 32-core Intel Xeon CPU E5-2650 @ 2.00GHz.

Inside a selected best run, for a complete resultant solution set X_c in each scheme, the specification-satisfied solutions forms a subset, $X_{s\text{-}s}$, leaving the complementary subset with specification-failed solutions. By considering the convergence-oriented nature of the single-objective methods, the complete solution set should be used to reflect the optimization quality. Therefore, average fitness and standard deviation are calculated inside X_c for Schemes 1-3. However, for the many-objective θ-DEA, in order to encourage the exploration of multiple clusters, the clusters have to be distributed across the entire solution space even in certain infeasible regions by managing a systematic construction of reference points [38]. Therefore, the average fitness is calculated inside $X_{s\text{-}s}$, which refrains from presenting senseless data penalized by infeasible solutions from certain poor regions. We employ the success-rate (i.e., $|X_{s\text{-}s}|/|X_c|$) to exhibit the diversity of the final solution set for the θ-DEA methods (i.e., Schemes 4-6).

In Table 16 for the two-stage Op-Amp circuit, the number of successful runs is 5/10 (5 out of 10 runs) when the elite and φ are employed in Scheme-3, which is much larger than those in Schemes 1-2 (i.e., 2/10 and 0/10, respectively) where there are no initial solutions or little specific

[1] best best-fitness: the first 'best' refers to the selection among 10 runs with individual best-fitness, while the second 'best-' refers to the selection among the solution sets with individual fitness in each run.

information assumed for setting the search space. In addition, Scheme-3 gains the best statistics compared with Schemes 1-2 for all of the five aspects exhibiting the effectiveness of adopting the initial solutions from Module-II. For the IGD statistics in Table 17, the numbers of successful runs for Schemes 4-5 are 7/10 and 6/10 respectively, which are larger than those in the single-objective schemes. This helps exhibit higher effectiveness after adopting the recommended many-objective θ-DEA method in Module-III. Moreover, the highest number of successful runs (i.e., 10/10) and better or equivalent best-fitness statistics also demonstrate the superiority of our proposed Scheme-6 over Schemes 4-5.

In Table 18 for the two-stage Op-Amp, all the schemes except Scheme-2 can manage to pass the specification. For the single-objective schemes, the small-scale configuration with elite and φ (i.e., Scheme-3) is able to derive a better solution in terms of best-fitness (i.e., 0.461) compared to 0.561 and 0.547 in Scheme-1 and Scheme-2, respectively. Even though Scheme-1 runs faster (i.e., 12.96min), its lower number of successful runs (i.e., 2/10 (2 out of 10 runs)) and poorer solution quality (i.e., best-fitness of 0.561) in comparison to those (i.e., 5/10 and 0.461) in Scheme-3 can hardly justify its adoption. Without any aid of initial solutions, Scheme-2 has to search in a huge variable space and thus may end up with useless outcome (i.e., number of successful runs as 0/10).

Table 16. Two-stage Op-Amp: Statistics of the LDE-aware sizing results for single-objective schemes

Statistics **(10 Runs)**	**Scheme-1** **(SFPC)** **Zhang** [59]	**Scheme-2** **Tlelo-Cuautle** [67]	**Scheme-3** **Vancorenland** [33]
Best (Best-Fitness)	0.561	0.547	0.461
Worst (Best-Fitness)	0.880	1.252	0.807
Median (Best-Fitness)	0.667	0.733	0.578
Mean (Best-Fitness)	0.697	0.761	0.602
# Successful Runs	2/10	0/10	5/10

Table 17. Two-stage Op-Amp: Statistics of the LDE-aware sizing results for many-objective schemes

Statistics (10 Runs)	**Scheme-4**	**Scheme-5**	**Scheme-6 This work**
Best (IGD)	0.180	0.142	0.123
Worst (IGD)	0.480	0.259	0.204
Median (IGD)	0.234	0.197	0.171
Mean (IGD)	0.249	0.194	0.165
# Successful Runs	7/10	6/10	10/10

Table 18. Settings and performance of the two-stage Op-Amp from the best run

Statistics (Best Run)		**g_m/I_D**	**Single-objective Methods**			**Many-objective θ-DEA Methods**		
		Scheme-0 LDE-aware Solution	**Scheme-1 (SFPC) Zhang** [59]	**Scheme-2 Tlelo-Cuautle** [67]	**Scheme-3 Vancorenland** [33]	**Scheme-4**	**Scheme-5**	**Scheme-6 This work**
Single/θ: Best-Fitness		0.559	0.561	0.547	0.461	0.465	0.469	0.466
θ: Average-Fitness		-	-	-	-	0.506	0.502	0.500
θ: Success-Rate		-	-	-	-	5.36%	15.625%	31.25%
Single: Average-Fitness		-	0.848	1.03	0.921	-	-	-
Single: Standard-Deviation		-	0.217	0.300	0.271	-	-	-
Run Time (mins)	**Module-I**	12.48	-	-	12.48	-	12.48	12.48
	Module-II	9.89	-	-	9.89	-	-	9.89
	MINLPs'	0.23	-	-	0.23	-	0.18	0.23
	Module-III	-	12.96	31.19	11.44	35.10	12.17	12.14
	Total	22.37	12.96	31.19	33.81	35.10	24.65	34.51
Specification & Objectives		**Circuit Performance (from the Representative Solution with the Smallest Fitness)**						
Gain > 60dB		62.97	61.01	48.81	61.60	60.52	63.04	61.22
UGB> 4MHz		8.53	27.02	18.22	17.11	44.95	15.44	18.82
PM > 60°		68.03	78.50	80.17	95.66	110.07	107.24	101.61
GM > 15dB		34.55	23.65	55.02	35.53	35.63	32.68	31.35
Area(μm^2)		153.07	762.73	47.10	132.12	778.77	319.55	198.36

For the many-objective schemes, the overall quality in terms of best-fitness for Scheme-6 is comparable to that of Scheme-4. However, without any focus especially inside a huge search space, even the sophisticated θ-DEA with sufficient evolutionary resources may encounter some difficulties in the search, and therefore a decreased number of successful runs (i.e., 7/10 in Table 17) is observed for the two-stage Op-Amp. Furthermore, the number of successful runs, IGD, and success-rate (i.e., 10/10 (statistic from the total 10 runs), 0.123, and 31.25% (statistic from the best single run)) in Scheme-6 are all better than those (i.e., 7/10, 0.180, and 5.36% correspondingly) in Schemes-4. This indicates that the small-scale configuration with the beneficial knowledge from the pre-optimized elite and φ is more efficient to provide competing solutions with similar level of run time (i.e., 34.51min vs. 35.10min). In comparison to Scheme-6, 28.66% (i.e., 9.89min/34.51min) of run time can be saved by removing the g_m/I_D-based LDE-aware sizing Module-II as configured in Scheme-5. However, the number of successful runs is reduced to 6/10 and the success-rate is halved (i.e., 15.625% vs. 31.25%). This is mainly due to the enlarged search space from the non-optimized *nf* and other LDE parameters, which can make strong impact on circuit performance. Therefore, with the assistance of our proposed comprehensive g_m/I_D-based LDE-aware sizing scheme in Module-II, more reliable performance can be beneficially gained over the other schemes yet at the cost of run time overhead.

Table 19. Differential comparator: Statistics of the LDE-aware sizing results for single-objective schemes

Statistics (10 Runs)	**Scheme-1 (SFPC) Zhang** [59]	**Scheme-2 Tlelo-Cuautle** [67]	**Scheme-3 Vancorenland** [33]
Best (Best-Fitness)	0.344	0.530	0.293
Worst (Best-Fitness)	3.00	1.514	0.627
Median (Best-Fitness)	0.720	0.702	0.380
Mean (Best-Fitness)	0.877	0.780	0.404
# Successful Runs	2/10	1/10	9/10

Table 20. Differential comparator: Statistics of the LDE-aware sizing results for many-objective schemes

Statistics (10 Runs)	**Scheme-4**	**Scheme-5**	**Scheme-6 This work**
Best (IGD)	0.150	0.124	0.115
Worst (IGD)	0.453	0.445	0.360
Median (IGD)	0.217	0.202	0.165
Mean (IGD)	0.280	0.240	0.212
# Successful Runs	10/10	7/10	10/10

Table 21. Settings and performance of the differential comparator from the best run

Statistics (Best Run)		g_m/I_D	Single-objective Methods			Many-objective θ-DEA Methods		
		Scheme-0 LDE-aware Solution	Scheme-1 (SFPC) Zhang [59]	Scheme-2 Tlelo-Cuautle [67]	Scheme-3 Vancorenland [33]	Scheme-4	Scheme-5	Scheme-6 This work
Single /θ: Best-Fitness		0.429	0.344	0.530	0.293	0.142	0.162	0.143
θ: Average-Fitness		-	-	-	-	0.182	0.213	0.205
θ: Success-Rate		-	-	-	-	21.15%	40.625%	53.125%
Single: Average-Fitness		-	1.17	0.769	0.642	-	-	-
Single: Standard-Deviation		-	0.374	0.141	0.242	-	-	-
Run Time (mins)	Module-I	5.19	-	-	5.19	-	5.19	5.19
	Module-II	6.64	-	-	6.64	-	-	6.64
	MINLPs'	0.35	-	-	0.35	-	0.27	0.35
	Module-III	-	13.99	33.01	12.48	39.93	13.01	12.80
	Total	11.83	13.99	33.01	24.31	39.93	18.20	24.63
Specification & Objectives		Circuit Performance (from the Representative Solution with the Smallest Fitness)						
Propagation Delay < 250ps		150.4	187.95	240.03	42.42	78.63	59.88	38.25
+Overshoot < 350mV		219.9	1.00	241.05	181.87	17.07	57.91	64.38
-Overshoot < 150mV		65.2	71.86	69.13	67.53	18.11	32.62	32.68
Area(μm^2)		148.93	403.67	31.34	93.71	235.88	74.68	37.13

Similar trends of statistics, as discussed for the two-stage Op-Amp in Table 16 and Table 17, can be observed in Table 19 and Table 20 for the comparator circuit. For the selected best runs as the representatives reported in Table 21, the integration of elite and φ (i.e., Scheme-3) still yields outstanding performance, especially in best-fitness and average-fitness (i.e., 0.293 and 0.642, respectively), among the single-objective schemes (comparing to 0.344 & 1.17 in Scheme-1 and 0.530 & 0.769 in Scheme-2, respectively), which further confirms the effectiveness of adopting the symbolic sizing phases via Modules I-II. By increasing the evolutionary resources but without any help from initial solutions, Scheme-4 just reaches similar best-fitness however with additional ((39.93min - 24.63min) / 39.93min = 38.32%) run time in comparison to our proposed Scheme-6. In addition, the IGD statistics and the number of successful runs in Table 20 as well as the reported best-run's performance in Table 21 from Scheme-6 are all superior to those in Scheme-5. This indicates that the g_m/I_D-based LDE-aware sizing optimization conducted in Module-II is essential for improving the quality and robustness of the final solutions yet at the cost of run time overhead (i.e., 6.64min/24.63min=26.96%).

Moreover, by observing the performance of Scheme-1 in Tables 16, 18, 19, 21, our experimental results illustrate that due to highly frequent floorplan variations in the SFPC scheme, the intractable parasitics and LDEs fail to cooperate well with sizing update along the course of evolution and thus increase the difficulty in exploring optimal solutions in practice. For our proposed Scheme-6, the resultant device sizes, whose performances are reported in the last column of Table 18 and Table 21 for the corresponding two experimental circuits, are provided as follows: For the two-stage Op-Amp, W_1=33.93µm, L_1=230nm, W_2=33.93µm, L_2=230nm, W_3=21.39µm, L_3=680nm, W_4=21.39µm, L_4=680nm, W_5=14.55µm, L_5=910nm, W_6=35.53µm, L_6=265nm, W_7=9.85µm, L_7=660nm, W_8=1.26µm, and L_8=260nm; For the comparator, W_1=0.97µm, L_1=200nm, W_2=0.97µm, L_2=200nm, W_3=0.97µm, L_3=200nm, W_4=0.97µm, L_4=200nm, W_5=12.73µm, L_5=70nm, W_6=12.73µm, L_6=70nm, W_7=1.66µm, L_7=65nm, W_8=1.66µm, L_8=65nm, W_9=0.27µm, L_9=80nm, W_{10}=0.27µm, L_{10}=80nm, W_{11}=0.27µm, L_{11}=80nm, W_{12}=0.27µm, and L_{12}=80nm. By following the sizing results associated with their floorplans, we used Cadence Layout-XL tool [68] to perform layout generation. Then we used Mentor Graphics Calibre tool [76] for parasitics extraction. The designs have been finally verified by the numerical simulator with satisfactory post-layout performance obtained.

5.6. Summary

In this chapter, we have proposed an efficient LDE-aware two-phase g_m/I_D-EA hybrid circuit sizing methodology for high-performance analog circuits. The proposed method firstly utilizes a symbolic nonlinear optimization in the g_m/I_D form by modeling the first-order performance equations, technology-dependent device characterization and other constraints in order to seek a global reference solution. Then it continues to optimize the parameters for LDEs in another nonlinear optimization by constraining the normalized current with the aid of sensitivity analysis. In the second sizing phase, one advanced many-objective EA called θ-dominance-based evolutionary algorithm with sensible configuration is adopted for a more focused and refined search under an informative guide implied from the LDE-aware elite solution as well as the intermediate solutions. The methodology was applied to common analog circuits, and our comparison with other layout-aware approaches clearly demonstrates its efficacy.

In the next chapter, we will introduce a machine-learning-based circuit sizing methodology with the consideration of LDEs. Only accurate numerical simulations will be involved due to the inaccuracy concern about circuit modeling discussed earlier.

Chapter 6 High-Dimensional Many-Objective Bayesian Optimization for LDE-Aware Analog Integrated Circuit Sizing

6.1. Introduction

In Section 2.1.2 , LDEs has been introduced. They are considered for analog circuit sizing by using the proposed g_m/I_D-EA hybrid sizing methodology in Chapter 5. To reinforce its detrimental impact on analog circuit performance, the following experimental results are reported. The LDEs that are intuitively reflected in the layout stage can actually be considered in the early schematic/netlist design stage. According to our experiments, LDEs can be reflected from the schematic design stage for the typical two-stage operational amplifier (Op-Amp) shown in Fig. 5(a) in a 65nm CMOS technology. If ignoring LDEs in the simulation, the DC gain, unit gain bandwidth (UGB), phase margin (PM), and gain margin (GM) were initially 64.87dB, 22.01MHz, 121.5°, and 21dB, respectively. They were changed to 55.8dB, 17.21MHz, 114.5°, and 23.51dB respectively after the LDEs were activated. In addition, when we further attempted different transistor finger number (*nf*) with the activated LDEs, the gain could drop to 41.96dB in the worst scenario. Further performance degradation may take place if such a design is placed and laid-out. Therefore, an LDE-aware circuit sizing method is highly demanded in the advanced technologies, preferably starting from the early schematic/netlist design stage.

A machine-learning-based approach called Bayesian optimization (BO) has recently emerged to handle black-box optimization problems that involve computation-intensive function evaluators (e.g., SPICE simulator). As introduced in Section 2.2.5, the prevalent surrogate model used for

BO is Gaussian process (GP). Gaussian process based Bayesian optimization (GP-BO) has the scalability problem of the input space as discussed in Section 2.2.5. The additive models proposed in [42] were later adopted in [82] to further improve the high-dimensional GP-BO. Its success has formed part of our inspiration for developing an LDE-inclusive sizing algorithm to address the large number of optimization variables incurred by LDEs. Furthermore, on top of [82] solely dedicated to single-objective optimization, we have developed a high-dimensional and real many-objective (i.e., >3 optimization objectives) GP-BO based (called HMBO) LDE-aware automated circuit sizing methodology in this chapter. In addition, we have proposed a more efficient dimension splitting scheme, which is especially beneficial when massive patterns exist in the high-dimensional variable space. The main contributions of this chapter are summarized as follows:

- To the best of our knowledge, this is the first work that utilizes machine-learning techniques to simultaneously optimize LDEs (i.e., WPE and STI) in the schematic-level automated analog circuit sizing process.
- Our proposed HMBO can deal with high-dimensional variable space and real many-objective optimization of GP-BO for analog circuit sizing applications.
- We have proposed a performance-driven parameter learning scheme for pattern generation and selection along with the adopted TileGPs [82] as the additive models.

The research work conducted in this chapter has been submitted to IEEE Transactions on Very Large Scale Integration Systems (TVLSI) seeking for publication [J1].

6.2. Bayesian Optimization

Bayesian optimization (BO) is suitable for optimizing objective functions which take longer time to evaluate, that is, expensive black-box functions. It adopts a surrogate model and an acquisition function. The surrogate model mimics the behavior of the objective functions by providing predictions and quantifying uncertainties, while the acquisition function is to determine where to sample (or query) in the search space. Thanks to the well-calibrated uncertainty of prediction, Gaussian process (GP) model is highly recommended to serve as the surrogate model. With some randomly sampled data, prior probability distribution (i.e., *prior* for short) that captures beliefs about the behavior of the objective function can be obtained from GP regression with a particular mean and covariance (kernel).

Let $f : \mathcal{X} \rightarrow \mathbb{R}$ be a black-box function to be optimized over a compact variable set $\mathcal{X} = [0, R]^D \subseteq \mathbb{R}^D$, where *D* is the number of *dimensions* for the input variable space. In this chapter, each dimension corresponds to one defined variable. The variable space implied by all *D* variables is referred to as the *D-dimensional space*. In the standard setting of BO, firstly consider optimizing a single objective function $f(\boldsymbol{x})$ (e.g., DC gain of an Op-Amp) over free variables $\boldsymbol{x}$ (e.g., sizing variables). A Gaussian process with zero mean and covariance *k* is denoted by $\mathcal{GP}(0, k)$, and let f be drawn from $\mathcal{GP}(0, k)$. Given *n* observations $\Omega_n = \{(\boldsymbol{x}_t, y_t)\}_{t=1}^{n}$ where y_t is drawn from normal distribution $\mathcal{N}(f(\boldsymbol{x}_t), \sigma^2)$ and σ^2 is the variance for Gaussian noise, we can obtain the log likelihood for observations Ω_n . The marginal likelihood can be used to estimate the hyperparameters in various kernels, such as the popular squared exponential (SE) and Matérn covariance functions. Then the posterior probability distribution (i.e., *posterior* for short) in terms of mean, variance, and covariance can be accordingly expressed [82]. Here the mean can be used to conduct the probabilistic prediction of y with the confidence of prediction as reflected by the variance.

The second component of BO is the acquisition function (Ψ), which is typically an inexpensive function representing how desirable evaluating f at a given point $\boldsymbol{x}$ is expected to be. By optimizing an acquisition function, the resultant $\boldsymbol{x}$ that has potential to achieve the best objective amount and reduce the uncertainty is selected as the location of the next observation (i.e., query point). To find such a promising $\boldsymbol{x}$, search algorithms are needed. For example, the limited memory Broyden-Fletcher-Goldfarb-Shanno for bound constrained optimization algorithm (L-BFGS-B) is normally used in the single-objective optimization setting, and NSGA-II or NSGA-III can be used for multi-objective (equal or less than 3 objectives) or many-objective (more than 3 objectives) optimization respectively. Based on various focuses on the balance between exploration and exploitation, common acquisition functions from the literature include probability of improvement (PI), expected improvement (EI), entropy search (ES), predictive entropy search (PES), max-value entropy search (MES), lower confidence bound (LCB), and upper confidence bound (UCB) [42].

We introduce the maximization-based UCB in the following due to our adoption in Algorithm 6 and Algorithm 7 as well as its compatibility with additive structure [42],

$$\Psi_t(\boldsymbol{x}) = \mu_{t-1}(\boldsymbol{x}) + \sqrt{\beta_t}\sigma_{t-1}(\boldsymbol{x}) \, , \tag{56}$$

where if we maximize Ψ_t, points with larger mean (μ_{t-1}) and larger uncertainty (σ_{t-1}) are preferred based on the previous t-1 observations. The user-defined factor $\sqrt{\beta_t}$ balances the exploitation for minimizing instantaneous regret and the exploration for querying at regions where we are uncertain of the objective values yet. It has been theoretically proved that, under certain conditions, iterative invocations of Ψ_t will make f converge to its true global optimum [83].

With the GP posterior and acquisition function, we use Algorithm 5 to form the basic GP-BO framework in this chapter. The probabilistic surrogate model, which provides the GP prior, is constructed after conducting random sampling in Line-1. The reward function that is characterized by using the acquisition function, is constructed with the current GP posterior in Line-3. Then the expected reward is optimized in order to find the next query point that will be evaluated for updating the GP prior to produce a more informative posterior distribution over the space of the objective functions. Eventually the global optimum would be approached by iteratively obtaining more query points and updating the prior in Lines 2-8.

Algorithm 5. Gaussian-process-based vanilla Bayesian optimization

Input: number N_{init} of initial samples, number T_{max} of maximum iteration

Output: best $f(\boldsymbol{x})$ evaluated until T_{max}

```
1. Place a Gaussian process prior on $f$;    // construct an initial GP model
2. while $t \leq T_{max}$                      // $t = 1, 2, \ldots, T_{max}$
3.     Construct $\Psi$ using current GP posterior;
4.     Solve $\boldsymbol{x}_t$ that optimizes $\Psi_t$;
5.     Evaluate the new point $y_t = f(\boldsymbol{x}_t)$;
6.     update $\Omega_t = \Omega_{t-1} \cup (\boldsymbol{x}_t, y_t)$ and the GP model;
7.     $t = t + 1$;
8. end while
```

6.3. High-Dimensional Many-Objective GP-BO

6.3.1. Additive Structure for High-Dimensional Gaussian Process

Scaling GP-BO to fit high-dimensional variable space is notoriously challenging as widely acknowledged in the literature. Kandasamy *et al.* [42] proposed additive Gaussian process (Add-GP) to tackle this problem by assuming function f is a summation of G disjoint yet additive components, each being an independent function $f^{(g)}$. By definition, $f(\boldsymbol{x}) = \sum_{g=1}^{G} f^{(g)}\left(\boldsymbol{x}^{(\mathcal{S}_g)}\right)$, where $\mathcal{S}_g$ is a subset of the original set $\{D\}$ having D dimensions such that $\boldsymbol{x}^{(\mathcal{S}_g)}$ includes some variables of $\boldsymbol{x}$. Precisely, $\{D\} = \bigcup_{g=1}^{G} \mathcal{S}_g$ and $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset, \forall i \neq j$, $i,j = 1,\ldots,G$. In this chapter, we use *sub-dimension* to refer to those specific variables contained in a subset $\mathcal{S}_g$, which is called *sub-dimensional group g*.

By the assumption of additive property, if $f^{(g)}$ is drawn independently from $\mathcal{GP}(\mu^{(g)}, k^{(g)})$, the resultant f would be a sample from the lumped GP distribution, $\mathcal{GP}(\sum_{g=1}^{G} \mu^{(g)}, \sum_{g=1}^{G} k^{(g)})$. With the same condition of Ω_n observations and σ^2 for Gaussian noise configured in the vanilla GP as listed in Algorithm 5, the log likelihood for observations Ω_n with additive feature can be expressed as,

$$\log p(\Omega_n | \{k^{(g)}, \mathcal{S}_g\}) = -\frac{1}{2}\log|\Sigma| - \frac{1}{2}\boldsymbol{y}_n^{\mathrm{T}}\Sigma^{-1}\boldsymbol{y}_n - \frac{n}{2}\log(2\pi)\,, \tag{57}$$

where matrix $\Sigma = \boldsymbol{K}_n + \sigma^2\boldsymbol{I}$, $\boldsymbol{I}$ is the identity matrix and kernel matrix $\boldsymbol{K}_n$ describes the covariance (i.e., a measure of similarities between points) of the GP random variables. $\boldsymbol{K}_n = \left[\sum_{g=1}^{G} k^{(g)}\left(\boldsymbol{x}_i{}^{(\mathcal{S}_g)}, \boldsymbol{x}_j{}^{(\mathcal{S}_g)}\right)\right]$ for $i,j \leq n$ and $\boldsymbol{x}_i$, $\boldsymbol{x}_j \in \Omega_n$, and $\boldsymbol{y}_n$ would be the shared observation amounts from y_t for all sub-dimensional groups of each $\boldsymbol{x}_t \in \Omega_n$. Thus the posterior mean, variance, and covariance can be accordingly expressed by,

$$\mu_n{}^{(g)}\left(\boldsymbol{x}^{(\mathcal{S}_g)}\right) = \boldsymbol{k}_n{}^{(g)}\left(\boldsymbol{x}^{(\mathcal{S}_g)}\right)^{\mathrm{T}}\Sigma^{-1}\boldsymbol{y}_n ,$$

$$\sigma_n^2\left(\boldsymbol{x}^{(\mathcal{S}_g)}\right) = k^{(g)}\left(\boldsymbol{x}^{(\mathcal{S}_g)}, \boldsymbol{x}^{(\mathcal{S}_g)}\right) - \boldsymbol{k}_n{}^{(g)}\left(\boldsymbol{x}^{(\mathcal{S}_g)}\right)^{\mathrm{T}}\Sigma^{-1}\boldsymbol{k}_n{}^{(g)}\left(\boldsymbol{x}^{(\mathcal{S}_g)}\right) , \tag{58}$$

$$k_n^{(g)}\left(\boldsymbol{x}_i{}^{(\mathcal{S}_g)}, \boldsymbol{x}_j{}^{(\mathcal{S}_g)}\right) = k^{(g)}\left(\boldsymbol{x}_i{}^{(\mathcal{S}_g)}, \boldsymbol{x}_j{}^{(\mathcal{S}_g)}\right) - \boldsymbol{k}_n{}^{(g)}\left(\boldsymbol{x}_i{}^{(\mathcal{S}_g)}\right)^{\mathrm{T}}\Sigma^{-1}\boldsymbol{k}_n{}^{(g)}\left(\boldsymbol{x}_j{}^{(\mathcal{S}_g)}\right)$$

where $\boldsymbol{k}_n{}^{(g)}\left(\boldsymbol{x}^{(\mathcal{S}_g)}\right) = \left[k^{(g)}\left(\boldsymbol{x}_i{}^{(\mathcal{S}_g)}, \boldsymbol{x}^{(\mathcal{S}_g)}\right)\right]$, $i \le n$ and $\boldsymbol{x}_i \in \Omega_n$ for any new observation point $\boldsymbol{x}$ because the function value of $y = f(\boldsymbol{x})$ and the historical observation values of $y_t = f(\boldsymbol{x}_t)$ should follow the joint Gaussian distribution [84]. Thus, the mean in (58) can be used to conduct the probabilistic prediction of y with the confidence of the prediction as reflected by the variance term but with the additive feature.

In addition, the proposed acquisition function of UCB with the additive GP (Add-GP-UCB) [42] can be maximized separately on $\boldsymbol{x}^{(\mathcal{S}_g)}$ for each group g given in (59),

$$\Psi_t^{(\mathcal{S}_g)}\left(\boldsymbol{x}^{(\mathcal{S}_g)}\right) = \mu_{t-1}^{(\mathcal{S}_g)}\left(\boldsymbol{x}^{(\mathcal{S}_g)}\right) + \sqrt{\beta_t}\,\sigma_{t-1}^{(\mathcal{S}_g)}\left(\boldsymbol{x}^{(\mathcal{S}_g)}\right) . \tag{59}$$

Thanks to the additive structure, the exponential sampling complexity can be significantly alleviated when handling high-dimensional space. Moreover, the optimization difficulty of the acquisition function, requiring exponential computation in terms of dimension size, can be largely reduced in the group-wise style.

As for our analog sizing problem, the additive structure naturally exists for the variable search space. That is, a portion of random sizing variables among all show more close relationships to performance than others. A number of groups of variables are formed as per such relationships in different extent. Optimizing the variables with a lower dimension group by group is equivalent to optimizing the target problem with all variables altogether with higher dimensionality [42]. Moreover, a sound dimension splitting scheme can even help bring several variables, which are more correlated and sensitive to some objective attributes, into one group. For instance, in the two-stage Miller Op-Amp shown in Fig. 5(a), the first-stage and second-stage DC gain can be symbolically expressed by $\frac{g_{m1}}{gds1+gds3}$ and $\frac{g_{m6}}{gds6+gds7}$, respectively. If maximizing the DC gain of both stages is one objective, it is more rational to include the design variables closely related to M1/M3 (e.g., W_1, L_1, W_3, and L_3) into one group and M6/M7 (e.g., W_6, L_6, W_7, and L_7) into another group rather than random selection for more effective Ψ optimization.

6.3.2. High-Dimensional Many-Objective GP-BO (HMBO)

The additive structure and the acquisition function (Add-GP-UCB) proposed in [42] provides a practical way to tackle the high dimensionality challenge for GP-BO. In addition, to deal with large observation data in the high-dimensional BO setting, the divide-and-conquer based strategy called Mondrian process [85] is conceived. It is a recursive generative process that partitions the input space in a hierarchical fashion (like decision trees) and divides the data samples by randomly making axis-aligned cuts. The locality of the samples is preserved by enclosing nearby points in one partition. In this chapter, by taking advantage of those advanced machine-learning techniques along with the strength of the ensemble Bayesian optimization (EBO) flow [82], we propose our HMBO as listed in Algorithm 6.

Algorithm 6. High-dimensional many-objective Gaussian-process-based Bayesian optimization (HMBO)

Input: N_{init} initial samples, maximum iteration T_{max}, maximum number of query points B_{max}, maximum number of Mondrian partitions P_{max}, and minimum number of observations in each partition A_{min}

Output: satisfactory $f(\boldsymbol{x})$'s recorded until termination as well as their corresponding $\boldsymbol{x}$'s

```
Generate Gaussian process priors on $\boldsymbol{f}|_{\boldsymbol{i}}$;              // $i$ = 1,…,$N_{obj}$
Initialize splitting parameter $c$, pattern information, and $t$ = 1;
while ($t \le T_{max}$ and performance margin has not been achieved)
    $P = min(\frac{|\Omega_{t-1}|}{A_{min}}, P_{max})$;
    Conduct a Mondrian process to slice the input space into $P$ partitions: $\mathcal{X} = \bigcup_{p=1}^{P} \mathcal{X}_p$, and distribute
    observations accordingly among sliced partitions: $\Omega_{t-1} = \bigcup_{p=1}^{P} \Omega_{t-1}^{p}$;
    for $p$ = 1,…,$P$
        Conduct Tile Coding to discretize $\Omega_{t-1}^{p}$ into feature vectors;
        Use Gibbs-UCB ($c$, pattern information) to derive $c^p$;              // invocation of Algorithm 7
        Construct TileGP$_p|_i$ using $\Omega_{t-1}^{p}$, $c^p$, and feature vectors;
        Use $c^p$ to split $D$-dimensional space to $\bigcup_{g=1}^{G} \mathcal{X}_p^{(\mathcal{S}_g)}$ sub-spaces;
        Construct acquisition function $\Psi_{t-1}^{p}|_i$ by using $\Omega_{t-1}^{p}$, $c^p$, and TileGP$_p|_i$;
        for $g$ = 1,…,$G$
            $\widetilde{\boldsymbol{x}}_t^{p,(\mathcal{S}_g)}$ ← many-objective $\max_{\boldsymbol{x} \in \mathcal{X}_p^{(\mathcal{S}_g)}} \{\Psi_{t-1}^{p,(\mathcal{S}_g)}(\boldsymbol{x})|_i\}_{i=1}^{N_{obj}}$;
        end for
    end for
    Apply correlation clustering on $c^p$ to update $c$;            // Merge $c^p$ to $c$
    if ($B > B_{max}$)       // $B$: number of solutions from all $P$ partitions
        $\{\boldsymbol{x}_t^b\}_{b=1}^{B_{max}} \leftarrow NondominateSort\ \Psi(\boldsymbol{x})|_i^{N_{obj}},\ \forall \boldsymbol{x} \in \{\widetilde{\boldsymbol{x}}_t^{p}\}_{p=1}^{P}$;
    else
        Select all $B$ solutions $\{\boldsymbol{x}_t^b\}_{b=1}^{B}$ as query points;
    end
    Perform multiple function evaluations to get $\{\boldsymbol{y}_t^b\} = f(\{\boldsymbol{x}_t^b\})$;
    Update pattern information based on $\{\boldsymbol{x}_t^b, \boldsymbol{y}_t^b\}$ and related $\{c^p\}_{p=1}^{P}$;
    Update the observation set $\Omega_t = \Omega_{t-1} \cup \{(\boldsymbol{x}_t^b, \boldsymbol{y}_t^b)\}_{b=1}^{N_{query}^{t}}$;
    $t = t + 1$;
end while
```

With N_{obj} objectives (i.e., target circuit performances in our circuit sizing problem), the objective $\boldsymbol{y}$ in bold is a vector, and we conduct N_{obj} independent Gaussian processes with some random samples (i.e., SPICE simulations) to obtain N_{obj} priors in Line-1. To make use of the additive structure, the full space (i.e., including all circuit sizing variables) is split into a number of disjoint groups, each having some unique sub-dimensions, which are controlled by splitting parameter *c* as randomly initialized in Line-2. The combination of sub-dimensions within a group, which we refer to as *pattern* formally defined in Section 6.4.1, has its unique impact on objectives.

Inside each iteration *t*, in Line-5 we conduct Mondrian process to partition the input space. The number of the partitions, *P*, is calculated in Line-4 by the number of observations at iteration *t*-1 (i.e., $|\Omega_{t-1}|$) divided by the user-defined minimum number of observations per partition (i.e., A_{min}), but not exceeding the maximum partitions (i.e., P_{max}) to avoid efficiency degradation. Then the observations from the previous *t*-1 iterations can be automatically collected into *P* partitions by forming $\bigcup_{p=1}^{P} \mathcal{X}_p$. In this way, a problem with larger observations can be divided into a number of sub-problems with smaller observations and later be conquered within each partition. Here we borrow the technique of Tile Coding [82] in Line-7 to discretize the continuous observations Ω_{t-1}^{p} into feature vectors, which are sparse and computationally cheaper to be used for training the TileGP_p (i.e., the additive GP model built inside each partition $\mathcal{X}_p$).

In Line-8, we conduct our proposed performance-driven Gibbs-UCB scheme as detailed in Algorithm 7 to generate dimension splitting parameter c^p for each partition *p*. Next, within each partition *p*, TileGP can be constructed for each objective *i* in Line-9. Then the *p*-th partition, $\mathcal{X}_p$, is further split into *G* sub-spaces ($\bigcup_{g=1}^{G} \mathcal{X}_p^{(\mathcal{S}_g)}$) by using c^p in Line-10. The size of each sub-dimensional group, $|\mathcal{S}_g|$, is constrained to be no more than 10 by default due to the high dimension challenges for GP-BO discussed in Section 2.2.5. Because the conventional acquisition function just works for a single objective, in Line-11 we create acquisition function $\Psi_{t-1}^{p}|_i$ for each objective *i*, and promote its many-objective strategy in the following.

Still inside partition *p*, for each group *g*, we will compose $\boldsymbol{x}_t^{p,(\mathcal{S}_g)}$, the candidate next-query point $\boldsymbol{x}_t^p$ for sub-dimensions only included in subset $\mathcal{S}_g$, by simultaneously optimizing multiple acquisition functions as per multiple objectives in Line-13. Thus, the many-objective focus is emphasized when generating $\boldsymbol{x}_t^{p,(\mathcal{S}_g)}$. Following this group-wise search for all *g*, the full dimension of $\boldsymbol{x}_t^p$ will be constructed after all subsets (i.e., $\{\mathcal{S}_g\}_{g=1}^{G}$) are attempted. Thanks to the additive structure, both the statistical difficulty of exponential sampling complexity for GP regression and the computational challenge of optimizing acquisition function due to high dimensionality can be significantly alleviated. In addition, due to the adopted many-objective acquisition function optimizer, the result for partition *p* is a set denoted by $\widetilde{\boldsymbol{x}}_t^p$ with a user-defined set size. We select the maximization-based Add-GP-UCB (59) with $\beta_t = 2\log\left(\frac{2t^2\pi^2}{\delta}\right) + 2D\log(Dt^3)$, where π is the Archimedes' constant and $\delta \in (0, 1)$, as the acquisition function. In Line-16, the correlation clustering scheme is adopted to merge c^p into *c*, which will be then used in the next iteration.

When the number of candidate query points collected from all the partitions, B=$|\{\widetilde{\boldsymbol{x}}_t^p\}_{p=1}^{P}|$, exceeds the maximum batch size, B_{max}, we conduct the nondominated sorting based on their performance of acquisition functions for all objectives in Line-18, and only obtain a subset solutions $\{\boldsymbol{x}_t^b\}_{b=1}^{B_{max}} \subset \{\widetilde{\boldsymbol{x}}_t^p\}_{p=1}^{P}$. Otherwise, all the candidate solutions from *P* partitions are used for querying. Then the objective function evaluator (i.e., the SPICE simulator in our circuit sizing problem) is called multiple times to evaluate all query points $\{\boldsymbol{x}_t^b\}$ in Line-22. In our case, when one sizing solution is simulated, all performance aspects can be obtained as denoted by $\boldsymbol{y}_t^b$. After querying, *pattern information* will be updated in Line-23, by tracing the patterns controlled by c^p of the corresponding query points $\{\boldsymbol{x}_t^b\}$, which will be detailed in Section 6.4.1.

After the objective values for $N_{query}^t = \text{Min}(B,\ B_{max})$ query points are obtained, we update the observation set Ω_t by including those new points in Line-24. The updated Ω_t will refine the GP surrogate models so as to better approximate the behavior of the objective functions in the subsequent BO iterations. In our circuit sizing problem, a satisfactory solution is the one that can fulfill all circuit specifications. As indicated in Line-3, if the over-constrained specification margin (10% by default) for all performance aspects is achieved by any attempted solution (query point), such a solution is sufficiently satisfactory, and so we break the while loop before reaching T_{max}. In addition, we record all satisfactory circuit sizing solutions along the iterations into a set as output and filter the set via a nondominated sorting operation in terms of the objective performance aspects to obtain a more refined set through post-processing.

6.4. LDE-aware HMBO-based Circuit Sizing

6.4.1. Performance-Driven Dimension-Based Pattern Learning

In Line-10 of Algorithm 6, dimension splitting parameter c^p splits D dimensions into a number of groups, each containing some unique sub-dimensions. As discussed in the example of maximizing the symbolically expressed DC gain in Section 6.3.1, many circuit performances are more closely related to only certain MOSFETs in the circuit. Since such correlation naturally exists in analog circuits, we call the combination of variables (i.e., certain selected sub-dimensions) within each group as *pattern*. As a result, a number of patterns are constructed from total *D* dimensions for each partition p, which is controlled by c^p as shown in Line-10 of Algorithm 6.

The key challenge of making the most of the additive structure is how to generate a good c^p in order to form a sound splitting structure. In [82], given the global dimension splitting parameter, the Gibbs sampling process is used to derive c^p for sampling or grouping sub-dimensions inside partition *p*. Since the partitions are not fixed among iterations, premium query points within partition *p* from the last iteration *t*-1 have chances to be included into newly generated partitions in future iterations, which can favorably bring about implicit exploration capability. For a large number of iterations, optimum patterns might be discovered with the aid of such a mechanism offered by this plain Gibbs sampling scheme. But with limited computation resources (e.g., only several hundred costly circuit SPICE simulations involved [41]), the search quality may strongly suffer. Therefore, we propose a circuit performance-driven dimension-based pattern learning scheme called *Gibbs-UCB* in Algorithm 7 (invoked in Line-8 of Algorithm 6) for learning c^p with reinforced exploitation strength especially efficient for the applications with a small BO iterations (i.e., $T_{max} \leq 100$).

As observed from Lines 10-14 of Algorithm 6, the selected query points $\{\boldsymbol{x}_t^b\}$ are structurally determined by the dimension splitting operation controlled by the splitting parameters $\{c^p\}_{p=1}^P$. That is to say, $\{c^p\}_{p=1}^P$ determines the structural composition of $\{\boldsymbol{x}_t^b\}$ in partition p. For example, assume the sizing task is to optimize a circuit with 10 variables. One c^p, which is represented by an ordered list [1, 2, 3, 2, 1, 2, 3, 2, 1, 2], is already obtained from Line-8 of Algorithm 6, and its list index k (k = 1,…,10) corresponds to one specific sub-dimension for one query point $\boldsymbol{x}_t^1$ generated later. Here there are three unique values (i.e., 1, 2, and 3) in c^p, which are referred to as group labels. The variables with the same group label value (i.e., c_k^p) form one pattern. So there are three patterns contained in c^p, namely, {1, 5, 9}, {2, 4, 6, 8, 10}, and {3, 7} denoted by E_1, E_2, and E_3, respectively. And there are three groups (i.e., G=3) of acquisition function optimization corresponding to E_1, E_2, and E_3, which will finally generate $\boldsymbol{x}_t^1$ (i.e., Lines 12-14 of Algorithm 6).

After function evaluation (i.e., SPICE simulation) with $\boldsymbol{x}_t^1$, we obtain performance $\boldsymbol{y}_t^1$, which can be associated with patterns E_1, E_2, and E_3. Our idea is to associate all attempted patterns with performance information (in t), which can facilitate the selection of premium patterns for deriving better c^p in subsequent iterations (from t+1). We firstly build the association by establishing a fitness metric for $\{c^p\}_{p=1}^P$ and specifically for the involved patterns using the following tactic. After evaluating a query point $\boldsymbol{x}_t^b$, each pattern involved can be associated with a fitness amount calculated by using all the objective attributes (i.e., $\boldsymbol{y}_t^b$) with weighting factors composed in a lumped form [78]. Initially there is no pattern and therefore no fitness information available. While evaluating each query point, all newly identified patterns are associated with the same fitness obtained from performance evaluation of this point. For an old pattern, its associated fitness is updated by the average fitness, which is calculated by the accumulated fitness divided by its occurrence number among all the evaluated query points.

For each pattern denoted by set $E_i \subset \{1, \ldots, D\}$, its associated fitness is denoted by H_i and the number of occurrences is denoted by M_i, where i =1,…,|E|, $E = \{E_i\}$ is the collection of all patterns in the record. For instance, after evaluating the first query point $\boldsymbol{x}_t^1$ if we continue to follow the example above, say we get $E = \{E_i\}_{i=1}^3$ = {E_1, E_2, E_3}, $\{M_i\}_{i=1}^3$ = {1, 1, 1} (i.e., one occurrence for each pattern), and $\{H_i\}_{i=1}^3$={10, 10, 10} (i.e., the same fitness amount of 10 obtained from $\boldsymbol{y}_t^1$). Given the second generated query point $\boldsymbol{x}_t^2$ controlled by c^p = [1, 2, 2, 2, 1, 2, 2, 2, 1, 2], it indicates two patterns {1, 5, 9} (i.e., an old pattern, namely, E_1) and {2, 3, 4, 6, 7, 8, 10} (i.e., a new pattern called E_4). After function evaluation, assuming the obtained fitness amount of $\boldsymbol{y}_t^2$ is 20, then we can get the updated $E = \{E_i\}_{i=1}^{4} = \{E_1, E_2, E_3, E_4\}$, $\{M_i\}_{i=1}^{4} = \{1+1=2, 1, 1, 1\}$, and $\{H_i\}_{i=1}^{4} = \{(10+20)/2 = 15, 10, 10, 20\}$. The *pattern information* in our fitness metric, including $\{E_i\}$, $\{M_i\}$, and $\{H_i\}$, is updated in each iteration *t* in Line-23 of Algorithm 6. It contains the performance-driven feedback from objective function evaluation (i.e., SPICE simulation results), which will determine the priority of pattern selection for deriving the upcoming c^p as illustrated in Algorithm 7 below.

In Line-1 of Algorithm 7, we initialize the ordered list $[c_k^p]_{k=1}^{D}$ as 0 (valid group label starting from 1) for all *D* elements, the pattern group label *l* as 1, as well as set E^0 (E^l for a collection of the attempted non-duplicate desirable patterns) and set *I* (for the indices of unwanted patterns) both as $\emptyset$. We need to have some data to start learning by conducting some Gibbs sampling in Line-3 for directly generating c^p when the current iteration *t* is smaller than $\tau_0 * T_{max}$, a small percentage (2% by default) of the total budget T_{max}. Otherwise, we carry out the dimension splitting process with balanced exploitation (i.e., Lines 6-17 for selecting good patterns from the record) and exploration (i.e., Line-18 for using Gibbs sampling to create new patterns). This is controlled by the dimension threshold parameter, d_{th}, a percentage factor calculated in Line-5. In the early iterations, the number of patterns, $|E|$, and *t* are relatively small. So d_{th} is also small, and thus only a small portion of all *D* dimensions (i.e., $d_{th} * D$) conducts exploitation leaving the remaining majority subject to exploration done in Line-18. As *t* increases, the number of the collected patterns significantly increases. To slow down the increment of d_{th}, the first term is logarithmized to prevent the process from over exploitation with the aid of the user-defined parameter τ_1. Another user-defined parameter $\tau_2 < 1$ can slightly slow down the increment of the 2nd term (i.e., $\tau_2 \frac{t}{T_{max}}$) in Line-5, which avoids exploration starvation as the iteration goes on.

Algorithm 7. Performance-driven pattern learning (Gibbs-UCB)

Input: global splitting parameter c, partition p, iteration t, user-defined τ_1 & τ_2, pattern information including sets $\{E_i\}$, $\{M_i\}$ and $\{H_i\}$ until t-1

Output: local dimension splitting parameter c^p for partition p at t

```
1. Initialize $c^p = [c_k^p]_{k=1}^D = [0]^D$, $l$ = 1, set $E^{l-1} = \emptyset$, and set $I = \emptyset$;
2. if ($t \le \tau_0 * T_{max}$)
3.     Conduct Gibbs sampling to derive $c^p$;
4. else
5.     Compute $d_{th} = \min(\tau_1 \log\frac{|E|}{D} + \tau_2 \frac{t}{T_{max}}, 1)$; // dimension threshold
6.     while ($|\{c_k^p, \forall c_k^p \neq 0\}| \le d_{th} * D$)        // for available sub-dimensions
7.         $i \leftarrow \text{argmax}_{i\in\{1,\dots,|E|\}\setminus I}$ (Eq. (60));
8.         Label $c_k^p$= $l$ for each $k$ location mapped by each element in $E_i$;
9.         $l = l + 1$;
10.        $E^l = E^{l-1} \cup E_i, I = I \cup \{i\}$;        // include desired pattern and index
11.        if ($E^l \cap E_j \neq \emptyset, \forall j \in \{1, \dots, |E|\}\setminus I$)
12.            $I = I \cup \{j\}$;        // include the index of a pattern due to overlap
13.        end if
14.        if ($|I| == |E|$)        // break when running out of available patterns
15.            break;
16.        end if
17.    end while        // composed exploitation-based $|E^l|$ dimensions of $c^p$
18.    $c^p$ ← conduct Gibbs sampling for remaining ($D$ - $|E^l|$) dimensions;
19. end
```

For pattern exploitation, we keep selecting promising patterns generated from the previous t-1 iterations, and labeling them by the group label l as shown in Line-8 as long as the number of the labeled variables from selected patterns does not exceed $d_{th}*D$ as indicated in Line-6. In each ***while*** iteration, the most promising pattern is selected by maximizing a score function that is inspired by the UCB scheme as follows,

$$H_i + \sqrt{\frac{2*\log(\sum_{j=1}^{t-1} N_{query}^j)}{M_i}}, \tag{60}$$

where H_i is the fitness amount of pattern E_i, the logarithm operand is the total number of query points attempted from iteration 1 to t-1, and M_i is E_i's occurrence from all evaluated query points so far. Thanks to Eq. (60), promising patterns in terms of high fitness (i.e., the first term H_i) and large uncertainty (i.e., the second square-root term) may be selected with high priority. That is to say, in our analog circuit sizing problem the performance-sensitive variables among all had better be grouped together since they can be simultaneously optimized for problem objectives in (59) to seek possible higher fitness. Thus, the sensitive groups would likely gain higher fitness much easier than the groups of non-sensitive variables so that the correlated sensitive variables can be intensively exploited within one group, which is in line with the concept of additive structure. In addition, the combinations of variables less frequently attempted (i.e., smaller M_i) would be also encouraged due to higher uncertainty.

After one good pattern E_i is selected, in Line-8 a number of elements on the ordered list $[c_k^p]$, which are mapped by sub-dimensions k's contained in E_i, will be labelled by l. After l increments by 1 in Line-9, the newly selected E_i will be merged to E^l, and E_i will not be considered in the subsequent pattern selection after including index i into I in Line-10. In Lines 11-12, the remaining patterns that contain any overlapping k's with the collected ones in E^l will be excluded in the next-iteration labelling process by updating I. As indicated in Lines 14-16, if we run out of the available patterns before violating the sub-dimension length condition in Line-6, the while loop breaks. After Line-17, associated with the labelled $[c_k^p]$ for the exploitation-based sub-dimensions, final c^p is determined with the aid of the exploration-based Gibbs sampling on the remaining sub-dimensions in Line-18. $|E^l|$ is the sum of set sizes for all selected E_i's, which are previously merged into E^l in Line-10.

In comparison with the plain Gibbs sampling method used in [82], in our proposed scheme the promising patterns are encouraged and prioritized by the UCB-based scoring function (60), which is closely related to the objective attributes (i.e., circuit performance) and uncertainty of candidate patterns (i.e., confidence regarding certain combinations of sizing variables). Such intelligence takes effect via d_{th} in Line-5, which can balance the allocation of optimization resources towards either exploration or exploitation. This tactic plays a core role in Algorithm 7, which enhances performance of Algorithm 6 when applying to our high-dimensional LDE-aware IC sizing problem.

As another benefit of the employed fitness metric for patterns, the optimization effort for the acquisition function in Lines 12-14 of Algorithm 6 can be tuned adaptively to improve algorithmic efficiency. If a pattern being attempted has a smaller score amount calculated with (60), we can assign a correspondingly less configuration of the utilized optimizer with reference to the default configuration (denoted by z_{df}). In this regard, all the attempted patterns in record are ranked according to their score amounts. For a pattern E_i, a ratio r_i is calculated by 1 - $\omega_i/|E|$, where ω_i is the rank of E_i. So z_i, which denotes the configuration for optimizing the variables involved in E_i, can be set as max(r_i*z_{df}, z_{min}), where z_{min} stands for the minimum configuration. As a result, those sizing variables that are more correlated and sensitive to circuit performance can be allocated with higher computational resources. In addition, newly generated patterns but not being evaluated have no entry in the record, and so would still use the default configuration.

6.4.2. Floorplanning and HMBO-Based LDE-Aware IC Sizing

To achieve LDE-awareness when applying the proposed HMBO to the analog circuit sizing problem, our free sizing variables include W, L, nf, LR_{ext}, and SC_t of every MOSFET, nominal values of passive devices (e.g., resistors, capacitors, and inductors) if any, as well as voltage or current biases in the circuit. With the same simplification scheme discussed in Section 5.3. SC_x and SC_y are set as the variables instead of SC_t. Next, when the geometric information contained in the candidate sizing solution (i.e., query point) is available, SAB_{eff} and SCA_{eff} can be calculated by using (43) and (42) to reflect STI effect and WPE. Associated with the basic MOSFET parameters of W, L, and nf as well as the other device properties calculated via Eqs. (52)-(54), the circuit simulation netlist can be constructed to fully take into account LDEs in the simulations invoked during the HMBO-based sizing optimization.

Similar to the floorplanning strategy employed in the previous chapters, the simulated annealing driven B*-tree-based floorplanning method [58] is deployed to generate optimal floorplans for each query point. As the input to the floorplanner, geometric information contained in each query point is utilized to transform device sizes into rectangular blocks. The resultant floorplans are obtained as per the floorplanning objectives (mainly including total area and wire length) and various constraints (e.g., signal flows, resemblance to circuit schematic, and symmetry). The floorplan expressed in the B*-tree representation for a query point can be used as a reference for the global routing. Thus, the definite interrelationship among device blocks can be acquired, which can be further used to derive the shortest wire path for estimating the interconnect parasitics. Because of the better flexibility as discussed in Section 5.3, we offer bonus score to a trial floorplan if any two neighboring MOSFETs share the same type (i.e., PMOS or NMOS).

6.5. Experimental Results

This section highlights the merits of our proposed HMBO-based LDE-aware circuit sizing method by providing our experimental results in comparison with other layout-aware circuit sizing approaches. All experiments in this chapter were conducted in the TSMC 65nm CMOS technology.

In order to show the effectiveness of our novel Gibbs-UCB-based pattern learning scheme used in our proposed HMBO, we keep track of the generated patterns and *hypervolume* along the iteration in comparison with the plain Gibbs sampling based ensemble Bayesian optimization (EBO) [82] and HMBO along with the plain Gibbs sampling (called HMBO-p hereafter) on the circuit sizing problem with LDE-awareness for the two-stage Op-Amp in Fig. 5(a). As one of the most popular quality indicators, hypervolume provides a way to assess and compare the resultant performance among various optimization approaches, especially for multi-objective and many-objective ones. As implied in [38], PF can be obtained by maximizing hypervolume. Therefore, we have adopted hypervolume as a performance metric, the larger the better, for comparing various approaches included in our experiment. In addition, the normalized specification serves as the reference point required for hypervolume calculation so that any solutions whose performance fails to pass any specification will not contribute to hypervolume. The maximum numbers of iterations (T_{max}) and query points per iteration (B_{max}) are set as 50 and 20 respectively in our experiment.

Fig. 15 and Fig. 16 depict the number of generated patterns and hypervolume variation respectively for the two-stage Op-Amp. We have two settings for our proposed Gibbs-UCB-based HMBO method, which are τ_1= τ_2= 0.25 (i.e., called HMBO-1 illustrated by the green curve with stars) and τ_1= τ_2= 0.5 (i.e., called HMBO-2 illustrated by the black curve with dots) where τ_1 and τ_2 (between 0.25 and 0.75 by default) are the factors of d_{th} in Algorithm 7 for balancing the exploration and exploitation towards pattern generation and reuse. In Fig. 15, for the plain Gibbs sampling based EBO approach (i.e., the blue curve with triangles), the number of generated patterns increases in approximately linear manner. The ratio is controlled by B_{max} because more patterns are generated when more query points are attempted in each iteration. For HMBO-p (i.e., the red curve with diamonds), there are more patterns generated due to the many-objective feature of Algorithm 6. As observed from Fig. 16 for both plain Gibbs-based schemes, even though the hypervolume in HMBO-p starts at a low level but improves late (i.e., at iteration 5), it manages to override EBO (i.e., at iteration 17) and maintains the leading state till the 50th iteration thanks to the many-objective feature designed for the proposed HMBO in comparison to EBO.

For any curve depicted in Fig. 15, if the slope of tangent at one point starts to become flat, it means that there are less new patterns being composed and more old patterns being reused, which accordingly indicates decreased exploration strength and increased exploitation momentum for optimum pattern search. The slope at different points of the curves varies more obviously for HMBO-1 and HMBO-2 than EBO and HMBO-p, which exhibits an adapted mechanism over pattern exploration and exploitation thanks to our proposed Gibbs-UCB scheme.

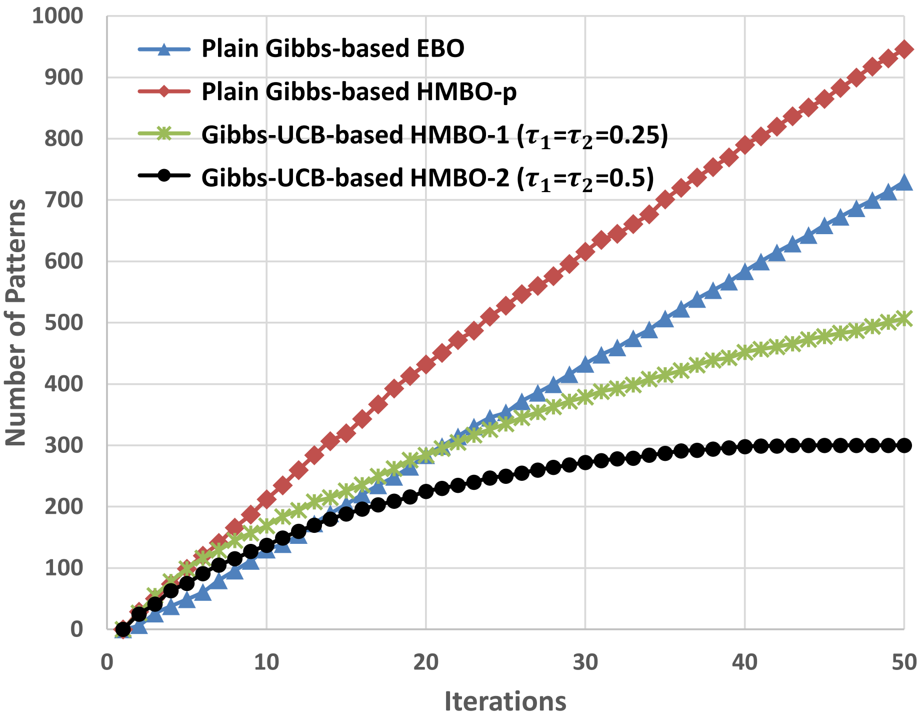


Fig. 15. Pattern generation along iterations for the two-stage Op-Amp

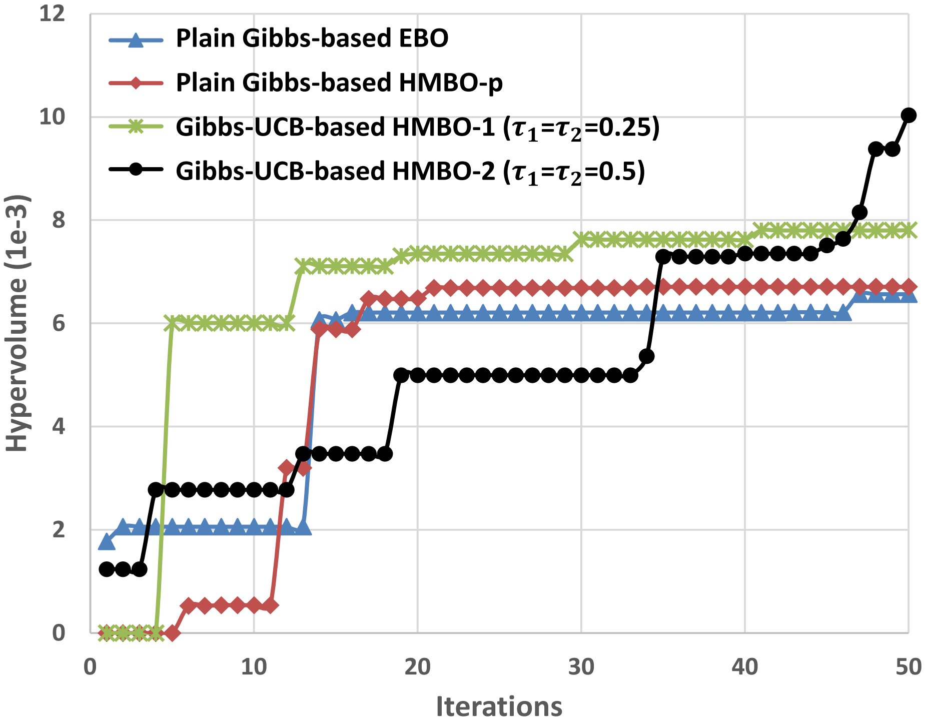


Fig. 16. Hypervolume variation along iterations for the two-stage Op-Amp

In comparison with HMBO-2 in Fig. 16, HMBO-1 holds the leading state until iteration 46 thanks to a large hypervolume jump achieved at iteration 5, which is intrinsically attributed to its larger exploration strength (i.e., larger slope of tangent in Fig. 15). As shown in Fig. 15 for HMBO-2, the exploitation starts to make larger impact yet still can enrich some pattern diversity around iteration 20, which is reflected by a flatter curve compared to HMBO-1. After iteration 30, there seems a strong exploitation strength for HMBO-2, and there is barely increment of new patterns after iteration 35. Accordingly in Fig. 16 for HMBO-2, there are two obvious boosts of hypervolume around iteration 20 and iteration 35, where the exploitation strength is becoming strong and then dominant. These observations suggest that the enhanced exploitation by reusing old patterns (around the 20th and 30th iterations) can contribute to the advancement of hypervolume. In addition, with even stronger exploitation strength (after iteration 35), there will be more opportunities to focus on attempted promising groups of performance-sensitive sizing variables with relative high scores (60). In addition, these focused groups of variables would stay being refined group by group via (59) with more optimization resources allocated (i.e., larger r_i*z_{df}) because of their higher scores (i.e., higher ranking ω_i's). This helps justify that after iteration 46, there is still potential for HMBO-2 to further improve hypervolume and finally surpass HMBO-1.

In contrast, the hypervolume of the plain Gibbs-based schemes (i.e., EBO and HMBO-p) is obviously inferior to that of the Gibbs-UCB-based schemes (HMBO-1 and HMBO-2). This suggests that without any care of pattern reuse for enhancing exploitation, performance-sensitive variables might not be readily optimally grouped and attentively refined, and thus it may take longer iterations for hypervolume to be improved. In contrast, the intelligence introduced by the UCB-based score function (60) in our proposed Algorithm 7 can help exploit sound patterns and contribute to the sizing performance enhancement.

In addition to the Op-Amp depicted in Fig. 5(a), a differential-pair comparator shown in Fig. 5(b) is also used in our experiment. For each experimental circuit, seven heuristic-based or statistical-based schemes are compared with one another. Among evolutionary algorithm based Schemes 1-3, Scheme-1 follows the Synthesis Flow for fast Parasitic Closure (called *SFPC* for short) [59], which includes placement and global routing inside a refined-sizing loop. One layout-aware sizing work [33], which uses differential evolution (DE), is implemented as Scheme-2. Scheme-3 imitates one state-of-the-art many-objective evolutionary algorithm called θ-DEA [38] applied to the layout-aware circuit sizing problem. For the Gaussian-process-based Bayesian optimization (GP-BO) related Schemes 4-7, the ensemble BO (EBO) [82] and the Multi-objective ACquisition Ensemble BO (MACE for short) [41] are applied to the LDE-aware circuit sizing as Scheme-4 and Scheme-5, respectively. Our proposed high-dimensional many-objective GP-BO (called HMBO) with adaptive Gibbs-UCB scheme for pattern learning is applied to the layout-aware circuit sizing as Scheme-7. To highlight the effectiveness of the proposed Gibbs-UCB scheme, a plain Gibbs sampling based scheme that replaces the Gibbs-UCB in HMBO is configured in Scheme-6 (i.e., HMBO-p) as the only difference from Scheme-7. For a fair comparison among all the seven LDE-aware circuit sizing schemes, the floorplan can be derived whenever a trial solution (EA chromosome or BO query point) is available, and be used to estimate the interconnect parasitics. In addition, by following Eqs. (42) (43) and (52)-(54), the LDE effects are considered via the netlist included in the circuit simulations for all schemes.

For the configuration of variable bounds, they are [0.5μm, 150μm] with step-size of 10nm for W, [60nm, 1μm] with step-size of 5nm for L, and [1, 50] with step-size of 1 for nf. nf will be automatically adjusted to ensure W/nf and L are compatible with the design rules of the used technology process. LR_{dum}, whose bound is set as [0, 750nm], is used to linearly adjust SAB_{edge} for

STI effect. For WPE, the bounds of SC_x and SC_y are configured by [0, 500nm] and [0, 1μm], respectively. The bounds for LR_{dum} as well as SC_x and SC_y are set so since the stress effect and proximity effect would quickly diminish after the distances of SAB_{edge} and SC_t are over 1μm in our adopted technology.

In both Table 22 and Table 23, the type of optimization including single-objective, multi-objective, and many-objective is specified for all Schemes 1-7. For Schemes 4-7, the size of the simulation-based initial training data set is 100. The number of query points per iteration (i.e., batch size) is set as 10 with maximum 50 iterations except for Scheme-5 where the batch size is set as 4 and the maximum iteration is set as 125 (i.e., a loyal implementation to [41]). Thus, the maximum number of the involved simulations for Schemes 4-7 is 600. For Schemes 2-3, the evolutionary size of population and the maximum generation are 36 and 17 respectively, which leads to 612 simulations. 600 simulations are reasonably configured in Scheme-1. For each scheme, the nondominated solutions are obtained only from valid solutions that pass the specification. The numbers of those solutions are reported in the fourth and fifth rows.

In addition to the adopted hypervolume metric in both Table 22 and Table 23, our maximization-based fitness (41), which is a summation of various normalized performance attributes similar to [78], is used as another figure of merit but negated as a negative value (i.e., thus 0 as the ideal maximum) in order to be compatible with the maximization-oriented UCB. In the seventh row, “Best-fitness” is the solution that has the largest fitness within the resultant solution set. This best-fitness solution is also selected as the representative solution for each scheme to exhibit detailed performance of various objective attributes.

For the two-stage Op-Amp in Table 22, among the EA-based schemes, the DE-based Scheme-2 has the worst hypervolume (i.e., 2.01) and moderate best-fitness (i.e., -0.631). The many-objective based θ-DEA (i.e., Scheme-3) is able to derive a solution set with a higher hypervolume (i.e., 5.70) over Schemes 1-2 (i.e., 5.36 and 2.01 respectively). However, its best-fitness is slightly inferior to those of Schemes 1-2 because the aim of the single-objective based approaches is to converge to a better fitness value while the goal of the many-objective approaches is to simultaneously improve all performance attributes, which may not necessarily lead to the best fitness. The sophisticated θ-DEA based Scheme-3 is expected to have better best-fitness solutions if provided with larger evolutionary resources and reasonable configuration of fitness function.

Table 22. Settings and performance of the two-stage Op-Amp

Two stage Op-Amp	**EA Based**			**GP-BO Based**			
Schemes/Performances	**Sch-1 SFPC** [59]	**Sch-2 DE** [33]	**Sch-3 θ-DEA** [38]	**Sch-4 EBO** [82]	**Sch-5 MACE** [41]	**Sch-6 Plain Gibbs-based HMBO-p**	**Sch-7 Gibbs-UCB based HMBO [This work]**
Optimization Type	Single-Obj.	Single-Obj.	Many-Obj.	Single-Obj.	Pseudo Multi-Obj.	Many-Obj.	Many-Obj.
Spec.-passed Solutions	10	7	10	8	2	6	6
Nondominated Solutions	8	6	4	8	2	5	4
Hypervolume (*1e-3)	5.36	2.01	5.70	4.72	2.05	5.32	7.81
Best-fitness	-0.615	-0.631	-0.653	-0.624	-0.613	-0.615	-0.586
Objectives & Specification	**Representative Solution (from the one with the largest fitness)**						
Gain > 60dB	64.84	60.81	61.60	61.89	60.50	62.17	62.23
UGB> 4MHz	17.91	8.20	40.69	12.52	18.15	26.31	21.03
PM > 60°	71.43	80.39	70.94	73.83	75.93	78.89	75.04
GM > 15dB	31.80	49.62	21.65	38.19	33.26	25.68	38.42
Runtime (hrs)	0.50	0.39	0.42	1.51	1.22	1.93	1.90

Among the GP-BO based Schemes 4-7, the best-fitness from Scheme-5 (i.e., -0.613) is not that bad because although multiple acquisition functions are employed, this MACE method is still able to focus on improving a single metric (i.e., fitness). However, since there is no consideration of handling high-dimensional variable space, it has only two specification-passed and nondominated solutions, which result in a low hypervolume of 2.05. Between Scheme-4 and Scheme-6 both having the consideration of high dimensionality, the many-objective feature included in Scheme-6 leads to a higher hypervolume (i.e., 5.32) and a slightly higher best-fitness (i.e., -0.615) than those (i.e., 4.72 and -0.624 respectively) in Scheme-4. As for our proposed Scheme-7, thanks to the proposed Gibbs-UCB scheme for pattern learning, the hypervolume (i.e., 7.81) and best-fitness (i.e., -0.586) could even outstrip those in Scheme-6 (i.e., 5.32 and -0.615 respectively) equipped with the plain Gibbs sampling scheme. Furthermore, when handling multiple optimization objectives, the single-objective based approaches, which rely on the user-defined fitness function, may have a risk of ending up to the solutions having good fitness but narrow performance margin. For example in Scheme-5, the representative solution with relatively good fitness of -0.613 among all seven schemes only has DC gain of 60.50dB, which might make it hard satisfy certain subsequent verification under PVT variations. For the runtime, since there is no involvement of model training and inference as well as acquisition function optimization, the EA-based Schemes 1-3 are 2.44 - 4.95 times faster than the GP-BO based Schemes 4-7. However the hypervolume (i.e., 2.01) from the fastest Scheme-2 is 3.89 times less than Schemes-7’s (i.e., 7.81). Compared to our proposed Scheme-7, there are less CPU resources required if a single-objective problem is targeted (i.e., Schemes 4-5) or there is no splitting/learning strategy on high-dimensional input (i.e., Scheme-5).

For the comparator circuit in Table 23, propagation delay is one of the most important circuit characteristics, while the positive and negative overshoots are given in absolute values. Among the EA based Schemes 1-3, the SFPC Scheme-1 could not locate any premium region, which ends up with only one valid solution as well as poor hypervolume and best-fitness (i.e., 0.012 and -

0.737, respectively). The search configuration space could be highly bumpy for the comparator circuit because it consists of multiple continuous regions but disjoint in between. This nature can be understood in that when the logic balance is broken by sufficient variations of device sizes, parasitics, and LDEs during the optimization, the charge and discharge of current paths for the output would immediately reverse. Scheme-2 successfully located several continuous regions as indicated by its large number of valid solutions and acceptable amount of nondominated solutions (i.e., 123 and 11, respectively). However, being trapped in suboptimal regions due to the nature of DE, its hypervolume and best-fitness are still inferior to the rest of Schemes 3-7. The θ-DEA Scheme-3 demonstrates its efficacy on this bumpy-search-space problem by yielding competitive hypervolume and best-fitness (i.e., 0.565 and -0.197) even in comparison with the GP-BO based Schemes 4-6.

Table 23. Settings and performance of the differential comparator

Two stage Op-Amp	**EA Based**			**GP-BO Based**			
Schemes/Performances	**Sch-1** **SFPC** [59]	**Sch-2** **DE** [33]	**Sch-3** **θ-DEA** [38]	**Sch-4** **EBO** [82]	**Sch-5** **MACE** [41]	**Sch-6** **Plain Gibbs-based HMBO-p**	**Sch-7** **Gibbs-UCB based HMBO** **[This work]**
Optimization Type	Single-Obj.	Single-Obj.	Many-Obj.	Single-Obj.	Pseudo Multi-Obj.	Many-Obj.	Many-Obj.
Spec.-passed Solutions	1	123	15	12	27	106	51
Nondominated Solutions	1	11	10	3	8	18	9
Hypervolume	0.012	0.263	0.565	0.535	0.544	0.581	0.625
Best-fitness	-0.737	-0.402	-0.197	-0.184	-0.259	-0.232	-0.180
Objectives & Specification	**Representative Solution (from the one with the largest fitness)**						
Propagation Delay < 250ps	168	147	81	88	131	108	68
+Overshoot < 350mV	316	119	35	36	79	48	34
-Overshoot < 150mV	96	41	15	15	4	19	25
Runtime (hrs)	0.54	0.45	0.49	1.12	0.95	1.44	1.43

Among all of GP-BO based Schemes 4-7, Scheme-5 features some multi-objective qualities thanks to the utilized multi-objective acquisition functions despite the plain single-objective FOM for circuit performance attributes. So higher numbers of valid solutions and nondominated solutions (i.e., 27 and 8) are obtained, which leads to slightly higher hypervolume of 0.544 in comparison with those (i.e., 12 specification-passed solutions, 3 nondominated solutions, and hypervolume of 0.535) in Scheme-4. In contrast, without any care about multi-objective benefits, the full optimization effort in Scheme-4 is solely devoted to improving the fitness along one search path in comparison with Scheme-5. Thus, the best-fitness of -0.184 in Scheme-4 is better than that (i.e., -0.259) in Scheme-5. In contrast, Scheme-6 with our proposed high-dimensional many-objective GP-BO framework could override the performance of Scheme-5 regarding the numbers of valid and nondominated solutions as well as hypervolume and best-fitness. In addition, despite less number of valid solutions and therefore less nondominated solutions observed in our propose Scheme-7 in comparison to Scheme-6, more superior performances of hypervolume and best-fitness (i.e., 0.625 and -0.180 over 0.581 and -0.232, respectively) are obtained in Scheme-7. This indicates that although the plain-Gibbs-sampling based pattern learning scheme utilized in Scheme-6 could find more solutions due to its significant exploration strength, our proposed Gibbs-UCB based scheme can actually improve the quality of solution set with balanced exploration and exploitation especially when the computation resources (e.g., allowable maximum batch size and BO iterations) are limited in the analog circuit sizing problem. These experimental results help justify the effectiveness of our proposed HMBO in Scheme-7.

6.6. Summary

In this chapter, an efficient high-dimensional many-objective Gaussian-process-based Bayesian optimization methodology called HMBO was presented to optimize the challenging LDE-aware analog circuit sizing problem. The layout dependent effects including STI and WPE were well modeled by accurate estimation of LDE parameters including *SA*/*SB* and *SCA*/*SCB*/*SCC*. Moreover, our developed performance-driven pattern learning Gibbs-UCB scheme can contribute to superior structural splitting of high dimensionality. Our experimental results clearly demonstrate its efficacy by comparing with other heuristic-based layout-aware circuit sizing approaches.

Chapter 7 Conclusion and Future Work

In this dissertation, we have first explained the necessity of EDA for analog/RF circuit synthesis and then proposed multiple novel methodologies to address critical challenges in the area of analog/RF integrated circuit sizing. We mainly identified two kinds of prominent layout effects, parasitics and layout-dependent effects (LDEs), which may incur severe impact on circuit performance degradation. They cannot be fully detected until a schematic is converted to its corresponding layout in the traditional analog IC design flow. Thus, analog designers may have to go back to the completed schematic stage to pursue another design solution if the performance degradation due to the parasitics and LDEs cannot be alleviated by any subsequent layout refinement. In such cases, plenty of tweaking effort including re-sizing, re-placement, and re-routing is expected to close the synthesis loop. As an appealing idea, early actions can be taken in the circuit sizing stage for early awareness of parasitics and LDEs, which is expected to alleviate the prospective trouble in the subsequent layout design stage. A widely accepted term for such an idea is called layout-aware circuit sizing for analog/RF integrated circuits. The main contribution of this dissertation is the proposed algorithms and methodologies to consider parasitics and LDEs in the early circuit schematic design stage.

We have first proposed a two-phase hybrid circuit sizing flow including a symbolic phase (i.e., geometric programming (GeoP)) and a heuristic optimization phase (i.e., evolutionary algorithm (EA)). Our proved theorem shows the GeoP-compatibility for floorplan and interconnect parasitic constraints considered in the symbolic circuit sizing of GeoP platform. After taking the quick solution solved as a global view from the first GeoP-based sizing phase, we apply an EA-based optimization phase for sizing refinement. It involves two EAs including a single-objective EA (i.e., DE) and a many-objective EA (i.e., θ-DEA) for adaptively fit the target problem. To effectively shrink the search scope caused by the variation of both device sizes and parasitics, we maintain a stable floorplan optimized by an SA-driven floorplanner with B*-tree representation in the EA sizing phase. Our experimental results show that the circuit knowledge information induced by the first GeoP phase tends to effectively facilitate the EA-phase optimization process, especially for the sizing problems with complex solution space. It also demonstrates the time efficiency and favorable circuit performance resolution of our proposed two-phase hybrid sizing method over the other similar works with supported pre-layout and post-layout simulation results for three analog/RF circuits in different technologies. Nevertheless, the modeling difficulty and limited accuracy pose a big challenge to fidelity of the first GeoP phase for the general analog/RF integrated circuits.

By maintaining the concept of symbolic plus heuristic two-phase sizing, we have improved the accessibility and accuracy of the first symbolic sizing phase by replacing the GeoP-based circuit modeling with g_m/I_D-based MINLP one, and updated the synthesis flow accordingly. As one of our main contributions, thanks to the involved numerical simulations, the advocated curve-fitting based equations are more accurate than the traditional equations for measuring device attributes (e.g., g_m, g_{ds}, and C_{ij}). For other contributions in this part of dissertation, we have firstly proposed the L-selection (or L-initialization) algorithm in order to avoid selecting improper L values that might account for repeated sizing failures in the subsequent g_m/I_D-based modules. We have then identified that each of the g_m/I_D-parameters (i.e., g_m/I_D, g_{ds}/I_D, I_{DN}, and C_{ij}/I_D) and a set of MOSFET node voltages (i.e., V_{GS} and V_{DS}) have one-to-one correspondence for a selected L. The relationships are obtained from accurate SPICE numerical simulations in a specific technology and then curved fitted into symbolic equations, which feature both generality and enhanced accuracy over the GeoP-based approach. Then we build the MINLP to include the free variables of I_D and node voltages in addition to the g_m/I_D-parameters symbolically expressed in free variables, and eventually solve for device sizes. Due to the accuracy concern on application region of W, we have proposed current density factor (*CDF*) to refine the original g_m/I_D sizing principle. In addition, because the curve-fitted relationships have some dependences on the selected reference W, multiple reference W's are used for improving fitting accuracy. In this way, we have connected those piecewise-based fitting equations in the mix-integer fashion and employed the MINLP solver for this purpose. Due to the disadvantages of either keeping one out-of-date floorplan template or performing floorplanning all the time, we have proposed the compatibility-aided adaptive floorplan variation scheme, which only need to rerun the floorplanner when the current floorplan is not compatible with varied device sizes during the second θ-DEA based sizing phase. Our experimental results have demonstrated the effectiveness of the adaptive floorplan variation scheme employed in our proposed g_m/I_D-EA two phase hybrid sizing methodologies over other similar works by delivering much better fitness and favorable run time for three analog/RF circuits in CMOS 65nm technology.

Beyond parasitics, we can also optimize LDEs by taking advantages of the trustworthy symbolic-based sizing platform (i.e., g_m/I_D-based MINLP sizing) developed before. Firstly, we have modeled the two types of LDEs including WPE and STI with formulations for the early design stage. Based on the g_m/I_D-based LDE-free sizing solution, we have integrated the sensitivity-analysis-based constraints regarding the normalized branch current and circuit performance to perform the LDE-aware symbolic sizing via another MINLP. We have advocated to optimize the two sets of design parameters (W and nf) and (LR_{ext} and SC_t) in two separate inner iterations of the LDE-aware sizing module due to their different degrees of impact on circuit performance. Furthermore, we have formulated the symbolic modeling and made it LDE-aware for calculating other device geometric parameters and included them in the second θ-DEA based sizing phase. In addition, we have refined our floorplanning strategy by favoring a unified larger well that encloses a bunch of devices with the same well type in comparison to multiple isolated wells. In our experimental results, we have used a case study on a single MOSFET to illustrate that our proposed LDE-aware device characterization model can reduce more than 50% estimation error on MOSFET electrical characteristics in comparison to Cadence (a commercial EDA tool). Moreover, the effectiveness of the proposed LDE-aware two-phase g_m/I_D-EA hybrid circuit sizing methodology is demonstrated in comparison to other similar works by reporting and analyzing statistically best-fitness amount, IGD, and successful runs among 10 runs as well as the detailed performance from the best run for several analog circuits in CMOS 65nm technology.

With emphasis on utilizing statistics to manage probabilistic models and uncertainty, machine-learning-based approaches as another heuristic-based category have been increasingly getting popular over the last decade. Gaussian-process-based Bayesian optimization (GP-BO) as a good candidate to solve black-box optimization problem has been applied to analog circuit sizing domain, however not with any consideration of *nf*, parasitics, or LDEs. Because of the intricacy when mingling the conventional sizing variables, LDE parameters, layout floorplan and parasitic considerations, the increased input dimension poses a critical challenge for applying the regular GP-BO to analog circuit sizing. Thus, we have been motivated to propose a high-dimensional many-objective GP-BO (called HMBO) algorithm for the layout-aware analog circuit sizing. We have appropriately employed the Mondrian process to cut the input space in terms of variables' bounds, utilized additive structure to split the input variables' dimension (i.e., grouping more correlated variables) in order to divide and conquer the high dimensionality, and taken the advantages of the EBO flow to devise our HMBO. In addition, we have proposed a performance-driven parameter learning scheme, called Gibbs-UCB, for pattern generation and selection. The UCB-based score function used in this scheme can help group or select performance-sensitive variables among all combinations and exploit the selection to seek for higher circuit performance. This pattern learning scheme can adaptively balance the optimization strength between exploration and exploitation regarding the available resources along the iteration. In addition, we have allocated more optimization resources on favored patterns with higher score during the group-wise acquisition function optimization. In our experimental results, higher best-fitness amount and hypervolume are achieved by using our proposed HMBO method with Gibbs-UCB scheme embedded in comparison to other similar works running for several analog circuits in CMOS 65nm technology.

Even though the topology-dependent circuit performance equations that do not alter with technology variation are relatively trustable for approximating the real circuit performance, manipulation of analytical-based sizing methods that utilize them can still be challenging in practice. This is because there are two levels of nonlinear relationships involved in circuit sizing. The first one is between device geometric sizes and MOSFET characteristics (e.g., g_m and g_{ds}) as well as intrinsic parasitics (e.g., C_{gs} and C_{ds}), while the other one lies between the MOSFET characteristics and circuit performance, not to mention the consideration of floorplan constraints, interconnect parasitics, and LDEs. This makes the analog/RF circuit sizing as a complex black-box optimization problem. In addition, even though the performance equations for popular circuit topologies are available from textbooks and literature, the development efforts of the performance equations for new topologies are normally not tractable in practice.

However, stochastic-based sizing approaches with accurate numerical simulations involved, which can discard those equations, seem to be a good alternative. In the era of artificial intelligence

(AI), plenty of novel machine-learning based methods and algorithms such as Bayesian network, reinforcement learning, and neural network have been already proposed and applied to a rich variety of disciplines and sub-fields. The concept of heuristics reflected from these techniques are well suited to meet the requirement of stochastic-based sizing approaches. Nevertheless, as a recommendation to the research in this domain, special emphasis should be placed onto their application to analog/RF EDA. Due to the costly simulation run time, we can only train those AI models with a limited number of real simulation data. It then becomes important to effectively search the solution space or manage the search exploration versus exploitation by integrating domain knowledge of analog/RF circuits (e.g., performance sensitivity) in order to put more focus on promising data to be sampled and used to improve the AI models. As one of our future works, the aforementioned machine-learning based techniques would be investigated for analog circuit optimization.

PVT stands for the process, supply voltage, and operating temperature. It is understood that in everyday operation, the supply voltage and operating temperature can be fluctuating. Process variation refers to the deviations in the semiconductor fabrication process, which can be caused by non-uniform conditions during depositions and/or diffusions of the impurities. This leads to the variations of sheet resistance and transistor parameters like V_{th}. In addition, W/L variations are found due to limited resolution of the photolithographic process. Analog circuit sizing with tolerance design consideration from inevitable variations of manufacturing process and operating conditions are called PVT-variation-aware sizing. It is further divided into two sub-tasks including minimizing the worst-case performance deviation and maximizing the yield, which satisfies the performance specification regardless of performance variations due to PVT [86]. Apparently, there would be more parameters, which need fine tuning optimizations in order to have a stable worst-case performance and maximum yield (by using the design centering technique [86], for example). Therefore, it becomes our next challenge in the future work to comprehensively and efficiently consider layout parasitics, LDEs, and PVT variations all in the early schematic sizing stage of circuit synthesis.

As a promising next-generation device, Fin Field-Effect Transistor (FinFET) has captured the attention of both digital and analog circuit designers. One key advantage of FinFET devices is that they have more drive current per unit area (i.e., current density) indicating a higher intrinsic gain than that of the planar CMOS devices at the same technology node. However, as the disadvantages of FinFETs, the conducting channel is harder to be controlled, and the higher source-to-drain resistance reduces transconductance. So the consideration of layout parasitics and LDEs in the context of FinFET circuit sizing would need further investigation in our future work.

References

[1] W. Kruiskamp and D. Leenaerts, "DARWIN: CMOS opamp synthesis by means of a genetic algorithm," in *Proc. IEEE Design Automation Conference (DAC)*, pp. 433-438, 1995.

[2] E. Martens and G. Gielen, "Classification of analog synthesis tools based on their architecture selection mechanisms," *Integration, the VLSI Journal,* vol. 41, no. 2, pp. 238-252, 2008.

[3] R. A. Rutenbar, "Analog layout synthesis: what's missing?," in *Proc. ACM/SIGDA, International Symposium on Physical Design (ISPD)*, p. 43, 2010.

[4] L. Zhang and U. Kleine, "A novel analog layout synthesis tool," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. V101-V104, 2004.

[5] T. Liao and L. Zhang, "Analog integrated circuit sizing and layout dependent effects: a review," *Microelectronics and Solid State Electronics,* vol. 3, no. 1A, pp. 17-29, 2014.

[6] L. Wei, F. Boeuf, T. Skotnicki, and H. S. P. Wong, "CMOS technology roadmap projection including parasitic effects," in *Proc. International Symposium on VLSI Technology, Systems, and Applications*, pp. 78-79, 2009.

[7] T. Liao and L. Zhang, "An LDE-Aware gm/ID-Based Hybrid Sizing Method for Analog Integrated Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* 2020.

[8] Calibre PEX, Mentor Graphics Inc., https://www.mentor.com/.

[9] L. Zhang and Z. Liu, "Directly performance-constrained template-based layout retargeting and optimization for analog integrated circuits," *Integration, the VLSI Journal,* vol. 1, pp. 1-11, 2011.

[10] G. Shomalnasab and L. Zhang, "New analytic model of coupling and substrate capacitance in nanometer technologies," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 23, no. 7, pp. 1268-1280, July 2015.

[11] M. V. Dunga, X.-M. (J.) Xi, J. He, W.-D. Liu, K.-Y. M. Cao, X.-D. Jin, J. J. Ou, M.-S. Chan, A. M. Niknejad and C.-M. Hu, "BSIM4.5.0 MOSFET model – user's manual,"

Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Chapters13-15, 2004.

[12] Compact Model Council (CMC), "Guidelines for extracting well proximity effect instance parameters," Version 2.0, March 10, 2006. [Online]. Available: http://www.si2.org/cmc_index.php.

[13] L. Zhang, R. Raut, and Y. Jiang, "A placement algorithm for implementation of analog LSI/VLSI systems," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. v77-v80, 2004.

[14] M. Torabi and L. Zhang, "Efficient ILP-based variant-grid analog router," in *Proc. 2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, Montreal, QC, pp. 1266-1269, 2016.

[15] M. d. M. Hershenson, S. P. Boyd, and T. H. Lee, "Optimal design of a CMOS OpAmp via geometric programming," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 20, no. 1, pp. 1-21, 2001.

[16] S. Boyd, S.-J. Kim, L. Vandenberghe, and A. Hassibi, "A tutorial on geometric programming," 2007. [Online]. Available: http://stanford.edu/~boyd/papers/gp_tutorial.html.

[17] A. A. I. Ahmed and L. Zhang, "Fast parasitic-aware synthesis methodology for high-performance analog circuits," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2155-2158, 2012.

[18] G. Shomalnasab, H. Heys, and L. Zhang, "Analytic modeling of interconnect capacitance in submicron and nanometer technologies," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2553-2556, 2013.

[19] W. Gao and R. Hornsey, "A power optimization method for CMOS Op-Amps using sub-space based geometric programming," in *Proc. Design, Automation and Test in Europe (DATE)*, pp. 508-513, 2010.

[20] Y. Zhang, B. Liu, B. Yang, J. Li, and S. Nakatake, "CMOS op-amp circuit synthesis with geometric programming models for layout-dependent effects," in *13th International Symposium Quality Electronic Design (ISQED)*, pp. 464-469, 2012.

[21] D. Flandre, A. Viviani, J. P. Eggermont, B. Gentinne, and P. G. A. Jespers, "Improved synthesis of gain-boosted regulated-cascode CMOS stages using symbolic analysis and

gm/ID methodology," *IEEE Journal of Solid-State Circuits,* vol. 32, no. 7, pp. 1006-1012, 1997.

[22] Y.-L. Chen, Y.-C. Ding, Y.-C. Liao, H.-J. Chang, and C.-N. J. Liu, "A layout-aware automatic sizing approach for retargeting analog integrated circuits," in *IEEE International Symposium on VLSI Design, Automation, and Test (VLSI-DAT)*, pp. 1-4, 2013.

[23] P. G. Jespers, The Gm/Id Methodology, a Sizing Tool for Low-Voltage Analog CMOS Circuits, Boston, MA, USA: Springer, 2010.

[24] M. N. Sabry, H. Omran, and M. Dessouky, "Systematic design and optimization of operational transconductance amplifier using gm/ID design methodology," *Microelectronics Journal,* vol. 75, pp. 87-96, 2018.

[25] A. Girardi, F. P. Cortes, and S.Bampi, "A tool for automatic design of analog circuits based on gm/ID methodology," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 4643–4646, 2006.

[26] E. Tlelo-Cuautle and A. C. Sanabria-Borbon, "Optimising operational amplifiers by evolutionary algorithms and gm/Id method," *International Journal of Electronics,* vol. 103, no. 10, pp. 1665-1684, 2016.

[27] C. Enz, F. Chicco, and A. Pezzotta, "Nanoscale MOSFET modeling: Part 1: The simplified EKV model for the design of low-power analog circuits," *IEEE Solid-State Circuits Magazine,* vol. 9, no. 3, pp. 26-35, 2017.

[28] D. M. Binkley, C. E. Hopper, S. D. Tucker, B. C. Moss, J. M. Rochelle, and D. P. Foty, "A CAD methodology for optimizing transistor current and sizing in analog CMOS design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 22, no. 2, pp. 225-237, 2003.

[29] C.-W. Lin, P.-D. Sue, Y.-T. Shyu, and S.-J. Chang, "A bias-driven approach for automated design of operational amplifiers," in *International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, pp. 118–121, 2009.

[30] Y.-C. Liao, Y.-L. Chen, X.-T. Cai, C.-N. Liu, and T.-C. Chen, "LASER: Layout-aware analog synthesis environment on laker," in *Proc. the 23rd ACM international conference on Great lakes symposium on VLSI*, New York, New York, USA: ACM Press, pp. 107-112, 2013.

[31] G. Pollissard-Quatremère, G. Gosset, and D. Flandre, "A modified gm/ID design methodology for deeply scaled CMOS technologies," *Analog Integrated Circuits and Signal Processing,* vol. 78, no. 3, pp. 771-784, 2014.

[32] R. Storn and K. Price, "Differential evolution–a simple and efficient adaptive scheme for global optimization over continuous spaces: technical report TR-95-012," International Computer Science, Berkeley, California, 1995.

[33] P. Vancorenland, G. V. d. Plas, M. Steyaert, G. Gielen, and W. Sansen, "A layout-aware synthesis methodology for RF circuits," in *Proc. IEEE International Conference on Computer-Aided Design (ICCAD)*, pp. 358-362, 2001.

[34] V. Aggarwal and U.-M. O'Reily, "COSMO: A Correlation Sensitive Mutation Operator for Multi-Objective Optimization," in *the 9th Annual Conference on Genetic and Evolutionary Computation, ACM*, pp. 741-748, 2007.

[35] K. Deb, A. Pratap, S. Agarwal, and T. A. M. T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation,* vol. 6, no. 2, pp. 182-197, 2002.

[36] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation,* vol. 18, no. 4, pp. 577-601, 2013.

[37] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation,* vol. 11, no. 6, pp. 712-731, 2007.

[38] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation,* vol. 20, no. 1, pp. 16-37, 2015.

[39] W. Lyu, P. Xue, F. Yang, C. Yan, Z. Hong, X. Zeng, and D. Zhou, "An efficient Bayesian optimization approach for automated optimization of analog circuits," *IEEE Transactions on Circuits and Systems I: Regular Paper,* vol. 65, no. 6, pp. 1954-1967, 2017.

[40] B. Paria, K. Kandasamy, and B. Póczos, "A flexible framework for multi-objective Bayesian optimization using random scalarizations," in *the 35th Uncertainty in Artificial Intelligence (UAI)*, PMLR, pp. 766-776, 2020.

[41] W. Lyu, F. Yang, C. Yan, D. Zhou, and X. Zeng, "Batch Bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design," in *International Conference on Machine Learning (ICML)*, pp. 3306-3314, 2018.

[42] K. Kandasamy, J. Schneider, and B. Póczos, "High dimensional Bayesian optimisation and bandits via additive models," in *International Conference on Machine Learning (ICML)*, pp. 295-304, 2015.

[43] R. A. Rutenbar, "Simulated annealing algorithms: An overview," *IEEE Circuits and Devices magazine,* vol. 5, no. 1, pp. 19-26, 1989.

[44] C. De Ranter, G. Van der Plas, M. Steyaert, G. Gielen, and W. Sansen, "CYCLONE: Automated design and layout of RF LC-oscillators," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 21, no. 10, pp. 1161-1170, 2002.

[45] A. Agarwal, H. Sampath, V. Yelamanchili, and R. Vemuri, "Fast and accurate parasitic capacitance models for layout-aware synthesis of analog circuits," in *Proc. the 41st Annual Design Automation Conference (DAC)*, pp. 145-150, 2004.

[46] X. Wang, S. McCracken, A. Dengi, K. Takinami, T. Tsukizawa, and Y. Miyahara, "A novel parasitic-aware synthesis and verification flow for RFIC design," in *Proc. European Microwave Conference*, pp. 664-667, 2006.

[47] M. Ranjan, W. Verhaegen, A. Agarwal, H. Sampath, R. Vemuri, and G. Gielen, "Fast, layout-inclusive analog circuit synthesis using pre-compiled parasitic-aware symbolic performance models," in *Proc. IEEE Design, Automation, and Test in Europe (DATE)*, vol. 1, pp. 604-609, 2004.

[48] A. Agarwal and R. Vemuri, "Layout-aware RF circuit synthesis driven by worst case parasitic corners," in *Proc. IEEE International Conference on Computer Design (ICCD)*, pp. 444-449, 2005.

[49] R. Schwencker, J. Eckmueller, H. Graeb, and K. Antreich, "Automating the sizing of analog CMOS circuits by consideration of structural constraints," in *Proc. IEEE Design, Automation, and Test in Europe (DATE)*, pp. 323-327, 1999.

[50] M. Dessouky, M. M. Louerat, and J. Porte, "Layout-oriented synthesis of high performance analog circuits," in *Proc. Design, Automation and Test in Europe (DATE)*, pp. 53-57, 2000.

[51] 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, no. 8, pp. 1089-1102, 2011.

[52] H. Graeb, S. Zizala, J. Eckmueller, and K. Antreich,, "The sizing rules method for analog integrated circuit design," in *Proc. IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pp. 343-349, 2001.

[53] R. Schwencker, F. Schenkel, H. Graeb, and K. Antreich, "The generalized boundary curve: a common method for automatic nominal design centering of analog circuits," in *Proc. Design, Automation, and Test in Europe (DATE)*, pp. 42–47, Mar. 2000.

[54] K. Antreich, J. Eckmueller, H. Graeb, M. Pronath, F. Schenkel, R. Schwencker, and S. Zizala, "Wicked: Analog circuit synthesis incorporating mismatch," in *Proc. IEEE Custom Integrated Circuits Conference (CICC)*, pp. 511–514., May 2000.

[55] R. Castro-Lopez, O. Guerra, E. Roca, and F. Fernandez, "An integrated layout-synthesis approach for analog ICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 27, no. 7, pp. 1179-1189, 2008.

[56] Cadence Design Systems, Inc., http://www.cadence.com/.

[57] T. Liao and L. Zhang, "Parasitic-aware GP-based many-objective sizing methodology for analog and RF integrated circuits," in *22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 475-480, 2017.

[58] P.-H. Lin, Y.-W. Chang, and S.-C. Lin, "Analog placement based on symmetry-island formulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 28, no. 6, pp. 791-804, 2009.

[59] G. Zhang, A. Dengi, R. Rohrer, R. Rutenbar, and L. Carley, "A synthesis flow toward fast parasitic closure for radio-frequency integrated circuits," in *Proc. the 41st Annual Design Automation Conference*, pp. 155-158, 2004.

[60] G. Shi, "A survey on binary decision diagram approaches to symbolic analysis of analog integrated circuits," *Analog Integrated Circuits and Signal Processing,* vol. 74, no. 2, pp. 331-343, 2013.

[61] M. Zhang, S. Zhao, and X. Wang, "Multi-objective evolutionary algorithm based on adaptive discrete differential evolution," in *IEEE Congress on Evolutionary Computation (CEC)*, pp. 614-621, 2009.

[62] A. W. Mohamed, "An improved differential evolution algorithm with triangular mutation for global numerical optimization," *Computers and Industrial Engineering,* vol. 85, pp. 359-375, 2015.

[63] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution – An updated survey," *Swarm and Evolutionary Computation,* vol. 27, pp. 1-30, 2016.

[64] S. M. Venske, R. A. Gonçalves, E. M. Benelli, and M. R. Delgado, "ADEMO/D: An adaptive differential evolution for protein structure prediction problem," *Expert System with Applications,* vol. 56, pp. 209-226, 2016.

[65] C. Chu and Y.-C. Wong, "FLUTE: fast lookup table based rectilinear steiner minimal tree algorithm for VLSI design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 27, no. 1, pp. 70-83, 2008.

[66] P. E. Allen and D. R. Holberg, CMOS Analog Circuit Design, Oxford University Press, 2002.

[67] E. Tlelo-Cuautle, I. Guerra-Gomez, et al., Evolutionary Algorithms in the Optimal Sizing of Analog Circuits, Intelligent Computational Optimization in Engineering, pp. 109-138, Berlin Heidelberg: Springer, 2011.

[68] Virtuoso Layout Suite, Cadence Design Systems, Inc., http://www.cadence.com/products/cic/layout_suite/pages/default.aspx.

[69] F. Silveira, D. Flandre, and P. Jespers., "A gm/ID-based methodology for the design of CMOS analog circuits and its application to the synthesis of a silicon-on-insulator micropower OTA," *IEEE Journal of Solid-State Circuits,* vol. 31, no. 9, pp. 1314-1319, 1996.

[70] D. M. Binkley, Tradeoffs and Optimization in Analog CMOS Design, Hoboken, New Jersey: Wiley, 2008.

[71] X. Dong and L. Zhang, "EA-based LDE-aware fast analog layout retargeting with device abstraction," *IEEE Transactions on Very Large Scale Integration Systems,* vol. 27, no. 4, pp. 854-863, 2019.

[72] R. Fiorelli, E. Peralías, and F. Silveira, An All-Inversion-Region gm/ID Based Design Methodology for Radiofrequency Blocks in CMOS Nanometer Technologies. In Wireless Radio-Frequency Standards and System Design: Advanced Techniques, pp. 15-39, IGI Global, 2012.

[73] L. Zhang, Y. Zhang, Y. Jiang and C. J. R. Shi, "Symmetry-aware placement with transitive closure graphs for analog layout design," *International Journal of Circuit Theory and Application,* vol. 38, no. 3, pp. 221-241, 2010.

[74] L. Zhang, R. Raut, L. Wang, and Y. Jiang, "Analog module placement realizing symmetry constraints based on a radiation decoder," in *IEEE International Midwest Symposium on Circuits and Systems*, pp. I481-I484, 2004.

[75] T. Liao and L. Zhang, "Parasitic-aware gm/ID-based many-objective analog/RF circuit sizing," in *19th International Symposium on Quality Electronic Design (ISQED)*, pp. 100-105, 2018.

[76] Calibre® xRC, Mentor Graphics, http://www.mentor.com/products/ic_nanometer_design/verification-signoff/circuit-verification/calibre-xrc/.

[77] K.-L. Yeh, C.-S. Chang, and J.-C. Guo, "Layout-dependent effects on high frequency performance and noise of sub-40nm multi-finger n-channel and p-channel MOSFETs," in *IEEE/MTT-S International Microwave Symposium Digest*, pp. 1-3, 2012.

[78] T. Liao and L. Zhang, "Efficient parasitic-aware hybrid sizing methodology for analog and RF integrated circuits," *Integration, the VLSI Journal,* vol. 62, pp. 301-313, 2018.

[79] T. Liao and L. Zhang, "Layout-dependent effects aware gm/ID-based many-objective sizing optimization for analog integrated circuits," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-5, 2018.

[80] R. He and L. Zhang, "Symmetry-aware TCG-based placement design under complex multi-group constraints for analog circuit layouts," in *Proc. 15th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Taipei, pp. 299-304, 2010.

[81] L. Zhang and U. Kleine, "A genetic approach to analog module placement with simulated annealing," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 345-348, 2002.

[82] Z. Wang, C. Gehring, P. Kohli, and S. Jegelka, "Batched large-scale Bayesian optimization in high-dimensional spaces," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 745-754, 2018.

[83] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani, "Predictive entropy search for efficient global optimization of black-box functions," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 918-926, 2014.

[84] C. K. Williams and C. E. Rasmussen, Gaussian Processes for Machine Learning, Vol. 2, no. 3, Cambridge, MA: MIT Press, 2006.

[85] D. M. Roy and Y. W. Teh, "The Mondrian process," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 1377-1384, 2008.

[86] H. E. Graeb, Analog Design Centering and Sizing, Berlin: Springer, vol. 64, 2007.

[87] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization,* vol. 11, no. 4, pp. 341-359, 1997.

Appendix A: User-defined Parameters

Each α_i = 1 and the summation of all β_i, i.e., $\sum_{i=1}^{m} \beta_i$ = 0.5 32

θ = 5 35

EA population size *NP*: 56 (large configuration) and 32 (small configuration in our setting) … 90

Each u_i = 1 and the summation of all v_i, i.e., $\sum_{v=1}^{K} \beta_v$ = 0.5 100

User-defined bounds: 0% and 90% of maximum performance variation based on reference ... 119

α_1 = 0.25 and β_1 = 0.75 127

α_2 = 0.25 and β_2 = 0.75 128

A_{min} = 10 156

User-defined set size: at least 10 (if PF has less than 10, randomly include other solutions) ... 157

Op-Amp: τ_1 = τ_2 = 0.5; comparator: τ_1 = τ_2 = 0.25 161

Appendix B: Published/Submitted Papers

1. International Journal Papers

[J1] T. Liao and L. Zhang, “High-dimensional many-objective Bayesian optimization for LDE-aware analog IC sizing,” *IEEE Transactions on Very Large Scale Integration Systems (TVLSI)*, submitted for review, 2021.

[J2] T. Liao and L. Zhang, “An LDE-aware g_m/I_D-based hybrid sizing method for analog integrated circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, DOI: 10.1109/TCAD.2020.3025068, 2020.

[J3] T. Liao and L. Zhang, “Efficient parasitic-aware g_m/I_D-based hybrid sizing methodology for analog and RF integrated circuits,” *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 26, no. 2, pp. 1-31, 2020.

[J4] T. Liao and L. Zhang, “Efficient parasitic-aware hybrid sizing methodology for analog and RF integrated circuits,” *Integration*, *the VLSI Journal*, vol. 62, pp. 301-313, 2018.

[J5] T. Liao and L. Zhang, “Analog integrated circuit sizing and layout dependent effects: A review,” *Journal of Microelectronics and Solid State Electronics*, vol. 3, no. 1A, pp. 17-29, 2014.

2. International/National Conference Papers

[C1] T. Liao and L. Zhang, “Layout-dependent effects aware g_m/I_D-based many-objective sizing optimization for analog integrated circuits,” in *Proc. IEEE International Symposium on Circuits & Systems* (*ISCAS*), pp. 1-5, 2018.

[C2] T. Liao and L. Zhang, “Parasitic-aware g_m/I_D-based many-objective analog/RF circuit sizing,” in *Proc. IEEE International Symposium on Quality Electronic Design* (*ISQED*), pp. 100-105, 2018.

[C3] Z. Zhao, T. Liao, and L. Zhang, “Fast performance evaluation for analog circuit synthesis frameworks,” in *Proc. IEEE International Symposium on Circuits & Systems* (*ISCAS*), pp. 1-5, 2018.

[C4] T. Liao and L. Zhang, “Parasitic-aware GP-based many-objective sizing methodology for analog and RF integrated circuits,” in *Proc. IEEE/ACM 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 475-480, 2017.

[C5] T. Liao and L. Zhang, “Differential evolution algorithm in analog integrated circuit sizing,” in *Proc. IEEE Newfoundland Electrical and Computer Engineering Conference (NECEC)*, Nov. 2014.

[C6] T. Liao and L. Zhang, “Analysis of layout dependent effects (LDE) and proposed optimization methodology,” in *Proc. IEEE Newfoundland Electrical and Computer Engineering Conference (NECEC)*, Nov. 2013.

[C7] T. Liao and L. Zhang, “MOSFET multi-finger layout structure in the advanced technology,” in *Proc. IEEE Newfoundland Electrical and Computer Engineering Conference (NECEC)*, Nov. 2012.