

TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Entwurfsautomatisierung

Constraint-Based Layout-Driven Sizing of Analog Circuits

Husni Habal

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Martin Buss (Univ. Tokio)

Prüfer der Dissertation: 1. Priv.-Doz. Dr.-Ing. Helmut Gräß

2. Prof. Dr. ir. Georges Gielen,
Katholieke Universiteit Leuven
Heverlee/Belgien

Die Dissertation wurde am 04.04.2012 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 05.02.2013 angenommen.

“Compassion is the basis of morality.”
-Arthur Schopenhauer

Acknowledgments

This work is the culmination of four years of research activity at the Institute for Electronic Design Automation at TU München. Firstly, I want to acknowledge Dr. Helmut Gräß for his guidance and supervision without which this dissertation would not be possible. I thank Prof. Ulf Schlichtmann for his confidence and for giving me the opportunity to pursue research work at his institute. I thank my colleagues Dr. Daniel Müller-Gritschneider, Dr. Martin Strasser, Dr. Michael Pehl, Anja Boos, and Michael Eick with whom I worked closely. My friend Patrick Birrer receives special gratitude for his help and support. Ultimately, I would like to thank my parents for their love and support throughout my life.

Contents

1	Introduction	1
1.1	Analog Integrated Circuit Design	2
1.1.1	Basic Analog Design Flow	2
1.1.2	Circuit Performances, Specifications, and Constraints	2
1.1.3	Process Parameters, Operating Conditions, and Reliability	4
1.1.4	Hierarchical Top-Down Design and Abstraction	4
1.1.5	Analog Circuit Design Automation	5
1.2	Motivation	6
1.2.1	Backtracking in the Analog Design Flow	6
1.2.2	Layout-Driven Circuit Sizing	7
1.3	State of the Art	9
1.4	Contributions of this Thesis	11
1.5	Related Publication	12
1.6	Organization of this Thesis	12
2	Formulation of the Circuit Sizing Problem	13
2.1	Basic Definitions	13
2.1.1	Electrical Circuit Topology	13
2.1.2	Electrical Test Bench Topology	14
2.1.3	Circuit Parameters	14
2.1.4	Circuit Performances	17
2.1.5	Circuit Sizing Rules	18
2.1.6	The Feasible Design Space and Performance Space	20
2.2	Circuit Problem Formulation	21
2.2.1	Feasibility Analysis	21
2.2.2	Circuit Sizing to Meet Performance Specifications	21
3	Overview of Layout Synthesis Steps	23
3.1	Introduction	23
3.2	Layout of Individual Physical Devices	24
3.2.1	Device Layout Automation	24
3.3	Device Placement	25
3.3.1	Circuit Placement Automation	26
3.4	Routing	27
3.4.1	Circuit Routing Automation	28
3.5	Post-Layout Electrical Model Extraction	28

4	New Automatic Constraint-Based Layout Synthesis Flow	31
4.1	Introduction	31
4.2	Enumeration of Device Layouts	32
4.2.1	Constrained Enumeration of CMOS Device Layouts	36
4.2.2	Constrained Enumeration of CMOS Devices in Common Centroid Layout	40
4.3	Enumeration of Circuit Placements	45
4.3.1	Placement Constraint Generation	45
4.3.2	Minimum Device Margins	46
4.3.3	Generation of Pareto-Optimal Placements	46
4.3.4	Geometric Placement Specifications	47
4.3.5	Ordering and Curtailing of Circuit Placements	51
4.4	Circuit Routing	52
4.4.1	Pin Assignment	53
4.4.2	Congestion Control	55
4.5	Post-Layout Satisfaction of Electrical Sizing Rules by Limiting Routing Resistance	57
4.5.1	Post-Layout Electrical Sizing Rules	57
4.5.2	Routing Limits to Satisfy Post-Layout Electrical Constraints	60
4.5.3	Maximization of R^u in the Feasible Effective Resistance Space	70
4.5.4	Acyclic Routing Network Graphs of Maximum Edge Number	72
4.5.5	Numerical Solution to (4.86) by Successive Linear Programming	76
4.6	Selection of a Final Layout	77
4.6.1	Post-Layout Circuit Extraction	78
4.6.2	Scalar Cost Metric Of Performance Specifications	78
4.7	Summary	80
5	Layout-Driven Circuit Sizing	81
5.1	Introduction	81
5.2	Review of the Search Algorithm Employed in Circuit Sizing	81
5.3	Technical Description of the Layout-Driven Circuit Sizing Problem	85
5.4	Issues in Numerical Function Evaluation	89
5.5	Geometric Inequality Constraint Functions	91
5.6	Electrical Performances and Constraints Without Layout Synthesis	91
5.6.1	Truncation Error	91
5.6.2	Computational Error	92
5.6.3	Adjustments to Palliate Truncation and Computational Error	93
5.7	Performances with Layout-Driven Circuit Sizing	95
5.7.1	Discretization Error	95
5.7.2	Placement Dependency	97
5.7.3	Solution Selection in the Design Space Under Consideration of Discretization and Placement Error	99
5.7.4	Partial Derivative Calculation Under Consideration of Discretization and Placement Error	102

5.8	On the Cost of Circuit Sizing	107
5.9	Summary	108
6	Circuit Sizing Examples	111
6.1	Description of the Example Circuits	111
6.1.1	Folded Cascode Operational Amplifier (FC-OA)	111
6.1.2	Tunable Operational Transconductance Amplifier (TOTA) . . .	114
6.1.3	Miller Operational Amplifier (MOA)	119
6.2	Experimental Setup	123
6.2.1	Computer Hardware and Software	123
6.2.2	Rules to Extract Layout Netlists	124
6.2.3	Selection of the Starting Vector for Circuit Sizing	124
6.3	Circuit Sizing Results and Comparison	125
6.3.1	Folded Cascode Operational Amplifier (FC-OA)	125
6.3.2	Tunable Operational Transconductance Amplifier (TOTA) . . .	130
6.3.3	Miller Operational Amplifier (MOA)	134
6.4	Summary	139
7	Conclusion	141
A	Area Estimation Without Layout Synthesis	143
B	Approximation to the Gradient of the Area Estimate	147
	Bibliography	149
	Nomenclature	163
	Lists	165
	List of Figures	165
	List of Tables	167
	Abstract in German	169

Chapter 1

Introduction

The market for commodity integrated circuit (IC) solutions is dominated by complementary metal-oxide-semiconductor (CMOS) fabrication technology. This is due to the low static power, high device density, and cheap manufacturing cost of CMOS chips. Analog circuits, such as analog-to-digital (A/D) converters, radio frequency (RF) front end interfaces, and frequency synthesizers are often implemented as components of a mixed-signal CMOS IC that is dominated by a large digital core, such as a microcontroller or digital signal processor (DSP) [KCJ⁺00]. The evolution of CMOS mixed-signal fabrication technology is focused on improving the specifications of the digital core, including higher gate and memory densities, lower power consumption, and a longer mean time to failure (MTTF). In order to improve the latter specifications, CMOS devices have been scaled down to deep sub-micron dimensions and are designed to operate at a low supply voltage [ANvLT05].

This course of technology progression has imposed many challenges on the analog designer. The designer must account for complex nonlinear device models, low threshold voltages, large process parameter variations, channel length modulation caused by short device length, and gate leakage when designing a circuit to meet a set of performance specifications, such as minimum gain and maximum power.

To aid the analog designer, research in analog electrical design automation (EDA) has focused on two tasks. The first task is to add more layers of hierarchy and abstraction in the design flow, while the second is to find means of automation in each design step, such as the dimensioning of components and layout synthesis.

High level programming languages and modeling tools are often used at the first stage of analog design, as they are fast and easy to set up [Mata]. Commercial toolboxes are available for some applications [Matb]. At a lower level, a hardware description language (HDL), such as Verilog-A [VLR] or VHDL-AMS [DV03], is used to create behavioral models of analog circuits. Tools for the automatic dimensioning of circuit components are available [AEG⁺00a, Cad03b], as are tools for automatic analog placement and routing [SEG⁺08, Cad03a]. These tools, however, still lag behind their digital counterparts – offering many opportunities for original research.

1.1 Analog Integrated Circuit Design

1.1.1 Basic Analog Design Flow

The design flow of an analog circuit at the device (transistor) level of detail consists of four standard and consecutive steps:

1. *Circuit topology selection*: A circuit topology, also referred to as a circuit structure, is a network of electrical devices; each device has at least two terminals and a behavioral model for analytical or numerical simulation. A circuit structure is selected that has the potential to fulfill the specified functional purpose of the circuit.
2. *Circuit sizing*: Constituent in each behavioral device model is one or more device design parameters, for example, in the CMOS device model, the drain to source current is a function of channel width and length. During circuit sizing values are assigned to the design parameters of each device in the circuit topology. Design parameter values are selected so that the circuit operates with the desired functionality.
3. *Circuit layout synthesis*: A circuit layout is the blueprint of planar geometric shapes used to create photo masks for the physical realization of the circuit in a specific fabrication technology using the technique of photo lithography. During layout synthesis, the geometric shapes representing sized topology devices are drawn on an IC floor plan. The network connections between devices are also drawn according to the circuit topology.
4. *Post-layout circuit extraction and electrical verification*: A new network of electrical devices is generated based on the circuit layout. This new network is a closer approximation to the physical circuit than the sized circuit topology. It is used to verify that the circuit still has the desired functionality after layout synthesis.

1.1.2 Circuit Performances, Specifications, and Constraints

A circuit performance is a descriptive quantity of circuit behavior deemed of value by the analog designer. It is useful to divide circuit performances into geometric and electrical categories. Electrical performances are selected based on the intended function of the analog circuit, for example, operational amplifier, low noise amplifier, or mixer. Geometric performances describe the spacial properties of the circuit, such as the area and aspect ratio of the circuit after layout.

A circuit specification is a functional equation or inequality of circuit performances. When the circuit specification is true, the circuit is said to exhibit the proper behavior. In practice most specifications take the form of an upper or lower bound on the value of a circuit performance.

Circuit specifications alone may not be sufficient to describe the behavior of an analog circuit [MCR00, MGS08]. Additional designer knowledge about the circuit topology can be translated into functional equations or inequalities of the circuit topology node voltages and branch currents, hereafter called electrical constraints, and functional equations or inequalities of the design parameters, hereafter called geometric constraints.

The circuit topology is electrically controllable through a subset of its network nodes, which are defined as *external nodes*. In order to calculate the electrical performances, specifications, and constraints, one or more test bench circuits are constructed by the analog designer and connected to the external circuit nodes. Test bench circuits are electrical networks that establish the electrical operating conditions under which the circuit is expected to operate. These include the voltage and current stimulus, the correct loading at each external node, and the correct external feedback paths between the external nodes.

Electrical performances and specifications are typically calculated as an expression or sequence of expressions from the voltages at and currents through the external topology nodes. The test bench setup for the measurement of electrical performances and specifications is normally independent of the circuit topology and depends only on the electrical signals at the external circuit nodes. In contrast, electrical constraints are often calculated from internal topology node voltages and branch currents. They are topology dependent and must be redefined for a change in the topology.

The circuit and test bench form a mathematical model of a closed electrical system. A numerical circuit simulator is used to study the behavior of this system, such as SPICE [Nag75], Spectre [Kun95a], or Titan [Inf08]. Numerical simulation requires detailed mathematical device models such as BSIM [SSKJ87] and EKV [EKV95].

The type of analysis method that is used in simulation is dependent on the type of stimulus sources present in the test bench network and the type of response that is to be observed. The analysis methods typically used for analog circuits include the following:

- DC analysis, or circuit quiescent (bias) point calculation.
- AC analysis, or linear small signal frequency domain circuit analysis.
- Transient simulation, or the time domain large-signal solution of differential algebraic circuit equations.
- Harmonic balance – to calculate the steady-state response of an electrical circuit without the need for a transient simulation.
- Periodic steady state (PSS) simulation and Periodic small-signal analysis [YP02]. PSS directly computes the periodic steady-state response of a circuit without transient analysis.

1.1.3 Process Parameters, Operating Conditions, and Reliability

In addition to design parameters, several process parameters constitute terms in a device model. As the name implies, the value of the process parameters depends on the semiconductor fabrication technology used to realize the IC [LJX⁺]. Due to the imperfections of sub-wavelength lithography, random dopant fluctuations, and line edge roughness during the IC fabrication procedure, process parameters are often not constant in value after manufacturing the IC. Process parameters values may change systematically or stochastically between silicon wafers, between dies on the same wafer, or between devices on a single die [AN07, BMR07, LBSG07, LDH⁺08]. The general trend in CMOS semiconductor fabrication technology is that the coefficient of variation of the process parameters increases with every new technology generation and reduction in device scale [tec09]. Variable process parameters increase the complexity of analog circuit design, as appropriate layout techniques, production yield levels, and margins of error in specifications and constraints must be considered in the design flow [Has01, AEG⁺00a, CLW10, GMGS09, YL08].

Circuit behavior is also dependent upon the operational conditions external to the circuit topology. Circuit stimulus, loading, and feedback conditions, as imposed by the test bench circuit, influence the circuit through electrical signals at the external circuit nodes. For example, the value of the DC supply voltage and the load impedance are set in the test bench circuit. In addition, environmental parameters, such as temperature, are normally considered within device models. These operating conditions are typically represented by a set of operational parameters; their value is typically not constant, but fluctuates over a range that needs to be taken into account during circuit design.

Reliability is defined as the ability of a circuit to conform to its specifications over a specified period of time under specific conditions [GDWM⁺08]. The effects of electromigration (EM), time-dependent-dielectric-breakdown (TDDB) and hot carrier degradation (HCI and NBTI) significantly affect circuit reliability in deep sub-micron CMOS fabrication technologies and must be taken into consideration during analog circuit design [WVN⁺06].

1.1.4 Hierarchical Top-Down Design and Abstraction

A large analog system, such as a frequency synthesizer or RF front end, may be comprised of many thousands of devices. For complete architectures, such as a WLAN physical layer, there may be digital and software components that are integral to the system and that must be designed in tandem with the analog sections. Constructing test benches, selecting appropriate constraints and specifications, then synthesizing and verifying such systems is intractable using only the basic analog design flow.

Large systems are therefore partitioned into sub-blocks by identifying the sub-tasks performed by the system. If the design of a sub-block is still infeasible, then further

partitions may be necessary until the basic analog design flow can be applied. The result of system partitioning is a hierarchy tree of circuit blocks. At each level in the hierarchy, constraints, performances, and specifications need to be identified according to the function of each block and its relation to the block above it in hierarchy. Abstraction of functionality can be employed to create a simplified behavioral model of a block in the hierarchy tree before it is designed at the detailed device level and to reduce the time needed to simulate large blocks [RGR07].

1.1.5 Analog Circuit Design Automation

Complete or partial automation techniques are available for each step of the basic analog design flow.

Two broad approaches are used to automate topology selection. In the first approach, designer knowledge or an iterative search algorithm is used to create a circuit of small functional blocks. Each functional block performs an elementary analog operation, for example a current mirror. Multiple structures for each functional block are predefined and saved in a library along with the constraints necessary to ensure correct behavior. The structure with the greatest potential to fulfill the purpose of the complete circuit is then chosen for each functional block. circuit sizing is often combined with topology selection in this approach [HRC89, ETP89, DCR05]. The second widely used method of topology selection is topology generation from basic devices, for example CMOS transistors, using graph grammar rewriting [DV09], or signal flow graphs [GE95].

For circuit sizing automation, a set of design parameters that satisfy the circuit constraints and specifications is sought out using a numerical optimization algorithm. Design parameter values are systematically selected by the optimization algorithm and the corresponding value of the constraints and performances are evaluated by numerical circuit simulation. The optimization algorithm terminates when a set of design parameters is found that evaluates all constraints and specifications to true. In this case the circuit is designated as feasible. Circuit sizing automation using a numerical optimization algorithm is illustrated in Figure 1.1.

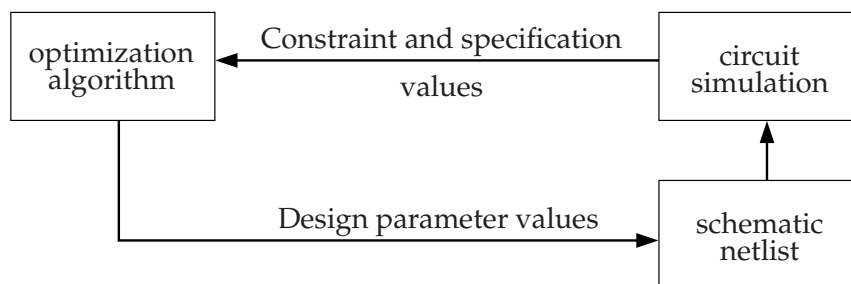


Figure 1.1: Sizing by an amalgamation of numerical simulation and optimization.

Many optimization algorithms have been used in analog circuit sizing, the most popular categorization of algorithms is to divide them into deterministic [LD81, NRSVT88, AEG⁺00a, GH10, Soo08, EDGS03, AEG⁺00b, LGXP04] and stochastic algorithms [Kes95, GWS90, SCP07, ORC96, MFDCRV94, ABD03, PKR⁺00]. Since the numerical simulation of analog circuit incurs a high computational cost, the number of constraint and performance evaluations needed to terminate the algorithm is an important measure of algorithm fitness.

Circuit layout synthesis is comprised of the device placement and routing operations. There are many design heuristics and constraints that must be fulfilled during the placement and routing of devices, such as device orientation, proximity, and symmetry conditions [Has01]. Several algorithms to automate placement and routing are available in literature [ESGS10, XY09, WCC03, PCLX01, RM08, HRM08, Cad03a].

Geometric and electrical verification ensure that the post-layout circuit fulfills the technology layout rules and that the circuit will operate correctly after layout synthesis. Mature commercial tools to extract a circuit model from a layout and to perform verification are readily available [Cad05].

1.2 Motivation

1.2.1 Backtracking in the Analog Design Flow

As stated in Section 1.1.1, the basic analog design flow consists of four steps that are completed consecutively. It may be necessary to backtrack one or more steps up the design flow if progress cannot be made towards completion, as shown in Figure 1.2.

Backtracking is costly, the problem blocking progress must be identified and a remedy determined; multiple iterations through the design flow may be necessary before successful completion.

If no combination of circuit topology and design parameter values exists to fulfill the specifications, then a redesign of the system at a higher level must be performed. This is illustrated by backtracking paths (1) and (2) in Figure 1.2. One remedy is to pursue a bottom-up design approach whereby all attainable performance values are ascertained before high system level design is begun.

A failure detected during the layout synthesis or electrical verification steps typically means circuit sizing and layout creation must be repeated. This is illustrated by backtracking paths (3) and (4) in Figure 1.2. A remedy is to consider or estimate the effects of layout synthesis during circuit sizing. This is the principle objective of this dissertation, and is expanded upon in Section 1.2.2.

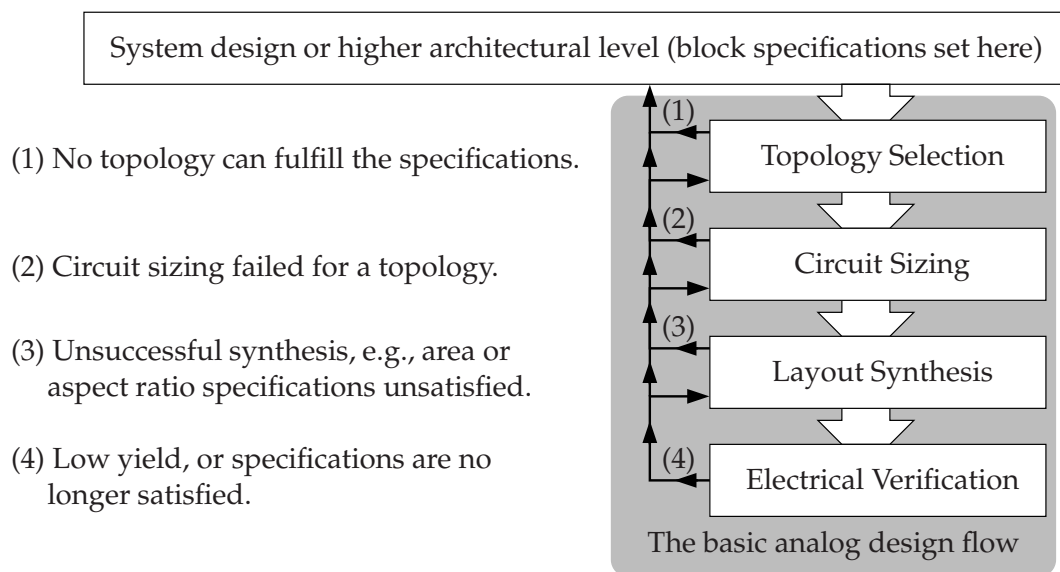


Figure 1.2: Backtracking and reiteration through the design flow may be necessary.

1.2.2 Layout-Driven Circuit Sizing

Layout synthesis may have a critical effect on circuit behavior:

- Layout-induced parasitic components, such as routing resistance and coupling capacitance, affect electrical performance.
- Systematic and intra-die random process parameters that depend on device placement, such as the distance between symmetric devices in Pelgrom's law [LBSG07], process gradients, and anisotropic effects, affect electrical performance and yield numbers.

Circuit performance values may change significantly after layout synthesis due to these two items, consequently a specification may become unsatisfied.

In top-down design, geometric specifications, such as the maximum area and aspect ratio of a circuit block, may be set at the system level. The location of pin connections on the boundary of the layout silhouette might also be fixed during chip floorplanning prior to circuit block design [KWY96]. If the geometric specifications cannot be satisfied during the layout synthesis step, then device dimensions, such as CMOS transistor widths, must be reduced and circuit sizing repeated.

Several remedies can be applied to mitigate these flaws:

- Design heuristics are applied during layout synthesis to help match the electrical performance of the circuit before and after layout synthesis [Has01]; for example, the use of common centroid device placement and symmetric signal path routing to improve matching and increase common mode signal rejection in differential signal paths.

- Circuit area is estimated from device dimensions, such as CMOS transistor width and length, before layout synthesis; maximum area is then set as a specification during circuit sizing.
- Performance specifications are tightened by an extra margin to account for the effects of layout synthesis.

It may not be feasible to negate the complete effects of layout synthesis or to estimate area with enough accuracy before actual layout.

For example, Table 1.1 states the specifications and lists the simulated performance values of a CMOS operational amplifier after both circuit sizing and layout synthesis. The value of some performances, such as common mode rejection ratio (CMRR), power supply rejection ratio (PSRR), and total harmonic distortion (THD), change significantly. The PSRR and THD specifications are unsatisfied after layout synthesis. Area estimation is too pessimistic – a more favorable tradeoff can probably be found between performances in the feasible performance space, and the result of circuit sizing is sub-par after layout synthesis.

Table 1.1: Performances and specifications of a CMOS operational amplifier

Specification	Unit	After Circuit Sizing	After Layout Synthesis
Gain ≥ 80	dB	83	83
CMRR ≥ 100	dB	114	111
PSRR ≥ 90	dB	90	86
Power ≤ 0.50	mW	0.41	0.42
THD ≤ 0.100	%	0.091	0.104
Area ≤ 3500	μm^2	3495*	3229

*Estimated layout area used during circuit sizing.

For problematic circuit problems where mitigation methods are unsatisfactory, layout synthesis can be integrated into the circuit sizing step to create a so-called layout-driven solution to the circuit sizing problem as illustrated in Figure 1.3. The result of layout-driven sizing is a layout that meets the circuit specifications and constraints.

Several layout-driven circuit sizing methods, as well as placement and routing algorithms, can be found in literature; they are reviewed in the state of the art section.

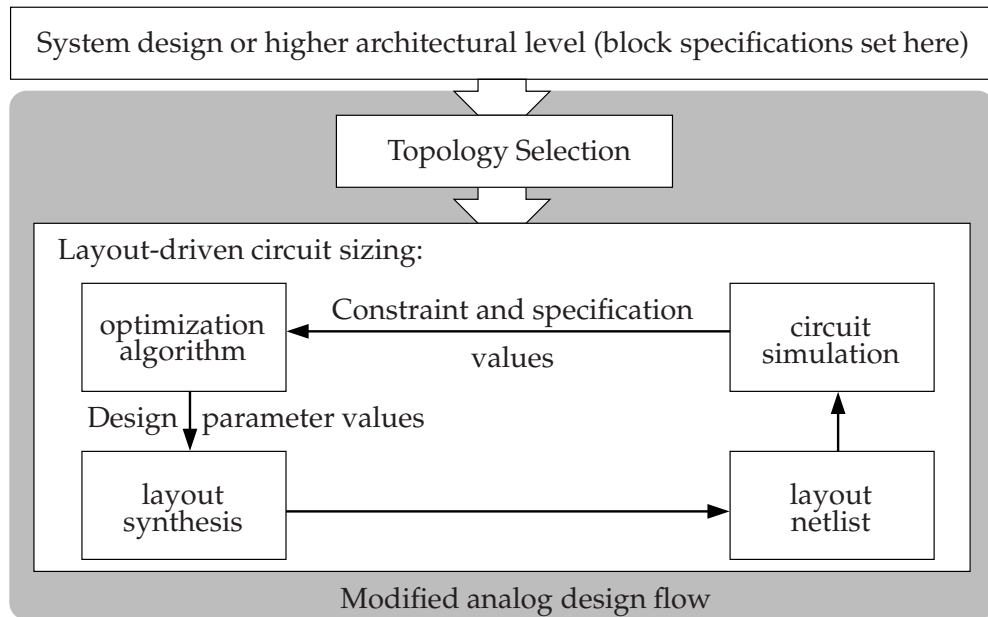


Figure 1.3: Analog design flow with layout-driven circuit sizing.

1.3 State of the Art

The state of the art in layout-driven (or layout-aware) circuit sizing can be divided into template-based and non-template-based methods.

As the name implies, template based methods rely on the use of layout templates [CLGRF08, BJS05, JZB⁺06]. A template specifies the spatial relation between circuit devices, such as transistors and capacitors, as well as fixed interconnect paths for routing. The template is created for each circuit topology prior to circuit sizing.

Template methods can be roughly categorized according to the data structure used to store the spatial relation between devices and the algorithm used for automatic circuit sizing. Used data structures include the slicing tree, O-tree, and B*-tree. Global optimization algorithms were used in the state of the art methods, including evolutionary algorithms and simulated annealing.

In [CLGRF08] a template defined by a slicing tree is used to estimate circuit area and layout parasitics. Interconnect parasitic estimates are stored in a lookup table associated with the template, while analytical-geometric techniques are used to extract the parasitics of placed devices. A simulated annealing algorithm is used for circuit sizing, requiring several thousand iterations for convergence in the given circuit examples.

Other methods, such as [HJBRS05, LZ10], are aimed at process migration or performance retargeting. An existing circuit layout is used as a template, device dimensions

are modified, after which interconnects are shrunk or extended to meet the layout and electrical design rules of the process technology.

For process migration, the technology layout rules may change prohibiting a direct downscaling of a template. As an example, in some 65 and 45nm technologies, transistor gates must be aligned on a grid while all gates must share the same orientation. It may also be difficult to avoid new routing conflicts or an increase in routing congestion when downscaling.

For performance retargeting, the aspect ratio of circuit devices may become extreme if the device parameters, such as CMOS width, change by a significant amount. This was solved in the references by the addition of geometric constraints on template devices. However, due to the fixed spatial relation between template devices, these geometric constraints must be severe, this will decrease the size of the optimization search space.

The non-template-based layout-driven sizing methods rely on simplifying approximations for performance evaluation, layout construction, and the modeling of layout parasitic devices in order to perform expeditious circuit sizing.

In [PV09] a linear regression model of the performances is used. The design space is sampled and a layout netlist is generated for each sample to define model parameters. The Pareto tradeoff [Par06] between performances is then explored using a multi-objective simulated annealing algorithm. Only layout parasitic devices are roughly approximated, while geometric constraints and matching are not considered.

In [YD09] performance sensitivity to node capacitance and device mismatch is used to direct placement using an algorithm based on slicing trees. Different shapes are considered for each device. A custom fast circuit simulator is used, however only DC and AC performance sensitivities can be calculated.

Several constraint-driven placement and routing algorithms can be found in [XY09, WCC03, PCLX01, SEG⁺08] and [RM08, HRM08] respectively. In [ESGS10], the circuit graph is subdivided into hierarchical proximity and symmetry groups and placement constraints are automatically generated. The tool of [SEG⁺08] was then used for the placement generation of several example circuits. Although not complete layout-driven circuit sizing solutions, these algorithms automate key layout synthesis steps.

1.4 Contributions of this Thesis

A design flow is presented for automatic layout synthesis starting with a topology and a set of circuit design parameter values. The flow is driven by geometric design, placement, and routing constraints and is not a template-based method. The new flow is integrated with the deterministic nonlinear optimization algorithm in [SSGA00] to perform layout-driven circuit sizing.

The novelty, in comparison to the state of the art, is summarized in six items:

- A deterministic optimization algorithm is used. In contrast to stochastic global search algorithms, such as evolutionary algorithms and simulated annealing, the deterministic algorithm has local scope, but converges to a solution within a small number of iterations, moreover, it requires a small number of performance evaluations. Less than 250 performance evaluations were needed for the most complicated circuit example. Theoretically, Q-superlinear convergence is possible with a smooth objective function.
- In the state of the art, simplifications are made to expedite performance evaluation. Knowledge-based equations, regression models, or a custom numerical simulator is used that is limited to DC and AC analysis. This is necessary as the used stochastic search algorithms demand thousands of performance evaluations. No simplifications are made in the proposed method; any numerical simulator can be used.
- The closest competitor in literature pursuing a method that is not template-based generates layout placements using a slicing tree algorithm. In the new method, a placement algorithm based on B*-trees is used [SEG⁺08]. It is known that a wider range of placement arrangements can be explored using B*-trees than slicing or O-tree algorithms [WCC03].
- Layout parasitics are extracted by an integral equation field solver with an permissible error of 3%. No analytical-geometric models are used to expedite parasitic estimation.
- DC electrical constraints are employed during layout synthesis to ensure correct circuit function and robustness. It has been shown in [MCR00, MGS08] that geometric and electrical circuit sizing rules are important for circuit function and robustness. Whilst almost all layout-driven methods in the state of the art implement the geometric constraints during layout synthesis, none check that the electrical constraints also remain satisfied. Parasitic resistance, however, can have an effect on the DC bias point of the circuit. In this thesis, the DC electrical constraints are ensured during routing by dynamically setting the upper bound on the allowed resistance of each routing path, by solving an optimization subproblem.
- The effect of routing congestion is considered in the new method. Layouts are adjusted to eliminate congestion.

The bulk of the state of the art methods and the ones that consider the most details in layout synthesis are template-based methods. In addition to the six items above, the presented method is distinguished from template-based methods in some categories:

- For each device in the circuit, a multi-valued mapping between device design parameters (such as CMOS transistor width and length) and possible device layouts is performed. Only layouts that satisfy certain geometric constraints and minimize discretization error due to manufacturing grid alignment are considered as possibilities for placement.
- For devices in a common centroid placement configuration, the number of divisions and the interleave pattern is selected during layout synthesis for an optimal layout. Traditionally, the number of divisions is fixed at the schematic level.
- In template-based methods, devices have a set location in the layout template that is fixed by a single slicing tree, O-tree, or B*-tree. During synthesis, only placements that conform to the fixed tree can be considered. In contrast, every possible B*-tree is considered by the new method.

1.5 Related Publication

Parts of the research work completed in this dissertation have been published in [HG11]. The principle steps of automatic constraint-based layout synthesis were described, as was the integration with a deterministic circuit sizing algorithm. The new layout-driven circuit sizing algorithm was demonstrated on two circuit examples, an operational amplifier and a tunable operational transconductance amplifier.

1.6 Organization of this Thesis

The remainder of this dissertation is organized as follows. In Chapter 2, mathematical definitions are given for circuit parameters, performances, and sizing rules. This is followed by a formulation of the circuit sizing problem. In Chapter 3, the basic steps of layout synthesis are detailed. Techniques to extract an electrical model from a geometric layout are also reviewed. In Chapter 4, the new automatic constraint-driven layout synthesis flow is presented. In Chapter 5, the new layout-driven circuit sizing procedure is presented. The issues resulting from numerical function evaluation and layout synthesis are described, as are techniques to handle these issues for successful sizing. Chapter 6 details the circuit sizing process for three circuit examples. The results of layout-driven sizing are compared to those of traditional circuit sizing without integrated layout synthesis. Chapter 7 concludes this dissertation.

Chapter 2

Formulation of the Circuit Sizing Problem

This chapter starts with mathematical definitions for circuit parameters, performances, and sizing rules. This is followed by a description of the circuit sizing problem for feasibility and for the fulfillment of performance specifications.

2.1 Basic Definitions

2.1.1 Electrical Circuit Topology

An electrical circuit topology, \mathcal{T} , is an interconnection of electrical devices, in which each device has two or more terminals. The topology can be represented by a hypergraph $HG(\mathcal{V}, \mathcal{E})$, where the vertices, \mathcal{V} , are the interconnects (circuit nodes) and the hyperedges, \mathcal{E} , are the devices [EGB06]:

$$\mathcal{T} \longrightarrow HG(\mathcal{V}, \mathcal{E}) \quad (2.1)$$

Each device $\delta \in \mathcal{E}$ is associated with a 3-tuple consisting of the device name, device type, and a list of device-terminal to vertex connections:

$$\delta = \begin{bmatrix} \text{"name"} \\ \text{"type"} \\ \text{"connections"} \end{bmatrix} \begin{array}{l} \text{e.g., MN1, MN2, MP1, C1, R1, ...} \\ \text{e.g., NMOS, PMOS, polysilicon-capacitor, ...} \\ \text{e.g., } [v_3, v_1, v_9] \text{ with } \{v_1, v_3, v_9\} \in \mathcal{V} \end{array} \quad (2.2)$$

The possible device types depend on the used technology. Each device type is associated with an electrical model for simulation and geometric rules for layout synthesis.

The circuit is electrically controllable through a subset of its vertices, $\mathcal{V}_e \subseteq \mathcal{V}$, defined as external circuit nodes.

2.1.2 Electrical Test Bench Topology

A test bench topology, TB , is an interconnection of electrical devices. It is designed to be connected to the external nodes, \mathcal{V}_e , of a circuit topology, \mathcal{T} , and establish the operating conditions under which electrical behavior is studied. This includes bias, stimulus, and external load and feedback conditions. Open paths in the hypergraph of \mathcal{T} must also be closed by connecting to a test bench.

As with the circuit topology, the test bench can be represented by a second hypergraph with vertices \mathcal{VB} and devices (edges) \mathcal{EB} :

$$TB \longrightarrow HG(\mathcal{VB}, \mathcal{EB}); \mathcal{V}_e = \mathcal{V} \cap \mathcal{VB} \quad (2.3)$$

2.1.3 Circuit Parameters

The circuit and test bench depend on a number of parameters that control how device models will behave during numerical simulation. These parameters can be classified into three separate categories:

Design Parameters

These are the parameters that can be freely adjusted by the circuit designer. A further distinction can be made between design parameters attached to circuit devices, \mathcal{E} , such as the width and length of a CMOS transistor, and the capacitance of a polysilicon capacitor; and design parameters attached to test bench devices, \mathcal{EB} , typically the DC voltage or current of power source used to bias the circuit.

As well as electrical behavior, design parameters attached to circuit devices will affect the geometric attributes of the circuit, such as layout area.

Let \mathbf{d}_δ denote the design parameters of a device $\delta \in \mathcal{E} \cup \mathcal{EB}$, \mathcal{D}_δ denote the associated domain. For example, if δ is CMOS device, then $\mathbf{d}_\delta = \mathbf{d}_{\text{CMOS}}$ and $\mathcal{D}_\delta = \mathcal{D}_{\text{CMOS}}$ as given in Table 2.1.

Table 2.1: CMOS device design parameters

$\mathbf{d}_{\text{CMOS}} \in \mathcal{D}_{\text{CMOS}}, \mathcal{D}_{\text{CMOS}} = \mathcal{D}_W \times \mathcal{D}_L$			
i	description	$\mathbf{d}_{\text{CMOS}}[i]$	Domain
1	total width	W	$\mathcal{D}_W = [W_{\min}, W_{\max}]$
2	total length	L	$\mathcal{D}_L = [L_{\min}, L_{\max}]$

The CMOS design parameters are the transistor width and length. The domain of each parameter is a bounded interval of real numbers. A bound may denote a technology constraint or a designer preference.

The *circuit design parameters* are ordered as vector $\mathbf{d}_\mathcal{E}$, with $n_{d\mathcal{E}} = |\mathbf{d}_\mathcal{E}|$:

$$\mathcal{E} = \{\delta_1, \delta_2, \dots\} \Rightarrow \begin{cases} \mathbf{d}_\mathcal{E} = [\mathbf{d}_{\delta_1}^T; \mathbf{d}_{\delta_2}^T; \dots]^T & \text{(circuit design parameters)} \\ \mathcal{D}_\mathcal{E} = \mathcal{D}_{\delta_1} \times \mathcal{D}_{\delta_2} \times \dots & \text{(associated design space)} \end{cases} \quad (2.4)$$

The *test bench design parameters* are ordered as vector $\mathbf{d}_{\mathcal{EB}}$, with $n_{d\mathcal{EB}} = |\mathbf{d}_{\mathcal{EB}}|$:

$$\mathcal{EB} = \{\delta_1, \delta_2, \dots\} \Rightarrow \begin{cases} \mathbf{d}_{\mathcal{EB}} = [\mathbf{d}_{\delta_1}^T; \mathbf{d}_{\delta_2}^T; \dots]^T & \text{(test bench design parameters)} \\ \mathcal{D}_{\mathcal{EB}} = \mathcal{D}_{\delta_1} \times \mathcal{D}_{\delta_2} \times \dots & \text{(associated design space)} \end{cases} \quad (2.5)$$

The *design parameters* are combined in vector \mathbf{d} , with $n_d = |\mathbf{d}|$; the complete design space is denoted by \mathcal{D} and is assumed to be a bounded subset of an Euclidean space:

$$\mathbf{d} \in \mathcal{D} \quad \text{such that} \quad \mathbf{d} = \begin{bmatrix} \mathbf{d}_\mathcal{E} \\ \mathbf{d}_{\mathcal{EB}} \end{bmatrix}, \quad \mathcal{D} = \mathcal{D}_\mathcal{E} \times \mathcal{D}_{\mathcal{EB}}, \quad \mathcal{D} \subset \mathbb{R}^{n_d} \quad (2.6)$$

It is necessary to normalize the design parameters, so that design parameters with different units and with widely different design space bounds are comparable. Normalization is also necessary to avoid ill-conditioned transformations during numerical analysis [TB97]. In general, normalization can be accomplished by a bijective linear transformation, and is represented here by a normalization matrix \mathbf{N} :

$$\mathbf{d}_{\text{normalized}} = \mathbf{N} \cdot \mathbf{d}_{\text{original}}; \quad \mathcal{D}_{\text{original}} \xrightarrow{\mathbf{N}} \mathcal{D}_{\text{normalized}} \quad (2.7)$$

$$\mathbf{d}_{\text{original}} = \mathbf{N}^{-1} \cdot \mathbf{d}_{\text{normalized}}; \quad \mathcal{D}_{\text{normalized}} \xrightarrow{\mathbf{N}^{-1}} \mathcal{D}_{\text{original}} \quad (2.8)$$

Unless explicitly mentioned, it will be assumed in subsequent analysis and discussion that the design parameters are suitably normalized.

Process Parameters

Process parameters denote properties of the semiconductor fabrication technology as represented in device electrical models. For example, the BSIM3 model for CMOS devices [LJX⁺] has 16 important process parameters [PDML94, MI92]. It is worth noting that the effect of the fabrication process on geometric properties, such as the effective channel width and length of CMOS devices, is normally considered in the electrical models with suitable relations and process parameters.

Due to manufacturing imperfections, the value of some process parameters may vary between fabricated circuits. If variability is large enough to have a measurable effect on electrical circuit behavior, then it must be accounted for during circuit design.

Process parameters can have components that vary systematically, such as across-field and layout dependent variation terms [AN07], as well as statistical components that are values of a random variable. A statistical component is global if it has the same value of random variable for all devices on the same die. A statistical component is local if the value can be different for each device on the same die. Global components can be represented by a single random variable, while local component values must be picked individually for each device.

In [PDML94], global variance and correlation is estimated for 16 CMOS process parameters in the BSIM model, while in [LHC86, PDW89] the mismatch in CMOS threshold voltage, current factor, and drain current due to the local variation of process parameters is studied. In [MI92], a stochastic model for the value of process parameters is developed that includes local variation components. Local variance and correlation was estimated for 16 CMOS process parameters in the BSIM model. Another model of local variation that takes into account the distance between the barycenter of devices according to Pelgrom's law is given in [LBSG07].

Examples of process parameters with a global statistical component in the BSIM3 model are gate oxide thickness (T_{ox}), channel doping concentration (N_{ch}), and drain-source sheet resistance (R_{sh}). Process parameters with a large enough local variation component to cause a mismatch in electrical properties, such as drain current, include mobility at nominal temperature (μ_0) and the nominal threshold voltage (V_{th0}).

It is possible to transform the random variables of an arbitrary probability density function (PDF) into random variables of a Gaussian distribution [Esh92]. This allows the global and local statistical component values to be selected from a Gaussian distribution.

Let \mathbf{s} , be the vector of transformed statistical component values of the complete circuit with $n_{xs} = |\mathbf{s}|$. The joint Gaussian PDF of \mathbf{s} is $\text{pdf}_N(\mathbf{s})$:

$$\mathbf{s} \in \mathbb{R}^{n_{xs}}; \mathbf{s} \sim \text{pdf}_N(\mathbf{s}); \text{pdf}_N(\mathbf{s}) = \frac{1}{\sqrt{2\pi}^{n_{xs}} \sqrt{\det(\mathbf{C})}} \cdot \exp\left(-\frac{\beta^2(\mathbf{s})}{2}\right) \quad (2.9)$$

$$\beta^2(\mathbf{s}) = (\mathbf{s} - \mathbf{s}^0)^T \cdot \mathbf{C}^{-1} \cdot (\mathbf{s} - \mathbf{s}^0) \quad (2.10)$$

where \mathbf{s}^0 is the mean value of the Gaussian PDF and \mathbf{C} is the covariance matrix.

When modeling nominal circuit behavior only, the value of \mathbf{s} is fixed to \mathbf{s}^0 .

Operating Parameters

These are test bench and environment parameters that depend on the operating conditions and cannot be adjusted freely.

Environmental operating parameters, such as temperature, are set in the simulation environment – to be used directly in device models. Test bench operating parameters, such as the supply voltage or a load capacitance, are attached to test bench devices.

The test bench and environment operating parameters are combined and ordered in a vector of operating parameters, $\boldsymbol{\theta}$, with $n_{\theta} = |\boldsymbol{\theta}|$. Operating conditions may vary; this must be taken into account during circuit design. For this purpose it will be assumed that each operating parameter varies within a bounded interval of real numbers. The lower and upper bounds are denoted by the vectors $\boldsymbol{\theta}^l$ and $\boldsymbol{\theta}^u$ respectively:

$$\boldsymbol{\theta} \in \mathbb{R}^{n_{\theta}} \wedge \boldsymbol{\theta}^l \preceq \boldsymbol{\theta} \preceq \boldsymbol{\theta}^u \quad (2.11)$$

The relation \preceq is defined for arbitrary vectors \mathbf{x} and \mathbf{y} with $|\mathbf{x}| = |\mathbf{y}|$ as follows:

$$\mathbf{x} \preceq \mathbf{y} \iff \forall_{1 \leq i \leq |\mathbf{x}|} x_i \leq y_i \quad (2.12)$$

The nominal value of the operating parameters is denoted by θ^0 and is used when simulating nominal circuit behavior.

2.1.4 Circuit Performances

A circuit performance is an indicator of circuit behavior that is important to the circuit designer or is useful in hierarchical design and function abstraction.

Electrical performances, such as gain and power consumption, depend on the electrical behavior of the circuit. Each is obtained by electrical simulation of the circuit using a suitable test bench, simulator, and analysis method.

Let \mathbf{f}_e denote the vector of electrical performances, such that $n_{\mathbf{f}_e} = |\mathbf{f}_e|$, and $\phi_{\mathbf{f}_e}$ denote the mapping of circuit parameters to electrical performances:

$$\phi_{\mathbf{f}_e} : \mathbb{R}^{n_d} \times \mathbb{R}^{n_s} \times \mathbb{R}^{n_\theta} \longrightarrow \mathbb{R}^{n_{\mathbf{f}_e}} : \begin{bmatrix} \mathbf{d} \\ \mathbf{s} \\ \boldsymbol{\theta} \end{bmatrix} \longmapsto \mathbf{f}_e \quad (2.13)$$

When only the nominal electrical behavior of the circuit as a function of the design parameters is of interest, the statistical and operational parameters are fixed to their nominal values:

$$\phi_{\mathbf{f}_e,0} : \mathbb{R}^{n_d} \longrightarrow \mathbb{R}^{n_{\mathbf{f}_e}} : \begin{bmatrix} \mathbf{d} \\ \mathbf{s}^0 \\ \boldsymbol{\theta}^0 \end{bmatrix} \longmapsto \mathbf{f}_e \quad (2.14)$$

Geometrical performances represent the geometrical properties of the circuit, such as area, width, length, and aspect ratio. Layout synthesis must be completed to get accurate values of geometric performances, as the graph representation of a topology has insufficient geometrical information for accurate calculation. Nevertheless, a model to estimate the geometrical performances from the circuit design parameters could be used, as is done for area in [Has01].

Let \mathbf{f}_g denote the vector of geometric performances, such that $n_{\mathbf{f}_g} = |\mathbf{f}_g|$, and $\phi_{\mathbf{f}_g}$ denote the mapping of circuit design parameters to geometric performances.

$$\phi_{\mathbf{f}_g} : \mathbb{R}^{n_{d\mathcal{E}}} \longrightarrow \mathbb{R}^{n_{\mathbf{f}_g}} : \mathbf{d}_{\mathcal{E}} \longmapsto \mathbf{f}_g \quad (2.15)$$

The nominal electrical and the geometric performances are combined in one vector:

$$\phi_{\mathbf{f}} : \mathbb{R}^{n_d} \longrightarrow \mathbb{R}^{n_{\mathbf{f}}} : \mathbf{d} \longmapsto \mathbf{f} \quad (2.16)$$

With

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_e \\ \mathbf{f}_g \end{bmatrix}; \quad n_{\mathbf{f}} = n_{\mathbf{f}_e} + n_{\mathbf{f}_g} \quad (2.17)$$

The image of the design space, \mathcal{D} , by the mapping $\phi_{\mathbf{f}}$ is denoted by \mathcal{F} :

$$\mathcal{D} \xrightarrow{\phi_{\mathbf{f}}} \mathcal{F} \quad (2.18)$$

2.1.5 Circuit Sizing Rules

An analog circuit topology, \mathcal{T} , is normally composed of smaller functional sub-blocks, each of which performs a recognized analog operation. For example, two CMOS devices can be connected as a simple current mirror or as a differential pair [GZEA01].

Each sub-block is associated with a set of sizing rules to ensure that it functions as intended and to reduce the mismatch due to statistical variation in parameters. These rules can be derived algebraically, often from simple analytical models, such as the Schichman-Hodges model for CMOS devices [SH68], and from mismatch models, such as the mismatch model of drain current [LHC86, PDW89].

At the circuit level, identification of all sub-blocks and the application of the associated sizing rules should improve overall circuit functionality and ensure that the circuit continues to operate when considering process and operating parameter variation. Identification of sub-blocks and the application of sizing rules may also be necessary for correct layout synthesis.

Functional blocks and sizing rules have been described under various names in a series of publications [HRC89, VLv⁺95, dMHBL01, DNAV99, dMHBL98, MV01, DGS03, BSV04, LHC86, GZEA01, SPS⁺03, MGS08]. In [MGS08], a library of more than 26 CMOS and Bipolar functional blocks is presented, along with a structure recognition algorithm to automatically identify them in a circuit topology. For example, the sizing rules of the NMOS differential pair in Figure 2.1.5 are listed in Table 2.2.

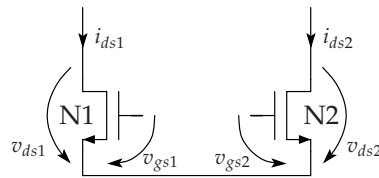


Figure 2.1: NMOS differential pair.

Sizing rules can be separated into two categories:

Table 2.2: Sizing rules of an NMOS differential pair based on [MGS08]

Geometric rules		Electrical rules	
/1/	$L_1 = L_2$	/9/	$ V_{ds2} - V_{ds1} \leq V^{(1)}$
/2/	$W_1 = W_2$	/10/	$ V_{gs2} - V_{gs1} \leq V^{(2)}$
/3/	$W_1 \cdot L_1 \geq Area_m$	/11/	$V_{ds1} - V_{gs1} + V_{th1} \geq V^{(3)}$
/4/	$L_1 \geq L_m$	/12/	$V_{ds1} \geq V^{(4)}$
/5/	$W_1 \geq W_m$	/13/	$V_{gs1} - V_{th1} \geq V^{(5)}$
/6/	$W_2 \cdot L_2 \geq Area_m$	/14/	$V_{ds2} - V_{gs2} + V_{th2} \geq V^{(3)}$
/7/	$L_2 \geq L_m$	/15/	$V_{ds2} \geq V^{(4)}$
/8/	$W_2 \geq W_m$	/16/	$V_{gs2} - V_{th2} \geq V^{(5)}$

$V^{(1)}$ to $V^{(5)} \in \mathbb{R}^+$ are electrical margins,
 $Area_m, L_m, W_m \in \mathbb{R}^+$ are geometric margins.

Electrical rules

These are inequality constraints that depend on the circuit DC bias point under nominal conditions. They can be formulated in terms of the design parameters:

$$\mathbf{d} \xrightarrow{\text{DC analysis}} \mathbf{V}(\mathbf{d}), \mathbf{I}(\mathbf{d}); \mathbf{h}_e(\mathbf{V}(\mathbf{d}); \mathbf{I}(\mathbf{d})) \succeq \mathbf{c}_e^m \quad (2.19)$$

where \mathbf{h}_e denotes the vector of all the electrical constraint functions, the elements of \mathbf{V} are the DC voltages of the topology vertices in \mathcal{V} , and the elements of \mathbf{I} are the DC terminal currents of the topology devices in \mathcal{E} . The constant \mathbf{c}_e^m is a vector of margins.

For abstract analysis, the formulation can be simplified:

$$\boldsymbol{\phi}_{ce}(\mathbf{d}) = \mathbf{h}_e(\mathbf{V}(\mathbf{d}); \mathbf{I}(\mathbf{d})); \quad (2.20)$$

$$\boldsymbol{\phi}_{ce} : \mathbb{R}^{nd} \longrightarrow \mathbb{R}^{nce} : \mathbf{d} \longmapsto \mathbf{c}_e; \quad \mathbf{c}_e \succeq \mathbf{c}_e^m \quad (2.21)$$

where \mathbf{c}_e denotes the vector of electrical constraints, such that $n_{ce} = |\mathbf{c}_e|$.

Geometric rules

These are algebraic equalities and inequalities involving the geometric properties of topology devices, such as the width, length, and area.

The geometric equalities are used to reduce the dimensions of the design space by identifying and eliminating dependent design parameters. For instance, for the set of linear equalities, the dependent parameters can be identified by the application of the Gaussian elimination algorithm.

Let \mathbf{c}_g denote the vector of geometric inequality constraints, such that $n_{cg} = |\mathbf{c}_g|$, and let $\boldsymbol{\phi}_{cg}$ denote the mapping of topology design parameters to geometric inequality constraints:

$$\boldsymbol{\phi}_{cg} : \mathbb{R}^{nd\mathcal{E}} \longrightarrow \mathbb{R}^{ncg} : \mathbf{d}_{\mathcal{E}} \longmapsto \mathbf{c}_g; \quad \mathbf{c}_g \succeq \mathbf{c}_g^m \quad (2.22)$$

Variable elimination methods can also be applied to the system of inequality constraints to create a reduced system of same kind, but with fewer variables:

$$\underbrace{\begin{bmatrix} \mathcal{D}_{\mathcal{E},\text{original}} \\ \mathbf{c}_{g,\text{original}} \succeq \mathbf{c}_{g,\text{original}}^m \end{bmatrix}}_{\text{original problem}} \xrightarrow{\text{elimination methods}} \underbrace{\begin{bmatrix} \mathcal{D}_{\mathcal{E},\text{reduced}} \\ \mathbf{c}_{g,\text{reduced}} \succeq \mathbf{c}_{g,\text{reduced}}^m \end{bmatrix}}_{\text{reduced problem}} \quad (2.23)$$

$$\mathbf{d}_{\mathcal{E},\text{original}} = [\mathbf{d}_{\delta 1}^T; \mathbf{d}_{\delta 2}^T; \dots]^T \xrightarrow{\text{elimination methods}} \mathbf{d}_{\mathcal{E},\text{reduced}}; \quad |\mathbf{d}_{\mathcal{E},\text{reduced}}| \leq |\mathbf{d}_{\mathcal{E},\text{original}}| \quad (2.24)$$

For example, the Fourier-Motzkin elimination algorithm [DE73, SGA07] can be applied to the set of linear inequalities to reduce the number of parameters.

A necessary condition of variable elimination is that the original and reduced systems have the same solutions over the remaining variables. This is necessary so that the individual device design parameters can be calculated and the circuit sized:

$$\mathbf{d}_{\mathcal{E},\text{reduced}} \xrightarrow{\text{inverse of elimination}} \mathbf{d}_{\mathcal{E},\text{original}} = [\mathbf{d}_{\delta 1}^T; \mathbf{d}_{\delta 2}^T; \dots]^T \quad (2.25)$$

Unless explicitly mentioned, it will be assumed in subsequent analysis and discussion that the mapping of (2.24) has been performed implicitly for the design parameters and inequality constraints and that $\mathbf{d} = [\mathbf{d}_{\mathcal{E},\text{reduced}}; \mathbf{d}_{\mathcal{E}\mathcal{B}}]$.

The electrical and the geometric constraints are combined in one vector:

$$\phi_{\mathbf{c}} : \mathbb{R}^{n_{\mathbf{d}}} \longrightarrow \mathbb{R}^{n_{\mathbf{c}}} : \mathbf{d} \longmapsto \mathbf{c}; \quad \mathbf{c} \succeq \mathbf{c}^m \quad (2.26)$$

With

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}_e \\ \mathbf{c}_g \end{bmatrix}; \quad \mathbf{c}^m = \begin{bmatrix} \mathbf{c}_e^m \\ \mathbf{c}_g^m \end{bmatrix}; \quad n_{\mathbf{c}} = n_{\mathbf{c}e} + n_{\mathbf{c}g} \quad (2.27)$$

2.1.6 The Feasible Design Space and Performance Space

The feasible design space, \mathcal{D} , is defined as the subset of the design space, \mathcal{D} , that fulfills the electrical and geometric circuit constraints (the circuit sizing rules):

$$\mathcal{D} = \{ \mathbf{d} \in \mathcal{D} \mid \phi_{\mathbf{c}}(\mathbf{d}) = \mathbf{c} \wedge \mathbf{c} \succeq \mathbf{c}^m \} \quad (2.28)$$

The feasible performance space, \mathcal{F} , consists of all elements in the performance space corresponding to elements of the feasible design space, \mathcal{D} , according to mapping $\phi_{\mathbf{f}}$:

$$\mathcal{F} = \left\{ \mathbf{f} \in \mathbb{R}^{n_{\mathbf{f}}} \mid \exists_{\mathbf{d} \in \mathcal{D}} \phi_{\mathbf{f}}(\mathbf{d}) = \mathbf{f} \right\} \quad (2.29)$$

The mapping of the feasible design space to the feasible performance space can be written as:

$$\mathcal{D} \xrightarrow{\phi_f} \mathcal{F} \quad (2.30)$$

Vectors \mathbf{f}^+ and \mathbf{f}^- denote the upper and lower bounds of the feasible design space and are defined below:

$$\mathbf{f}^+ = \inf \mathcal{F} \quad (\text{infimum of } \mathcal{F} \text{ in } \mathbb{R}^{nf}) \quad (2.31)$$

$$\mathbf{f}^- = \sup \mathcal{F} \quad (\text{supremum of } \mathcal{F} \text{ in } \mathbb{R}^{nf}) \quad (2.32)$$

It is assumed that these bounds always exist for analog circuit design problems.

2.2 Circuit Problem Formulation

In this section, a series of related circuit problems will be defined based on the mapping of the feasible design space to the performance space.

2.2.1 Feasibility Analysis

Feasibility Analysis is the problem of finding any element of the feasible design space:

$$\text{Find any } \mathbf{d} \in \mathcal{D} \quad (2.33)$$

Conversely, the sizing rules are feasible if they are satisfied by at least one design parameter vector.

2.2.2 Circuit Sizing to Meet Performance Specifications

A general specification is an inequality involving a function of circuit performances. Specifications are set at the system design level for each sub-block circuit. The correct operation of a system depends on specification satisfaction in each sub-block. Let $\mathbf{s}(\mathbf{f})$ denote a general vector of specification functions, and let \mathbf{f}^l and \mathbf{f}^u denote lower and upper specifications respectively:

$$\mathbf{f}^l \preceq \mathbf{s}(\mathbf{f}) \preceq \mathbf{f}^u \quad (2.34)$$

Without loss of generality, it is assumed that the specifications take the form of an upper bound, \mathbf{f}^u , on the value of the performances:

$$\mathbf{f} \preceq \mathbf{f}^u \quad (2.35)$$

2 Formulation of the Circuit Sizing Problem

If necessary, a performance space can be transformed so that the specifications take the form of (2.35):

$$\underbrace{\mathbf{f}^l \preceq \mathbf{s}(\mathbf{f}) \preceq \mathbf{f}^u}_{\text{specifications in original performance space}} \implies \underbrace{\mathbf{f}_{\text{new}} \preceq \mathbf{f}_{\text{new}}^u}_{\text{specifications in a new performance space}} \quad (2.36)$$

For example, general specifications $\mathbf{f}^l \preceq \mathbf{s}(\mathbf{f}) \preceq \mathbf{f}^u$ can be transformed as follows:

$$\mathbf{f}^l \preceq \mathbf{s}(\mathbf{f}) \preceq \mathbf{f}^u \implies \underbrace{\begin{bmatrix} \mathbf{s}(\mathbf{f}) \\ -\mathbf{s}(\mathbf{f}) \end{bmatrix}}_{\mathbf{f}_{\text{new}}} \preceq \underbrace{\begin{bmatrix} \mathbf{f}^u \\ -\mathbf{f}^l \end{bmatrix}}_{\mathbf{f}_{\text{new}}^u} \quad (2.37)$$

This transformation preserves the differentiability class of $\mathbf{s}(\mathbf{f})$, but has twice as many dimensions after transformation, such that $|\mathbf{f}_{\text{new}}| = 2|\mathbf{s}(\mathbf{f})|$.

The circuit sizing problem is formulated as follows in the performance space:

$$\text{Find any } \mathbf{f} \in \mathcal{F} \text{ subject to } \mathbf{f} \preceq \mathbf{f}^u \quad (2.38)$$

The solution to the circuit sizing problem in the design space is usually of interest, since it can be readily used to synthesize a circuit:

$$\text{Find any } \mathbf{d} \in \mathcal{D} \text{ subject to } \mathbf{f} \preceq \mathbf{f}^u \text{ where } \phi_{\mathbf{f}}(\mathbf{d}) = \mathbf{f} \quad (2.39)$$

Chapter 3

Overview of Layout Synthesis Steps

3.1 Introduction

A layout is blueprint of planar geometric shapes used to create photo masks for the physical realization of an IC in a specific fabrication technology using the technique of photo lithography.

The layout of a circuit block can be separated into three steps:

- Layout of individual physical devices, such as MOS transistors, and polysilicon capacitors.
- Compact placement of device layout polygons relative to each other on a plane.
- Routing of connections between device terminals, as well as the circuit pins.

Layout synthesis is followed by electrical and geometric verification to ensure the correctness of the synthesis process.

Electrical behavior may change in ways that may not be accounted for during circuit sizing at the topology level. Electrical verification aims to check if the electrical constraints and the specifications set on electrical performances are satisfied by the circuit post-layout synthesis. In order to complete electrical verification, an electrical model must be extracted from the layout geometry and the technology information.

Geometric verification checks if the layout geometry fulfills the technology layout rules (design rule checking or DRC), as well as the specifications set on geometric performances, such as width, length, area, and aspect ratio. Layout design rules specify geometric and connectivity restrictions at the layout level. They are particular to a semiconductor manufacturing process and ensure the correctness of a mask set. They also ensure sufficient margins to account for variability in semiconductor manufacturing processes.

The remainder of this chapter is organized in four sections. Sections 3.2 through 3.4 describe the three steps of layout synthesis in detail. The conditions necessary to

ensure circuit functionality and robustness after layout are explained. Algorithms and techniques found in literature to automate each step are also listed. Section 3.5 reviews the techniques used to extract an electrical model from a geometric layout.

3.2 Layout of Individual Physical Devices

For each device in the circuit topology, the planar geometric layers needed to create the physical realization using the process of photolithography are drawn.

For example, Figure 3.1(a) shows a simplified cross section of a fabricated NMOS transistor, while Figure 3.1(b) shows the geometric layout information needed to correctly fabricate the NMOS device. The NMOS device layout, as pictured, is formed of 16 rectangles in five different layers.

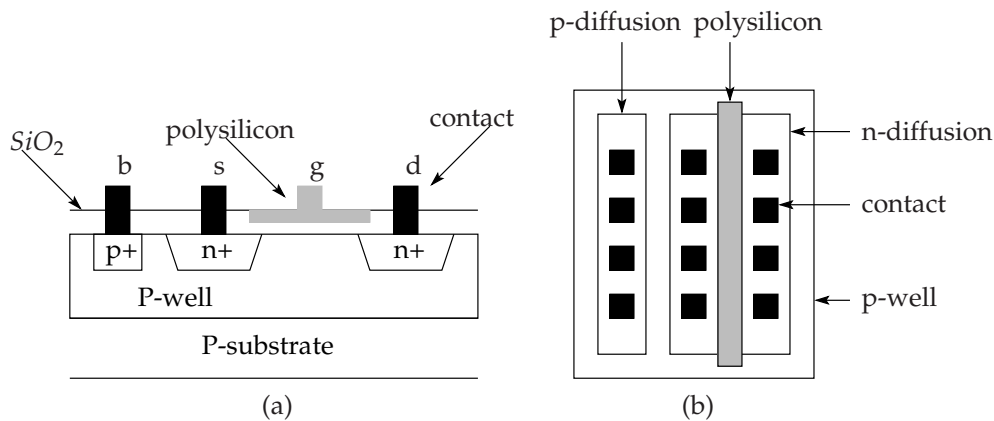


Figure 3.1: Physical cross section and layout of an NMOS transistor.

In order to systematically create device layouts, each device in the circuit topology is attached to a list of layout parameters that depend on the device type, such as NMOS, PMOS, polysilicon capacitor, etc. For example, the number of folds and gate orientation are two layout parameters of a CMOS device. The layout parameters are used to draw the geometric layout of the device according to the layout rules of the used fabrication technology.

3.2.1 Device Layout Automation

A procedure can be used to map a list of layout parameters to a device layout in a systematic manner that meets all technology layout rules. For example, in the commercial Cadence analog design framework, parametric cells (PCELLS) are created for each type of device, and used to map layout parameters to a device layout [Cad08].

3.3 Device Placement

The individual device layouts are placed relative to each other on a plane. There may be innumerable possible arrangements to place the devices, however certain considerations must be taken into account during placement; these considerations are listed below:

- Circuit geometric constraints must be satisfied by the layout; these constraints include the technology layout rules and the circuit dimension, area, and aspect ratio specifications.
- The placement should be compact – maximizing area utilization.
- The parasitic devices on both halves of a differential circuit must be balanced. At the placement level, this is ensured by placing devices symmetrically along differential signal paths.
- In order to be successful, circuit routing should be taken into account during the placement step. Connected devices should be placed in proximity, thereby reducing total wire length and parasitic routing resistance. Noise sensitive signal paths should be kept away from noise sources to avoid coupling through layout parasitic coupling capacitors. Margins of space must be left between devices to insure that device terminal are unblocked and reachable by routing layers. These margins must also be wide enough to avoid routing congestion [Sax07]. Symmetric device placement is also necessary to create symmetric routing.
- The variation in the electrical behavior between matched devices is dependent on device placement, as will be described here: Process anisotropic effects are caused by certain manufacturing steps, such as plasma etching, ion implant angle, or from lattice orientation. Adjacent structures to matched devices may have a systematic influence on the value of the process parameters. Die stress from packaging or thermal gradients may cause considerable systematic drift in process parameter values. Finally, the variance in the value of some process parameters, such as CMOS nominal threshold voltage V_{th0} , depends on the distance between devices; this is modeled by the distance term in Pelgrom's law [LBSG07]. It is difficult to numerically model the effect of placement-dependent variation at the device level, since information about the spacial variation in the value of the process parameters is typically not supplied by the fabrication technology foundry. However, the effect of placement on electrical performance can be minimized by using appropriate layout techniques [Has01]. For example, the appropriate techniques for matched CMOS devices are tabulated in Table 3.1, along with the source of mismatch that is minimized by each technique. The techniques are adjacent parallel placement, matching drain-to-source orientation, splitting and interleaving of devices fingers, splitting and common centroid layout, and the use of dummy elements to surround matched devices.

Table 3.1: Layout techniques for matched CMOS devices

Layout techniques	Local process parameters	Systematic gradients	Adjacent structures	Anisotropic effects
Adjacent parallel placement	X	X		
Surround by dummy elements			X	
Common centroid	X	X		
Split and interleave	X	X		
Match drain-to-source orientation				X

In the state of the art, placement considerations have been formulated as geometric constraints to limit the space of possible placement arrangements. In [EK96], the proposed constraints are maximum area, the deviation from a specified circuit aspect ratio, and a maximum path delay for each routing path. Common centroid and symmetry constraints are formulated in [XY09], while in [SEG⁺08], device proximity, symmetry, common centroid, and minimum distance constraints are used.

The type and number of placement constraints considered during placement will affect the electrical circuit performance values as demonstrated in [ESGS10, ESL⁺11].

3.3.1 Circuit Placement Automation

Automation can be applied to two aspects of circuit placement:

The first aspect is the automatic formulation of geometric placement constraints. A successful automation method will recognize the possible constraints involving two or more devices, then rank conflicting constraints according to importance in optimizing the layout. In [LCL09, SEG⁺08], placement constraints are grouped according to the clusters of devices to which they are applied. The constraint groups are then hierarchically ordered according to the importance of individual constraints. In [CMSV93, CSV93, MCFSV96, CS92], circuit sensitivity analysis is performed prior to placement in order to identify the matching and symmetry constraints that must be used to improve electrical behavior. The method in [HDC⁺04, KSH94, Ars96] performs a structural analysis of the circuit topology to recognize basic circuit sub-blocks and generate symmetry constraints. The algorithm in [ESGS10] generalizes the structural analysis method to the recognition of proximity, alignment, symmetry, and common centroid constraints; these constraints are ordered hierarchically according to importance and circuit topology.

The second aspect is the automatic generation of circuit placements that satisfy the placement constraints. Two approaches towards placement generation can be found in literature. In the first approach, the position of each device is stored as

an (x, y) planar coordinate, and the space of circuit placements is set of all possible combinations of device coordinates. Circuit placement can commence by using a search algorithm to find a set of device coordinates that satisfied all placement constraints [NSS85, CGRC91, LGS95, MCFVS96]. The overlap of devices on the plane is avoided by introducing additional placement constraints. The disadvantage of this approach is the high dimensionality of the search space (the space of possible placements), which is $\mathbb{R}^{2|\mathcal{E}|}$, where $|\mathcal{E}|$ is the number of devices. Each additional device increases the dimensions of the search space by two. High dimensionality will result in a high computational time to find a feasible circuit placement; furthermore, it is impossible to find the complete set of feasible placements with this approach.

In the second approach to automatic placement, a topological representation is used to encode placements. Topological representation does not allow device overlap and the number of possible placement arrangements is finite [GCY99]. There are many mathematical structures for the topological representation of planar rectangular shapes, they include the Sequence Pair [MFNK96], Bounded Sliceline Grid [NFMK96], O-Tree [GCY99, PCLX01], Corner Block List [HHC⁺00, MYP07], and B*-tree [CCWW00, BMM⁺04, WCC03, SEG⁺08] structures. The B*-tree structure has the lowest solution space redundancy and can represent the largest space of possible placements when compared to other topological representations [CCWW00]. All the topological methods found in literature use non-deterministic simulated annealing [BSMD08, SK06] to search the topological placement space for feasible placements that meet the placement constraints; the only exception is the method of [SEG⁺08] that uses a deterministic enumeration using enhanced shape functions and B*-trees.

3.4 Routing

After the individual device layouts are drawn and compactly placed, the connections between device terminals, as well as device terminals and the (external) pin connections of the circuit block are routed. Typically two or more metal layers in a fabrication technology are designated for circuit routing.

As with device placement, the routing operation is restricted by a set of geometric constraints:

- The technology layout rules must remain satisfied after routing.
- Geometric constraints, beyond what is included in the technology layout rules, are set to improve post-layout electrical behavior in terms of functionality. A maximum wire length and minimum wire width are specified for each metal layer to limit connection resistance and total load capacitance. The allowed number of contacts (vias), wire corners, and wire crossovers along a connection may be limited to reduce resistance and coupling capacitance. The minimum separation between parallel and between tandem wires is specified to limit coupling

capacitance. For symmetrically placed devices and differential signal paths, resistance and load capacitance is matched, or the routing geometry is mirrored when possible. For noise sensitive nodes in the circuit topology, the maximum coupling capacitance to other nodes may be specified. Routing congestion can also be estimated and minimized [Sax07, CSX⁺05, AKSW06, SYL09].

- Additional rules may be specified to improve post-layout robustness and reliability and to address failure mechanisms. For example, the minimum wire width of a routing connection may be increased to insure that the maximum current density through the connection does not exceed the preset technology limit. This is important to avoid metal migration, in which the atoms move within the wire, leaving a break in the conductor [WVN⁺06, CLL⁺06].

3.4.1 Circuit Routing Automation

Several automatic routing algorithms are found in literature [CGRC91, RM08, HRM08, Cad03a].

The dominant automatic routing methodology is shape-based routing. Shape-based routing algorithms can handle complex constraints such as differential pair routing, wire shielding, bounds on parasitic coupling capacitance and wire resistance, as well as other custom design requirements.

The method in [Cad03a] uses adaptive routing. In a first run of the shape-based routing algorithm, the auto-router tries to wire all connections while ignoring some routing constraints, such as minimum wire separation and the clearance rules between components. In subsequent runs, connections with constraint violations are ripped off and routing is retried.

3.5 Post-Layout Electrical Model Extraction

In order to simulate the post-layout electrical performances, an electrical model of the circuit is extracted from the layout geometry. A review of layout electrical model extraction can be found in [KLBS01].

Circuit topology devices are identified directly from the layout geometry layers (LVS extraction). Some parasitic components, such as CMOS coupling capacitors c_{db} and c_{sb} , depend on the area and perimeter of the topology devices and are accounted for directly within the device models.

The layout features that must be considered in the electrical model depend on circuit application, operating environment, and the level of accuracy desired in the calculated value of the electrical performances. The computational effort the designer is willing to expend in model extraction is an additional factor.

The substrate structure needs to be modeled if there are noise generating devices on the chip, and the coupling of noise through the substrate has a significant effect on the value of the electrical performances [vHBD⁺02, Hey04, HMF05, AM08]. Algorithms to extract and model the substrate can be found in [Cad, OBA⁺03]. Some applications may require eddy currents in the substrate to be modeled.

For routing interconnects, self and mutual inductance is significant in circuits operating at a relatively high frequency, such as radio frequency (RF) circuits, while only parasitic wire resistance and coupling capacitance is important in low frequency analog circuits. The methodology used to extract the inductance and coupling capacitance will affect the accuracy of electrical performance calculation as well as the computational cost of extraction.

For high accuracy and high computational cost, a three dimensional (3D) electromagnetic field solver based on the finite difference or finite element methods is used, such as the algorithm and commercial tool in [Mag06].

For low accuracy and low computational cost, analytical-geometric models of capacitance may be used. The models need only be generated once for a fabrication technology, after which they can be applied to any circuit layout. Examples of analytical-geometric model generation and application can be found in [LGS95, ARSR96, CHA⁺92]. The algorithm in [LGS95] claims a 10% error in the value of coupling capacitors in comparison to a 3D solver.

Algorithms based on the boundary element method (BEM) or integral equations offer a compromise in the tradeoff between accuracy and cost. The algorithm in [YLWH04] implements a hierarchical form of the BEM to extract the whole interconnect capacitance matrix with one computation and with an average error of 2.7% compared to a 3D solver. The method in [KL00] is an integral equation method with a new representation for charge distributions that decouples charge variation from conductor geometry. In this method, the error is claimed to be below 3% compared to the 3D solver in [Mag06] for capacitors of a value greater than 2fF.

Diffusion area impedance can be calculated by formulating and solving the diffusion equation. In [Cad05], the diffusion equation is formulated as a two dimensional (2D) problem that can be expediently solved using a 2D Laplace solver.

Lossy or lossless model order reduction can be used to reduce the complexity of the passive parasitic network of extracted devices; this is done under consideration of the circuit bandwidth [FF95, OCP98, PCL96, PS05].

Chapter 4

New Automatic Constraint-Based Layout Synthesis Flow

4.1 Introduction

In this chapter, a novel automatic layout synthesis flow is presented. The new flow combines placement and routing algorithms from the state of the art with new concepts to create a functional and robust circuit layout. This is done starting from a circuit topology, \mathcal{T} , and a list of design parameters, \mathbf{d} . Steps that require the input of a decision maker in traditional layout synthesis are replaced by automata.

Each step in the synthesis flow is completely constraint-driven, such that layout selection is completed under consideration of a predefined set of device, placement, and routing constraints – collectively called the *synthesis rules*.

The synthesis rules can be selected automatically or set up by the designer. They need only be defined once for a circuit topology and fabrication technology, after which they can be applied for different values of the circuit design parameters.

The space of all possible layouts that satisfy the synthesis rules is thoroughly explored. From this exploration, a final layout is selected that best meets the electrical and geometric performance specifications. The steps of the new flow are illustrated in Figure 4.1, while the details of each step are given in Sections 4.2 through 4.6 of this chapter.

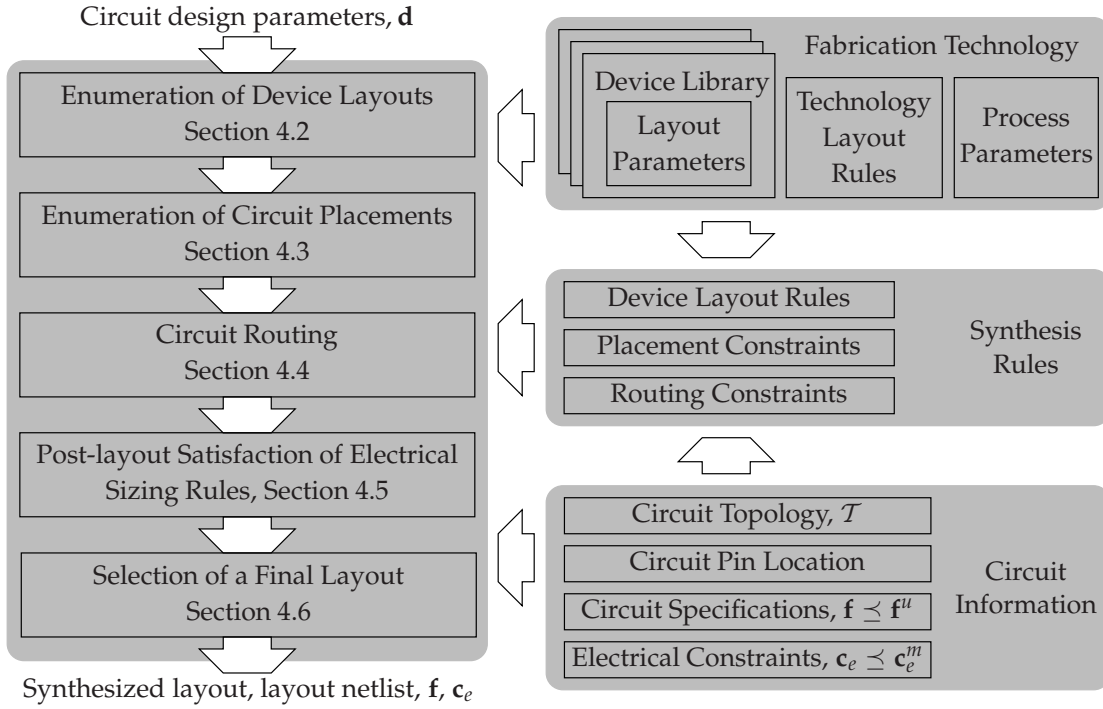


Figure 4.1: A new automatic layout synthesis flow.

4.2 Enumeration of Device Layouts

The design parameters of each circuit device, $\delta \in \mathcal{E}$, are elements of the circuit design parameters, such that $\mathbf{d}_{\mathcal{E}} = [\mathbf{d}_{\delta 1}^T; \mathbf{d}_{\delta 2}^T; \dots]^T$. If the design parameter space is normalized according to (2.7) or transformed according to (2.23) and (2.24), then these mappings are inverted:

$$\mathbf{d}_{\mathcal{E}, \text{reduced, normalized}} \xrightarrow{(2.8), (2.25)} \mathbf{d}_{\mathcal{E}, \text{original}} = [\mathbf{d}_{\delta 1}^T; \mathbf{d}_{\delta 2}^T; \dots]^T \quad (4.1)$$

Let λ_{δ} denote the vector of layout parameters of device $\delta \in \mathcal{E}$. The elements of λ_{δ} depend on the device type, such as NMOS, PMOS, polysilicon capacitor, etc. The space of all valid device layouts is denoted by \mathcal{L}_{δ} , such that $\lambda_{\delta} \in \mathcal{L}_{\delta}$.

For example, if δ is an CMOS device, then $\lambda_{\delta} = \lambda_{\text{CMOS}}$ as given in Table 4.1. The CMOS layout parameters in this example are the number of device fingers, n_f , and the transistor finger width and length, W_f and L_f respectively. Additional layout parameters define the device orientation, ORE, and the location of substrate taps, STL. In Figure 4.2, the layout parameters in combination with the technology layout rules are used to create the layout of an NMOS device. Geometric dimensions not explicitly fixed by the value of the layout parameters or the technology layout rules offer additional degrees of freedom during layout creation.

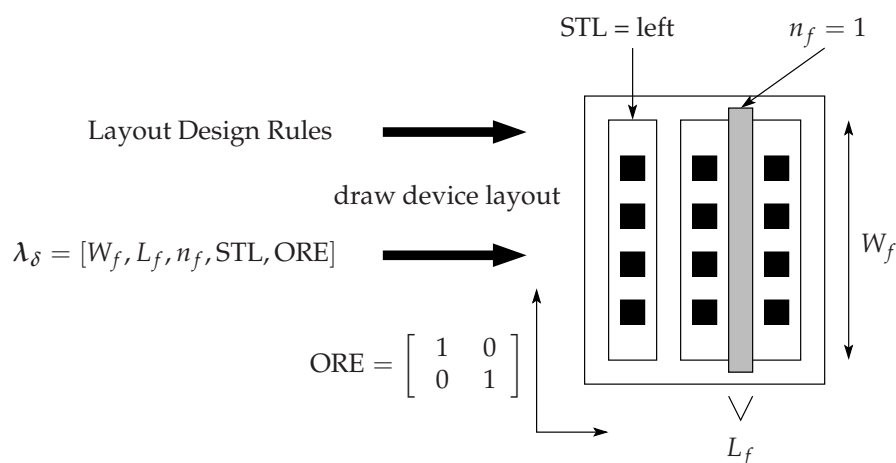


Figure 4.2: Layout parameters mapped to the layout of an NMOS transistor.

Table 4.1: CMOS device layout parameters

$\lambda_{\text{CMOS}} \in \mathcal{L}_{\text{CMOS}}, \mathcal{L}_{\text{CMOS}} = \mathcal{L}_{W_f} \times \mathcal{L}_{L_f} \times \mathcal{L}_{n_f} \times \mathcal{L}_{\text{STL}} \times \mathcal{L}_{\text{ORE}}$			
i	description	$(\lambda_{\text{CMOS}})_i$	Domain
1	finger width	W_f	$\mathcal{L}_{W_f} = [W_{\min} : W_{\text{step}} : W_{\max}]$
2	finger length	L_f	$\mathcal{L}_{L_f} = [L_{\min} : L_{\text{step}} : L_{\max}]$
3	# of fingers	n_f	$\mathcal{L}_{n_f} = \mathbb{N}^+$
4	Substrate Tap location	STL	$\mathcal{L}_{\text{STL}} = \{\text{left, right, both, none}\}$
5	Orientation and Reflection	ORE	$\mathcal{L}_{\text{ORE}} = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \right\}$ (described by rotation and reflection matrices)

The device design parameters, \mathbf{d}_δ , are mapped to device layout parameters, λ_δ , in a manner that preserves the electrical characteristics of the device. In general, multiple valid layouts can be realized for the same value of the device design parameters.

Let \mathcal{V}_δ denote the set of layout parameters possible for a single value of the device design parameters:

$$\mathbf{d}_\delta \xrightarrow{\text{multivalued mapping}} \mathcal{V}_\delta; \quad \mathcal{V}_\delta = \{\lambda_\delta^{(1)}, \lambda_\delta^{(2)}, \dots, \lambda_\delta^{(n)}\}; \quad n \geq 1 \quad (4.2)$$

For the purpose of illustration, Figure 4.3 shows five different layouts of an NMOS device, denoted by parameter vectors $\lambda_\delta^{(1)}$ to $\lambda_\delta^{(5)}$, that are valid for the same value of device design parameter vector \mathbf{d}_δ .

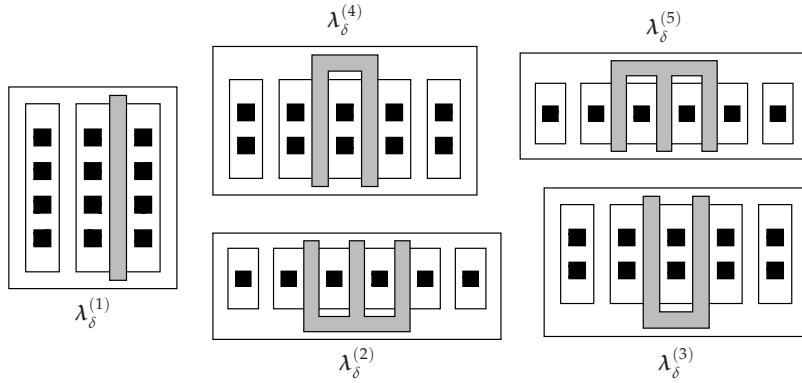


Figure 4.3: Many device layouts are possible for the same device design parameter values: $\mathbf{d}_\delta \mapsto \{\lambda_\delta^{(1)}, \lambda_\delta^{(2)}, \lambda_\delta^{(3)}, \lambda_\delta^{(4)}, \lambda_\delta^{(5)}\}$.

The complete set, $\mathcal{V}_{\text{CMOS}}$, of valid layouts parameters for a CMOS device with design parameters, $\mathbf{d}_{\text{CMOS}} = [W, L]$, is defined in (4.3)*:

$$\mathcal{V}_{\text{CMOS}} = \left\{ \left[\begin{array}{c} W_f \\ L_f \\ n_f \\ \text{STL} \\ \text{ORE} \end{array} \right] \left| \begin{array}{l} W_f \in \mathcal{L}_{W_f}, \quad L_f \in \mathcal{L}_{L_f}, \quad n_f \in \mathcal{L}_{n_f}, \\ \text{STL} \in \mathcal{L}_{\text{STL}}, \quad \text{ORE} \in \mathcal{L}_{\text{ORE}}, \\ W_f = \left\lfloor \frac{W}{n_f \cdot W_{\text{step}}} \right\rfloor \cdot W_{\text{step}}, \quad L_f = \left\lfloor \frac{L}{L_{\text{step}}} \right\rfloor \cdot L_{\text{step}} \end{array} \right. \right\} \quad (4.3)$$

The CMOS layout parameters and domains \mathcal{L}_{W_f} , \mathcal{L}_{L_f} , \mathcal{L}_{n_f} , \mathcal{L}_{STL} , and \mathcal{L}_{ORE} are defined in Table 4.1. Constants W_{step} and L_{step} are the minimum increment step for width and length allowed because of layout manufacturing grid alignment.

The folding of a single CMOS device into a number of fingers, n_f , connected in parallel helps to create compact placements – thereby improving the value of the geometric circuit performances, \mathbf{f}_g , such as area and aspect ratio. Folding also changes the parasitic gate, drain, and source resistance, as well as each extrinsic device capacitance; this, in turn, alters the drain current, small signal transconductance, and frequency response of the CMOS device [YKC⁺05, KKC⁺08]. As a result, electrical

* $\lfloor \dots \rfloor$: is used to denote rounding to the nearest integer.

performances, f_e , must be evaluated to find the optimal number of fingers for each circuit device. The electrical constraints, $\mathbf{c}_e \succeq \mathbf{c}_e^m$, must also remain satisfied.

The number of possible folding combinations for the complete circuit is exponential in the number of CMOS devices:

$$\text{possible folding combinations} = |\mathcal{L}_{n_f, \delta_1} \times \cdots \times \mathcal{L}_{n_f, \delta_i} \times \cdots \times \mathcal{L}_{n_f, \delta_m}| \quad (4.4)$$

where $\{\delta_1, \dots, \delta_i, \dots, \delta_m\}$ are the CMOS devices in the circuit and $\mathcal{L}_{n_f, \delta_i}$ is the set possible fingers allowed for the i -th device.

A solution to find the optimal number of fingers for each device is to replace W with $[W_f, n_f]$ as device design parameters. Parameters n_f and W_f have discrete domains, therefore discrete algorithms are needed to explore this revised design space.

Stochastic algorithms are used in [ABD03, ORC96, GWS90, PKR⁺00, SCP07] for discrete circuit sizing, while a deterministic approach was employed in [PMGS08, PZG10, PG11]. Using circuit examples, stochastic approaches have been shown to converge slowly. The referenced deterministic approaches can be used to solve a problem, only if the circuit performances and constraint functions can be evaluated in between the discrete elements of the design space. If the discrete design space cannot be extended to a continuous domain and the circuit performances and constraint functions cannot be evaluated over this extended domain, then the deterministic approaches cannot be applied directly. A principal problem of all the discrete sizing methods is that the effect of circuit placement is not considered. Many possible folding combinations can be readily discarded post placement, since they do not lead to compact circuit placements – wasting effort in design space exploration.

A second solution to find the optimal number of fingers for each device, and the one used here, is to retain total width, W , as a design parameter, then enumerate and collectively assay all possible folding combinations.

Additional constraints can be applied to the multivalued mapping of (4.2) and to membership in \mathcal{V}_δ in order to preclude layout realizations that will not result in compact device placement, good routing quality and proper electrical behavior after synthesis. Hereinafter, these constraints will be called *device layout rules*. The placement exploration algorithm, discussed in Section 4.3, is then applied to identify the possible folding combinations that result in the most compact of circuit placements only. Finally, electrical constraints and performances need only be considered for the remaining fraction of possible finger combinations

The constrained multi-valued mapping between design and layout parameters is described in Section 4.2.1 for single CMOS devices. In Section 4.2.2 constrained enumeration is extended to CMOS functional blocks, such as matched devices, current mirrors, and level shifters, that are split and laid out in a common centroid configuration to improve performance.

A similar constrained mapping is possible for other types of devices, such as polysilicon capacitors and resistors; and for other CMOS layout configurations, such as merged fingers.

4.2.1 Constrained Enumeration of CMOS Device Layouts

Device layout rules along with a procedure for the constrained mapping of design parameters to layout parameters for a single CMOS device is described below; Algorithm-1 is an equivalent given in pseudo code.

First, designer preferences are applied to restrict the space of valid layouts. The designer specifies three subsets, $\mathcal{L}'_{nf} \subseteq \mathcal{L}_{nf}$, $\mathcal{L}'_{STL} \subseteq \mathcal{L}_{STL}$, and $\mathcal{L}'_{ORE} \subseteq \mathcal{L}_{ORE}$ as inputs to the procedure. These sets are used to restrict the space of valid layouts, such that $[n_f, STL, ORE] \in \mathcal{L}'_{nf} \times \mathcal{L}'_{STL} \times \mathcal{L}'_{ORE}$.

For example, the designer may specify that the number of fingers is to be even and up to 30, such that $\mathcal{L}'_{nf} \leftarrow \{2, 4, 6, 8, \dots, 30\}$, and that left substrate taps only are to be used, such that $\mathcal{L}'_{STL} \leftarrow \{\text{left}\}$. To reduce anisotropic layout effects and improve circuit routing, all CMOS device gates are often oriented identically (vertically, for example); if the devices also have one reflection symmetry along an axis, then the number of orientations is reduced from eight to two, for instance $\mathcal{L}'_{ORE} \leftarrow \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \right\}$.

Designer preferences are handled in lines /2/ and /4/ of Algorithm-1.

A geometric constraint is applied to ensure compliance with the layout design rule prescribing minimum device width:

$$W_{\min} \leq W_f \quad (4.5)$$

W_{\min} is the minimum finger width, as given in Table 4.1, and W_f is defined in (4.3).

Constraint (4.5) is handled in line /9/ of Algorithm-1.

Skewed device geometries will not result in compact circuit placements and are discarded:

$$As_{\min} \leq \frac{\text{device length}}{\text{device width}} \leq As_{\max} \quad (4.6)$$

As_{\min} and As_{\max} are the minimum and maximum device aspect ratio respectively, for example $As_{\min} = 1/3$ and $As_{\max} = 3/1$.

Constraint (4.6) is handled in line /13/ of Algorithm-1.

A disadvantage of folding is a larger statistical variation in the *effective* device width, as well as a larger discretization error in width due to manufacturing grid alignment. These disadvantages can offset advantages of folding by reducing nominal circuit performance as well as robustness to manufacturing variations. Constraints to reduce these disadvantages are derived below.

The effective finger width, $W_{f,eff}$, is defined in the I-V modeling section of the BSIM model [SSKJ87]:

$$W_{f,eff} = W_f - 2\delta W \quad (4.7)$$

where $2\delta W$ is the difference between specified and effective finger width; δW is composed of a constant, W_{int} , a contribution to model the effect of gate, source, and bulk

voltage bias, $\delta W_{f,b}$, a contribution to model the effect of device width, length, and area, $\delta W_{f,g}$, and a statistical component to model manufacturing variation, $\delta W_{f,s}$:

$$\delta W = W_{int} + \delta W_{f,b} + \delta W_{f,g} + \delta W_{f,s} \quad (4.8)$$

W_{int} can be adjusted for systematically prior to device layout generation:

$$W_f \leftarrow W_f + 2W_{int} \quad (4.9)$$

In the current generation of fabrication technologies, the bias and geometric components are an order of magnitude smaller than the constant W_{int} and can be neglected. What remains of importance to consider is the statistical component in δW :

$$W_{f,eff} = W_f - 2\delta W_{f,s} \quad (\text{adjusting for } W_{int} \text{ and neglecting } W_{f,b} \text{ \& } \delta W_{f,g}) \quad (4.10)$$

In [PDML94], $\delta W_{f,s}$ is represented by a global circuit process parameter.

Following from the equation of finger width, W_f , in (4.3) and the equation for effective finger width, $W_{f,eff}$ in (4.10), the effective *total* width of a device, denoted by W_{eff} , is derived as follows:

$$\begin{aligned} W_{eff} &= (W_f - 2\delta W_{f,s}) \cdot n_f \\ &= \underbrace{\left[\frac{W}{n_f \cdot W_{step}} \right] \cdot W_{step} \cdot n_f}_{W_{discrete}} - \underbrace{2\delta W_{f,s} \cdot n_f}_{\delta W_s} \\ &= W_{discrete} + \delta W_s \end{aligned} \quad (4.11)$$

where $W_{discrete}$ represents the discretization of total width and is a modulated function of n_f , and δW_s represents the statistical variation in total width and is a linear function of n_f .

Let W_{error} denote the magnitude of error due to discretization:

$$W_{error} = |W_{discrete} - W| \quad (4.12)$$

The number of fingers, n_f , is limited to values that result in as small error magnitude:

$$W_{error} \leq W_{error-max} \quad (4.13)$$

Constraint (4.13) is handled in lines /8/ and /11/ of Algorithm-1.

Let $\sigma(\delta W_{f,s})$ denote the standard deviation of $\delta W_{f,s}$. The standard deviation of total width, denoted by $\sigma(\delta W_s)$, increases linearly with n_f :

$$\sigma(\delta W_s) = 2\sigma(\delta W_{f,s}) \cdot n_f \quad (4.14)$$

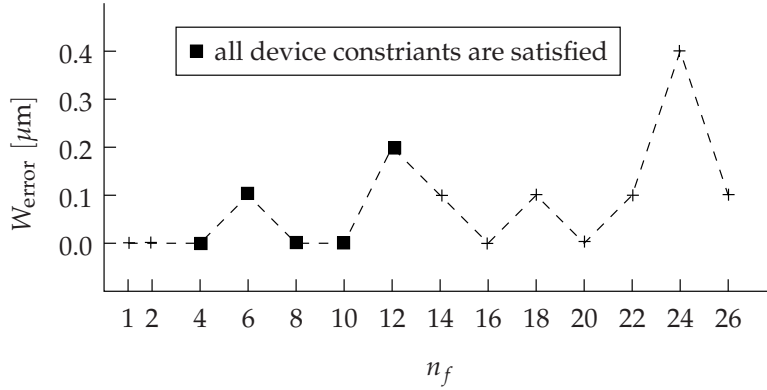


Figure 4.4: W_{error} vs. $n_f = [1, 2, 4, 6, \dots, 26]$ for $W = 100\mu\text{m}$ and $L = 0.7\mu\text{m}$. Constraints (4.5), (4.6), (4.13), and (4.16) are satisfied for $n_f \in \{4, 6, 8, 10, 12\}$.

Without loss of generality, let $\mu(\delta W_{f,s}) = 0$. The term W_{discrete} has no statistical component, therefore $\mu(W_{\text{discrete}}) = W_{\text{discrete}}$ and $\sigma(W_{\text{discrete}}) = 0$. The coefficient of variation in total effective width, denoted by $CV_{W,eff}$, can be calculated as follows:

$$CV_{W,eff} = \frac{\sigma(W_{eff})}{\mu(W_{eff})} = \frac{2\sigma(\delta W_{f,s}) \cdot n_f}{W_{\text{discrete}}} \stackrel{(4.3), (4.11)}{=} \frac{2\sigma(\delta W_{f,s})}{W_f} \quad (4.15)$$

For robustness, statistical variation in $W_{f,eff}$ is limited by selecting a sufficiently large minimum value, W_m , for W_f :

$$W_m \leq W_f \quad (4.16)$$

Constraint (4.16) is handled in line /10/ of Algorithm-1.

Figure 4.4 plots W_{error} versus $n_f = [1, 2, 4, 6, \dots, 26]$ for $[W, L] = [100, 0.7]\mu\text{m}$. If $W_{\text{min}} = 5\mu\text{m}$, $As_{\text{min}} = 1/3$, $As_{\text{max}} = 3$, and $W_{\text{error-max}} = 0.3\mu\text{m}$, then constraints (4.5), (4.6), (4.13), and (4.16) are satisfied for $n_f \in \{4, 6, 8, 10, 12\}$. If, in addition, $\mathcal{L}'_{\text{ORE}} = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \right\}$, and $\mathcal{L}'_{\text{STL}} = \{\text{left}\}$, then the total number of valid layouts is $5 \times 2 \times 1 = 10$.

Algorithm-1 constrained-enumeration-of-CMOS-layouts

```

/1/ input:  $[W, L]$  (device design parameters)
/2/            $\mathcal{L}'_{nf}, \mathcal{L}'_{STL}, \mathcal{L}'_{ORE}$  (designer preferences)
/3/ output: set,  $\mathcal{V}$ , of acceptable device layout parameters

      (apply the designer preferences to reduce the layout space)
/4/  $\mathcal{L}_{nf} \leftarrow \mathcal{L}'_{nf} \cup \{1\}, \mathcal{L}_{STL} \leftarrow \mathcal{L}'_{STL}, \mathcal{L}_{ORE} \leftarrow \mathcal{L}'_{ORE}$ 

      (initialize the output set)
/5/  $\mathcal{V} \leftarrow \emptyset$ 
      for each  $[n_f, STL, ORE]$  in  $\mathcal{L}_{nf} \times \mathcal{L}_{STL} \times \mathcal{L}_{ORE}$  do

/6/    $W_f \leftarrow \left\lfloor \frac{W}{n_f \cdot W_{step}} \right\rfloor \cdot W_{step}$ 

/7/    $L_f \leftarrow \left\lfloor \frac{L}{L_{step}} \right\rfloor \cdot L_{step}$ 

/8/    $W_{error} \leftarrow |W_f \cdot n_f - W|$ 

/9/   if  $W_f < W_{min}$  then next iteration
/10/  if  $W_f < W_m$  then next iteration
/11/  if  $W_{error-max} < W_{error}$  then next iteration

/12/  Map  $[W_f, L_f, n_f, STL, ORE]$  to a geometric layout
      (e.g., call a PCELL in the Cadence framework)

/13/  Aspect-ratio  $\leftarrow \frac{\text{device length}}{\text{device width}}$ 
/14/  if Aspect-ratio  $< A_{smin}$  or  $A_{smax} < \text{Aspect-ratio}$  then next iteration

      (add the current layout parameters to the output set)
/15/   $\mathcal{V} \leftarrow \mathcal{V} \cup \{[W_f, L_f, n_f, STL, ORE]\}$ 

return

```

4.2.2 Constrained Enumeration of CMOS Devices in Common Centroid Layout

A single device in the circuit topology may be divided into a number of smaller identical devices in the layout. This is typical when two or more devices are to be laid out in a common centroid configuration to improve the matching of device properties post fabrication.

The number of device divisions is often fixed in the circuit topology prior to circuit sizing. If the value of the device design parameters is allowed to vary within a wide range, then a fixed number of divisions may produce a sub-optimal layout.

An example is given in Figure 4.5. Two matched NMOS devices, δ_1 and δ_2 , are laid out in a common centroid configuration; variable M denotes the number of device divisions. If $W = 100\mu\text{m}$ for each of the two devices, then $M = 4$ results in devices with suitable layout dimensions. If $W = 50\mu\text{m}$, then $M = 4$ results in devices that violate constraints (4.5) and (4.6), furthermore, the common centroid block has a skewed aspect ratio and a large area; setting $M = 2$ produces a better layout.

In order to solve the problem illustrated in the example above, an extension to the constrained mapping procedure of Section 4.2.1 is described here for a CMOS device placed in a common centroid configuration. In this extension, the number of device divisions is taken into consideration. Algorithm-2 is an equivalent in pseudo code.

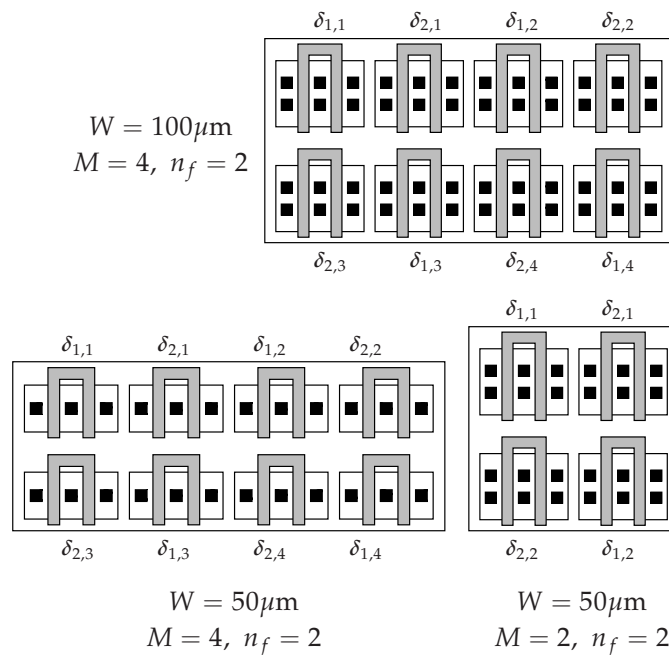


Figure 4.5: Common centroid configurations for two matched NMOS devices, δ_1 and δ_2 ; n_f is the number of fingers and M is the number of divisions.

First, designer preferences for n_f , STL, and ORE are imposed as in Section 4.2.1. Here, n_f refers to the number of fingers *per division*, therefore \mathcal{L}'_{n_f} is typically a set of small positive integers, for example $\mathcal{L}'_{n_f} \leftarrow \{1, 2, 4\}$.

Designer preferences are handled in lines /2/ and /4/ of Algorithm-2.

A set of division values, \mathcal{L}_M , is defined. The set \mathcal{L}_M depends on the interleave patterns sanctioned during common centroid layout. For example, if $\mathcal{L}_M = \{2, 4, 10, 18\}$, then a device may be divided into two, four, 10, or 18 devices in the layout.

This step handled in line /5/ of Algorithm-2.

Geometric constraints (4.5), (4.13), and (4.16), defined in Section 4.2.1, are reapplied here. Only number of divisions, denoted by M , is added to the calculation of W_f in (4.3) and to (4.11), to become (4.17) and (4.18) respectively:

$$W_f = \left\lfloor \frac{W}{M \cdot n_f \cdot W_{\text{step}}} \right\rfloor \cdot W_{\text{step}} \quad (4.17)$$

$$W_{\text{eff}} = \underbrace{\left\lfloor \frac{W}{M \cdot n_f \cdot W_{\text{step}}} \right\rfloor \cdot W_{\text{step}} \cdot M \cdot n_f}_{W_{\text{discrete}}} + \underbrace{2\delta W_{f,s} \cdot M \cdot n_f}_{\delta W_s} \quad (4.18)$$

The geometric constraints are tested in lines /10/ through /17/ of Algorithm-2.

If W is very small, then the device will default to a layout with a single gate and no divisions, so that $[n_f, M] = [1, 1]$. This is achieved, indirectly, by the steps in lines /7/, /8/, and /9/ of Algorithm-2.

In general, multiple combinations of finger and division count may fulfill geometric constraints (4.5), (4.6), (4.13), and (4.16). For the case of an NMOS device with $[W, L] = [100, 0.7]\mu\text{m}$, the shaded cells in Table 4.2 denote combinations of $[n_f, M]$ that fulfill the four geometric constraints, and populate the output set, \mathcal{V} , of Algorithm-2.

Further elimination of layout parameter vectors from the output set; for instance, to improve the aspect ratio of the common centroid layout block; will not be made at the level of an individual device. Aspects of the complete block that is laid out in common centroid configuration will be considered.

Table 4.2: The shaded cells fulfill the geometric constraints (4.5), (4.6), (4.13), and (4.16) for an example NMOS device with $[W, L] = [100, 0.7]\mu\text{m}$

$\mathcal{L}_{n_f} \times \mathcal{L}_M$	1	2	4	10	18
1	-	-	-	+	-
2	-	-	-	+	-
4	+	+	+	-	-

Algorithm-2 enumerate-CMOS-layouts-common-centroid

```

/1/ input:  $[W, L]$  (device design parameters)
/2/       $\mathcal{L}'_{nf}, \mathcal{L}'_{STL}, \mathcal{L}'_{ORE}$  (designer preferences)
/3/ output: set,  $\mathcal{V}$ , of acceptable device layout parameters

      (apply the designer preferences to reduce the layout space)
/4/  $\mathcal{L}_{nf} \leftarrow \mathcal{L}'_{nf} \cup \{1\}, \mathcal{L}_{STL} \leftarrow \mathcal{L}'_{STL}, \mathcal{L}_{ORE} \leftarrow \mathcal{L}'_{ORE}$ 
      (define the set of device dividers)
/5/  $\mathcal{L}_M \leftarrow \{1, 2, 4, 10, 18\}$ 

      (initialize the output set)
/6/  $\mathcal{V} \leftarrow \emptyset$ 
      for each  $n$  in  $\mathcal{L}_{nf}$  do

          (find the largest possible number of device multiples)
/7/ if  $\frac{W}{\max(\mathcal{L}_M) \cdot n} \geq W_{\min}$  then  $M_{\max} \leftarrow \max(\mathcal{L}_M)$ 
          else  $M_{\max} \leftarrow \sup\{x \in \mathcal{L}_M : x \leq \max(\frac{W}{M_{\min} \cdot n}, 1)\}$ 
          (from step /4/,  $1 \in \mathcal{L}_M$ , and the supremum is always an element of  $\mathcal{L}_M$ )

          for each  $M$  in  $\mathcal{L}_M$  with  $M \leq M_{\max}$  do

/8/       $W_{\text{temp}} \leftarrow \max(\frac{W}{M \cdot n}, W_{\min})$ 
/9/       $n_f \leftarrow \sup\{x \in \mathcal{L}_{nf} : x \leq \max(\frac{W}{W_{\text{temp}} \cdot M}, 1)\}$ 

          for each [STL, ORE] in  $\mathcal{L}_{STL} \times \mathcal{L}_{ORE}$  do

/10/      $W_f \leftarrow \left\lfloor \frac{W_{\text{temp}}}{M \cdot n_f \cdot W_{\text{step}}} \right\rfloor \cdot W_{\text{step}}$ 
/11/      $L_f \leftarrow \left\lfloor \frac{L}{L_{\text{step}}} \right\rfloor \cdot L_{\text{step}}$ 
/12/      $W_{\text{error}} \leftarrow |W_f \cdot M \cdot n_f - W|$ 

/13/     if  $W_f < W_{\min}$  or  $W_f < W_m$  then next iteration
/14/     if  $W_{\text{error-max}} < |W_{\text{error}}|$  then next iteration

/15/     Map  $[W_f, L_f, n_f, \text{STL}, \text{ORE}]$  to a geometric layout
          (e.g., call a PCELL in the Cadence framework)

/16/     Aspect-ratio  $\leftarrow$  device length/device width
/17/     if Aspect-ratio  $< A_{s_{\min}}$  or  $A_{s_{\max}} <$  Aspect-ratio then next iteration

          (add the current layout parameters to the output set)
/18/      $\mathcal{V} \leftarrow \mathcal{V} \cup \{[W_f, L_f, n_f, \text{STL}, \text{ORE}; M]\}$ 

return

```

Three cases are identified for analog functional blocks that can take advantage of the common centroid configuration:

Case 1: the devices in common centroid configuration have equal device design parameter values.

This is case of two devices δ_1 and δ_2 forming a differential pair, such that $[W_1, L_1] = [W_2, L_2]$ according to the geometric sizing rules covered in Section 2.1.5.

In this case, the number of divisions, M , and fingers, n_f , as well as parameters W_f , L_f , STL, and ORE are equal for each device, and Algorithm-2 need only be called once. A common centroid interleave pattern with the ratio $M_1 : M_2 = 1 : 1$ is used. The following items are considered when selecting from the output set, \mathcal{V} , of Algorithm-2:

- The area and aspect ratio of the common centroid array, as well as overall circuit compactness, improve with an increase in the number of fingers, n_f , and the number of divisions, M , therefore $[n_f, M]$ should be maximized.
- When $M = 1$ the benefits of common centroid layout are lost, therefore combinations with $M = 1$ should be avoided if possible.
- To benefit from device folding, n_f is to be maximized when possible.

For illustration, the shaded cells in Table 4.2 are ranked as shown in Table 4.3 according to the considerations itemized above.

Table 4.3: The shaded cells of Table 4.2 are ranked for two matched devices

$\mathcal{L}_{n_f} \times \mathcal{L}_M$	1	2	4	10	18
1	–	–	–	4	–
2	–	–	–	3	–
4	5	2	1	–	–

Case 2: the devices in common centroid configuration have equal lengths, but device widths are independent design parameters.

This is case of two devices δ_1 and δ_2 forming a simple current mirror or level shifter, such that $L_1 = L_2$ according to the sizing rules of [MGS08].

In this case, Algorithm-2 is called independently for devices δ_1 and δ_2 . The result is two outputs sets, \mathcal{V}_{δ_1} and \mathcal{V}_{δ_2} . The elements of each set can be ranked in an analogous manner to Case 1, however the ratio of M_1 and M_2 must fit the desired interleave pattern. For example, to use an interleave pattern with $M_1 : M_2 = 1 : 1$, the elements selected from \mathcal{V}_{δ_1} and \mathcal{V}_{δ_2} must satisfy $M_1 = M_2$, and to use an interleave pattern with $M_1 : M_2 = 1 : 2$, the elements selected from \mathcal{V}_{δ_1} and \mathcal{V}_{δ_2} must satisfy $2M_1 = M_2$.

If dummy devices can be used to complete the interleave pattern, then the pattern ratio can be modified. For example, $(M_1 + M_{\text{dummy},1}) : (M_2 + M_{\text{dummy},2}) = 1 : 1$, where $M_{\text{dummy},1}$ and $M_{\text{dummy},2}$ are the number of dummy devices used to complete the ratio of M_1 and M_2 to 1:1.

The cascode, Wilson, improved Wilson, and wide swing cascode current mirrors can be deconstructed into a simple current mirror and a level shifter that are identical in placement, but differ in the routing of connections only.

For level shifter and current mirror banks consisting of many devices, $\delta_1, \delta_2, \dots, \delta_n$, Algorithm-2 is first called independently for each device, after which the common centroid interleave pattern and the ratios $M_1 : M_2, M_1 : M_3, \dots, M_1 : M_n$ must be selected appropriately. Dummy devices may also be used.

Case 3: the devices in common centroid configuration have equal lengths, while the ratio of device widths is a rational value.

This is case of two devices δ_1 and δ_2 forming a simple current mirror or level shifter, such that $L_1 = L_2$ according to the sizing rules of [MGS08]. In contrast to Case 2, only W_1 is independent, while $W_2 = \frac{a}{b}W_1$, where $a, b \in \mathbb{N}^+$.

Without loss of generality, it is assumed that $\frac{a}{b} \geq 1$. The common centroid can be constructed with equal finger widths, $W_{f,1} = W_{f,2}$, if the equation in (4.20) is satisfied:

$$W_2 = \frac{a}{b}W_1 \xrightarrow{(4.17), (4.18)} W_{f,2} \cdot n_{f,2} \cdot M_2 = \frac{a}{b}W_{f,1} \cdot n_{f,1} \cdot M_1 \quad (4.19)$$

$$W_{f,1} = W_{f,2} \xrightarrow{(4.19)} \frac{n_{f,2} \cdot M_2}{n_{f,1} \cdot M_1} = \frac{a}{b}; \quad n_{f,1}, M_1, n_{f,2}, M_2 \in \mathbb{N}^* \quad (4.20)$$

In this case, Algorithm-2 is called for device δ_1 . The result is output set \mathcal{V}_{δ_1} . The elements for which $(\frac{a}{b}n_{f,1} \cdot M_1)$ is a natural number are selected from \mathcal{V}_{δ_1} . The elements of the second device, δ_2 , are constructed by selecting $n_{f,2}$ and M_2 to satisfy the rational equation in (4.20). For example, let the shaded cells in Table 4.2 denote combinations of $n_{f,1}$ and M_1 that fulfill constraints (4.5), (4.6), (4.13), and (4.16) for device δ_1 and let $\frac{a}{b} = \frac{5}{4}$. The elements of \mathcal{V}_{δ_1} with $[n_{f,1}, M_1] = [4, 4]$ are valid for δ_1 , since $\frac{5}{4} \times 4 \times 4 = 20 \in \mathbb{N}$. The corresponding elements for δ_2 are constructed directly by selecting $[n_{f,2}, M_2] = [4, 5]$, as this satisfies the rational equation in (4.20).

If dummy devices can be used, then equation (4.20) can be modified:

$$W_{f,1} = W_{f,2} \xrightarrow{(4.19)} \frac{n_{f,2} \cdot (M_2 + M_{\text{dummy},2})}{n_{f,1} \cdot (M_1 + M_{\text{dummy},1})} = \frac{a}{b}; \quad n_{f,1}, M_1, n_{f,2}, M_2 \in \mathbb{N}^* \quad (4.21)$$

where $M_{\text{dummy},1}$ and $M_{\text{dummy},2}$ are the number of dummy devices used to complete the ratio to $a:b$.

When dummy devices can be used, the elements for which $(\frac{a}{b}n_{f,1} \cdot (M_1 + M_{\text{dummy},1}))$ is a natural number are selected from \mathcal{V}_{δ_1} . Variable $M_{\text{dummy},1}$ provides an additional

degree of freedom. The elements of the second device, δ_2 , are constructed by selecting $n_{f,2}$ and M_2 to satisfy the rational equation in (4.21). Variable $M_{\text{dummy},2}$ provides an additional degree of freedom.

As in Case 2, the cascode, Wilson, improved Wilson, and wide swing cascode current mirrors can be deconstructed into a simple current mirror and a level shifter.

For level shifter and current mirror banks consisting of many devices, $\delta_1, \delta_2, \dots, \delta_n$, only the first device width, W_1 , is independent, while $[W_2, \dots, W_n] = [\frac{a_2}{b}, \dots, \frac{a_n}{b}] \cdot W_1$. In this case, Algorithm-2 is called for device δ_1 . From the output set, \mathcal{V}_{δ_1} , the elements for which $\{(\frac{a_2}{b}n_{f,1} \cdot M_1), \dots, (\frac{a_n}{b}n_{f,1} \cdot M_1)\}$ are natural numbers are retained. The elements of each subsequent device, $\delta_2, \dots, \delta_n$, are constructed by selecting $[n_{f,2}, M_2], \dots, [n_{f,n}, M_n]$ to satisfy the rational equation in (4.20). Dummy devices can also be used, in which case the rational equation in (4.21) must be satisfied for each subsequent device.

4.3 Enumeration of Circuit Placements

For circuit topology \mathcal{T} with devices $\mathcal{E} = \{\delta_1, \delta_2, \dots, \delta_{|\mathcal{E}|}\}$, the possible layout variants for each device, $\{\mathcal{V}_{\delta_1}, \mathcal{V}_{\delta_2}, \dots, \mathcal{V}_{\delta_{|\mathcal{E}|}}\}$, can be generated by the methods discussed in Section 4.2.

The next step in the flow of Figure 4.1 is the enumeration of possible circuit placements given the possible variations of each individual circuit device.

Placement constraints and the minimum margins between devices must be set prior to placement. This is discussed in Sections 4.3.1 and 4.3.2. A formulation for placement enumeration that can be used to generate the most compact circuit placements is described in Section 4.3.3. In Section 4.3.4, a scalar objective function is defined to amalgamate multiple geometric performance specifications in one quality measure; this measure is used in Section 4.3.5 to rank the generated circuit placements.

4.3.1 Placement Constraint Generation

In Section 3.3 the advantages of constrained device placement were discussed. The algorithm of [ESGS10] is used here to generate placement constraints. Structural analysis of circuit topology is performed and the circuit subdivided into hierarchical proximity and symmetry groups, after which proximity, alignment, symmetry, and common centroid constraints are automatically generated. These constraints are ordered hierarchically according to importance and order of application. The automatically generated constraints can then be edited or adjusted by the designer if necessary; layout-bound components such as guard rings and well trenches can be specified as additional placement constraints.

4.3.2 Minimum Device Margins

Minimum margins are specified for each device with regards to every other device in the layout. This is to ensure compliance with the technology layout rules and to ensure that all device terminals are reachable for routing (unblocked). Routing congestion will also be reduced by increasing the margin values.

Minimum margins are illustrated in Figure 4.6 with a diagram and a table of margins for a circuit consisting of three devices. Characters (T, B, L, R) stand for (Top, Bottom, Left, Right). The term $M_L(j, i)$ refers to the margin between device i and device j when device i is placed directly to the *left of* device j in a layout. The terms $M_T(j, i)$, $M_B(j, i)$, and $M_R(j, i)$ have analogous definitions.

Every ordered combination of devices, (i, j) , has four distinct margins: $M_L(j, i)$, $M_T(j, i)$, $M_L(i, j)$, and $M_T(i, j)$. This is because $M_B(j, i) = M_T(i, j)$, $M_R(j, i) = M_L(j, i)$, $M_B(i, j) = M_T(j, i)$, and $M_R(i, j) = M_L(j, i)$, as illustrated in Figure 4.6.

A minimum margin table can be generated automatically for a topology, \mathcal{T} , given the set of possible layout variants $\{\mathcal{V}_{\delta_1}, \mathcal{V}_{\delta_2}, \dots, \mathcal{V}_{\delta_{|\mathcal{E}|}}\}$. The minimum margins about layout-bound components, such as wells or guard rings, and pin connections must also be added to the margin table.

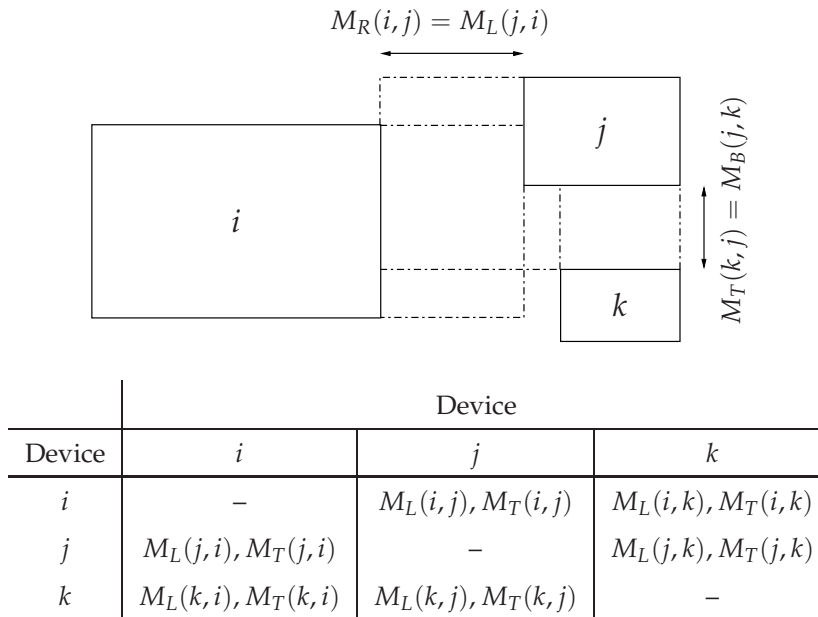


Figure 4.6: Minimum margins between three devices.

4.3.3 Generation of Pareto-Optimal Placements

For circuit topology \mathcal{T} with devices $\mathcal{E} = \{\delta_1, \delta_2, \dots, \delta_{|\mathcal{E}|}\}$ and corresponding device layout variants $\{\mathcal{V}_{\delta_1}, \mathcal{V}_{\delta_2}, \dots, \mathcal{V}_{\delta_{|\mathcal{E}|}}\}$; let \mathcal{C}_p denote the list of placement constraints

and minimum margins that must be satisfied; \mathbf{p} denote any possible circuit placement; and $W(\mathbf{p})$ and $L(\mathbf{p})$ denote the width and length of placement \mathbf{p} respectively. Compact constrained placement can be expressed as a multi-objective minimization problem:

$$\min_{\mathcal{V}_{\delta_1} \times \mathcal{V}_{\delta_2} \times \dots \times \mathcal{V}_{\delta_{|\mathcal{E}|}}} \begin{bmatrix} W(\mathbf{p}) \\ L(\mathbf{p}) \end{bmatrix} \text{ subject to } \mathcal{C}_p \quad (4.22)$$

Problem (4.22) is a two-objective minimization problem. In general, there is no placement solution that simultaneously minimizes both circuit width and length; instead there is a finite set of Pareto-optimal solutions [Par06]:

$$\mathcal{P} = \{\mathbf{p}^1, \mathbf{p}^2, \dots, \mathbf{p}^n\} \text{ is the set of circuit placements to solve (4.22)} \implies \neg \exists_{\{\mathbf{p}^i, \mathbf{p}^j\} \in \mathcal{P}} \begin{bmatrix} W(\mathbf{p}^i) \\ L(\mathbf{p}^i) \end{bmatrix} \prec \begin{bmatrix} W(\mathbf{p}^j) \\ L(\mathbf{p}^j) \end{bmatrix} \quad (4.23)$$

The relation \prec is defined for arbitrary vectors \mathbf{x} and \mathbf{y} with $|\mathbf{x}| = |\mathbf{y}|$ as follows:

$$\mathbf{x} \prec \mathbf{y} \iff \mathbf{x} \preceq \mathbf{y} \wedge \exists_{1 \leq i \leq |\mathbf{x}|} x_i < y_i \quad (4.24)$$

All possible circuit placements not in \mathcal{P} will be worse in satisfying the geometric circuit specifications and are discarded.

The placement exploration algorithm in [SEG⁺08] is used to solve (4.22) and generate set of placement solutions, \mathcal{P} . This algorithm is deterministic and performs a quasi-complete exploration of the possible placement space. It is considered quasi-complete because the B*-tree topological structure it uses can represent the largest space of possible placement arrangements when compared to other topological representations [CCWW00].

Figure 4.7 plots the width and length of a set of example Pareto-optimal placements found using the placement exploration algorithm in [SEG⁺08].

If the device layout variants, $\{\mathcal{V}_{\delta_1}, \dots, \mathcal{V}_{\delta_{|\mathcal{E}|}}\}$, and the placement constraints and minimum margins, \mathcal{C}_p , are well constructed, then \mathcal{P} is not empty.

4.3.4 Geometric Placement Specifications

In a top-down design methodology, maximum circuit area, permissible aspect ratio, and/or maximum circuit width and length may be set as geometric circuit specifications. Boundary values may be assigned at the system or chip floor planning stage.

Let $W(\mathbf{p})$, $L(\mathbf{p})$, $A(\mathbf{p})$, and $As(\mathbf{p})$, denote the width, length, area, and aspect ratio of circuit placement \mathbf{p} respectively, such that:

$$A(\mathbf{p}) = W(\mathbf{p}) \cdot L(\mathbf{p}); \quad As(\mathbf{p}) = L(\mathbf{p})/W(\mathbf{p}) \quad (4.25)$$

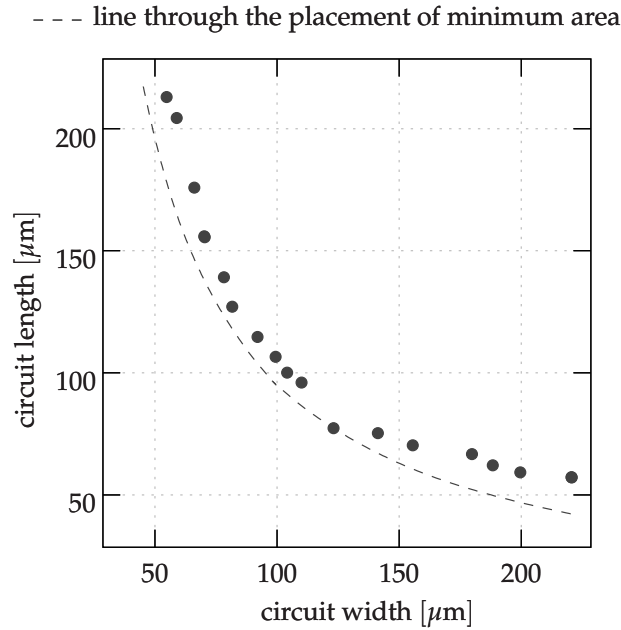


Figure 4.7: A Pareto-optimal set of circuit placements that differ in width and length.

Without loss of generality, two groups of geometric circuit specifications are considered in this section. In the first group, an upper bound is set on the value of circuit width and length:

$$W(\mathbf{p}) \leq W_{\max}, \quad L(\mathbf{p}) \leq L_{\max} \quad (4.26)$$

In the second group, an upper bound is set on circuit area, while circuit aspect ratio is constrained to a range:

$$A(\mathbf{p}) \leq A_{\max}, \quad A_{s_{\min}} \leq As(\mathbf{p}) \leq A_{s_{\max}} \quad (4.27)$$

If the circuit aspect ratio is fixed such that $A_{s_{\min}} = A_{s_{\max}} = k$, then the specifications in (4.27) become a special case of the specifications in (4.26), since:

$$W \cdot L \stackrel{(4.25)}{=} A \stackrel{(4.27)}{\leq} A_{\max}, \quad As \stackrel{(4.25)}{=} \frac{L}{W} = k \implies W \leq \underbrace{\sqrt{A_{\max}/k}}_{W_{\max}}, \quad L \leq \underbrace{\sqrt{A_{\max} \cdot k}}_{L_{\max}} \quad (4.28)$$

In the worst case, the geometric specifications are not satisfied by any placement found with the placement exploration algorithm, as illustrated in Figure 4.8.

It is desired to rank the Pareto-optimal set of placements according to how well the geometric specifications are satisfied. After ranking it is possible to discard the *worst* placements and reduce the cost of further synthesis steps. For example, the rightmost placement in Figure 4.8(b) has an aspect ratio of $60/225 \approx .27$, while $A_{s_{\min}} = 4/5$; the aspect ratio of the aforementioned placement is clearly too skewed to be further considered in synthesis, especially since other placements with much better aspect ratio and similar area are present in the Pareto-optimal set of placements.

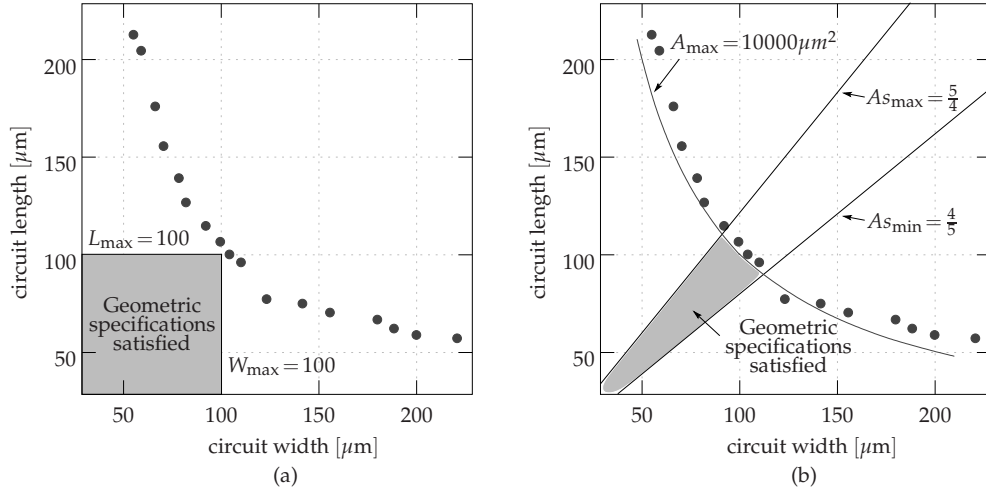


Figure 4.8: (a) Width and length specifications, as given by (4.26), are not satisfied by any placement. (b) Area and aspect ratio specifications, as given by (4.27), are not satisfied by any placement.

In addition to (ordinal) ranking, it is desirable to have a suitable metric combining the geometric performances – width and length, or area and aspect ratio – so that meaningful comparison can be made with the electrical performances, such as gain and bandwidth, as is done, for example, in Section 4.6.2.

Towards the goal of systematic ranking and suitable metric, a scalar objective is defined that combines the value of the geometric performances. The new objective function is called the *modified area* and is denoted by \hat{A} . The minimum value of \hat{A} is set to A_{\max} – attained when the geometric specifications are met.

Two different functions will be used for the two groups of geometric specifications given in (4.26) and (4.27):

Case 1: circuit width and length specifications are given by (4.26).

In this case, the modified area will penalize the normalized maximum deviation in circuit width or length beyond the specification bounds:

$$\begin{aligned} \hat{A}(\mathbf{p}) &= \left(1 + \max \left(\frac{W(\mathbf{p}) - W_{\max}}{W_{\max}}, \frac{L(\mathbf{p}) - L_{\max}}{L_{\max}}, 0 \right) \right) \cdot A_{\max} \\ &= \max (W(\mathbf{p}) \cdot L_{\max}, L(\mathbf{p}) \cdot W_{\max}, A_{\max}) \end{aligned} \quad (4.29)$$

where $A_{\max} \stackrel{(4.25)}{=} W_{\max} \cdot L_{\max}$.

There is an equivalence between the original specifications in (4.26) and \hat{A} as given by (4.29) such that:

$$\begin{aligned} (W(\mathbf{p}) \leq W_{\max} \wedge L(\mathbf{p}) \leq L_{\max}) &\iff (\hat{A}(\mathbf{p}) = A_{\max} \wedge A(\mathbf{p}) \leq A_{\max}) \\ \neg (W(\mathbf{p}) \leq W_{\max} \wedge L(\mathbf{p}) \leq L_{\max}) &\iff (\hat{A}(\mathbf{p}) > A_{\max}) \end{aligned} \quad (4.30)$$

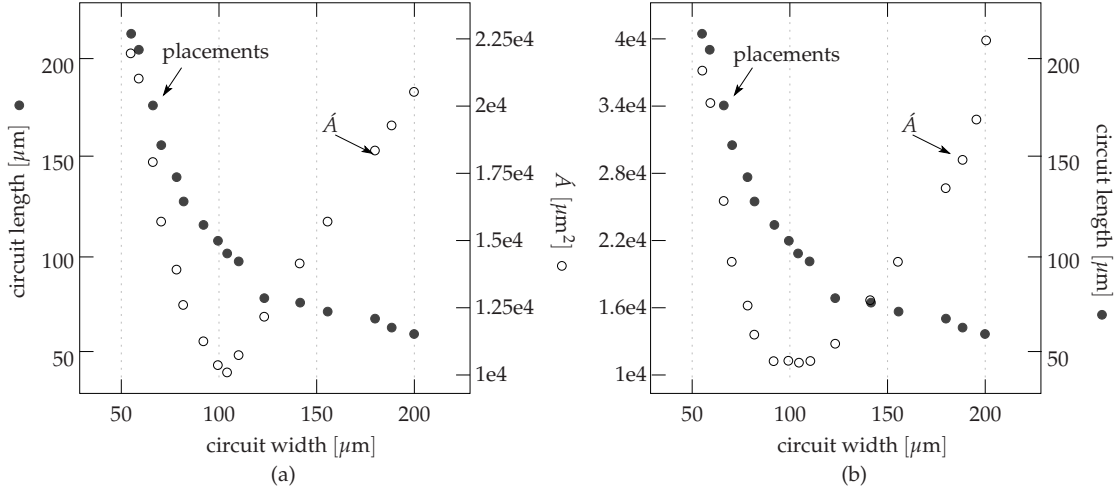


Figure 4.9: (a) $W \leq 100\mu\text{m}$ and $L \leq 100\mu\text{m}$, \hat{A} is calculated according to (4.29). (b) $A \leq 10000\mu\text{m}^2$ and $\frac{4}{5} \leq As \leq \frac{5}{4}$, \hat{A} is calculated according to (4.33).

In Figure 4.9(a), the value of \hat{A} as calculated by (4.29) is plotted for the placements and geometric specifications illustrated in the example of Figure 4.8(a).

Case 2: circuit area and aspect ratio specifications are given by (4.27).

If $As(\mathbf{p}) < As_{\min}$, then circuit length is increased till the minimum aspect ratio is satisfied; this is done mathematically by replacing $L(\mathbf{p})$ with

$$\hat{L}(\mathbf{p}) = \max(L(\mathbf{p}), W(\mathbf{p}) \cdot As_{\min}) \quad (4.31)$$

If $As(\mathbf{p}) > As_{\max}$, then circuit width is increased till the minimum aspect ratio is satisfied, this is done mathematically by replacing $W(\mathbf{p})$ with

$$\hat{W}(\mathbf{p}) = \max\left(W(\mathbf{p}), \frac{L(\mathbf{p})}{As_{\max}}\right) \quad (4.32)$$

After adjustment of width and length, the modified area is set to the larger of A_{\max} and $\hat{W}(\mathbf{p}) \cdot \hat{L}(\mathbf{p})$:

$$\begin{aligned} \hat{A}(\mathbf{p}) &= \max(\hat{W}(\mathbf{p}) \cdot \hat{L}(\mathbf{p}), A_{\max}) \\ &\stackrel{(4.32), (4.31)}{=} \max\left(A(\mathbf{p}), \frac{L^2(\mathbf{p})}{As_{\max}}, W^2(\mathbf{p}) \cdot As_{\min}, A_{\max}\right) \\ &\stackrel{(4.25)}{=} \max\left(A(\mathbf{p}), \frac{A(\mathbf{p}) \cdot As(\mathbf{p})}{As_{\max}}, \frac{A(\mathbf{p}) \cdot As_{\min}}{As(\mathbf{p})}, A_{\max}\right) \end{aligned} \quad (4.33)$$

There is an equivalence between the original specifications in (4.27) and \hat{A} as given by (4.33) such that:

$$\begin{aligned} (A(\mathbf{p}) \leq A_{\max} \wedge As_{\min} \leq As(\mathbf{p}) \leq As_{\max}) &\iff (\hat{A}(\mathbf{p}) = A_{\max}) \\ \neg(A(\mathbf{p}) \leq A_{\max} \wedge As_{\min} \leq As(\mathbf{p}) \leq As_{\max}) &\iff (\hat{A}(\mathbf{p}) > A_{\max}) \end{aligned} \quad (4.34)$$

In Figure 4.9(b), the value of \hat{A} calculated by (4.33) is plotted for the placements and geometric specifications illustrated in the example of Figure 4.8(b).

4.3.5 Ordering and Curtailing of Circuit Placements

If the geometric specifications are given by (4.26) or (4.27), then the Pareto-optimal set of placements, \mathcal{P} is ordered according to the value of \hat{A} given in (4.29) or (4.33) respectively:

$$\mathbf{P} = \text{ordered elements of } \mathcal{P} \text{ such that } \hat{A}(\mathbf{P}[i]) \leq \hat{A}(\mathbf{P}[i+1]), \quad i=1, \dots, |\mathcal{P}|-1 \quad (4.35)$$

Since all placements $\mathbf{p} \in \mathcal{P}$ for which $\hat{A}(\mathbf{p}) = A_{\max}$ satisfy the geometric specifications, an additional measure must be applied to ensure strict total order amongst placements, for example circuit area, A , can be used:

$$\begin{aligned} &\mathbf{P} = \text{ordered elements of } \mathcal{P} \text{ such that} \\ &A_{\max} < \hat{A}(\mathbf{P}[i]) \leq \hat{A}(\mathbf{P}[i+1]) \text{ or } \left\{ \begin{array}{l} A_{\max} = \hat{A}(\mathbf{P}[i]) = \hat{A}(\mathbf{P}[i+1]) \\ \wedge A(\mathbf{P}[i]) \leq A(\mathbf{P}[i+1]) \end{array} \right\} \\ &\text{with } i=1, \dots, |\mathcal{P}|-1 \end{aligned} \quad (4.36)$$

If the device layout variants and placement constraints are well constructed, then the list $|\mathbf{P}|$ can be large. The computational cost of the subsequent synthesis steps – circuit routing, extraction of an electrical model, electrical performance evaluation, and final layout selection – is proportional to the number of placements, $|\mathbf{P}|$, that are considered. Furthermore, the value of \hat{A} , indicating the degree to which placements meet the geometric placement constraints may increase quickly, as demonstrated by the examples in Figure 4.9. In consideration of these three points, the list of placements, \mathbf{P} , is truncated before passing on to circuit routing.

Two control parameters are set to control the truncation of \mathbf{P} . The first parameter, m , denotes the maximum number of placements to consider during synthesis; list \mathbf{P} is truncated to a maximum length of m . The second parameter, ϵ , is a small fraction; list \mathbf{P} is truncated so that all elements $\mathbf{p} \in \mathbf{P}$ satisfy $\hat{A}(\mathbf{p}) \leq (1 + \epsilon) \cdot \hat{A}(\mathbf{P}[1])$. Since \mathbf{P} is ordered according to increasing values of \hat{A} and A , as established by (4.36), the truncation of list \mathbf{P} can be performed as follows:

$$\begin{aligned} &\mathbf{P} \leftarrow [\mathbf{P}[1], \mathbf{P}[2], \dots, \mathbf{P}[\min(|\mathbf{P}|, m, n)]] \text{ such that} \\ &\forall_{n < i \leq |\mathcal{P}|} (1 + \epsilon) \cdot \hat{A}(\mathbf{P}[1]) < \hat{A}(\mathbf{P}[i]); \quad m, \epsilon \text{ are predefined constants} \end{aligned} \quad (4.37)$$

For example, if the 17 placements illustrated in Figure 4.8(b) are labeled p^1 to p^{17} from the left to the right of the figure, then the ordered vector of placements according to (4.36) is $\mathbf{P} = [p^{10}, p^9, p^8, p^7, p^{11}, p^6, p^5, p^{12}, p^4, p^{13}, p^3, p^{14}, p^{15}, p^{16}, p^2, p^1, p^{17}]$. The ordered placements are plotted in Figure 4.10. If $m = 10$ and $\epsilon = 0.02$, then according to (4.37) $n = 4$ and the list of placements, \mathbf{P} , is truncated to four elements.

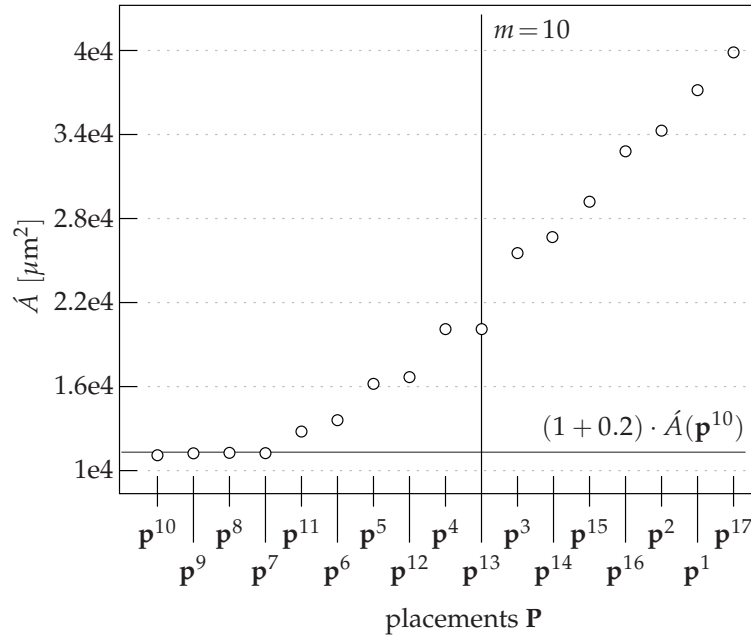


Figure 4.10: The placements in Figure 4.8(b) are ordered according to (4.36); the first four placements, $\{p^{10}, p^9, p^8, p^7\}$, are considered according to (4.37).

The subsequent steps in the layout synthesis flow are computationally costly. Conversely, these steps can be performed in parallel for the elements of \mathbf{P} . Truncation may also discard viable placements. The designer must balance these considerations when assigning values to m and ϵ .

4.4 Circuit Routing

For circuit topology \mathcal{T} with devices $\mathcal{E} = \{\delta_1, \delta_2, \dots, \delta_{|\mathcal{E}|}\}$, and possible layout variants for each device, $\{\mathcal{V}_{\delta_1}, \mathcal{V}_{\delta_2}, \dots, \mathcal{V}_{\delta_{|\mathcal{E}|}}\}$, a set of Pareto-optimal placements is generated then ordered and truncated in a vector, \mathbf{P} , according to how well the geometric specifications are met as discussed in Section 4.3.

The next step in the flow of Figure 4.1 is the routing of device connections, as well as the connections to the circuit pins for each of the considered placements in \mathbf{P} .

Geometric routing constraints are set up for the circuit as discussed in Section 3.4. General constraints are set up to meet the technology layout rules and to address robustness and reliability issues, while constraints specific to a circuit topology are set up to improve circuit matching and electrical behavior post-layout synthesis.

The industrial tool [Cad03a] is used to perform circuit routing. The underlying routing algorithm is a constraint-driven shape-based router. This router is sufficiently fast for small analog circuit topologies with less than 100 vertices and a large number of custom geometric routing constraints.

Two issues associated with routing are pin assignment and the problem of routing congestion. They are handled as described in the following two subsections.

A third issue is the satisfaction of the circuit electrical sizing rules after routing. A method is introduced whereby the circuit electrical constraints, $\mathbf{c}_e \succeq \mathbf{c}_e^m$, are indirectly transformed into geometric constraints on routing resistance. Section 4.5. is dedicated to a description of this method.

4.4.1 Pin Assignment

Circuit pin location affects internal circuit routing, as well as the chip floor plan and and system-level (global) routing. Three scenarios for pin assignment are considered here, one of which must be selected by the designer before layout synthesis is begun:

- Pin assignment is performed prior to circuit layout synthesis. This is typical in top-down design, such as in [KWY96]. In this case instructions are given for the permissible shape and location of each pin on the circuit layout boundary.

Without loss of generality, an example of this scenario is illustrated in Figure 4.11. The maximum dimensions of the circuit are set in the floorplan, and the placement specifications follow (4.26). Pin locations have also been assigned during global system routing: power and ground pins trunks are laid along the left and right of the circuit, the location of pin- i for $i = 1, \dots, 4$, is $[x_i \cdot W_{\max}, y_i \cdot L_{\max}]$ relative to the lower left corner of the circuit layout.

If the circuit does not meet the geometric specifications, such that $W > W_{\max}$ or $L > L_{\max}$, then the pin locations are rescaled according to the dimensions of the circuit:

$$\begin{aligned} \text{geometric specifications: } W \leq W_{\max} \wedge L \leq L_{\max} \\ \implies \\ \text{location of pin-}i = [x_i \cdot \max(W_{\max}, W), y_i \cdot \max(L_{\max}, L)] \end{aligned} \quad (4.38)$$

- Pin assignment is performed independent of circuit layout synthesis, and pin location is unknown at time of circuit synthesis. In this case only the internal circuit connections are routed; no pin assignment or routing is performed during circuit synthesis layout.

Example routing for this scenario is illustrated in Figure 4.12(a). An algorithm such as [ZS02] is used for subsequent pin assignment and to complete both the internal circuit and the global system-level routing.

- Pin assignment can be performed freely during circuit layout synthesis. In this case the pins are placed during the routing procedure to best fulfill the routing constraints, and ensure routing symmetry.

The outcome of pin assignment in this scenario would be similar to the example in Figure 4.12(b). In this example only the location of the power and ground trunks is set prior to layout synthesis; pin-1 through pin-4 are placed freely by the routing procedure.

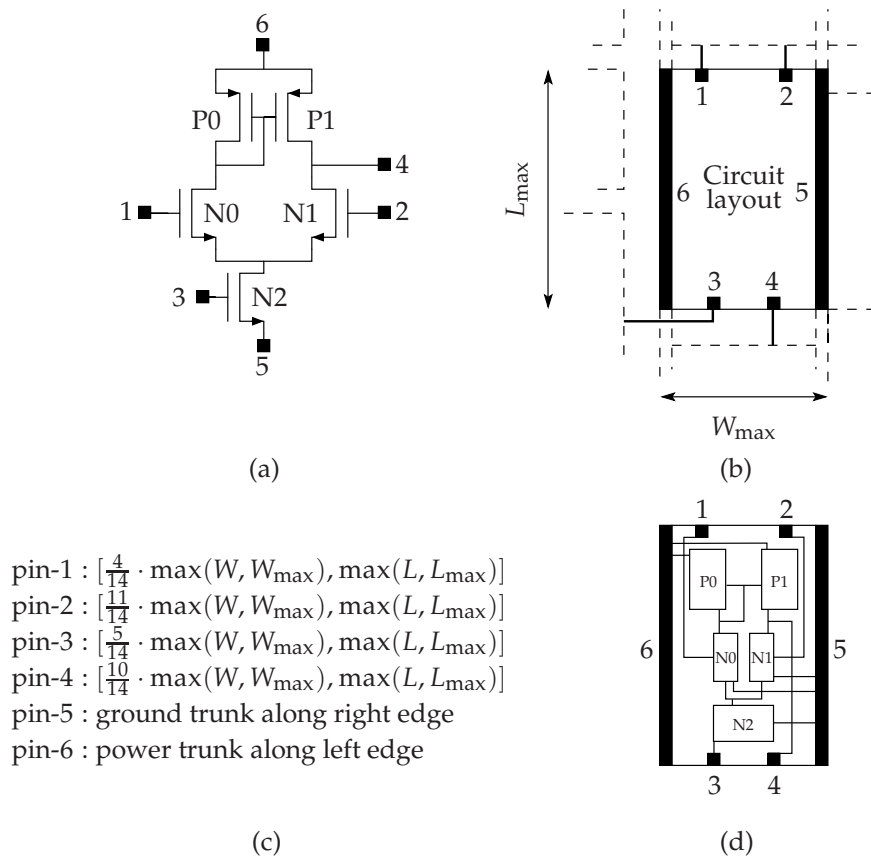


Figure 4.11: (a) Example circuit topology. (b) Circuit dimensions and pin locations are set during floorplanning. (c) Pin locations relative to lower left corner of the circuit layout. (d) Routing performed according to the fixed pin locations.

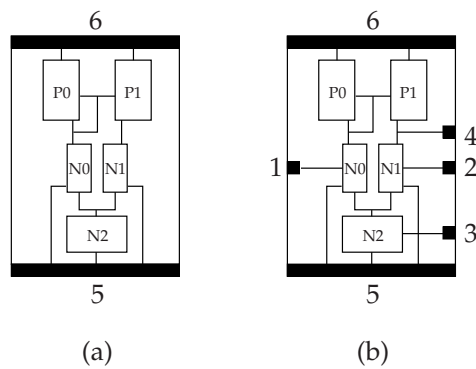


Figure 4.12: (a) pin-1 through pin-4 are added and connected externally after layout synthesis. (b) Pin assignment performed in conjunction with routing.

4.4.2 Congestion Control

If all device terminals are reachable (unblocked) and the routing constraints are consistent, then the probable reason for the failure to complete circuit routing is routing congestion [Sax07]. Barring failure to find a feasible routing solution, congestion will degrade electrical circuit performances after layout synthesis.

Congestion occurs when the minimum margins between devices, illustrated in Figure 4.6, are too small to properly fit circuit routing in the same layout along with the placed devices.

There are many methods of congestion estimation in literature [Sax07, SYL09]. Here, fast grid-based maze routing is applied with relaxed routing constraints to quickly estimate congestion, after which congested placements are adjusted.

Congestion estimation: The grid-based maze routing algorithm in [Cad03a] is applied to the circuit layout. Evenly spaced tracks, called *grids*, are superimposed horizontally and vertically over the complete layout area. The intersection of a vertical and horizontal track is called a *grid point*. Any routing operation that is performed must take into account all the grid points in the complete layout area. Maze routing is then performed considering only connectivity; technology layout rules; wire width, length, and separation; geometric symmetry; and path resistance.

Let $M[i, j]$ denote the width of the margin between adjacent devices i and j in a placement. The width of each margin is divided into tracks of pitch α . The value of α is process and layer dependent; for example, the value of α can be set so that a wire of minimum width can be laid along a track without breaking any clearance rules. The number of unused tracks between adjacent devices i and j is denoted by $T[i, j]$. If there are not enough tracks to complete routing, then $M[i, j]$ is congested and $T[i, j] < 0$. To add a safety margin, the minimum number of unused tracks is set to a small positive value, T_{min} , and $M[i, j]$ is *nearly* congested if $T[i, j] < T_{min}$; for example, $T_{min} = 2$

Placement adjustment: If the margin between adjacent devices i and j is congested, then $M[i, j]$ is increased to $\hat{M}[i, j]$ according to (4.39):

$$T[i, j] < T_{min} \Rightarrow \hat{M}[i, j] = M[i, j] + \alpha \cdot (T_{min} - T[i, j]) \quad (4.39)$$

For placement, \mathbf{p} , with devices $\mathcal{E} = \{\delta_1, \dots, \delta_{|\mathcal{E}|}\}$ and corresponding layout parameters $\{\lambda_{\delta_1}, \dots, \lambda_{\delta_{|\mathcal{E}|}}\}$, the margins between adjacent devices are checked for congestion. If congestion is found, then the margins are adjusted and circuit placement is repeated to produce an adjusted placement, $\hat{\mathbf{p}}$. Repeated placement with adjusted margins is cheap in comparison to the original placement generation process, since there are no device variants or placement possibilities to enumerate and the relative location of the devices is already fixed in the placement.

Algorithm-3 details congestion control and placement adjustment in pseudo code.

Algorithm-3 placement-with-congestion-control

```
/1/ input: placement,  $\mathbf{p}$  with a set of margin values  $M[i, j]$ 
      for each pair  $[i, j]$  of adjacent devices in  $\mathbf{p}$ 
/2/ output: adjusted placement,  $\hat{\mathbf{p}}$ 

/3/ adjust-placement  $\leftarrow$  FALSE (the placement does not need adjustment)

/4/ Perform grid-based maze routing using the algorithm in [Cad03a]

      (check the placement margins and perform margin adjustment if necessary)
      for each pair  $[i, j]$  of adjacent devices in  $\mathbf{p}$  do
/5/   Get the number of unused tracks,  $T[i, j]$ 
/6/   if  $T[i, j] < T_{min}$  then
/7/      $\hat{M}[i, j] \leftarrow M[i, j] + \alpha \cdot (T_{min} - T[i, j])$ 
/8/     adjust-placement  $\leftarrow$  TRUE (the placement needs adjustment)

      if adjust-placement = TRUE then
/9/   repeat circuit placement with adjusted margins to get  $\hat{\mathbf{p}}$ 
      else
/10/  accept placement  $\mathbf{p}$  as is ( $\hat{\mathbf{p}} = \mathbf{p}$ )
      return
```

4.5 Post-Layout Satisfaction of Electrical Sizing Rules by Limiting Routing Resistance

The electrical sizing rules for CMOS functional blocks are reviewed in Section 2.1.5; they depend on the circuit DC bias point, as formulated in (2.19). As discussed in Section 4.2, the folding of a single CMOS device into multiple fingers changes the parasitic gate, drain, and source resistance and alters the DC drain current of each finger. The wire interconnects that route device terminals contribute an additional parasitic resistance. As a result of these changes, the electrical sizing rules may be violated post-layout

Electrical sizing rules are applied because they help ensure the functionality and robustness of the circuit, it would be favorable if they are still satisfied post-layout synthesis without the need to change the value of the design parameters, \mathbf{d} .

In this section, a method will be presented to attempt and rectify electrical sizing rules that are violated post-layout, by setting an upper bound on the routing resistance between device terminals. This method is computationally cheap to implement because of two facts: Firstly, the gate, active area, and interconnect resistance values are cheap to extract post-layout in comparison to complete parasitic device extraction (including inductance and capacitance), as discussed in Section 3.5. Secondly, the circuit DC bias point is cheap to compute in comparison to other types of analysis, such as AC or transient analysis.

To implement the new rectification method, post-layout electrical sizing rules must be defined, after which the algorithm to set a boundary on routing resistance can be explained.

4.5.1 Post-Layout Electrical Sizing Rules

Each device, $\delta \in \mathcal{E}$, in the original circuit topology, \mathcal{T} , is represented by one or more devices in the placement, while topology vertices, \mathcal{V} , are replaced by resistor networks for post-layout DC analysis. Each resistor network is a connected undirected graph.

There is no one-to-one correspondence between the vertices and edges of the pre and post layout circuit topologies. The electrical sizing rules of analog functional blocks, addressed in [MGS08], need to be modified, so that they can be applied to the post placement circuit.

An example is given in Figure 4.13(a). Two NMOS devices, N1 and N2, form a differential pair; they are connected to the remainder of the circuit topology by vertices $\{A, B, C, D, E\} \subseteq \mathcal{V}$. To improve matching, N1 and N2 are laid out in a common centroid configuration, as shown in Figure 4.13(b); the number of device divisions is

$M=2$ and the number of fingers in each division is $n_f=2$. The circuit topology of Figure 4.13(c) is extracted from the layout for post-layout DC analysis. Each individual NMOS finger in the layout is represented by an intrinsic NMOS device model in the post-layout circuit topology; N1 is represented by devices $\{N1-1, N1-2, N1-3, N1-4\}$, and N2 is represented by devices $\{N2-1, N2-2, N2-3, N2-4\}$. Gate, source, drain, and interconnect resistance are represented by a resistor network corresponding to each of the original topology vertices, $\{A, B, C, D, E\}$. Bulk tap connections have been omitted from the figure to simplify the illustration.

The original electrical sizing rules of an NMOS differential pair are reviewed in Table 2.2. New modified rules are given in Table 4.4. To create new post-layout electrical

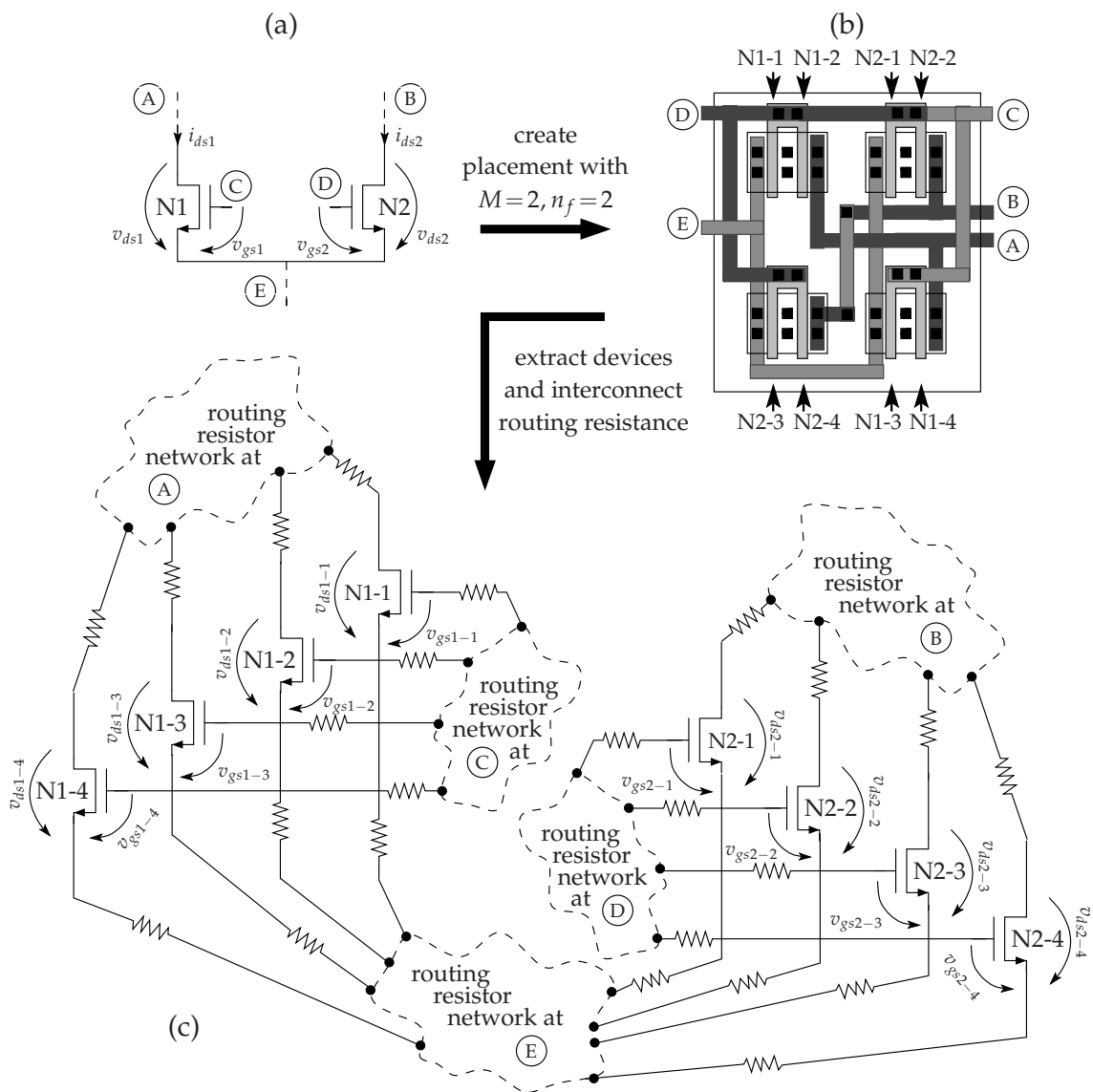


Figure 4.13: (a) An NMOS differential pair. (b) A common centroid layout configuration. (c) Post-layout topology of the differential pair for DC analysis.

Table 4.4: Post-layout electrical sizing rules of an NMOS differential pair

Electrical sizing rules	
for $i, j = 1, \dots, (M \cdot n_f)$	
/1/	$ V_{ds2-j} - V_{ds1-i} \leq \hat{V}^{(1)}$
/2/	$ V_{gs2-j} - V_{gs1-i} \leq \hat{V}^{(2)}$
for $i = 1, \dots, (M \cdot n_f)$	
/3/	$V_{ds1-i} - V_{gs1-i} + V_{th1-i} \geq \hat{V}^{(3)}$
/4/	$V_{ds1-i} \geq \hat{V}^{(4)}$
/5/	$V_{gs1-i} - V_{th1-i} \geq \hat{V}^{(5)}$
/6/	$V_{ds2-i} - V_{gs2-i} + V_{th2-i} \geq \hat{V}^{(3)}$
/7/	$V_{ds2-i} \geq \hat{V}^{(4)}$
/8/	$V_{gs2-i} - V_{th2-i} \geq \hat{V}^{(5)}$

$\hat{V}^{(1)}$ to $\hat{V}^{(5)} \in \mathbb{R}^+$ are electrical margins,
 M is the number of device divisions,
 n_f is the number of fingers in each division.

sizing rules, each original electrical inequality constraint is applied to the electrical model of every intrinsic NMOS device extracted from the layout. For instance, constraint /11/ in Table 2.2 is to ensure that device N1 is in the saturation region of operation: $V_{ds1} - V_{gs1} + V_{th1} \geq V^{(3)}$. The corresponding post-layout constraint in Table 4.4 is constraint /3/, whereby each intrinsic device of N1 is tested.

In the example of Figure 4.13, N1 is represented by four NMOS devices and the post-layout saturation constraints are: $V_{ds1-i} - V_{gs1-i} + V_{th1-i} \geq \hat{V}^{(3)}$, with $i \in [1, 2, 3, 4]$.

The post-layout electrical margins, $\hat{V}^{(1)}$ to $\hat{V}^{(5)}$ in Table 4.4, can be reduced by a certain amount of the corresponding pre-layout margins, $V^{(1)}$ to $V^{(5)}$, defined in Table 2.2. This is because a portion of the original margin values, denoted by $V_{\text{routing-margin}}^{(1)}$ to $V_{\text{routing-margin}}^{(5)}$, was to hedge DC electrical constraints against the effect of parasitic routing resistance:

$$\underbrace{V^{(\kappa)}}_{\text{pre-layout margin}} = \underbrace{\hat{V}^{(\kappa)}}_{\text{post-layout margin}} + \underbrace{V_{\text{routing-margin}}^{(\kappa)}}_{\text{discarded after layout}}; \quad \kappa \in \{1, 2, 3, 4, 5\} \quad (4.40)$$

Post-layout electrical sizing rules can be created in a similar manner for other functional blocks, such as current mirrors, level shifters, as well as current mirror and level shifter banks.

Let $\hat{\mathbf{c}}_e$ denote the vector collecting and ordering all the post-layout circuit DC electrical constraint values, such that $n_{\hat{\mathbf{c}}_e} = |\hat{\mathbf{c}}_e|$; and let $\hat{\mathbf{c}}_e^m$ denote the corresponding vector of electrical margins:

$$\mathbb{R}^{nd} \longrightarrow \mathbb{R}^{n_{\hat{\mathbf{c}}_e}} : \mathbf{d} \xrightarrow{\text{layout \& DC analysis}} \hat{\mathbf{c}}_e; \quad \hat{\mathbf{c}}_e \succeq \hat{\mathbf{c}}_e^m \quad (4.41)$$

Let $\mathbf{c}_{e,\text{routing-margin}}^m$ be the margin to hedge against parasitic layout resistance; the relationship between pre and post layout constraint margins can then be written:

$$\mathbf{c}_e^m = \hat{\mathbf{c}}_e^m + \mathbf{c}_{e,\text{routing-margin}}^m \quad (4.42)$$

4.5.2 Routing Limits to Satisfy Post-Layout Electrical Constraints

For any specific placement, $\mathbf{p} \in \mathbf{P}$, if the pre-layout electrical constraints are satisfied, then the goal is to ensure the post-layout electrical constraints are also satisfied:

$$(\mathbf{c}_e \succeq \mathbf{c}_e^m) \implies (\hat{\mathbf{c}}_e \succeq \hat{\mathbf{c}}_e^m) \quad (4.43)$$

An attempt to satisfy (4.43) is made by adjusting the routing resistor networks: Firstly, the *effective resistance* between connected device terminals is defined. Secondly, boundary constraints are placed on the effective resistance values. Thirdly, the post-layout electrical constraints are parametrized as a function of effective resistance. Finally, a novel algorithm is presented to satisfy (4.43) by adjusting the boundary constraints placed on effective resistance during circuit routing. This last step is described in Section 4.5.3.

Abstract representation of routing resistor networks

Some definitions are first in order with regards to the structure of routing resistor networks, these definitions are made with the aid of the circuit example in Figure 4.14.

Each vertex, $v \in \mathcal{V}$, of the original (pre-layout) circuit topology, \mathcal{T} , is a connected undirected graph. It is assumed here that the graphs are simple with no self loops at a vertex or multiple edges between vertices. In an electrical impedance network, shunt loops can be readily discarded, while a parallel to series impedance transformation can be used to reduce multiple edges between two vertices to a single edge. Electrically, stimulus can be applied at the subset of the graph vertices connected to circuit devices, such as CMOS transistors; this subset will be called the terminal vertices or *terminals*; the remainder of the vertices, referred to here as *internal vertices*, are not externally stimulated (DC floating nodes).

Let \mathbf{T}_v and \mathbf{U}_v denote the ordered vectors of terminal and internal vertices respectively; the complete set of vertices is ordered as $\mathbf{V}_v^T = [\mathbf{T}_v^T; \mathbf{U}_v^T]$. Let \mathbf{E}_v denote the

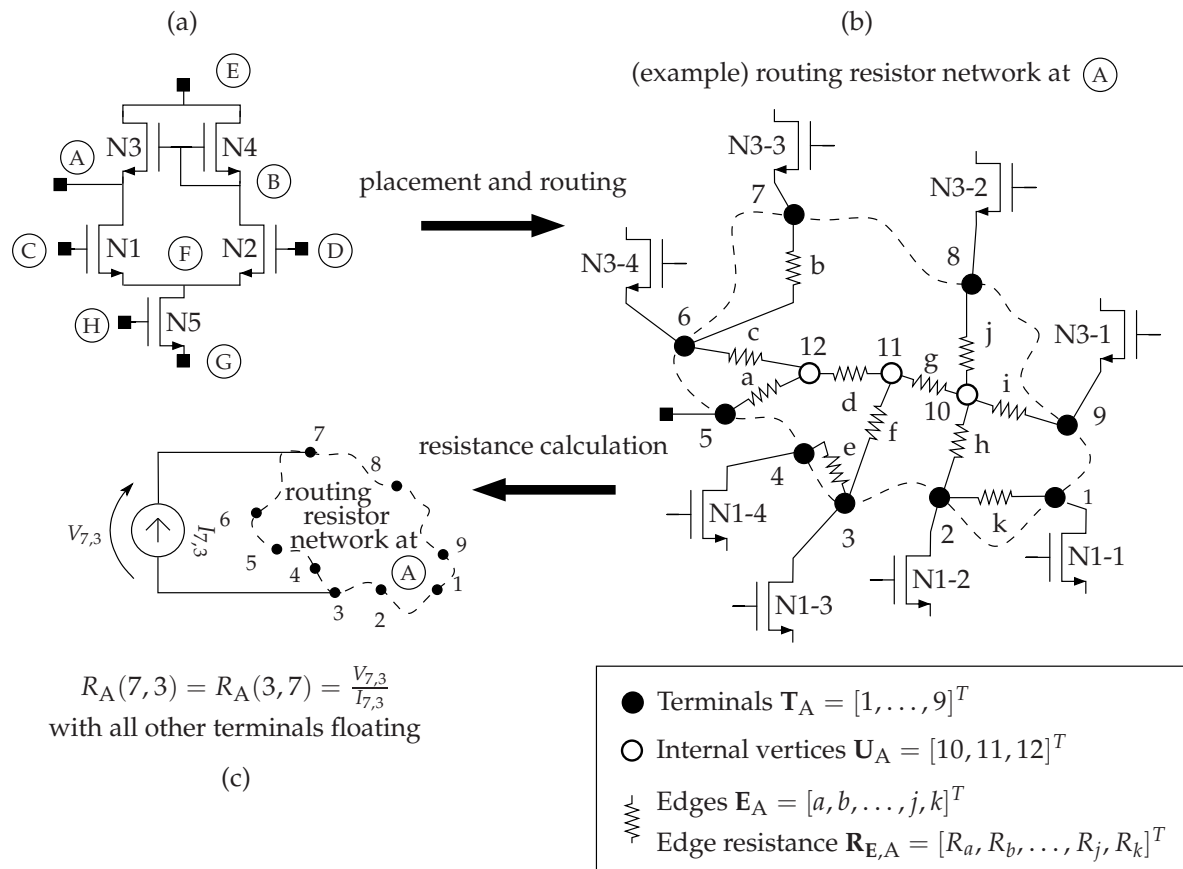


Figure 4.14: (a) Simple gain stage circuit with seven vertices, A to G. (b) An example post-layout routing resistor network corresponding to original vertex A is drawn; $G_A = G([\mathbf{T}_A^T; \mathbf{U}_A^T]^T, \mathbf{E}_A)$. (c) The effective resistance, $R_A(7,3)$, between terminals 3 and 7 of network A is calculated.

ordered vector of edges. Let G_v be the graph of the routing resistor network corresponding to original vertex v , such that $G_v = G([\mathbf{T}_v^T; \mathbf{U}_v^T]^T, \mathbf{E}_v)$.

Each graph edge is associated with a resistance value; let $\mathbf{R}_{\mathbf{E},v}$ be the vector ordering the edge resistances, such that $\mathbf{R}_{\mathbf{E},v}[i]$ is the resistance of edge $\mathbf{E}_v[i]$, for $i = 1, \dots, |\mathbf{E}_v|$.

In Figure 4.14(b), the terminal vertices corresponding to original vertex A are labeled 1 to 9 and ordered in \mathbf{T}_A , the internal vertices are labeled 10 to 12 and are ordered in \mathbf{U}_A , while the edges are labeled alphabetically from a to k and are ordered in \mathbf{E}_A . In this example, $|\mathbf{T}_A| = 9$, $|\mathbf{U}_A| = 3$, $|\mathbf{V}_A| = |\mathbf{T}_A| + |\mathbf{U}_A| = 12$, and $|\mathbf{E}_A| = 11$.

The graph of each routing resistor network is simply connected, therefore every terminal vertex in \mathbf{T}_v is connected by at least one edge:

$$\text{degree}(\mathbf{T}_v[i]) \geq 1; \quad i = 1, \dots, |\mathbf{T}_v| \quad (4.44)$$

where $\text{degree}(x)$ denotes the number of edges connected to vertex x in a graph.

Each internal vertex in \mathbf{U}_v is assumed to be connected by at least three edges. If an internal vertex is connected to a single edge, then it is floating and does not contribute electrically to the resistance network. If an internal vertex is connected by two edges, then it can be eliminated from the graph by resistor series combination:

$$\text{degree}(\mathbf{U}_v[i]) \geq 3; \quad i = 1, \dots, |\mathbf{U}_v| \quad (4.45)$$

Let \mathbf{L}_v be the admittance (weighted Laplacian) matrix of routing resistor network v , as used in nodal analysis; \mathbf{L}_v is a function of $\mathbf{R}_{\mathbf{E},v}$. Since each routing resistor network is assumed to be a simple graph, \mathbf{L}_v can be written as follows:

$$i, j = 1, \dots, |\mathbf{V}_v|; \quad \mathbf{L}_v[i, j] = \begin{cases} -\frac{1}{\mathbf{R}_{\mathbf{E},v}[k]} & \text{if } i \neq j \wedge \mathbf{E}_v[k] \text{ connects } i, j \\ -\sum_{j \neq i} \mathbf{L}_v[i, j] & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (4.46)$$

Matrix \mathbf{L}_v is irreducible symmetric and positive semi-definite with dimensions $|\mathbf{V}_v| \times |\mathbf{V}_v|$ and rank $|\mathbf{V}_v| - 1$. The smallest eigenvalue of \mathbf{L}_v is 0, the corresponding eigenvector is $\mathbf{1}_{|\mathbf{V}_v|}$, where $\mathbf{1} \in \{1\}^{|\mathbf{V}_v|}$.

In the definition of equation (4.46) the rows and columns of \mathbf{L}_v correspond to the elements of \mathbf{V}_v . Let $\mathbf{j}(\mathbf{V}_v)$ be the vector of currents injected into the vertices, and let $\mathbf{p}(\mathbf{V}_v)$ denote the corresponding vector of potentials:

$$\mathbf{j}(\mathbf{V}_v) = \mathbf{L}_v \cdot \mathbf{p}(\mathbf{V}_v) \quad (4.47)$$

By observing that $\mathbf{j}^T(\mathbf{V}_v) = [\mathbf{j}^T(\mathbf{T}_v); \mathbf{j}^T(\mathbf{U}_v)]$, $\mathbf{p}^T(\mathbf{V}_v) = [\mathbf{p}^T(\mathbf{T}_v); \mathbf{p}^T(\mathbf{U}_v)]$, and $\mathbf{j}(\mathbf{U}_v) = \mathbf{0}$, since no external current is injected into the internal vertices; the admittance matrix can be decomposed into four blocks, and (4.47) can be written as:

$$\begin{bmatrix} \mathbf{j}(\mathbf{T}_v) \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{v,TT} & \mathbf{L}_{v,TU} \\ \mathbf{L}_{v,TU}^T & \mathbf{L}_{v,UU} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{p}(\mathbf{T}_v) \\ \mathbf{p}(\mathbf{U}_v) \end{bmatrix} \quad (4.48)$$

The square matrix $\mathbf{L}_{v,UU}$ is positive definite and therefore invertible. An equation with reduced dimensions can be derived from (4.48):

$$\mathbf{j}(\mathbf{T}_v) = \underbrace{(\mathbf{L}_{v,TT} - \mathbf{L}_{v,TU} \cdot \mathbf{L}_{v,UU}^{-1} \cdot \mathbf{L}_{v,TU}^T)}_{\tilde{\mathbf{L}}_v} \cdot \mathbf{p}(\mathbf{T}_v) = \tilde{\mathbf{L}}_v \cdot \mathbf{p}(\mathbf{T}_v) \quad (4.49)$$

Matrix $\tilde{\mathbf{L}}_v$ is the Schur complement of $\mathbf{L}_{v,UU}$ in \mathbf{L}_v , and is also a weighted Laplacian matrix [vdS10]. Let \tilde{G}_v be the graph corresponding to $\tilde{\mathbf{L}}_v$; \tilde{G}_v is a complete graph with vertices \mathbf{T}_v . The graph reduction $G_v \rightarrow \tilde{G}_v$ is called a Kron reduction, and $\tilde{\mathbf{L}}_v$ is called the Kron-reduced matrix of \mathbf{L}_v . An iterative technique for Kron reduction based on internal vertex elimination is available [DB10].

An important result of Kron reduction is that the systems in (4.47) and (4.49) are electrically equivalent at the terminal vertices. Furthermore, two arbitrary graphs \tilde{G}_v^1 and \tilde{G}_v^2 with the same number of corresponding terminals are electrically equivalent if $\tilde{\mathbf{L}}_v^1 = \tilde{\mathbf{L}}_v^2$. This is true irrespective of the internal graph structure or edge resistances.

Effective resistance between terminals

The *effective resistance* between two terminals in a resistor network is defined as the difference in potential that appears across the terminals when a unit current source is applied between them. The effective resistance between two terminals is a measure of how close the terminals are in the network graph [KR93].

Let $R_v(p, q)$ denote the effective resistance between terminals p and q of G_v . Effective resistance is independent of order, such that $R_v(p, q) = R_v(q, p)$. In Figure 4.14, the effective resistance, $R_A(3, 7)$, between terminals 3 and 7 of network A is illustrated.

The effective resistance between any two terminals in resistor network v is non-negative if the vector of edge resistances satisfies $\mathbf{R}_{E,v} \succeq \mathbf{0}$. With respect to $\mathbf{R}_{E,v}$, the effective resistance between any two terminals is a homogeneous function of degree 1 and a concave function [GBS]; it is also a non-decreasing function [YZ08].

The effective resistance between terminals p and q of G_v can be calculated from the Moore-Penrose inverse, $\tilde{\mathbf{L}}_v^+$, of the Kron-reduced matrix, $\tilde{\mathbf{L}}_v$ [Ell11, DB10]:

$$R_v(p, q) = R_v(q, p) = \tilde{\mathbf{L}}_v^+[p, p] + \tilde{\mathbf{L}}_v^+[q, q] - 2\tilde{\mathbf{L}}_v^+[p, q] \quad (4.50)$$

It has been shown in [DB10] that $\tilde{\mathbf{L}}_v^+ \cdot \mathbf{1} = \mathbf{0}$, therefore:

$$\tilde{\mathbf{L}}_v^+[i, i] = - \sum_{j \neq i} \tilde{\mathbf{L}}_v^+[i, j]; \quad i, j = 1, \dots, |\mathbf{T}_v| \quad (4.51)$$

The Moore-Penrose inverse of the Kron-reduced matrix is symmetric, therefore:

$$\tilde{\mathbf{L}}_v^+[i, j] = \tilde{\mathbf{L}}_v^+[j, i] \quad (4.52)$$

Using (4.51) and (4.52), the effective resistance between terminals p and q of G_v can be written in terms of the off-diagonal upper-triangular elements of $\tilde{\mathbf{L}}_v^+$ only:

$$\begin{aligned} \text{for } p < q: \quad R_v(p, q) &= R_v(q, p) = -4 \cdot \tilde{\mathbf{L}}_v^+[p, q] \\ &\quad -1 \cdot \sum_{i \in \{1, \dots, p-1\}} \tilde{\mathbf{L}}_v^+[i, p] \\ &\quad -1 \cdot \sum_{i \in \{p+1, \dots, |\mathbf{T}_v|\} \setminus \{q\}} \tilde{\mathbf{L}}_v^+[p, i] \\ &\quad -1 \cdot \sum_{i \in \{1, \dots, q-1\} \setminus \{p\}} \tilde{\mathbf{L}}_v^+[i, q] \\ &\quad -1 \cdot \sum_{i \in \{q+1, \dots, |\mathbf{T}_v|\}} \tilde{\mathbf{L}}_v^+[q, i] \end{aligned} \quad (4.53)$$

Let \mathbf{R}_v denote the vector ordering the effective resistance between each two terminals of the routing network corresponding to original vertex v according to (4.54):

$$\begin{aligned}
 &/1/ \quad k \leftarrow 1 \\
 &/2/ \quad \text{for } i = 1 \text{ to } |\mathbf{T}_v| - 1 \\
 &/3/ \quad \quad \text{for } j = i + 1 \text{ to } |\mathbf{T}_v| \\
 &/4/ \quad \quad \quad \mathbf{R}_v[k] \leftarrow R_v(i, j) \\
 &/5/ \quad \quad \quad k \leftarrow k + 1
 \end{aligned} \tag{4.54}$$

The length of \mathbf{R}_v is:

$$n_{\mathbf{R}_v} = |\mathbf{R}_v| = \binom{|\mathbf{T}_v|}{2} = \frac{|\mathbf{T}_v| \cdot (|\mathbf{T}_v| - 1)}{2} \tag{4.55}$$

For example, in Figure 4.14, $n_{\mathbf{R}_A} = |\mathbf{R}_A| = 36$ and

$$\mathbf{R}_A = [R_A(1, 2), R_A(1, 3), \dots, R_A(1, 9), R_A(2, 3), R_A(2, 4), \dots, R_A(7, 8), R_A(8, 9)]$$

The two index sets $\mathcal{I}_{|\mathbf{T}_v|}$ and $\mathcal{K}_{|\mathbf{T}_v|}$ are defined as follows:

$$\mathcal{I}_{|\mathbf{T}_v|} = \{(i, j) \mid (i \in \{1, \dots, |\mathbf{T}_v| - 1\}) \wedge (j \in \{2, \dots, |\mathbf{T}_v|\}) \wedge (i < j)\} \tag{4.56}$$

$$\mathcal{K}_{|\mathbf{T}_v|} = \left\{ 1, \dots, \frac{|\mathbf{T}_v| \cdot (|\mathbf{T}_v| - 1)}{2} \right\} \tag{4.57}$$

An explicit bijective mapping, M , can be defined between $\mathcal{I}_{|\mathbf{T}_v|}$ and $\mathcal{K}_{|\mathbf{T}_v|}$:

$$M : \mathcal{I}_{|\mathbf{T}_v|} \longrightarrow \mathcal{K}_{|\mathbf{T}_v|} : (i, j) \longmapsto -\frac{i^2}{2} + (|\mathbf{T}_v| - \frac{1}{2}) \cdot i + j - |\mathbf{T}_v| \tag{4.58}$$

The mapping in (4.58) is equivalent to the iterative definition in (4.54), such that $\mathbf{R}_v[M(i, j)] \leftarrow R_v(i, j)$.

The elements of $\tilde{\mathbf{L}}_v^+$ have the unit of ohms, since the matrix is the pseudo inverse of an admittance matrix. Since $\tilde{\mathbf{L}}_v^+$ has dimensions $|\mathbf{T}_v| \times |\mathbf{T}_v|$, the number of off-diagonal upper-triangular elements in $\tilde{\mathbf{L}}_v^+$ is equal to $n_{\mathbf{R}_v}$ as given in (4.55). Let $\mathbf{R}_{\tilde{\mathbf{L}}_v^+}$ denote the vector ordering the off-diagonal upper-triangular elements of $\tilde{\mathbf{L}}_v^+$ according to the index mapping in (4.58), and let $\chi_{\tilde{\mathbf{L}}_v^+}$ be the $n_{\mathbf{R}_v} \times n_{\mathbf{R}_v}$ matrix that relates $\mathbf{R}_{\tilde{\mathbf{L}}_v^+}$ and \mathbf{R}_v according to (4.53):

$$\mathbf{R}_v = \chi_{\tilde{\mathbf{L}}_v^+} \cdot \mathbf{R}_{\tilde{\mathbf{L}}_v^+} \tag{4.59}$$

The elements of $\chi_{\tilde{\mathbf{L}}_v^+}$ can be derived using (4.53) as follows:

$$\begin{aligned}
 &\text{for all } (p, q), (i, j) \in \mathcal{I}_{|\mathbf{T}_v|}: \\
 \chi_{\tilde{\mathbf{L}}_v^+}[M(p, q), M(i, j)] &= \begin{cases} -4 & \text{if } (p, q) = (i, j) \\ 0 & \text{if } (p \neq i) \wedge (p \neq j) \wedge (q \neq i) \wedge (q \neq j) \\ -1 & \text{otherwise} \end{cases} \tag{4.60}
 \end{aligned}$$

Matrix $\chi_{\tilde{\mathbf{L}}+v}$ is a non-positive integer square matrix with constant diagonal values. It is also full rank. This can be checked by noting that the pattern of zero valued elements is different in each row of $\chi_{\tilde{\mathbf{L}}+v}$. Since $\chi_{\tilde{\mathbf{L}}+v}$ is full rank, the mapping between $\mathbf{R}_{\tilde{\mathbf{L}}+v}$ and \mathbf{R}_v is a bijection, and:

$$\mathbf{R}_{\tilde{\mathbf{L}}+v} = \chi_{\tilde{\mathbf{L}}+v}^{-1} \cdot \mathbf{R}_v \quad (4.61)$$

Using (4.51), (4.52), (4.58), (4.61), and the property $(\tilde{\mathbf{L}}_v^+)^+ = \tilde{\mathbf{L}}_v$, it is possible to construct the matrix $\tilde{\mathbf{L}}_v$ and corresponding complete graph \tilde{G}_v from a given value of \mathbf{R}_v and vice versa.

The construction of $\chi_{\tilde{\mathbf{L}}+v}$ in (4.60) is dependent only on the number of graph terminals, $|\mathbf{T}_v|$, and not on the graph structure or edge resistances. Two arbitrary graphs G_v^1 and G_v^2 with the same number of corresponding terminals have $\chi_{\tilde{\mathbf{L}}+v}^1 = \chi_{\tilde{\mathbf{L}}+v}^2$. If $\mathbf{R}_v^1 = \mathbf{R}_v^2$, then $\mathbf{R}_{\tilde{\mathbf{L}}+v}^1 = \mathbf{R}_{\tilde{\mathbf{L}}+v}^2$, and $\tilde{\mathbf{L}}_v^1 = \tilde{\mathbf{L}}_v^2$, therefore graphs G_v^1 and G_v^2 are electrically equivalent at the terminal vertices and have the same Kron-reduced graph \tilde{G}_v .

This result is important, because it means that a resistor network can be represented electrically by the value of \mathbf{R}_v regardless of internal structure.

An important question to ask at this point is for what values of \mathbf{R}_v can a routing resistor network be constructed, or alternatively, what are the possible values of \mathbf{R}_v .

If a routing resistor network can be constructed for any non-negative value of \mathbf{R} , then each effective resistance value can be selected independently of all others, such that the domain of \mathbf{R}_v is the non-negative elements of $\mathbb{R}^{n_{Rv}}$. This would require that up to a complete graph can be constructed and that the edge resistances can have arbitrary non-negative real values.

However, routing resistor networks have a geometric implementation in circuit layout; the set of networks that can be implemented in a geometric layout is dependent on the routing algorithm, the circuit placement, and the geometric constraints. This means that, in practice, the set of possible values of \mathbf{R}_v is a subset of the non-negative elements of $\mathbb{R}^{n_{Rv}}$.

Setup of boundary constraints on effective resistance

Let \mathbf{R} be the vector ordering all the effective resistances $\{\mathbf{R}_v | v \in \mathcal{V}\}$, $n_{\mathbf{R}} = |\mathbf{R}|$. For the example circuit of Figure 4.14 with original vertices A to G, $\mathbf{R} = [\mathbf{R}_A^T; \mathbf{R}_B^T; \dots; \mathbf{R}_G^T]^T$.

Effective resistance is a superior measure of the influence of circuit routing on post-layout electrical behavior than simple wire length or shortest path resistance between connected terminals.

To limit the adverse effect of routing resistance on post-layout electrical behavior, an upper bound, \mathbf{R}^u , on the value of \mathbf{R} is typically set as a geometric routing constraint, while a lower bound of $\mathbf{0}$ is set so that the routing edge resistances are positive and physically realizable:

$$\mathbf{0} \preceq \mathbf{R} \preceq \mathbf{R}^u \quad (4.62)$$

For matched devices and balanced signal propagation paths, a bound may also be placed on the difference in effective resistance. For example, to match effective resistances $R_A(1, 9)$ and $R_B(1, 8)$:

$$-\Delta_R^u \preceq [1 \quad -1] \cdot \begin{bmatrix} R_A(1, 9) \\ R_B(1, 8) \end{bmatrix} \preceq \Delta_R^u; \quad \Delta_R^u \geq 0 \text{ is a small constant} \quad (4.63)$$

If there are N effective resistance pairs to be matched, then a system of matching constraints can be defined:

$$-\Delta_R^u \cdot \mathbf{1}_N \preceq \Delta_{\mathbf{R}} \cdot \mathbf{R} \preceq \Delta_R^u \cdot \mathbf{1}_N \quad (4.64)$$

where

$$\begin{aligned} &\Delta_R^u \geq 0 \text{ is a small constant; } \mathbf{1}_N \in \{1\}^N; \quad \Delta_{\mathbf{R}} \in \{-1, 0, 1\}^{N \times n_{\mathbf{R}}}; \\ &\text{for } i \in [1, N] \text{ and } j, k, r \in [1, n_{\mathbf{R}}], \\ &\text{if the } i\text{-th matched pair is } (\mathbf{R}[j], \mathbf{R}[k]) \text{ then } \begin{cases} \Delta_{\mathbf{R}}[i, j] = 1 \\ \Delta_{\mathbf{R}}[i, k] = -1 \\ \Delta_{\mathbf{R}}[i, r] = 0 \text{ for } r \neq j, k \end{cases} \end{aligned} \quad (4.65)$$

Parametrized post-layout electrical constraints

The value of \mathbf{R} will influence the DC bias point of the post-layout circuit, and, as a consequence, the value of the post-layout DC electrical constraints. To account for this, post-layout DC electrical constraints are parametrized by \mathbf{R} .

Pursuing the example of Figure 4.13, the post-layout routing-dependent saturation constraints of N1 are: $V_{ds1-i}(\mathbf{R}) - V_{gs1-i}(\mathbf{R}) + V_{th1-i}(\mathbf{R}) \geq \hat{V}^{(3)}$, with $i \in [1, 2, 3, 4]$.

If all the post-layout electrical constraints are parametrized by \mathbf{R} , then (4.43) can be rewritten as follows:

$$(\mathbf{c}_e \succeq \mathbf{c}_e^m) \implies (\hat{\mathbf{c}}_e(\mathbf{R}) \succeq \hat{\mathbf{c}}_e^m) \quad (4.66)$$

The inequality $\hat{\mathbf{c}}_e(\mathbf{R}) \succeq \hat{\mathbf{c}}_e^m$ can be added as an electrical routing constraint.

The *feasible effective resistance space* is defined to be the values of $\mathbf{R} \in \mathbb{R}^{n_{\mathbf{R}}}$ that satisfy (4.62), (4.64), and (4.66). This definition implicitly assumes that a complete graph can be constructed by the routing algorithm subject to the circuit placement and geometric constraints, and that the network edge resistances can assume arbitrary positive real values.

An illustration is given in Figure 4.15 of the effective resistance space with $n_{\hat{\mathbf{c}}_e} = 3$, $n_{\mathbf{R}} = 2$, and with effective resistances $\mathbf{R}[1]$ and $\mathbf{R}[2]$ set to be matched.

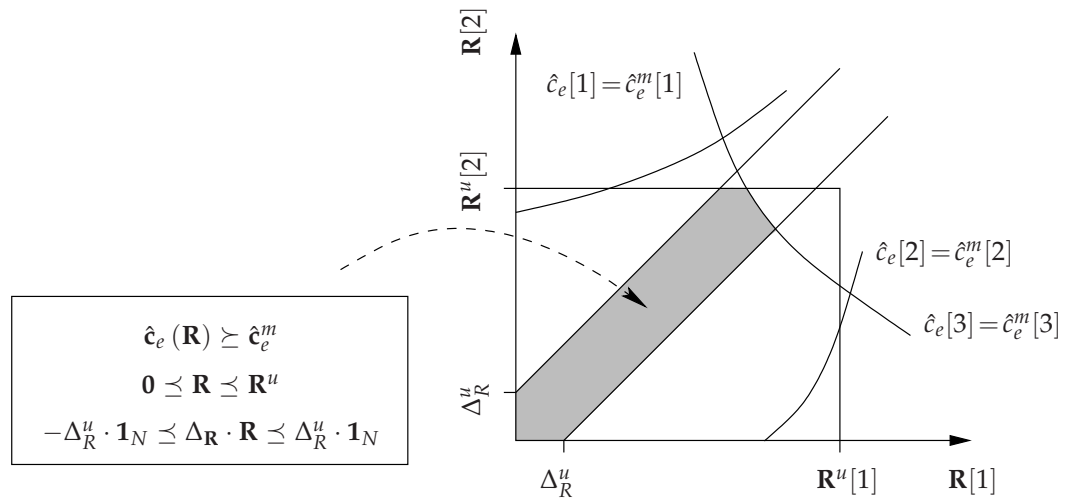


Figure 4.15: An example with one pair of matched effective resistances and three electrical constraints. The feasible effective resistance space is shaded.

Algorithm to ensure the satisfaction of (4.66) during routing

One method to satisfy (4.66) is to repeat circuit routing with ever decreasing values of \mathbf{R}^u until $\hat{\mathbf{c}}_e(\mathbf{R}) \succeq \hat{\mathbf{c}}_e^m$. The value of \mathbf{R}^u can be scaled down as demonstrated in Figure 4.16. If the routing algorithm fails to successfully complete circuit routing following each recall, then the value of $\mathbf{c}_{e,\text{routing-margin}}^m$ is increased, after which circuit sizing and layout synthesis must be repeated.

The advantage of the above method to satisfy (4.66) is that the electrical constraints need not be considered directly by the routing algorithm. There are two disadvantages: First, the routing algorithm must be recalled multiple times, increasing computational cost. Secondly, a simple scaling of vector \mathbf{R}^u may discard regions of the effective resistance space that contain valid solutions which satisfy (4.66), and the value of \mathbf{R}^u may not be maximal as illustrated in Figure 4.16.

A maximal upper bound value, \mathbf{R}^u , on routing resistance gives greater flexibility to the routing algorithm in selecting a good routing solution.

A more elaborate method to ensure (4.66) and obtain a maximal value of \mathbf{R}^u is described below. With this new method, the routing algorithm is called twice in the worst-case scenario; furthermore, the adjustments necessary to circuit routing are often minor in the second call. Algorithm-4 is an equivalent to the new method in pseudo code:

First, $\mathbf{c}_{e,\text{routing-margin}}^m$ is fixed and upper bound \mathbf{R}^u is initialized to a maximum value, $\mathbf{R}^u \leftarrow \mathbf{R}^{u0}$, based on designer experience or estimated wire length from global routing. This is handled in lines /2/ and /3/ of Algorithm-4.

Secondly, circuit routing is performed subject to $\mathbf{R} \preceq \mathbf{R}^{u0}$, (4.64), and any other geometric routing constraints defined as discussed in Section 3.4. If the routing algorithm fails to complete circuit routing successfully, then \mathbf{R}^{u0} or the other geometric

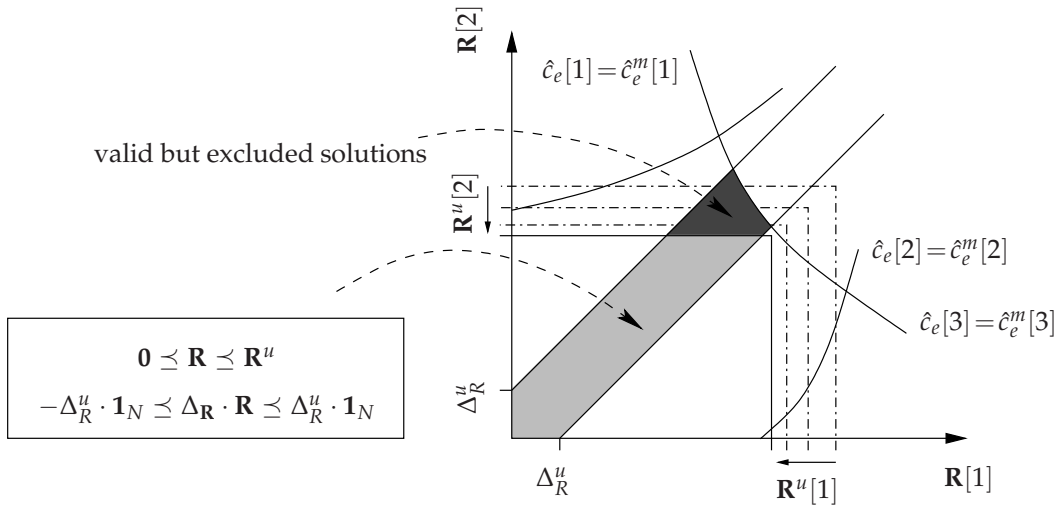


Figure 4.16: \mathbf{R}^u is rescaled till routing can be performed without post-layout electrical constraint violations. The region of valid but excluded solutions is marked.

routing constraints are too stringent to find a feasible routing solution; routing of the current placement is considered to have failed. This is handled in lines /5/ and /6/ of Algorithm-4.

Thirdly, if circuit routing is successful, then the post-layout DC circuit is extracted. The value of \mathbf{R} is then calculated from the routing resistor networks, this value is labeled \mathbf{R}^0 . The value of the post-layout electrical constraint function, $\hat{c}_e(\mathbf{R}^0)$, is calculated by DC simulation. If $\hat{c}_e(\mathbf{R}^0) \succeq \hat{c}_e^m$, then (4.66) is satisfied and the routing solution is accepted. This is handled in lines /7/ to /10/ of Algorithm-4.

Fourthly, if $\hat{c}_e(\mathbf{R}^0) \prec \hat{c}_e^m$, then \mathbf{R}^u is adjusted to a maximal value in the feasible effective resistance space. The problem of adjusting \mathbf{R}^u is formulated as constrained optimization problem that can be solved efficiently using a local optimization algorithm, such as successive linear programming (SLP) [GS61]. This step constitutes the fundamental component of the new algorithm, and is described in detail in Sections 4.5.3, 4.5.4, and 4.5.5. If an adjusted value of \mathbf{R}^u cannot be found, then the routing of the current placement is considered to have failed and the placement is discarded. This is handled in lines /11/ and /12/ of Algorithm-4.

Fifthly, if a suitable value of \mathbf{R}^u is found, then this value is labeled \mathbf{R}^{u1} ; circuit routing is repeated subject to $\mathbf{R} \preceq \mathbf{R}^{u1}$, (4.64), and the other predefined geometric routing constraints. This is handled in line /13/ of Algorithm-4.

Sixthly, if the routing algorithm fails to complete circuit routing successfully, then the new constraint $\mathbf{R} \preceq \mathbf{R}^{u1}$ is too stringent so that there is no feasible routing solution for the current placement. Otherwise, if circuit routing is successful, then the routing solution is accepted. This is handled in lines /14/ and /15/ of Algorithm-4.

As postlude to the methodology in this section, if circuit routing fails for many placements in the set of placements, \mathbf{P} , then this is empirical evidence that the value

of $\mathbf{c}_{e,\text{routing-margin}}^m$ is too small and needs to be increased. A change in $\mathbf{c}_{e,\text{routing-margin}}^m$ will adjust the boundaries of the feasible design space, \mathcal{D} . A new design parameter vector, \mathbf{d} , can then be selected from the adjusted feasible design space, \mathcal{D} , after which the layout synthesis flow of Chapter 4 can be recalled.

Algorithm-4 detailed-routing

```

/1/ input: placement  $\mathbf{p} \in \mathbf{P}$  to be routed
/2/      $\mathbf{c}_{e,\text{routing-margin}}^m$  (margin value to hedge against parasitic layout resistance)
/3/      $\mathbf{R}^{u0}$  (initial bound on the effective resistance vector)
/4/ output: a circuit routing solution that satisfies (4.66) and all other
        geometric routing constraints as described in Section 3.4

        (routing is performed as described in Section 4.4)
/5/ call the routing algorithm with  $\mathbf{R} \preceq \mathbf{R}^{u0}$ 
/6/ if circuit routing failed then return failed

/7/ extract the post-layout circuit DC netlist from the layout
/8/ extract the effective resistance vector,  $\mathbf{R}^0$ , from the DC netlist
/9/ calculate  $\hat{\mathbf{c}}_e(\mathbf{R}^0)$  (DC simulation)

        (no electrical constraint is violated  $\Rightarrow$  the layout and routing solution is accepted)
/10/ if  $\hat{\mathbf{c}}_e(\mathbf{R}^0) \succeq \hat{\mathbf{c}}_e^m$ , then return circuit routing solution

        (if  $\hat{\mathbf{c}}_e(\mathbf{R}^0) \prec \hat{\mathbf{c}}_e^m$ , then there are electrical constraint violations)
/11/ Find an maximal upper bound,  $\mathbf{R}^{u1}$ , on  $\mathbf{R}$  as described in Section 4.5.3
/12/ if an maximal bound on  $\mathbf{R}$  is not found then return failed

        (routing is performed as described in Section 4.4)
/13/ if an maximal bound on  $\mathbf{R}$  is found then call the routing algorithm with  $\mathbf{R} \preceq \mathbf{R}^{u1}$ 

        (circuit routing will fail if the new upper bound  $\mathbf{R}^{u1}$  is too stringent)
/14/ if circuit routing failed then return failed
/15/ else return the circuit routing solution

```

4.5.3 Maximization of \mathbf{R}^u in the Feasible Effective Resistance Space

The maximization of \mathbf{R}^u in the feasible effective resistance space can be written as a nonlinear optimization problem in \mathbf{R} :

$$\begin{aligned} \mathbf{R}^{u1} = \max_{\mathbf{R} \in \mathbb{R}^{n_R}} \mathbf{R} \\ \text{subject to } \begin{cases} \hat{c}_e(\mathbf{R}) \preceq \hat{c}_e^m \\ -\Delta_R^u \cdot \mathbf{1}_N \preceq \Delta_R \cdot \mathbf{R} \preceq \Delta_R^u \cdot \mathbf{1}_N \\ \mathbf{0} \preceq \mathbf{R} \preceq \mathbf{R}^{u0} \end{cases} \end{aligned} \quad (4.67)$$

Only the post-layout electrical constraint function, $\hat{c}_e(\mathbf{R})$, is nonlinear in (4.67).

In general, the solution to (4.67), if it exists, is not unique; there is a set of maximal solutions that form a Pareto-optimal set [Par06]. This is illustrated in Figure 4.17 for a two dimensional case; the set of solutions is indicated.

If the difference $\mathbf{R}^{u0} - \mathbf{R}^{u1}$ is small, then the change in the geometric routing constraint on \mathbf{R} from $\mathbf{R} \preceq \mathbf{R}^{u0}$ to $\mathbf{R} \preceq \mathbf{R}^{u1}$ is also small. As a consequence, the necessary routing adjustments on the second call to the routing algorithm, as handled in line /13/ of Algorithm-4, are expected to be easier to make and incur less computational cost. Furthermore, relatively small changes in the bound value will be easier to adjust for in routing than relatively large changes. For example, a change from 100Ω to 110Ω in the maximum allowed resistance of a long wire connection is (typically) easier to accommodate than a change from 10Ω to 20Ω in a short connection.

In consideration of the arguments above, the solution to (4.67) that minimizes the maximum relative distance from the original bound \mathbf{R}^{u0} is preferred over other solu-

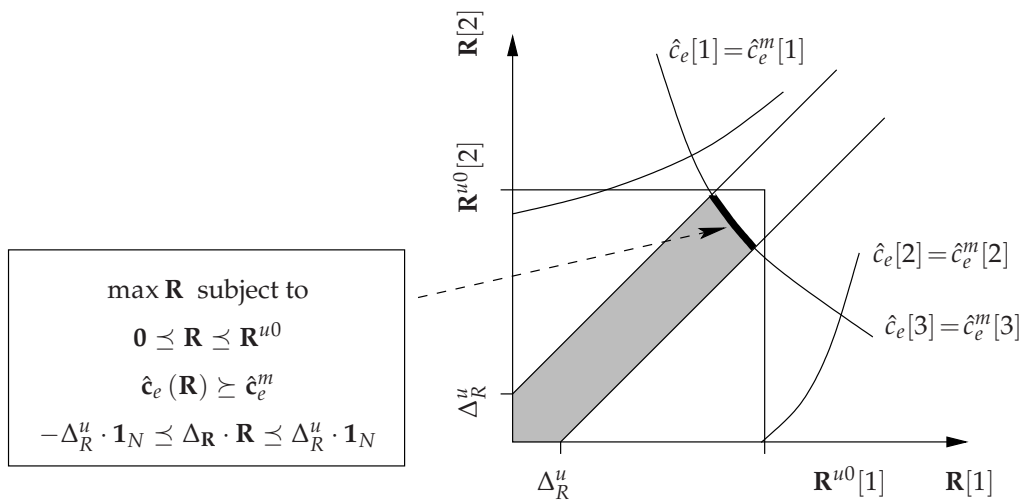


Figure 4.17: $\mathbf{R}^u \leftarrow \mathbf{R}^{u0}$, the set of vectors $\{\mathbf{R}^{u1}\}$ that satisfy (4.67) is indicated.

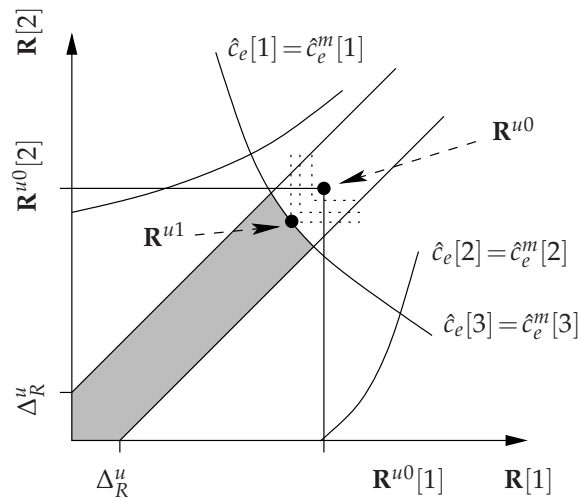


Figure 4.18: $\mathbf{R}^u \leftarrow \mathbf{R}^{u0}$, the vector \mathbf{R}^{u1} that satisfies (4.68) is indicated.

tions. This preferred solution can be obtained directly by solving the following scalar minimization problem:

$$\begin{aligned} \mathbf{R}^{u1} = \operatorname{argmin}_{\mathbf{R} \in \mathbb{R}^{n_R}} \max_{i=1, \dots, n_R} \frac{\mathbf{R}^{u0}[i] - \mathbf{R}[i]}{\mathbf{R}^{u0}[i]} \\ \text{subject to } \begin{cases} \hat{\mathbf{c}}_e(\mathbf{R}) \succeq \hat{\mathbf{c}}_e^m \\ -\Delta_R^u \cdot \mathbf{1}_N \preceq \Delta_R \cdot \mathbf{R} \preceq \Delta_R^u \cdot \mathbf{1}_N \\ \mathbf{0} \preceq \mathbf{R} \preceq \mathbf{R}^{u0} \end{cases} \end{aligned} \quad (4.68)$$

A unique solution exists to the problem in (4.68), if and only if there is at least one solution to (4.67). The solution to (4.68) is illustrated in Figure 4.18. The vector $\mathbf{R} = \mathbf{R}^0$ can be used as an initial starting point to solve (4.68).

The objective of the min-max problem in (4.68) is nonlinear and discontinuous over the problem domain \mathbb{R}^{n_R} . The problem can be rewritten in the Goal Attainment formulation [MGS07]: The objective is replaced by a new bound parameter, t , and a set of inequality constraints, called goal attainment (GA) constraints, are added:

$$\begin{aligned} \begin{bmatrix} t^* \\ \mathbf{R}^{u1} \end{bmatrix} = \operatorname{argmin}_{t, \mathbf{R} \in \mathbb{R}^{n_R}} t \\ \text{subject to } \begin{cases} \hat{\mathbf{c}}_e(\mathbf{R}) \succeq \hat{\mathbf{c}}_e^m \\ -\Delta_R^u \cdot \mathbf{1}_N \preceq \Delta_R \cdot \mathbf{R} \preceq \Delta_R^u \cdot \mathbf{1}_N \\ \mathbf{0} \preceq \mathbf{R} \preceq \mathbf{R}^{u0} \\ \mathbf{R}^{u0} \cdot (1 - t) \preceq \mathbf{R} \quad (\text{GA constraints}) \end{cases} \end{aligned} \quad (4.69)$$

The vector $\begin{bmatrix} t \\ \mathbf{R} \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{R}^0 \end{bmatrix}$ can be used as an initial starting point to solve (4.69).

An antecedent to problems (4.67), (4.68), and (4.69), is that the unconstrained domain of vector \mathbf{R} is \mathbb{R}^{n_R} . In Section 4.5.4, the domain of \mathbf{R} is restricted to the subset possible when each routing resistor network is a tree. The solution space of problem (4.69) is then restricted to this new domain.

4.5.4 Acyclic Routing Network Graphs of Maximum Edge Number

It is recalled from Section 4.5.2 that the vectors $\{\mathbf{R}_v | v \in \mathcal{V}\}$ construct vector \mathbf{R} , and that the domain of possible values of \mathbf{R} is dependent on the constraints placed on the routing resistor network graphs $\{G_v | v \in \mathcal{V}\}$.

In this section, two assumptions are made with regards to the structure of the routing resistor networks. As a consequence of these assumptions, the domain of the effective resistances, \mathbf{R} , is restricted to a subspace of \mathbb{R}^{n_R} . The complex algebraic manipulations of Section 4.5.2 are also simplified.

Assumption 1: Each of the original routing resistor graphs, $\{G_v | v \in \mathcal{V}\}$, is a tree. Basic impedance transformations, such as parallel to series combination and the delta to star transformation, can be used to remove *some* cycles from a resistor network graph and create a tree.

Under assumption 1, there is exactly one simple path between any two terminals in \mathbf{T}_v . In this case, the equivalent resistance $R_v(p, q)$ between terminals p and q of G_v can be calculated by an inner product:

$$R_v(p, q) = \chi_{v,p,q} \cdot \mathbf{R}_{E,v} \quad (4.70)$$

where $\chi_{v,p,q}$ is an indicator vector, such that:

$$\chi_{v,p,q}[i] = \begin{cases} 1 & \text{if } \mathbf{E}_v[i] \text{ is on the path between } p \text{ and } q \\ 0 & \text{otherwise} \end{cases} \quad (4.71)$$

For \mathbf{R}_v , a comprehensive indicator matrix can be defined:

$$\mathbf{R}_v = \chi_v \cdot \mathbf{R}_{E,v} \quad (4.72)$$

where χ_v is an indicator matrix, such that:

$$\chi_v[i, j] = \begin{cases} 1 & \text{if } \mathbf{R}_v[i] = R_v(p, q) \text{ and } \mathbf{E}_v[j] \text{ is on the path between } p \text{ and } q \\ 0 & \text{otherwise} \end{cases} \quad (4.73)$$

If $\mathcal{V} = \{A, B, C, \dots\}$, $\mathbf{R} = [\mathbf{R}_A^T, \mathbf{R}_B^T, \mathbf{R}_C^T, \dots]^T$, and $\mathbf{R}_E = [\mathbf{R}_{E,A}^T, \mathbf{R}_{E,B}^T, \mathbf{R}_{E,C}^T, \dots]^T$, then by (4.72):

$$\begin{aligned} \mathbf{R} &= \begin{bmatrix} \mathbf{R}_A \\ \mathbf{R}_B \\ \mathbf{R}_C \\ \vdots \end{bmatrix} = \underbrace{\begin{bmatrix} \chi_A & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \chi_B & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \chi_C & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{\chi} \cdot \underbrace{\begin{bmatrix} \mathbf{R}_{E,A} \\ \mathbf{R}_{E,B} \\ \mathbf{R}_{E,C} \\ \vdots \end{bmatrix}}_{\mathbf{R}_E} \\ &= \chi \cdot \mathbf{R}_E \end{aligned} \quad (4.74)$$

Under assumption 1, the problem of calculating the value of \mathbf{R}_v for a post-layout DC circuit is the problem of finding the simple paths between the terminal vertices, \mathbf{T}_v , so as to build the matrix χ_v . Step /8/ of Algorithm-4 can be subdivided into three sub-steps:

1. For each original vertex $v \in \mathcal{V}$, the value of vector $\mathbf{R}_{\mathbf{E},v}$, denoted by $\mathbf{R}_{\mathbf{E},v}^0$, is extracted from the post-layout netlist.
2. The value of the indicator function χ_v , denoted by χ_v^0 , is calculated by finding the simple paths in each network.
3. By (4.72), $\mathbf{R}_v^0 = \chi_v^0 \cdot \mathbf{R}_{\mathbf{E},v}^0$; \mathbf{R}^0 is then constructed from the sub-vectors.

Under assumption 1 and from (4.72), when solving (4.69), the degree of freedom in selecting the resistance vector \mathbf{R}_v (and by extension the complete vector \mathbf{R}) depends on the column rank of χ_v .

The dimensions of χ_v are $\left\lfloor \frac{|\mathbf{T}_v| \cdot (|\mathbf{T}_v| - 1)}{2} \right\rfloor \times |\mathbf{E}_v|$, since $|\mathbf{R}_v|$ is given by (4.55) and $|\mathbf{R}_{\mathbf{E},v}| = |\mathbf{E}_v|$ by definition. Maximum column rank is achieved when $|\mathbf{E}_v|$ is maximized and the column vectors are linearly independent. Lemma 4.1 below finds the tree with the maximum number of edges for a fixed number of terminal vertices \mathbf{T}_v .

Lemma 4.1. *Given a fixed vector of terminals \mathbf{T}_v , such that $|\mathbf{T}_v| \geq 1$, satisfying (4.44), and zero or more internal vertices \mathbf{U}_v satisfying (4.45), the maximum possible number of edges, $|\mathbf{E}_v|$, that a tree with vertices $[\mathbf{T}_v^T; \mathbf{U}_v^T]^T$ can have is $|\mathbf{E}_{v,max}| = 2|\mathbf{T}_v| - 3$; this is achieved when the number of internal vertices, \mathbf{U}_v , is maximized, furthermore, $|\mathbf{U}_{v,max}| = |\mathbf{T}_v| - 2$.*

According to the properties of a graph:

$$|\mathbf{E}_v| = \frac{1}{2} \left(\sum_i^{|\mathbf{T}_v|} \text{degree}(\mathbf{T}_v[i]) + \sum_i^{|\mathbf{U}_v|} \text{degree}(\mathbf{U}_v[i]) \right) \quad (4.75)$$

According to the properties of a tree:

$$|\mathbf{E}_v| = |\mathbf{T}_v| + |\mathbf{U}_v| - 1 \quad (4.76)$$

From (4.76), $|\mathbf{E}_{v,max}| = \max(|\mathbf{T}_v| + |\mathbf{U}_v| - 1)$. Since $|\mathbf{T}_v|$ is constant:

$$|\mathbf{E}_{v,max}| = |\mathbf{T}_v| + \max(|\mathbf{U}_v|) - 1 = |\mathbf{T}_v| + |\mathbf{U}_{v,max}| - 1 \quad (4.77)$$

Therefore if $|\mathbf{U}_v|$ is maximized, then $|\mathbf{E}_v|$ is also maximized. As a consequence, in equation (4.75), if $|\mathbf{E}_v| = |\mathbf{E}_{v,max}|$ in the left hand side, then $|\mathbf{U}_v| = |\mathbf{U}_{v,max}|$ in the right hand side. Since $|\mathbf{T}_v|$ is constant, to maximize $|\mathbf{U}_v|$ in the right hand side of (4.75), $\text{degree}(\mathbf{T}_v[i])$ with $i \in [1, |\mathbf{T}_v|]$ and $\text{degree}(\mathbf{U}_v[i])$ with $i \in [1, |\mathbf{U}_v|]$ must be minimized whilst satisfying (4.44) and (4.45), therefore $\text{degree}(\mathbf{T}_v[i]) = 1$ with $i \in [1, |\mathbf{T}_v|]$, and $\text{degree}(\mathbf{U}_v[i]) = 3$ with $i \in [1, |\mathbf{U}_v|]$. By substituting the latter results into (4.75), the following equation is obtained:

$$|\mathbf{E}_{v,max}| = \frac{1}{2} \sum_i^{|\mathbf{T}_v|} 1 + \frac{1}{2} \sum_i^{|\mathbf{U}_{v,max}|} 3 = \frac{|\mathbf{T}_v| + 3|\mathbf{U}_{v,max}|}{2} \quad (4.78)$$

Substituting (4.77) into (4.78):

$$|\mathbf{U}_{v,max}| = |\mathbf{T}_v| - 2 \quad (4.79)$$

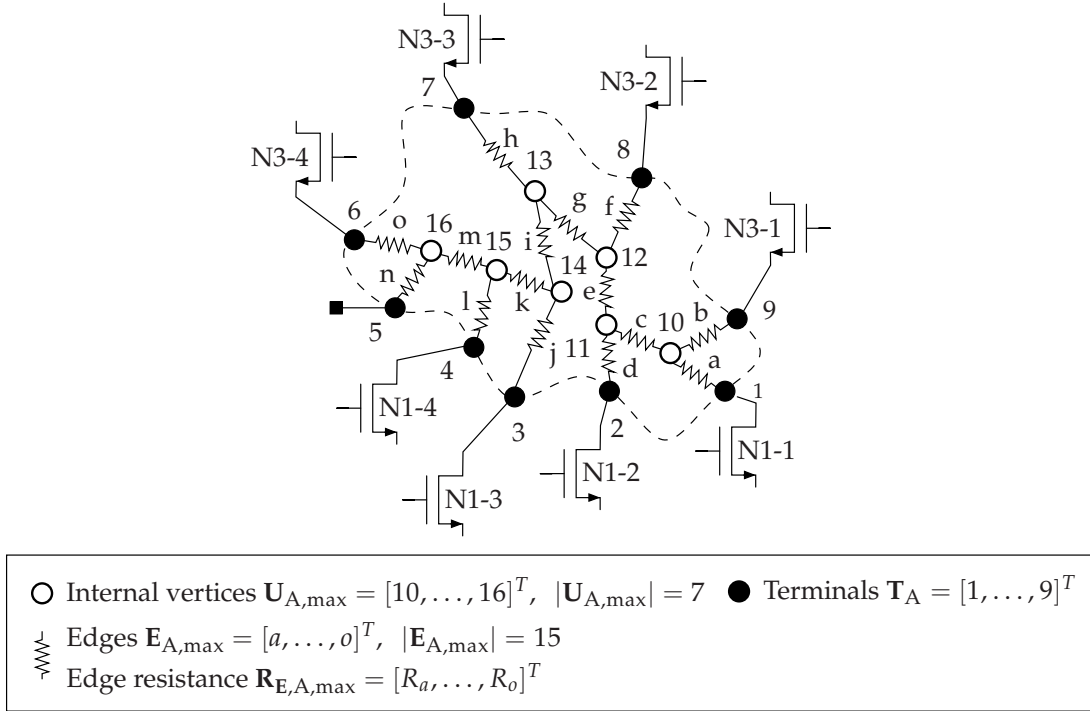


Figure 4.19: A tree with the maximum possible number of edges, $\mathbf{E}_{v,\max}$, for a fixed number of terminal vertices \mathbf{T}_v ; $|\mathbf{T}_v| = 9$ in this example.

Substituting (4.79) into (4.77):

$$|\mathbf{E}_{v,\max}| = 2|\mathbf{T}_v| - 3 \quad (4.80)$$

End of Lemma 4.1.

Let $G_{v,\max} = G([\mathbf{T}_v^T; \mathbf{U}_{v,\max}^T]^T, \mathbf{E}_{v,\max})$ be the tree with the maximum number of edges, as described in Lemma 4.1. The structure of $G_{v,\max}$ depends only on the value of $|\mathbf{T}_v|$. Categorically, $G_{v,\max}$ is the Caterpillar tree [HS73] with internal vertices $\mathbf{U}_{v,\max}$ along a central path and terminals \mathbf{T}_v as leaves. Figure 4.19 illustrates $G_{v,\max}$ for $|\mathbf{T}_v| = 9$.

Let $\chi_v = \chi_{v,\max}$ be the indicator matrix corresponding to the graph $G_{v,\max}$ according to the definition in (4.73). From (4.55) and (4.80), the dimensions of $\chi_{v,\max}$ are $\left\lfloor \frac{|\mathbf{T}_v| \cdot (|\mathbf{T}_v| - 1)}{2} \right\rfloor \times (2|\mathbf{T}_v| - 3)$. Furthermore, $\chi_{v,\max}$ has full column rank when the underlying field is \mathbb{R} . Let $\mathbf{R}_{E,v,\max}$ be the vector of edge resistances in the graph $G_{v,\max}$.

Assumption 2: Prior to solving problem (4.69), each of the original graphs, G_v , is replaced by the corresponding Caterpillar tree, $G_{v,\max}$. Algebraically, this means the domain of effective resistance vector \mathbf{R}_v (with $v \in \mathcal{V}$) is restricted to $\text{col}(\chi_{v,\max})$.

Equation (4.72) can be rewritten for the specific case when routing network v is a Caterpillar tree with indicator matrix $\chi_v = \chi_{v,\max}$ and edge resistances $\mathbf{R}_{E,v} = \mathbf{R}_{E,v,\max}$:

$$\mathbf{R}_v = \chi_{v,\max} \cdot \mathbf{R}_{E,v,\max} \quad (4.81)$$

Similarly, if the set of original circuit topology vertices is $\mathcal{V} = \{A, B, C, \dots\}$, then (4.74) can be rewritten for the specific case when *all* the circuit routing resistor networks are Caterpillar trees:

$$\begin{aligned} \mathbf{R} &= \begin{bmatrix} \mathbf{R}_A \\ \mathbf{R}_B \\ \mathbf{R}_C \\ \vdots \end{bmatrix} = \underbrace{\begin{bmatrix} \chi_{A,\max} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \chi_{B,\max} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \chi_{C,\max} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{\chi_{\max}} \cdot \underbrace{\begin{bmatrix} \mathbf{R}_{E,A,\max} \\ \mathbf{R}_{E,B,\max} \\ \mathbf{R}_{E,C,\max} \\ \vdots \end{bmatrix}}_{\mathbf{R}_{E,\max}} \\ &= \chi_{\max} \cdot \mathbf{R}_{E,\max} \end{aligned} \quad (4.82)$$

In (4.82), χ_{\max} is block diagonal and each block has full column rank, therefore χ_{\max} also has full column rank. Let χ_{\max}^+ denote the left inverse of χ_{\max} :

$$\mathbf{R}_{E,\max} = \chi_{\max}^+ \cdot \mathbf{R} = (\chi_{\max}^T \cdot \chi_{\max})^{-1} \cdot \chi_{\max}^T \cdot \mathbf{R} \quad (4.83)$$

To transform between general routing resistor trees and Caterpillar trees with corresponding terminals and equal effective resistance between terminals, equations (4.74) and (4.82) are equated, and the inverse in (4.83) is used to find the edge resistances $\mathbf{R}_{E,\max}$ of the Caterpillar tree:

$$\begin{aligned} \chi \cdot \mathbf{R}_E = \mathbf{R} &= \chi_{\max} \cdot \mathbf{R}_{E,\max} \xrightarrow{(4.83)} \mathbf{R}_{E,\max} = (\chi_{\max}^T \cdot \chi_{\max})^{-1} \cdot \chi_{\max}^T \cdot \mathbf{R} \\ &= (\chi_{\max}^T \cdot \chi_{\max})^{-1} \cdot \chi_{\max}^T \cdot \chi \cdot \mathbf{R}_E \end{aligned} \quad (4.84)$$

If, as under Assumption 2, problem (4.69) is solved for $\{\mathbf{R}_v \in \text{col}(\chi_{v,\max}) | v \in \mathcal{V}\}$ and each routing resistor network is replaced by a Caterpillar tree, then the domain of \mathbf{R} is restricted to the column space, $\text{col}(\chi_{\max})$, of χ_{\max} :

$$\begin{aligned} \begin{bmatrix} t^* \\ \mathbf{R}^{u1} \end{bmatrix} &= \underset{t, \mathbf{R} \in \text{col}(\chi_{\max})}{\text{argmin}} t \\ \text{subject to} &\begin{cases} \hat{\mathbf{c}}_e(\mathbf{R}) \succeq \hat{\mathbf{c}}_e^m \\ -\Delta_R^u \cdot \mathbf{1}_N \preceq \Delta_R \cdot \mathbf{R} \preceq \Delta_R^u \cdot \mathbf{1}_N \\ \mathbf{0} \preceq \mathbf{R} \preceq \mathbf{R}^{u0} \\ \mathbf{R}^{u0} \cdot (1-t) \preceq \mathbf{R} \end{cases} \end{aligned} \quad (4.85)$$

Since χ_{\max} has full column rank, the linear mapping in (4.82) is injective. By a change of variable, the solution space of (4.85) is written in terms of $\mathbf{R}_{E,\max}$:

$$\begin{aligned} \begin{bmatrix} t^* \\ \mathbf{R}_{E,\max}^{u1} \end{bmatrix} &= \underset{t, \mathbf{R}_{E,\max} \in \mathbb{R}^{\mathbf{R}_{E,\max}}}{\text{argmin}} t \\ \text{subject to} &\begin{cases} \hat{\mathbf{c}}_e(\chi_{\max} \cdot \mathbf{R}_{E,\max}) \succeq \hat{\mathbf{c}}_e^m \\ -\Delta_R^u \cdot \mathbf{1}_N \preceq \Delta_R \cdot \chi_{\max} \cdot \mathbf{R}_{E,\max} \preceq \Delta_R^u \cdot \mathbf{1}_N \\ \mathbf{0} \preceq \chi_{\max} \cdot \mathbf{R}_{E,\max} \preceq \mathbf{R}^{u0} \\ \mathbf{R}^{u0} \cdot (1-t) \preceq \chi_{\max} \cdot \mathbf{R}_{E,\max} \end{cases} \end{aligned} \quad (4.86)$$

Using (4.82), the solution to the original problem in (4.85) is $\mathbf{R}^{u1} = \chi_{\max} \cdot \mathbf{R}_{\mathbf{E},\max}^{u1}$, while the vector $\begin{bmatrix} t \\ \mathbf{R}_{\mathbf{E},\max} \end{bmatrix} = \begin{bmatrix} 1 \\ \chi_{\max}^+ \cdot \mathbf{R}^0 \end{bmatrix}$ can be used as an initial starting point to solve (4.86).

The post-layout electrical constraint function $\hat{c}_e(\chi_{\max} \cdot \mathbf{R}_{\mathbf{E},\max})$ is nonlinear, therefore a nonlinear constrained optimization algorithm is required to solve (4.86). In Section 4.5.5, problem (4.86) is solved using successive linear programming (SLP).

4.5.5 Numerical Solution to (4.86) by Successive Linear Programming

There are two reasons for using Successive Linear Programming [GS61] to solve (4.86): Firstly, the residual of a linear approximation to $\hat{c}_e(\chi_{\max} \cdot \mathbf{R}_{\mathbf{E},\max})$ is small and only a few SLP iterations are needed, in practice, to find a solution by SLP. Secondly, only first-order derivatives need to be calculated, and the cost of minimization, in terms of the number of DC simulations, can be kept low.

Let τ label the SLP steps, such that the initial starting point is denoted by $\tau = 0$. To minimize (4.86) by SLP, the constraint $\hat{c}_e(\chi_{\max} \cdot \mathbf{R}_{\mathbf{E},\max})$ must be linearized. Let $\mathbf{R}_{\mathbf{E},\max}^{(\tau)}$ denote the value of $\mathbf{R}_{\mathbf{E},\max}$ at the beginning of step τ , $\mathbf{R}_{\mathbf{E},\max}^{(0)} = \chi_{\max}^+ \cdot \mathbf{R}^0$, and let $\bar{c}_e^{(\tau)}(\mathbf{R}_{\mathbf{E},\max})$ denote the linear approximation to $\hat{c}_e(\chi_{\max} \cdot \mathbf{R}_{\mathbf{E},\max})$ at $\mathbf{R}_{\mathbf{E},\max}^{(\tau)}$:

$$\begin{aligned} \bar{c}_e^{(\tau)}(\mathbf{R}_{\mathbf{E},\max}) &= \hat{c}_e(\chi_{\max} \cdot \mathbf{R}_{\mathbf{E},\max}^{(\tau)}) + \mathbf{J}(\mathbf{R}_{\mathbf{E},\max}^{(\tau)}) \cdot (\mathbf{R}_{\mathbf{E},\max} - \mathbf{R}_{\mathbf{E},\max}^{(\tau)}); \\ \mathbf{J}(\mathbf{R}_{\mathbf{E},\max}^{(\tau)}) &= \frac{\partial \hat{c}_e}{\partial \mathbf{R}_{\mathbf{E},\max}^T}(\mathbf{R}_{\mathbf{E},\max}^{(\tau)}); \quad \tau = 0, 1, 2, \dots \end{aligned} \quad (4.87)$$

where $\mathbf{J}(\mathbf{R}_{\mathbf{E},\max}^{(\tau)})$ is the Jacobian matrix of $\hat{c}_e(\chi_{\max} \cdot \mathbf{R}_{\mathbf{E},\max})$ evaluated at $\mathbf{R}_{\mathbf{E},\max}^{(\tau)}$.

To calculate the Jacobian matrix, the sensitivity of the circuit DC bias point to the resistor values, $\mathbf{R}_{\mathbf{E},\max}$, must be calculated. The number of node voltages and branch currents for which sensitivity information must be acquired to calculate the Jacobian matrix is smaller than $|\mathbf{R}_{\mathbf{E},\max}|$, therefore an adjoint sensitivity method can be employed to improve the efficiency of calculation [DR69]. A flavor of the adjoint method is normally supported for sensitivity analysis in most commercial circuit simulators, such as Titan [Inf08].

The linear program to solve in step τ of SLP is given below:

$$\begin{aligned} \begin{bmatrix} \psi^{(\tau+1)} \\ t^{(\tau+1)} \\ \mathbf{R}_{\mathbf{E},\max}^{(\tau+1)} \end{bmatrix} &= \underset{\psi, t, \mathbf{R}_{\mathbf{E},\max} \in \mathbb{R}^{\mathbf{R}_{\mathbf{E},\max}}}{\operatorname{argmin}} w_1 \cdot t + w_2 \cdot \psi \\ \text{subject to} &\begin{cases} \bar{\mathbf{c}}_e^{(\tau)}(\mathbf{R}_{\mathbf{E},\max}) + \mathbf{1}_{n_{\hat{\mathbf{c}}_e}} \cdot \psi \succeq \hat{\mathbf{c}}_e^m \\ -\Delta_R^u \cdot \mathbf{1}_N \preceq \Delta_R \cdot \chi_{\max} \cdot \mathbf{R}_{\mathbf{E},\max} \preceq \Delta_R^u \cdot \mathbf{1}_N \\ \mathbf{0} \preceq \chi_{\max} \cdot \mathbf{R}_{\mathbf{E},\max} \preceq \mathbf{R}^{u0} \\ \mathbf{R}^{u0} \cdot (1-t) \preceq \chi_{\max} \cdot \mathbf{R}_{\mathbf{E},\max} \\ \mathbf{a} \preceq \mathbf{R}_{\mathbf{E},\max} - \mathbf{R}_{\mathbf{E},\max}^{(\tau)} \preceq \mathbf{b} \\ \psi \geq 0 \end{cases} \end{aligned} \quad (4.88)$$

Vectors \mathbf{a} and \mathbf{b} are move bounds to control the accuracy of the linear approximation to $\hat{\mathbf{c}}_e$. The new parameter ψ is introduced to relax the electrical constraints, thereby it becomes easier to find a feasible solution to the linear program for $\mathbf{a} \preceq \mathbf{R}_{\mathbf{E},\max} - \mathbf{R}_{\mathbf{E},\max}^{(\tau)} \preceq \mathbf{b}$ when the starting vector is infeasible. Constants w_1 and w_2 are weights selected to ensure that the minimization of ψ takes precedence over the minimization of t , for example, $w_2/w_1 = 10/1$. Finally, $\mathbf{1}_{n_{\hat{\mathbf{c}}_e}} \in \{1\}^{n_{\hat{\mathbf{c}}_e}}$.

4.6 Selection of a Final Layout

Each placement in the vector of placements, \mathbf{P} , was routed and adjusted for congestion as discussed in Section 4.4, thereby completing the layout synthesis procedure. Placements were also ranked according to how well the geometric specifications are met, as defined by (4.36).

In addition to geometric specifications, specifications are typically set on the electrical performances of the circuit; this is discussed in Section 2.2.2. The electrical behavior of a circuit may change post-layout synthesis; electrical behavior may be sensitive to the layout of each circuit device, as well as differences in circuit placement and routing [Has01, YD09, ESL⁺11]. The effect of folding a CMOS device into multiple fingers was discussed in Section 4.2.1.

The selection of a final layout from list \mathbf{P} will be made in consideration of both the geometric as well as the electrical performance specifications calculated post-layout synthesis.

Two steps are needed to complete the final selection task: First, a post-layout electrical model is extracted for each layout in \mathbf{P} and the electrical performances are calculated by numerical simulation. Secondly, layouts are ranked using a scalar cost metric that takes into account both the geometric and electrical specifications. These two steps are detailed in the following two subsections.

4.6.1 Post-Layout Circuit Extraction

An electrical circuit model (netlist) is extracted from each layout in \mathbf{P} using the commercial tool in [Cad05].

In general, the extraction rules are particular to each circuit and to the requirement of the designer, as discussed in Section 3.5.

An example of extraction rules suitable for low frequency circuits can be found in the results chapter under Section 6.2.

4.6.2 Scalar Cost Metric Of Performance Specifications

Vectors \mathbf{f}_e^u and \mathbf{f}_g^u are the respective upper specification bounds on electrical performances, \mathbf{f}_e , and geometric performances, \mathbf{f}_g . Equation (2.35) can be rewritten as (4.89):

$$\underbrace{\text{electrical specifications}}_{\mathbf{f}_e \preceq \mathbf{f}_e^u} \wedge \underbrace{\text{geometric specifications}}_{\mathbf{f}_g \preceq \mathbf{f}_g^u} \quad (4.89)$$

The electrical performances, \mathbf{f}_e , are calculated for each layout in \mathbf{P} by simulating the extracted layout netlist. Post-layout electrical performance is layout dependent. This is denoted by adding a reference to the placement, \mathbf{p} , used in simulation, such that $\mathbf{f}_e = \mathbf{f}_e(\mathbf{p})$. The layout dependent electrical specifications are written as follows:

$$\mathbf{f}_e(\mathbf{p}) \preceq \mathbf{f}_e^u \quad (4.90)$$

Geometric specifications are set on placement area, aspect ratio, width, length, etc. In Section 4.3.4, and without loss of generality, these specifications were transformed into the modified area objective, \hat{A} , such that for placement \mathbf{p} :

$$\left(\mathbf{f}_g \preceq \mathbf{f}_g^u \right) \iff \left(\hat{A}(\mathbf{p}) = A_{\max} \right); \quad \neg \left(\mathbf{f}_g \preceq \mathbf{f}_g^u \right) \iff \left(\hat{A}(\mathbf{p}) > A_{\max} \right) \quad (4.91)$$

If the geometric specifications are unsatisfied, then a placement does not fit into the allotted space on the chip floorplan. Unsatisfied geometric specifications, other than the constraint placed on layout area, are penalized in \hat{A} .

A scalar exponential cost metric, φ , is defined combining the electrical performance specifications, $\mathbf{f}_e(\mathbf{p}) \preceq \mathbf{f}_e^u$, and the modified area specification, $\hat{A}(\mathbf{p}) = A_{\max}$:

$$\varphi(\mathbf{p}) = \exp \left(-\mathbf{w}[0] \cdot \left(A_{\max} - \hat{A}(\mathbf{p}) \right) \right) + \sum_{i=1}^{n_{f_e}} \exp \left(-\mathbf{w}[i] \cdot \left(\mathbf{f}_e^u[i] - \mathbf{f}_e(\mathbf{p})[i] \right) \right) \quad (4.92)$$

where $\mathbf{w} = [\mathbf{w}[0], \mathbf{w}[1], \dots, \mathbf{w}[n_{f_e}]]$ is a vector of weights denoting the significance of each performance. Vector \mathbf{w} is set by the designer prior to layout synthesis to control the tradeoff between performances in the cost metric.

The electrical performances and the modified area are calculated for each layout $\mathbf{p} \in \mathbf{P}$; from these, the value of φ is subsequently calculated. The layout corresponding to the lowest value of φ is selected as the final layout.

To calculate φ , the electrical performances, \mathbf{f}_e , must be obtained by simulation. This is a costly prospect, as it must be repeated for each layout in \mathbf{P} . If it is known a priori which electrical performances are sensitive to layout synthesis or changes between placement arrangements, then the cost of final selection can be reduced. Let \mathbf{f}_s denote the list of layout-sensitive performances in \mathbf{f}_e , such that $n_{\mathbf{f}_s} = |\mathbf{f}_s|$ and $n_{\mathbf{f}_s} < n_{\mathbf{f}_e}$. Let \mathbf{f}_s^u denote the corresponding vector of upper specification bounds.

For example, if it is known a priori that common mode rejection ratio (CMRR), power supply rejection ratio (PSRR), and input offset voltage (IOV) are the layout-sensitive performances of a certain operational amplifier topology, then $n_{\mathbf{f}_s} = 3$ and $\mathbf{f}_s = [\text{CMRR}, \text{PSRR}, \text{IOV}]$.

If the layout-sensitive performances are known, then cost metric, φ , can be rewritten as given in (4.93), which is less costly to calculate than (4.92).

$$\varphi(\mathbf{p}) = \exp(-\mathbf{w}[0] \cdot (A_{\max} - \hat{A}(\mathbf{p}))) + \sum_{i=1}^{n_{\mathbf{f}_s}} \exp(-\mathbf{w}[i] \cdot (\mathbf{f}_s^u[i] - \mathbf{f}_s(\mathbf{p})[i])) \quad (4.93)$$

Algorithm-5 details final layout selection in pseudo code.

Algorithm-5 final-layout-selection

```

/1/ input: list of layouts,  $\mathbf{P}$ 
/2/ output: final layout,  $\mathbf{p}^*$ 

  for  $k = 1, \dots, |\mathbf{P}|$  do
/3/    $\mathbf{p} = \mathbf{P}[k]$ 

/4/   Extract the layout netlist corresponding to layout  $\mathbf{p}$ 
/5/   Simulate and calculate the electrical performances,  $\mathbf{f}_s$ , for layout  $\mathbf{p}$ 

/6/    $\varphi(\mathbf{p}) = \exp(\mathbf{w}[0] \cdot (\hat{A}(\mathbf{p}) - A_{\max})) + \sum_{i=1}^{n_{\mathbf{f}_s}} \exp(\mathbf{w}[i] \cdot (\mathbf{f}_s(\mathbf{p})[i] - \mathbf{f}_s^u[i]))$ 

/7/    $\mathbf{p}^* = \underset{\mathbf{p} \in \mathbf{P}}{\operatorname{argmin}} \varphi(\mathbf{p})$ 

  return

```

4.7 Summary

In this chapter, a new flow is presented for the automatic layout synthesis of analog integrated circuits. Each synthesis step is completely constraint-driven, and is performed under consideration of a predefined list of device, placement, and routing constraints. For any vector of design parameter values, the set of layouts that meet the device, placement, and routing constraints is generated. An optimal layout is then selected that best fits the performance specifications. Examples are presented for CMOS devices.

The folding of a large CMOS device into multiple fingers will alter the device electrical behavior and change the electrical performance of the complete circuit. A solution to find the optimal number of fingers for each device is to enumerate every folding possibility, then employ a rectangle packing algorithm to generate all possible circuit placements. The optimal device layouts will be used in optimal circuit placement.

Algorithm-1 is presented for the constrained enumeration of CMOS device layouts. Device layout constraints are used to improve robustness and geometric performance. Multiple CMOS devices can be divided, so that they can be laid out in a common centroid configuration to improve matching. Algorithm-2 is presented in order to find the optimal number of divisions for matching. The common centroid configurations for different analog functional blocks are also discussed.

A new metric, modified area, was defined combining the geometric performances – such as width and length – so as to geometrically rank placements and make meaningful comparisons with electrical performances.

Routing congestion occurs when the margins between devices are too small to properly fit circuit routing with the placed devices. Barring failure to route, congestion will degrade the electrical performances. A procedure is presented to adjust congested placements. The unused space between devices is found using a cheap maze routing algorithm. If the space is very small, then the margins are increased and circuit placement is repeated. The procedure is given in Algorithm-3.

The folding of a CMOS device into fingers will change the device terminal resistance and alter the DC drain current. Circuit routing contributes an additional parasitic resistance. As a result of these effects, the electrical sizing rules may be violated after layout. In Algorithm-4, a procedure is given to rectify violated sizing rules. This is done by setting an upper bound on routing resistance. Principle to this procedure is that DC sensitivity analysis is relatively cheap to perform. The effective resistance between every two vertices in the layout is obtained using a graph representation of circuit routing. The electrical constraints are then parameterized in terms of effective resistance. An optimization problem to find the maximum effective resistance subject to the electrical constraints is set up, then solved by successive linear programming.

Chapter 5

Layout-Driven Circuit Sizing

5.1 Introduction

In this chapter, a new procedure is presented to solve the circuit sizing problem described in Section 2.2.2 and formulated in equation (2.39).

The new method combines a deterministic search algorithm derived from the work in [SSGA00, AEG⁺00b, SEGA99] with the new automatic constraint-based layout synthesis flow presented in Chapter 4. When applied to the sizing of a circuit topology, the outcome of the new algorithm is a circuit layout that meets all the geometric constraints and specifications and a corresponding electrical model (netlist) that meets all the electrical constraints and specifications.

The remainder of this chapter is organized as follows. The deterministic search algorithm is reviewed in Section 5.2. The technical steps needed for the amalgamation of the search algorithm and the automatic synthesis flow are summarized in Section 5.3. The differences between layout-driven circuit sizing and traditional circuit sizing – without consideration of layout synthesis – are also drafted. In Sections 5.4 through 5.7, the issues resulting from the numerical evaluation of functions and partial derivatives, as well as layout synthesis, are presented in detail; techniques to handle these issues are also discussed. Computational cost is summated in Section 5.8.

5.2 Review of the Search Algorithm Employed in Circuit Sizing

The deterministic search algorithm described in [SSGA00, AEG⁺00b, SEGA99] is used to numerically search for a solution to the circuit sizing problem formulated in (2.39). An iterative approach is undertaken; during each iteration, the circuit sizing problem of (2.39) is reformulated as a constrained scalar minimization problem

of Euclidean distances in the design space. The search algorithm is terminated when all specifications and constraints are met, once no improvement is possible, or once a predefined maximum computational cost is reached.

Let κ denote iteration number and m denote total number of iterations, such that $\kappa = 0, \dots, m-1$. The solution in the design space, \mathcal{D} , to the minimization problem in iteration κ is denoted by $\mathbf{d}^{\kappa+1}$; this is used as the starting vector of iteration $\kappa+1$. The starting vector, \mathbf{d}^0 , of the first iteration ($\kappa = 0$) is provided as an input to the optimization algorithm. The final solution after m iterations is denoted by \mathbf{d}^m :

$$\mathbf{d}^0 \xrightarrow[\text{iteration 0}]{\text{solution in}} \mathbf{d}^1 \xrightarrow[\text{iteration 1}]{\text{solution in}} \mathbf{d}^2 \dots \mathbf{d}^\kappa \xrightarrow[\text{iteration } \kappa]{\text{solution in}} \mathbf{d}^{\kappa+1} \dots \mathbf{d}^{m-2} \xrightarrow[\text{iteration } m-2]{\text{solution in}} \mathbf{d}^{m-1} \xrightarrow[\text{iteration } m-1]{\text{solution in}} \mathbf{d}^m$$

In iteration κ , the linear approximation to the performance function, $\phi_{\mathbf{f}}$, is constructed at the iteration starting vector, \mathbf{d}^κ ; it is denoted by $\bar{\phi}_{\mathbf{f},\kappa}$:

$$\bar{\phi}_{\mathbf{f},\kappa} : \mathbb{R}^{n_{\mathbf{d}}} \longrightarrow \mathbb{R}^{n_{\mathbf{f}}} : \mathbf{d} \longmapsto \bar{\mathbf{f}}_\kappa; \quad \bar{\mathbf{f}}_\kappa = \mathbf{f}^\kappa + \mathbf{J}_{\mathbf{f}}(\mathbf{d}^\kappa) \cdot (\mathbf{d} - \mathbf{d}^\kappa) \quad (5.1)$$

where

$$\mathbf{d}^\kappa \xrightarrow{\phi_{\mathbf{f}}} \mathbf{f}^\kappa \quad (5.2)$$

and $\mathbf{J}_{\mathbf{f}}(\mathbf{d})$ is the Jacobian matrix of the performance function $\phi_{\mathbf{f}}$:

$$\mathbf{J}_{\mathbf{f}} : \mathbb{R}^{n_{\mathbf{d}}} \longrightarrow \mathbb{R}^{n_{\mathbf{f}}} \times \mathbb{R}^{n_{\mathbf{d}}} : \mathbf{d} \longmapsto \frac{\partial \phi_{\mathbf{f}}}{\partial \mathbf{d}^T}(\mathbf{d}) \quad (5.3)$$

Let i denote the i -th performance function, such that $i = 1, \dots, n_{\mathbf{f}}$, and j denote the j -th design parameter, such that $j = 1, \dots, n_{\mathbf{d}}$:

$$\mathbf{J}_{\mathbf{f}}(\mathbf{d})[i, j] = \frac{\partial \phi_{\mathbf{f}}[i]}{\partial \mathbf{d}[j]}(\mathbf{d}); \quad \mathbf{J}_{\mathbf{f}}(\mathbf{d})[i] = \frac{\partial \phi_{\mathbf{f}}[i]}{\partial \mathbf{d}^T}(\mathbf{d}) \quad (5.4)$$

Let \mathbf{x} denote a pre-image of upper specification bound, \mathbf{f}^u , according to $\bar{\phi}_{\mathbf{f},\kappa}$:

$$\mathbf{x} \xrightarrow{\bar{\phi}_{\mathbf{f},\kappa}} \mathbf{f}^u; \quad \mathbf{f}^u \stackrel{(5.1)}{=} \mathbf{f}^\kappa + \mathbf{J}_{\mathbf{f}}(\mathbf{d}^\kappa) \cdot (\mathbf{x} - \mathbf{d}^\kappa) \quad (5.5)$$

The difference $(\mathbf{x} - \mathbf{d}^\kappa)$ is of interest, as it is the step that needs to be taken from \mathbf{d}^κ in the design space to fulfill the performance specifications, $\mathbf{f} \preceq \mathbf{f}^u$, post linearization. The unknown variable in equation (5.5) is \mathbf{x} . Since $\text{rank}(\mathbf{J}_{\mathbf{f}}(\mathbf{d}^\kappa)) \leq \min(n_{\mathbf{f}}, n_{\mathbf{d}})$, (5.5) cannot be solved in the general case for a unique value of \mathbf{x} . In addition, unless $\mathbf{J}_{\mathbf{f}}(\mathbf{d}^\kappa)$ has full row rank, there may be no value of \mathbf{x} that concurrently solves all rows (individual equations) in (5.5); only an approximate solution to (5.5) is possible.

In consideration of the existence and uniqueness issues stated in the last paragraph, a two step approach is taken in [SSGA00] to determine an approximate solution to (5.5):

First, the Moore-Penrose pseudo inverse of each *individual* row vector in $\mathbf{J}_{\mathbf{f}}(\mathbf{d}^\kappa)$ is calculated; assuming $\|\mathbf{J}_{\mathbf{f}}(\mathbf{d}^\kappa)[i]\| > 0$:

$$(\mathbf{J}_{\mathbf{f}}(\mathbf{d}^\kappa)[i])^+ = (\mathbf{J}_{\mathbf{f}}(\mathbf{d}^\kappa)[i])^T \cdot \left(\mathbf{J}_{\mathbf{f}}(\mathbf{d}^\kappa)[i] \cdot (\mathbf{J}_{\mathbf{f}}(\mathbf{d}^\kappa)[i])^T \right)^{-1} = \frac{(\mathbf{J}_{\mathbf{f}}(\mathbf{d}^\kappa)[i])^T}{\|\mathbf{J}_{\mathbf{f}}(\mathbf{d}^\kappa)[i]\|^2} \quad (5.6)$$

The pseudo inverse is used to obtain the unique minimum length solution of the corresponding row in equation (5.5). Let $(\mathbf{x}_{\min,i} - \mathbf{d}^\kappa)$ denote the minimum length solution corresponding to the i -th row in equation (5.5):

$$\begin{aligned} \mathbf{x}_{\min,i} - \mathbf{d}^\kappa &= (\mathbf{J}_f(\mathbf{d}^\kappa)[i])^+ \cdot (\mathbf{f}^\mu[i] - \mathbf{f}^\kappa[i]) \\ &\stackrel{(5.6)}{=} \underbrace{\left(\frac{\mathbf{f}^\mu[i] - \mathbf{f}^\kappa[i]}{\|\mathbf{J}_f(\mathbf{d}^\kappa)[i]\|} \right)}_{\alpha_{\kappa,i} \text{ (scalar value)}} \cdot \underbrace{\left(\frac{\mathbf{J}_f(\mathbf{d}^\kappa)[i]}{\|\mathbf{J}_f(\mathbf{d}^\kappa)[i]\|} \right)^T}_{\text{unit vector}} = \alpha_{\kappa,i} \cdot \left(\frac{\mathbf{J}_f(\mathbf{d}^\kappa)[i]}{\|\mathbf{J}_f(\mathbf{d}^\kappa)[i]\|} \right)^T \end{aligned} \quad (5.7)$$

The minimum length solution is in the direction of steepest ascent or descent of linear function $\bar{\phi}_{f,\kappa}[i]$. Scalar value $\alpha_{\kappa,i}$ denotes the length and direction of $(\mathbf{x}_{\min,i} - \mathbf{d}^\kappa)$ along $\mathbf{J}_f(\mathbf{d}^\kappa)[i]/\|\mathbf{J}_f(\mathbf{d}^\kappa)[i]\|$. If $\alpha_{\kappa,i} < 0$, then performance value $\mathbf{f}^\kappa[i]$ does not fulfill the i -th specification. If $\alpha_{\kappa,i} \geq 0$, then $\mathbf{f}^\kappa[i]$ fulfills the i -th specification, and the magnitude $|\alpha_{\kappa,i}| = \|\mathbf{x}_{\min,i} - \mathbf{d}^\kappa\|$ is a safety margin. The values $\{\mathbf{x}_{\min,i} \mid i = 1, \dots, n_f\}$, will not be equal in general.

For the second step in the approach of [SSGA00], a single approximate solution is sought to replace the individual exact minimum length solutions. The starting vector, \mathbf{f}^κ , is replaced by an approximation, \mathbf{f} , in the performance space, while the scalar value $\alpha_{\kappa,i}$ becomes a scalar function of the performance space elements:

$$\alpha_{\kappa,i}(\mathbf{f}) = \frac{\mathbf{f}^\mu[i] - \mathbf{f}[i]}{\|\mathbf{J}_f(\mathbf{d}^\kappa)[i]\|} \quad (5.8)$$

Since $\mathbf{f} = \phi_f(\mathbf{d})$, equation (5.8) can be rewritten as a function, $\beta_{\kappa,i}$, of the design space elements:

$$\beta_{\kappa,i}(\mathbf{d}) = \alpha_{\kappa,i}(\mathbf{f}) = \frac{\mathbf{f}^\mu[i] - \phi_f(\mathbf{d})[i]}{\|\mathbf{J}_f(\mathbf{d}^\kappa)[i]\|} \quad (5.9)$$

To find an approximate solution, \mathbf{d} , in the design space that takes into account all the performances, the functions $\{\beta_{\kappa,i}(\mathbf{d}) \mid i = 1, \dots, n_f\}$ are combined in a single objective function to be minimized. Priority is given to the fulfillment of the performance specifications, then to the increase in the value of the safety margins [KD95, AGW94, LD89]:

$$\gamma_\kappa(\mathbf{d}) = \sum_{i=1}^{n_f} \exp(-\beta_{\kappa,i}(\mathbf{d})) \quad (5.10)$$

The value to minimize the objective function, γ_κ , is the approximate solution used to replace the individual solutions $\{\mathbf{x}_{\min,i} \mid i = 1, \dots, n_f\}$ in the design space. It is also the final solution, $\mathbf{d}^{\kappa+1}$, of iteration, κ :

$$\mathbf{d}^{\kappa+1} = \underset{\mathbf{d} \in \mathcal{D}}{\operatorname{argmin}} \gamma_\kappa(\mathbf{d}) \quad (5.11)$$

Only a solution in the feasible design space, \mathcal{D} , is valid:

$$\mathbf{d}^{\kappa+1} = \underset{\mathbf{d} \in \mathcal{D}}{\operatorname{argmin}} \gamma_\kappa(\mathbf{d}) \quad (5.12)$$

From (2.28), (5.12) becomes:

$$\mathbf{d}^{\kappa+1} = \underset{\mathbf{d} \in \mathcal{D}}{\operatorname{argmin}} \gamma_{\kappa}(\mathbf{d}) \quad \text{subject to} \quad \mathbf{c} \succeq \mathbf{c}^m \quad \text{where} \quad \boldsymbol{\phi}_{\mathbf{c}}(\mathbf{d}) = \mathbf{c} \quad (5.13)$$

The original circuit sizing problem of (2.39) has been reformulated as the constrained minimization in (5.13) with a scalar objective function.

In [SSGA00], a convex model is constructed to approximate (5.13) at \mathbf{d}^{κ} . The model is minimized within a suitable trust region around \mathbf{d}^{κ} , as discussed below.

The function $\boldsymbol{\phi}_{\mathbf{f}}$ is replaced by the linearization of (5.1) in (5.9) and (5.10):

$$\bar{\boldsymbol{\phi}}_{\mathbf{f},\kappa}(\mathbf{d}) \stackrel{(5.1)}{=} \mathbf{f}^{\kappa} + \mathbf{J}_{\mathbf{f}}(\mathbf{d}^{\kappa}) \cdot (\mathbf{d} - \mathbf{d}^{\kappa}) \quad (5.14)$$

$$\bar{\beta}_{\kappa,i}(\mathbf{d}) \stackrel{(5.9),(5.14)}{=} \frac{\mathbf{f}^u[i] - \mathbf{f}^{\kappa}[i]}{\|\mathbf{J}_{\mathbf{f}}(\mathbf{d}^{\kappa})[i]\|} - \frac{\mathbf{J}_{\mathbf{f}}(\mathbf{d}^{\kappa})[i]}{\|\mathbf{J}_{\mathbf{f}}(\mathbf{d}^{\kappa})[i]\|} \cdot (\mathbf{d} - \mathbf{d}^{\kappa}) \quad (5.15)$$

$$\bar{\gamma}_{\kappa}(\mathbf{d}) \stackrel{(5.10),(5.14)}{=} \sum_{i=1}^{n_{\mathbf{f}}} \exp(-\bar{\beta}_{\kappa,i}(\mathbf{d})) \quad (5.16)$$

The linear approximation to the constraint function, $\boldsymbol{\phi}_{\mathbf{c}}$, is constructed at the iteration starting vector, \mathbf{d}^{κ} ; it is denoted by $\bar{\boldsymbol{\phi}}_{\mathbf{c},\kappa}$:

$$\bar{\boldsymbol{\phi}}_{\mathbf{c},\kappa} : \mathbb{R}^{n_{\mathbf{d}}} \longrightarrow \mathbb{R}^{n_{\mathbf{c}}} : \mathbf{d} \longmapsto \bar{\mathbf{c}}_{\kappa}; \quad \bar{\mathbf{c}}_{\kappa} = \mathbf{c}^{\kappa} + \mathbf{J}_{\mathbf{c}}(\mathbf{d}^{\kappa}) \cdot (\mathbf{d} - \mathbf{d}^{\kappa}) \quad (5.17)$$

where

$$\mathbf{d}^{\kappa} \xrightarrow{\boldsymbol{\phi}_{\mathbf{c}}} \mathbf{c}^{\kappa} \quad (5.18)$$

and $\mathbf{J}_{\mathbf{c}}(\mathbf{d})$ is the Jacobian matrix of the constraint function $\boldsymbol{\phi}_{\mathbf{c}}$:

$$\mathbf{J}_{\mathbf{c}} : \mathbb{R}^{n_{\mathbf{d}}} \longrightarrow \mathbb{R}^{n_{\mathbf{c}}} \times \mathbb{R}^{n_{\mathbf{d}}} : \mathbf{d} \longmapsto \frac{\partial \boldsymbol{\phi}_{\mathbf{c}}}{\partial \mathbf{d}^T}(\mathbf{d}) \quad (5.19)$$

The approximation model to (5.13) is constructed using (5.16) and (5.17):

$$\mathbf{d}_{\text{model}}^{\kappa+1} = \underset{\mathbf{d} \in \mathcal{D}}{\operatorname{argmin}} \bar{\gamma}_{\kappa}(\mathbf{d}) \quad \text{subject to} \quad \bar{\mathbf{c}}_{\kappa} + \mathbf{J}_{\mathbf{c}}(\mathbf{d}^{\kappa}) \cdot (\mathbf{d} - \mathbf{d}^{\kappa}) \succeq \mathbf{c}^m \quad (5.20)$$

Objective function $\bar{\gamma}_{\kappa}$ is convex, as the Hessian matrix $\frac{\partial^2 \bar{\gamma}_{\kappa}}{\partial \mathbf{d} \partial \mathbf{d}^T}$ is positive semidefinite. Furthermore, the feasible region is defined by a system of linear inequalities and is a convex polyhedron. Therefore (5.20) can be solved to within any degree of precision using a convex programming algorithm [Nes83].

To solve the constrained minimization in (5.13) by a trust region approach using (5.20) as an approximation model, a new term is added to the objective function:

$$\begin{aligned} \mathbf{d}_{\text{model}}^{\kappa+1,\tau} &= \underset{\mathbf{d} \in \mathcal{D}}{\operatorname{argmin}} \bar{\gamma}_{\kappa}^2(\mathbf{d}) + \lambda_{\tau} \cdot \|\mathbf{d} - \mathbf{d}^{\kappa}\|^2 \\ &\text{subject to} \quad \bar{\mathbf{c}}_{\kappa} + \mathbf{J}_{\mathbf{c}}(\mathbf{d}^{\kappa}) \cdot (\mathbf{d} - \mathbf{d}^{\kappa}) \succeq \mathbf{c}^m \quad \wedge \quad \lambda_{\tau} \geq 0 \end{aligned} \quad (5.21)$$

The factor λ_τ controls the size of the trust region over which a solution is sought to the problem in (5.13) [Mar63]; $\mathbf{d}_{\text{model}}^{\kappa+1,\tau}$ is the solution corresponding to λ_τ . As λ_τ increases, the size of the trust region decreases and the magnitude of the corresponding parameter correction step, $\|\mathbf{d}_{\text{model}}^{\kappa+1,\tau} - \mathbf{d}^\kappa\|$, is reduced.

To select the size of the trust region, a compromise is made between the value of λ_τ and the reduction ratio measured at $\mathbf{d}^{\kappa+1}$. The reduction ratio is denoted by ρ and is defined as follows:

$$\rho = \frac{\text{actual reduction at } \mathbf{d}_{\text{model}}^{\kappa+1,\tau}}{\text{model reduction at } \mathbf{d}_{\text{model}}^{\kappa+1,\tau}} = \frac{\left(\gamma_\kappa(\mathbf{d}^\kappa) - \gamma_\kappa(\mathbf{d}_{\text{model}}^{\kappa+1,\tau})\right)}{\left(\gamma_\kappa(\mathbf{d}^\kappa) - \tilde{\gamma}_\kappa(\mathbf{d}_{\text{model}}^{\kappa+1,\tau})\right)}; \quad \rho \in (0, 1) \quad (5.22)$$

If ρ is smaller than a predesignated value, for example $\rho \leq 1/4$, then the trust region is decreased in size and a new parameter correction step is taken with $\lambda_{\tau+1} \geq \lambda_\tau$. Several steps, $\tau = 1, \dots, q_\kappa$, may be necessary to select a suitable trust region. After each step, if $\phi_c(\mathbf{d}_{\text{model}}^{\kappa+1,\tau}) < \mathbf{c}^m$, then a feasible correction step is taken to return to the feasible region.

The final feasible solution to satisfy (5.22), is $\mathbf{d}_{\text{model}}^{\kappa+1,q_\kappa}$ corresponding to λ_{q_κ} , such that:

$$\underbrace{\mathbf{d}^{\kappa+1}}_{\text{solution to (5.13)}} \longleftarrow \mathbf{d}_{\text{model}}^{\kappa+1,q_\kappa} \quad (5.23)$$

5.3 Technical Description of the Layout-Driven Circuit Sizing Problem

A summary of the technical steps necessary to evaluate circuit performances and constraints with and without layout synthesis is given in this section. This is based on the description of the circuit sizing problem given in Chapter 2 and the layout synthesis flow of Chapter 4.

A layout-driven circuit sizing problem is then described, which amalgamates the layout synthesis flow of Chapter 4 and the search algorithm of Section 5.2.

Finally, a *traditional* circuit sizing problem that does not employ layout synthesis is described, so that a direct comparison can be made between results of circuit sizing without layout synthesis and with the new layout-driven synthesis flow.

The equations in this summary will be referred to in the subsequent sections of this and the results chapter, so as to simplify the discussion.

The initial problem input consists of a circuit topology, a set of test benches, and a set of electrical and geometric performance specifications to be realized:

$$\text{initial problem input} = \left\{ \begin{array}{l} \mathcal{T}, \{\mathcal{TB}\}, \\ \left[\begin{array}{l} \mathbf{f}_e \\ \mathbf{f}_g \end{array} \right] \preceq \left[\begin{array}{l} \mathbf{f}_e^u \\ \mathbf{f}_g^u \end{array} \right] \end{array} \right. \quad (5.24)$$

Design parameters are extracted from the circuit and test bench devices, \mathcal{E} and \mathcal{EB} respectively, as done in Section 2.1.3. Electrical and geometric sizing rules are extracted from the circuit topology, \mathcal{T} , as done in Section 2.1.5:

$$\mathcal{T}, \{\mathcal{TB}\} \longrightarrow \left\{ \begin{array}{l} \mathbf{d}_{\text{original}} = \left[\begin{array}{l} \mathbf{d}_{\mathcal{E},\text{original}} \\ \mathbf{d}_{\mathcal{EB},\text{original}} \end{array} \right], \\ \mathcal{D}_{\text{original}} = \mathcal{D}_{\mathcal{E},\text{original}} \times \mathcal{D}_{\mathcal{EB},\text{original}}, \\ \text{equality constraints,} \\ \mathbf{c}_{g,\text{original}} \succeq \mathbf{c}_{g,\text{original}}^m \\ \mathbf{c}_e \succeq \mathbf{c}_e^m \end{array} \right. \quad (5.25)$$

The design parameters are normalized, as given in (2.7), and geometric sizing rules are employed to reduce the dimensions of the circuit design space by variable elimination methods, as given by (2.23) and (2.24):

$$\left\{ \begin{array}{l} \mathbf{d}_{\mathcal{E},\text{original}}, \mathcal{D}_{\mathcal{E},\text{original}}, \\ \mathbf{d}_{\mathcal{EB},\text{original}}, \mathcal{D}_{\mathcal{EB},\text{original}}, \\ \text{equality constraints,} \\ \mathbf{c}_{g,\text{original}} \succeq \mathbf{c}_{g,\text{original}}^m \end{array} \right\} \begin{array}{l} \text{elimination} \\ \text{methods} \\ \implies \\ \text{parameter} \\ \text{normalization} \end{array} \left\{ \begin{array}{l} \mathbf{d}_{\mathcal{E}}, \mathcal{D}_{\mathcal{E}}, \\ \mathbf{d}_{\mathcal{EB}}, \mathcal{D}_{\mathcal{EB}}, \\ \mathbf{c}_g \succeq \mathbf{c}_g^m \end{array} \right\} \quad (5.26)$$

Conversely, given a vector of reduced and normalized design parameters, \mathbf{d} , the original parameters must be calculated prior to numerical simulation or layout synthesis. This is achieved by calculating the inverse of (5.26):

$$\underbrace{\left[\begin{array}{l} \mathbf{d}_{\mathcal{E}} \\ \mathbf{d}_{\mathcal{EB}} \end{array} \right]}_{\mathbf{d}} \begin{array}{l} \text{inverse of} \\ \text{elimination methods} \\ \implies \\ \text{inverse of} \\ \text{parameter normalization} \end{array} \underbrace{\left[\begin{array}{l} \mathbf{d}_{\mathcal{E},\text{original}} \\ \mathbf{d}_{\mathcal{EB},\text{original}} \end{array} \right]}_{\mathbf{d}_{\text{original}}} \quad (5.27)$$

Geometric inequality constraint functions take the form of explicit analytical expressions of the circuit design parameters, while electrical constraints are obtained by circuit DC bias point calculation with a suitable DC test bench:

$$\mathbf{d}_{\mathcal{E}} \xrightarrow[\text{(5.27), analytical expressions}]{\phi_{\mathbf{c}_g}} \mathbf{c}_g \quad (5.28)$$

$$\mathbf{d} \xrightarrow[\text{(5.27), } \mathcal{T}, \mathcal{TB}\text{-DC, DC simulation}]{\phi_{ce}} \mathbf{c}_e \quad (5.29)$$

The pre-layout value of the electrical performances is obtained by numerical simulation of the circuit topology, \mathcal{T} , connected to the set of test benches $\{\mathcal{TB}\}$:

$$\mathbf{d} \xrightarrow[\text{(5.27), } \mathcal{T}, \{\mathcal{TB}\}, \text{simulation}]{\phi_{f_{e,wos}}} \mathbf{f}_{e,wos} \quad (\text{pre-layout value}) \quad (5.30)$$

The subscript "wos" – short for "without (layout) synthesis" – is used to indicate a pre-layout value.

To obtain the post-layout value of the electrical performances, the layout synthesis flow of Chapter 4 is executed. The result is a set \mathbf{P} of layouts corresponding to $\mathbf{d}_{\mathcal{E}}$:

$$\mathbf{d}_{\mathcal{E}} \xrightarrow[\text{(5.27), layout synthesis}]{\mathbf{P}} \quad (5.31)$$

A post-layout netlist is extracted for any layout in \mathbf{P} and simulated in connection to the set of test benches, $\{\mathcal{TB}\}$, in consideration of the test bench parameters, $\mathbf{d}_{\mathcal{EB}}$. The result is the electrical performances parametrized by placement:

$$\mathbf{d} \xrightarrow[\text{(5.27), layout synthesis}]{\phi_{f_{e,ws}}} (\mathbf{P}, \mathbf{d}_{\mathcal{EB}}) \xrightarrow[\text{and simulation for } \mathbf{p} \in \mathbf{P}]{\{\mathcal{TB}\}, \text{netlist extraction}} \mathbf{f}_{e,ws}(\mathbf{p}) \quad (\text{post-layout value}) \quad (5.32)$$

The subscript "ws" – short for "with (layout) synthesis" – is used to indicate a post-layout value.

Geometric performances, typically layout area, aspect ratio, width, and length, require layout synthesis to be calculated. They are independent of the test bench parameters, $\mathbf{d}_{\mathcal{EB}}$. The flow of Chapter 4 is also used:

$$\mathbf{d}_{\mathcal{E}} \xrightarrow[\text{(5.27), layout synthesis}]{\phi_{f_{g,ws}}} \mathbf{P} \xrightarrow[\text{for } \mathbf{p} \in \mathbf{P}]{\text{measurement}} \mathbf{f}_{g,ws}(\mathbf{p}) \quad (\text{exact value}) \quad (5.33)$$

In Section 4.3.4, a scalar objective function called modified area, \hat{A} , was defined that combines multiple geometric specifications in a single scalar and penalizes the deviation beyond geometric specification bounds. Two specific cases were considered. If the geometric specifications are given by (4.26), then \hat{A} is given by (4.29). If the geometric specifications are given by (4.27), then \hat{A} is given by (4.33):

$$\mathbf{d}_{\mathcal{E}} \xrightarrow[\text{(5.27), layout synthesis}]{\phi_{f_{g,ws}}} \mathbf{f}_{g,ws}(\mathbf{p}) \xrightarrow[\text{(4.29) or (4.33)}]{\phi_{\hat{A},ws}} \hat{A}_{ws}(\mathbf{p}) \quad (5.34)$$

An equivalence was shown in (4.30) and (4.34) between the geometric specifications and an inequality applied to \hat{A} , such that

$$\neg \left(\mathbf{f}_g(\mathbf{p}) \preceq \mathbf{f}_g^u \right) \stackrel{(4.30) \text{ or } (4.34)}{\iff} \left(\hat{A}(\mathbf{p}) > A_{\max} \right) \quad (5.35)$$

From (5.35), a sufficient condition for geometric performance satisfaction is that $\hat{A}(\mathbf{p}) \leq A_{\max}$. This condition will be used in lieu of $\mathbf{f}_g \preceq \mathbf{f}_g^u$ when using the search algorithm of Section 5.2.

Without resorting to layout synthesis, a range for circuit area unassociated with any specific placement is derived from the value of the device design parameters by a suitable procedure:

$$\mathbf{d}_{\mathcal{E}} \xrightarrow[\text{(5.27), estimation procedure}]{\phi_{A, \text{wos}, \text{min}}} A_{\text{wos}, \text{min}}; \quad \mathbf{d}_{\mathcal{E}} \xrightarrow[\text{(5.27), estimation procedure}]{\phi_{A, \text{wos}, \text{max}}} A_{\text{wos}, \text{max}}; \quad (5.36)$$

$$A_{\text{ws}} \in [A_{\text{wos}, \text{min}}, A_{\text{wos}, \text{max}}] \quad (\text{estimated range})$$

A value is selected from $[A_{\text{wos}, \text{min}}, A_{\text{wos}, \text{max}}]$ to estimate the modified area objective:

$$\hat{A}_{\text{ws}} = (1 - \rho) \cdot A_{\text{wos}, \text{min}} + \rho \cdot A_{\text{wos}, \text{max}}; \quad \rho \in [0, 1] \quad (5.37)$$

A procedure to derive the range $[A_{\text{wos}, \text{min}}, A_{\text{wos}, \text{max}}]$ and calculate \hat{A}_{ws} is given in Appendix A. A procedure to approximate the gradient of \hat{A}_{ws} is given in Appendix B.

From the discussion of this section, two new circuit sizing problems can be crafted to replace the original circuit sizing problem of (2.39).

In the first problem, \mathbf{f}_e and \hat{A} are calculated post layout synthesis:

$$\text{circuit sizing with layout synthesis} = \left\{ \begin{array}{l} \mathcal{T}, \{\mathcal{TB}\}, \mathbf{d} = \begin{bmatrix} \mathbf{d}_{\mathcal{E}} \\ \mathbf{d}_{\mathcal{EB}} \end{bmatrix}, \mathcal{D} = \mathcal{D}_{\mathcal{E}} \times \mathcal{D}_{\mathcal{EB}}, \\ \text{Find any} \\ \mathbf{d} \in \mathcal{D} \\ \text{subject to} \\ \mathbf{c}_g \preceq \mathbf{c}_g^m \wedge \mathbf{c}_e \preceq \mathbf{c}_e^m \wedge \begin{bmatrix} \mathbf{f}_{e, \text{ws}}(\mathbf{p}) \\ \hat{A}_{\text{ws}}(\mathbf{p}) \end{bmatrix} \preceq \begin{bmatrix} \mathbf{f}_e^u \\ A_{\max} \end{bmatrix} \\ \text{where} \\ \mathbf{d}_{\mathcal{E}} \xrightarrow[\text{(5.27), layout synthesis}]{\phi_{\text{cg}}} \mathbf{P}, \mathbf{p} \in \mathbf{P} \\ \mathbf{d}_{\mathcal{E}} \xrightarrow{\phi_{\text{cg}}} \mathbf{c}_g, \mathbf{d} \xrightarrow{\phi_{\text{ce}}} \mathbf{c}_e, \mathbf{d} \xrightarrow{\phi_{\text{fe}, \text{ws}}} \mathbf{f}_{e, \text{ws}}(\mathbf{p}), \\ \mathbf{d}_{\mathcal{E}} \xrightarrow{\phi_{\hat{A}, \text{ws}}} \hat{A}_{\text{ws}}(\mathbf{p}) \end{array} \right. \quad (5.38)$$

In the second problem, \mathbf{f}_e and \hat{A} are calculated prior to layout synthesis:

$$\text{circuit sizing without layout synthesis} = \left\{ \begin{array}{l} \mathcal{T}, \{\mathcal{TB}\}, \mathbf{d} = \begin{bmatrix} \mathbf{d}_{\mathcal{E}} \\ \mathbf{d}_{\mathcal{EB}} \end{bmatrix}, \mathcal{D} = \mathcal{D}_{\mathcal{E}} \times \mathcal{D}_{\mathcal{EB}}, \\ \text{Find any} \\ \mathbf{d} \in \mathcal{D} \\ \text{subject to} \\ \mathbf{c}_g \preceq \mathbf{c}_g^m \wedge \mathbf{c}_e \preceq \mathbf{c}_e^m \wedge \begin{bmatrix} \mathbf{f}_{e,\text{wos}} \\ \hat{A}_{\text{wos}} \end{bmatrix} \preceq \begin{bmatrix} \mathbf{f}_e^u \\ A_{\text{max}} \end{bmatrix} \\ \text{where} \\ \mathbf{d}_{\mathcal{E}} \xrightarrow{\phi_{cg}} \mathbf{c}_g, \mathbf{d} \xrightarrow{\phi_{ce}} \mathbf{c}_e, \mathbf{d} \xrightarrow{\phi_{fe,\text{wos}}} \mathbf{f}_{e,\text{wos}}, \\ \mathbf{d}_{\mathcal{E}} \xrightarrow{\phi_{A,\text{wos},\text{min}}} A_{\text{wos},\text{min}}, \mathbf{d}_{\mathcal{E}} \xrightarrow{\phi_{A,\text{wos},\text{max}}} A_{\text{wos},\text{max}}, \\ \hat{A}_{\text{wos}} = (1 - \rho) \cdot A_{\text{wos},\text{min}} + \rho \cdot A_{\text{wos},\text{max}}, \rho \in [0, 1] \end{array} \right. \quad (5.39)$$

In order to compare anecdotal results from real circuit sizing problems, both (5.38) and (5.39) must to be solved for each problem.

5.4 Issues in Numerical Function Evaluation

Due to the use of floating point numbers, design parameter values have finite precision, while functions are subject to round-off error.

As shown in Section 5.3, only the geometric inequality constraint functions are described by analytical expressions. Electrical performance and constraint functions are evaluated numerically by circuit simulation; this will add a computational error to their value.

Post-layout evaluation of performances will add a discretization error, furthermore, behavior will also depend on layout geometry. Pre-layout estimation of the modified area objective is given, procedurally, to be within a range of values. It is dependent on the possible layouts of individual devices; this is presented in Appendix A.

The search algorithm of Section 5.2 is gradient-based. It requires the evaluation of the partial derivatives of the circuit performance and constraint functions at the starting vector of each algorithm iteration. In practice, it is not always possible to construct the partial derivative functions analytically or to evaluate them by a direct numerical method, such as the adjoint sensitivity method [DR69]. In such cases, a numerical approximation can be employed to replace the exact value of the partial derivatives.

Here, finite difference functions [Tre96, CM10] are used to approximate the partial derivatives of $\boldsymbol{\phi}_f$, and hence the Jacobian matrix $\mathbf{J}_f(\mathbf{d})$ defined in (5.3). A general finite difference approximation to $\mathbf{J}_f(\mathbf{d})$ is expressed as follows:

$$\mathbf{J}_f(\mathbf{d})[i, j] = \frac{\partial \boldsymbol{\phi}_f[i]}{\partial \mathbf{d}[j]}(\mathbf{d}) \approx \frac{1}{h_{i,j}} \cdot \sum_{k=-l}^r \mu_k \cdot \boldsymbol{\phi}_f(\mathbf{d} + k \cdot h_{i,j} \cdot \mathbf{e}_j)[i]; \quad (5.40)$$

$$i = 1, \dots, n_f; \quad j = 1, \dots, n_d$$

Performances and design parameters are indexed by i and j respectively, $h_{i,j}$ is the step size or grid spacing used in approximating the partial derivative indexed by $[i, j]$, $[-l, r]$ is a range of integers corresponding to multiples of $h_{i,j}$, and $[\mu_{-l}, \dots, \mu_r]$ is a vector of finite difference coefficients. The vector of coefficients must be selected such that the finite difference approximation converges to $\mathbf{J}_f(\mathbf{d})[i, j]$ as $h_{i,j} \rightarrow 0$. Suitable coefficients can be found algorithmically, for example, by the method in [For98].

The accuracy of a finite difference approximation is limited by truncation error. Pending differentiability class, it is possible to approximate a partial derivative to an arbitrary order of accuracy by increasing the number of terms in (5.40). The tradeoff is with the increase in the number of function evaluations.

Finite difference approximation is also prone to stability problems and round-off error [GMW81]. Numerical function evaluation will contribute a computational error. A decrease in step size, $h_{i,j}$, will reduce truncation error and increase the effect of round-off error and computational error.

The selection of an optimal step size and order of accuracy is costly. Here, in order to check the number of function evaluations, partial derivatives are approximated by the first-order forward difference function; while each step size, $h_{i,j}$, is fixed by the designer to a small value prior to the initiation of the circuit sizing. Heuristics as well as necessary constraints on the value of $h_{i,j}$ are pointed out in the subsequent sections of this chapter. By substitution in (5.40), the approximation to $\boldsymbol{\phi}_f$ by first-order forward difference is given by:

$$\mathbf{J}_f(\mathbf{d})[i, j] \approx \frac{\overbrace{\boldsymbol{\phi}_f(\mathbf{d} + h_{i,j} \cdot \mathbf{e}_j)[i]}^{\mathbf{f}_j[i]} - \overbrace{\boldsymbol{\phi}_f(\mathbf{d})[i]}^{\mathbf{f}_e[i]}}{h_{i,j}} = \frac{\mathbf{f}_j[i] - \mathbf{f}_e[i]}{h_{i,j}}; \quad (5.41)$$

$$i = 1, \dots, n_f; \quad j = 1, \dots, n_d$$

The partial derivative approximations of the electrical constraint functions, $\boldsymbol{\phi}_{ce}$, can be given in a similar manner to (5.40) and (5.41).

With regards to function evaluation and partial derivative approximation, three categories of functions are distinguished in the technical description of Section 5.3. Each category will be handled individually in the following three sections. A suitable estimate to $\mathbf{J}_{A_{\text{wos}}}$ is delegated to Appendix B.

5.5 Geometric Inequality Constraint Functions

The mapping of geometric inequality constraint functions, represented in (5.28), is an explicit expression of the device design parameters.

The source of error when evaluating geometric constraint functions according to (5.28) is finite precision and round-off error.

The partial derivatives of the geometric constraint functions can be constructed analytically without the need for approximation functions. For example, the partial derivative functions of rule /3/ in Table 2.2 are given by:

$$\frac{\partial}{\partial W_1} (W_1 \cdot L_1) = L_1; \quad \frac{\partial}{\partial L_1} (W_1 \cdot L_1) = W_1 \quad (5.42)$$

5.6 Electrical Performances and Constraints Without Layout Synthesis

The numerical evaluation of electrical constraint functions and pre-layout electrical performance functions is represented by the mappings in (5.29) and (5.30).

The numerical evaluation of functions and partial derivatives will introduce truncation and computational error, in addition to the precision and round-off error caused by the use of floating point numbers.

The following subsections will refer to the electrical performances, $\mathbf{f}_{e,\text{wos}}$. Derivations can be applied in a similar manner to the electrical constraints, \mathbf{c}_e .

5.6.1 Truncation Error

For the gradient-based algorithm of Section 5.2, each function is assumed to be continuous and differentiable. More specifically, the Jacobian matrix, $\mathbf{J}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})$, must exist at the starting vector, \mathbf{d}^κ , of each algorithm iteration, κ . If the partial derivative indexed by $[i, j]$ in $\mathbf{J}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})$ is replaced by a first-order forward difference approximation, then the second order partial derivative, with respect to the j -th design parameter, must be continuous along the line segment from \mathbf{d} to $(\mathbf{d} + h_{i,j} \cdot \mathbf{e}_j)$ for the intermediate value theorem to apply and the local truncation error to be bounded:

$$\mathbf{J}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})[i, j] - \frac{\mathbf{f}_{e,\text{wos},j}[i] - \mathbf{f}_{e,\text{wos}}[i]}{h_{i,j}} = \frac{R_{\mathbf{f}_{e,\text{wos},i,j}}(\mathbf{d})}{h_{i,j}} \quad (5.43)$$

such that

$$\frac{R_{\mathbf{f}_{e,\text{wos},i,j}}(\mathbf{d})}{h_{i,j}} = -\frac{h_{i,j}}{2} \cdot \frac{\partial^2 \phi_{\mathbf{f}_{e,\text{wos}}[i]}}{\partial^2 \mathbf{d}[j]} (\mathbf{d} + \xi \cdot \mathbf{e}_j) \quad \text{with } \xi \in [0, h_{i,j}] \quad (5.44)$$

In this case, an upper bound can be derived for the magnitude of truncation error:

$$\frac{|R_{\mathbf{f}_{e,\text{wos},i,j}(\mathbf{d})}|}{h_{i,j}} \leq \frac{h_{i,j}}{2} \cdot \overbrace{\sup_{\xi \in [0, h_{i,j}]} \left| \frac{\partial^2 \phi_{\mathbf{f}_{e,\text{wos},i,j}}[i]}{\partial^2 \mathbf{d}[j]} (\mathbf{d} + \xi \cdot \mathbf{e}_j) \right|}^{M_{e,\text{wos},i,j}(\mathbf{d})}} \quad (5.45)$$

$$\left| \mathbf{J}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})[i, j] - \frac{\widetilde{\mathbf{f}_{e,\text{wos},j}[i]} - \widetilde{\mathbf{f}_{e,\text{wos},j}[i]}}{h_{i,j}} \right| \leq \frac{h_{i,j}}{2} \cdot M_{e,\text{wos},i,j}(\mathbf{d}) \quad (5.46)$$

5.6.2 Computational Error

Numerical simulation is used in the mapping of design parameters to electrical performance values. A computational error will be introduced as a result of numerical simulation. For example, in direct time integration methods [CC96, Wei02], local error can be estimated and controlled, while trapezoidal rule ringing in stiff circuits and the effect of switching between integration methods may contribute an additional error that cannot be reduced by tightening error tolerance limits [Kun95b].

The effect of computational error can be treated in a similar manner to round-off error in classical error analysis. Let the new function $\widetilde{\phi}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})$ represent the function $\phi_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})$ including computational error, and let $\eta_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})$ be the function representing the error:

$$\eta_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d}) = \widetilde{\phi}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d}) - \phi_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d}) \quad (5.47)$$

Let $\Delta\eta_{\mathbf{f}_{e,\text{wos},i,j}}$ be the function representing computational error in the first-order forward difference approximation:

$$\begin{aligned} \Delta\eta_{\mathbf{f}_{e,\text{wos},i,j}}(\mathbf{d}) &= \eta_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d} + h_{i,j} \cdot \mathbf{e}_j) - \eta_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d}) \\ &= \underbrace{\widetilde{\phi}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d} + h_{i,j} \cdot \mathbf{e}_j)}_{\widetilde{\mathbf{f}_{e,\text{wos},j}[i]}} - \underbrace{\widetilde{\phi}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})}_{\widetilde{\mathbf{f}_{e,\text{wos},j}[i]}} - \\ &\quad \underbrace{\phi_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d} + h_{i,j} \cdot \mathbf{e}_j)}_{\mathbf{f}_{e,\text{wos},j}[i]} + \underbrace{\phi_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})}_{\mathbf{f}_{e,\text{wos},j}[i]} \\ &= \widetilde{\mathbf{f}_{e,\text{wos},j}[i]} - \widetilde{\mathbf{f}_{e,\text{wos},j}[i]} - \mathbf{f}_{e,\text{wos},j}[i] + \mathbf{f}_{e,\text{wos},j}[i] \end{aligned} \quad (5.48)$$

From (5.43) and (5.48), truncation and computational error in the first-order forward difference approximation can be represented in a single equation:

$$\mathbf{J}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})[i, j] - \frac{\widetilde{\mathbf{f}_{e,\text{wos},j}[i]} - \widetilde{\mathbf{f}_{e,\text{wos},j}[i]}}{h_{i,j}} = \frac{R_{\mathbf{f}_{e,\text{wos},i,j}}(\mathbf{d})}{h_{i,j}} - \frac{\Delta\eta_{\mathbf{f}_{e,\text{wos},i,j}}(\mathbf{d})}{h_{i,j}} \quad (5.49)$$

As with truncation error, an upper bound is placed on computational error in the first-order forward difference approximation:

$$|\Delta\eta_{\mathbf{f}_{e,\text{wos}},i,j}(\mathbf{d})| \leq 2 \cdot \sup_{\xi \in [0, h_{i,j}]} \overbrace{\left| \boldsymbol{\phi}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d} + \xi \cdot \mathbf{e}_j)[i] - \boldsymbol{\phi}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d} + \xi \cdot \mathbf{e}_j)[i] \right|}^{N_{e,\text{wos},i,j}(\mathbf{d})}} \quad (5.50)$$

such that

$$\left| \widetilde{\mathbf{f}_{e,\text{wos},j}[i]} - \widetilde{\mathbf{f}_{e,\text{wos}}[i]} - \mathbf{f}_{e,\text{wos},j}[i] + \mathbf{f}_{e,\text{wos}}[i] \right| \leq 2N_{e,\text{wos},i,j}(\mathbf{d}) \quad (5.51)$$

From (5.46), (5.49) and (5.51):

$$\left| \mathbf{J}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})[i,j] - \frac{\widetilde{\mathbf{f}_{e,\text{wos},j}[i]} - \widetilde{\mathbf{f}_{e,\text{wos}}[i]}}{h_{i,j}} \right| \leq \frac{h_{i,j}}{2} \cdot M_{e,\text{wos},i,j}(\mathbf{d}) + \frac{2N_{e,\text{wos},i,j}(\mathbf{d})}{h_{i,j}} \quad (5.52)$$

5.6.3 Adjustments to Palliate Truncation and Computational Error

Let vector $\mathbf{N}_{e,\text{wos}}$ denote the upper bound on the magnitude of computational error in electrical performance values in the feasible design space; using (5.47):

$$\mathbf{N}_{e,\text{wos}}[i] = \sup_{\mathbf{d} \in \mathcal{D}} |\eta_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})[i]|; \quad i = 1, \dots, n_{\mathbf{f}_e} \quad (5.53)$$

Vector $\mathbf{N}_{e,\text{wos}}$ is a global bound. An estimate $\hat{\mathbf{N}}_{e,\text{wos}}$ to $\mathbf{N}_{e,\text{wos}}$ can be manually assigned by the designer according to the accuracy of the numerical simulations needed to calculate each electrical performance. If $\widetilde{\mathbf{f}_{e,\text{wos}}} = \widetilde{\boldsymbol{\phi}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})}$, then:

$$\text{abs} \left(\widetilde{\mathbf{f}_{e,\text{wos}}} - \mathbf{f}_{e,\text{wos}} \right) \preceq \hat{\mathbf{N}}_{e,\text{wos}} \quad (5.54)$$

The electrical performance specifications can be adjusted to take into account the effect of computational error in numerical function evaluation:

$$\left. \begin{array}{l} \mathbf{f}_e \preceq \mathbf{f}_e^u \\ \text{abs} \left(\widetilde{\mathbf{f}_{e,\text{wos}}} - \mathbf{f}_{e,\text{wos}} \right) \stackrel{(5.54)}{\preceq} \hat{\mathbf{N}}_{e,\text{wos}} \end{array} \right\} \implies \underbrace{\hat{\mathbf{N}}_{e,\text{wos}} + \widetilde{\mathbf{f}_{e,\text{wos}}}}_{\text{to guarantee } \mathbf{f}_e \preceq \mathbf{f}_e^u} \preceq \mathbf{f}_e^u \quad (5.55)$$

$$\mathbf{f}_e^u \stackrel{(5.55)}{\longleftarrow} (\mathbf{f}_e^u - \hat{\mathbf{N}}_{e,\text{wos}}) \quad (\text{modified specification bound}) \quad (5.56)$$

When applying the substitution of (5.56) to the numerator in (5.9), the worst-case computational error will be taken into consideration when solving problem (5.13).

When using the search algorithm of Section 5.2, the partial derivatives of $\boldsymbol{\phi}_{\mathbf{f}_e}$ are approximated at the starting vector, \mathbf{d}^κ , of each iteration, κ , by the first-order forward difference function presented in (5.41).

From (5.52), a decrease in each step size, $h_{i,j}$, will increase the computational error in the corresponding approximation and decrease the local truncation error. For a fixed step size, if the value of the difference $\mathbf{f}_{e,\text{wos},j}[i] - \mathbf{f}_{e,\text{wos}}[i]$ in (5.43) is small at \mathbf{d}^k such that it will be dominated by $R_{\mathbf{f}_{e,\text{wos},i,j}}$, then first-order forward difference can be replaced with a higher order finite difference approximation. A reduction in truncation error, however, is contingent upon the differentiability class of the function within the suitable neighborhood around the value of \mathbf{d}^k in the design space. An increase of order will increase the number of needed function evaluations and the cost of partial derivative approximation by finite difference. Computational error, added in equation (5.49), will also limit the accuracy and absorb the benefit of a high order approximation.

From (5.15), if each difference $\widetilde{\mathbf{f}_{e,\text{wos},j}[i]} - \widetilde{\mathbf{f}_{e,\text{wos}}[i]}$, with $i \in 1, \dots, n_f$ and $j \in 1, \dots, n_d$, has the same sign as $\mathbf{J}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})[i, j]$, then each gradient approximation will point in the general direction of improvement in the design space, though the weight of each performance in the objective function (5.16) will change. From (5.49), this is equivalent to the following condition:

$$\begin{aligned} \text{sign}(\mathbf{J}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})[i, j]) &= \text{sign}\left(\frac{\widetilde{\mathbf{f}_{e,\text{wos},j}[i]} - \widetilde{\mathbf{f}_{e,\text{wos}}[i]}}{h_{i,j}}\right) \\ \implies \text{sign}(\mathbf{J}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})[i, j]) &= \\ \text{sign}\left(\mathbf{J}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})[i, j] - \left(\frac{R_{\mathbf{f}_{e,\text{wos},i,j}}(\mathbf{d})}{h_{i,j}} - \frac{\Delta\eta_{\mathbf{f}_{e,\text{wos},i,j}}(\mathbf{d})}{h_{i,j}}\right)\right) & \end{aligned} \quad (5.57)$$

Only when a gradient direction is small in magnitude relative to the error and opposite in sign will a step be taken outside the general direction of improvement.

It is suggested, here, that truncation error can be considered a useful correction term when added to $\mathbf{J}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})[i, j]$ so as to produce the first-order forward difference function. The first-order forward difference function gives the average of the partial derivative $\mathbf{J}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})[i, j]$ over the range $[\mathbf{d}[j], \mathbf{d}[j] + h_{i,j}]$:

$$\frac{\mathbf{f}_{e,\text{wos},j}[i] - \mathbf{f}_{e,\text{wos}}[i]}{h_{i,j}} = \frac{1}{h_{i,j}} \int_{\tau=0}^{\tau=h_{i,j}} \frac{\partial \phi_{\mathbf{f}_{e,\text{wos}}[i]}}{\partial \mathbf{d}[j]} (\mathbf{d} + \tau \cdot \mathbf{e}_j) d\tau \quad (5.58)$$

From (5.43) and (5.58):

$$\underbrace{\mathbf{J}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})[i, j] - \frac{R_{\mathbf{f}_{e,\text{wos},i,j}}(\mathbf{d})}{h_{i,j}}}_{\text{partial derivative + truncation error}} = \underbrace{\frac{1}{h_{i,j}} \int_{\tau=0}^{\tau=h_{i,j}} \frac{\partial \phi_{\mathbf{f}_{e,\text{wos}}[i]}}{\partial \mathbf{d}[j]} (\mathbf{d} + \tau \cdot \mathbf{e}_j) d\tau}_{\text{first-order finite difference approximation}} \quad (5.59)$$

Truncation error compensates for large variations in value over small distances and improves algorithm robustness. The source of large sensitivity may be the real circuit response or may be due to the computational error.

Each step size, $h_{i,j}$, is selected in consideration of the estimated upper bound $\hat{N}_{e,\text{wos}}[i]$ on computational error for the i -th performance, so that $\hat{N}_{e,\text{wos}}[i]/h_{i,j}$ is small; and in consideration of the partial derivative $\mathbf{J}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})[i,j]$, so that it is averaged over small distances in the design space.

For example, if the j -th design parameter is the width, W , of a CMOS device, such that $W_{\min} = 0.2\mu\text{m}$, and the i -th performance is circuit gain, such that $\hat{N}_{e,\text{wos}}[i] = 2\text{dB}$, then the designer may select $h_{i,j} = 5 \cdot W_{\min}$.

The estimated upper bound, $\hat{N}_{e,\text{wos}}$, on computational error will be used to set a lower limit on the approximation to the partial derivatives:

$$\left(\left| \widetilde{\mathbf{f}_{e,\text{wos},j}[i]} - \widetilde{\mathbf{f}_{e,\text{wos}}[i]} \right| < 2\hat{N}_{e,\text{wos}} \right) \implies (\mathbf{J}_{\mathbf{f}_{e,\text{wos}}}(\mathbf{d})[i,j] \leftarrow 0); \quad (5.60)$$

$$j \in 1, \dots, n_{\mathbf{d}}; \quad i \in 1, \dots, n_{\mathbf{f}_e}$$

5.7 Performances with Layout-Driven Circuit Sizing

The post-layout evaluation of electrical performances and the modified area objective is represented by the mappings in (5.32) and (5.34) respectively. Prior to numerical evaluation by circuit simulation, a layout is synthesized and a post-layout circuit model is extracted using the synthesis flow described in Chapter 4.

Layout synthesis and model extraction will introduce a discretization error and a placement-specific error, as will be described below. This is in addition to the truncation and computational errors caused by the numerical evaluation of functions and the precision and round-off error caused by the use of floating point numbers.

Insight into the cloaked steps of the layout synthesis flow can be used to improve the first-order forward difference approximations to post-layout partial derivatives.

The following discussion will refer to the electrical performances, denoted by \mathbf{f}_e , but can be applied in an exact manner to the modified area objective, \hat{A} .

Without loss of generality, examples using CMOS devices will be presented in this section.

5.7.1 Discretization Error

Discretization error is introduced by the layout synthesis flow when continuous device design parameters are mapped to discontinuous layout parameters, as is done in Section 4.2.

For example, the mapping of CMOS design parameters to layout parameters is represented in (4.3), while the discretization of CMOS device width is given by (4.11). The magnitude of discretization error is given by (4.12) and illustrated in Figure 4.4. Device length can be discretized in a similar manner so as to obtain the complete vector of discrete CMOS design parameters:

$$\underbrace{[W, L]}_{\mathbf{d}_{\text{CMOS}}} \xrightarrow{(4.3)} \underbrace{[W_f, L_f, n_f]}_{\lambda_{\text{CMOS}}} \xrightarrow{(4.11)} \underbrace{[W_{\text{discrete}}, L_{\text{discrete}}]}_{\mathbf{d}_{\text{CMOS,discrete}}} \quad (5.61)$$

The circuit design parameters, $\mathbf{d}_{\mathcal{E}}$, are obtained from the device design parameters by renormalization and variable elimination, as represented by (5.26), while the opposite operation is represented by (5.27). By application of (5.27), followed by discretization at the device level, followed by (5.26), the vector of discrete circuit design parameters is obtained:

$$\mathbf{d}_{\mathcal{E}} \xrightarrow{(5.27)} \mathbf{d}_{\mathcal{E},\text{original}} \xrightarrow{\text{discretization}} \mathbf{d}_{\mathcal{E},\text{original,discrete}} \xrightarrow{(5.26)} \mathbf{d}_{\mathcal{E},\text{discrete}} \quad (5.62)$$

From (5.62), a discrete circuit design space can be defined:

$$\mathcal{D}_{\mathcal{E},\text{discrete}} = \left\{ \mathbf{d}_{\mathcal{E},\text{discrete}} \in \mathcal{D}_{\mathcal{E}} \mid \exists_{\mathbf{d}_{\mathcal{E}} \in \mathcal{D}} \mathbf{d}_{\mathcal{E}} \xrightarrow{(5.62)} \mathbf{d}_{\mathcal{E},\text{discrete}} \right\}; \quad \mathcal{D}_{\mathcal{E},\text{discrete}} \subset \mathcal{D}_{\mathcal{E}} \quad (5.63)$$

Without loss of generality, the test bench design parameters, $\mathbf{d}_{\mathcal{E}\mathcal{B}}$, are assumed, here, to remain continuous. A partially discrete design space can therefore be defined, the diminution "pd" will be used for "partially discrete" in subscripts:

$$\mathcal{D}_{\text{pd}} = \mathcal{D}_{\mathcal{E},\text{discrete}} \times \mathcal{D}_{\mathcal{E}\mathcal{B}}; \quad \mathcal{D}_{\text{pd}} \subset \mathcal{D} \quad (5.64)$$

$$\underbrace{\begin{bmatrix} \mathbf{d}_{\mathcal{E}} \\ \mathbf{d}_{\mathcal{E}\mathcal{B}} \end{bmatrix}}_{\mathbf{d}} \in \mathcal{D} \xrightarrow[(5.64)]{(5.62), (5.63)} \underbrace{\begin{bmatrix} \mathbf{d}_{\mathcal{E},\text{discrete}} \\ \mathbf{d}_{\mathcal{E}\mathcal{B}} \end{bmatrix}}_{\mathbf{d}_{\text{pd}}} \in \mathcal{D}_{\text{pd}} \quad (5.65)$$

Let $\mathbf{d}_{\mathcal{E},\text{error}}$ denote the discretization error in the circuit design parameters:

$$\mathbf{d}_{\mathcal{E},\text{error}} = \mathbf{d}_{\mathcal{E},\text{discrete}} - \mathbf{d}_{\mathcal{E}} \quad (5.66)$$

$$\underbrace{\begin{bmatrix} \mathbf{d}_{\mathcal{E},\text{discrete}} \\ \mathbf{d}_{\mathcal{E}\mathcal{B}} \end{bmatrix}}_{\mathbf{d}_{\text{pd}}} \stackrel{(5.66)}{=} \underbrace{\begin{bmatrix} \mathbf{d}_{\mathcal{E}} \\ \mathbf{d}_{\mathcal{E}\mathcal{B}} \end{bmatrix}}_{\mathbf{d}} + \begin{bmatrix} \mathbf{d}_{\mathcal{E},\text{error}} \\ \mathbf{0} \end{bmatrix} = \mathbf{d} + \begin{bmatrix} \mathbf{d}_{\mathcal{E},\text{error}} \\ \mathbf{0} \end{bmatrix} \quad (5.67)$$

In Section 4.2.1, an upper bound was placed on the magnitude of discretization error in each design parameter during device layout synthesis. This was exemplified in (4.13) for CMOS device width. At the circuit level, the upper bound on error can be represented by a vector $\mathbf{d}_{\mathcal{E},\text{error-max}}$, such that:

$$\text{abs}(\mathbf{d}_{\mathcal{E},\text{error}}) \preceq \mathbf{d}_{\mathcal{E},\text{error-max}} \quad (5.68)$$

Let $\mathbf{f}_{e,\text{error},\text{wos}}$ denote the change in the value of the electrical performance vector due to discretization error in the circuit design parameters; using (5.30):

$$\mathbf{f}_{e,\text{pd},\text{wos}} = \boldsymbol{\phi}_{\mathbf{f}e,\text{wos}}(\mathbf{d}_{\text{pd}}) \stackrel{(5.67)}{=} \boldsymbol{\phi}_{\mathbf{f}e,\text{wos}}\left(\mathbf{d} + \begin{bmatrix} \mathbf{d}_{\mathcal{E},\text{error}} \\ \mathbf{0} \end{bmatrix}\right) \quad (5.69)$$

$$\mathbf{f}_{e,\text{error},\text{wos}} = \mathbf{f}_{e,\text{pd},\text{wos}} - \mathbf{f}_{e,\text{wos}} \stackrel{(5.69)}{=} \boldsymbol{\phi}_{\mathbf{f}e,\text{wos}}(\mathbf{d}_{\text{pd}}) - \boldsymbol{\phi}_{\mathbf{f}e,\text{wos}}(\mathbf{d}) \quad (5.70)$$

Using (5.68), an upper bound can be derived for $\mathbf{f}_{e,\text{error},\text{wos}}$; for $i = 1, \dots, n_{\mathbf{f}e}$:

$$|\mathbf{f}_{e,\text{error},\text{wos}}[i]| \leq \overbrace{\sup_{\text{abs}(\mathbf{d}_{\mathcal{E},\text{error}}) \leq \mathbf{d}_{\mathcal{E},\text{error-max}}} \left| \boldsymbol{\phi}_{\mathbf{f}e,\text{wos}}\left(\mathbf{d} + \begin{bmatrix} \mathbf{d}_{\mathcal{E},\text{error}} \\ \mathbf{0} \end{bmatrix}\right)[i] - \boldsymbol{\phi}_{\mathbf{f}e,\text{wos}}(\mathbf{d})[i] \right|}^{Q_{\text{wos}}(\mathbf{d})[i]} \quad (5.71)$$

5.7.2 Placement Dependency

As discussed in Section 4.2, the mapping between device design and layout parameters is not unique. From the possible device layout variants generated for a value of $\mathbf{d}_{\mathcal{E}}$, a set \mathbf{P} of apposite circuit placements is enumerated and routed. This is represented by the mapping in (5.31). As a consequence, the discretization of design parameters is placement dependent. It is convenient to note this dependence by a parametrization of the discrete circuit design parameters by placement $\mathbf{p} \in \mathbf{P}$:

$$\mathbf{d}_{\mathcal{E}} \xrightarrow{(5.27), \text{ layout synthesis}} \mathbf{P} \xrightarrow[\text{for } \mathbf{p} \in \mathbf{P}]{\text{ discretization}} \mathbf{d}_{\mathcal{E},\text{original,discrete}}(\mathbf{p}) \xrightarrow{(5.26)} \mathbf{d}_{\mathcal{E},\text{discrete}}(\mathbf{p}) \quad (5.72)$$

$$\mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}) = \mathbf{d}_{\mathcal{E},\text{discrete}}(\mathbf{p}) - \mathbf{d}_{\mathcal{E}} \quad (5.73)$$

$$\underbrace{\begin{bmatrix} \mathbf{d}_{\mathcal{E},\text{discrete}}(\mathbf{p}) \\ \mathbf{d}_{\mathcal{E}\mathcal{B}} \end{bmatrix}}_{\mathbf{d}_{\text{pd}}(\mathbf{p})} \stackrel{(5.73)}{=} \underbrace{\begin{bmatrix} \mathbf{d}_{\mathcal{E}} \\ \mathbf{d}_{\mathcal{E}\mathcal{B}} \end{bmatrix}}_{\mathbf{d}} + \begin{bmatrix} \mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}) \\ \mathbf{0} \end{bmatrix} = \mathbf{d} + \begin{bmatrix} \mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}) \\ \mathbf{0} \end{bmatrix} \quad (5.74)$$

The change in the value of the electrical performances due to design parameter discretization for a placement \mathbf{p} is denoted by $\mathbf{f}_{e,\text{error},\text{wos}}(\mathbf{p})$; using (5.30):

$$\mathbf{f}_{e,\text{pd},\text{wos}}(\mathbf{p}) \stackrel{(5.74)}{=} \boldsymbol{\phi}_{\mathbf{f}e,\text{wos}}(\mathbf{d}_{\text{pd}}(\mathbf{p})) = \boldsymbol{\phi}_{\mathbf{f}e,\text{wos}}\left(\mathbf{d} + \begin{bmatrix} \mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}) \\ \mathbf{0} \end{bmatrix}\right) \quad (5.75)$$

$$\mathbf{f}_{e,\text{error},\text{wos}}(\mathbf{p}) \stackrel{(5.75)}{=} \boldsymbol{\phi}_{\mathbf{f}e,\text{wos}}(\mathbf{d}_{\text{pd}}(\mathbf{p})) - \boldsymbol{\phi}_{\mathbf{f}e,\text{wos}}(\mathbf{d}) \quad (5.76)$$

The change in the value of the electrical performances due to design parameter discretization for a placement \mathbf{p} , as well as layout parasitic devices is denoted by $\mathbf{f}_{e,\text{error},\text{ws}}(\mathbf{p})$; using (5.32):

$$\mathbf{f}_{e,\text{pd},\text{ws}}(\mathbf{p}) \stackrel{(5.74)}{=} \boldsymbol{\phi}_{\mathbf{f}e,\text{ws}}(\mathbf{d}_{\text{pd}}(\mathbf{p})) = \boldsymbol{\phi}_{\mathbf{f}e,\text{ws}}\left(\mathbf{d} + \begin{bmatrix} \mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}) \\ \mathbf{0} \end{bmatrix}\right) \quad (5.77)$$

$$\mathbf{f}_{e,\text{error},\text{ws}}(\mathbf{p}) \stackrel{(5.77)}{=} \boldsymbol{\phi}_{\mathbf{f}_{e,\text{ws}}}(\mathbf{d}_{\text{pd}}(\mathbf{p})) - \boldsymbol{\phi}_{\mathbf{f}_{e,\text{ws}}}(\mathbf{d}) \quad (5.78)$$

The change in the value of the electrical performances for a placement \mathbf{p} *due uniquely to* layout parasitic devices is denoted by $\mathbf{f}_{e,\text{error},\Delta}(\mathbf{p})$:

$$\mathbf{f}_{e,\text{error},\Delta}(\mathbf{p}) \stackrel{(5.76),(5.78)}{=} \mathbf{f}_{e,\text{error},\text{ws}}(\mathbf{p}) - \mathbf{f}_{e,\text{error},\text{ws}}(\mathbf{p}) \quad (5.79)$$

Without layout synthesis, the value of $\mathbf{f}_{e,\text{error},\text{ws}}(\mathbf{p})$ is bounded by (5.71), since

$$\text{abs}(\mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p})) \preceq \mathbf{d}_{\mathcal{E},\text{error-max}} \text{ for all } \mathbf{p} \in \mathbf{P} \quad (5.80)$$

so that for $i=1, \dots, n_{\mathbf{f}_e}$:

$$|\mathbf{f}_{e,\text{error},\text{ws}}(\mathbf{p})[i]| \leq \underbrace{Q_{\text{wos}}(\mathbf{d})[i]}_{(5.71)} \quad (5.81)$$

With layout synthesis, the elements of $\mathbf{f}_{e,\text{error},\text{ws}}(\mathbf{p})$ are unbounded. This is because the change in device location in the placement, the differences in routing, and other layout specific attributes are hidden and unaccounted for when mapping from the design to the performance space using the layout synthesis flow of Chapter 4.

In practical circuit examples, the unbounded error, $\mathbf{f}_{e,\text{error},\text{ws}}(\mathbf{p})[i]$, can dominate the pre-layout performance value, $\mathbf{f}_{e,\text{wos}}[i]$, if the i -th performance is layout-sensitive. More specifically, the error, $\mathbf{f}_{e,\text{error},\Delta}(\mathbf{p})[i]$, uniquely due to layout parasitic devices, dominates over the bounded discretization error, $\mathbf{f}_{e,\text{error},\text{wos}}(\mathbf{p})[i]$.

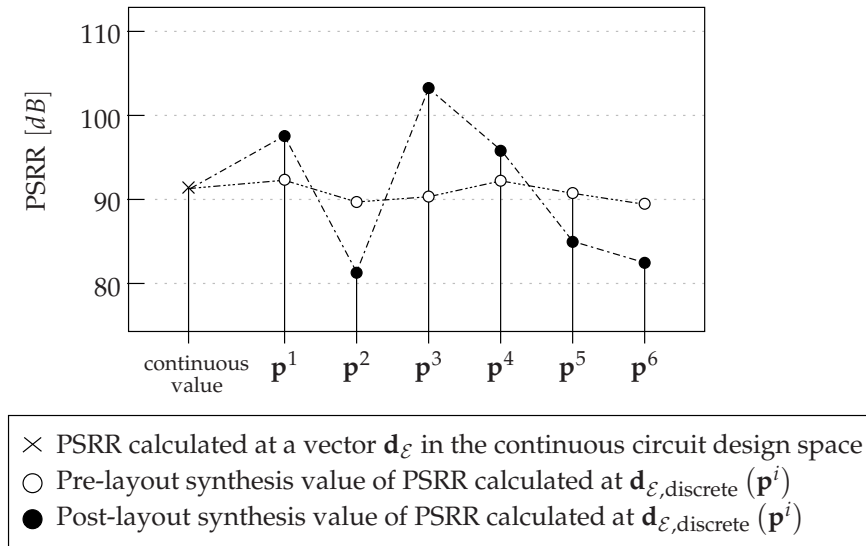


Figure 5.1: The effect of circuit design parameter discretization and layout synthesis on the value of the power supply rejection ratio (PSRR) of an operational amplifier is illustrated.

A practical illustration of error is given in Figure 5.1. First, the power supply rejection ratio (PSRR) of an operational amplifier is calculated by circuit simulation for

a value of $\mathbf{d}^T = [\mathbf{d}_{\mathcal{E}}^T; \mathbf{d}_{\mathcal{E}\mathcal{B}}^T]$ in the feasible design space. Secondly, the layout synthesis flow of Chapter 4 is called; for this example, the set of layouts produced by the flow is $\mathbf{P} = \{\mathbf{p}^1, \dots, \mathbf{p}^6\}$. Thirdly, for the layouts in \mathbf{P} , the corresponding discretized circuit design parameter vectors, $\{\mathbf{d}_{\mathcal{E},\text{discrete}}(\mathbf{p}^i) \mid i = 1, \dots, 6\}$, are calculated as described in (5.72). Fourthly, PSRR is calculated for the discretized circuit design parameter vectors with and without layout synthesis; this corresponds to the use of equations (5.75) and (5.77) respectively. The amount of post-layout error in PSRR is relatively large and reached 14% for \mathbf{p}^3 . This will affect gradient direction and step size calculation during the use of a search algorithm.

5.7.3 Solution Selection in the Design Space Under Consideration of Discretization and Placement Error

It is recalled that during iteration, κ , of the search algorithm, and in each sub-iteration, $\tau = 1, \dots, q_\kappa$, problem (5.21) is solved in the continuous design space, \mathcal{D} , to obtain $\mathbf{d}_{\text{model}}^{\kappa+1,\tau}$. The solution $\mathbf{d}^{\kappa+1}$ to problem (5.13) – also the starting vector of iteration $(\kappa+1)$ – is approximated, such that $\mathbf{d}^{\kappa+1} \approx \mathbf{d}_{\text{model}}^{\kappa+1,q_\kappa}$.

Computational error in function evaluation at $\mathbf{d}^{\kappa+1,\tau}$ is taken into consideration as done in (5.56) for the case of traditional circuit sizing without layout synthesis.

Post layout synthesis, the subspace $\mathcal{D}_{\mathcal{E}}$ of \mathcal{D} is discretized according to (5.63). As a consequence, the solution to (5.13) must be selected from the partially discrete design space, \mathcal{D}_{pd} . A four step approach is taken, here, to select a discrete solution.

First, in each sub-iteration, τ , problem (5.21) is solved in the continuous design space to obtain $\mathbf{d}_{\text{model}}^{\kappa+1,\tau}$. The feasible correction step is then applied, if necessary, in order to ensure the satisfaction of the inequality constraints.

Secondly, the solution is discretized by calling the layout synthesis flow of Chapter 4 with input $\mathbf{d}_{\text{model}}^{\kappa+1,\tau}$ to determine the set of valid circuit placements. The best final placement, as determined in Section 4.6, is left to the third step coming forthwith.

Let $\mathbf{P}_{\kappa,\tau}$ denote the set of valid circuit placements corresponding to $\mathbf{d}_{\text{model}}^{\kappa+1,\tau}$. For each placement $\mathbf{p}_{\kappa,\tau}$ in $\mathbf{P}_{\kappa,\tau}$, the placement-dependent circuit design parameter vector $\mathbf{d}_{\mathcal{E},\text{model},\text{discrete}}(\mathbf{p}_{\kappa,\tau})$ is obtained by applying (5.72):

$$\begin{array}{ccc}
 \mathbf{d}_{\mathcal{E},\text{model}}^{\kappa+1,\tau} & \xrightarrow{(5.27), \text{ layout synthesis}} & \mathbf{P}_{\kappa,\tau} \\
 & \text{discretization} \downarrow \text{ for } \mathbf{p}_{\kappa,\tau} \in \mathbf{P}_{\kappa,\tau} & \\
 \mathbf{d}_{\mathcal{E},\text{model},\text{discrete}}(\mathbf{p}_{\kappa,\tau}) & \xleftarrow{(5.26)} & \mathbf{d}_{\mathcal{E},\text{model},\text{original},\text{discrete}}(\mathbf{p}_{\kappa,\tau})
 \end{array} \tag{5.82}$$

The stages of the mapping in (5.82) are illustrated in Figure 5.2.

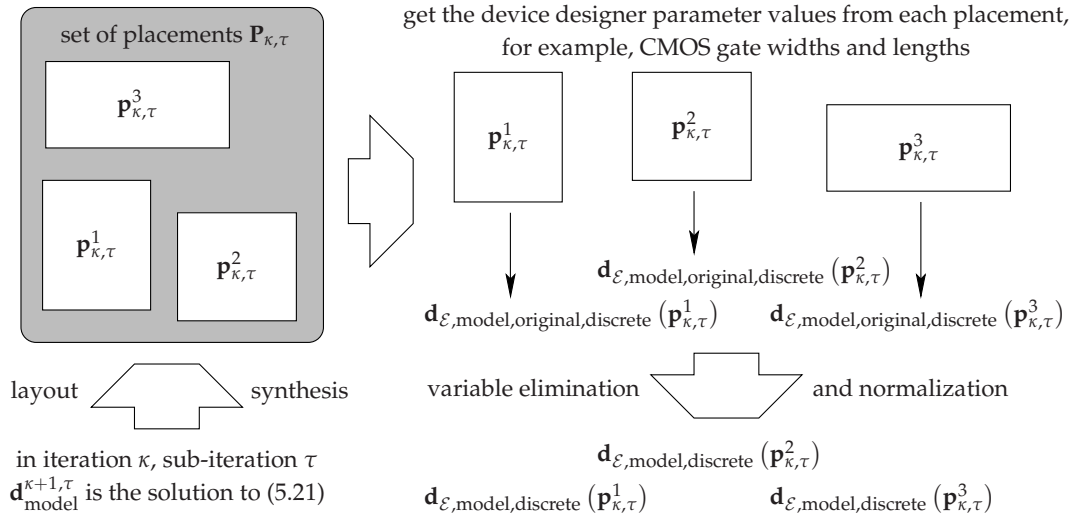


Figure 5.2: Illustration of the mapping in (5.82). The layout synthesis flow of Chapter 4 is used to generate the set of placements $\mathbf{P}_{\kappa,\tau} = \{\mathbf{p}_{\kappa,\tau}^1, \mathbf{p}_{\kappa,\tau}^2, \mathbf{p}_{\kappa,\tau}^3\}$; from $\mathbf{P}_{\kappa,\tau}$, a set of discrete placement-dependent circuit design parameter values are extracted.

The partially discrete solution is then constructed by adding the test bench design parameters:

$$\mathbf{d}_{\text{model,pd}}^{\kappa+1,\tau}(\mathbf{p}_{\kappa,\tau}) = \begin{bmatrix} \mathbf{d}_{\mathcal{E},\text{model,discrete}}(\mathbf{p}_{\kappa,\tau}) \\ \mathbf{d}_{\mathcal{E}\mathcal{B},\text{model}}^{\kappa+1,\tau} \end{bmatrix}; \quad (5.83)$$

$$\mathbf{d}_{\text{model,pd}}^{\kappa+1,\tau}(\mathbf{p}_{\kappa,\tau}) \in \mathcal{D}_{\text{pd}}; \quad \mathbf{p}_{\kappa,\tau} \in \mathbf{P}_{\kappa,\tau}$$

Thirdly, a *best* placement, denoted by $\mathbf{p}_{\kappa,\tau}^*$, is selected from $\mathbf{P}_{\kappa,\tau}$, and the continuous solution $\mathbf{d}_{\text{model}}^{\kappa+1,\tau}$ is replaced by $\mathbf{d}_{\text{model,pd}}^{\kappa+1,\tau}(\mathbf{p}_{\kappa,\tau}^*)$ as the *partially discrete* solution. The selection of $\mathbf{p}_{\kappa,\tau}^*$ is accomplished by looking at the performance values, as follows.

Using (5.77), the electrical performance values corresponding to $\mathbf{p}_{\kappa,\tau} \in \mathbf{P}_{\kappa,\tau}$ are calculated using the post-layout circuit:

$$\mathbf{f}_{e,\text{pd,ws}}(\mathbf{p}_{\kappa,\tau}) \stackrel{(5.77)}{=} \boldsymbol{\phi}_{\mathbf{f}_{e,\text{ws}}}(\mathbf{d}_{\text{model,pd}}^{\kappa+1,\tau}(\mathbf{p}_{\kappa,\tau})) \quad (5.84)$$

The difference between the placement-dependent electrical performance values $\{\mathbf{f}_{e,\text{pd,ws}}(\mathbf{p}_{\kappa,\tau}) \mid \mathbf{p}_{\kappa,\tau} \in \mathbf{P}_{\kappa,\tau}\}$ is due to discretization and placement-dependent error.

The objective function, γ_{κ} , of the search algorithm, defined in (5.10), is a function of the performances, and is minimized in problem (5.13). The best placement $\mathbf{p}_{\kappa,\tau}^*$ is, therefore, selected from $\mathbf{P}_{\kappa,\tau}$ so as to minimize γ_{κ} :

$$\begin{aligned}
 \mathbf{p}_{\kappa,\tau}^* &= \underset{\mathbf{p}_{\kappa,\tau} \in \mathbf{P}_{\kappa,\tau}}{\operatorname{argmin}} \left[\gamma_{\kappa} \left(\mathbf{d}_{\text{model,pd}}^{\kappa+1,\tau} (\mathbf{p}_{\kappa,\tau}) \right) \right] \\
 &\stackrel{(5.9)}{=} \underset{\mathbf{p}_{\kappa,\tau} \in \mathbf{P}_{\kappa,\tau}}{\operatorname{argmin}} \left[\exp \left(-\frac{A_{\max} - \hat{A}_{\text{ws}} (\mathbf{p}_{\kappa,\tau})}{\|\mathbf{J}_{\hat{A},\text{ws}} (\mathbf{d}^{\kappa})\|} \right) + \sum_{i=1}^{n_f} \exp \left(-\frac{\mathbf{f}^u [i] - \mathbf{f}_{e,\text{pd,ws}} (\mathbf{p}_{\kappa,\tau}) [i]}{\|\mathbf{J}_{\mathbf{f}e,\text{ws}} (\mathbf{d}^{\kappa}) [i]\|} \right) \right]
 \end{aligned} \tag{5.85}$$

This can be conveniently accomplished using the scalar cost metric φ suggested in Section 4.6 and used to select the final placement in the layout synthesis flow. To exactly match the expressions for γ_{κ} and φ , the weight vector \mathbf{w} in (4.92) is set as follows at the starting point of each algorithm iteration, κ :

$$\begin{aligned}
 \mathbf{w}[i] &= \frac{1}{\|\mathbf{J}_{\mathbf{f}e,\text{ws}} (\mathbf{d}^{\kappa}) [i]\|}; \quad i = 1, \dots, n_{\mathbf{f}e} \\
 \mathbf{w}[0] &= \frac{1}{\|\mathbf{J}_{\hat{A},\text{ws}} (\mathbf{d}^{\kappa})\|} \quad (\text{for modified area, } \hat{A})
 \end{aligned} \tag{5.86}$$

The Jacobian $\mathbf{J}_{\mathbf{f}e,\text{ws}} (\mathbf{d}^{\kappa})$ and the gradient $\mathbf{J}_{\hat{A},\text{ws}} (\mathbf{d}^{\kappa})$ are derived in Section 5.7.4.

Since $\mathbf{d}_{\text{model}}^{\kappa+1,\tau}$ is replaced by $\mathbf{d}_{\text{model,pd}}^{\kappa+1,\tau} (\mathbf{p}_{\kappa,\tau}^*)$ as the partially discrete subproblem solution, it is also used in lieu of $\mathbf{d}_{\text{model}}^{\kappa+1,\tau}$ in the reduction ratio defined in (5.22) and used to select the size of the trust region:

$$\rho = \frac{\left(\gamma_{\kappa} (\mathbf{d}^{\kappa}) - \gamma_{\kappa} \left(\mathbf{d}_{\text{model,pd}}^{\kappa+1,\tau} (\mathbf{p}_{\kappa,\tau}^*) \right) \right)}{\left(\gamma_{\kappa} (\mathbf{d}^{\kappa}) - \bar{\gamma}_{\kappa} \left(\mathbf{d}_{\text{model,pd}}^{\kappa+1,\tau} (\mathbf{p}_{\kappa,\tau}^*) \right) \right)}; \quad \rho \in (0, 1) \tag{5.87}$$

The geometric constraints, $\mathbf{c}_g \succeq \mathbf{c}_g^m$, are satisfied by layout construction. Post-layout satisfaction of the DC electrical constraints, $\mathbf{c}_e \succeq \mathbf{c}_e^m$, was ensured during synthesis by the procedure in Section 4.5.2. Therefore, if $\boldsymbol{\phi}_c \left(\mathbf{d}_{\text{model}}^{\kappa+1,\tau} \right) \succeq \mathbf{c}^m$, then the constraints \mathbf{c} are also satisfied post layout synthesis.

Fourthly, after the q_{κ} sub-iterations are completed and condition (5.87) is satisfied, the solution to (5.13), which is the starting vector of iteration $\kappa+1$, is assigned the discrete placement-dependent value:

$$\mathbf{d}^{\kappa+1} \leftarrow \mathbf{d}_{\text{model,pd}}^{\kappa+1,q_{\kappa}} (\mathbf{p}_{\kappa,q_{\kappa}}^*); \quad \mathbf{p}_{\kappa,q_{\kappa}}^* \in \mathbf{P}_{\kappa,q_{\kappa}}; \quad \mathbf{d}^{\kappa+1} \in \mathcal{D}_{\text{pd}} \tag{5.88}$$

Finally, it is noted here that the initial starting vector \mathbf{d}^0 (for $\kappa = 0$) is discretized in a similar manner using the layout synthesis flow, however the weight vector \mathbf{w} used in the selection of the initial best placement is set by the designer directly. Let $\mathbf{p}_{-1,-1}^*$ be the best placement for the initial continuous starting vector \mathbf{d}^0 :

$$\mathbf{d}^0 \xleftarrow{\text{reassigned}} \mathbf{d}_{\text{pd}}^0 (\mathbf{p}_{-1,-1}^*) \tag{5.89}$$

5.7.4 Partial Derivative Calculation Under Consideration of Discretization and Placement Error

In the search algorithm of Section 5.2, the performance partial derivatives are approximated at the starting vector, \mathbf{d}^κ , of each iteration, κ .

By (5.88), the starting vector, \mathbf{d}^κ , of iteration κ is set to $\mathbf{d}_{\text{model,pd}}^{\kappa,q_{\kappa-1}} \left(\mathbf{p}_{\kappa-1,q_{\kappa-1}}^* \right)$ and is already in the partially discrete design space, \mathcal{D}_{pd} . The value $\phi_{\text{fe,ws}}(\mathbf{d}^\kappa)$ has already been calculated in iteration $\kappa-1$ while selecting $\mathbf{p}_{\kappa-1,q_{\kappa-1}}^*$ in (5.85).

If the first-order forward difference approximation is used, as described in (5.41), then the approximation to $\mathbf{J}_{\text{fe,ws}}(\mathbf{d}^\kappa)[i,j]$ – the partial derivative of the i -th electrical performance to the j -th design parameter – requires a new vector ($\mathbf{d}^\kappa + h_{i,j} \cdot \mathbf{e}_j$) to be chosen in the design space and $\phi_{\text{fe,ws}}(\mathbf{d}^\kappa + h_{i,j} \cdot \mathbf{e}_j)[i]$ to be calculated.

There are several approaches to complete the partial difference approximation with respect to the circuit design parameters. The main difference between approaches depends on whether or not layout synthesis is repeated at ($\mathbf{d}^\kappa + h_{i,j} \cdot \mathbf{e}_j$), so that discretization error and placement dependency are accounted for in performance evaluation. Layout synthesis is computationally costly. In the approach described below, layout synthesis is used in partial derivative approximation, such that a total of n_{dE} calls are made to the synthesis flow of Chapter 4.

Let it be assumed that the step lengths are equal for all performance functions:

$$h_{-,j} = h_{1,j} = \dots = h_{i,j} = \dots = h_{n_{\text{fe}},j} \quad (5.90)$$

It will be shown later that this assumption reduces the number of needed calls to the layout synthesis flow from ($n_{\text{fe}} \cdot n_{\text{dE}}$) to n_{dE} .

By (5.90), the new vectors ($\mathbf{d}^\kappa + h_{i,j} \cdot \mathbf{e}_j$) taken for first-order forward difference are independent of performance index, i . Vector $\mathbf{d}^{\kappa,j}$ is now defined as follows:

$$\mathbf{d}^{\kappa,j} = \mathbf{d}^\kappa + h_{-,j} \cdot \mathbf{e}_j \quad (5.91)$$

Let $\mathbf{P}_{\kappa,j}$ denote the set of circuit placements corresponding to $\mathbf{d}^{\kappa,j}$. For each placement $\mathbf{p}_{\kappa,j}$ in $\mathbf{P}_{\kappa,j}$, the placement-dependent circuit design parameter vector $\mathbf{d}_{\mathcal{E},\text{discrete}}(\mathbf{p}_{\kappa,j})$ is obtained by applying (5.72):

$$\begin{array}{ccc} \mathbf{d}_{\mathcal{E}}^{\kappa,j} & \xrightarrow{(5.27), \text{ layout synthesis}} & \mathbf{P}_{\kappa,j} \\ & \text{discretization} \downarrow \text{ for } \mathbf{p}_{\kappa,j} \in \mathbf{P}_{\kappa,j} & \\ \mathbf{d}_{\mathcal{E},\text{discrete}}(\mathbf{p}_{\kappa,j}) & \xleftarrow{(5.26)} & \mathbf{d}_{\mathcal{E},\text{original,discrete}}(\mathbf{p}_{\kappa,j}) \end{array} \quad (5.92)$$

The complete discrete solution is then constructed:

$$\mathbf{d}_{\text{pd}}^{\kappa,j}(\mathbf{p}_{\kappa,j}) = \begin{bmatrix} \mathbf{d}_{\mathcal{E},\text{discrete}}(\mathbf{p}_{\kappa,j}) \\ \mathbf{d}_{\mathcal{E}\mathcal{B}}^{\kappa,j} \end{bmatrix}; \quad (5.93)$$

$$\mathbf{d}_{\text{pd}}^{\kappa,j}(\mathbf{p}_{\kappa,j}) \in \mathcal{D}_{\text{pd}}; \quad \mathbf{p}_{\kappa,j} \in \mathbf{P}_{\kappa,j}$$

Using (5.74), the discretization error $\mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}_{\kappa,j})$ is given implicitly by:

$$\underbrace{\begin{bmatrix} \mathbf{d}_{\mathcal{E},\text{discrete}}(\mathbf{p}_{\kappa,j}) \\ \mathbf{d}_{\mathcal{E}\mathcal{B}}^{\kappa,j} \end{bmatrix}}_{\mathbf{d}_{\text{pd}}^{\kappa,j}(\mathbf{p}_{\kappa,j})} \stackrel{(5.73)}{=} \underbrace{\begin{bmatrix} \mathbf{d}_{\mathcal{E}}^{\kappa,j} \\ \mathbf{d}_{\mathcal{E}\mathcal{B}}^{\kappa,j} \end{bmatrix}}_{\mathbf{d}^{\kappa,j}} + \begin{bmatrix} \mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}_{\kappa,j}) \\ \mathbf{0} \end{bmatrix} \quad (5.94)$$

$$\mathbf{d}_{\text{pd}}^{\kappa,j}(\mathbf{p}_{\kappa,j}) \stackrel{(5.91)}{=} \mathbf{d}^{\kappa} + h_{-,j} \cdot \mathbf{e}_j + \begin{bmatrix} \mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}_{\kappa,j}) \\ \mathbf{0} \end{bmatrix}$$

From (5.92), layout synthesis is only called when the circuit design parameters, $\mathbf{d}_{\mathcal{E}}$, are changed; this corresponds to $j = 1, \dots, n_{\mathcal{d}\mathcal{E}}$. The test bench design parameters do not directly alter layout geometry; thereby for $j = (n_{\mathcal{d}\mathcal{E}} + 1), \dots, n_{\mathcal{d}}$, the layouts produced for the iteration starting vector, \mathbf{d}^{κ} , are used:

$$\mathbf{d}^{\kappa,j} \quad \begin{cases} j \in [1, \dots, n_{\mathcal{d}\mathcal{E}}] \implies \text{obtain } \mathbf{P}_{\kappa,j} \text{ by synthesis as in (5.92)} \\ j \in [(n_{\mathcal{d}\mathcal{E}} + 1), \dots, n_{\mathcal{d}}] \implies \mathbf{P}_{\kappa,j} \leftarrow \mathbf{P}_{\kappa-1,q_{\kappa-1}} \text{ from iteration } \kappa-1 \end{cases} \quad (5.95)$$

From (5.95) and for $j = (n_{\mathcal{d}\mathcal{E}} + 1), \dots, n_{\mathcal{d}}$:

$$\begin{aligned} \mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}_{\kappa,j}) &= \mathbf{d}_{\text{pd}}^{\kappa,j}(\mathbf{p}_{\kappa-1,q_{\kappa-1}}) = \mathbf{0} \\ \mathbf{d}_{\text{pd}}^{\kappa,j}(\mathbf{p}_{\kappa,j}) &= \mathbf{d}_{\text{pd}}^{\kappa,j}(\mathbf{p}_{\kappa-1,q_{\kappa-1}}) \stackrel{(5.91)}{=} \mathbf{d}^{\kappa} + h_{-,j} \cdot \mathbf{e}_j \end{aligned} \quad (5.96)$$

Using (5.77) and considering

$$\mathbf{d}^0 \stackrel{(5.89)}{=} \mathbf{d}_{\text{pd}}^0(\mathbf{p}_{-1,-1}^*), \quad \mathbf{d}^{\kappa} \stackrel{(5.88)}{=} \mathbf{d}_{\text{model,pd}}^{\kappa,q_{\kappa-1}}(\mathbf{p}_{\kappa-1,q_{\kappa-1}}^*) \quad \text{for } \kappa = 1, \dots, m-1$$

the value of the performances at \mathbf{d}^{κ} and $\mathbf{d}_{\text{pd}}^{\kappa,j}(\mathbf{p}_{\kappa,j})$ for each $\mathbf{p}_{\kappa,j} \in \mathbf{P}_{\kappa,j}$ is defined:

$$\mathbf{f}_{e,\text{pd,ws}}^{\kappa}(\mathbf{p}_{\kappa-1,q_{\kappa-1}}^*) = \phi_{\mathbf{f}_{e,\text{ws}}}(\mathbf{d}^{\kappa}) \quad (5.97)$$

$$\mathbf{f}_{e,\text{pd,ws}}^{\kappa,j}(\mathbf{p}_{\kappa,j}) = \phi_{\mathbf{f}_{e,\text{ws}}}(\mathbf{d}_{\text{pd}}^{\kappa,j}(\mathbf{p}_{\kappa,j})) \quad (5.98)$$

Using (5.79), the change due uniquely to layout parasitic devices is defined:

$$\mathbf{f}_{e,\text{error},\Delta}^{\kappa}(\mathbf{p}_{\kappa-1,q_{\kappa-1}}^*) = \phi_{\mathbf{f}_{e,\text{ws}}}(\mathbf{d}^{\kappa}) - \phi_{\mathbf{f}_{e,\text{ws}}}(\mathbf{d}^{\kappa}) \quad (5.99)$$

$$\mathbf{f}_{e,\text{error},\Delta}^{\kappa,j}(\mathbf{p}_{\kappa,j}) = \boldsymbol{\phi}_{\mathbf{f}_{e,\text{ws}}}(\mathbf{d}_{\text{pd}}^{\kappa,j}(\mathbf{p}_{\kappa,j})) - \boldsymbol{\phi}_{\mathbf{f}_{e,\text{ws}}}(\mathbf{d}_{\text{pd}}^{\kappa,j}(\mathbf{p}_{\kappa,j})) \quad (5.100)$$

Let $\Delta\boldsymbol{\eta}_{\mathbf{f}_{e,\text{ws}},i,j}$ be the function representing computational error in the first-order forward difference of electrical performances calculated post layout synthesis. Similar to (5.48), the value of $\Delta\boldsymbol{\eta}_{\mathbf{f}_{e,\text{ws}},i,j}$ at \mathbf{d}^κ is calculated:

$$\begin{aligned} \Delta\boldsymbol{\eta}_{\mathbf{f}_{e,\text{ws}},i,j}(\mathbf{d}^\kappa) = & \widetilde{\mathbf{f}_{e,\text{pd,ws}}^{\kappa,j}(\mathbf{p}_{\kappa,j})}[i] - \mathbf{f}_{e,\text{pd,ws}}^\kappa(\mathbf{p}_{\kappa-1,q_{\kappa-1}}^*)[i] \\ & - \mathbf{f}_{e,\text{pd,ws}}^{\kappa,j}(\mathbf{p}_{\kappa,j})[i] + \mathbf{f}_{e,\text{pd,ws}}^\kappa(\mathbf{p}_{\kappa-1,q_{\kappa-1}}^*)[i] \end{aligned} \quad (5.101)$$

For any placement $\mathbf{p}_{\kappa,j} \in \mathbf{P}_{\kappa,j}$, the following equation can be derived for the first-order representation of the finite difference in the value of the i -th electrical performance between \mathbf{d}^κ and $\mathbf{d}_{\text{pd}}^{\kappa,j}(\mathbf{p}_{\kappa,j})$. The equation considers discretization, truncation, and computational error, as well as the error uniquely due to layout parasitic devices:

$$\begin{aligned} \mathbf{J}_{\mathbf{f}_{e,\text{ws}}}(\mathbf{d}^\kappa)[i] \cdot \overbrace{\left(\left[\begin{array}{c} \mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}_{\kappa,j}) \\ \mathbf{0} \end{array} \right] + h_{i,j} \cdot \mathbf{e}_j \right)}^{(\mathbf{d}_{\text{pd}}^{\kappa,j}(\mathbf{p}_{\kappa,j}) - \mathbf{d}^\kappa) \text{ by (5.94)}} = & \\ + \left(\widetilde{\mathbf{f}_{e,\text{pd,ws}}^{\kappa,j}(\mathbf{p}_{\kappa,j})}[i] - \mathbf{f}_{e,\text{pd,ws}}^\kappa(\mathbf{p}_{\kappa-1,q_{\kappa-1}}^*)[i] \right) \Big\} & \text{calculated value} \\ - \left(\mathbf{f}_{e,\text{error},\Delta}^{\kappa,j}(\mathbf{p}_{\kappa,j})[i] - \mathbf{f}_{e,\text{error},\Delta}^\kappa(\mathbf{p}_{\kappa-1,q_{\kappa-1}}^*)[i] \right) \Big\} & \text{placement dependency} \\ - \Delta\boldsymbol{\eta}_{\mathbf{f}_{e,\text{ws}},i,j}(\mathbf{d}^\kappa) \Big\} & \text{computational error} \\ - \frac{1}{2} \cdot \mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}_{\kappa,j})^T \cdot \frac{\partial^2 \boldsymbol{\phi}_{\mathbf{f}_{e,\text{ws}}}[i]}{\partial \mathbf{d}_{\mathcal{E}} \partial \mathbf{d}_{\mathcal{E}}^T}(\boldsymbol{\xi}^{\kappa,j}) \cdot \mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}_{\kappa,j}) \Big\} & \\ - \frac{h_{i,j}^2}{2} \cdot \frac{\partial^2 \boldsymbol{\phi}_{\mathbf{f}_{e,\text{ws}}}[i]}{\partial^2 \mathbf{d}[j]}(\boldsymbol{\xi}^{\kappa,j}) \Big\} & \text{truncation error} \end{aligned} \quad (5.102)$$

For some point $\boldsymbol{\xi}^{\kappa,j}$ on the line segment joining \mathbf{d}^κ and $\mathbf{d}_{\text{pd}}^{\kappa,j}(\mathbf{p}_{\kappa,j})$, and assuming that the underlying function $\boldsymbol{\phi}_{\mathbf{f}_{e,\text{ws}}}[i]$ is twice differentiable along this line segment.

The following equation can be derived from (5.102) to obtain the approximation $\mathbf{J}_{fe,ws}(\mathbf{d}^K)$ to $\mathbf{J}_{fe,ws}(\mathbf{d}^K)$, which includes placement dependency and corrects for discretization error:

$$\mathbf{J}_{fe,ws}(\mathbf{d}^K)[i] \cdot \overbrace{\left(\begin{bmatrix} \mathbf{d}_{\mathcal{E},error}(\mathbf{p}_{\kappa,j}) \\ \mathbf{0} \end{bmatrix} + h_{i,j} \cdot \mathbf{e}_j \right)}^{\left(\mathbf{d}_{pd}^{\kappa,j}(\mathbf{p}_{\kappa,j}) - \mathbf{d}^K \right) \text{ by (5.94)}} = \left(\widetilde{\mathbf{f}_{e,pd,ws}^{\kappa,j}(\mathbf{p}_{\kappa,j})[i]} - \widetilde{\mathbf{f}_{e,pd,ws}^{\kappa}(\mathbf{p}_{\kappa-1,q_{\kappa-1}^*})[i]} \right) \left. \vphantom{\mathbf{J}_{fe,ws}(\mathbf{d}^K)[i]} \right\} \text{calculated value} \quad (5.103)$$

For $j=1, \dots, n_d$ and $i=1, \dots, n_{fe}$, and since $h_{-,j}=h_{1,j}=\dots=h_{i,j}=\dots=h_{n_{fe},j}$ by the assumption in (5.90), $(n_{fe} \times n_d)$ equations of the form in (5.103) can be constructed and combined into a single system to calculate the complete Jacobian matrix $\mathbf{J}_{fe,ws}(\mathbf{d}^K)$:

$$\mathbf{J}_{fe,ws}(\mathbf{d}^K) \cdot \overbrace{\left(\begin{bmatrix} \mathbf{d}_{\mathcal{E},error}(\mathbf{p}_{\kappa,1}), \dots, \mathbf{d}_{\mathcal{E},error}(\mathbf{p}_{\kappa,n_d}) & \mathbf{0}_{\langle n_d \mathcal{E} \times n_d \mathcal{E} \mathcal{B} \rangle} \\ \mathbf{0}_{\langle n_d \mathcal{E} \mathcal{B} \times n_d \mathcal{E} \rangle} & \mathbf{0}_{\langle n_d \mathcal{E} \mathcal{B} \times n_d \mathcal{E} \mathcal{B} \rangle} \end{bmatrix} + \mathbf{H} \right)}^{\Delta D} \approx \underbrace{\left[\left(\widetilde{\mathbf{f}_{e,pd,ws}^{\kappa,1}(\mathbf{p}_{\kappa,1})} - \widetilde{\mathbf{f}_{e,pd,ws}^{\kappa}(\mathbf{p}_{\kappa-1,q_{\kappa-1}^*})} \right), \dots, \left(\widetilde{\mathbf{f}_{e,pd,ws}^{\kappa,n_d}(\mathbf{p}_{\kappa,n_d})} - \widetilde{\mathbf{f}_{e,pd,ws}^{\kappa}(\mathbf{p}_{\kappa-1,q_{\kappa-1}^*})} \right) \right]}_{\Delta F} \quad (5.104)$$

where

$$\mathbf{H} = \begin{bmatrix} h_{-,1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & h_{-,n_d} \end{bmatrix} \quad (5.105)$$

Placement $\mathbf{p}_{\kappa-1,q_{\kappa-1}^*}$ is the best from the previous iteration ($\kappa-1$), as selected in (5.85), and \mathbf{d}^K is the corresponding design parameter vector; from (5.97):

$$\mathbf{f}_{e,pd,ws}^{\kappa}(\mathbf{p}_{\kappa-1,q_{\kappa-1}^*}) = \boldsymbol{\phi}_{fe,ws}(\mathbf{d}^K) \quad (5.106)$$

For $j=1, \dots, n_d$, $\mathbf{d}_{pd}^{\kappa,j}(\mathbf{p}_{\kappa,j})$ is the discrete design parameter vector corresponding to $\mathbf{p}_{\kappa,j} \in \mathbf{P}_{\kappa,j}$ as defined in (5.93), and $\mathbf{d}_{\mathcal{E},error}(\mathbf{p}_{\kappa,j})$ is the discretization error in the circuit design parameters as defined in (5.94); from (5.98):

$$\mathbf{f}_{e,pd,ws}^{\kappa,j}(\mathbf{p}_{\kappa,j}) = \boldsymbol{\phi}_{fe,ws}(\mathbf{d}_{pd}^{\kappa,j}(\mathbf{p}_{\kappa,j})) \quad (5.107)$$

By the assumption in (5.90) and from the definition in (5.91), the value of $\mathbf{d}_{\text{pd}}^{\kappa,j}(\mathbf{p}_{\kappa,j})$ is independent of the performance index, i . From (5.104):

$$\mathbf{J}_{\text{fe,ws}}(\mathbf{d}^{\kappa}) \cdot \Delta \mathbf{D} \approx \Delta \mathbf{F} \quad (5.108)$$

In order to solve (5.108) for $\mathbf{J}_{\text{fe,ws}}(\mathbf{d}^{\kappa})$, the step sizes, $\{h_{-,1}, \dots, h_{-,j}, \dots, h_{-,n_{\mathbf{d}}}\}$, must be selected so that the square matrix $\Delta \mathbf{D}$ is nonsingular.

One method of selection is suggested here. By the Levy-Desplanques theorem, if $\Delta \mathbf{D}$ is a strictly diagonally dominant matrix, then it is nonsingular. The step sizes are therefore selected so that $\Delta \mathbf{D}$ is a strictly column diagonally dominant matrix:

$$|\mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}_{\kappa,j})[j] + h_{-,j}| > \sum_{k \neq j}^{n_{\mathbf{d}\mathcal{E}}} |\mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}_{\kappa,j})[k]|; \quad j = 1, \dots, n_{\mathbf{d}\mathcal{E}} \quad (5.109)$$

This requires each step, $h_{-,j}$, to be large relative to the first norm of the discretization error, $\mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}_{\kappa,j})$, for $j = 1, \dots, n_{\mathbf{d}\mathcal{E}}$ (the circuit design parameters), while no such requirement is imposed for $j = (n_{\mathbf{d}\mathcal{E}} + 1), \dots, n_{\mathbf{d}}$ (the test bench design parameters).

For the reasonable assumption that $\mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}_{\kappa,j})[j] + h_{-,j} > 0$:

$$h_{-,j} > \sum_{k \neq j}^{n_{\mathbf{d}\mathcal{E}}} |\mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}_{\kappa,j})[k]| - \mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}_{\kappa,j})[j]; \quad j = 1, \dots, n_{\mathbf{d}\mathcal{E}} \quad (5.110)$$

Using (5.80), it is possible to define a lower bound for the step size so as to *guarantee* that the matrix $\Delta \mathbf{D}$ is nonsingular:

$$h_{-,j} > \underbrace{\mathbf{1}_{n_{\mathbf{d}\mathcal{E}}}^T \cdot \mathbf{d}_{\mathcal{E},\text{error-max}}}_{\text{a constant}}; \quad j = 1, \dots, n_{\mathbf{d}\mathcal{E}}; \quad \mathbf{1}_{n_{\mathbf{d}\mathcal{E}}} \in \{1\}^{n_{\mathbf{d}\mathcal{E}}} \quad (5.111)$$

In (5.102), the step $h_{-,j}$ is proportional to truncation error. As discussed in Section 5.6, truncation error has an averaging effect on each partial difference approximation. The lower bound set in (5.111) on each step size, $h_{-,j}$ is very large and can overly smooth the Jacobian approximation due to truncation error. Important function extrema between \mathbf{d}^{κ} and $\mathbf{d}^{\kappa,j}$ may be missed in the resulting linear model. Furthermore, it is unlikely that the maximum error magnitude, $\mathbf{d}_{\mathcal{E},\text{error-max}}$, will be attained by any placement $\mathbf{p}_{\kappa,j}$ in $\mathbf{P}_{\kappa,j}$, so that $\text{abs}(\mathbf{d}_{\mathcal{E},\text{error}}(\mathbf{p}_{\kappa,j})) = \mathbf{d}_{\mathcal{E},\text{error-max}}$. In consequence, the lower bound on step size is set to a fraction, ϱ , of $(\mathbf{1}_{n_{\mathbf{d}\mathcal{E}}}^T \cdot \mathbf{d}_{\mathcal{E},\text{error-max}})$:

$$h_{-,j} > \varrho \cdot \mathbf{1}_{n_{\mathbf{d}\mathcal{E}}}^T \cdot \mathbf{d}_{\mathcal{E},\text{error-max}}; \quad j = 1, \dots, n_{\mathbf{d}\mathcal{E}}; \quad \mathbf{1}_{n_{\mathbf{d}\mathcal{E}}} \in \{1\}^{n_{\mathbf{d}\mathcal{E}}}; \quad 0 < \varrho \leq 1 \quad (5.112)$$

If $\Delta \mathbf{D}$ is singular after synthesis, then ϱ is increased and synthesis is repeated.

In (5.102), each step size is inversely proportional to the computational error and the effect of placement dependency. Computational error can be regarded as done in Section 5.6 for the partial derivative approximations without layout synthesis.

Similar to computational error, a large step size reduces the effect of placement dependency (and hence the local trends in layout geometry) to the approximation of each partial derivative. However, the quantified placement dependency can be unbounded in value – a circuit example was given in Figure 5.1. For best results, it cannot be ignored completely, for example, by performance evaluation without layout synthesis, or by the random selection of $\mathbf{p}_{\kappa,j}^*$ from $\mathbf{P}_{\kappa,j}$.

In consideration of the method in which the discrete solution, \mathbf{d}^κ , is found to (5.13), represented by equations (5.82) to (5.88), placement dependency is a benefit if it results in an improvement in the descent direction. By extending the linear model of the previous iteration $\kappa-1$, used to find \mathbf{d}^κ , to the vector $\mathbf{d}^{\kappa,j} = \mathbf{d}^\kappa + h_{-,j} \cdot \mathbf{e}_j$, a *best* placement $\mathbf{p}_{\kappa,j}^*$ is selected from $\mathbf{P}_{\kappa,j}$ so as to minimize $\gamma_{\kappa-1}$ over the values of $\mathbf{P}_{\kappa,j}$. This is, once more, accomplished using the scalar cost metric φ suggested in Section 4.6 and used to select the final placement in the layout synthesis flow. The Jacobian matrix from the previous iteration, $\mathbf{J}_{f_{e,ws}}(\mathbf{d}^{\kappa-1})$, is used to set the weight vector \mathbf{w} :

$$\begin{aligned} \mathbf{w}[i] &= \frac{1}{\|\mathbf{J}_{f_{e,ws}}(\mathbf{d}^{\kappa-1})[i]\|}; \quad i = 1, \dots, n_{fe} \\ \mathbf{w}[0] &= \frac{1}{\|\mathbf{J}_{\hat{A},ws}(\mathbf{d}^{\kappa-1})\|} \quad (\text{for modified area, } \hat{A}) \end{aligned} \quad (5.113)$$

5.8 On the Cost of Circuit Sizing

The preponderant cost of the search algorithm in Section 5.2 is the evaluation of electrical performances and constraints by circuit simulation. In layout-driven circuit sizing, the cost of layout synthesis must also be considered. Therefore, the number of performance and constraint evaluations needed, as well as the number of calls to the layout synthesis flow of Chapter 4 will be used as a measure of circuit sizing cost. An account of the number of calls and evaluations is given below.

First, the performances and constraints are evaluated at the starting vector, \mathbf{d}^0 , of the first search algorithm iteration; a layout is synthesized in layout-driven sizing.

Secondly, performances and constraints are linearized at the starting vector, κ , of each algorithm iteration to obtain the subproblem in (5.20), with $(\kappa = 0, \dots, m-1)$. If a first-order forward difference function is used to approximate the Jacobian of the electrical performances and constraints, then n_d electrical performance and constraint evaluations are needed in each iteration. In layout-driven circuit sizing, the synthesis flow is called for each circuit design parameter – for a total of n_{dE} calls in each iteration.

Thirdly, in order to select the optimal trust region, the problem in (5.21) is solved q_κ times in iteration κ , and the reduction ratio in (5.22) is evaluated. This requires a total of q_κ performance evaluations. In layout-driven circuit sizing, q_κ calls are made to the synthesis flow. Typically, $3 \leq q_\kappa \leq 10$, in the circuit sizing examples of Chapter 6.

Fourthly, if a feasible correction step is necessary in iteration κ , sub-iteration τ , then an additional $n_{\mathbf{d}}+1$ constraint evaluations and one additional performance evaluation are needed. Let r_{κ} denote the number of times a feasible correction step is needed in iteration κ .

From the account above, a tally of the number of electrical performance and constraint evaluations as well as calls to the layout synthesis flow is given below:

$$\text{number of performance evaluations} = 1 + m \cdot n_{\mathbf{d}} + \sum_{\kappa=0}^{m-1} (q_{\kappa} + r_{\kappa}) \quad (5.114)$$

$$\text{number of constraint evaluations} = 1 + m \cdot n_{\mathbf{d}} + (n_{\mathbf{d}} + 1) \cdot \sum_{\kappa=0}^{m-1} (r_{\kappa}) \quad (5.115)$$

$$\text{number of calls to layout synthesis} = 1 + m \cdot n_{\mathbf{d}\mathcal{E}} + \sum_{\kappa=0}^{m-1} q_{\kappa} \quad (5.116)$$

5.9 Summary

In this Chapter, a new procedure is presented to solve the circuit sizing problem formulated in (2.39). It combines a deterministic search algorithm with the constraint-based layout synthesis flow presented in Chapter 4. The outcome of circuit sizing is a layout that meets all the geometric constraints and specifications and a corresponding electrical model that meets all the electrical constraints and specifications.

Electrical performances and constraints are evaluated numerically by circuit simulation; this contributes a computational error. The search algorithm is gradient-based. Each partial derivative of the electrical performance and constraint functions is approximated a forward finite difference function; this contributes a truncation error.

Layout synthesis adds a discretization error and a placement-dependency to the value of each performance. Discretization error is introduced by layout synthesis when continuous device parameters are mapped to discontinuous layout parameters. Placement-dependency is understood to be uniquely due to layout parasitic devices. Discretization error is bounded by constraints placed on device layout, while placement-dependency can be large and unbounded, as shown in Figure 5.1. This is because the change in device location in the placement, the differences in routing, and other layout specific attributes are hidden and unaccounted for when mapping from the design space to the performance space.

In order to account for the sources of error due to numerical evaluation and layout synthesis, adjustments are made to the standard search algorithm. The upper performance specification bound is increased by an extra margin to account for the effect of computational error. Computational error is also used to set a lower bound

on the magnitude of partial derivatives. In each search algorithm iteration, the best placement is selected from the set generated by the layout synthesis flow so that the solution to (5.13) is minimized. In (5.104), the discretization error term is accounted for in the system of linear equations used in Jacobian approximation.

The principle computational cost of circuit sizing is the numerical evaluation (simulation) of electrical performances and constraints. In layout-driven circuit sizing, the cost of calling the layout synthesis flow of Chapter 4 is also significant. The cost of executing the search algorithm code is relatively small and can be neglected. Equations are derived for the number of performance and constraint evaluations that are necessary, as well as the number of calls to the layout synthesis flow.

Chapter 6

Circuit Sizing Examples

In this chapter the layout-driven circuit sizing algorithm described in Chapter 5 is applied to size three low frequency CMOS example circuits. Results are compared to the outcome of traditional circuit sizing without the integration of layout synthesis.

The example circuits are first described in Section 6.1. In Section 6.2, the general implementation details are described for the experimental setup. Experimental results for each example circuit are given in Section 6.3.

6.1 Description of the Example Circuits

This section gives an overview of the example circuits, as well as the number of design parameters, inequality constraints, and performances. The layout constraints, used for layout-driven sizing, are also described for each example.

6.1.1 Folded Cascode Operational Amplifier (FC-OA)

The first example is the CMOS folded cascode operational amplifier (FC-OA) with the circuit topology shown in Figure 6.1. The FC-OA consists of 19 CMOS devices.

Each CMOS device has two design parameters, total gate width and length ($\mathbf{d}_{\text{CMOS}} = [W, L]$), for a total of 38 circuit design parameters. An externally supplied bias current, labeled I_{bias} , is the only test bench design parameter.

Each device must operate as a voltage controlled current source and be in the saturation region of operation. Ten analog functional circuit sub-blocks are identified in the FC-OA topology and revealed in Figure 6.2: 4 current mirrors, 3 level shifters, 2 differential pairs, and 1 cascode current mirror. The device tuples (P8,P3,P6,N2,N4,N7) and (P7,P4,P5,N3,N5,N6) constitute the two branches of a balanced differential signal path; the design parameters of corresponding devices are matched.

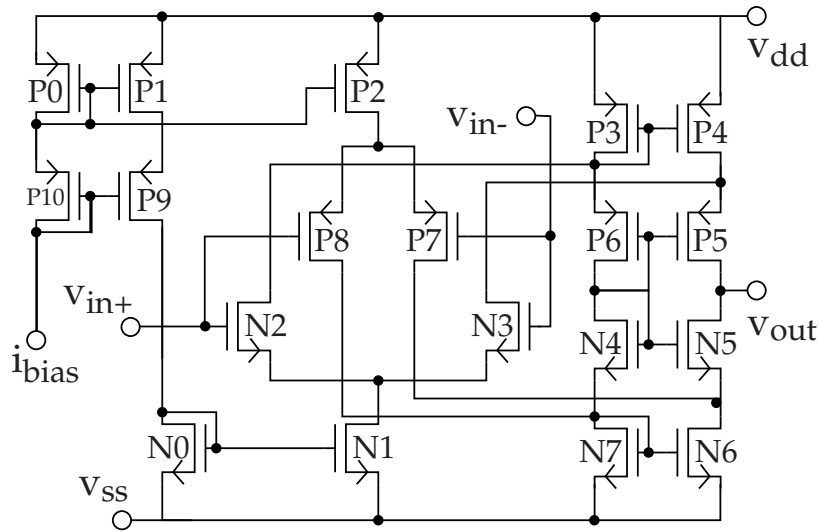


Figure 6.1: Folded cascode operational amplifier (FC-OA) topology.

Circuit sizing rules are specified from the device region of operation, the functional circuit sub-blocks, and circuit symmetry, as explained in Section 2.1.5 and [GZEA01]. There are 90 electrical inequality constraints, 57 geometric inequality constraints, and 26 geometric equality constraints.

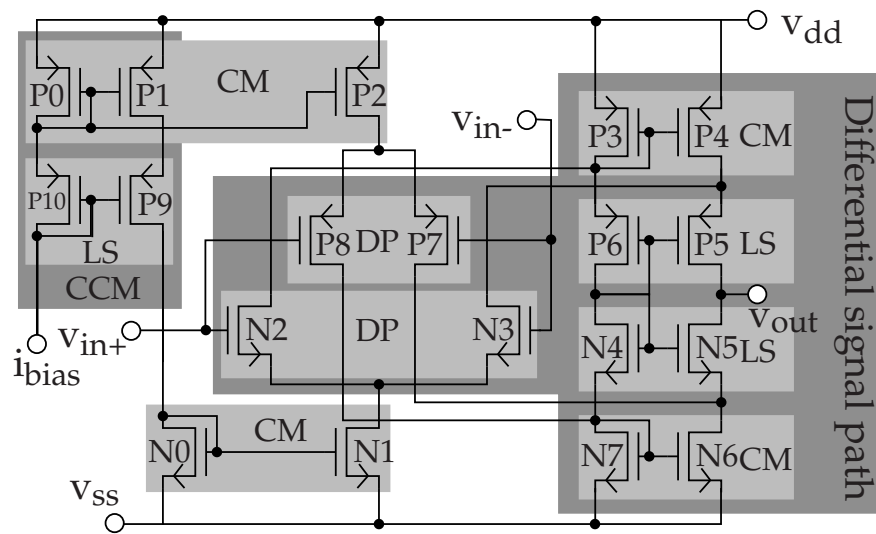
By using the geometric equality constraints and applying elimination methods to the inequality constraints, the number of electrical inequality constraints is reduced to $n_{ce} = 56$, the number of geometric inequality constraints is reduced to $n_{cg} = 27$, and the number of circuit design parameters is reduced to $n_{d\epsilon} = 22$. The total number of design parameters is $n_d = 23$.

The placement constraints for the FC-OA are overlaid on the circuit topology in Figure 6.3. The devices along the differential signal path are placed symmetrically and in proximity, while the bias circuit can be placed independently. The device pairs (P3,P4), (P6,P5), (N4,N5), (N7,N6), (P8,P7), and (N2,N3) are placed in common centroid configuration to improve matching along the differential signal path. As a result, 14 proximity, 6 common centroid, and 3 symmetry constraints are imposed according to the rules described in [ESGS10]. The minimum margins between each pair of devices constitute additional placement constraints.

Devices N2 to N7, and P3 to P8 are placed in grounded guard rings, while bulk taps are used to ground the remaining devices.

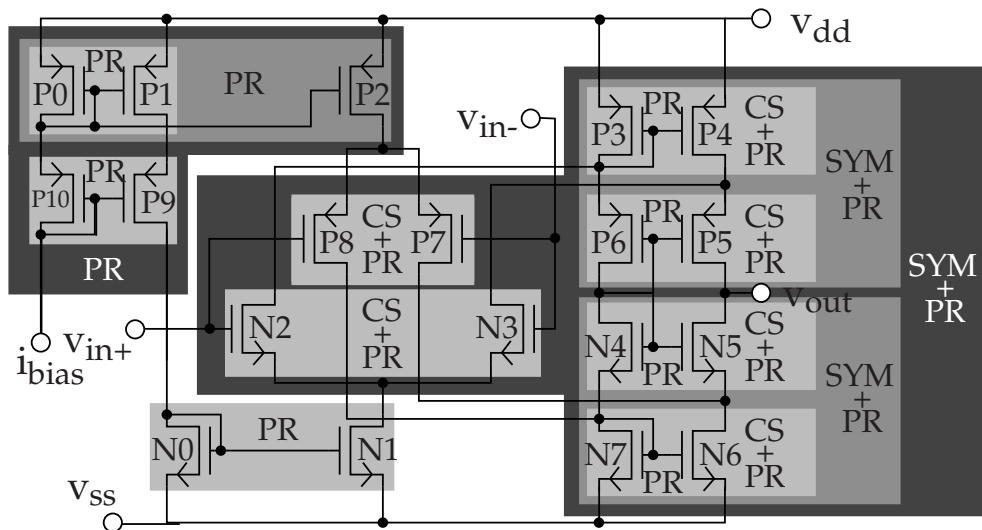
Interface pin location for the input and output signals, v_{in+} , v_{in-} , and v_{out} ; the DC bias current, i_{bias} ; and the DC supply and ground potentials, v_{dd} and v_{ss} ; is fixed to a location on the layout boundary prior to the initiation of layout synthesis.

Two metal layers are used for routing. Geometric routing constraints are set to meet DRC rules. The minimum wire width, the maximum wire length, the number of vias



CM: Current mirror, DP: Differential pair,
LS: Level shifter, CCM: Cascode current mirror

Figure 6.2: FC-OA topology with analog functional sub-blocks identified.



PR: Proximity, CS: Common centroid, SYM: Symmetry

Figure 6.3: FC-OA topology with superimposed placement constraints.

and corners is specified for each connection between devices. Maximum parallel and tandem wire separation and length is also specified. The maximum resistance and load capacitance of each connection between devices is also specified, as is the maximum coupling capacitance between critical nodes. For symmetrically placed devices, a preference for symmetrical signal routing paths is registered. Resistance and load capacitance is specified to be matched along the two branches of the differential signal path. The effective resistance between the terminal of each topology vertex $v \in \mathcal{V}$

Table 6.1: FC-OA test benches and electrical performances

Test Bench	Analysis	Electrical Performances Simulated	Abbreviation
TB_1	DC	Electrical Constraints	c_e
TB_2	AC	low frequency gain phase margin unity gain bandwidth	Gain PM UGBW
TB_3	AC	common mode rejection ratio power supply rejection ratio from v_{dd} power supply rejection ratio from v_{ss}	CMRR PSRR- v_{dd} PSRR- v_{ss}
TB_4	Transient	slew rate for rising signal settling time	SR-Rising Settling
TB_5	Transient	slew rate for falling signal	SR-Falling
TB_6	DC	input offset voltage power consumption	IOV Power
TB_7	PSS	total harmonic distortion	THD

is dependent upon on the number of fingers n_f and the number of divisions M for each device, (for common centroid pairs) and is determined after placement.

Seven test bench circuits are constructed to calculate 12 electrical performances in addition to the DC electrical constraints, as shown in Table 6.1.

6.1.2 Tunable Operational Transconductance Amplifier (TOTA)

The second example is the CMOS tunable operational transconductance amplifier (TOTA) with circuit topology shown in Figure 6.4. It is based on the circuit in [LHI09] with some topology modifications and it consists of 52 CMOS devices.

Operational amplifiers A+ and A- are identical. When combined with the bias circuit, each represents an instance of the FC-OA of Section 6.1.1

There is a total of 104 circuit design parameters (CMOS gate widths and lengths). An externally supplied DC bias current, I_{bias} , common mode reference voltage, V_{cm} , and tuning current, $I_{tune,low}$ are the three test bench design parameters.

Devices P5, N3, and N6 operate as voltage controlled resistors and must be in the triode region of operation. The channel length of N3 and N6 is constrained to have a large minimum; this is to reduce short channel effects on the linearity of voltage to current conversion in these devices. Minimum width, length, and area constraints

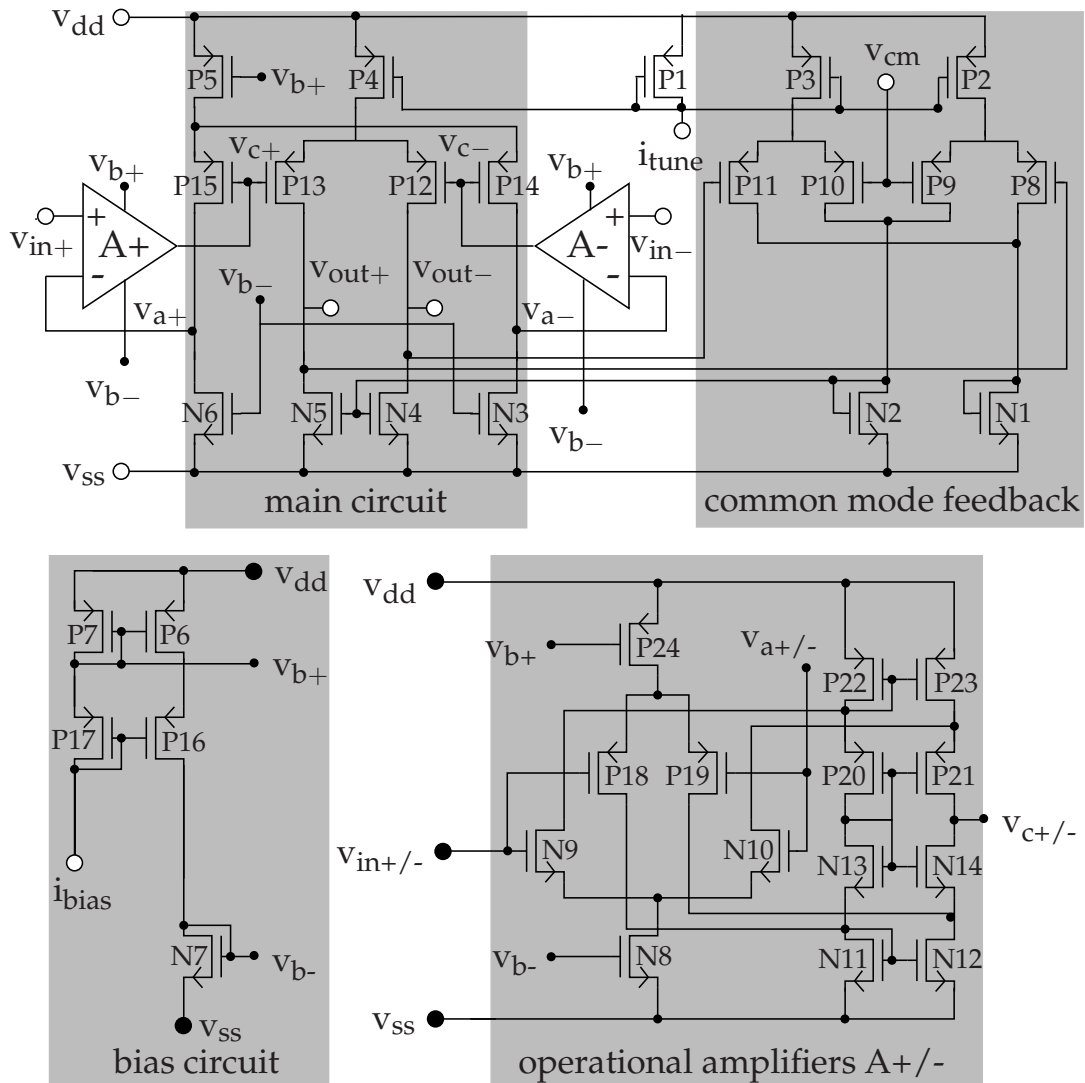


Figure 6.4: Tunable operational transconductance amplifier (TOTA) topology.

are also set on N3 and N6 to reduce drain current mismatch due to process parameter variations, and to reduce the amount of flicker noise.

Devices P8, P9, P10, P11, P12, and P13 are allowed to swing from weak to strong inversion but with the drain-to-source voltage, V_{ds} , of each device satisfying $V_{ds} \gg V_T$, where V_T is the thermal voltage. The remaining devices operate as voltage controlled current sources and must be in the saturation region of operation.

The voltages V_{ds} and V_{gs} (gate-to-source voltage) across each device pair; (P8,P9), (P10,P11), (P12,P13), (N6,N3), and (N1,N2); are matched.

Nineteen analog functional circuit sub-blocks are identified in the TOTA topology for the CMOS devices in saturation: 8 current mirrors, 5 level shifters, 5 differential pairs, and 1 cascode current mirror.

Device design parameters are matched in value along balanced differential signal paths. Within operational amplifiers A+ and A-, the design parameters of corresponding devices in the tuples (P22,P20,P18,N9,N11,N13) and (P23,P21,P19,N10,N12,N14) are matched. Within the main circuit, the design parameters of corresponding devices in the tuples (P12,P14,N4,N3) and (P13,P15,N5,N6) are matched. Within the common feedback circuit, the design parameters of the devices P8, P9, P10, P11 are set equal, as are the parameters of P2 and P3, and of N1 and N2.

Circuit sizing rules are specified from the device region of operation, the functional circuit sub-blocks, and circuit symmetry, as explained in Section 2.1.5 and [GZEA01]. There are 244 electrical inequality constraints, 129 geometric inequality constraints, and 107 geometric equality constraints.

Four additional DC electrical constraints are added to ensure sufficiently high gain and low input offset voltage (IOV) for A+ and A-. Referring to Figure 6.2:

$$\begin{aligned} -V_{\text{bound}} &\leq V_{\text{in}+} - V_{a+} \leq V_{\text{bound}} \\ -V_{\text{bound}} &\leq V_{\text{in}-} - V_{a-} \leq V_{\text{bound}} \end{aligned} \quad (6.1)$$

By using the geometric equality constraints and applying elimination methods to the inequality constraints, the number of electrical inequality constraints is reduced to $n_{ce} = 183$, the number of geometric inequality constraints is reduced to $n_{cg} = 45$, and the number of circuit design parameters is reduced to $n_{d\mathcal{E}} = 39$. The total number of design parameters is $n_d = 42$.

The main, bias, and CMFB circuits, as well as A+ and A- are placed concurrently. In order to route symmetrically, balance signal paths, and ensure electrical matching the main and CMFB circuit devices are placed symmetrically or in common centroid configuration, and in close layout proximity. In addition to the FC-OA placement constraints of Section 6.1.1, corresponding devices in A+ and A- are placed symmetrically. This is possible to accomplish because device dimensions have already been matched by geometric equality constraints, as discussed above. In all, 14 symmetry constraints, 33 proximity constraints, and 7 common centroid constraints are imposed on placement. The NMOS devices are also placed within a grounded guard ring, while bulk taps are used to ground the PMOS devices.

Interface pin location for the input and output signals, $V_{\text{in}+}$, $V_{\text{in}-}$, $V_{\text{out}+}$, $V_{\text{out}-}$; as well as the DC supply and ground potentials, V_{dd} , V_{ss} ; is fixed prior to the initiation of layout synthesis. Pin location for the DC common mode voltage level, V_{cm} ; the bias current, i_{bias} ; and the DC tuning current, i_{tune} ; is remains to be selected on the placement border by the routing algorithm

Routing constraints are specified in a similar manner to the first circuit example in Section 6.1.1. Symmetric routing is made possible by the strict geometric device design and circuit placement constraints.

Five test bench circuits are constructed to calculate 13 electrical performances in addition to the DC electrical constraints, as shown in Table 6.2.

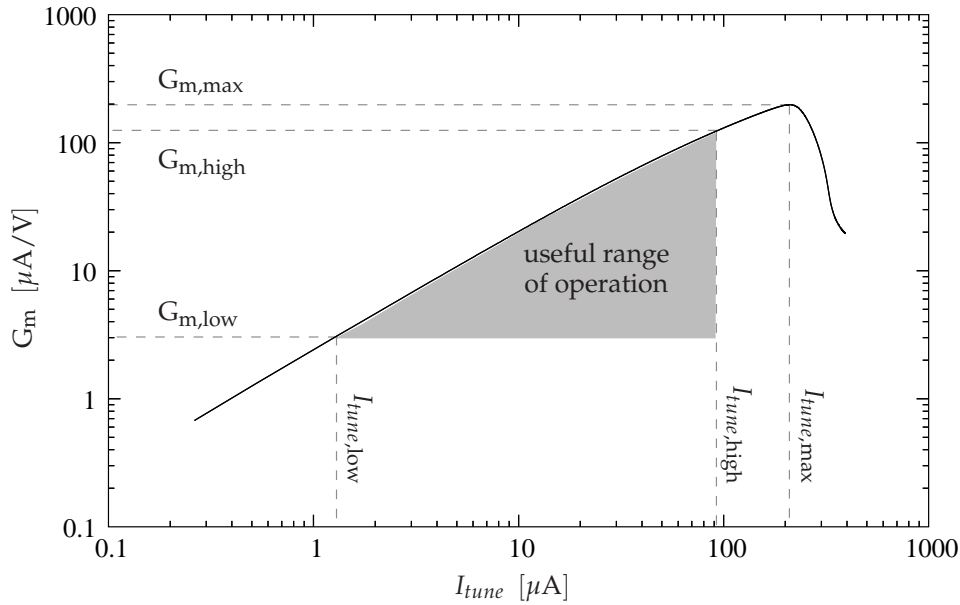


Figure 6.5: TOTA transconductance, G_m , versus tuning current, I_{tune} for a correctly sized circuit. The specifications in (6.2) as well as the electrical constraints are satisfied in the useful range of operation.

The TOTA linearly transforms a differential input voltage signal, $V_{in,diff}$, to a differential output current signal, $V_{out,diff}$, by a transconductance factor of magnitude G_m . In turn, G_m is controlled through the DC tuning current, I_{tune} , applied at pin i_{tune} .

In the feasible region of circuit operation – reached by correct circuit sizing – G_m increases monotonically with I_{tune} , as illustrated in Figure 6.5. The transconductance magnitude, $G_{m,max}$, and corresponding tuning current, $I_{tune,max}$, denote the end of the feasible operating region.

To be useful in wide tuning range filter design, the feasible region of operation, defined in terms of G_m and I_{tune} , must satisfy a set of additional specifications. The transconductance magnitudes, $G_{m,low}$ and $G_{m,high}$, and corresponding tuning currents $I_{tune,low}$ (a design parameter) and $I_{tune,high}$, define the useful range of operation that must be met. In the example of Figure 6.5, the useful range of operation corresponds to the following set of specifications:

$$\begin{aligned} 1\mu\text{A} &\leq I_{tune,low} \leq I_{tune,high} \leq 100\mu\text{A}; \\ G_{m,high} &= 40 \cdot G_{m,low}; \\ 1\mu\text{A/V} &\leq G_{m,low} \leq G_{m,high} \leq G_{m,max} \end{aligned} \quad (6.2)$$

The electrical inequality constraints should be satisfied for $I_{tune} \in [I_{tune,low}, I_{tune,high}]$:

$$\mathbf{c}_{e,min}[i] = \min_{I_{tune,low} \leq I_{tune} \leq I_{tune,high}} \mathbf{c}_e[i]; \quad i = 1, \dots, n_{ce}; \quad \mathbf{c}_{e,min} \succeq \mathbf{c}_e^m \quad (6.3)$$

In this manner, the additional tuning current, $(I_{tune,high} - I_{tune,low})$, beyond the design value, $I_{tune,low}$, is considered a circuit operating parameter, as defined in Section 2.1.3.

From the design of the CMFB circuit, the DC common-mode output voltage should equal the common mode reference voltage, V_{cm} , applied at pin v_{cm} :

$$V_{out,cm} = \frac{V_{out+} + V_{out-}}{2} = V_{cm} \quad (6.4)$$

Otherwise, a common mode offset adjustment would be required.

A performance is added to measure the common mode voltage at the output and compare it with the intended output common mode voltage, V_{cm} . As with the electrical constraints, the worst-case offset is measured for $I_{tune} \in [I_{tune,low}, I_{tune,high}]$.

To avoid output signal compression at a high differential input signal magnitude, the transconductance, G_m , must remain a constant factor over the differential input signal swing range.

The maximum differential input voltage magnitude is selected, here, to be 0.75V. A linearity measure is defined as the percentage change in the value of G_m when $V_{in,diff}$ changes from 0 to 0.75V:

$$\text{Linearity measure} = 100 \cdot \left| \frac{G_m(V_{in,diff} = 0 \text{ V}) - G_m(V_{in,diff} = 0.75 \text{ V})}{G_m(V_{in,diff} = 0 \text{ V})} \right| \quad (6.5)$$

For the TOTA in the feasible range of operation, the linearity measure becomes monotonically worse from $I_{tune} = I_{tune,low}$ to $I_{tune} = I_{tune,high}$; it is calculated for $I_{tune} = I_{tune,high}$.

From [PT03], the input referred noise power spectrum density of an OTA is given by:

$$S_n(f) = S_t/G_m + S_f/f; \quad f \text{ is frequency in Hz} \quad (6.6)$$

The thermal, S_t , and flicker, S_f , components depend on the technology and OTA topology. As flicker noise was a considerable problem for this OTA, a specification is set on the root mean square (RMS) input referred noise (IRN) from 1 to 500 kHz.

When used in filter design, the finite low frequency gain of the OTA will result in a passband loss (instead of an ideal 0dB), therefore voltage gain and bandwidth for the unloaded OTA must also be considered.

Noise, gain, and bandwidth are simulated at the corners for which $I_{tune} = I_{tune,low}$ and $I_{tune} = I_{tune,high}$.

According to [LHI09], there is a tradeoff between OTA noise, linearity, and power consumption.

The simulations of test bench \mathcal{TB}_1 must be executed before test benches \mathcal{TB}_2 to \mathcal{TB}_5 , in order to calculate the tuning current $I_{tune,high}$ corresponding to $G_{m,high}$.

Table 6.2: TOTA test benches and electrical performances

Test Bench	Analysis	Electrical Performances Simulated	Abbreviation
TB_1^*	AC, DC, parameter sweep of I_{tune}	smallest transconductance gain for $I_{tune} = I_{tune,low}$	$G_{m,low}$
		greatest transconductance gain for $I_{tune} = I_{tune,high}$	$G_{m,high}$
		power consumption for $I_{tune} = I_{tune,low}$	Power@1×
		power consumption for $I_{tune} = I_{tune,high}$	Power@40×
		output offset	$(V_{out,cm} - V_{cm})$
TB_2	DC	Electrical Constraints	c_e
TB_3	DC (sweep)	linearity measure defined in (6.5) for $I_{tune} = I_{tune,high}$	Linearity@40×
TB_4	AC	RMS input referred noise (1–500) kHz for $I_{tune} = I_{tune,low}$	IRN@1×
		low frequency voltage gain for $I_{tune} = I_{tune,low}$	Gain@1×
		bandwidth for $I_{tune} = I_{tune,low}$	BW@1×
TB_5	AC	RMS input referred noise (1–500) kHz for $I_{tune} = I_{tune,high}$	IRN@40×
		low frequency voltage gain for $I_{tune} = I_{tune,high}$	Gain@40×
		bandwidth for $I_{tune} = I_{tune,high}$	BW@40×

* TB_1 must be called before TB_2 to TB_5 so as to calculate $G_{m,high}$, $I_{tune,high}$.

6.1.3 Miller Operational Amplifier (MOA)

The third example is the CMOS miller operational amplifier (MOA) with circuit topology shown in Figure 6.6. It consists of 8 CMOS devices and a polysilicon-to-polysilicon capacitor.

There is a total of 17 circuit design parameters, the gate widths and lengths of the CMOS devices and the capacitance of C0, denoted by C. An externally supplied DC bias current, labeled I_{bias} , is the only test bench design parameter.

Each CMOS device must operate as a voltage controlled current source and be in the saturation region of operation. Three analog functional circuit sub-blocks are

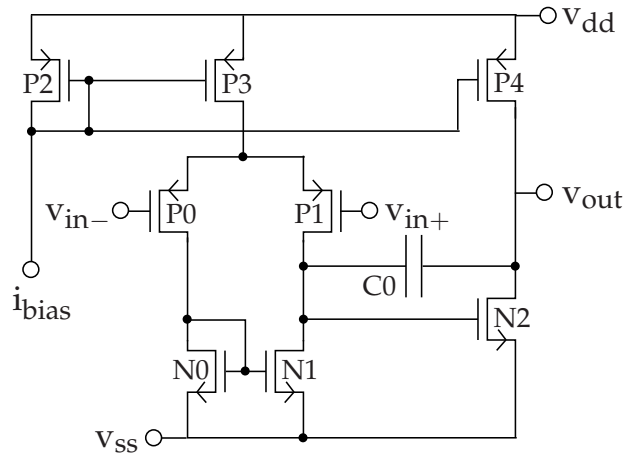


Figure 6.6: Miller operational amplifier (MOA) topology.

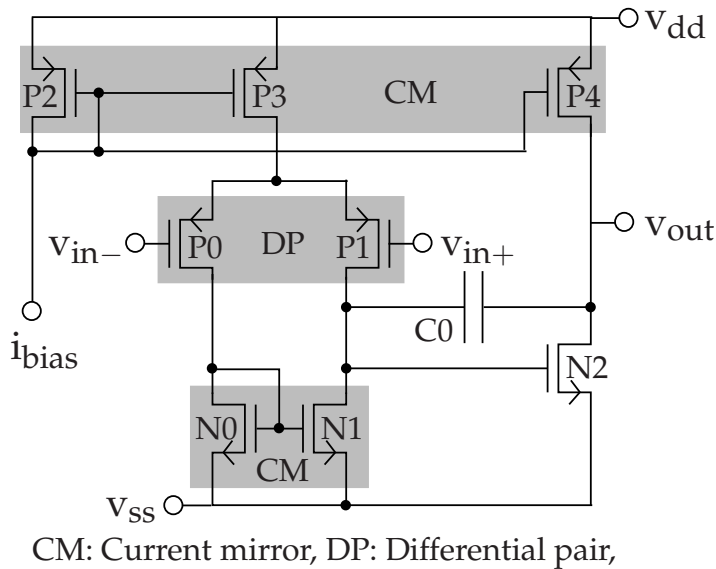


Figure 6.7: MOA topology with analog functional sub-blocks identified.

identified in the MOA topology and revealed in Figure 6.7: two current mirrors and one differential pair. The width of devices N0 and N1 must be equal to balance the drain currents of the differential pair. Circuit sizing rules for the CMOS devices are specified from the device region of operation and the functional circuit sub-blocks as explained in Section 2.1.5 and [GZEA01].

As is the case with CMOS device design parameters, statistical variations and manufacturing grid alignment will change the effective capacitance of C0 from the value selected during circuit sizing. In consequence, the value of circuit performances can change and robustness to manufacturing variations is reduced. Suitable sizing rules and placement constraints are derived below for C0.

A polysilicon-polysilicon capacitor is formed by laying a second plate of polysilicon over gate polysilicon. In this circuit example, C0 is placed as a single large capacitor, since matching is not an issue. Under nominal temperature and voltage bias conditions the capacitance, C , of C0 is modeled by the following equation:

$$C = C_A \cdot (W \cdot L) + C_F \cdot (2(W + L)) = C_A \cdot A + 2 \cdot C_F \cdot \sqrt{A} \cdot \left(\sqrt{As} + \frac{1}{\sqrt{As}} \right) \quad (6.7)$$

where C_A is the capacitance per unit area and C_F is the fringe capacitance per unit length, while W and L are the dimensions of the top polysilicon plate. Capacitor area, $A = W \cdot L$, and aspect ratio, $As = W/L$, are selected as the device layout parameters.

Let W_{step} and L_{step} denote the minimum increment steps for the dimensions of the top plate, and let W_{discrete} and L_{discrete} denote the dimensions after grid alignment:

$$W_{\text{discrete}} = \left\lfloor \frac{W}{W_{\text{step}}} \right\rfloor \cdot W_{\text{step}}; \quad L_{\text{discrete}} = \left\lfloor \frac{L}{L_{\text{step}}} \right\rfloor \cdot L_{\text{step}}; \quad (6.8)$$

$$C_{\text{discrete}} = C_A \cdot (W_{\text{discrete}} \cdot L_{\text{discrete}}) + C_F \cdot (2(W_{\text{discrete}} + L_{\text{discrete}}))$$

If $W \gg W_{\text{step}}$ and $L \gg L_{\text{step}}$, then $(C - C_{\text{discrete}})$ is relatively small and can be neglected.

Process parameters C_A and C_F and plate dimensions W and L have a statistical component; they vary about their nominal values with standard deviations σ_{C_A} , σ_{C_F} , σ_W , and σ_L respectively. Assuming the statistical components are uncorrelated and have suitably small coefficients of variation [Goo60], the variation in capacitance, C , can be derived as follows from (6.7):

$$\frac{\sigma_C^2}{C^2} \approx \frac{1}{\left(1 + 2 \cdot \frac{C_F}{C_A} \cdot \frac{1}{\sqrt{A}} \cdot \left(\sqrt{As} + \frac{1}{\sqrt{As}} \right) \right)^2} \cdot \left(\frac{\sigma_{C_A}^2}{C_A^2} + \sigma_W^2 \cdot \frac{1}{A \cdot As} + \sigma_L^2 \cdot \frac{As}{A} + 4 \cdot \frac{\sigma_{C_F}^2}{C_A^2} \cdot \frac{(As + \frac{1}{As})}{A} + 4 \cdot (\sigma_W^2 + \sigma_L^2) \cdot \frac{C_F^2}{C_A^2} \cdot \frac{1}{A^2} \right) \quad (6.9)$$

Most terms in (6.9) can be discarded; this is dependent upon the nominal and statistical parameters of the manufacturing process that is used. In general, to minimize the variation in C and reduce the contribution of fringe capacitance, capacitor area, A , should be suitably large, while the aspect ratio As should be kept close to 1. A well proportioned capacitor will also help in creating a compact circuit placement.

$$A_{\min} \leq A; \quad As_{\min} \leq As \leq As_{\max} \quad (6.10)$$

From (6.7), minimum capacitance for a fixed area is obtained when $As = 1$. Using this knowledge, the placement constraint $A_{\min} \leq A$ can be replaced by a device sizing rule applied directly to the design parameter C during circuit sizing:

$$A_{\min} \leq A \xrightarrow{(6.7)} \underbrace{C_A \cdot A_{\min} + 4 \cdot C_F \cdot \sqrt{A_{\min}}}_{C_{\min}} \leq \underbrace{C_A \cdot A + 4 \cdot C_F \cdot \sqrt{A}}_{\text{the value of } C \text{ when } As=1} \leq C \quad (6.11)$$

$$\implies C_{\min} \leq C$$

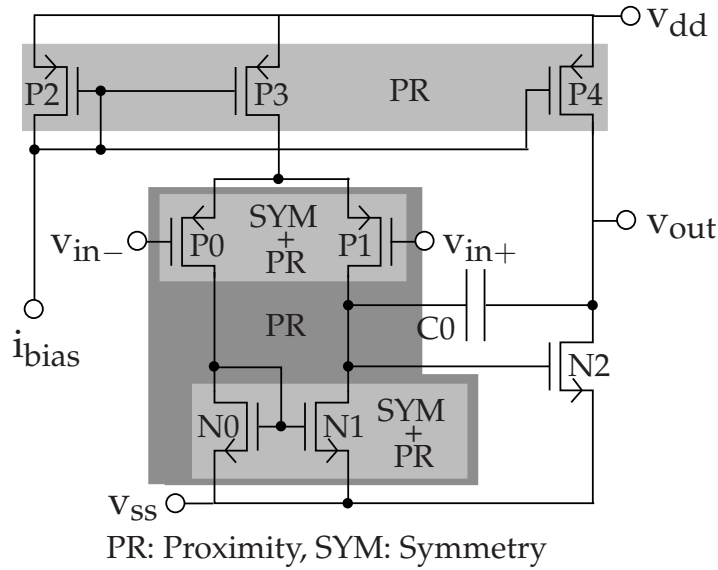


Figure 6.8: MOA topology with superimposed placement constraints.

For the complete circuit, there are 31 electrical inequality constraints, 6 geometric equality constraints, and 25 geometric inequality constraints, including the rule in (6.11) for C_0 .

By using the geometric equality constraints and applying elimination methods to the inequality constraints, the number of electrical inequality constraints is reduced to $n_{ce} = 21$, the number of geometric inequality constraints is reduced to $n_{cg} = 17$, and the number of circuit design parameters is reduced to $n_{d\mathcal{E}} = 11$. The total number of design parameters is $n_d = 12$.

The placement constraints for the MOA are overlaid on the circuit topology in Figure 6.8. Imposed are 4 proximity and 2 symmetry constraints. The minimum margins between each pair of devices constitute additional placement constraints.

The capacitor is placed within a guard ring to reduce noise injection; bulk taps are used to ground the CMOS devices.

For this example, only the interface pin location for the DC bias current, i_{bias} , the DC supply and ground potentials, v_{dd} and v_{ss} , is fixed to a location on the layout boundary prior to the initiation of layout synthesis. No pin assignment is performed for the input and output signals, $v_{\text{in+}}$, $v_{\text{in-}}$, and v_{out} ; and only internal circuit connections will be routed. It is thereby assumed that the external connection for these signals will be performed at a later stage.

Routing constraints are specified in a similar manner to the first circuit example in Section 6.1.1.

Eight test bench circuits are constructed to calculate 14 electrical performances in addition to the DC electrical constraints, as shown in Table 6.3.

Table 6.3: MOA test benches and electrical performances

Test Bench	Analysis	Electrical Performances Simulated	Abbreviation
TB_1	DC	Electrical Constraints	c_e
TB_2	AC	low frequency gain phase margin unity gain bandwidth	Gain PM UGBW
TB_3	AC	common mode rejection ratio power supply rejection ratio from v_{dd} power supply rejection ratio from v_{ss}	CMRR PSRR- v_{dd} PSRR- v_{ss}
TB_4	Transient	slew rate for rising signal settling time	SR-Rising Settling
TB_5	Transient	slew rate for falling signal	SR-Falling
TB_6	DC	input offset voltage power consumption output resistance	IOV Power R_{out}
TB_7	PSS	total harmonic distortion	THD
TB_8	Transient	output voltage swing	OVS

6.2 Experimental Setup

6.2.1 Computer Hardware and Software

All computations were run on a dedicated PC with four quad-core 2.67GHz Intel processors and 12GB of RAM.

CMOS devices were electrically modeled using BSIM3 models [LJX⁺], while Spectre [Kun95a] from Cadence Design Systems was used for numerical analog circuit simulation.

The tool WiCkeD [AEG⁺00a, Cad03b] from MunEDA was used as a simulation server; extensions to implement the nuances of layout-driven circuit sizing, as described in Chapter 5, were written in Python [Pyt09].

The Cadence Virtuoso analog design system was used as a platform for schematic and layout synthesis [Cad08]. Device layouts and geometries were modeled using parametric cells (PCELLS) within the platform. The adaptive routing algorithm in industrial tool of [Cad03a] was used for placement routing. The grid-based maze routing algorithm in the same tool was used in congestion estimation. Electrical circuit

models (netlists) were extracted from layouts using the commercial tool in [Cad05]. SKILL [Bar90], a Lisp dialect, was used to implement the automatic layout synthesis flow of Chapter 4.

A limited number of licenses were available for each commercial tool that was used.

6.2.2 Rules to Extract Layout Netlists

The netlist extraction rules described below were used to extract netlists from layout geometries for the example circuits. They are also suitable for most low frequency circuits with a bandwidth of interest below 1GHz, and strike a suitable balance between the accuracy of the extracted netlist in modeling electrical behavior and the computational cost of extraction.

Routing interconnects are partitioned into segments and a resistance is calculated for each segment. Partitions are made at contacts, line intersections, vias, and device terminals. Long lines are fractured into smaller segments; maximum segment length is $5\mu\text{m}$. For each segment, a lumped parasitic coupling capacitance to each other segment and to the substrate is calculated. The commercial integral equation field solver, RCX-FS, is used to extract coupling capacitance. It is based on the algorithm, Nebula, described in [KL00].

The RC network formed of segment resistors and coupling capacitors is simplified by series and parallel device combinations and the elimination of dangling and small elements; for example, resistors smaller than 0.01Ω are discarded. RC model order reduction is performed considering a maximum frequency of 1GHz.

Diffusion area impedance is accounted for and solved for using a 2D Laplace solver. Parasitic inductance and the effect of eddy currents in the substrate are not modeled.

6.2.3 Selection of the Starting Vector for Circuit Sizing

The selection of initial starting vector, \mathbf{d}^0 , in the design space will influence circuit sizing results and the comparison between traditional circuit sizing (without layout synthesis integration) and the layout-driven circuit sizing algorithm of Chapter 5.

The layout-driven circuit sizing algorithm requires that the starting vector be feasible, so that $\mathbf{d}^0 \in \mathcal{D}$. Furthermore, if \mathbf{d}^0 is arbitrarily chosen, then a large number of iterations, m , may be needed to find the final solution, \mathbf{d}^m . If m is large, then the computational cost of layout-driven circuit sizing will be inflated.

A solution to the problems above is to exert effort into the selection of \mathbf{d}^0 . The feasible starting vector is selected in the design space by traditional circuit sizing, as represented by the formulation in (5.39), but with *relaxed* performance specifications:

$$\mathbf{f} \preceq \mathbf{f}^u \xrightarrow{\text{specification relaxation}} \mathbf{f} \preceq \mathbf{f}^u + \Delta_{\mathbf{f},\text{relaxation}}; \Delta_{\mathbf{f},\text{relaxation}} \succeq \mathbf{0} \quad (6.12)$$

If \mathbf{d}^0 maps to vector \mathbf{f}^0 in the performance space, then $\mathbf{f}^0 \preceq \mathbf{f}^u + \Delta_{\mathbf{f},\text{relaxation}}$ and an improvement of $\mathbf{f}^0 - \mathbf{f}^u \preceq \Delta_{\mathbf{f},\text{relaxation}}$ must be made in the performance space to satisfy the original performance specifications ($\mathbf{f} \preceq \mathbf{f}^u$). The number of algorithm iterations, m , to accomplish a maximum improvement of $\Delta_{\mathbf{f},\text{relaxation}}$ is typically smaller than what is needed with an arbitrary starting point.

The initial starting vector, \mathbf{d}^0 , will be selected by the procedure described above in Sections 6.3.1 through 6.3.3.

6.3 Circuit Sizing Results and Comparison

Both traditional and layout-driven circuit sizing are applied to the example circuits of Section 6.1. The results and costs of circuit sizing are presented, analyzed, and compared in this section.

6.3.1 Folded Cascode Operational Amplifier (FC-OA)

The results of circuit sizing are given in Table 6.4:

- Column 1 and 2: The performance specifications used in circuit sizing.
- Column 3: Initial performance values before circuit sizing.
- Column 4: Performance values obtained by traditional circuit sizing without layout synthesis. Circuit layout area is estimated using the procedure in Appendix A and equation (A.12) with $\rho = 0.5$.
- Column 5: A layout is synthesized for the result of traditional circuit sizing and the post-layout performance values are listed.
- Column 6: Performance values obtained by layout-driven circuit sizing.

Performance values that fail to meet a specification are gray shaded in Table 6.4.

Traditional circuit sizing of the FC-OA circuit took three iterations using the algorithm of Section 5.2. In each iteration, three to four sub-iterations were necessary.

Figure 6.9 compares estimated and actual layout area during the progression of traditional circuit sizing. The starting vector of the first algorithm iteration is labeled "initial" in the graph. The upper, lower, and average area estimate with $\rho = 0.5$ are plotted for each iteration. For comparison, the area of the layouts generated by the synthesis flow of Chapter 4 are also plotted, and the area of the best layout in each iteration is highlighted.

As discussed in Appendix A, it is difficult to accurately estimate layout area without using a computationally costly placement algorithm. Each device can have several

Table 6.4: FC-OA performance specifications and circuit sizing results

Performance Specification	Unit	Initial Value	Traditional Circuit Sizing	After Layout*	Layout-Driven Circuit Sizing
Gain ≥ 80	dB	80	83	83	81
$60 \leq PM \leq 120$	deg	86	84	82	82
UGBW ≥ 7.00	MHz	5.46	6.11	6.14	7.01
CMRR ≥ 100	dB	113	114	111	109
PSRR- $v_{dd} \geq 90$	dB	87	89	86	96
PSRR- $v_{ss} \geq 90$	dB	90	95	105	88
Settling ≤ 250	ms	291	298	297	212
SR-Rising ≥ 4.00	V/ μ s	3.36	3.28	3.29	4.60
SR-Falling ≥ 4.00	V/ μ s	3.44	4.05	4.09	4.91
Power ≤ 0.50	mW	0.39	0.41	0.42	0.49
$ IOV \leq 100$	μ V	37	34	42	76
THD ≤ 0.100	%	0.038	0.091	0.104	0.080
Area ≤ 3500	μm^2	3752	3595 [†]	3229 ^{††}	3417 ^{††}
$\frac{4}{5} \leq \text{Aspect Ratio} \leq \frac{5}{4}$	-				

*A layout is synthesized for the result of traditional circuit sizing.

[†]Estimated layout area using (A.12) with $\rho=0.5$.

^{††}Actual layout area whilst meeting the aspect ratio specification.

valid layouts and area utilization is heavily dependent on the placement constraints. As a result, the range between upper and lower area estimates is large.

From Figure 6.9, the average area estimate was too pessimistic in traditional sizing – estimated area is greater than actual layout area in each iteration. This bias could not be perceived from the initial starting point, nor would it be possible to ascertain without performing layout synthesis and plotting the trend for the actual layout area.

As explained in Section 4.6.2, some performances, such as PSRR- v_{dd} and PSRR- v_{ss} , are sensitive to circuit placement and routing. For the FC-OA results with traditional circuit sizing, the difference between the value of PSRR- v_{ss} (ground node to output) obtained from the schematic netlist and after layout synthesis is 10dB. Similar to inaccurate area estimation, this reduces the usefulness of the results of traditional circuit sizing, as performance values are overestimated or underestimated

In attempting to fulfill the area specification with a pessimistic estimate during traditional circuit sizing, the value of the other performances with a hard-to-meet spec-

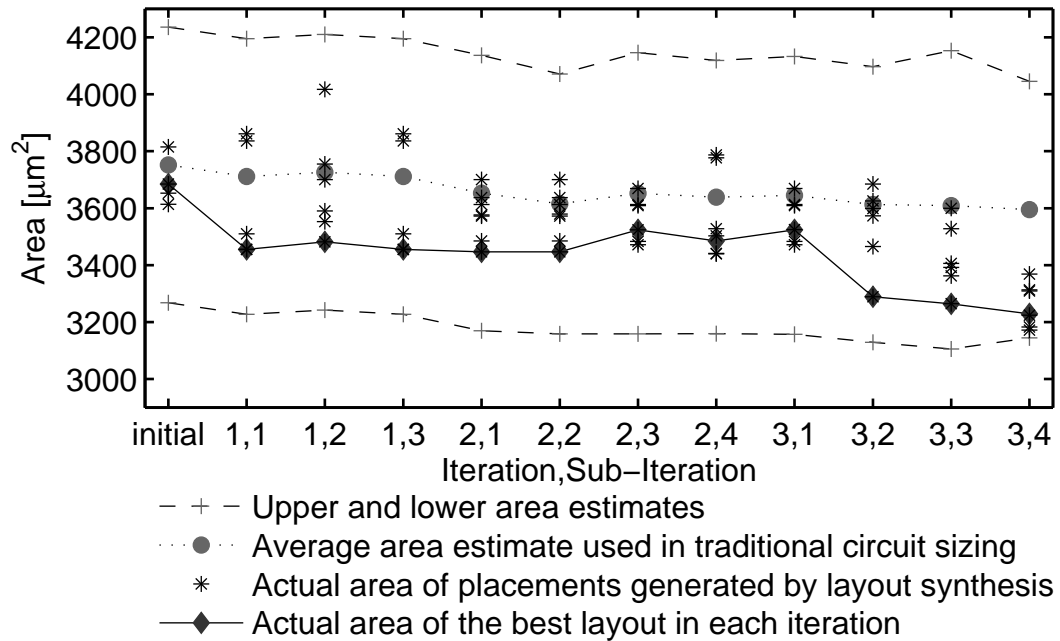


Figure 6.9: The estimated area before and actual area after layout synthesis is shown during the progress of traditional circuit sizing; the difference between actual and estimated area affects the results of circuit sizing.

ification; namely UGBW, PSRR- v_{dd} , SR-Rising, Settling Time, and THD; suffered and the search algorithm converged on a sub-optimal solution in the design space.

With layout-driven circuit sizing, the exact area is calculated after layout synthesis. It was easy to fulfill the specifications of this circuit example. Only PSRR- v_{ss} fell short of the set specification by 2dB.

A breakdown of sizing cost is given in Table 6.5. Cost is given by the CPU time needed for completing each task.

Layout synthesis, with a mean CPU time of 93.18 seconds, constitutes 72% of the cost of layout-driven circuit sizing. On average, each extracted layout netlist contained 176 parasitic resistors and 816 parasitic capacitors. Test benches TB_4 and TB_5 require transient analysis and constitute most of the remaining cost.

The mean cost of a *single* search algorithm iteration is 672 seconds and 3214 seconds respectively for traditional and layout-driven circuit sizing; their ratio is 1:4.8. Layout-driven circuit sizing run took more iterations to converge, five in contrast to three, therefore the complete process of layout-driven circuit sizing took eight times the CPU time of traditional circuit sizing.

Parallelism in the optimization steps was exploited. The determination of the performance Jacobian matrix requires the synthesis of $n_{d\epsilon} = 22$ layouts, which can be performed independently and in parallel. Further parallelism can be exploited when

Table 6.5: FC-OA breakdown of circuit sizing cost; unless otherwise labeled, cost is given by the CPU time needed for each task

Test Bench	Traditional Circuit Sizing		Layout-Driven Circuit Sizing	
	number of calls	mean cost of a call [seconds]	number of calls	mean cost of a call [seconds]
\mathcal{TB}_1 – Electrical Constraints	209	1.46	521	1.46
\mathcal{TB}_2 – Gain, PM, UGBW	81	1.48	132	1.61
\mathcal{TB}_3 – CMRR, PSRR- v_{dd} , PSRR- v_{ss}	81	1.44	132	1.58
\mathcal{TB}_4 – SR-Rising, Settling	81	6.58	132	10.42
\mathcal{TB}_5 – SR-Falling	81	6.64	132	10.42
\mathcal{TB}_6 – IOV, Power	81	1.50	132	1.58
\mathcal{TB}_7 – THD	81	2.16	132	2.87
Layout Synthesis	1	94.10	127	93.18
Optimization Iterations	3		5	
Mean Cost of 1 Iteration [seconds]	672		3270	
Total Cost [hours]	0.56		4.54	
Elapsed <i>Wall Clock</i> Time [hours]	0.35		2.10	

performing the sub-steps of the layout synthesis flow, such as placement routing, layout netlist extraction, and electrical simulation. This wall clock reduction in cost is already reflected in the numbers of Table 6.5. On the used PC and with a limited number of licenses for commercial tools (limiting the type and number of steps that can be completed in parallel), traditional circuit sizing took 0.35 hours of wall clock time, while layout-driven sizing took 2.1 hours; their ratio is 1:6.

For a circuit placement, the B*-tree records the relative location of each device. The device in the lower left corner of the placement is represented by the root tree node, while the remaining devices are represented by the children of the root node as explained in [BMM⁺04].

In order to compare layout-driven circuit sizing to the template-based methods in the state of the art, the B*-trees of all placements generated by the layout synthesis flow of Chapter 4 are reconstructed in Figure 6.10, along with the number of occurrences of each tree during layout-driven circuit sizing of the FC-OA.

For the FC-OA example, placements corresponding to six different B*-tree structures were used during layout-driven circuit sizing. Furthermore, the best placements corresponding to the initial design parameter vector and final solution after layout-driven circuit sizing are represented by different B*-trees. This is shown in Table 6.6.

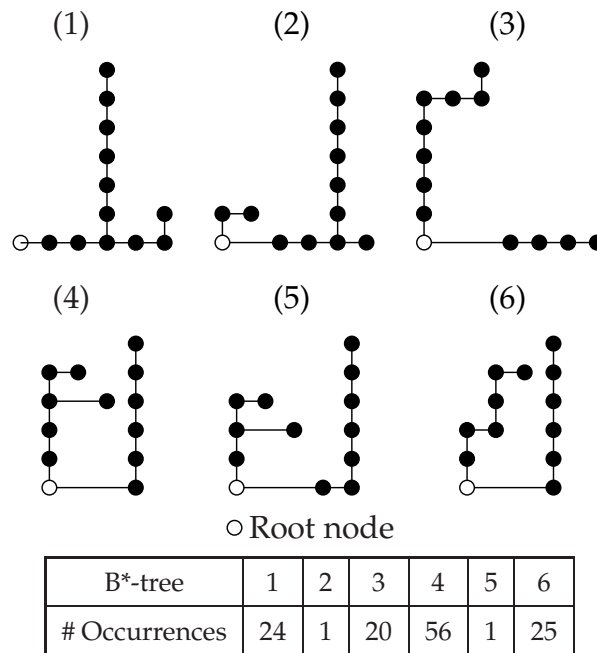


Figure 6.10: B*-trees used during layout-driven sizing; there are 13 nodes, devices in common centroid configuration are represented by a single node.

Table 6.6: FC-OA comparison of placement structure between the initial layout before layout-driven sizing and the final layout

Best Placement	B*-tree (Figure 6.10)	Common Centroid Divisions					
		(N2,N3)	(N7,N6)	(N4,N5)	(P3,P4)	(P6,P5)	(P7,P8)
initial	4	4	2	1	1	1	4
post sizing	6	4	1	1	2	1	2

In template-based methods, the relative location of each device is fixed and does not change during circuit sizing, therefore the corresponding B*-tree structure (or other representation such as a slicing tree or O-tree) is also fixed. As a result, any template-based method would not be able to reach the layout solution found by the layout-driven algorithm used here.

The flexibility to select the divider, M , of CMOS devices in common centroid configuration in the layout synthesis flow, according to Algorithm-2 in Section 4.2.2, was utilized in FC-OA sizing. The number of divisions changed for some common centroid pairs before and after circuit sizing. This is also shown in Table 6.6.

An example FC-OA layout produced by the layout synthesis flow is shown in Figure 6.11.

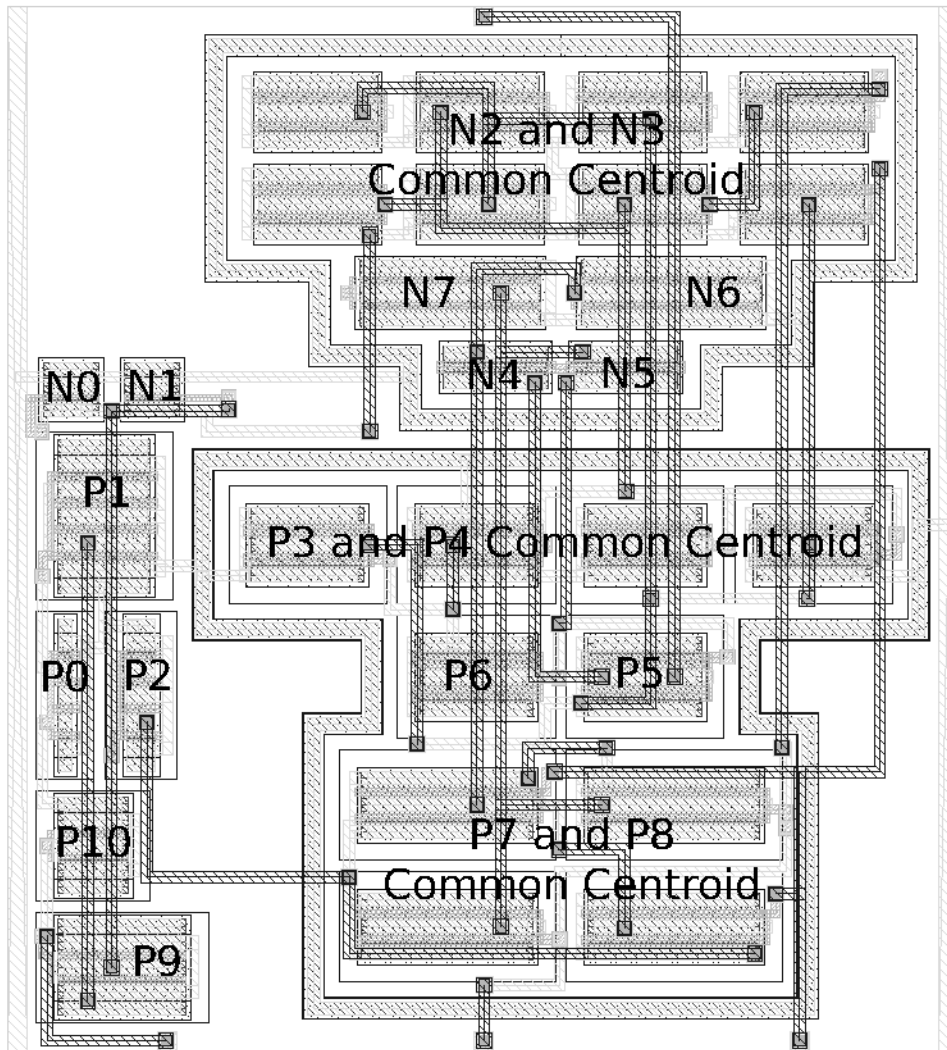


Figure 6.11: Example of an FC-OA layout created by the layout synthesis flow of Chapter 4 and using the placement and routing constraints discussed in Section 6.1.1; this layout has the 6-th B*-tree structure shown in Figure 6.10.

6.3.2 Tunable Operational Transconductance Amplifier (TOTA)

The results of circuit sizing are given in Table 6.7. Performance values that fail to meet a specification are gray shaded.

For the TOTA, the estimated area, used in traditional sizing without layout synthesis, was optimistic. Specifically, the average area estimate with $\rho = 0.5$ is smaller than actual circuit area in each iteration of search algorithm execution. This is due to the strict placement constraints applied to the TOTA to force layout symmetry. Application of the placement constraints results in a reduction of area utilization; it also makes it difficult to meet the aspect ratio constraints, thereby the modified area,

Table 6.7: TOTA performance specifications and circuit sizing results

Performance Specification	Unit	Initial Value	Traditional Circuit Sizing	After Layout*	Layout-Driven Circuit Sizing
$G_{m,low} \geq 1.00$	$\mu A/V$	2.93	3.12	3.28	3.53
$G_{m,high}/G_{m,low} \geq 40$	-	45	62	61	61
Power@1x ≤ 1	mW	0.80	0.78	0.79	0.82
Power@40x ≤ 3	mW	3.75	2.15	2.22	2.73
$ V_{out,cm} - V_{cm} \leq 1.0$	mV	0.5	0.8	2.2	1.2
Linearity@40x ≤ 5.0	%	4.0	6.5	11.1	4.6
IRN@1x ≤ 500	$\frac{nV}{\sqrt{Hz}}$	977	417	402	501
Gain@1x ≥ 25	dB	19	23	28	25
BW@1x ≥ 500	kHz	124	617	491	524
IRN@40x ≤ 600	$\frac{nV}{\sqrt{Hz}}$	1136	598	576	585
Gain@40x ≥ 25	dB	18	31	32	35
BW@40x ≥ 5.00	MHz	3.14	5.74	8.94	6.60
Area ≤ 10000	μm^2	11169	9793 [†]	10660 ^{††}	10030 ^{††}
$0.5 \leq \text{Aspect Ratio} \leq 2$	-				

*A layout is synthesized for the result of traditional circuit sizing.

[†]Estimated layout area using (A.12) with $\rho=0.5$.

^{††}Actual layout area whilst meeting the aspect ratio specification.

calculated from the geometric specifications using (4.33), is also larger for the actual circuit layout.

In traditional circuit sizing, the estimated area of $9793\mu m^2$ was close to the specification bound of $10000\mu m^2$, so that the actual area after layout synthesis increased to $10660\mu m^2$ – passing above the specification bound.

In layout-driven sizing, the circuit area was $10030\mu m^2$, or 0.30% above the specification bound. The minimization of circuit area reached a limit due to the placement constraints and the aspect ratio specification. Any new decrease in area would require a significant reduction in the width of matched devices P12 and P13. This reduction would edge the electrical performance values out of specification. The trade-off between circuit area and the electrical performances cannot be improved without changing the placement constraints or the minimum margins between devices. To meet the area specification after circuit sizing is completed, the designer can change the circuit margins slightly so as to reduce the area by the trivial amount of $30\mu m^2$.

The RMS input referred noise and the bandwidth are the electrical performances most sensitive to circuit placement and routing, while the linearity measure was systematically higher after layout synthesis. These three performances benefited from the use of layout-driven sizing in comparison to traditional circuit sizing.

Table 6.8: TOTA breakdown of circuit sizing cost; cost is given by the CPU time needed for each task

Test Bench	Traditional Circuit Sizing		Layout-Driven Circuit Sizing	
	number of calls	mean cost of a call [seconds]	number of calls	mean cost of a call [seconds]
$\mathcal{TB}_1 - G_{m,low}, G_{m,high}, (V_{out,cm} - V_{cm}), \text{Power}$	483 + 551*	3.77	246	8.45
$\mathcal{TB}_2 - \text{Electrical Constraints}$	1034	2.05	229	2.24 + 3.87†
$\mathcal{TB}_3 - \text{linearity measure}$	483	4.94	246	8.28
$\mathcal{TB}_4 - \text{IRN, Gain, BW for } I_{tune} = I_{tune,low}$	483	1.49	246	3.46
$\mathcal{TB}_5 - \text{IRN, Gain, BW for } I_{tune} = I_{tune,high}$	483	1.49	246	3.46
Layout Synthesis	1	150.20	231	145.30
Optimization Iterations	10		5	
Mean Cost of 1 Iteration [seconds]	999		8156	
Total Cost [hours]	2.78		11.33	

*Additional executions of \mathcal{TB}_1 to calculate c_e .

† \mathcal{TB}_1 is executed without layout synthesis to calculate c_e .

The DC output common-mode offset, $(V_{out} - V_{cm})$, is sensitive to the electrical mismatch of devices as well as routing symmetry along the common-mode feedback loop, and was typically much higher after layout synthesis. For this reason, the value of the output common-mode level for the result of traditional sizing jumped from 0.8mV to 2.22mV after layout synthesis. The result of layout-driven sizing, 1.20mV, is an improvement on what was possible with traditional sizing. It was difficult, during performance optimization to keep this performance within a specification bound lower than 1.50mV in each algorithm iteration and over the current tuning range ($I_{tune} \in [I_{tune,low}, I_{tune,high}]$). This is despite the effort made in the construction of placement and routing constraints for automatic synthesis. With the insight gained by the results of layout-driven sizing, it can be said that the specification $|V_{out} - V_{cm}| \leq 1\text{mV}$ is in a range too small to realize and retain after layout.

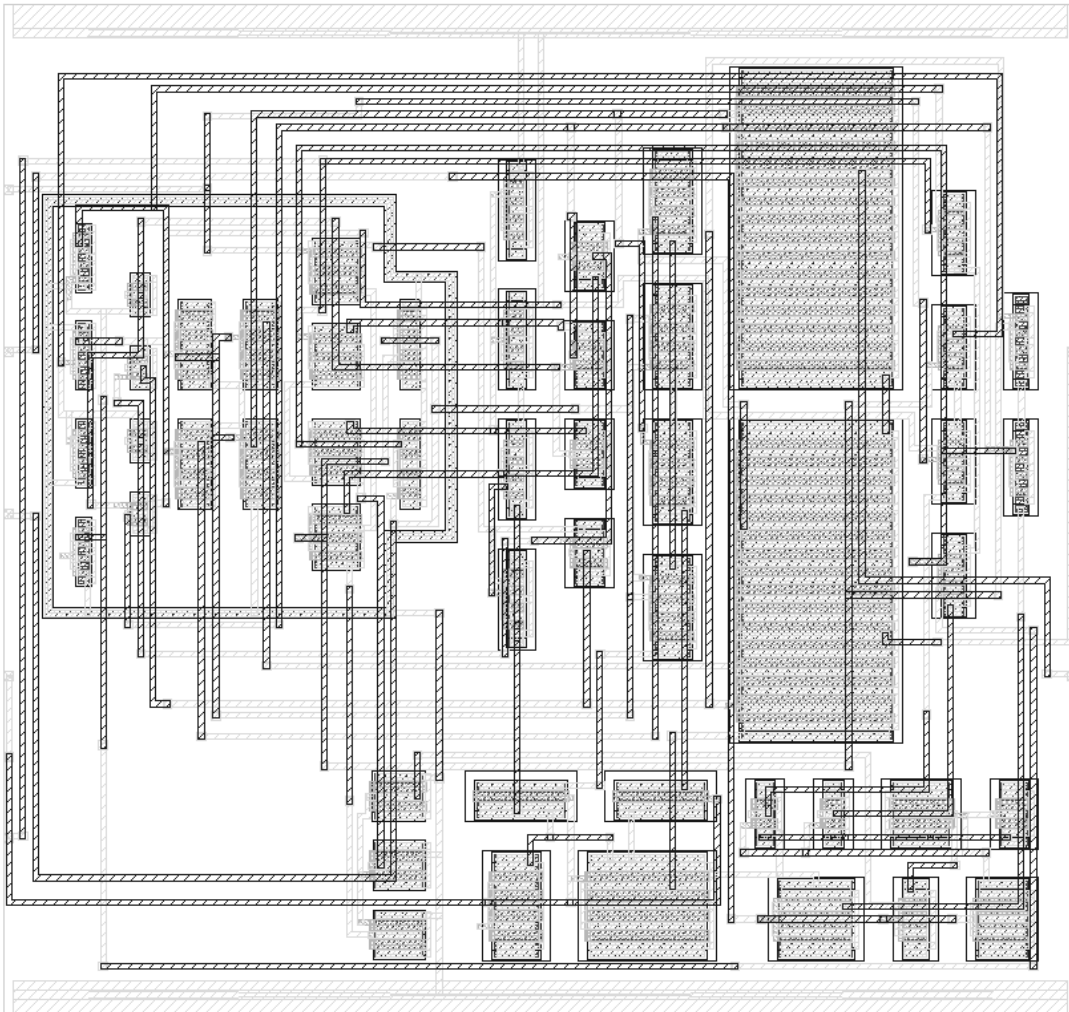


Figure 6.12: Example of an TOTA layout created by the layout synthesis flow of Chapter 4 and using the placement and routing constraints discussed in Section 6.1.2.

A breakdown of sizing cost is given in Table 6.8. Cost is given by the CPU time needed for completing each task. For layout-driven circuit sizing, TB_1 is executed without layout synthesis to calculate c_e .

For this larger circuit, layout synthesis took a mean time of 145.3 seconds and constitutes 82% of the cost of the layout-driven flow. On average, each extracted layout netlist contained 327 parasitic resistors and 1678 parasitic capacitors.

The mean cost of a *single* optimization iteration is 999 seconds and 8156 seconds respectively for traditional and layout-driven sizing; their ratio is approximately 1:8. Layout-driven circuit sizing progress terminated after 5 iterations, while traditional sizing edged on with small improvements in the performances for 10 iterations. The

number of iterations completed by the algorithm will depend on the initial starting point and the value of the performance specifications; this will skew comparison results. For the circuit comparison made here, the complete layout-driven flow cost approximately 4 times the CPU time of the traditional flow.

An example TOTA layout is shown in Figure 6.12.

6.3.3 Miller Operational Amplifier (MOA)

The results of circuit sizing are given in Table 6.9. Performance values that fail to meet a specification are gray shaded.

Table 6.9: MOA performance specifications and circuit sizing results

Performance Specification	Unit	Initial Value	Traditional Circuit Sizing	After Layout*	Layout-Driven Circuit Sizing
Gain ≥ 75	dB	75	76	76	76
$60 \leq \text{PM} \leq 120$	deg	61	61	65	60
UGBW ≥ 10.00	MHz	9.74	10.35	9.70	11.02
CMRR ≥ 75	dB	73	75	76	74
PSRR- $v_{\text{dd}} \geq 80$	dB	80	81	83	83
PSRR- $v_{\text{ss}} \geq 80$	dB	91	84	87	92
Settling ≤ 150	ms	0.151	148	175	145
SR-Rising ≥ 7.00	V/ μs	6.96	7.02	5.99	7.12
SR-Falling ≥ 7.00	V/ μs	7.64	8.12	6.95	8.06
Power ≤ 1.00	mW	0.84	0.79	0.80	0.87
$ \text{IOV} \leq 100$	μV	37	34	52	76
$R_{\text{out}} \geq 50.0$	k Ω	50.2	52.7	54.1	53.91
OVS ≥ 2.60	V	2.59	2.60	2.59	2.59
THD ≤ 0.0100	%	0.0114	0.0105	0.0131	0.0113
Area ≤ 7500	μm^2	7796	7482 [†]	7607 ^{††}	7406 ^{††}
Aspect Ratio = 1	-				

*A layout is synthesized for the result of traditional circuit sizing.

[†]Estimated layout area using (A.12) with $\rho = 0.5$.

^{††}Actual layout area whilst meeting the aspect ratio specification.

In order to illustrate the difficulty of circuit sizing for this example, the deterministic multi-objective goal attainment algorithm presented in [MGS07] is used to *estimate*

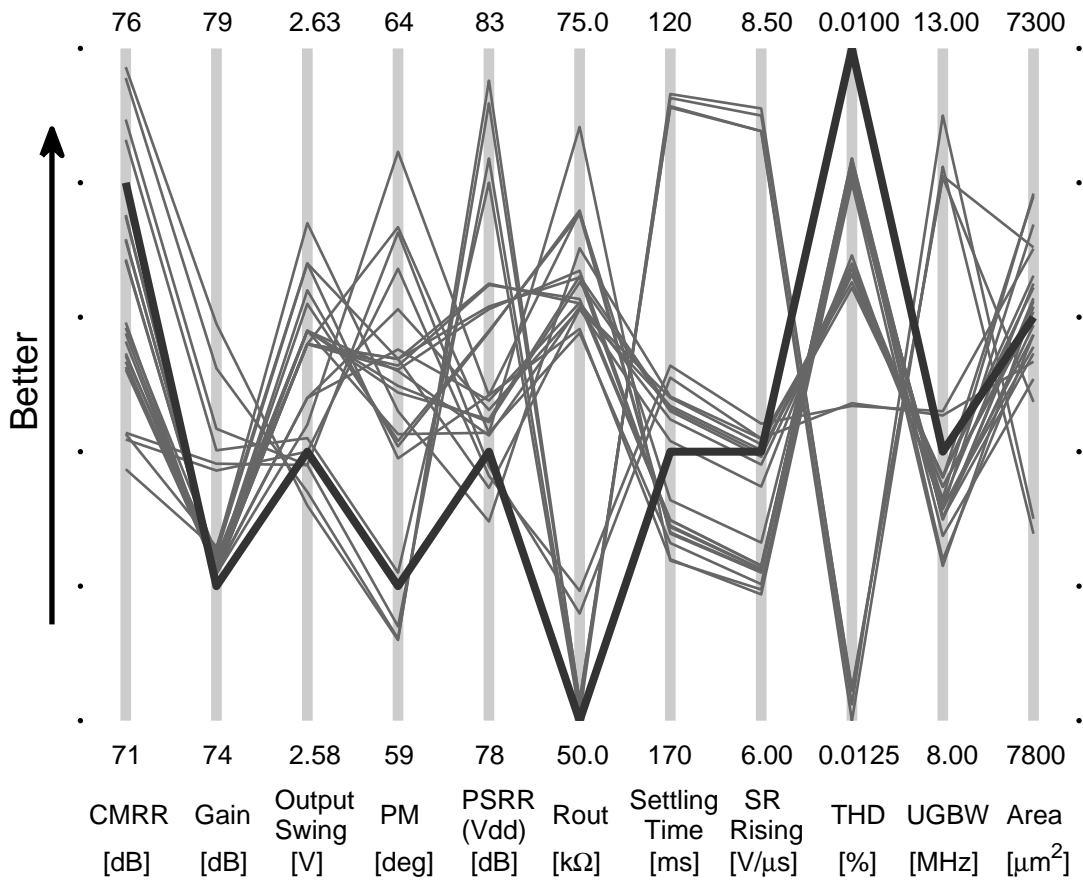


Figure 6.13: Estimated Pareto-optimal performance tradeoffs for the MOA circuit plotted in parallel coordinates; the black line identifies the performance specifications set in Table 6.9; the electrical constraints as well as the remaining performance specifications are satisfied for the set of tradeoffs.

the Pareto-optimal tradeoffs [Par06] between Gain, UGBW, PM, CMRR, PSRR- v_{dd} , OVS, R_{out} , SR-Rising, Settling Time, THD, and Area; under the condition that the electrical constraints as well as the specifications PSRR- $v_{ss} \geq 80$ dB, Power ≤ 1.00 mW, and $|IOV| \leq 100\mu$ V are satisfied. The performances PSRR- v_{ss} , Power, and IOV satisfy their specifications with a large safety margin; they are not critical and will contribute an inconsequential term to objective (5.10) used by the search algorithm of Section 5.2.

The result of multi-objective optimization is a set of 23 non-dominated vectors in the feasible performance space; these vectors are graphed in Figure 6.13 using parallel coordinates [Ins10]. The vector of performance specifications, obtained from Table 6.9, is also graphed in the figure.

Multi-objective optimization required over 30000 performance evaluations to complete. The set of 23 non-dominated vectors does not completely cover the Pareto-optimal boundary between the 11 critical performances, however, it roughly indi-

cates the possible combinations of performances, in addition to the position of the performance specifications vector relative to the Pareto-optimal boundary.

It is apparent from Figure 6.13 that the performance specifications in Table 6.9 are formidable. Foremost, the specification $\text{THD} \leq 0.0100\%$ is infeasible. This means that the search algorithm will only terminate once no improvement is possible or once performance gradient approximations cannot be calculated with sufficient accuracy. Secondly, when THD is omitted, the vector of remaining specifications is very close to the estimated Pareto-optimal boundary. The region in which the search algorithm can maneuver so as to satisfy the specifications is very small and the objective in (5.10) is very sensitive to the value of each performance. Thirdly, there is a steep tradeoff between performances, such as the tradeoff between PM and [Settling Time, SR-Rising, UGBW]. For the 23 non-dominated performance vectors of Figure 6.13, the correlation coefficients between PM and [Settling Time, SR-Rising, UGBW] are $[0.73509, -0.75736, -0.77908]$. It is due to this steep tradeoff in combination with the sensitivity of objective (5.10) to the value of each performance that PM must remain near the lower specification bound of 60 degrees for the circuit sizing results in Table 6.9. Any small increase in PM will cause the other performance values to change rapidly and invalidate the specifications. Finally, in layout-driven circuit sizing discretization and placement error must be successfully palliated in each search iteration, whilst taking very small steps in the performance space to improve the sensitive objective.

All performance specifications except THD are satisfied in Table 6.9 for traditional circuit sizing. After layout synthesis, UGBW, Settling Time, SR-Rising, SR-Falling, OVS, and Area also fail to satisfy the specifications.

The change in [UGBW, Settling Time, SR-Rising, SR-Falling] is related to the 4 degree change in PM and the sensitive tradeoff between these performances. In terms of the circuit frequency response, the difference in PM and UGBW can be explained by the change in the dominant pole locations due, in turn, to the change in the value of coupling and load capacitance before and after layout synthesis; and by undesired frequency compensation due to parasitic layout devices.

The low limit of 0.0100% set on THD makes it very sensitive to circuit placement and routing. The relative difference between THD before and after layout synthesis for traditional circuit sizing is 25%.

The compensation capacitor, C_0 , dominates the layout area of the MOA. This, in conjunction with the square aspect ratio that is specified and the small number of circuit devices kept the range of area utilization small and made for a good area estimate using the method of Appendix A. The difference between the pre-layout area estimate and the actual area after layout is $125\mu\text{m}^2$ for traditional circuit sizing in Table 6.9. The actual area is $7607\mu\text{m}^2$ or $107\mu\text{m}^2$ beyond the specification bound of $7500\mu\text{m}^2$. In practice, this small transgression is not critical and can be corrected for manually by the designer, for example by a small adjustment of the circuit margins.

Finally, the amount of 10mV by which OVS is below the specification limit is negligible in consideration of the tradeoff with the remaining performances.

For layout-driven circuit sizing, only the CMRR, OVS, and THD specifications are unsatisfied. Because of the ambitious specification on THD, a compromise solution is obtained to minimize as much as possible the objective in (5.10).

A breakdown of sizing cost is given in Table 6.10. Cost is given by the CPU time needed for completing each task.

Table 6.10: MOA breakdown of circuit sizing cost; unless otherwise labeled, cost is given by the CPU time needed for each task

Test Bench	Traditional Circuit Sizing		Layout-Driven Circuit Sizing	
	number of calls	mean cost of a call [seconds]	number of calls	mean cost of a call [seconds]
TB_1 – Electrical Constraints	161	1.41	87	1.41
TB_2 – Gain, PM, UGBW	149	1.44	88	1.54
TB_3 – CMRR, PSRR- v_{dd} , PSRR- v_{ss}	149	1.45	88	1.51
TB_4 – SR-Rising, Settling	149	3.78	88	5.68
TB_5 – SR-Falling	149	3.79	88	5.77
TB_6 – IOV, Power, R_{out}	149	1.41	88	1.54
TB_7 – THD	149	1.70	88	1.96
TB_8 – OVS	149	3.74	88	5.69
Layout Synthesis	1	115.85	83	106.96
Optimization Iterations	9		5	
Mean Cost of 1 Iteration [seconds]	325		2217	
Total Cost [hours]	0.82		3.08	

For this circuit, layout synthesis took a mean time of 106.96 seconds, this constitutes 80% of the cost of the layout-driven flow. On average, each extracted layout netlist contained 38 parasitic resistors and 234 parasitic capacitors. Although the MOA has fewer devices than the FC-OA, a longer time, on average, was needed to complete layout synthesis. The cause of high synthesis cost was identified to be the placement generation step described in Section 4.3.3. The space of valid circuit placements was large, such that a long time was needed to enumerate the best placements.

The mean cost of a *single* search algorithm iteration was 325 seconds and 2217 seconds respectively for traditional and layout-driven circuit sizing; their ratio is 1:6.8.

Traditional circuit sizing of the MOA circuit took nine iterations. The last three algorithm iterations were spent in attempt to minimize THD and satisfy the specification $\text{THD} \leq 0.0100\%$. The search terminated once no new improvement could be made, Layout-driven circuit sizing took five iterations. The algorithm terminated after five iterations because of inaccuracy in performance gradient calculation due to placement error, as described in Section 5.7.4.

The complete process of layout-driven circuit sizing took approximately 3.8 times the CPU time of traditional circuit sizing.

An example MOA layout produced by the layout synthesis flow is shown in Figure 6.14.

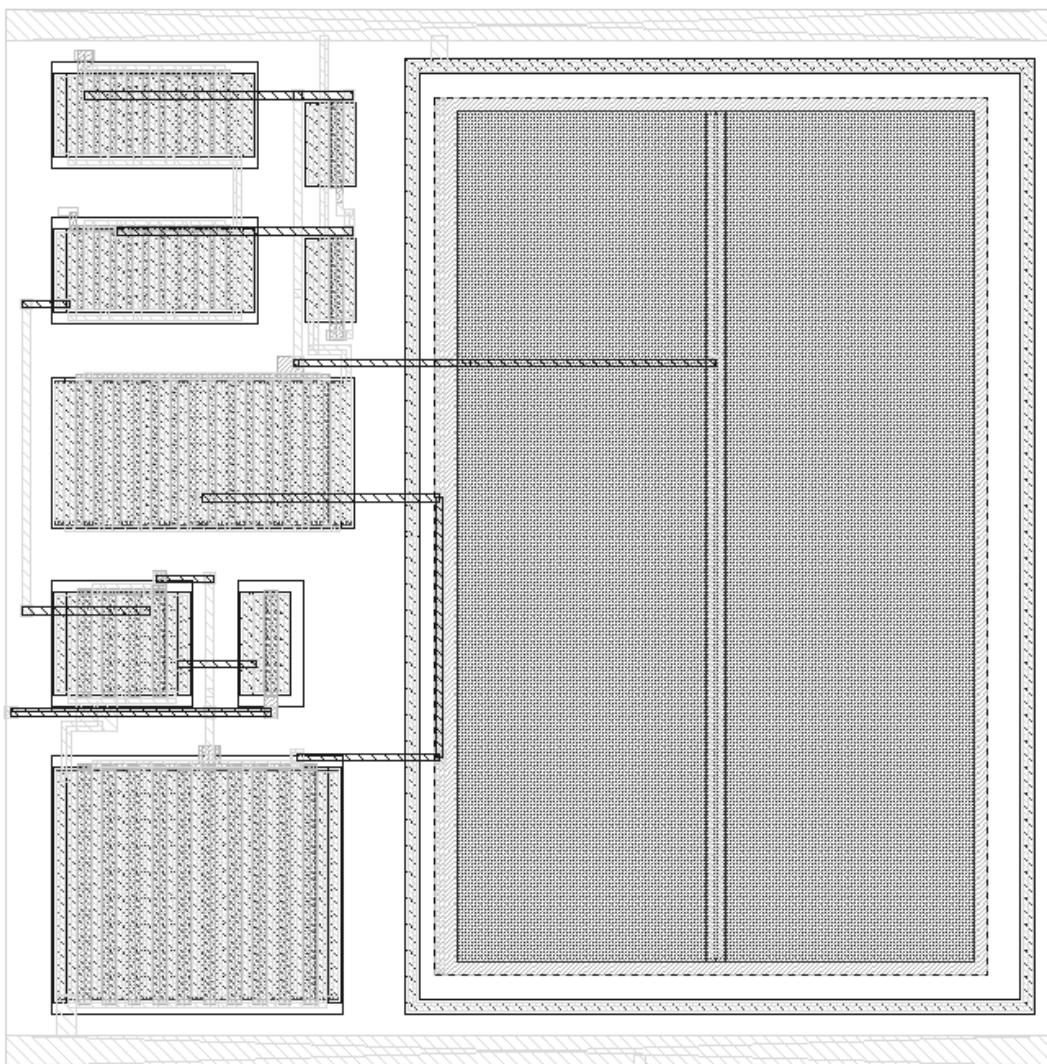


Figure 6.14: Example of an MOA layout created by the layout synthesis flow of Chapter 4 and using the placement and routing constraints discussed in Section 6.1.3.

6.4 Summary

The layout-driven circuit sizing algorithm described in Chapter 5 is used to size three CMOS example circuits.

Prior to traditional sizing, substantial design effort is necessary to recognize functional sub-blocks, such as current mirrors and level shifters, and extract the sizing rules that define the boundaries of the feasible design space and ensure circuit robustness. For layout-driven circuit sizing, the device, placement, and routing constraints needed by the layout synthesis flow of Chapter 4 must also be defined. These setup steps were presented in detail for the FC-OA circuit example.

The TOTA is an elaborate example, with 52 devices, a complex performance measurement procedure, and the hierarchical reuse of the pre-designed FC-OA circuit. The FC-OA is retargeted to be used by the input stage of the TOTA; it is concurrently placed with the other circuit devices to improve layout symmetry and compactness.

In the MOA example, new geometric sizing rules are derived for polysilicon capacitors, so as to ensure performance robustness towards statistical variation in the process parameter values.

The results of layout-driven sizing are compared to the outcome of traditional circuit sizing without the in line integration of layout synthesis, as done in [SSGA00].

In brief, it is shown that the electrical performances are typically met when using the layout-driven algorithm. For the cases when a specification is unsatisfied, either the margin to the unmet specification was small or the specification vector was infeasible. With traditional sizing, electrical performances that are sensitive to layout parasitic devices or take values close to the specification bounds will fail to meet the specifications once layout synthesis is performed.

It is difficult to accurately estimate circuit area without layout-synthesis. This is because area utilization is heavily dependent on the placement constraints and because each device may have many valid placements.

Inaccuracy in area estimation during traditional circuit sizing was presented in detail for the FC-OA. For this circuit, many of the electrical performances were in a tradeoff situation with circuit area. In the attempt to fulfill the area specification with a pessimistic estimate, the value of the electrical performances suffered and the algorithm converged on a sub-optimal solution.

Because of the hard-to-meet layout symmetry constraints and aspect ratio specification, the area estimate of the TOTA circuit was too optimistic in traditional circuit sizing of the TOTA. With layout-driven circuit sizing, the exact area is calculated, and the impact of bad area estimation is removed.

In template-based layout-driven circuit sizing methods, the relative location of each device is fixed and does not change during circuit sizing. For the FC-OA example,

placements corresponding to six different B*-tree structures were used during the progression of the new (constraint-based) layout-driven circuit sizing procedure. As a result, any template-based method would not be able to reach the layout solution found by the algorithm proposed in this dissertation.

For each example circuit, the computational cost, in CPU time, was compared between traditional and layout-driven circuit sizing. Cost is dominated by electrical performance and constraint evaluation, as well as the cost of layout synthesis. The cost ratio of one iteration of traditional sizing to one iteration of layout-driven sizing was 1:4.8, 1:8, and 1:6.8 respectively for the FC-OA, TOTA, and MOA circuits.

The cost of electrical simulation is higher for post-layout circuit models. This is due to the higher complexity of devices and the parasitic resistors and capacitors that are added to the circuit topology. However, the execution cost of the layout synthesis flow of Chapter 4 always dominated the cost of layout-driven circuit sizing. Layout synthesis constituted 72%, 82%, and 80% of the total cost of layout-driven circuit sizing for the FC-OA, TOTA, and MOA circuits respectively.

Abundant computer resources can be used to complete independent steps of the circuit sizing procedure in parallel. As a result, the actual wall clock time that is needed to complete circuit sizing can be lower than the CPU time.

Chapter 7

Conclusion

The analog integrated circuit (IC) design flow consists of the following steps. First, a circuit topology – a network of devices – with the potential to meet the functional purpose of the circuit, such as voltage signal amplification, is selected. Secondly, sizing of device dimensions is performed to meet the specifications, such as minimum gain and maximum power requirements. Thirdly, a layout is synthesized to create the geometric masks for IC fabrication. Post-layout verification is also necessary to ensure that the layout will meet the specifications placed on electrical performances.

Electrical design automation tools for analog circuits still lag behind their digital counterparts. In its current report, the international technology roadmap for semiconductors (ITRS), the leading organization for technology assessment in the semiconductor industry, emphasized the need for better tools to successfully and expeditiously design analog circuits [tec09]. Of the unsolved problems particularly relevant in CMOS fabrication technology for commodity markets, the report emphasized the need for closer modes of interaction between circuit sizing and layout synthesis.

When using contemporary CMOS technologies, it is impossible to account for the effect of layout synthesis on electrical behavior by design heuristics alone. Difficult-to-meet performance requirements leave no room for error margins in specifications.

Bounds on layout dimensions are often severe, so as to fit the circuit in a compact system on a chip (SOC) solution. This can lead to decisive tradeoffs between layout area and the electrical performances, as the adverse effects of layout on electrical behavior tend to increase with layout compactness. To further complicate the matter, tradeoff assessment during the circuit sizing step may be incorrect. This is because of the difficulty of precise estimation of layout dimensions prior to layout synthesis.

If the circuit layout is too big or a failure to meet all electrical specifications is detected during the electrical verification step, then it will be necessary to backtrack up the design flow and repeat circuit sizing; this is a tarrying procedure.

In this dissertation, a procedure was presented to integrate the layout synthesis and circuit sizing steps in analog circuit design. The novelty in comparison to the state of the art lies in the following items:

- A deterministic optimization algorithm was employed in circuit sizing.

This algorithm is more efficient than nondeterministic algorithms, such as simulated annealing, because the number of necessary circuit simulations is smaller. Circuit simulation replicates the electronic response of a circuit to a given input signal by numerical analysis. From this response, circuit performance can be predicted. Since the computational time of analog circuit simulation is relatively high, the effort expended in circuit sizing is measured by the number of necessary circuit simulations.

- Discrete parameters, such as the number of gates in a multi-gate transistor, were handled without the need for discrete optimization algorithms.

Discrete stochastic optimization algorithms converge slowly, while deterministic algorithms can only be employed if the space of discrete design parameters can be extended to a continuous domain. The need for discrete optimization algorithms was avoided by the procedure of this dissertation. All discrete possibilities are enumerated, then filtered based on the suitability for layout synthesis. It was shown that most discrete possibilities can be readily discarded, and only a few additional circuit simulations are needed to decide upon the best discrete parameter values.

- Circuit layouts are synthesized from scratch using a numerical procedure driven by a set of design constraints and layout directives.

No layout template is needed. It was shown, by a circuit example, that template-based methods restrict the space of layout possibilities; they are not able to reach the layout solutions found by the procedure in this dissertation.

- A novel plan was used to impose electrical constraints during layout synthesis.

The electrical constraints are parameterized as a function of parasitic routing resistance. Resistance is then controlled to ensure the satisfaction of imposed electrical constraints. Principle to this approach is that the generation of a DC circuit model is fast, and that quiescent point sensitivity analysis is relatively cheap to perform.

The procedure of this dissertation was used in the layout-driven circuit sizing of several CMOS circuits. These included a large analog block with 52 transistors, which was sized in 11.33 hours of CPU time on a contemporary workstation. Significant improvements in post-layout specification satisfaction were shown in the comparison to traditional circuit sizing without layout synthesis integration.

Statistically significant process variations during IC manufacturing will impact electrical behavior. In what are termed layout-dependent proximity effects, the adjacent structures to a CMOS device will have a systematic influence on drain current and threshold voltage. The phenomena listed above may cause an IC to violate the imposed specifications after production; such an IC cannot be sold, thereby production yield is reduced. An intuitive step to improve upon the work in this dissertation would be to take into account statistical process and proximity effects at the layout level, so as to optimize post-layout production yield.

Appendix A

Area Estimation Without Layout Synthesis

In order to compare circuit sizing results with and without layout synthesis, problem (5.39) must be solved without resorting to layout synthesis. An estimate \hat{A}_{ws} of A_{ws} must be made from the value of the circuit design parameters, $\mathbf{d}_{\mathcal{E}}$, and independent of any specific layout.

The modified area can be a critical circuit performance, since the step taken in the design space by the search algorithm in each iteration to improve \hat{A}_{ws} is affected by the difference $(A_{\text{max}} - \hat{A}_{\text{ws}})$ and by the magnitude $\|\mathbf{J}_{\hat{A}_{\text{ws}}}(\mathbf{d}^k)\|$.

If \hat{A}_{ws} is replaced by an estimation \hat{A}_{wos} , then the search algorithm may fail to find a feasible solution or could consume too many iterations – incurring additional computational cost. The counterargument in favor of pre-layout estimation is that layout synthesis in each iteration of the search algorithm is too costly and that the estimation \hat{A}_{wos} is good enough to find a feasible solution in most circuit design problems.

If only the design parameters are used in estimation, then only a very crude estimate can be made of circuit area. For example, for a circuit with CMOS devices $\mathcal{E} = \{\delta_1, \delta_2, \dots\}$, circuit area can be estimated from the size of the active regions when all of the devices are laid out without folding, while the circuit can be assumed square to calculate width, length, and aspect ratio:

$$\begin{aligned} \text{area} &\approx \sum_{\delta_k \in \mathcal{E}} (W_{\delta_k} \cdot L_{\delta_k}) \quad (\text{crude area estimate}) \\ \text{width} = \text{length} &= \sqrt{\text{area}}; \quad \text{aspect Ratio} = \frac{\text{length}}{\text{width}} = 1 \end{aligned} \tag{A.1}$$

In actuality, each circuit device can have multiple valid layouts, subject to the device layout constraints, for the same value of the design parameters. This is explained in Section 4.2. From the set of valid device layouts, multiple circuit placements can be generated. This is explained in Section 4.3. Area utilization (layout compactness); as well as placement width, length, and aspect ratio are subject to geometric

placement constraints (Section 4.3.1) and minimum device margins (Section 4.3.2). Adjustments made for congestion control (Section 4.4.2) can also change the geometric performances. Finally, electrical performances can be sensitive to placement and routing parasitic effects. Therefore both the geometric properties and the electrical behavior of the post-layout circuit are weighted to select the best layout, \mathbf{p}^* , from the set of valid placements, \mathbf{P} . This is explained in Section 4.6.

In order to make an honest comparison with and without the use of layout synthesis, as many considerations as possible will be taken into account, here, in the calculation of \hat{A}_{wos} . Expressly, device layout constraints and the minimum margins between device layouts are considered. The execution of a rectangle packing algorithm is considered too costly for estimation, therefore compact placement generation under consideration of placement constraints, as done in Section 4.3.3, is not performed. The subsequent steps of congestion control and electrical performance evaluation are not considered.

The specific steps to estimate a range for \hat{A}_{wos} are listed below. Without loss of generality, equations are presented for CMOS devices with transistor folding.

First, the parameters of each device are extracted from the circuit design parameters:

$$\mathbf{d}_{\mathcal{E}} \xrightarrow{(5.27)} \mathbf{d}_{\mathcal{E},\text{original}} \xrightarrow{\text{extract}} \mathbf{d}_{\delta} = [W_{\delta} \cdot L_{\delta}]; \delta \in \mathcal{E} \quad (\text{A.2})$$

Secondly, the procedure of Algorithm-1 in Section 4.2 is executed to obtain the set of layout parameters, \mathcal{V}_{δ} , which satisfy the layout constraints for each device $\delta \in \mathcal{E}$:

$$\mathbf{d}_{\delta} \xrightarrow{\text{Section 4.2: Algorithm-1}} \mathcal{V}_{\delta} = \{\lambda_{\delta}^{(1)}, \lambda_{\delta}^{(2)}, \dots\} \quad (\text{A.3})$$

From the value of the device layout parameters, an accurate device layout can be synthesized according to the technology design rules; this layout includes substrate taps and internal device routing, such as the routing of connections between individual fingers in a folded CMOS device.

Thirdly, the layout dimensions corresponding to each vector of device layout parameters, $\lambda_{\delta} \in \mathcal{V}_{\delta}$, are calculated. Let $\text{width}(\lambda_{\delta})$ and $\text{length}(\lambda_{\delta})$ denote the layout dimensions of the device synthesized from λ_{δ} .

$$\lambda_{\delta} \xrightarrow[\text{of device } \delta]{\text{layout synthesis}} \text{width}(\lambda_{\delta}), \text{length}(\lambda_{\delta}) \quad (\text{A.4})$$

Fourthly, the margins between devices are considered. As discussed in Section 4.3.2 and illustrated in Figure 4.6, a minimum margin is specified for the distance between every pair of devices in a circuit topology; this is specific to each device edge (Top, Bottom, Left, Right).

Compact placement generation in consideration of the placement constraints is not performed during estimation, therefore it is unknown which device will abut another

and along which edge abutment occurs in the circuit layout. The first estimation step made here is to take the average of the margins for each device and edge. The average margin for the left edge of device $\delta \in \mathcal{E}$ is denoted by $\hat{M}_L(\delta)$:

$$\hat{M}_L(\delta) = \sum_{\delta^* \in \mathcal{E} \setminus \delta} \frac{M_L(\delta, \delta^*)}{|\mathcal{E}| - 1}; \quad M_L(\cdot, \cdot) \text{ is defined in Section 4.3.2} \quad (\text{A.5})$$

The averages $\hat{M}_T(\delta)$, $\hat{M}_B(\delta)$, and $\hat{M}_R(\delta)$ have analogous definitions. The average margins are added to the width and length of the corresponding device:

$$\begin{aligned} \text{width}(\lambda_\delta) &\longleftarrow \text{width}(\lambda_\delta) + \hat{M}_L(\delta) + \hat{M}_R(\delta) \\ \text{length}(\lambda_\delta) &\longleftarrow \text{length}(\lambda_\delta) + \hat{M}_T(\delta) + \hat{M}_B(\delta) \end{aligned} \quad (\text{A.6})$$

For the fifth step, the the minimum and maximum area is identified for each device $\delta \in \mathcal{E}$ from the layout parameter vectors in \mathcal{V}_δ :

$$\begin{aligned} \mathcal{V}_\delta &\longmapsto A_{\min}(\delta) = \min_{\lambda_\delta \in \mathcal{V}_\delta} (\text{width}(\lambda_\delta) \cdot \text{length}(\lambda_\delta)) \\ \mathcal{V}_\delta &\longmapsto A_{\max}(\delta) = \max_{\lambda_\delta \in \mathcal{V}_\delta} (\text{width}(\lambda_\delta) \cdot \text{length}(\lambda_\delta)) \end{aligned} \quad (\text{A.7})$$

Since the device layout constraints are satisfied for each vector $\lambda_\delta \in \mathcal{V}_\delta$, each device layout with the dimensions given by (A.6) has equal probability to be used in circuit placement. Furthermore, the range $[A_{\min}(\delta), A_{\max}(\delta)]$ for each device δ is also small as it reflects realistic device layouts.

At this point, a rectangle packing algorithm would be used to enumerate the possible circuit placements over the set $\mathcal{V}_{\delta_1} \times \mathcal{V}_{\delta_2} \times \dots \times \mathcal{V}_{\delta_{|\mathcal{E}|}}$, so as to examine the geometric performances, such as placement area, aspect ratio, width, and length. This was done in Section 4.3.3 using the constrained placement exploration algorithm of [SEG⁺08]. The use of a computationally costly rectangle packing algorithm is avoided here. Only circuit layout area will be estimated, while it will be assumed that any circuit width, length, and aspect ratio can be realized.

In the sixth step, lower and upper area estimates are derived by the mappings $\phi_{A, \text{wos}, \min}$ and $\phi_{A, \text{wos}, \max}$:

$$\begin{aligned} \mathbf{d}_\mathcal{E} &\xrightarrow{\phi_{A, \text{wos}, \min}} A_{\text{wos}, \min} = \sum_{\delta_k \in \mathcal{E}} A_{\min}(\delta_k) \\ \mathbf{d}_\mathcal{E} &\xrightarrow{\phi_{A, \text{wos}, \max}} A_{\text{wos}, \max} = \sum_{\delta_k \in \mathcal{E}} A_{\max}(\delta_k) \end{aligned} \quad (\text{A.8})$$

Complete utilization of placement area was assumed in (A.8). Area utilization is formally defined as follows for any arbitrary circuit placement \mathbf{p} :

$$u(\mathbf{p}) = \frac{\text{total area of device layout rectangles including margins in } \mathbf{p}}{\text{total area of circuit placement } \mathbf{p}} \quad (\text{A.9})$$

Actual area utilization is incomplete and is dependent on the placement constraints and the geometric specifications; these allow for only certain arrangements of the devices to construct valid placements. Area utilization is estimated, here, to be in a range denoted by $[u_{\min}, u_{\max}]$; (A.8) is modified to model incomplete area utilization:

$$\begin{aligned} \mathbf{d}_{\mathcal{E}} \xrightarrow{\phi_{A_{\text{wos,min}}}} A_{\text{wos,min}} &= \frac{1}{u_{\max}} \cdot \sum_{\delta_k \in \mathcal{E}} A_{\min}(\delta_k) \\ \mathbf{d}_{\mathcal{E}} \xrightarrow{\phi_{A_{\text{wos,max}}}} A_{\text{wos,max}} &= \frac{1}{u_{\min}} \cdot \sum_{\delta_k \in \mathcal{E}} A_{\max}(\delta_k) \end{aligned} \quad (\text{A.10})$$

The area utilization range must be estimated by the designer. Here, the layout synthesis flow of Chapter 4 is called for the starting vector, \mathbf{d}^0 , from which circuit sizing is initiated. The range of area utilization is selected from the set of placements generated for \mathbf{d}^0 and denoted by $\mathbf{P}_{-1,-1}$:

$$\mathbf{d}^0 = \begin{bmatrix} \mathbf{d}_{\mathcal{E}}^0 \\ \mathbf{d}_{\mathcal{EB}}^0 \end{bmatrix}; \quad \mathbf{d}_{\mathcal{E}}^0 \xrightarrow{(5.27), \text{ layout synthesis}} \mathbf{P}_{-1,-1}; \quad (\text{A.11})$$

$$u_{\min} = \min_{\mathbf{p} \in \mathbf{P}_{-1,-1}} u(\mathbf{p}); \quad u_{\max} = \max_{\mathbf{p} \in \mathbf{P}_{-1,-1}} u(\mathbf{p}); \quad u(\mathbf{p}) \text{ calculated by (A.9)}$$

As long as the range of area utilization does not change from $[u_{\min}, u_{\max}]$ during the progress of the search algorithm, then $A_{\text{ws}}(\mathbf{p}) \in [A_{\text{wos,min}}, A_{\text{wos,max}}]$ for any placement, \mathbf{p} , generated during the search.

If the circuit layout is surrounded by an additional margin for pin placement and routing, for using a guard ring, or for any similar addition, then the estimates in (A.10) can be augmented to account for this in a straightforward manner.

Only area is considered in the estimate of the modified area objective, \hat{A}_{wos} , since it is assumed that any circuit width, length, and aspect ratio can be realized. In the seventh step, a single area estimate is selected from the range $[A_{\text{wos,min}}, A_{\text{wos,max}}]$ by a linear combination of the range bounds:

$$\hat{A}_{\text{wos}} = (1 - \rho) \cdot A_{\text{wos,min}} + \rho \cdot A_{\text{wos,max}}; \quad \rho \in [0, 1] \quad (\text{A.12})$$

The constant ρ must be selected by the designer.

A high value of ρ means the area estimate is pessimistic. A pessimistic estimate means that the corresponding circuit layout is more likely to meet the geometric specifications in lieu of the unconsidered placement constraints and changes in area utilization during the execution of the search algorithm. However, tradeoff with the electrical performances, represented by objective function of (5.10), must also be recognized. More effort would be needed to find a circuit sizing solution to meet both the geometric and the electrical performance specification when ρ is large.

A low value of ρ means the area estimate is optimistic. In this case, it is easier to find a circuit sizing solution. However, the geometric specification may be unsatisfied after layout synthesis at the solution. In which case, circuit sizing would have to be repeated with a larger value of ρ .

Appendix B

Approximation to the Gradient of the Area Estimate

A procedure to estimate the modified area objective is given in Appendix A. From (A.12), the estimate, \hat{A}_{wos} , is a linear combination of an upper and a lower area bound. These bounds are denoted by $A_{\text{wos,min}}$ and $A_{\text{wos,max}}$ respectively. The gradient (transposed) of \hat{A}_{wos} with respect to the circuit design parameters is given by:

$$\begin{aligned} \mathbf{J}_{\hat{A}_{\text{wos}},\mathbf{d}_{\mathcal{E}}}(\mathbf{d}) &= (1 - \rho) \cdot \frac{\partial \phi_{A_{\text{wos,min}}}(\mathbf{d})}{\partial \mathbf{d}_{\mathcal{E}}^T} + \rho \cdot \frac{\partial \phi_{A_{\text{wos,max}}}(\mathbf{d})}{\partial \mathbf{d}_{\mathcal{E}}^T} \\ &= (1 - \rho) \cdot \mathbf{J}_{A_{\text{wos,min}},\mathbf{d}_{\mathcal{E}}}(\mathbf{d}) + \rho \cdot \mathbf{J}_{A_{\text{wos,max}},\mathbf{d}_{\mathcal{E}}}(\mathbf{d}) \end{aligned} \quad (\text{B.1})$$

The test bench design parameters, $\mathbf{d}_{\mathcal{B}}$, do not alter the layout geometry, so that:

$$\mathbf{J}_{\hat{A}_{\text{wos}},\mathbf{d}_{\mathcal{B}}}(\mathbf{d}) = \mathbf{0}; \quad \mathbf{J}_{\hat{A}_{\text{wos}}} = \begin{bmatrix} \mathbf{J}_{\hat{A}_{\text{wos}},\mathbf{d}_{\mathcal{E}}}(\mathbf{d}); \mathbf{0} \end{bmatrix} \quad (\text{B.2})$$

The functions $\phi_{A_{\text{wos,min}}}$ and $\phi_{A_{\text{wos,max}}}$ are discontinuous, so that $\mathbf{J}_{\hat{A}_{\text{wos}},\mathbf{d}_{\mathcal{E}}}(\mathbf{d})$ does not strictly exist. Nevertheless, a forward finite difference approximation, $\widehat{\mathbf{J}}_{\hat{A}_{\text{wos}},\mathbf{d}_{\mathcal{E}}}(\mathbf{d})$, to $\mathbf{J}_{\hat{A}_{\text{wos}},\mathbf{d}_{\mathcal{E}}}(\mathbf{d})$ is derived in this Appendix.

In (A.10), the circuit area estimates $A_{\text{wos,min}}$ and $A_{\text{wos,max}}$ are constructed from the design parameter vector, $\mathbf{d}_{\mathcal{E}}$. Let $\mathbf{d}_{\mathcal{E},\text{discrete,min}}$ and $\mathbf{d}_{\mathcal{E},\text{discrete,max}}$ denote the vectors of discrete circuit design parameters corresponding to $A_{\text{wos,min}}$ and $A_{\text{wos,max}}$. These discrete vectors can be extracted and stored during the execution of the area estimation algorithm of Appendix A:

$$\begin{array}{l} \mathbf{d}_{\mathcal{E}} \xrightarrow{\phi_{A_{\text{wos,min}}}} A_{\text{wos,min}} \xrightarrow{\text{discretization}} \mathbf{d}_{\mathcal{E},\text{original,discrete,min}} \xrightarrow{(5.26)} \mathbf{d}_{\mathcal{E},\text{discrete,min}} \\ \mathbf{d}_{\mathcal{E}} \xrightarrow{\phi_{A_{\text{wos,max}}}} A_{\text{wos,max}} \xrightarrow{\text{discretization}} \mathbf{d}_{\mathcal{E},\text{original,discrete,max}} \xrightarrow{(5.26)} \mathbf{d}_{\mathcal{E},\text{discrete,max}} \end{array} \quad (\text{B.3})$$

For $j = 1, \dots, n_{\mathbf{d}_{\mathcal{E}}}$, let h_j be the finite difference step taken in the direction of the j -th circuit design parameter. In a similar manner, let $A_{\text{wos,min},j}$ and $A_{\text{wos,max},j}$ be the

area estimates derived for $(\mathbf{d}_\mathcal{E} + h_j \cdot \mathbf{e}_j)$, and let $\mathbf{d}_{\mathcal{E},\text{discrete},\text{min},j}$ and $\mathbf{d}_{\mathcal{E},\text{discrete},\text{max},j}$ be the corresponding discrete parameter vectors. Once more, the estimation algorithm in Appendix A can be used:

$$\begin{aligned} \mathbf{d}_\mathcal{E} + h_j \cdot \mathbf{e}_j &\xrightarrow{\phi_{A_{\text{wos},\text{min},j}}} A_{\text{wos},\text{min},j} \xrightarrow{\text{discretization}} \mathbf{d}_{\mathcal{E},\text{original},\text{discrete},\text{min},j} \xrightarrow{(5.26)} \mathbf{d}_{\mathcal{E},\text{discrete},\text{min},j} \\ \mathbf{d}_\mathcal{E} + h_j \cdot \mathbf{e}_j &\xrightarrow{\phi_{A_{\text{wos},\text{max},j}}} A_{\text{wos},\text{max},j} \xrightarrow{\text{discretization}} \mathbf{d}_{\mathcal{E},\text{original},\text{discrete},\text{max},j} \xrightarrow{(5.26)} \mathbf{d}_{\mathcal{E},\text{discrete},\text{max},j} \end{aligned} \quad (\text{B.4})$$

Using (5.66), the discretization error in $\mathbf{d}_\mathcal{E}$ and $(\mathbf{d}_\mathcal{E} + h_j \cdot \mathbf{e}_j)$ is calculated:

$$\mathbf{d}_{\mathcal{E},\text{error},\text{min}} = \mathbf{d}_{\mathcal{E},\text{discrete},\text{min}} - \mathbf{d}_\mathcal{E}; \quad \mathbf{d}_{\mathcal{E},\text{error},\text{min},j} = \mathbf{d}_{\mathcal{E},\text{discrete},\text{min},j} - (\mathbf{d}_\mathcal{E} + h_j \cdot \mathbf{e}_j) \quad (\text{B.5})$$

From (B.5):

$$\mathbf{d}_{\mathcal{E},\text{discrete},\text{min},j} - \mathbf{d}_{\mathcal{E},\text{discrete},\text{min}} = \underbrace{\mathbf{d}_{\mathcal{E},\text{error},\text{min},j} - \mathbf{d}_{\mathcal{E},\text{error},\text{min}}}_{\Delta \mathbf{d}_{\mathcal{E},\text{error},\text{min},j}} + h_j \cdot \mathbf{e}_j \quad (\text{B.6})$$

Define

$$\Delta A_{\text{wos},j} = A_{\text{wos},\text{min},j} - A_{\text{wos},\text{min}} \quad (\text{B.7})$$

To approximate the Jacobian $\mathbf{J}_{A_{\text{wos},\text{min},\mathbf{d}_\mathcal{E}}(\mathbf{d})}$, a linear system of equations is built to relate the difference in the discrete design parameter vectors, given in (B.6), and the corresponding difference in area estimate, defined in (B.7). This is similar to what was constructed in equation (5.104) for the electrical performances. For $j=1, \dots, n_{\mathbf{d}_\mathcal{E}}$, the approximation $\widehat{\mathbf{J}}_{A_{\text{wos},\text{min},\mathbf{d}_\mathcal{E}}(\mathbf{d})}$ to $\mathbf{J}_{A_{\text{wos},\text{min},\mathbf{d}_\mathcal{E}}(\mathbf{d})}$ is given by the following system:

$$\widehat{\mathbf{J}}_{A_{\text{wos},\text{min},\mathbf{d}_\mathcal{E}}(\mathbf{d})} \cdot ([\Delta \mathbf{d}_{\mathcal{E},\text{error},\text{min},1}, \dots, \Delta \mathbf{d}_{\mathcal{E},\text{error},\text{min},n_{\mathbf{d}_\mathcal{E}}}] + \mathbf{H}) = [\Delta A_{\text{wos},\text{min},1}, \dots, \Delta A_{\text{wos},\text{min},n_{\mathbf{d}_\mathcal{E}}}] \quad (\text{B.8})$$

Where

$$\mathbf{H} = \begin{bmatrix} h_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & h_{n_{\mathbf{d}_\mathcal{E}}} \end{bmatrix} \quad (\text{B.9})$$

In order to solve (B.8) for $\widehat{\mathbf{J}}_{A_{\text{wos},\text{min},\mathbf{d}_\mathcal{E}}(\mathbf{d})}$, the step sizes, $\{h_1, \dots, h_j, \dots, h_{n_{\mathbf{d}_\mathcal{E}}}\}$, must be selected so that the matrix $[\Delta \mathbf{d}_{\mathcal{E},\text{error},\text{min},1}, \dots, \Delta \mathbf{d}_{\mathcal{E},\text{error},\text{min},n_{\mathbf{d}_\mathcal{E}}}]$ is nonsingular. Using (5.68), it is possible to define a lower bound for the step size, so as to *guarantee* that the latter matrix is nonsingular:

$$h_j > 2 \cdot \underbrace{\mathbf{1}_{n_{\mathbf{d}_\mathcal{E}}}^T \cdot \mathbf{d}_{\mathcal{E},\text{error-max}}}_{\text{a constant}}; \quad j = 1, \dots, n_{\mathbf{d}_\mathcal{E}}; \quad \mathbf{1}_{n_{\mathbf{d}_\mathcal{E}}} \in \{1\}^{n_{\mathbf{d}_\mathcal{E}}} \quad (\text{B.10})$$

The approximation $\widehat{\mathbf{J}}_{A_{\text{wos},\text{max},\mathbf{d}_\mathcal{E}}(\mathbf{d})}$ to $\mathbf{J}_{A_{\text{wos},\text{max},\mathbf{d}_\mathcal{E}}(\mathbf{d})}$ can be derived in a similar manner and is omitted here. Finally, the complete approximation is given by:

$$\widehat{\mathbf{J}}_{A_{\text{wos},\mathbf{d}_\mathcal{E}}(\mathbf{d})} = (1 - \rho) \cdot \widehat{\mathbf{J}}_{A_{\text{wos},\text{min},\mathbf{d}_\mathcal{E}}(\mathbf{d})} + \rho \cdot \widehat{\mathbf{J}}_{A_{\text{wos},\text{max},\mathbf{d}_\mathcal{E}}(\mathbf{d})} \quad (\text{B.11})$$

Bibliography

- [ABD03] G. Alpaydin, S. Balkir, G. Dunder: *An evolutionary approach to automatic synthesis of high-performance analog integrated circuits*, IEEE Transactions on Evolutionary Computation, volume 7(3), pages 240–252, June 2003.
- [AEG⁺00a] K. Antreich, J. Eckmueller, H. Graeb, M. Pronath, F. Schenkel, R. Schwencker, S. Zizala: *WiCkeD: analog circuit synthesis incorporating mismatch*, in: *Proceedings of the IEEE Custom Integrated Circuits Conference*, pages 511–514, May 2000.
- [AEG⁺00b] K. Antreich, J. Eckmueller, H. Graeb, M. Pronath, F. Schenkel, R. Schwencker, S. Zizala: *WiCkeD: Analog circuit synthesis incorporating mismatch*, in: *Proceedings of the IEEE Custom Integrated Circuits Conference*, pages 511–514, May 2000.
- [AGW94] K. Antreich, H. Graeb, C. Wieser: *Circuit analysis and optimization driven by worst-case distances*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 13(1), pages 57–71, January 1994.
- [AKSW06] C. J. Alpert, A. B. Kahng, C. N. Sze, Q. Wang: *Timing-driven steiner trees are (practically) free*, in: *Proceedings of the 43rd annual ACM Conference on Design Automation*, pages 389–392, 2006.
- [AM08] S. Aghnout, N. Masoumi: *The effect of substrate noise on a 5.2 GHz LC-Tank VCO performance in a lightly doped substrate*, in: *The 3rd International Conference on Design and Technology of Integrated Systems in Nanoscale Era*, pages 1–6, March 2008.
- [AN07] K. Agarwal, S. Nassif: *Characterizing process variation in nanometer CMOS*, in: *Proceedings of the 44th Design Automation Conference*, pages 396–399, 2007.
- [ANvLT05] A.-J. Annema, B. Nauta, R. van Langevelde, H. Tuinhout: *Analog circuits in ultra-deep-submicron cmos*, IEEE Journal of Solid-State Circuits, volume 40(1), pages 132–143, January 2005.
- [Ars96] B. G. Arsintescu: *A method for analog circuits visualization*, in: *Proceedings of the International Conference on Computer Design, VLSI in Computers and Processors*, pages 454–459, 1996.
- [ARSR96] N. Arora, K. Raol, R. Schumann, L. Richardson: *Modeling and extraction of interconnect capacitances for multilayer VLSI circuits*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 15(1), pages 58–67, January 1996.

- [Bar90] T. Barnes: *SKILL: a CAD system extension language*, in: *Proceedings of the 27th IEEE/ACM International Conference on Design Automation*, pages 266–271, June 1990.
- [BJS05] S. Bhattacharya, N. Jangkrajarn, C.-J. Shi: *Template-driven parasitic-aware optimization of analog integrated circuit layouts*, in: *Proceedings of the 42nd Design Automation Conference*, pages 644–647, June 2005.
- [BMM⁺04] F. Balasa, S. C. Maruvada, S. Member, K. Krishnamoorthy, S. Member: *On the exploration of the solution space in analog placement with symmetry constraints*, *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, volume 23, pages 177–191, February 2004.
- [BMR07] S. Bhunia, S. Mukhopadhyay, K. Roy: *Process variations and process-tolerant design*, in: *VLSI Design, 2007. Held jointly with the 6th International Conference on Embedded Systems*, pages 699–704, January 2007.
- [BSMD08] S. Bandyopadhyay, S. Saha, U. Maulik, K. Deb: *A simulated annealing-based multiobjective optimization algorithm: AMOSA*, *IEEE Transactions on Evolutionary Computation*, volume 12(3), pages 269–283, June 2008.
- [BSV04] F. D. Bernardinis, A. Sangiovanni-Vincentelli: *A methodology for system-level analog design space exploration*, in: *Proceedings of the IEEE Conference on Design, Automation and Test in Europe*, pages 676–677, February 2004.
- [Cad] Cadence Design Systems, Inc, Available at www.cadence.com, *SubstrateStorm*.
- [Cad03a] Cadence Design Systems, Inc, *IC Shape-Based Technology Chip Assembly User Guide*, June 2003.
- [Cad03b] Cadence Design Systems, Inc, *Neolinear Inc. (acquired by Cadence in 2004)*, June 2003.
- [Cad05] Cadence Design Systems, Inc, Available at www.cadence.com, *Assura Physical Verification Developer Guide*, May 2005.
- [Cad08] Cadence Design Systems, Inc, Available at www.cadence.com, *Virtuoso Relative Object Design User Guide*, Oct 2008.
- [CC96] C.-K. Choi, H.-J. Chung: *Error estimates and adaptive time stepping for various direct time integration methods*, *Computers & Structures*, volume 60(6), pages 923–944, 1996.
- [CCWW00] Y.-C. Chang, Y.-W. Chang, G.-M. Wu, S.-W. Wu: *B*-trees: a new representation for non-slicing floorplans*, in: *Proceedings of the 37th Design Automation Conference*, pages 458–463, 2000.
- [CGRC91] J. Cohn, D. Garrod, R. Rutenbar, L. Carley: *KOAN/ANAGRAM II: new tools for device-level analog placement and routing*, *IEEE Journal of Solid-State Circuits*, volume 26(3), pages 330–342, March 1991.

- [CHA⁺92] J.-H. Chern, J. Huang, L. Arledge, P.-C. Li, P. Yang: *Multilevel metal capacitance models for CAD design synthesis systems*, IEEE Electron Device Letters, volume 13(1), pages 32–34, January 1992.
- [CLGRF08] R. Castro-Lopez, O. Guerra, E. Roca, F. Fernandez: *An integrated layout-synthesis approach for analog ICs*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 27(7), pages 1179–1189, July 2008.
- [CLL⁺06] F. Chen, B. Li, T. Lee, C. Christiansen, J. Gill, M. Angyal, M. Shinosky, C. Burke, W. Hasting, R. Austin, T. Sullivan, D. Badami, J. Aitken: *Technology reliability qualification of a 65nm CMOS Cu/Low-k BEOL interconnect*, in: *The 13th International Symposium on Physical and Failure Analysis of Integrated Circuits*, pages 97–105, July 2006.
- [CLW10] J.-E. Chen, P.-W. Luo, C.-L. Wey: *Placement optimization for yield improvement of switched-capacitor analog integrated circuits*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 29(2), pages 313–318, February 2010.
- [CM10] D. M. Causon, C. G. Mingham: *Introductory Finite Difference Methods for PDEs*, BookBoon, 2010.
- [CMSV93] E. Charbon, E. Malavasi, A. Sangiovanni-Vincentelli: *Generalized constraint generation for analog circuit design*, in: *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 408–414, November 1993.
- [CS92] D. Chen, B. Sheu: *Generalised approach to automatic custom layout of analogue ICs*, Circuits, Devices and Systems, IEE Proceedings G, volume 139(4), pages 481–490, August 1992.
- [CSV93] U. Choudhury, A. Sangiovanni-Vincentelli: *Automatic generation of parasitic constraints for performance-constrained physical design of analog circuits*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 12(2), pages 208–224, February 1993.
- [CSX⁺05] J. Cong, J. R. Shinnerl, M. Xie, T. Kong, X. Yuan: *Large-scale circuit placement*, ACM Transactions on Design Automation of Electronic Systems, volume 10(2), pages 389–430, 2005.
- [DB10] F. Dorfler, F. Bullo: *Spectral analysis of synchronization in a lossless structure-preserving power network model*, in: *First IEEE International Conference on Smart Grid Communications*, pages 179–184, October 2010.
- [DCR05] T. Dastidar, P. Chakrabarti, P. Ray: *A synthesis system for analog circuits based on evolutionary search and topological reuse*, IEEE Transactions on Evolutionary Computation, volume 9(2), pages 211–224, April 2005.
- [DE73] G. Dantzig, B. Eaves: *Fourier-Motzkin elimination and its dual*, Journal of Combinatorial Theory, volume 14, pages 288–297, 1973.

- [DGS03] W. Daems, G. Gielen, W. Sansen: *Simulation-based generation of posynomial performance models for the sizing of analog integrated circuits*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 22(5), pages 517–534, May 2003.
- [dMHBL98] M. del Mar Hershenson, S. P. Boyd, T. H. Lee: *GPCAD: A tool for CMOS Op-Amp synthesis*, in: *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 1998.
- [dMHBL01] M. del Mar Hershenson, S. P. Boyd, T. H. Lee: *Optimal design of a CMOS Op-Amp via geometric programming*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 20(1), pages 1–21, January 2001.
- [DNAV99] N. Dhanwada, A. Nunez-Aldana, R. Vemuri: *Hierarchical constraint transformation using directed interval search for analog system synthesis*, in: *Proceedings of the IEEE Conference on Design, Automation and Test in Europe*, 1999.
- [DR69] S. Director, R. Rohrer: *The generalized adjoint network and network sensitivities*, IEEE Transactions on Circuit Theory, volume 16(3), pages 318–323, August 1969.
- [DV03] A. Daboli, R. Vemuri: *Behavioral modeling for high-level synthesis of analog and mixed-signal systems from VHDL-AMS*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 22(11), pages 1504–1520, November 2003.
- [DV09] A. Das, R. Vemuri: *A graph grammar based approach to automated multi-objective analog circuit design*, in: *Proceedings of the IEEE Conference on Design, Automation and Test in Europe*, pages 700–705, April 2009.
- [EDGS03] T. Eeckelaert, W. Daems, G. Gielen, W. Sansen: *Generalized posynomial performance modeling [analog ics]*, in: *Proceedings of the IEEE Conference on Design, Automation and Test in Europe*, pages 250–255, 2003.
- [EGB06] T. Eschbach, W. Günther, B. Becker: *Orthogonal hypergraph drawing for improved visibility*, Journal of Algorithms and Applications, volume 10, pages 141–157, 2006.
- [EK96] H. Esbensen, E. Kuh: *Design space exploration using the genetic algorithm*, in: *IEEE International Symposium on Circuits and Systems*, volume 4, pages 500–503, May 1996.
- [EKV95] C. C. Enz, F. Krummenacher, E. A. Vittoz: *An analytical MOS transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications*, Analog Integrated Circuits and Signal Processing Journal, volume 8(1), pages 83–114, July 1995.
- [Ell11] W. Ellens: *Effective resistance and other graph measures for network robustness*, Master’s thesis, Mathematical Institute, University of Leiden, 2011.

- [ESGS10] M. Eick, M. Strasser, H. E. Graeb, U. Schlichtmann: *Automatic generation of hierarchical placement rules for analog integrated circuits*, in: *Proceedings of the 19th international symposium on Physical design*, pages 47–54, 2010.
- [Esh92] K. Eshbaugh: *Generation of correlated parameters for statistical circuit simulation*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 11(10), pages 1198–1206, October 1992.
- [ESL⁺11] M. Eick, M. Strasser, K. Lu, U. Schlichtmann, H. E. Graeb: *Comprehensive generation of hierarchical placement rules for analog integrated circuits*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 30(2), pages 180–193, February 2011.
- [ETP89] F. El-Turky, E. Perry: *BLADES: an artificial intelligence approach to analog circuit design*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 8(6), pages 680–692, June 1989.
- [FF95] P. Feldmann, R. Freund: *Efficient linear circuit analysis by pade approximation via the lanczos process*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 14(5), pages 639–649, May 1995.
- [For98] B. Fornberg: *Calculation of weights in finite difference formulas*, SIAM Review, volume 40, pages 685–691, 1998.
- [GBS] A. Ghosh, S. Boyd, A. Saberi: *Minimizing effective resistance of a graph*, SIAM Review.
- [GCY99] P.-N. Guo, C.-K. Cheng, T. Yoshimura: *An O-tree representation of non-slicing floorplan and its applications*, in: *Proceedings of the 36th Design Automation Conference*, pages 268–273, 1999.
- [GDWM⁺08] G. Gielen, P. De Wit, E. Maricau, J. Loeckx, J. Martín-Martínez, B. Kaczer, G. Groeseneken, R. Rodríguez, M. Nafría: *Emerging yield and reliability challenges in nanometer CMOS technologies*, in: *Proceedings of the IEEE Conference on Design, Automation and Test in Europe*, pages 1322–1327, March 2008.
- [GE95] R. Guindi, M. Elmasry: *High-level analog synthesis using signal flow graph transformations*, in: *Proceedings of the 8th IEEE International ASIC Conference*, pages 366–369, September 1995.
- [GH10] W. Gao, R. Hornsey: *A power optimization method for CMOS Op-Amps using sub-space based geometric programming*, in: *Proceedings of the IEEE Conference on Design, Automation and Test in Europe*, pages 508–513, March 2010.
- [GMGS09] H. Graeb, D. Mueller-Gritschneider, U. Schlichtmann: *Pareto optimization of analog circuits considering variability*, International Journal of Circuit Theory and Applications, volume 37(2), pages 283–299, March 2009.

- [GMW81] P. Gill, W. Murray, M. H. Wright: *Practical Optimization*, Academic Press, London, 1981.
- [Goo60] L. A. Goodman: *On the exact variance of products*, *Journal of the American Statistical Association*, volume 55(292), pages 708–713, 1960.
- [GS61] R. E. Griffith, R. A. Stewart: *A nonlinear programming technique for the optimization of continuous processing systems*, *Management Science*, volume 7, pages 379–392, 1961.
- [GWS90] G. Gielen, H. Walscharts, W. Sansen: *Analog circuit design optimization based on symbolic simulation and simulated annealing*, *IEEE Journal of Solid-State Circuits*, volume 25(3), pages 707–713, June 1990.
- [GZEA01] H. Graeb, S. Zizala, J. Eckmueller, K. Antreich: *The sizing rules method for analog integrated circuit design*, in: *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 343–349, November 2001.
- [Has01] A. Hastings: *The Art of Analog Layout*, Prentice Hall, 2001.
- [HDC⁺04] Q. Hao, S. Dong, S. Chen, X. Hong, Y. Su, Z. Qu: *Constraints generation for analog circuits layout*, in: *IEEE International Conference on Communications, Circuits and Systems*, volume 2, pages 1339–1343, June 2004.
- [Hey04] P. Heydari: *Analysis of the PLL jitter due to power/ground and substrate noise*, *IEEE Transactions on Circuits and Systems I: Regular Papers*, volume 51(12), pages 2404–2416, December 2004.
- [HG11] H. Habal, H. Graeb: *Constraint-based layout-driven sizing of analog circuits*, *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, volume 30(8), pages 1089–1102, August 2011.
- [HHC⁺00] X. Hong, G. Huang, Y. Cai, J. Gu, S. Dong, C.-K. Cheng, J. Gu: *Corner block list: an effective and efficient topological representation of non-slicing floorplan*, in: *Proceedings of the IEEE/ACM international conference on Computer-aided design*, pages 8–12, 2000.
- [HJBRS05] R. Hartono, N. Jangkrajarn, S. Bhattacharya, C. Richard Shi: *Automatic device layout generation for analog layout retargeting*, in: *Proceedings of the 18th IEEE International Conference on VLSI Design*, pages 457–462, January 2005.
- [HMF05] H. Habal, K. Mayaram, T. Fiez: *Accurate and efficient simulation of synchronous digital switching noise in systems on a chip*, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, volume 13(3), pages 330–338, March 2005.
- [HRC89] R. Harjani, R. Rutenbar, L. Carley: *OASYS: A framework for analog circuit synthesis*, *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, volume 8, pages 1247–1266, 1989.

- [HRM08] J. Hu, J. A. Roy, I. L. Markov: *Sidewinder: a scalable ILP-based router*, in: *Proceedings of the international workshop on System level interconnect prediction*, pages 73–80, 2008.
- [HS73] F. Harary, A. Schwenk: *The number of caterpillars*, *Discrete Math*, volume 6, pages 359–365, 1973.
- [Inf08] Infineon Technologies AG, *TML – Titan Modeling Language*, January 2008.
- [Ins10] A. Inselberg: *Parallel coordinates: Visual multidimensional geometry and its applications*, *ACM Special Interest Group on Software Engineering Notes*, volume 35, pages 39–39, May 2010.
- [JZB⁺06] N. Jangkrajarn, L. Zhang, S. Bhattacharya, N. Kohagen, C.-J. Shi: *Template-based parasitic-aware optimization and retargeting of analog and rf integrated circuit layouts*, in: *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 342–348, November 2006.
- [KCJ⁺00] K. Kundert, H. Chang, D. Jefferies, G. Lamant, E. Malavasi, F. Sendig: *Design of mixed-signal systems-on-a-chip*, *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, volume 19(12), pages 1561–1571, December 2000.
- [KD95] K. Krishna, S. Director: *The linearized performance penalty (LPP) method for optimization of parametric yield and its reliability*, *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, volume 14(12), pages 1557–1568, December 1995.
- [Kes95] G. Kesidis: *Analog optimization with Wong’s stochastic neural network*, *IEEE Transactions on Neural Networks*, volume 6(1), pages 258–260, January 1995.
- [KKC⁺08] H.-S. Kim, J. Kim, C. Chung, J. Lim, J. Jeong, J. H. Joe, J. Park, K.-W. Park, H. Oh, J. S. Yoon: *Effects of parasitic capacitance, external resistance, and local stress on the RF performance of the transistors fabricated by standard 65-nm cmos technologies*, *IEEE Transactions on Electron Devices*, volume 55(10), pages 2712–2717, October 2008.
- [KL00] S. Kapur, D. Long: *Large-scale capacitance calculation*, in: *Proceedings of the Design Automation Conference*, pages 744–749, 2000.
- [KLBS01] W. Kao, C.-Y. Lo, M. Basel, R. Singh: *Parasitic extraction: current state of the art and future trends*, *Proceedings of the IEEE*, volume 89(5), pages 729–739, May 2001.
- [KR93] D. J. Klein, M. Randi: *Resistance distance*, *Mathematical Chemistry*, volume 12(1), pages 81–95, 1993.
- [KSH94] M. Kole, J. Smit, O. Herrmann: *Modeling symmetry in analog electronic circuits*, in: *IEEE International Symposium on Circuits and Systems*, volume 1, pages 315–318, May 1994.

- [Kun95a] K. S. Kundert: *The Designer's Guide to SPICE & SPECTRE*, Kluwer Academic Publishers, 1995.
- [Kun95b] K. S. Kundert: *The Designer's Guide to Spice and Spectre*, Kluwer Academic Publishers, Norwell, MA, USA, 1995.
- [KWY96] T. Koide, S. Wakabayashi, N. Yoshida: *Pin assignment with global routing for VLSI building block layout*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 15(12), pages 1575–1583, December 1996.
- [LBSG07] B. Linares-Barranco, T. Serrano-Gotarredona: *On an efficient CAD implementation of the distance term in Pelgrom's mismatch model*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 26(8), pages 1534–1538, August 2007.
- [LCL09] P.-H. Lin, Y.-W. Chang, S.-C. Lin: *Analog placement based on symmetry-island formulation*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 28, pages 791–804, June 2009.
- [LD81] M. R. Lightner, S. W. Director: *Multiple criterion optimization for the design of electronic circuits*, IEEE Transactions on Circuits and Systems I, volume 28(3), pages 169–179, March 1981.
- [LD89] K. Low, S. Director: *A new methodology for the design centering of ic fabrication processes*, in: *IEEE International Conference on Computer-Aided Design, Digest of Technical Papers*, pages 194–197, November 1989.
- [LDH⁺08] J. Liu, S. Dong, X. Hong, Y. Wang, O. He, S. Goto: *Symmetry constraint based on mismatch analysis for analog layout in SOI technology*, in: *Proceedings of the Asia and South Pacific Design Automation Conference*, pages 772–775, March 2008.
- [LGS95] K. Lampaert, G. Gielen, W. Sansen: *A performance-driven placement tool for analog integrated circuits*, IEEE Journal of Solid-State Circuits, volume 30(7), pages 773–780, July 1995.
- [LGXP04] X. Li, P. Gopalakrishnan, Y. Xu, L. T. Pileggi: *Robust analog RF circuit design with projection-based posynomial modeling*, in: *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 855–862, November 2004.
- [LHC86] K. Lakshmikummar, R. Hadaway, M. Copeland: *Characterisation and modeling of mismatch in MOS transistors for precision analog design*, IEEE Journal of Solid-State Circuits, volume 21(6), pages 1057–1066, December 1986.
- [LHI09] T.-Y. Lo, C.-C. Hung, M. Ismail: *A wide tuning range GmC filter for multi-mode CMOS direct-conversion wireless receivers*, IEEE Journal of Solid-State Circuits, volume 44(9), pages 2515–2524, September 2009.

- [LJX⁺] W. Liu, X. Jin, X. Xi, J. Chen, M.-C. Jeng, Z. Liu, Y. Cheng, K. Chen, M. Chan, K. Hui, J. Huang, R. Tu, P. K. Ko, C. Hu: *BSIM3v3.3 MOSFET Model*, Available at www.device.eecs.berkeley.edu/~bsim3.
- [LZ10] Z. Liu, L. Zhang: *A performance-constrained template-based layout retargeting algorithm for analog integrated circuits*, in: *Proceedings of the 15th Asia and South Pacific Design Automation Conference*, pages 293–298, January 2010.
- [Mag06] Magma Design Automation, Inc, Available at www.magma-da.com, *QuickCap*, 2006.
- [Mar63] D. W. Marquardt: *An algorithm for least-squares estimation of nonlinear parameters*, *Journal of the Society for Industrial and Applied Mathematics*, volume 11(2), pages 431–441, 1963.
- [Mata] MathWorks, Available at www.mathworks.com, *Simulink – Simulation and Model-Based Design*.
- [Matb] MathWorks, Available at www.mathworks.com, *Simulink RF Toolbox 2.8*.
- [MCFSV96] E. Malavasi, E. Charbon, E. Felt, A. Sangiovanni-Vincentelli: *Automation of IC layout with analog constraints*, *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, volume 15(8), pages 923–942, August 1996.
- [MCR00] T. Mukherjee, L. Carley, R. Rutenbar: *Efficient handling of operating range and manufacturing line variations in analog cell synthesis*, *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, volume 19(8), pages 825–839, August 2000.
- [MFDCRV94] F. Medeiro, F. V. Fernandez, Dominguez-Castro, A. Rodriguez-Vasquez: *A statistical optimization-based approach for automated sizing of analog cells*, in: *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 1994.
- [MFNK96] H. Murata, K. Fujiyoshi, S. Nakatake, Y. Kajitani: *VLSI module placement based on rectangle-packing by the sequence-pair*, *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, volume 15(12), pages 1518–1524, December 1996.
- [MGS07] D. Mueller, H. Graeb, U. Schlichtmann: *Trade-off design of analog circuits using goal attainment and "wave front" sequential quadratic programming*, in: *Proceedings of the conference on Design, automation and test in Europe*, pages 75–80, 2007.
- [MGS08] T. Massier, H. Graeb, U. Schlichtmann: *The sizing rules method for CMOS and bipolar analog integrated circuit synthesis*, *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, volume 27(12), pages 2209–2222, December 2008.

- [MI92] C. Michael, M. Ismail: *Statistical modeling of device mismatch for analog MOS integrated circuits*, IEEE Journal of Solid-State Circuits, volume 27(2), pages 154–166, February 1992.
- [MV01] P. Mandal, V. Visvanathan: *CMOS Op-Amp sizing using a geometric programming formulation*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 20(1), pages 22–38, January 2001.
- [MYP07] Q. Ma, E. F. Y. Young, K. P. Pun: *Analog placement with common centroid constraints*, in: *Proceedings of the IEEE/ACM international conference on Computer-aided design*, pages 579–585, 2007.
- [Nag75] L. W. Nagel: *SPICE2: A computer program to simulate semiconductor circuits*, Technical report, University of California, Berkeley, Electronic Research Lab., May 1975.
- [Nes83] Y. Nesterov: *A method of solving a convex programming problem with convergence rate $o(1/k^2)$* , Soviet Mathematics Doklady, volume 2(27), pages 372–376, 1983.
- [NFMK96] S. Nakatake, K. Fujiyoshi, H. Murata, Y. Kajitani: *Module placement on BSG-structure and IC layout applications*, in: *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 484–491, November 1996.
- [NRSVT88] W. Nye, D. Riley, A. Sangiovanni-Vincentelli, A. Tits: *DELIGHT.SPICE: An optimization-based system for the design of integrated circuits*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 7, pages 501–519, April 1988.
- [NSS85] S. Nahar, S. Sahni, E. Shragowitz: *Experiments with simulated annealing*, in: *Proceedings of the 22nd ACM/IEEE Conference on Design Automation*, pages 748–752, 1985.
- [OBA⁺03] B. Owens, P. Birrer, S. Adluri, R. Shreeve, S. Arunachalam, H. Habal, S. Hsu, A. Sharma, K. Mayaram, T. Fiez: *Strategies for simulation, measurement and suppression of digital noise in mixed-signal circuits*, in: *Proceedings of the IEEE Custom Integrated Circuits Conference*, pages 361–364, September 2003.
- [OCP98] A. Odabasioglu, M. Celik, L. Pileggi: *PRIMA: passive reduced-order interconnect macromodeling algorithm*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 17(8), pages 645–654, August 1998.
- [ORC96] E. Ochotta, R. Rutenbar, L. Carley: *Synthesis of high-performance analog circuits in ASTRX/OBLX*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 15(3), pages 273–294, March 1996.
- [Par06] V. Pareto: *Manuale d'economia politica*, Societa Editrice Libratia, Milano, 1906.

- [PCL96] J. Phillips, E. Chiprout, D. Ling: *Efficient full-wave electromagnetic analysis via model-order reduction of fast integral transforms*, in: *Proceedings of the 33rd Design Automation Conference*, pages 377–382, June 1996.
- [PCLX01] Y. Pang, C.-K. Cheng, K. Lampaert, W. Xie: *Rectilinear block packing using O-tree representation*, in: *Proceedings of the International Symposium on Physical Design*, pages 156–161, 2001.
- [PDML94] J. Power, B. Donnellan, A. Mathewson, W. Lane: *Relating statistical MOSFET model parameter variabilities to ic manufacturing process fluctuations enabling realistic worst case design*, *IEEE Transactions on Semiconductor Manufacturing*, volume 7(3), pages 306–318, August 1994.
- [PDW89] M. Pelgrom, A. Duinmaijer, A. Welbers: *Matching properties of MOS transistors*, *IEEE Journal of Solid-State Circuits*, volume 24(5), pages 1433–1439, October 1989.
- [PG11] M. Pehl, H. Graeb: *An SQP and Branch-and-Bound Based Approach for Discrete Sizing of Analog Circuits*, chapter 13, pages 297–316, InTech, February 2011.
- [PKR⁺00] R. Phelps, M. Krasnicki, R. Rutenbar, L. Carley, J. Hellums: *Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search*, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, volume 19(6), pages 703–717, June 2000.
- [PMGS08] M. Pehl, T. Massier, H. Graeb, U. Schlichtmann: *A random and pseudo-gradient approach for analog circuit sizing with non-uniformly discretized parameters*, in: *IEEE International Conference on Computer Design*, pages 188–193, October 2008.
- [PS05] J. Phillips, L. Silveira: *Poor man's TBR: a simple model reduction scheme*, *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, volume 24(1), pages 43–55, January 2005.
- [PT03] Y. Palaskas, Y. Tsvividis: *Dynamic range optimization of weakly nonlinear, fully balanced, Gm-C filters with power dissipation constraints*, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, volume 50(10), pages 714–727, October 2003.
- [PV09] A. Pradhan, R. Vemuri: *Efficient synthesis of a uniformly spread layout aware pareto surface for analog circuits*, in: *Proceedings of the 22nd IEEE International Conference on VLSI Design*, pages 131–136, January 2009.
- [Pyt09] *Python v2.7.2 documentation*, Available at docs.python.org, 2009.
- [PZG10] M. Pehl, M. Zwerger, H. Graeb: *Sizing analog circuits using an sqp and branch and bound based approach*, in: *IEEE International Conference on Electronics, Circuits and Systems*, December 2010.
- [RGR07] R. Rutenbar, G. Gielen, J. Roychowdhury: *Hierarchical modeling, optimization, and synthesis for system-level analog and RF designs*, *Proceedings of the IEEE*, volume 95(3), pages 640–669, March 2007.

- [RM08] J. Roy, I. Markov: *High-performance routing at the nanometer scale*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 27(6), pages 1066–1077, June 2008.
- [Sax07] P. Saxena: *Routing Congestion in VLSI Circuits: Estimation and Optimization*, Springer Publishing Company, Incorporated, 2007.
- [SCP07] A. Somani, P. Chakrabarti, A. Patra: *An evolutionary algorithm-based approach to automated design of analog and rf circuits using adaptive normalized cost functions*, IEEE Transactions on Evolutionary Computation, volume 11(3), pages 336–353, June 2007.
- [SEG⁺08] M. Strasser, M. Eick, H. Graeb, U. Schlichtmann, F. M. Johannes: *Deterministic analog circuit placement using hierarchically bounded enumeration and enhanced shape functions*, in: *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 306–313, November 2008.
- [SEGA99] R. Schwencker, J. Eckmueller, H. Graeb, K. Antreich: *Automating the sizing of analog CMOS circuits by consideration of structural constraints*, in: *Proceedings of the IEEE Conference on Design, Automation and Test in Europe*, pages 323–327, 1999.
- [SGA07] G. Stehr, H. Graeb, K. Antreich: *Analog performance space exploration by Normal-Boundary intersection and by Fourier-Motzkin Elimination*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 26(10), pages 1733–1748, October 2007.
- [SH68] H. Shichman, D. Hodges: *Modeling and simulation of insulated-gate field-effect transistor switching circuits*, IEEE Journal of Solid-State Circuits, volume 3(3), pages 285–289, September 1968.
- [SK06] B. Suman, P. Kumar: *A survey of simulated annealing as a tool for single and multiobjective optimization*, Journal of the Operational Research Society, volume 57, pages 1143–1160, October 2006.
- [Soo08] T. Soorapanth: *Multi-objective analog design via geometric programming*, in: *Proceedings of the 5th International IEEE Conference on Electrical Engineering, Electronics, Computer, Telecommunications and Information Technology*, volume 2, pages 729–732, May 2008.
- [SPS⁺03] G. Stehr, M. Pronath, F. Schenkel, H. Graeb, K. Antreich: *Initial sizing of analog integrated circuits by centering within topology-given implicit specifications*, Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pages 241–246, 2003.
- [SSGA00] R. Schwencker, F. Schenkel, H. Graeb, K. Antreich: *The generalized boundary curve — a common method for automatic nominal design centering of analog circuits*, in: *Proceedings of the IEEE Conference on Design, Automation and Test in Europe*, pages 42–47, 2000.

- [SSKJ87] B. Sheu, D. Scharfetter, P.-K. Ko, M.-C. Jeng: *BSIM: Berkeley short-channel IGFET model for MOS transistors*, IEEE Journal of Solid-State Circuits, volume 22(4), pages 558–566, August 1987.
- [SYL09] C.-W. Sham, E. F. Y. Young, J. Lu: *Congestion prediction in early stages of physical design*, ACM Transactions on Design Automation of Electronic Systems, volume 14, pages 12:1–12:18, January 2009.
- [TB97] L. N. Trefethen, D. Bau: *Numerical Linear Algebra*, SIAM, 1997.
- [tec09] *International Technology Roadmap for Semiconductors*, Available at www.public.itrs.net, 2009.
- [Tre96] L. N. Trefethen: *Finite Difference and Spectral Methods for Ordinary and Partial Differential Equations*, Available at web.comlab.ox.ac.uk/oucl/work/nick.trefethen, 1996.
- [vdS10] A. van der Schaft: *Characterization and partial synthesis of the behavior of resistive circuits at their terminals*, Systems & Control Letters, volume 59(7), pages 423–428, 2010.
- [vHBD⁺02] M. van Heijningen, M. Badaroglu, S. Donnay, G. Gielen, H. De Man: *Substrate noise generation in complex digital systems: efficient modeling and simulation methodology and experimental verification*, IEEE Journal of Solid-State Circuits, volume 37(8), pages 1065–1072, August 2002.
- [VLR] *Verilog-AMS Language Reference Manual: Analog & Mixed-Signal Extensions to Verilog-HDL – version 2.1.*, Available at www.verilog-ams.com.
- [VLv⁺95] P. Veselinovic, D. Leenaerts, W. van Bokhoven, F. Leyn, G. Gielen, W. Sansen: *A flexible topology selection program as part of an analog synthesis system*, in: *Proceedings of the European Design and Test Conference*, pages 119–123, 1995.
- [WCC03] G.-M. Wu, Y.-C. Chang, Y.-W. Chang: *Rectilinear block placement using b*-trees*, ACM Transactions on Design Automation of Electronic Systems, volume 8, pages 188–202, April 2003.
- [Wei02] J. A. C. Weideman: *Numerical integration of periodic functions: A few examples*, The American Mathematical Monthly, volume 109(1), pages 21–36, 2002.
- [WVN⁺06] M. White, D. Vu, D. Nguyen, R. Ruiz, Y. Chen, J. Bernstein: *Product reliability trends, derating considerations and failure mechanisms with scaled CMOS*, IEEE International Integrated Reliability Workshop Final Report, pages 156–159, September 2006.
- [XY09] L. Xiao, E. F. Y. Young: *Analog placement with common centroid and 1-D symmetry constraints*, in: *Proceedings of the Asia and South Pacific Design Automation Conference*, pages 353–360, 2009.

- [YD09] E. Yilmaz, G. Dunder: *Analog layout generator for CMOS circuits*, IEEE Transactions on Computer-Aided Design of Integrated Circuits, volume 28(1), pages 32–45, January 2009.
- [YKC⁺05] W.-K. Yeh, C.-C. Ku, S.-M. Chen, Y.-K. Fang, C. Chao: *Effect of extrinsic impedance and parasitic capacitance on figure of merit of RF MOSFET*, IEEE Transactions on Electron Devices, volume 52(9), pages 2054–2060, September 2005.
- [YL08] G. Yu, P. Li: *Yield-aware hierarchical optimization of large analog integrated circuits*, in: *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 79–84, November 2008.
- [YLWH04] W. Yu, L. Li, Z. Wang, X. Hong: *Improved 3-d hierarchical interconnect capacitance extraction for the analog integrated circuit*, in: *IEEE International Conference on Communications, Circuits and Systems*, volume 2, pages 1305–1309, June 2004.
- [YP02] B. Yang, J. Phillips: *Time-domain steady-state simulation of frequency-dependent components using multi-interval chebyshev method*, in: *Proceedings of the 39th Design Automation Conference*, pages 504–509, 2002.
- [YZ08] Y. Yang, H. Zhang: *Some rules on resistance distance with applications*, Journal of Physics A: Mathematical and Theoretical, volume 41(44), pages 445203, 2008.
- [ZS02] T. Zhang, S. S. Sapatnekar: *Optimized pin assignment for lower routing congestion after floorplanning phase*, in: *Proceedings of the international workshop on System-level interconnect prediction*, pages 17–21, 2002.

Nomenclature

\mathcal{S}	Set \mathcal{S}
$ \mathcal{S} $	Cardinality of set \mathcal{S}
\mathbb{R}	The set of real numbers
\mathbb{N}	The set of natural numbers
x	Scalar x
x^a	Specific value of scalar x from a predefined set
$ x $	Absolute value of scalar x
$\lfloor x \rfloor$	Round scalar x to the nearest integer
\mathbf{b}	Vector \mathbf{b}
$ \mathbf{d} $	Cardinality of vector \mathbf{d}
\mathbf{b}^a	Specific value of vector \mathbf{b} from a predefined set
$\mathbf{b}[k]$	k -th (scalar) element of vector \mathbf{b}
$\mathbf{b}^a[k]$	k -th (scalar) element of \mathbf{b}^a
\mathbf{b}^T	Transpose of vector \mathbf{b}
$\ \mathbf{d}\ $	Euclidean norm of vector \mathbf{d}
$\text{abs}(\mathbf{b})$	$ \text{abs}(\mathbf{d}) = \mathbf{b} $; for all $1 \leq i \leq \mathbf{b} $, $\text{abs}(\mathbf{d})[i] = \mathbf{d}[i] $
$\mathbf{a} \preceq \mathbf{b}$	$ \mathbf{a} = \mathbf{b} $; for all $1 \leq i \leq \mathbf{a} $, $a_i \leq b_i$
$\mathbf{a} \prec \mathbf{b}$	$ \mathbf{a} = \mathbf{b} $; $\mathbf{a} \preceq \mathbf{b}$; for some $1 \leq i \leq \mathbf{b} $, $a_i < b_i$
\mathbf{A}	Matrix \mathbf{A}
$ \mathbf{A} $	Cardinality of matrix \mathbf{A}
\mathbf{A}^a	Specific value of matrix \mathbf{A} from a predefined set
$\mathbf{A}[i, j]$	(Scalar) element of matrix \mathbf{A} at the i -th row and j -th column
$\mathbf{A}[i]$	Vector equal to the i -th row of matrix \mathbf{A}
$\mathbf{A}^a[i, j]$	(Scalar) element of matrix \mathbf{A}^a at the i -th row and j -th column
$\mathbf{A}^a[i]$	Vector equal to the i -th row of matrix \mathbf{A}^a
\mathbf{A}^T	Transpose of matrix \mathbf{A}
\mathbf{A}^{-1}	Inverse of Matrix \mathbf{A}
\mathbf{A}^+	Moore-Penrose pseudoinverse of Matrix \mathbf{A}
$\mathbf{0}$	Zero-valued vector or matrix; cardinality is inferred from context
$\mathbf{1}$	Vector with all elements equal to 1; cardinality is inferred from context
\mathbf{e}_k	Vector with k -th element equal to 1, other elements equal to 0

List of Figures

1.1	Simulator and optimizer interaction in automatic circuit sizing	5
1.2	Backtracking in the basic analog design flow	7
1.3	Analog design flow with layout-driven circuit sizing	9
2.1	NMOS differential pair	18
3.1	Cross section and layout of an NMOS transistor	24
4.1	New automatic layout synthesis flow	32
4.2	Layout parameters mapped to the layout of an NMOS transistor	33
4.3	Multiple device layouts for the same value of design parameters	34
4.4	Device example: W_{error} vs. n_f for $W = 100\mu\text{m}$ and $L = 0.7\mu\text{m}$	38
4.5	Common centroid configurations for two matched NMOS devices	40
4.6	Minimum margins between three devices	46
4.7	Pareto-optimal set of circuit placements	48
4.8	Circuit placements with unsatisfied geometric specifications	49
4.9	Plot of the modified area for a set of placements	50
4.10	A total order relation is applied to a set of placements	52
4.11	Placement routing with predefined pin locations	54
4.12	Two additional pin assignment options	54
4.13	NMOS differential pair in common centroid layout	58
4.14	Routing resistance network definitions	61
4.15	Example effective resistance space	67
4.16	Example effective resistance space: rescaling of \mathbf{R}^u to satisfy the electrical constraints	68
4.17	Example resistor space: the set of solutions to (4.67) are indicated	70
4.18	Example resistor space: the solution to (4.68) is indicated	71
4.19	Example tree with the maximum possible number of edges	74
5.1	The effect of circuit design parameter discretization and layout synthesis on PSRR	98
5.2	Illustration of the mapping in equation (5.82)	100
6.1	Folded cascode operational amplifier (FC-OA) topology	112
6.2	FC-OA topology with identified analog functional sub-blocks	113
6.3	FC-OA topology with superimposed placement constraints	113
6.4	Tunable operational transconductance amplifier (TOTA) topology	115

6.5	TOTA transconductance, G_m , versus tuning current, I_{tune}	117
6.6	Miller operational amplifier (MOA) topology	120
6.7	MOA topology with identified analog functional sub-blocks	120
6.8	MOA topology with superimposed placement constraints	122
6.9	Difference between estimated and actual area during FC-OA sizing . .	127
6.10	B*-trees used during FC-OA layout-driven sizing	129
6.11	FC-OA circuit layout example	130
6.12	TOTA circuit layout example	133
6.13	MOA Pareto-optimal performance tradeoffs	135
6.14	MOA circuit layout example	138

List of Tables

1.1	Performances and specifications of a CMOS operational amplifier . . .	8
2.1	CMOS device design parameters	14
2.2	Sizing rules of an NMOS differential pair based on [MGS08]	19
3.1	Layout techniques for matched CMOS devices	26
4.1	CMOS device layout parameters	33
4.2	Combinations of n_f and M that satisfy the geometrical constraints . . .	41
4.3	Ranking the combinations of n_f and M for matched devices	43
4.4	Post-layout electrical sizing rules of an NMOS differential pair	59
6.1	FC-OA test benches and electrical performances	114
6.2	TOTA test benches and electrical performances	119
6.3	MOA test benches and electrical performances	123
6.4	FC-OA circuit sizing results	126
6.5	FC-OA computational cost of circuit sizing	128
6.6	Comparison of FC-OA placement structure pre and post layout-driven sizing	129
6.7	TOTA circuit sizing results	131
6.8	TOTA computational cost of circuit sizing	132
6.9	MOA circuit sizing results	134
6.10	MOA computational cost of circuit sizing	137

Abstract in German

Vorge stellt wird ein Verfahren zur automatischen Synthese analoger Schaltungen ausgehend von einer Netzliste und einer Menge von Entwurfsparametern. Die neue Methode basiert ausschließlich auf Entwurfs-, Platzierungs- und Verdrahtungsbedingungen, und benötigt, im Gegensatz zu existierenden Verfahren, keine Layout-Vorlagen. Die Synthese ist Teil einer nichtlinearen, Layout-orientierten Optimierung zur Schaltungsdimensionierung. Das Verfahren wurde zur Layout-orientierten Dimensionierung von mehreren CMOS-Schaltungen, darunter ein großes Analog-Subsystem mit 52 Transistoren, angewandt. Im Vergleich zu Verfahren, die Platzierung und Verdrahtung nicht berücksichtigen, konnten deutliche Verbesserungen hinsichtlich der Erfüllung der Optimierungsziele erreicht werden.

