

Universidade Federal do Pampa

Lucas Compassi Severo

Uma Ferramenta para o Dimensionamento Automático de Circuitos Integrados
Analógicos Considerando Análise de Produtividade

Alegrete

2012

Lucas Compassi Severo

**Uma Ferramenta para o Dimensionamento Automático
de Circuitos Integrados Analógicos Considerando Análise
de Produtividade**

Dissertação apresentada ao Programa de Pós-graduação Stricto Sensu em Engenharia Elétrica da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Mestre em Engenharia Elétrica.

Orientador: Alessandro Gonçalves Girardi

Alegrete

2012

Ficha catalográfica elaborada automaticamente com os dados fornecidos
pelo(a) autor(a) através do Módulo de Biblioteca do
Sistema GURI (Gestão Unificada de Recursos Institucionais) .

S498f Severo, Lucas Compassi

Uma Ferramenta para o Dimensionamento Automático de
Circuitos Integrados Analógicos Considerando Análise de
Produtividade / Lucas Compassi Severo.

127 p.

Dissertação(Mestrado)-- Universidade Federal do Pampa,
MESTRADO EM ENGENHARIA ELÉTRICA, 2012.

"Orientação: Alessandro Gonçalves Girardi".

1. CMOS. 2. Circuitos Integrados Analógicos. 3. Ferramentas
de CAD. 4. Otimização de Produtividade. I. Título.

Lucas Compassi Severo

**Uma Ferramenta para o Dimensionamento Automático de Circuitos Integrados Analógicos
Considerando Análise de Produtividade**

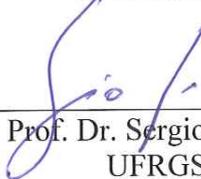
Dissertação apresentada ao Programa de Pós-graduação
Stricto Sensu em Engenharia Elétrica da Universidade
Federal do Pampa, como requisito parcial para obtenção
do Título de Mestre em Engenharia Elétrica.

Área de concentração: Sistemas de Energia

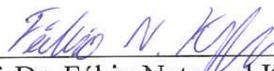
Dissertação defendida e aprovada em: 22 de Novembro de 2012.
Banca examinadora:



Prof. Dr. Alessandro Gonçalves Girardi
Orientador
UNIPAMPA



Prof. Dr. Sergio Bampi
UFRGS



Prof. Dr. Fábio Natanael Kepler
UNIPAMPA



Prof. Dr. Jumar Luis Russi
UNIPAMPA



Prof. Dr. Marcos Vinicio Thomas Heckler
UNIPAMPA

A minha querida Tanísia, meus pais Altamir e Marli
e a minha irmã Tamíris.

Agradecimentos

Ao nos depararmos com os momentos de conclusões de trabalhos, é de grande importância lembrar que todos os desafios e oportunidades que temos no dia-a-dia, sejam profissionais ou pessoais, nós não os vencemos sozinhos, pois sempre há alguém para, de qualquer modo, ajudar-nos a atingir nossas metas e realizarmos nossos sonhos.

Desta forma, em homenagem a todos que de alguma forma colaboraram com o desenvolvimento deste trabalho, desde a Graduação até o Mestrado, eu manifesto o meu sincero agradecimento.

Agradeço ao professor Alessandro Girardi pela orientação junto ao Grupo de Arquitetura de Computadores e Microeletrônica (GAMA), aos desafios em mim impostos e a dicas de organização, os quais com certeza foram de grande importância para o desenvolvimento do trabalho e para o meu processo de aprendizagem. Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro ao projeto. Ao Programa de Pós-Graduação em Engenharia Elétrica (PPGEE) pela oportunidade e o financiamento para participação em eventos acadêmicos. Agradeço aos colegas de Mestrado pelo apoio nas disciplinas bem como a amizade.

A meus pais, Altamir e Marli, e minha irmã Tamirís, que em toda a minha vida estiveram sempre me apoiando e dando-me forças para seguir na “batalha” de buscar conhecimento e concretizar um sonho. E a minha querida noiva Tanísia que está sempre ao meu lado nestas etapas de minha vida.

*“Os problemas nunca vão desaparecer, mesmo na mais bela existência.
Problemas existem para serem resolvidos, e não para perturbar-nos.”*

Augusto Cury

Resumo

A indústria de microeletrônica tem a sua evolução ditada pela necessidade cada vez maior de integração de circuitos como memórias e processadores, fazendo com que os dispositivos semicondutores sejam cada vez mais miniaturizados. Esta miniaturização implica processos de fabricação cada vez mais complexos, resultando em uma grande variabilidade de parâmetros. O projeto de circuitos analógicos torna-se cada vez mais complexo, pois em geral é altamente suscetível às variações de processo, o que afeta a sua produtividade. Uma das partes mais complexas deste projeto é o dimensionamento dos dispositivos que compõem o circuito, pois o espaço de projeto é altamente não-linear e nem sempre se conhece a localização do seu ponto ótimo. Neste contexto, este trabalho tem como objetivo o desenvolvimento de uma ferramenta para o dimensionamento automático de circuitos integrados analógicos, capaz de lidar com a variabilidade dos parâmetros e visando aumentar a produtividade do circuito gerado. Esta ferramenta baseia-se no dimensionamento do circuito como um problema de otimização baseado em simulação elétrica SPICE. O objetivo principal é receber as especificações requeridas de uma topologia de circuito e, através de técnicas de inteligência artificial, explorar o espaço de soluções em busca de soluções otimizadas que atendam às restrições impostas. Além disso, espera-se obter soluções que atendam às especificações requeridas mesmo com variações no processo de fabricação. Para isso, são empregadas técnicas de design centering de modo a maximizar a produtividade do circuito. A ferramenta desenvolvida foi implementada de maneira modular, permitindo que a análise do dimensionamento do circuito possa ser realizada sob diferentes aspectos. Como resultado, este trabalho apresenta duas topologias de amplificadores operacionais automaticamente dimensionadas em tecnologia CMOS, tendo como objetivo a minimização da área de gate e da potência dissipada, além da maximização da produtividade. Os circuitos gerados apresentaram melhor desempenho em comparação com resultados descritos na literatura.

Palavras-chaves: Circuitos Integrados Analógicos, Ferramenta de CAD, Dimensionamento Automático, Design Centering

Abstract

The microelectronics industry has the CMOS technology evolution dictated by the capability of integration of digital circuits such as memories and processors, causing the semiconductor devices miniaturization. The miniaturization leads to complex manufacturing processes with high parameters variation. Analog circuit designs are complex and highly susceptible to process variations, affecting the circuit yield. One of the most complex part of the analog design is the circuit sizing, since the possible solutions have a highly nonlinear design space and the optimal solution is not known. In this context, this work aims at developing a tool for the automatic sizing of analog integrated circuits that is able to deal with parameter variation in order to yield maximization. This tool is based on the circuit sizing as an optimization problem based on electrical SPICE simulations. The main objective is to receive the required specifications of a circuit topology and, by means of artificial intelligence techniques, to explore the design space for optimized solutions that meet the circuit constraints. Furthermore, it is expected to obtain solutions which meet the specifications required even with the presence of variations in the manufacturing process. For this purpose, design centering techniques are implemented for yield maximization. The tool is implemented with modular functions, enabling the sizing process on different configurations. As results, this work present the automatic design of two CMOS operational amplifiers topologies, with the goal to reduce the power dissipation and the gate area and to maximize the yield. The results present good performance when compared to similar designs found in literature.

Key-words: Analog Integrated Circuits, CAD Tools, Automatic Sizing, Design Centering.

Lista de ilustrações

Figura 1	Estrutura física de um transistor em tecnologia CMOS.	28
Figura 2	Fluxo de projeto de circuitos integrados analógicos.	29
Figura 3	Diagrama de um sistema automatizado para o projeto automático de circuitos integrados analógicos.	30
Figura 4	Diagrama de dimensionamento de circuitos integrados analógicos baseado na experiência do projetista.	32
Figura 5	Diagrama de dimensionamento de circuitos integrados analógicos baseado em um problema de otimização.	32
Figura 6	Diagrama de dimensionamento de circuitos integrados analógicos baseado em um problema de otimização com avaliação de desempenho.	34
Figura 7	Definição do dimensionamento do circuito como um problema de otimização. (a) otimização de um objetivo sujeito a restrições e (b) otimização de uma função custo composta por objetivos e restrições.	35
Figura 8	Principais variações no processo de fabricação de circuitos integrados em tecnologia CMOS. (a) flutuação aleatória de dopantes (YE et al., 2008) e (b) variação nas dimensões do canal de um transistor CMOS (DRENNAN; MCANDREW, 2003).	37
Figura 9	Análise de pior caso com relação à velocidade dos transistores CMOS.	40
Figura 10	Curva de uma função de densidade de probabilidade normal.	41
Figura 11	<i>Yield</i> em um <i>wafer</i> de silício.	42
Figura 12	Design centering através da maximização da distância entre as especificações requeridas e a encontrada.	43
Figura 13	Esquemático de um amplificador diferencial CMOS com carga ativa.	46
Figura 14	Modelo de pequenos sinais simplificado para o circuito amplificador diferencial CMOS.	47
Figura 15	Espaço de projeto para o amplificador diferencial de duas variáveis. O valor mínimo é igual a 0,1195802 e é obtido com $W = 1,62\mu m$ and $L = 0,55\mu m$. (a) espaço tridimensional e (b) espaço bidimensional com especificação da solução ótima.	51
Figura 16	Fluxograma de dimensionamento automático utilizando SA.	53
Figura 17	Evolução das funções de decaimento da temperatura do SA.	55

Figura 18	Número de soluções ótimas encontradas com relação ao tempo de execução para o processo de otimização global de um amplificador diferencial, considerando três funções de decaimento de temperatura e a função G_{FAST}	57
Figura 19	Número de soluções ótimas com relação ao tempo de otimização para a otimização global seguida de local para o amplificador diferencial utilizando função G_{FAST}	58
Figura 20	Número de resultados ótimos com relação ao tempo de otimização global do amplificador diferencial com diferentes valores de intervalo de <i>reannealing</i> , utilizando decaimento de temperatura pela função T_{BOLTZ}	59
Figura 21	Número de resultados ótimos com relação ao tempo de otimização global do amplificador diferencial com diferentes valores de intervalo de <i>reannealing</i> , utilizando decaimento de temperatura pela função T_{EXP}	60
Figura 22	Número de resultados ótimos com relação ao tempo de otimização global do amplificador diferencial com diferentes valores de intervalo de <i>reannealing</i> , utilizando decaimento de temperatura pela função T_{FAST}	61
Figura 23	Máximo número de soluções ótimas obtidas com a otimização global com relação ao intervalo de <i>reannealing</i> utilizado com função de decaimento de temperatura T_{FAST}	62
Figura 24	Número de resultados ótimos com relação ao tempo de otimização global do amplificador diferencial para as diferentes funções de geração de novas soluções.	63
Figura 25	Fluxograma de dimensionamento automático utilizando algoritmos genéticos (GA).	64
Figura 26	Comparação da probabilidade de seleção com as funções <i>roulette wheel</i> e ranqueamento geométrico normalizado.	65
Figura 27	Recombinação e mutação em cromossomos binários.	66
Figura 28	Função custo final obtida com a execução do GA com a combinação dos valores $5 \leq NPOP \leq 200$ com passos de 10 unidades e $10 \leq NGEN \leq 600$ com passos de 50 unidades.	69
Figura 29	Espaço de produtividade para o amplificador diferencial de duas variáveis. (a) espaço tridimensional e (b) espaço bidimensional.	75
Figura 30	Histograma de frequência para a análise de Monte Carlo com variações globais e locais, simultaneamente. (a) Histograma do ganho (Avo), (b) Histograma da Margem de fase (PM), (c) Histograma do GBW e (d) Histograma do ICMR+.	77

Figura 31	Histograma de frequência para a análise de Monte Carlo com variação local de parâmetros. (a) Histograma do ganho (Avo), (b) Histograma da Margem de fase (PM), (c) Histograma do GBW e (d) Histograma do ICMR+.	78
Figura 32	Espaço de Projeto do amplificador diferencial de duas variáveis para a função f_c^* . (a) espaço tridimensional e (b) espaço bidimensional com especificação da solução ótima.	80
Figura 33	Fluxograma da estratégia 1 de <i>design centering</i> baseado na simulação de Monte Carlo.	81
Figura 34	Fluxograma da estratégia 2 de <i>design centering</i> baseado na simulação Monte Carlo.	82
Figura 35	Fluxograma da metodologia proposta.	85
Figura 36	Diagrama de blocos da ferramenta proposta.	86
Figura 37	Fluxograma de execução do núcleo da ferramenta.	88
Figura 38	Tipos de topologias de circuitos presentes na ferramenta. (a) circuitos de entrada e saída em terminação simples (<i>single-ended</i>), (b) circuitos de entrada diferencial e saída em terminação simples e (c) circuitos de entrada e saída diferencial (<i>fully differential</i>).	89
Figura 39	Inserção de uma topologia de circuito através do seu <i>netlist</i> . (a) esquemático de um OTA Miller CMOS e (b) <i>netlist</i> referente ao esquemático mostrado em (a).	90
Figura 40	<i>Script</i> necessário para a configuração da topologia de circuito a ser dimensionado.	91
Figura 41	<i>Script</i> necessário para a configuração da heurística de otimização. (a) algoritmos genéticos e (b) <i>Simulated Annealing</i>	93
Figura 42	Função $R(S_j)$. (a) Função para especificação que requer valor superior ao valor mínimo e (b) Função para especificação que requer valor inferior ao valor máximo.	94
Figura 43	<i>Script</i> de configuração das especificações.	97
Figura 44	<i>Testbench</i> para medição das características AC em malha aberta.	98
Figura 45	Diagrama de Bode gerado com a simulação do <i>testbench</i> da Figura 44.	99
Figura 46	<i>Testbench</i> para medição do <i>slew rate</i> (<i>SR</i>).	99
Figura 47	Sinal de tensão de entrada e saída gerado com a simulação do <i>testbench</i> da Figura 46 para a medição do <i>slew rate</i> (<i>SR</i>).	100
Figura 48	<i>Testbench</i> para medição da faixa de tensão de entrada em modo comum (<i>ICMR</i>).	100
Figura 49	Sinal de tensão de saída gerado com a simulação do <i>testbench</i> da Figura 48.	101
Figura 50	<i>Testbench</i> para medição da faixa de tensão de saída (<i>OS</i>).	102

Figura 51	Sinal de tensão de saída gerado com a simulação do <i>testbench</i> da Figura 50 para a medição da faixa de tensão de saída.	102
Figura 52	Esquemático do OTA Miller em tecnologia CMOS.	105
Figura 53	Evolução da função custo para o melhor indivíduo da população durante o processo de otimização.	110
Figura 54	Histogramas de frequência para a simulação Monte Carlo no circuito OTA Miller projetado sem a utilização de <i>design centering</i> . (a) Avo, (b) GBW, (c) PM, (d) ICMR-, (e) ICMR+, (f) OS e (g) SR.	112
Figura 55	Histogramas de frequência para a simulação Monte Carlo no circuito OTA Miller projetado com a utilização de <i>design centering</i> . (a) Avo, (b) GBW, (c) PM, (d) ICMR-, (e) ICMR+, (f) OS e (g) SR.	115
Figura 56	Esquemático do amplificador <i>folded cascode</i> em tecnologia CMOS. . .	116
Figura 57	Fontes de corrente de polarização do circuito amplificador <i>folded cascode</i> da Figura 56. (a) fonte de polarização <i>vb</i> e (b) fonte de polarização <i>vbp</i>	116

Lista de tabelas

Tabela 1	Faixa de desvio padrão e probabilidade de uma distribuição normal . . .	41
Tabela 2	Especificações de projeto e restrições para o circuito amplificador diferencial da Figura 13	49
Tabela 3	Especificações nominais do circuito amplificador diferencial simplificado para a solução ótima no ponto $W = 1,62\mu m$ e $L = 0,55\mu m$. . .	52
Tabela 4	Funções de geração de novas soluções do SA	53
Tabela 5	Funções de decaimento de temperatura do SA	55
Tabela 6	Valores médios da otimização global do circuito amplificador diferencial para diferentes funções de decaimento de temperatura usando G_{FAST} .	56
Tabela 7	Valores médios do projeto do amplificador diferencial após a otimização global seguida de otimização local	57
Tabela 8	Resultados de execução do GA com cromossomos binários e reais . . .	70
Tabela 9	Resultados de execução do GA para as diferentes funções de seleção . .	71
Tabela 10	Resultados de execução do GA com as diferentes formas de recombinação	72
Tabela 11	Resultados de execução do GA com a utilização dos diferentes métodos de mutação	73
Tabela 12	Resultados obtidos com a otimização global seguida de local	73
Tabela 13	Análise do resultados de variações de parâmetros do circuito amplificador diferencial através da simulação Monte Carlo	76
Tabela 14	Resultados de execução da estratégia 1 de <i>design centering</i>	81
Tabela 15	Resultados de execução da estratégia 2 de <i>design centering</i>	83
Tabela 16	Especificações requeridas e encontradas na automação do projeto do circuito OTA Miller da Figura 52.	109
Tabela 17	Parâmetros encontrados na automação do projeto do circuito OTA Miller da Figura 52.	109
Tabela 18	Especificações requeridas e encontradas na automação do projeto do circuito OTA Miller da Figura 52 com a utilização de <i>design centering</i>	114
Tabela 19	Parâmetros encontrados na automação do projeto do circuito OTA Miller da Figura 52 utilizando <i>design centering</i>	114
Tabela 20	Especificações obtidas com o dimensionamento automático do circuito amplificador <i>folded cascode</i> da Figura 56.	118

Tabela 21	Resultado final para as variáveis livres de projeto obtidas com o dimensionamento automático do circuito amplificador <i>folded cascode</i> da Figura 56.	120
-----------	---	-----

Sumário

1	Introdução	23
1.1	Motivação	24
1.2	Objetivo	25
1.3	Organização do Trabalho	25
2	Projeto Automático de Circuitos Integrados Analógicos CMOS	27
2.1	Fluxo de Projeto de Circuitos Integrados Analógicos	27
2.2	Ferramentas de Automação do Projeto de Circuitos Integrados Analógicos	30
2.2.1	Dimensionamento de Circuitos Integrados Analógicos	31
2.2.2	Ferramentas Comerciais	35
2.3	Design Centering	36
2.4	Conclusão	44
3	Automação do Projeto de um Amplificador Diferencial	45
3.1	Amplificador Diferencial	45
3.2	Modelagem do Amplificador Diferencial para o Dimensionamento Auto- mático	49
3.3	Exploração do Espaço de Projeto de um Amplificador Diferencial através de Otimização	51
3.3.1	Otimização com Simulated Annealing (SA)	52
3.3.2	Otimização com Algoritmos Genéticos (GA)	60
3.4	Design Centering	74
3.4.1	Estratégias de Design Centering Adotadas	77
3.5	Conclusão	83
4	A Ferramenta UCAF	85
4.1	Interfaces de Entrada e Saída	86
4.2	Núcleo da Ferramenta	87
4.3	Tecnologia de Fabricação	87
4.4	Biblioteca de Circuitos	88
4.5	Otimização	91
4.5.1	Algoritmos Genéticos (GA)	92
4.5.2	Simulated Annealing (SA)	92
4.6	Função Custo	93
4.7	Especificações	95
4.8	Simulação Elétrica	96

4.8.1	Análise no domínio da frequência em malha aberta (SP001AC_type1) e potência dissipada (SP001PowerBias)	97
4.8.2	Velocidade de resposta (SP001TR_type1)	98
4.8.3	Tensão de entrada em modo comum (SP001DC_type1)	99
4.8.4	Faixa de tensão de saída (SP002DC_type1)	101
4.9	<i>Design Centering</i>	101
4.10	Conclusões	103
5	Resultados	105
5.1	Projeto Automático de um OTA Miller	105
5.1.1	Projeto Manual	107
5.1.2	Projeto Automático	107
5.1.3	Comparação dos Resultados	109
5.1.4	Aplicando o Design Centering ao Projeto	111
5.2	Projeto de um Amplificador Folded Cascode	113
5.3	Conclusão	119
	Referências	123
	ANEXO A Script de Interface de Entrada	127
	ANEXO B Relatório de Saída	129

1 Introdução

A era da tecnologia da informação começou após o desenvolvimento da microeletrônica. Este desenvolvimento possibilitou que os circuitos eletrônicos fossem cada vez mais integrados de forma a possibilitar o processamento cada vez maior da informação. Neste contexto, os circuitos integrados representaram uma grande importância no desenvolvimento tecnológico e na cultura da sociedade, além de representar uma grande fatia dos recursos econômicos de um país.

Os circuitos integrados são circuitos eletrônicos construídos em escalas micrométricas ou nanométricas sobre elementos semicondutores, tipicamente silício. Estes circuitos são classificados principalmente de acordo com o tipo de sinal que trabalham (circuitos analógicos, digitais ou mistos). Na atualidade, a maior parte dos circuitos são digitais, porém os circuitos analógicos sempre serão necessários, seja para a interface entre os sinais físicos, polarização de circuitos ou para a geração de sinais para circuitos digitais. Os circuitos analógicos são mais complexos do que os circuitos digitais quanto ao dimensionamento dos transistores que os compõem, uma vez que em circuitos digitais todos os transistores possuem tamanhos pré-definidos e em circuitos analógicos cada transistor deve ser dimensionado individualmente. Com relação ao número de transistores, os circuitos integrados digitais são mais complexos, podendo possuir até alguns bilhões de transistores, enquanto que os circuitos analógicos não passam de algumas centenas de transistores.

Segundo Graeb (2007), em 2005 cerca de 75% dos circuitos integrados possuíam blocos analógicos em sua construção. Nestes chips, cerca de 20% da área é ocupada por circuitos analógicos. Embora pequena, esta área representa 40% de todo o esforço de projeto. Além disso, as partes analógicas são responsáveis por cerca de 50% dos defeitos ocorridos no processo de desenvolvimento de um circuito integrado. Este percentual se deve basicamente ao fato de as ferramentas de automação do projeto de circuitos integrados analógicos serem pouco desenvolvidas quando comparado aos circuitos digitais. Assim, o projeto de circuitos mistos, compostos por partes analógicas e por partes digitais, apresenta uma grande desigualdade, uma vez que a parte digital é feita de forma totalmente automatizada com ferramentas de CAD (*Computer Aided Design*) e a parte analógica é feita em sua maioria manualmente (GIELEN; RUTENBAR, 2000).

Neste contexto, torna-se cada vez mais necessário o desenvolvimento de ferramentas para o projeto de circuitos integrados analógicos que explorem de maneira eficiente o espaço de projeto em busca de soluções que atinjam os requisitos impostos em sua apli-

cação. Nestas ferramentas deve-se sempre manter o foco na miniaturização dos circuitos e na menor dissipação de potência (LIU et al., 2009).

O projeto de um circuito integrado analógico, de maneira geral, pode ser dividido em três grandes partes: nível de sistema, nível de circuito e nível de leiaute (BALKIR; DUNDAR; OGRENCI, 2003). Estes níveis se referem à abstração do projeto. No primeiro nível (sistema) o circuito é tratado como um bloco funcional. No segundo nível o circuito é representado pelo esquemático de transistores. Já no nível de leiaute o circuito é tratado fisicamente através do desenho do dispositivo semicondutor. No nível de circuito, o objetivo do projeto é dimensionar cada um dos transistores que compõem um dado circuito de forma a satisfazer as restrições impostas ou especificações mínimas de projeto. Como as metodologias de projeto não são bem definidas, a qualidade do projeto manual de um circuito integrado analógico depende da experiência do projetista (CORTES et al., 2003; INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS - ITRS, 2011). Na automação do projeto de circuitos integrados analógicos existem inúmeros trabalhos de pesquisa com o objetivo de explorar o espaço de projeto com a utilização de técnicas de otimização não-linear e de inteligência artificial (HAGGLUND, 2003; JAFARI et al., 2010; LIU et al., 2009). Nestas ferramentas, comumente são utilizados simuladores elétricos tipo SPICE (*Simulated Program with Integrated Circuits Emphasis*) para prover informações acerca das soluções encontradas para um dado circuito baseando-se em modelos que representam o comportamento físico do dispositivo.

1.1 Motivação

A evolução da tecnologia de fabricação de circuitos integrados é acelerada pela indústria de memórias e processadores, as quais tem o foco cada vez maior na miniaturização dos dispositivos de forma que seja possível a inserção de um maior número de blocos lógicos em um mesmo circuito integrado.

Impulsionados pela evolução digital, os circuitos analógicos devem ser implementados nas mesmas tecnologias que os circuitos digitais de forma que seja possível a construção de circuitos mistos. Um dos problemas refere-se ao fato de os circuitos analógicos serem altamente susceptíveis a variações de parâmetros físicos dos dispositivos, variações estas fortemente presentes em tecnologias de maior escala de integração. A motivação do trabalho é colaborar para o desenvolvimento de circuitos integrados analógicos que consumam menor potência, ocupem menor área e atinjam as especificações impostas mesmo com a ocorrência de variações de parâmetros nos processos de fabricação.

1.2 Objetivo

Este trabalho tem por objetivo o desenvolvimento de uma ferramenta de CAD, chamada UCAF, para a automação do projeto de blocos analógicos integrados básicos voltada à exploração do espaço de projeto em nível de circuito através de heurísticas de inteligência artificial, de forma a encontrar soluções otimizadas que satisfaçam as especificações impostas ao circuito.

Além disso, este trabalho também tem por objetivo explorar o espaço de produtividade do circuito através de técnicas de *design centering*, possibilitando encontrar soluções que atinjam as especificações impostas aos circuitos mesmo com a ocorrência de variações nos processos de fabricação de circuitos integrados.

Como objetivo específico, deseja-se obter uma ferramenta composta por funções modulares, para que um mesmo circuito possa ter o seu espaço de projeto explorado de diversas formas. Com isso espera-se obter uma ferramenta científica para a pesquisa e análise de formas de automação do projeto analógico.

1.3 Organização do Trabalho

Esta dissertação está organizada em 6 capítulos. O Capítulo 2 apresenta uma revisão bibliográfica referente ao projeto de circuitos analógicos, ferramentas existentes para este tipo de projeto, análise das variações de parâmetros e técnicas de *design centering*. O Capítulo 3 trata da exploração do projeto de um amplificador diferencial simplificado utilizando heurísticas de otimização e técnicas de *design centering*. O Capítulo 4 aborda o desenvolvimento e implementação da ferramenta UCAF. O Capítulo 5 trata da análise dos resultados obtidos com a UCAF no projeto de duas topologias de amplificadores operacionais. Finalmente, o Capítulo 6 apresenta as conclusões deste trabalho.

2 Projeto Automático de Circuitos Integrados Analógicos CMOS

Este capítulo analisa o estado da arte do desenvolvimento de ferramentas de automação do projeto de circuitos integrados analógicos CMOS. Nesta análise, o projeto de circuitos analógicos será contextualizado primeiramente através do fluxo de projeto manual e, em seguida, através da utilização de ferramentas de automação do projeto.

Neste capítulo, é mostrada também uma breve análise da produtividade (*yield*) de um circuito analógico através das técnicas de *design centering*.

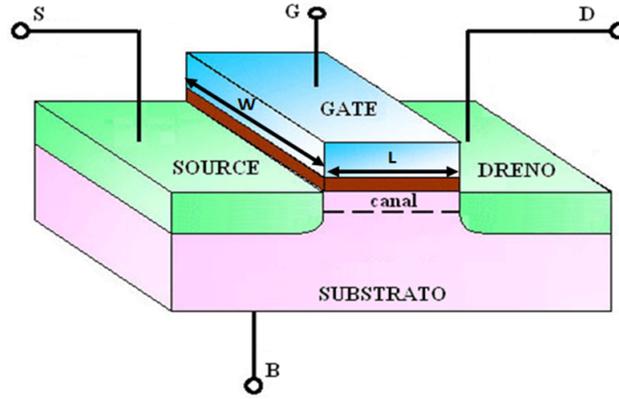
2.1 Fluxo de Projeto de Circuitos Integrados Analógicos

Os blocos analógicos de um circuito integrado representam a etapa mais crítica de um projeto (GRAEB, 2007). Atualmente, o projeto de um circuito integrado analógico ou dos blocos analógicos é feito, em sua maioria, manualmente por um projetista. Este projetista utiliza algumas ferramentas computacionais de auxílio ao projeto (CAD), como ambientes de simulação e editores de leiaute. Desta forma, cabe ao projetista entender o problema de aplicação do circuito integrado, a escolha e o projeto da topologia que possivelmente será implementada.

Segundo Balkir, Dundar e Ogrenci (2003), de maneira geral, o projeto de um circuito integrado analógico pode ser dividido em três grandes níveis de abstração: nível de sistema, nível de circuito e nível de leiaute. A Figura 2 mostra o fluxograma de projeto de um circuito integrado analógico. No de sistema, com base nas especificações do circuito, uma topologia de circuito é escolhida. Esta escolha leva em conta a experiência do projetista e também novas topologias propostas. Dada a topologia escolhida, os transistores CMOS que compõem o circuito e as fontes de polarização devem ser dimensionados em nível de circuito. O dimensionamento de um transistor CMOS consiste na obtenção do valor da largura (W) e o do comprimento (L) do canal do dispositivo, conforme mostra a Figura 1. Para o dimensionamento inicial, normalmente o projetista utiliza metodologias simplificadas de projeto, como a equação da corrente I_D do modelo quadrático (SPICE nível 1 ou nível 3) para transistores CMOS (ALLEN; HOLBERG, 2002), dada na Equação 2.1, ou metodologias mais elaboradas que analisam o circuito com base em curvas físicas válidas para todos os níveis de inversão (SILVEIRA; FLANDRE; JESPERS, 1996; CORTES et al., 2003). Na Equação 2.1, μ_0 é o parâmetro de mobilidade de portadores no canal, C_{OX}

é a capacitância de óxido isolante por unidade de área, V_T é a tensão de limiar, λ é o parâmetro de modulação no comprimento do canal e v_{GS} e v_{DS} são as tensões entre os terminais de gate e source e dreno e source, respectivamente.

Figura 1: Estrutura física de um transistor em tecnologia CMOS.



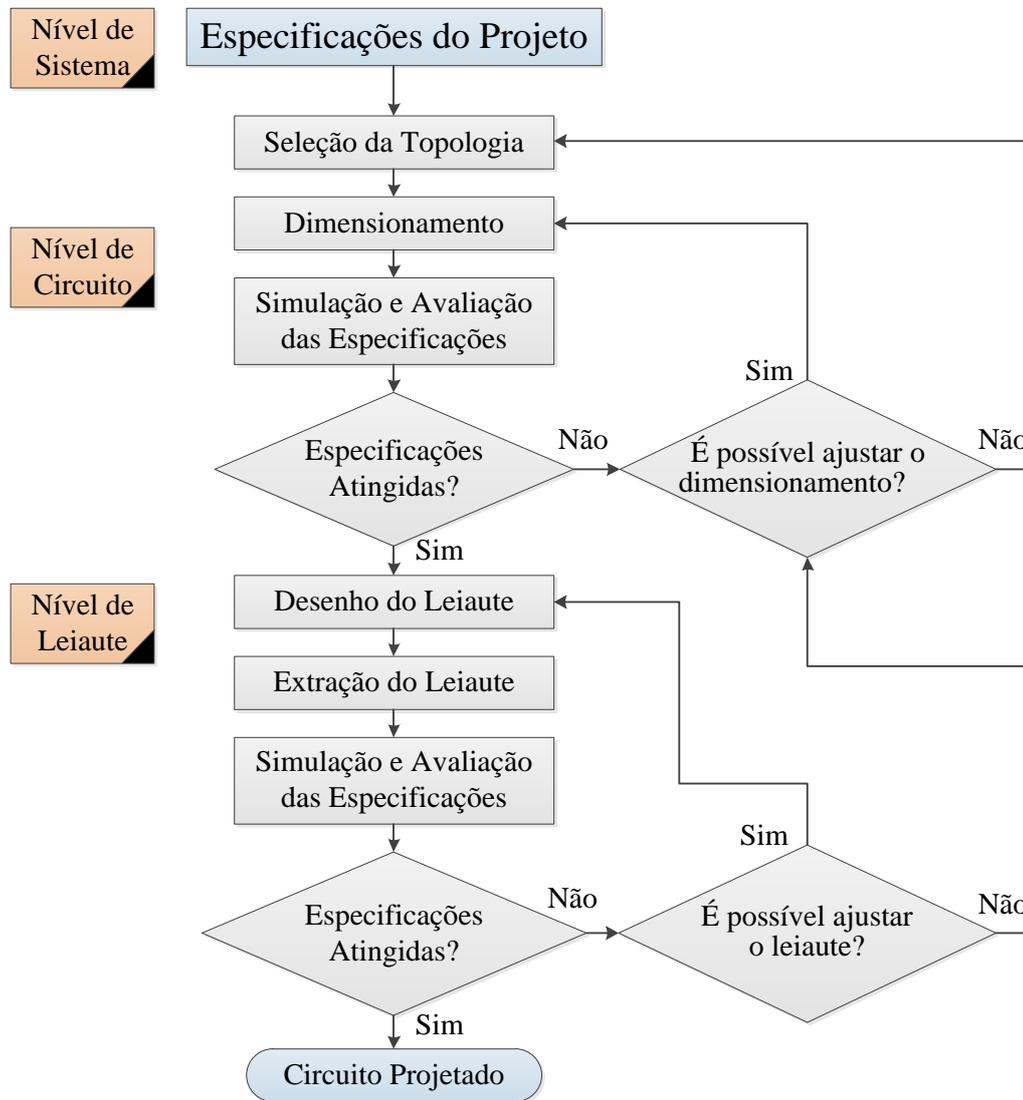
$$I_D = \begin{cases} \frac{\mu_0 \cdot C_{OX} \cdot W}{L} [(v_{GS} - V_T) - (\frac{v_{DS}}{2})] \cdot v_{DS} & \text{se } 0 \leq v_{DS} < v_{GS} - V_T \\ \frac{\mu_0 \cdot C_{OX} \cdot W}{2L} (v_{GS} - V_T)^2 \cdot (1 + \lambda v_{DS}) & \text{se } v_{DS} \geq v_{GS} - V_T \end{cases} \quad (2.1)$$

Após o dimensionamento inicial, as especificações do circuito devem ser obtidas através de simulação elétrica utilizando para isso simuladores elétricos tipo SPICE. Caso as especificações requeridas não sejam satisfeitas após o dimensionamento inicial, o dimensionamento deve sofrer ajustes até atingir as especificações desejadas. Neste passo, caso não seja possível atingir as especificações, deve-se voltar e escolher uma nova topologia. Se as especificações forem atingidas, o leiaute do circuito, em nível de dispositivos, deve ser feito. Para isso, utilizam-se editores de leiaute pré-configurados com as regras de projeto fornecidas pela *foundry*. Para o leiaute, o projetista deve analisar estratégias de desenho para que os circuitos mantenham suas características elétricas próximas dos valores nominais e tenham baixa variabilidade após a fabricação.

Após o desenho do leiaute, um circuito contendo os elementos parasitas, tais como capacitâncias e resistências, é extraído a partir do leiaute. Com a simulação deste circuito é possível verificar a funcionalidade da topologia projetada com resultados mais próximos da implementação física. Caso as especificações não sejam atingidas, existem três possibilidades de ajuste: ajuste do leiaute e repetição da extração do circuito e simulação, ajuste do dimensionamento do circuito ou a escolha de uma nova topologia. Caso as especificações sejam atingidas o circuito está projetado.

No fluxo de projeto de um circuito integrado analógico, mostrado na Figura 2, deve ainda haver algumas etapas como simulações do circuito sujeito à variabilidades no processo de fabricação, como análises de Monte Carlo e de pior caso.

Figura 2: Fluxo de projeto de circuitos integrados analógicos.



Como detalhado, o projeto de um circuito analógico necessita de uma série de passos e comumente todas as decisões e análises são feitas manualmente pelo projetista. A parte mais crítica do projeto trata-se do dimensionamento, uma vez que encontrar um ponto de polarização dos circuitos que atinja as especificações do projeto demanda inúmeras iterações de redimensionamentos e simulações. Na etapa de dimensionamento, devido às metodologias como um todo utilizarem equações simplificadas, os resultados iniciais tendem a estar longe dos resultados reais. Outro fato importante é que dados como potência e área do circuito são difíceis de ser otimizados por parte do projetista, uma vez que o principal problema é encontrar uma solução que atinja as especificações impostas ao circuito. Caso seja possível, são analisadas alternativas que reduzam a potência e a área do circuito.

O grande problema das técnicas de projeto manual de um circuito integrado analó-

gico é que o resultado depende do conhecimento ou da experiência por parte do projetista de circuitos integrados. Além disso, as metodologias de projeto manual são baseadas em tentativa e erro de forma que uma solução encontrada representa uma solução que atinge as especificações requeridas (GIRARDI, 2007). Neste processo não é possível afirmar se a solução encontrada é a melhor possível ou se existem soluções melhores.

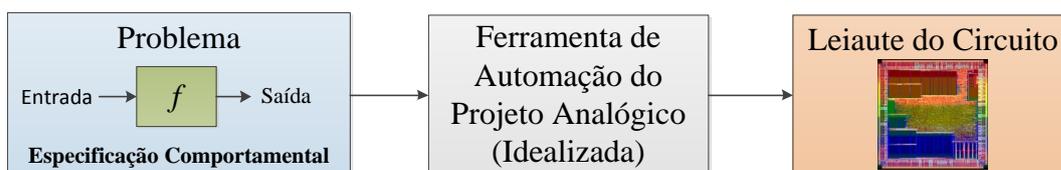
2.2 Ferramentas de Automação do Projeto de Circuitos Integrados Analógicos

Como abordado na seção anterior, a qualidade do projeto manual de um circuito integrado analógico está diretamente relacionado à experiência do projetista. Além disso, em técnicas de projeto manual, o projetista fica restrito a equações simplificadas e aproximações para o projeto. Assim, bons resultados são obtidos através de inúmeras iterações de ajuste de valores e simulação elétrica.

Partindo da ideia apresentada na seção anterior, uma ferramenta computacional para automação do projeto analógico poderia ser idealizada como o diagrama mostrado na Figura 3. Neste diagrama, partindo da descrição comportamental de um problema, obtém-se o leiaute do circuito que executa a função comportamental. Esta ferramenta idealizada serve como base para o desenvolvimento e análise das ferramentas existentes.

Partindo do problema que necessita de um bloco funcional analógico, o primeiro passo representa a escolha de uma topologia de circuito que pode executar tal função comportamental. Um circuito, ao executar tal função, apresentará não-idealidades, as quais podem ser estimadas através de simulação. O próximo passo é o dimensionamento do circuito. Neste passo, os parâmetros como W (largura do canal), L (comprimento do canal) e as fontes de polarização são dimensionadas baseando-se nas especificações requeridas para o circuito. Após o dimensionamento, o leiaute do circuito é gerado. Em alguns casos várias topologias de circuito podem resolver o problema, porém nem todas são capazes de atingir as especificações de projeto.

Figura 3: Diagrama de um sistema automatizado para o projeto automático de circuitos integrados analógicos.



Neste contexto, a automação do projeto analógico envolve uma sequência de projetos e análises de forma a encontrar um circuito realizável para o problema. Martens

e Gielen (2008) e Barros, Guilherme e Horta (2010) trazem análises das principais ferramentas desenvolvidas desde a década de 80 para a automação do projeto de circuitos integrados analógicos. Nestas análises, as ferramentas podem ser divididas de acordo com o nível de abstração. Nestes níveis, uma das partes mais complexas refere-se ao dimensionamento dos transistores CMOS do circuito (BALKIR; DUNDAR; OGRENCI, 2003). Neste conceito, a subseção seguinte analisa as principais formas de dimensionamento automático de circuitos integrados analógicos.

2.2.1 Dimensionamento de Circuitos Integrados Analógicos

No fluxo de projeto de um circuito analógico, após a escolha ou geração de uma topologia de circuito, o dimensionamento deve ser feito, de forma a obter os parâmetros de projeto da topologia. Este dimensionamento, de maneira geral, pode ser realizado de duas formas: baseado no conhecimento (*knowledge-based*) ou baseado em um problema de otimização (*optimization-based approach*). Na primeira forma, o dimensionamento do circuito é feito com base no conhecimento e experiência de um projetista. Conforme mostra a Figura 4, neste tipo de dimensionamento, dada uma topologia, é descrita uma metodologia de projeto. Esta metodologia utiliza equações simplificadas de projeto e estratégias de dimensionamento baseadas no conhecimento de um projetista. Com a metodologia de projeto e com os requisitos do projeto, o circuito é dimensionado. Uma ferramenta que pode ser classificada como dimensionamento baseado no conhecimento é a ferramenta PAD (*Procedural Analog Design*) proposta por Stefanovic e Kayal (2009). Nesta ferramenta as topologias de circuitos presentes em sua biblioteca podem ser dimensionadas através de equações do modelo elétrico EKV (ENZ, 2008) e de um procedimento de dimensionamento de cada bloco básico do circuito. Além do dimensionamento, a ferramenta possui uma interface para o ajuste fino de alguns parâmetros do circuito e possibilita a visualização do nível de inversão de cada transistor, *mismatch* do circuito, entre outras análises. O processo de dimensionamento baseado no conhecimento possui algumas desvantagens, uma vez que para cada topologia de circuito é necessário o desenvolvimento de uma metodologia de projeto. O desenvolvimento de uma metodologia de projeto exige um grande conhecimento da topologia e da tecnologia de fabricação por parte do projetista. Além disso, a metodologia baseia-se em equações, as quais tendem a ser simplificadas em circuitos mais complexos, gerando valores muitas vezes distantes de valores realizáveis (BARROS; GUILHERME; HORTA, 2010).

A outra forma de efetuar o dimensionamento automático de um circuito é transformar este processo em um problema de otimização. Conforme mostrado na Figura 5, a etapa de dimensionamento do circuito recebe uma topologia de circuito e um modelo de otimização. Através de um método de otimização o espaço de projeto do circuito é explorado em busca de soluções otimizadas que atinjam os requisitos impostos ao projeto.

Figura 4: Diagrama de dimensionamento de circuitos integrados analógicos baseado na experiência do projetista.

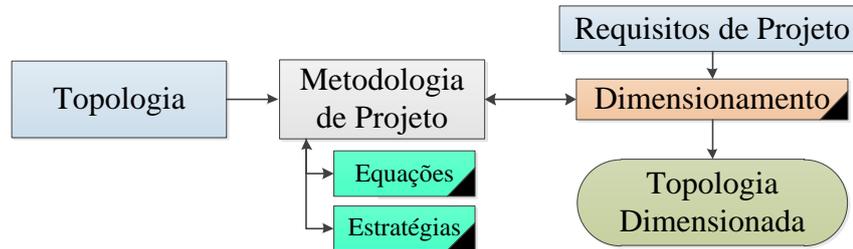
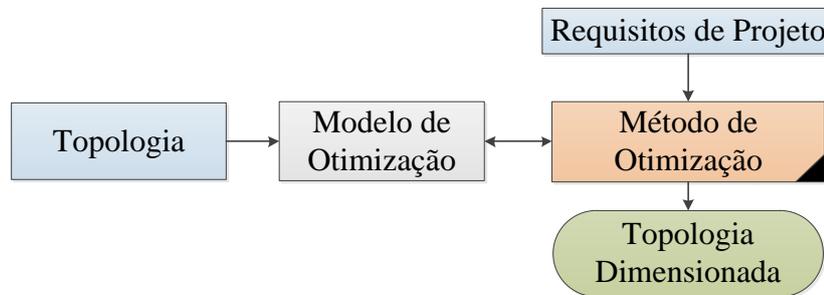


Figura 5: Diagrama de dimensionamento de circuitos integrados analógicos baseado em um problema de otimização.



Segundo Barros, Guilherme e Horta (2010) o método de otimização utilizado é dependente do modelo de otimização do projeto. Os modelos de otimização podem ser classificados em: baseado em equações (*equation-based*), baseado em simulação elétrica (*simulation-based*) e baseado em aprendizado (*learning-based*). O método baseado em equações utiliza equações simplificadas originadas da análise da topologia de circuito para grandes e pequenos sinais, como dados em Allen e Holberg (2002) e Razavi (2000). A utilização de modelos baseados em equações permite uma avaliação rápida das características elétricas do circuito, porém nem todas as especificações podem ser obtidas. Além disso, devido às simplificações, estes modelos podem ter pouca precisão. Os trabalhos de Jafari et al. (2010), Hershenson, Boyd e Lee (2001) e Deyati e Mandal (2011) utilizam modelos baseados em equações para o processo de dimensionamento. Jafari et al. (2010) utiliza a heurística dos algoritmos genéticos para explorar o espaço de projeto, dado por equações simplificadas, para um amplificador operacional de dois estágios. Em Hershenson, Boyd e Lee (2001) e Deyati e Mandal (2011), são utilizadas equações posinomiais para modelar circuitos amplificadores. Com esta abordagem, a exploração do projeto com algoritmos de programação geométrica garantem encontrar a solução ótima.

Os métodos baseados em simulação elétrica utilizam simuladores elétricos comerciais tipo SPICE como modelos de otimização. Para a simulação elétrica são utilizados

modelos elétricos para a tecnologia de fabricação fornecidos pela *foundry*. Este tipo de modelo de otimização tende a consumir um grande tempo computacional no processo de otimização, uma vez que para cada simulação o simulador elétrico executa inúmeras iterações de cálculo e, normalmente, é necessário medir várias especificações sobre diferentes estruturas de simulação. Embora o processo necessite de um grande tempo computacional, este método tende a ser preciso, uma vez que baseia-se em modelos elétricos complexos para a modelagem dos dispositivos. A simulação elétrica permite que qualquer especificação do circuito seja analisada, fazendo com que este tipo de modelo de otimização seja flexível. Além disso, este tipo de modelo permite a análise da variabilidade do circuito que está sendo dimensionado através de modelos de pior caso, Monte Carlo, entre outros. Outro fator importante da utilização de simulação elétrica é que esta pode ser utilizada por qualquer técnica de otimização. As ferramentas propostas por Phelps et al. (2000), Smedt e Gielen (2002) utilizam modelos baseados em simulação elétrica. Em Phelps et al. (2000), são utilizadas heurísticas de *Simulated Annealing* para explorar uma função de avaliação (função custo) multiobjetivo utilizando o simulador elétrico Cadence Spectre®. Smedt e Gielen (2002) abordam a exploração do espaço de projeto de circuitos analógicos e RFIC utilizando otimização multiobjetivo. Para definir a função custo multiobjetivo o trabalho analisa o *trade-off* entre as especificações.

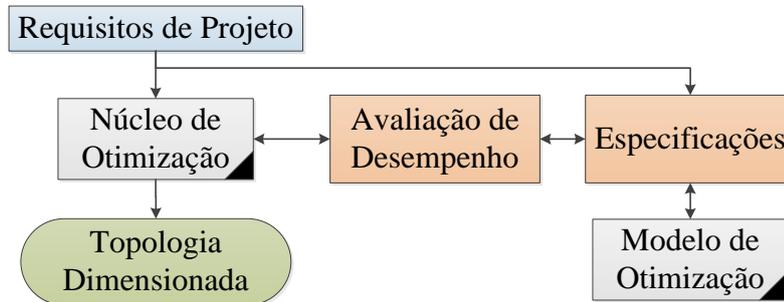
Além de modelos baseados em equações e simulação elétrica, existem modelos baseados em aprendizado de máquina, que fornecem uma avaliação rápida, como os modelos baseados em equações, e tem uma boa precisão, como os modelos baseados em simulação. Isto é obtido com a utilização de técnicas como Máquinas de Vetores de Suporte, como utilizado em Barros, Guilherme e Horta (2010), e Redes Neurais Artificiais, como utilizado em (ALPAYDIN; BALKIR; DUNDAR, 2003). Nestes métodos, primeiramente os modelos são treinados utilizando simulação elétrica e, após isso, o sistema pode ser utilizado para estimar as especificações de um dado circuito.

Com base no modelo de otimização, escolhe-se o método de otimização. Nos modelos baseados em equações pode-se utilizar métodos de otimização que explorem a continuidade deste tipo de modelo. Um exemplo disso são modelos de otimização baseados em equação posinomial (DEYATI; MANDAL, 2011), onde a aplicação de métodos de otimização baseados em programação geométrica garante a obtenção do resultado ótimo. Com modelos de otimização baseados em simulação elétrica, em geral, qualquer método de otimização pode ser utilizado. Segundo Martens e Gielen (2008), os métodos de otimização estocásticos podem ser utilizados em qualquer tipo de modelo de otimização. Em alguns métodos, a convergência para o resultado ótimo é garantida e em alguns casos existe uma grande probabilidade de o método encontrar a solução ótima global após várias iterações. Dentre os métodos de otimização se destacam a utilização de métodos baseados em otimização evolutiva, como Algoritmos Genéticos (JAFARI et al., 2010), e métodos baseados em *Simulated Annealing* (MARTENS; GIELEN, 2008). Estes dois métodos

possuem uma grande probabilidade de encontrar a solução ótima após várias iterações. Outro fator importante é que estes métodos, de maneira geral, têm uma solução final não dependente da solução inicial, sendo desnecessário um dimensionamento inicial da topologia. A grande desvantagem destes métodos é o fato de necessitar a execução de inúmeras iterações, fazendo com que o tempo de otimização seja longo.

A Figura 6 mostra o fluxograma de execução de uma ferramenta baseada em otimização com a avaliação do desempenho de cada solução. Nesta figura, a ferramenta recebe como entrada os requisitos de projeto. O núcleo de otimização gera soluções para o problema de otimização de acordo com a técnica de otimização envolvida. Para cada solução gerada é necessário avaliar o desempenho da solução. A avaliação do desempenho é feita através de uma função custo que contém as especificações do circuito e a relação com os valores requeridos para o projeto. Para obter os valores das especificações é utilizado o modelo de otimização. No caso de modelos baseados na simulação elétrica estas especificações são mensuradas através de configurações da topologia (*testbenches*) que permitem a análise das suas especificações.

Figura 6: Diagrama de dimensionamento de circuitos integrados analógicos baseado em um problema de otimização com avaliação de desempenho.



Neste contexto, o dimensionamento de circuitos analógicos baseados em um problema de otimização, tem sua formulação mostrada na Figura 7. A parte (a) desta figura apresenta a formulação básica de um processo de otimização, onde deve-se minimizar o somatório de todos os objetivos de otimização e satisfazer algumas restrições impostas ao projeto. Estas restrições referem-se às especificações ($S(X)$) de um dado circuito que devem satisfazer os valores mínimos ou máximos S_{ref} . Para a exploração do espaço de projeto do dimensionamento de circuitos integrados analógicos este tipo de função tem um problema (LIU et al., 2009), pois soluções que não atingem as restrições, são descartadas. E assim, torna-se difícil encontrar uma solução válida em projetos fortemente restritos. Neste intuito, surge o processo de otimização mostrado na parte (b) da Figura 7, em que a função custo, além dos objetivos, apresenta o valor das especificações dada por uma função de restrição ($R(x)$). Esta inserção das especificações refere-se à distância entre a especificação alcançada e a especificação desejada. Assim, soluções que não atingem as

especificações impostas são aceitas, mas inserem um grande valor de $R(X)$. À medida que as especificações se aproximam do valor desejado $R(X)$ é reduzido, tornando-se igual a zero quando as especificações forem atingidas.

Figura 7: Definição do dimensionamento do circuito como um problema de otimização. (a) otimização de um objetivo sujeito a restrições e (b) otimização de uma função custo composta por objetivos e restrições.

$$\begin{array}{ll} \text{Min } \sum_{i=1}^N \text{objetivo}_i(X) & \text{Min } \sum_{i=1}^N \text{objetivo}_i(X) + \sum_{j=1}^M R_j(X) \\ \text{Sujeito a:} & \\ Si(X) \geq Si_{ref} & \\ Sj(X) \leq Sj_{ref} & \\ \text{(a)} & \text{(b)} \end{array}$$

A função custo de avaliação do desempenho representa um dos principais pontos no processo de otimização, pois deve aliar as especificações que deseja-se otimizar, como potência e área, e as especificações que são restrições do circuito. Desta forma, a função custo deve ser uma função multiobjetivo para que possam ser otimizadas várias especificações e, ao mesmo tempo, atingir várias especificações requeridas (PHELPS et al., 2000; SMEDT; GIELEN, 2002). A função custo ainda pode possuir técnicas de ponderação das especificações que apresentam maior dificuldade para serem atingidas (LABRAK et al., 2007).

2.2.2 Ferramentas Comerciais

As ferramentas de automação do projeto de circuitos integrados analógicos estão cada vez mais sendo inseridas no fluxo de projeto, mas ainda estão longe de serem utilizadas tal como ferramentas de automação do projeto de circuitos digitais. Uma das principais causas é a grande complexidade envolvida e também o tempo necessário de execução da ferramenta. O que tem sido feito é a utilização de ferramentas em certas etapas do projeto como dimensionamento e geração do leiaute.

De acordo com Barros, Guilherme e Horta (2010) as principais ferramentas comerciais de automação do projeto analógico são a Barcelona Design® e a Neo Linear NeoCircuit®. A ferramenta Barcelona Design® foi inicialmente desenvolvida por Hershen-son, Boyd e Lee (1998) e é baseada no dimensionamento automático de circuitos através de otimização convexa com base na simulação elétrica. Barcelona Design® possui uma interface gráfica para o projetista e é compatível com a maior parte dos simuladores elétricos comerciais. Atualmente, a ferramenta encontra-se integrada às ferramentas da empresa Synopsys. A ferramenta Neo Linear NeoCircuit® atualmente está inserida no pacote

Neonlinear® da empresa Cadence. Neocircuit® utiliza otimização baseada em *Simulated Annealing* para dimensionar o circuito e gerar o leiaute automaticamente.

Outra ferramenta que está sendo amplamente utilizada pela indústria de microeletrônica é a ferramenta WiCkeD® da empresa MunEDA. Esta ferramenta possibilita o dimensionamento e o leiaute do circuito baseados na otimização da produtividade do circuito sobre variações de parâmetros, fator este pouco explorado pelas ferramentas atuais. A WiCkeD® possui uma interface amigável e pode ser integrada às principais ferramentas de CAD para o fluxo de projeto analógico, como Cadence Virtuoso® e Synopsys Custom Designer®, além de ser compatível com grande parte dos kits de projetos fornecidos pela *foundry* (SOBE et al., 2009).

2.3 Design Centering

O processo de fabricação de circuitos integrados possui uma alta complexidade envolvida, de forma que a cada dia novos desafios tecnológicos são superados e cada vez mais chega-se aos limites da tecnologia CMOS convencional. Devido a este fato, as tecnologias sub-micrométricas possuem alta incidência de variabilidades nos processos de fabricação. Estas variações acabam afetando o desempenho dos circuitos integrados tanto digitais como analógicos. Nos circuitos digitais as principais causas são as variações no tempo de propagação e transição dos sinais digitais (BUEHLER et al., 2006). Em circuitos analógicos a variação faz com que o circuito opere em um nível de polarização diferente e sofra variações de *mismatch* (variações no casamento de dispositivos) que podem levar à perda da funcionalidade do circuito (DRENNAN; MCANDREW, 2003).

Segundo Orshansky, Nassif e Boning (2008) as variações no processo de fabricação de um circuito integrado podem ser classificadas como: *front-end*, *back-end* e impostas pelo ambiente de utilização do circuito integrado.

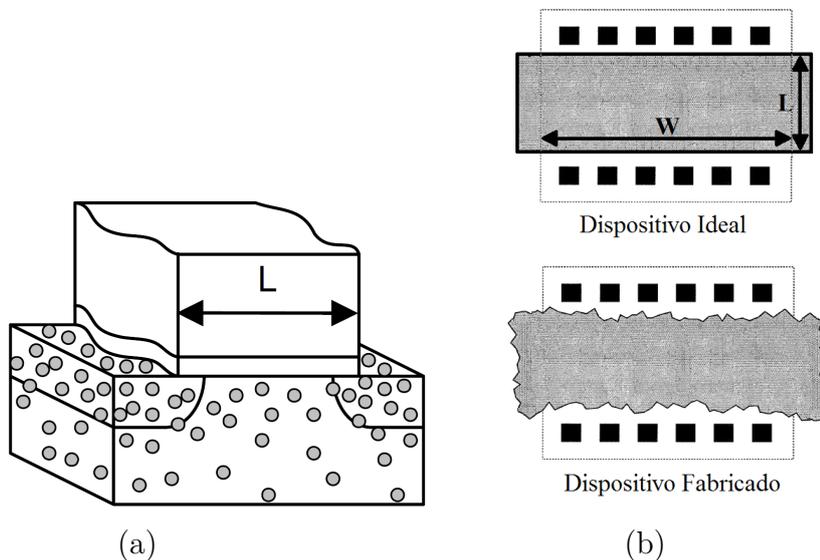
As variações *front-end* ocorrem na fabricação do dispositivo, nas etapas de implantação iônica, oxidação, construção das linhas de polisilício, entre outras. Nesta parte, podem ocorrer variações nas dimensões do canal do transistor (W e L), na tensão de limiar do dispositivo (V_{TH}), na espessura do óxido de silício (T_{ox}), entre outros. Estas variações possuem uma característica aleatória e em sua maioria estão relacionadas aos limites da tecnologia, uma vez que em alguns processos de fabricação as dimensões estão próximas à dimensão de alguns átomos.

As variações *back-end* são caracterizadas pela variação de processo causadas na metalização das vias e interconexões internas do circuito integrado. Neste tipo de variações as linhas de metal e vias de interconexão, nos diversos níveis de metais, sofrem alterações. Ao mesmo tempo, dispositivos passivos como capacitores e indutores integrados, por serem construídos em geral por linhas de metais, sofrem variações nos seus valores nominais.

As variações impostas pelo ambiente de utilização do circuito integrado referem-se às diferenças entre os valores nominais e reais de operação do circuito. Como exemplo desta classe de variação destacam-se as variações da temperatura de operação do circuito e da tensão de alimentação. Estas variações são sistemáticas e podem ser corrigidas no projeto de forma a atenuar suas causas.

A Figura 8 ilustra duas das principais variações ocorridas em um processo de fabricação. Na parte (a) ocorre a flutuação aleatória de dopantes devido à alta complexidade de controle da implantação iônica, fazendo com que os íons sejam também implantados em regiões indesejadas. Na parte (b) é mostrada a variação nas dimensões (W e L) do canal de um transistor CMOS em relação aos dispositivos ideal e real. Com esta figura pode-se perceber que para dispositivos de grande largura do canal (W), as variações em L tendem a se cancelar de forma que o seu valor torna-se próximo ao valor nominal. A variação no polisilício não provoca fortes variações em W , pois normalmente esta dimensão é dada pela região de difusão, uma vez que a camada de polisilício normalmente é maior que a região de difusão (DRENNAN; MCANDREW, 2003).

Figura 8: Principais variações no processo de fabricação de circuitos integrados em tecnologia CMOS. (a) flutuação aleatória de dopantes (YE et al., 2008) e (b) variação nas dimensões do canal de um transistor CMOS (DRENNAN; MCANDREW, 2003).



Segundo Orshansky, Nassif e Boning (2008), as variações do tipo *front-end* são as mais importantes em um projeto. Para exemplificar uma variação no processo de fabricação de um circuito integrado consideramos a tensão de limiar (V_{TH}). A tensão de limiar para transistores de canal longo e dopagem uniforme (XI et al., 2003) pode ser estimada com a seguinte equação:

$$V_{TH} = V_{TH0} + \gamma(\sqrt{2|\Phi_F| - V_{bs}} - \sqrt{2|\Phi_F|}) \quad (2.2)$$

onde V_{TH0} é a tensão de limiar para um dispositivo de canal longo e com tensão de substrato igual a zero, Φ_F é o nível de Fermi e γ é obtido com a seguinte equação:

$$\gamma = \frac{\sqrt{2q\epsilon_{Si}N_{sub}}}{C_{ox}} \quad (2.3)$$

na qual q é a carga de um elétron, ϵ_{Si} é a permissividade elétrica do silício, N_{sub} representa a densidade de elétrons implantados no substrato e C_{ox} é a capacitância de óxido. Neste contexto, verifica-se que uma variação no número de elétrons implantados na pastilha de silício (N_{sub}) representa uma grande influência no valor da tensão de limiar. Devido à complexidade da implantação iônica, em dispositivos de pequenas dimensões não é possível controlar o número exato de elétrons na região implantada, causando uma flutuação aleatória de dopantes (YE et al., 2008), como ilustrado na Figura 8. Além disso, a tensão de limiar é também dependente da capacitância de óxido, que por sua vez depende da espessura do óxido de silício sob o gate. A espessura do óxido de silício também possui variações, apresentando diferentes valores em cada região do *wafers*. É importante notar que em um mesmo *wafers* de silício pode haver regiões com características diferentes, fazendo com que circuitos de uma mesma rodada de fabricação apresentem características distintas. Em Orshansky, Nassif e Boning (2008) a seguinte equação é mostrada para o desvio padrão de variação em VTH para um processo de fabricação de circuito integrados:

$$\sigma_{V_{TH}} = 3,19 \times 10^{-8} \frac{T_{ox} N_{sub}^{0,4}}{\sqrt{L_{eff} W_{eff}}} \quad (2.4)$$

Nesta equação, verifica-se que o desvio padrão da variação de VTH está diretamente relacionado com os parâmetros espessura do óxido (T_{ox}) e N_{sub} . Além disso, verifica-se uma relação inversa à raiz quadrada da área efetiva de gate do circuito ($L_{eff}W_{eff}$). Nesta relação, verifica-se que, para dispositivos grandes, o efeito de variação é reduzido. Já para dispositivos pequenos, a variação tende a ser alta.

As variações no processo de fabricação podem ser divididas em variações locais e globais de parâmetros, também chamadas de *intra-die* e *inter-die* (MUTLU; RAHMAN, 2005) ou variações em lote (*LOT*) e dispositivo (*DEV*). As variações locais fazem com que a variação de parâmetros afete cada transistor de um mesmo circuito individualmente. Este tipo de variação é muito importante em circuitos analógicos, principalmente em circuitos como espelhos de corrente e pares diferenciais, onde o casamento (*matching*) entre os dispositivos é fundamental (SCHNEIDER; GALUP-MONTORO, 2010). As variações globais fazem com que ocorram variações nos parâmetros em relação a circuitos diferentes. Neste caso, todos os transistores de um circuito possuem a mesma variação de parâmetros. As variações globais, são importantes para análise de circuitos digitais (SAXENA et al., 2008). Com relação ao grau de variação, as variações globais apresentam maiores variações em relação às locais, porém em circuitos analógicos as pequenas variações locais provocam

maior efeito de variações nas especificações do que as grandes variações globais (DRENNAN; MCANDREW, 2003).

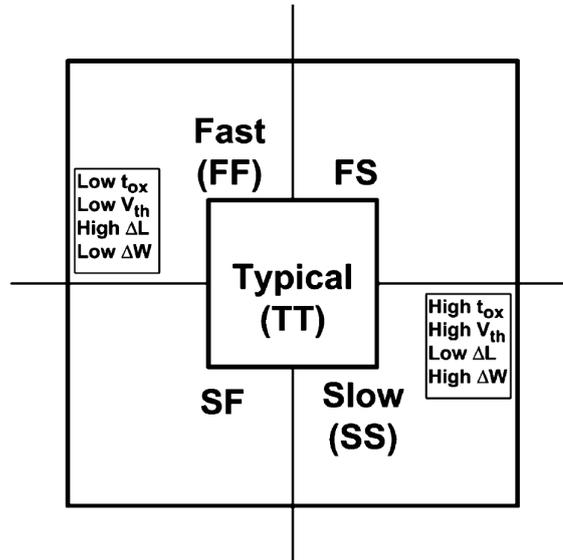
Como as variações no processo de fabricação influenciam os circuitos produzidos, deve-se analisar o comportamento do circuito em fase de projeto de forma que especificações como estas sejam levadas em conta já nas primeiras fases de projeto. Para analisar o comportamento do circuito utiliza-se um modelo elétrico e simulações elétricas do tipo SPICE. Um modelo elétrico, durante a etapa de simulação, é tido como um modelo determinístico, de forma que para um mesmo circuito sempre são encontradas as mesmas especificações. Porém, os parâmetros do modelo sofrem variações no processo de fabricação, de forma que seu valor na prática seja estocástico (não determinístico). Como os parâmetros não têm valores fixos, o modelo elétrico do dispositivo também deve ser um modelo estocástico. Para a análise do comportamento de um circuito através de simulação elétrica, deve-se conhecer o comportamento da variação dos parâmetros do modelo.

Para a análise das características do circuito com relação à variação de parâmetros podem ser utilizadas análises estatísticas. Estas análises são geridas pela observação (caracterização) do processo de fabricação. Com a caracterização do processo, especificações como os valores mínimos, máximos, média e desvio padrão de variação de cada parâmetro podem ser analisados.

Segundo Graeb (2007), com a simulação elétrica, dois tipos de análises à variações de parâmetros podem ser utilizadas: a análise de pior caso (*worst case*) e análise de Monte Carlo. Na primeira forma, o circuito é simulado com os piores casos de variações. De acordo com Mutlu e Rahman (2005), a análise de pior caso mais voltada a circuitos digitais está relacionada à velocidade do circuito. Nesta configuração, os parâmetros do circuito são alterados dentro a sua faixa de variação para valores que causam as maiores variações de velocidade. Para isso, os dispositivos do tipo NMOS e PMOS podem assumir três classes: rápida, lenta ou típica. Com estas combinações, cinco possíveis formas podem ser analisadas, como mostra a Figura 9, na qual *Fast* (FF) ou *Slow* (SS) significa que ambos os tipos de transistores NMOS e PMOS estão em uma mesma configuração, rápida ou lenta. A configuração rápida é chamada de pior caso em termos de potência e a configuração lenta é chamada de pior caso de velocidade. Já FS ou SF indica configurações diferentes. Caso um tipo de transistor esteja em configuração lenta o outro estará em configuração rápida. A configuração FS é chamada de condição um de pior caso e a configuração SF é chamada de condição zero de pior caso. A análise de pior caso fornece uma forma fácil de verificar a funcionalidade do circuito com relação às variações impostas ao projeto, uma vez que é necessário somente uma simulação para cada caso. Se o circuito estiver dentro das características desejadas no pior caso, em outros casos também irá manter as suas funcionalidades. A desvantagem da análise de pior caso é que esta tende a superestimar a variação de parâmetro e, assim, muitas vezes acaba dificultando

o projeto. Além disso, o pior caso tem uma pequena probabilidade de acontecer, uma vez que inúmeros outros casos melhores existem. Os modelos de pior caso normalmente utilizam uma faixa de 3 ou 6 desvios padrão de cada parâmetro.

Figura 9: Análise de pior caso com relação à velocidade dos transistores CMOS.



Neste conceito, surge a outra forma de análise mais abrangente, chamada Monte Carlo, em que o circuito é verificado através de uma distribuição estatística. Nesta análise, números aleatórios dentro da faixa de variação conhecida são gerados para cada parâmetro através de uma função de densidade de probabilidade (PDF).

A função de densidade de probabilidade pode ser vista como a forma em que números aleatórios são gerados ou amostrados. A função de distribuição de probabilidade mais utilizada é a distribuição normal (ou gaussiana). Esta distribuição obedece a seguinte equação:

$$PDF_{NORMAL} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2.5)$$

em que σ representa o desvio padrão e μ a média de um parâmetro x . Esta equação, para $\mu = 0$, fornece a representação gráfica mostrada na Figura 10. De acordo com esta figura, verifica-se que a probabilidade de serem amostrados valores próximos a média é alta e já valores afastados da média apresentam uma pequena probabilidade de existência. Esta análise está de acordo com a maior parte dos parâmetros físicos, uma vez que a maior parte dos parâmetros estão próximos do valor nominal e somente uma minoria estão longe (KHOUAS; DERIEUX, 2000).

Em uma distribuição gaussiana, de acordo com o valor da média e do desvio padrão, a probabilidade de existência de uma solução pode ser analisada. A Tabela 1 sintetiza os valores de probabilidade acumulada em cada um dos intervalos de desvio padrão. Nesta

Figura 10: Curva de uma função de densidade de probabilidade normal.

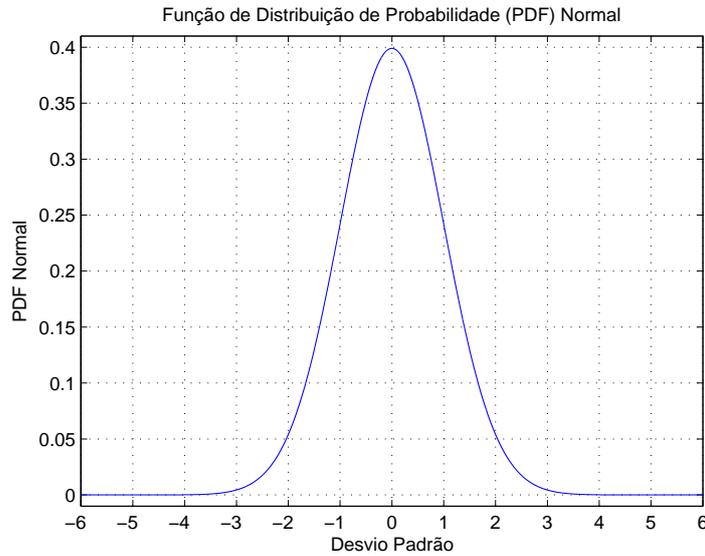


Tabela 1: Faixa de desvio padrão e probabilidade de uma distribuição normal

Faixa de Desvio Padrão	Probabilidade de distribuição
$-\sigma$ a σ	68,268949213 %
-2σ a 2σ	95,44997361 %
-3σ a 3σ	99,73002039 %
-4σ a 4σ	99,99366575 %
-5σ a 5σ	99,99994266 %
-6σ a 6σ	99,99999980 %

tabela, é possível verificar que existe uma probabilidade de 68,27% de o parâmetro estar situado entre $-\sigma$ e σ em relação à média ($\sigma = 0$). Além disso, verifica-se que a chance de um parâmetro estar situado no intervalo de -3σ a 3σ é próxima de 100%.

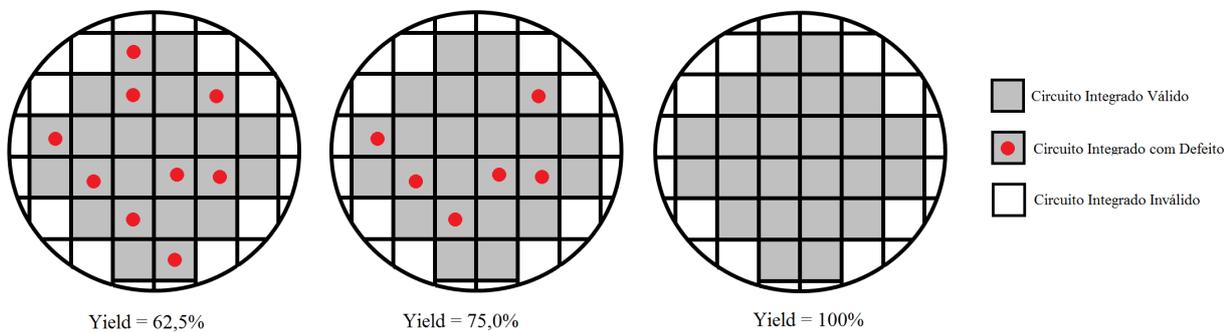
Além da função de densidade de probabilidade gaussiana, existem inúmeras outras funções que geram valores de base aleatória para a simulação Monte Carlo, dentre as funções destacam-se: variação uniforme e *Latin Hypercube Sampling* (LIU; MESSAOUDI; GIELEN, 2012). A principal diferença entre as funções de densidade de probabilidade refere-se à forma em que os valores aleatórios para cada parâmetro são amostrados. Devido à diferença em que os números são gerados, são necessárias maiores ou menores números de rodadas de simulações para haver uma análise com precisão. De acordo com Liu, Messaoudi e Gielen (2012), a simulação elétrica de variação de parâmetros baseado em Monte Carlo com PDF normal tende a representar melhor as variações físicas, porém às custas da necessidade de um número maior de simulações, o que tende a afetar o tempo de análise. De acordo com Graeb (2007), a precisão da análise Monte Carlo é proporcional à raiz quadrada do número de amostras de uma simulação, enquanto que a complexidade da análise cresce linearmente com o número de amostras.

Baseando-se na densidade da probabilidade, a produtividade do circuito pode ser analisada. A produtividade (ou *yield*) refere-se à percentagem de circuitos integrados que mantêm as especificações desejadas após a fabricação, de acordo com a seguinte equação:

$$Yield = \frac{N^{\circ} \text{ de circuitos integrados válidos}}{N^{\circ} \text{ total de circuitos integrados}} \cdot 100\% \quad (2.6)$$

Como um exemplo, a Figura 11 mostra o *yield* para três diferentes rodadas de fabricação. Verifica-se que, dependendo do número de circuitos que apresentam defeitos ou possuem especificações fora dos valores requeridos, o *yield* é alterado.

Figura 11: *Yield* em um *wafer* de silício.



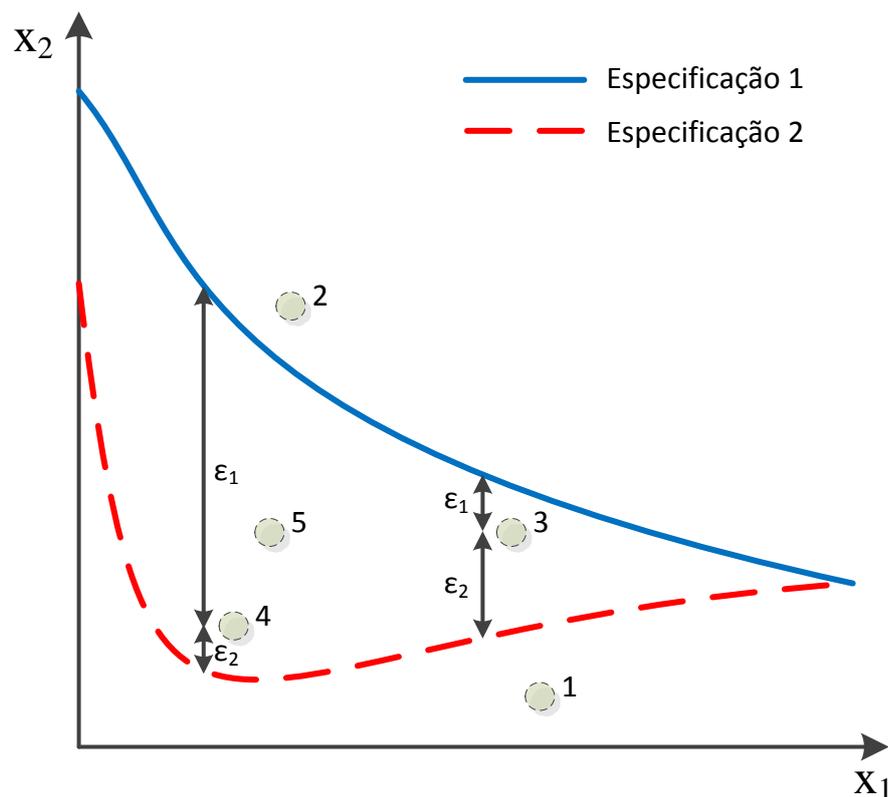
O *yield* de um circuito pode ser facilmente analisado com a simulação Monte Carlo, pois inúmeras simulações são geradas e a percentagem de simulações cujos resultados atendem às especificações impostas pode ser verificada. Na análise de pior caso não é possível analisar um valor de *yield*, pois com o resultado da simulação pode-se somente verificar se a simulação atinge ou não as especificações impostas ao circuito.

Baseando-se na estimativa de variação de parâmetros através da simulação Monte Carlo, o valor da produtividade pode ser otimizado. Esta otimização recebe o nome de *design centering*. Desta forma, junto ao fluxo de dimensionamento, é possível inserir formas em que o espaço de projeto possa ser explorado com o intuito de encontrar uma região em que a produtividade seja máxima. Neste caso os resultados obtidos no dimensionamento do circuito são válidos mesmo na ocorrência de variações nos parâmetros do circuito.

De acordo com Deyati e Mandal (2011) e Li et al. (2005) o *design centering* pode ser feito também sem a necessidade da análise do *yield* durante o processo de otimização. Para isso, pode-se utilizar técnicas geométricas. A exploração do projeto pode ser feita de forma a maximizar a distância (ϵ) entre uma especificação requerida e o valor encontrado. Para exemplificar, a Figura 12 mostra um espaço de projeto de duas variáveis, onde é requerido que a solução esteja entre as curvas de especificação 1 e especificação 2. As soluções 1 e 2 estão fora dos valores requeridos e por isso não são soluções válidas. As soluções 3 e 4 apresentam uma maior proximidade à especificação 1 ou à especificação

2. Esta maior proximidade faz com que, na ocorrência de variações de parâmetros, a especificação de maior proximidade não seja satisfeita, caindo fora do espaço de projeto. Já a solução 5 está posicionada em uma região central, onde a distância entre as curvas de especificações 1 e 2 é a mesma, e está mais distantes das bordas do espaço de projeto. Assim, devido à maior distância, a solução tem uma maior probabilidade de se manter válida na ocorrência de variações de parâmetros. Esta técnica, embora simples e rápida, não tem a maximização do *yield* garantida, uma vez que existem especificações que sofrem menos ou mais o efeito das variações de processo. Em Deyati e Mandal (2011), ao final do processo de *design centering* de forma geométrica, a análise de Monte Carlo é executada para o cálculo efetivo do *yield*.

Figura 12: Design centering através da maximização da distância entre as especificações requeridas e a encontrada.



Para a simulação SPICE das variações de processos (pior caso ou Monte Carlo), em geral a *foundry* fornece os dados de variações de parâmetros em seu processo de progresso. Em geral, dois modelos são fornecidos pela *foundry* o modelo para simulação

de Monte Carlo e modelo de pior caso. Quando estes dados não estão disponíveis, devem ser fabricados vários transistores de testes para ser feita uma caracterização da tecnologia.

2.4 Conclusão

Neste capítulo, foi apresentada uma revisão bibliográfica sobre o projeto automático de circuitos integrados analógicos.

Nesta revisão foi constatado que muitas vezes a qualidade do projeto manual (baseado no conhecimento do projetista) depende diretamente da experiência do projetista.

As ferramentas de automação do projeto, em geral, podem ser classificadas de acordo com a forma de obtenção de uma topologia de circuito. Uma topologia de circuito é projetada através de um processo de dimensionamento que pode ser baseado no conhecimento por parte do projetista (metodologia de projeto) ou através de um processo de otimização. Com um processo de otimização é possível explorar o espaço de soluções. Neste tipo de dimensionamento, quando é utilizada simulação elétrica, não é necessário grande conhecimento do circuito a ser projetado pelo projetista. Após o dimensionamento, o leiaute do circuito pode ser gerado.

Neste capítulo foi também apresentada a otimização da produtividade com a técnica de *design centering*. Nesta técnica, foi verificado que uma forma de verificar o efeito da variação de parâmetros é a verificação da produtividade através de simulação elétrica, como a análise de Monte Carlo.

3 Automação do Projeto de um Amplificador Diferencial

Este capítulo analisa algumas estratégias de dimensionamento automático aplicadas ao projeto de um circuito analógico integrado básico, o amplificador diferencial. Para isso, um circuito simplificado do amplificador diferencial, contendo somente duas variáveis livres de projeto, é utilizado de forma que o espaço de projeto seja analisado em termos de otimização. Para a otimização, as técnicas de *Simulated Annealing* e Algoritmos Genéticos são apresentadas e analisadas.

Ao mesmo tempo, este capítulo propõem algumas técnicas e formas de otimização da produtividade (*design centering*) do circuito.

As técnicas apresentadas neste capítulo são utilizadas na implementação da ferramenta UCAF, mostrada no Capítulo 4.

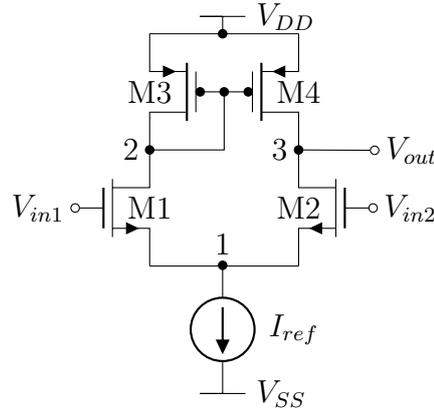
3.1 Amplificador Diferencial

O amplificador diferencial é um bloco analógico básico utilizado, de maneira geral, como estágio de entrada dos circuitos amplificadores operacionais. Apesar de sua simplicidade, este circuito é muito utilizado como o primeiro estágio de amplificação de tensão de muitos dispositivos eletrônicos e tem se tornado uma das principais escolhas nos circuitos analógicos de alta performance e circuitos mistos (RAZAVI, 2000).

De maneira ideal, o circuito amplifica a diferença de tensão entre as suas entradas, sem amplificar a tensão de modo comum. Uma implementação do amplificador diferencial com transistores CMOS e carga ativa é mostrada na Figura 13. Este circuito é composto por um par diferencial, formado por dois transistores de entrada (M1 e M2), um espelho de corrente ativo (M3 e M4) e uma fonte de corrente ideal (I_{ref}). A tensão de saída (V_{out}) depende da diferença de tensão entre suas entradas V_{in1} e V_{in2} . Para pequenas diferenças entre V_{in1} e V_{in2} , ambos transistores M2 e M4 estão saturados, provendo alto ganho. De outra forma, se $|V_{in1} - V_{in2}|$ é grande o bastante, M1 ou M2 estará em estado de corte, de forma que a tensão de saída possa estar próxima da tensão de alimentação V_{SS} ou V_{DD} .

A tensão de saída do amplificador diferencial pode ser expressa em termos dos

Figura 13: Esquemático de um amplificador diferencial CMOS com carga ativa.



modos comum e diferencial, como

$$V_{out} = A_{VD}(V_{in1} - V_{in2}) \pm A_{VC} \left(\frac{V_{in1} + V_{in2}}{2} \right) \quad (3.1)$$

Nesta equação, A_{VD} é o ganho de tensão em modo diferencial e A_{VC} é o ganho de tensão em modo comum.

Um amplificador operacional ideal tem um valor infinito de A_{VD} e um valor zero de A_{VC} . As implementações práticas sempre tentam aproximar estes valores, porém a implementação física do circuito insere algumas não-idealidades que limitam os valores de A_{VD} e A_{VC} . Outra característica importante do amplificador diferencial é a faixa de entrada em modo comum ($ICMR$). Para estimar o valor de $ICMR$, pode-se assumir $V_{in1} = V_{in2}$ e variar a tensão de modo comum (componente DC de V_{in1} e V_{in2}) até que um dos transistores não se mantenha saturado (ALLEN; HOLBERG, 2002). O maior valor da tensão de entrada em modo comum ($ICMR^+$) é dado pela seguinte equação:

$$ICMR^+ = V_{DD} - V_{SG3} + V_{TN1} \quad (3.2)$$

em que V_{SG3} é a tensão entre os terminais de source e gate do transistor $M3$ e V_{TN1} é a tensão de limiar de $M1$. O menor valor de tensão de entrada ($ICMR^-$) no terminal de gate de $M1$ (ou $M2$) é dado pela equação seguinte:

$$ICMR^- = V_{SS} + V_1 + V_{GS2} \quad (3.3)$$

A tensão no nó 1 (V_1) é determinada pela implementação física da fonte de corrente I_{ref} , a qual, em geral, é formada por um único transistor cuja corrente de dreno é controlada por uma tensão de gate. V_{GS2} é a tensão entre os terminais de gate e source do transistor $M2$.

As propriedades de pequenos sinais do amplificador diferencial podem ser analisadas com o modelo simplificado mostrado na Figura 14, que ignora os efeitos de corpo

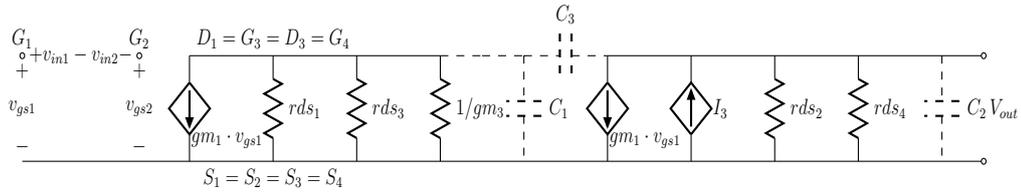
do substrato (*body effects*). Nesta figura, gm é a transcondutância de gate dada pela derivada parcial da corrente de dreno em relação à tensão entre gate e source.

$$gm = \frac{\partial I_D}{\partial V_{GS}} \quad (3.4)$$

A resistência série rds é dada pelo inverso da condutância de saída gds e pode ser estimada na análise de pequenos sinais como a derivada parcial da corrente de dreno em relação à tensão entre dreno e source.

$$\frac{1}{rds} = gds = \frac{\partial I_D}{\partial V_{DS}} \quad (3.5)$$

Figura 14: Modelo de pequenos sinais simplificado para o circuito amplificador diferencial CMOS.



O ganho de tensão em pequenos sinais (A_{vo}), ou seja, a relação entre a tensão de saída V_{out} e a entrada diferencial $V_{in1} - V_{in2}$, pode ser estimada em baixa frequência como

$$A_{vo} = \frac{gm_1}{gds_2 + gds_4} \quad (3.6)$$

Para altas frequências, o ganho de tensão é afetado pelas inúmeras capacitâncias parasitas em cada um dos nós do circuito, modeladas como C_1 , C_2 e C_3 , as quais são calculadas da seguinte forma:

$$C_1 = C_{gd1} + C_{bd1} + C_{bd3} + C_{gs3} + C_{gs4} \quad (3.7)$$

$$C_2 = C_{bd2} + C_{bd4} + C_{gd2} + C_L \quad (3.8)$$

$$C_3 = C_{gd4} \quad (3.9)$$

Considerando C_3 aproximadamente igual a zero, a função de transferência pode ser escrita como

$$V_{out}(s) \cong \frac{gm_1}{gds_2 + gds_4} \left[\left(\frac{gm_3}{gm_3 + sC_1} \right) V_{gs1}(s) - V_{gs2}(s) \right] \frac{\omega_2}{s + \omega_2} \quad (3.10)$$

em que ω_2 é dado pela seguinte equação:

$$\omega_2 = \frac{gds_2 + gds_4}{C_2} \quad (3.11)$$

o polo ω_2 determina a frequência de corte do amplificador, também chamada de ω_{-3dB} .

Assumindo que

$$\frac{gm_3}{C_1} \gg \frac{gds_2 + gds_4}{C_2} \quad (3.12)$$

então, a resposta em frequência do amplificador diferencial se reduz a

$$\frac{V_{out}(s)}{V_{in1}(s) - V_{in2}(s)} \cong \left(\frac{gm_1}{gds_2 + gds_4} \right) \left(\frac{\omega_2}{s + \omega_2} \right) \quad (3.13)$$

esta análise de primeira ordem possui um único polo na saída, dado por $-(gds_2 + gds_4)/C_2$. Alguns zeros ocorrem devido a C_{gd1} , C_{gd2} e C_{gd4} , mas estes podem ser ignorados nesta análise.

O produto ganho-largura de faixa (GBW), que é igual à frequência de ganho unitário, pode ser expresso como

$$GBW = A_{vo} \cdot \omega_{-3dB} \quad (3.14)$$

O *slew rate* (SR), que caracteriza a velocidade de resposta do amplificador, possui seu valor dependente de I_{ref} e da capacitâncias associadas ao nó de saída e o terminal comum (GND) e é dado pela seguinte equação:

$$SR = \frac{I_{ref}}{C} \quad (3.15)$$

em que C é a capacitância total conectada no nó de saída (nesta análise, aproximada por C_2).

Outras especificações importantes do comportamento elétrico do amplificador diferencial incluem dissipação de potência e área de gate, calculadas com as Equações 3.16 e 3.17, respectivamente.

$$P_{diss} = I_{ref} \cdot (V_{DD} - V_{SS}) \quad (3.16)$$

$$Area = \sum_i^N W_i \cdot L_i \quad (3.17)$$

em que N refere-se ao número de transistores do circuito, neste caso igual a 4.

Todo projeto analógico possui uma tecnologia de fabricação e um tipo de dispositivo associado. Em tecnologia CMOS, o projetista deve dimensionar os transistores do circuito, pois W (largura do canal) e L (comprimento do canal) são os únicos parâmetros que podem ser alterados na fabricação de um dispositivo CMOS.

3.2 Modelagem do Amplificador Diferencial para o Dimensionamento Automático

A modelagem do amplificador diferencial da Figura 13 para o dimensionamento automático é dada na sequência. Utilizando um processo baseado em otimização, as especificações do circuito são calculadas através de simulações elétricas SPICE. A escolha por utilizar a simulação SPICE, deve-se ao fato de não serem necessárias equações simplificadas que modelam um dado circuito, uma vez que equações simplificadas estão condicionadas a uma dada região de operação e, devido à simplificação, apresentam resultados diferentes dos resultados reais. Com isso, a simulação elétrica é uma alternativa flexível, pois os modelos elétricos tendem a modelar o circuito em todas as regiões de operação e é possível analisar qualquer circuito. Como exemplo de modelagem, para o dimensionamento, vamos considerar o projeto de um amplificador diferencial que deve ser otimizado em termos da área de gate do circuito e satisfazer algumas restrições de projeto. Além da área, outras especificações poderiam ser também otimizadas, formando assim um projeto multiobjetivo. A Tabela 2 mostra os objetivos e restrições para este problema.

Tabela 2: Especificações de projeto e restrições para o circuito amplificador diferencial da Figura 13

Especificação	Valor Requerido
Área de gate	<i>Minimizar</i>
Ganho em Baixas Frequências (A_v)	$\geq 40,00 \text{ dB}$
Margem de Fase (PM)	$\geq 70,00^\circ$
Produto Ganho-Largura de Faixa (GBW)	$\geq 1,00 \text{ MHz}$
Máximo Valor de Entrada em Modo Comum ($ICMR^+$)	$\geq 0,40 \text{ V}$

A função custo $f_c(X)$ é formulada como a soma das especificações e restrições de projetos em termos do vetor de variáveis livres de projeto X :

$$f_c(X) = \frac{Area(X)}{Area_{ref}} + R(X) \quad (3.18)$$

As especificações são calculadas para um dado X e são normalizadas por um valor de referência. Neste exemplo, $Area_{ref} = 1\mu m^2$. A ponderação de cada especificação pode ser implementada como pesos individuais que indicam a importância relativa do parâmetro. Neste exemplo, foram adotados todos os pesos iguais à unidade. $R(X)$ é a função de restrição em função de X , calculada como:

$$R(X) = R_{min}(A_v(X), A_{vref}) + R_{min}(PM(X), PM_{ref}) + R_{min}(GBW(X), GBW_{ref}) \\ + R_{min}(ICMR^+(X), ICMR^+_{ref}) \quad (3.19)$$

em que, $R_{min}(S(X), S_{ref})$ é uma função de restrição de mínimo em termos de uma especificação $S(X)$ e de um valor de referência S_{ref} . Pode haver ainda especificações que devem ser menores do que um valor de referência. Neste caso existe uma função $R_{max}(S(X), S_{ref})$. Ambas funções de restrições inserem uma penalidade na função custo $f_c(X)$, caso a especificação esteja fora da faixa de valores desejados. Caso contrário, a função retorna o valor zero. As equações seguintes mostram como as funções de restrição estão implementadas:

$$R_{max}(S(X), S_{ref}) = \begin{cases} 0 & \text{se } S(X) \leq S_{ref} \\ \frac{S_{ref}-S(X)}{S_{ref}} & \text{se } S(X) > S_{ref} \end{cases} \quad (3.20)$$

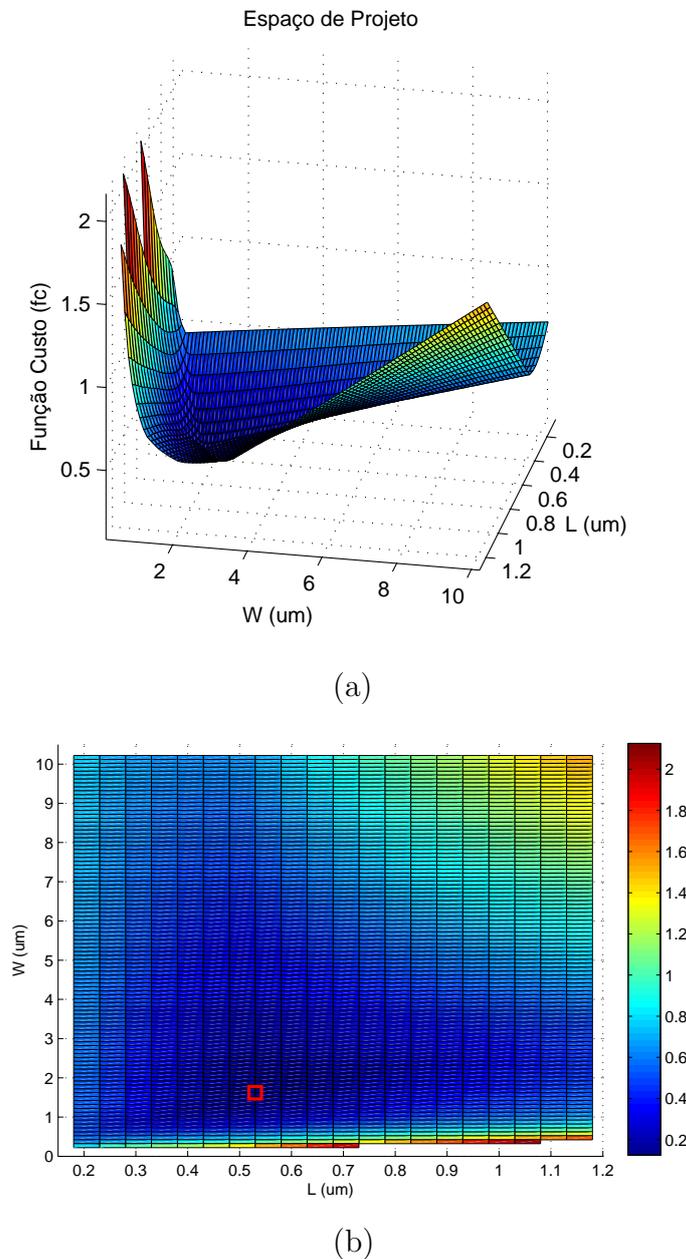
$$R_{min}(S(X), S_{ref}) = \begin{cases} 0 & \text{se } S(X) \geq S_{ref} \\ \frac{S(X)-S_{ref}}{S_{ref}} & \text{se } S(X) < S_{ref} \end{cases} \quad (3.21)$$

Neste exemplo, foram utilizados como restrições de referência os valores mostrados na Tabela 2. Os valores de $Avo(X)$, $PM(X)$, $GBW(X)$ e $ICMR^+(X)$ são obtidos através de simulação elétrica SPICE.

Com o intuito de simplificar a análise, considera-se que todos os transistores tenham o mesmo tamanho. Esta consideração não é um procedimento prático, uma vez que neste circuito $M1$ deve ser igual a $M2$, mas não necessariamente igual a $M3$ e $M4$. Porém, esta simplificação fornece uma visualização do problema em duas dimensões, de forma que possam ser exemplificados os *trade-offs* e o valor ótimo de busca, provendo uma noção intuitiva do problema. Portanto, neste projeto foram consideradas duas variáveis livres de projeto: $L = L_1 = L_2 = L_3 = L_4$ e $W = W_1 = W_2 = W_3 = W_4$. O vetor de variáveis livres de projeto é $X = [W \ L]$.

O espaço de projeto para a Equação 3.18 foi mapeado através de simulação elétrica variando W de $0,22\mu m$ a $10\mu m$ e L de $0,18\mu m$ a $1\mu m$ com passos de $0,01\mu m$. Para isso, foram utilizados os parâmetros da tecnologia XFAB de $0,18\mu m$ CMOS com tensão de alimentação ($V_{DD} - V_{SS}$) igual a $1,8V$. A Figura 15 mostra o espaço de projeto traçado como função de W e L . Nesta figura, é possível notar a alta não-linearidade da função gerada e também a existência de um vale onde está localizado o valor mínimo. A solução ótima para o problema de dimensionamento, ou seja, o valor mínimo da função custo, está localizado em $W = 1,62\mu m$ e $L = 0,55\mu m$, com um valor de $0,119580$. Neste ponto, o circuito amplificador diferencial apresenta as especificações mostradas na Tabela 3. Verifica-se que a solução ótima atende a todas as especificações requeridas ao projeto e apresenta um valor mínimo de área igual a $3,564\mu m^2$.

Figura 15: Espaço de projeto para o amplificador diferencial de duas variáveis. O valor mínimo é igual a 0,1195802 e é obtido com $W = 1,62\mu m$ and $L = 0,55\mu m$. (a) espaço tridimensional e (b) espaço bidimensional com especificação da solução ótima.



3.3 Exploração do Espaço de Projeto de um Amplificador Diferencial através de Otimização

Na seção anterior, o espaço de projeto do amplificador diferencial simplificado foi obtido através de simulação elétrica e o cálculo da função custo foi feito através da Equação 3.18.

Nesta seção, o objetivo será a análise da exploração do espaço de projeto por heurísticas de otimização. Como o espaço de projeto do circuito analisado tem a sua

Tabela 3: Especificações nominais do circuito amplificador diferencial simplificado para a solução ótima no ponto $W = 1,62\mu m$ e $L = 0,55\mu m$

Especificação	Valor Requerido	Solução Ótima
$Area$	$Minimizar$	$3,564\mu m^2$
A_v	$\geq 40,00 dB$	$49,97 dB$
PM	$\geq 70,00^\circ$	$90,07^\circ$
GBW	$\geq 1,00 MHz$	$1,00 MHz$
$ICMR^+$	$\geq 0,40 V$	$0,59 V$

solução ótima conhecida, esta será utilizada como base para análise dos resultados obtidos com as heurísticas.

Como heurísticas de otimização, esta seção aborda as técnicas de otimização baseada em Algoritmos Genéticos e *Simulated Annealing*.

3.3.1 Otimização com Simulated Annealing (SA)

A heurística *Simulated Annealing* (SA) é uma meta-heurística de otimização para sistemas não-lineares inspirada na analogia com um sistema termodinâmico ao simular o resfriamento de um conjunto de átomos aquecidos até o ponto de mínima energia, fenômeno este conhecido como recozimento (*Annealing*).

O fluxograma da heurística SA aplicado ao dimensionamento de circuitos integrados analógicos é mostrado na Figura 16. Neste fluxograma, o núcleo de otimização com *Simulated Annealing* recebe como entrada os parâmetros de configuração da heurística, as especificações de projeto (neste caso os dados da Tabela 2) e os dados da tecnologia de fabricação (neste caso os parâmetros de simulação da tecnologia XFAB 0,18 μm).

O primeiro passo do fluxograma refere-se a gerar uma solução inicial - neste caso aleatória - e fazer a avaliação da solução gerada. A avaliação é feita através do cálculo da função custo (Equação 3.18). Para o cálculo da função custo é necessário estimar as especificações do circuito através de simulação elétrica. Após a solução inicial, o parâmetro de temperatura é inicializado em um valor informado pelo usuário.

O próximo passo refere-se à geração de uma nova solução X_N . Para isso é utilizada uma função de geração de soluções. Esta função gera uma nova solução baseando-se na solução atual e no parâmetro de temperatura. A Tabela 4 apresenta duas possibilidades de construção desta função: *Fast* e *Boltzmann* (disponíveis na Heurística de SA do *Optimization Toolbox* do Matlab (MATHWORKS, 2012b)). Nestas funções, X_C refere-se ao vetor da solução atual, $T(i)$ é o valor do parâmetro de temperatura da iteração i e X_R é um vetor de valores aleatórios normalizados na faixa de -1 a +1. A diferença entre as funções *Fast* e *Boltzmann* está relacionada ao grau de variação possível no valor gerado

Figura 16: Fluxograma de dimensionamento automático utilizando SA.

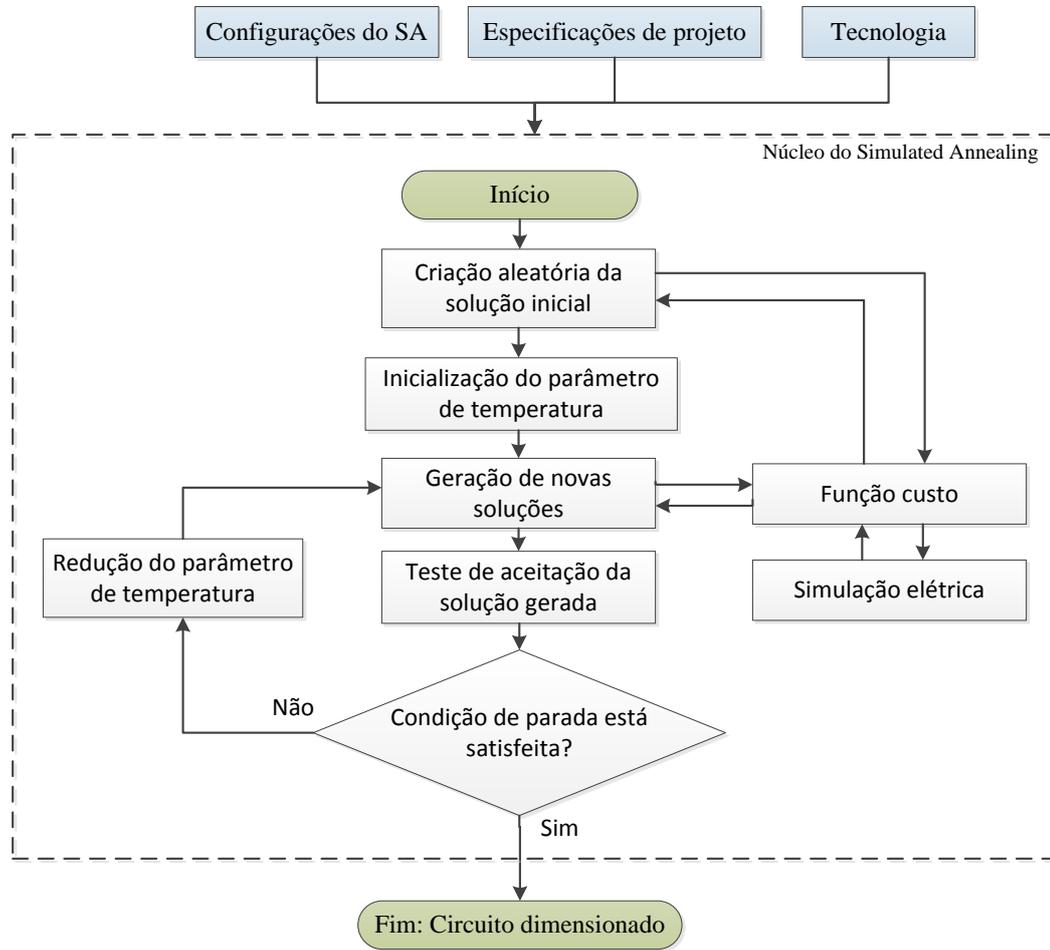


Tabela 4: Funções de geração de novas soluções do SA

Nome	Função
Fast (G_{FAST})	$X_N = X_C + T(i).X_R$
Boltzmann (G_{BOLTZ})	$X_N = X_C + \sqrt{T(i)}.X_R$

em relação ao valor atual. A primeira é proporcional a $T(i)$ e a outra é proporcional a $\sqrt{T(i)}$. Após a geração da nova solução, esta deve ser avaliada com a função custo.

Após a geração de uma nova solução e sua avaliação, o teste de aceitação é executado. Este teste refere-se primeiramente ao valor da função custo. Se este valor for melhor (menor, no caso de um problema de minimização) que o valor da solução atual, a solução gerada torna-se a solução atual do problema. Caso a solução gerada apresente um valor pior, em relação ao valor atual, uma probabilidade de distribuição de aceitação (H) é calculada e comparada com um valor aleatório. Se o valor de probabilidade calculado for maior do que o número aleatório, a solução gerada torna-se a solução atual do problema. Caso contrário, a solução gerada é descartada e o procedimento continua. Como função

de distribuição de probabilidade pode-se utilizar a distribuição de Boltzmann dada pela seguinte equação (FLOUDAS; PARDALOS, 2008):

$$H = \frac{1}{1 + e^{\frac{f_c(X_N) - f_c(X_C)}{T(i)}}} \quad (3.22)$$

em que f_c é a função custo da Equação 3.18, X_N e X_C representam o vetor de solução gerada e solução atual, respectivamente, e $T(i)$ é o valor da temperatura na iteração i . Nesta equação, pode-se perceber que H reduz à medida que $T(i)$ reduz. Como $T(i)$ é reduzida ao longo das iterações, a probabilidade de aceitação de soluções ruins se reduz ao longo das iterações. Desta forma, inicialmente a heurística SA tende a ter maior chance de escapar de soluções mínimas locais e ter uma variabilidade de geração de soluções maior (MICHALEWICZ; FOGEL, 2000). Além disso, se $f_c(X_N) - f_c(X_C)$ for grande, a probabilidade de aceitação da nova solução é pequena.

No próximo passo do fluxograma, o critério de parada é testado e, se satisfeito, a heurística finaliza a busca e retorna o melhor valor encontrado durante a exploração do espaço de projeto. São exemplos de critérios de parada: um número máximo de soluções geradas, valor mínimo tolerável de variação na função custo e valor mínimo de temperatura. Caso o critério de parada não seja satisfeito, o parâmetro de temperatura $T(i)$ é reduzido por uma função de decaimento de temperatura. As funções de decaimento de temperatura mostradas na Tabela 5 estão disponíveis no Matlab *Optimization Toolbox*. Estas funções têm as evoluções traçadas na Figura 17, na qual pode-se perceber que a função que possui o decaimento mais rápido é a *Fast* e a que tem o decaimento mais lento é a função *Boltzmann*.

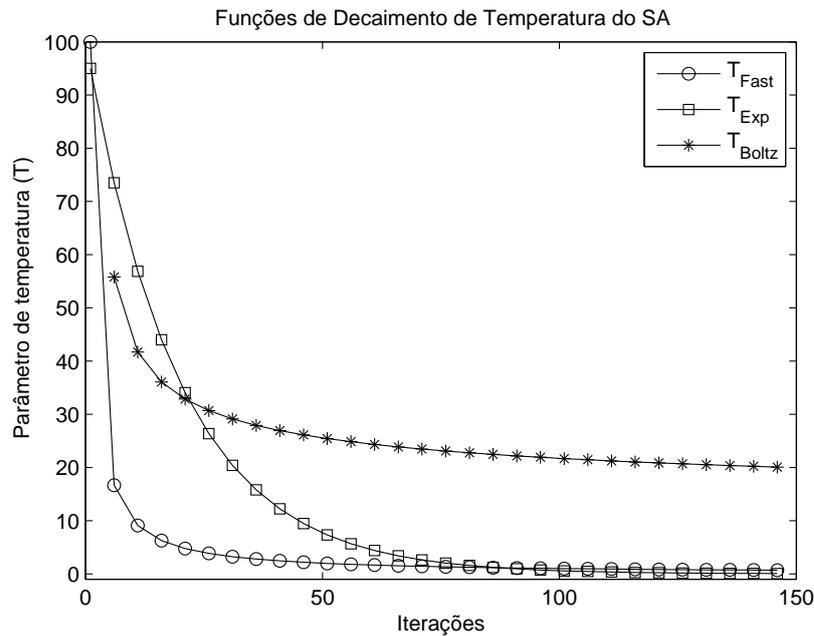
Após o decaimento da temperatura, os procedimentos do fluxograma são repetidos de maneira que novas soluções sejam geradas e avaliadas. O processo se repete até que a condição de parada seja satisfeita. No fluxograma da Figura 16 pode haver algumas etapas extras de otimização com *Simulated Annealing*. A primeira refere-se ao *Reannealing*, no qual pode-se configurar a heurística para que, se em um certo número de iterações o critério de parada não tiver sido satisfeito, o parâmetro de temperatura seja reinicializado. Com esta reinicialização, a probabilidade de aceitação de novas soluções é maior, aumentando assim a chance de escapar de mínimos locais. Porém, o algoritmo pode sofrer com variabilidade caso esteja já próximo à solução global.

Outra alternativa que pode ser inserida ao processo de otimização com SA é a utilização de uma busca híbrida. A busca híbrida utiliza otimização local e global simultaneamente. Com isso, é possível encontrar a solução mínima local mais próxima da solução atual. Este procedimento é importante ao término do processo de otimização com SA, pois a solução tende a estar próxima de um valor mínimo, mas dificilmente exatamente no ponto mínimo (LIU et al., 2009).

Tabela 5: Funções de decaimento de temperatura do SA

Nome	Função
Fast (T_{FAST})	$\frac{T_0}{i}$
Exponencial (T_{EXP})	$T_0 \cdot 0,95^i$
Boltzmann (T_{BOLTZ})	$\frac{T_0}{\ln(i)}$

Figura 17: Evolução das funções de decaimento da temperatura do SA.



Para analisar o efeito das configurações do SA quando aplicado ao dimensionamento automático de circuitos integrados analógicos, serão apresentadas a seguir inúmeras execuções da heurística no problema de dimensionamento mostrado na Seção 3.2. Para isso, foram executadas 1000 rodadas de otimização com diferentes configurações de funções de geração de novas soluções, funções de decaimento de temperatura, utilização de *Reannealing* e otimização híbrida. Para cada execução, a solução inicial foi gerada aleatoriamente e, ao final das execuções, uma análise estatística foi realizada para a análise e comparação dos resultados. Para as execuções foi utilizado o software Matlab® e o simulador elétrico Synopsys HSPICE®. Nesta exploração do projeto foi utilizado o mapeamento discreto do espaço de projeto utilizado para traçar o gráfico da Figura 15. Esta escolha deve-se ao fato de eliminar a simulação elétrica a cada iteração, de forma a reduzir o tempo de dimensionamento e tornar possível a análise de 1000 execuções para cada configuração.

A Tabela 6 mostra a média do valor de função custo encontrado após 1000 execuções do SA utilizando a função de geração G_{FAST} e as três funções de decaimento de

temperatura. Nesta tabela é possível verificar que a solução que possui uma média mais próxima da solução ótima foi obtida com a função T_{BOLTZ} , que possui \bar{f}_c igual a 0,12024 sendo que a solução ótima é 0,119580. Isto deve-se ao fato de que a maior parte das 1000 execuções com T_{BOLTZ} são próximas ou iguais à solução ótima. As funções de decaimento de temperatura T_{EXP} e T_{FAST} , apresentaram os piores resultados. Os resultados com T_{BOLTZ} , embora bons, tiveram um maior número de iterações executadas, o que resultou em um tempo de execução maior.

Tabela 6: Valores médios da otimização global do circuito amplificador diferencial para diferentes funções de decaimento de temperatura usando G_{FAST}

Função	\bar{f}_c	\bar{W} (μm)	\bar{L} (μm)	Tempo de execução (s)	Iterações
<i>Boltz</i>	0,12024	1,634	0,551	4,08	1777,44
<i>Exp</i>	0,19179	2,323	0,616	2,39	1043,89
<i>Fast</i>	0,65983	6,978	0,705	3,25	1416,36
Valor ótimo	0,119580	1,62	0,55	-	-

A Figura 18 mostra um gráfico comparando as três funções de decaimento de temperatura, considerando somente o número de soluções ótimas encontradas em relação ao tempo de otimização. É possível verificar que a função T_{BOLTZ} encontrou 400 soluções ótimas nas 1000 execuções em torno de 6 segundos de execução. Após este tempo, o número de soluções ótimas encontradas não crescem consideravelmente, com saturação em 430 soluções ótimas com um tempo de 9 segundos. O mesmo comportamento é encontrado com as funções T_{EXP} e T_{FAST} , porém com um número muito menor de soluções ótimas encontradas e com um tempo execução muito menor.

As próximas execuções referem-se à análise da otimização global seguida de otimização local. Para isso, os valores encontrados na análise anterior foram inseridos como solução inicial de uma heurística de otimização local. Com isso, pode-se verificar se é possível melhorar ainda mais o resultado final após a busca global com SA. Como técnica de busca local foi utilizado o algoritmo de Pontos Interiores (PRESS et al., 2007). Este método é adequado para otimização de espaços de projeto convexos lineares e não-lineares. O algoritmo foi implementado em Matlab através da função *fmincon*. Os resultados podem ser vistos na Tabela 7. Nestes resultados, é possível verificar o melhoramento das soluções após a inserção da busca local. A média da função custo após as 1000 execuções com as três funções de decaimento de temperatura estão mais próximas da solução ótima, com relação à análise anterior. O tempo total de execução (incluindo o tempo de otimização global e local) foi aumentado em cerca de 50%, porém são encontrados valores razoáveis já próximo aos 7 segundos.

Comparando os valores de \bar{W} e \bar{L} , é possível notar que houve uma melhora significativa em todas as funções de decaimento de temperatura, pois os valores encontrados

Figura 18: Número de soluções ótimas encontradas com relação ao tempo de execução para o processo de otimização global de um amplificador diferencial, considerando três funções de decaimento de temperatura e a função G_{FAST} .

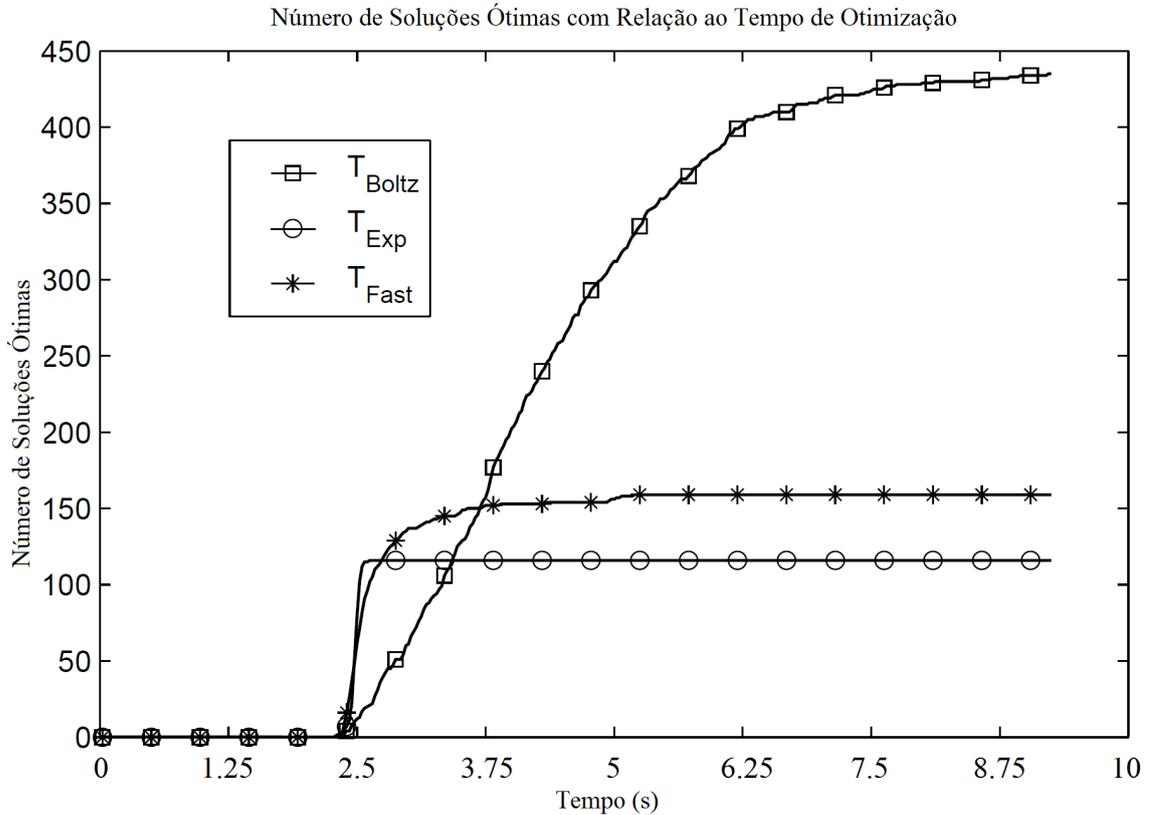


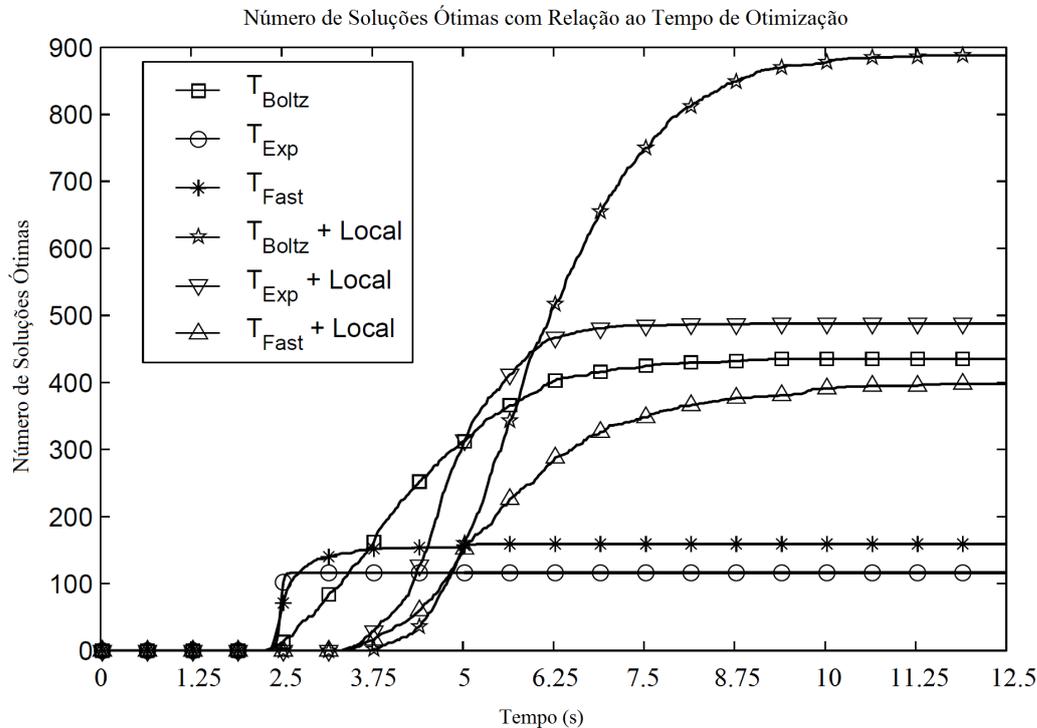
Tabela 7: Valores médios do projeto do amplificador diferencial após a otimização global seguida de otimização local

Função	\bar{f}_c	\bar{W}	\bar{L}	Tempo de Execução (s)
Boltz	0,11988	1,625	0,550	7,83
Exp	0,12103	1,636	0,562	4,63
Fast	0,12205	1,638	0,572	6,61
Valor ótimo	0,119580	1,62	0,55	-

estão mais próximos da solução ótima.

No que se refere ao número de soluções ótimas encontradas durante as 1000 execuções, a busca local demonstrou uma enorme melhoria, conforme pode-se observar na Figura 19. De acordo com esta figura, 90% dos resultados encontrados com a função T_{BOLTZ} representam a solução ótima do problema, uma melhora maior do que 50% em relação à exploração do espaço de projeto utilizando somente otimização global. O mesmo ocorre com as outras funções de decaimento de temperatura.

Figura 19: Número de soluções ótimas com relação ao tempo de otimização para a otimização global seguida de local para o amplificador diferencial utilizando função G_{FAST} .

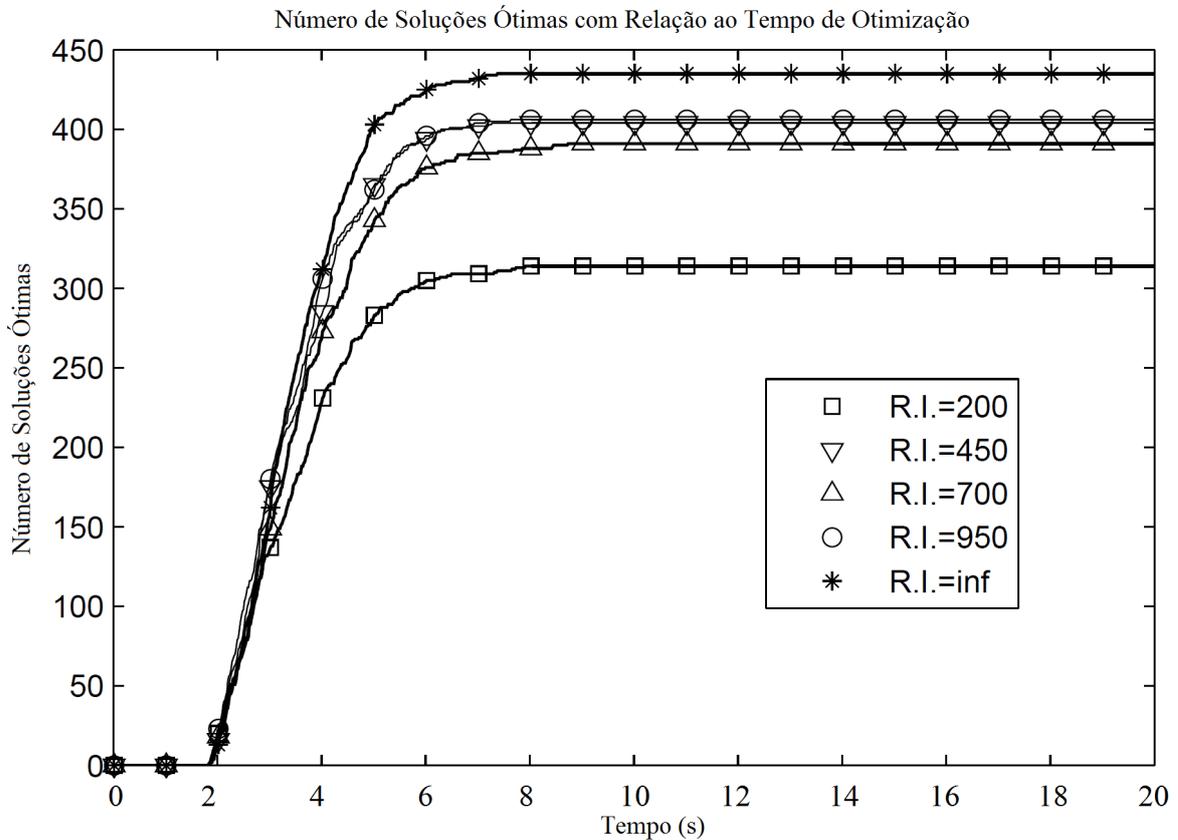


Para a análise da influência do *reannealing* no processo de otimização, foram executados alguns experimentos com *Simulated Annealing*. Os experimentos tratam de executar a análise anterior utilizando intervalos de *reannealing* (R.I.) iguais a 200, 450, 700 e 950 iterações. Esta análise faz com que a cada intervalo o parâmetro de temperatura seja reinicializado. Novamente, 1000 execuções foram feitas para a análise dos resultados.

A Figura 20 mostra a relação entre o número de soluções ótimas encontradas em função do tempo de otimização para a função de decaimento de temperatura T_{BOLTZ} com todos os intervalos de *reannealing*. Nesta figura, pode-se perceber que o *reannealing* afeta o desempenho da busca global pela heurística. A melhor solução é obtida sem a utilização de *reannealing* (intervalo de *reannealing* igual a infinito), demonstrando assim que a estratégia não colabora quando se utiliza a função de decaimento de temperatura T_{BOLTZ} .

Quando é utilizada a função de decaimento de temperatura T_{EXP} , o comportamento do processo de otimização em relação ao intervalo de *reannealing* é oposto ao encontrado com a função T_{BOLTZ} . Com a redução no intervalo de *reannealing*, um maior número de soluções ótimas são encontradas, de acordo com a Figura 21. O mesmo ocorre para a função de decaimento de temperatura T_{FAST} , como mostrado na Figura 22. Nesta figura, pode-se verificar que, à medida que o intervalo de *reannealing* é reduzido, o número de soluções ótimas encontradas aumenta. Com o intuito de verificar um intervalo

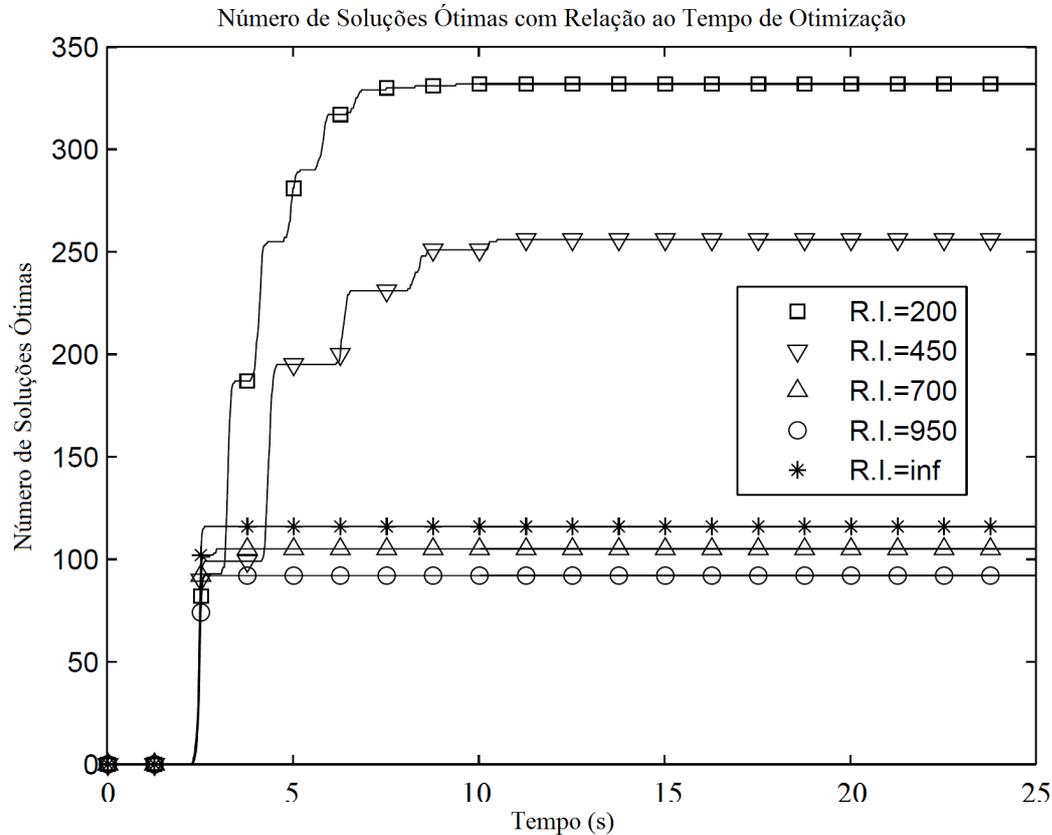
Figura 20: Número de resultados ótimos com relação ao tempo de otimização global do amplificador diferencial com diferentes valores de intervalo de *reannealing*, utilizando decaimento de temperatura pela função T_{BOLTZ} .



de *reannealing* ótimo, no qual o maior número de soluções ótimas é encontrado com a busca global, mais algumas análises foram executadas. Estas análises estão mostradas na Figura 23, na qual pode-se perceber que o intervalo de *reannealing* que produz o maior número de soluções ótimas está próximo de 100. Abaixo e acima deste valor o número de soluções ótimas encontradas é reduzido.

Para as análises mostradas anteriormente, foi utilizada como função de geração de novas soluções a função G_{FAST} . Para analisar o efeito da segunda função de geração de novas soluções, função G_{BOLTZ} , o processo de otimização com as três funções de decaimento de temperatura foi executado novamente. A Figura 24 mostra o número de soluções ótimas encontradas em função do tempo de otimização para as 6 combinações formadas pelas funções de geração de soluções e funções de decaimento de temperatura. Com estes resultados, pode-se verificar que a qualidade das soluções encontradas utilizando funções de decaimento de temperatura e geração de novas soluções com a função G_{BOLTZ} é muito melhor em relação às análises anteriores. Esta melhora deve-se ao fato de que esta combinação apresenta a melhor analogia com o sistema termodinâmico. Devido a isso, a probabilidade de encontrar a solução ótima é alta (MICHALEWICZ; FOGEL, 2000). Para

Figura 21: Número de resultados ótimos com relação ao tempo de otimização global do amplificador diferencial com diferentes valores de intervalo de *reannealing*, utilizando decaimento de temperatura pela função T_{EXP} .



as funções de decaimento de temperatura T_{EXP} e T_{FAST} , a variação é pequena quando se altera a função de geração de novas soluções, porém são encontrados melhores resultados quando é utilizada a geração de novas soluções com a função G_{FAST} .

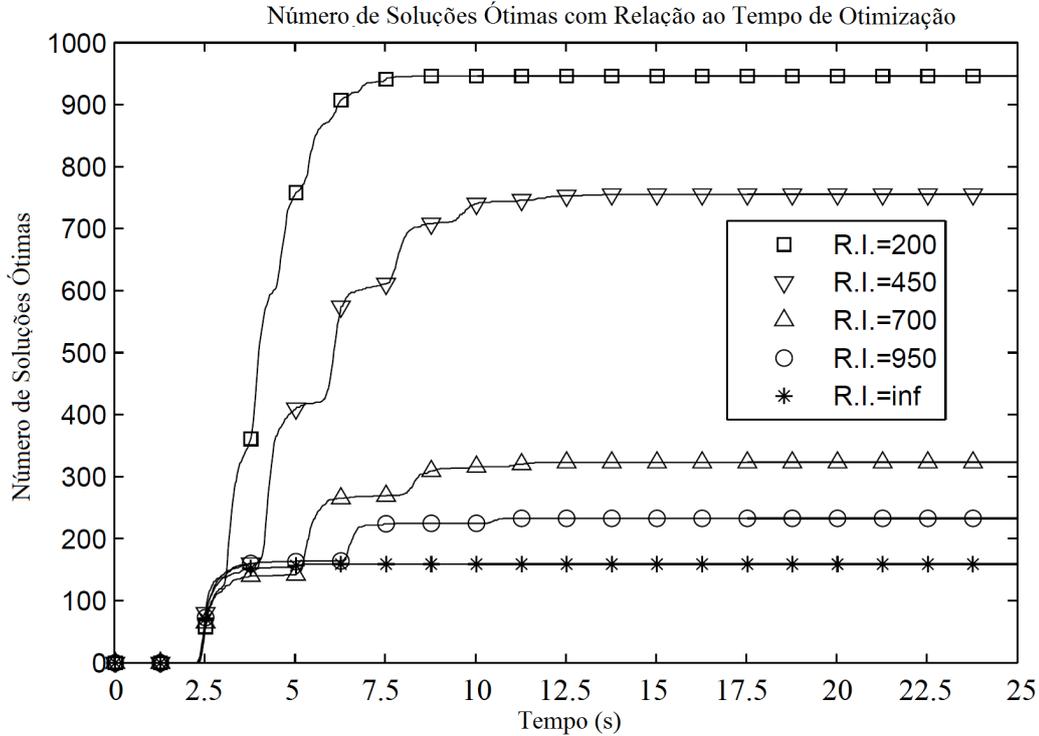
As análises do SA mostradas nesta seção serviram de base para a publicação de um capítulo de livro (SEVERO et al., 2012).

3.3.2 Otimização com Algoritmos Genéticos (GA)

Os algoritmos genéticos (GA) baseiam-se nos modelos das teorias biológicas da evolução natural e genética para realizar a otimização de funções não-lineares (FLOUDAS; PARDALOS, 2008). O GA é uma meta-heurística não-determinística que trabalha com uma população de soluções simultaneamente. Cada solução pertencente à população é chamada de indivíduo e a cada indivíduo é associado um valor de função custo. Nos algoritmos genéticos, a função custo é comumente chamada de função de aptidão (*fitness*).

As iterações do processo de otimização com GA são chamadas de gerações, em analogia à evolução natural. Ou seja, a cada iteração os indivíduos da população são cruzados, gerando novos indivíduos que herdam características de ambos os pais (recom-

Figura 22: Número de resultados ótimos com relação ao tempo de otimização global do amplificador diferencial com diferentes valores de intervalo de *reannealing*, utilizando decaimento de temperatura pela função T_{FAST} .



binacão), podendo este novo indivíduo sofrer ou não mutações.

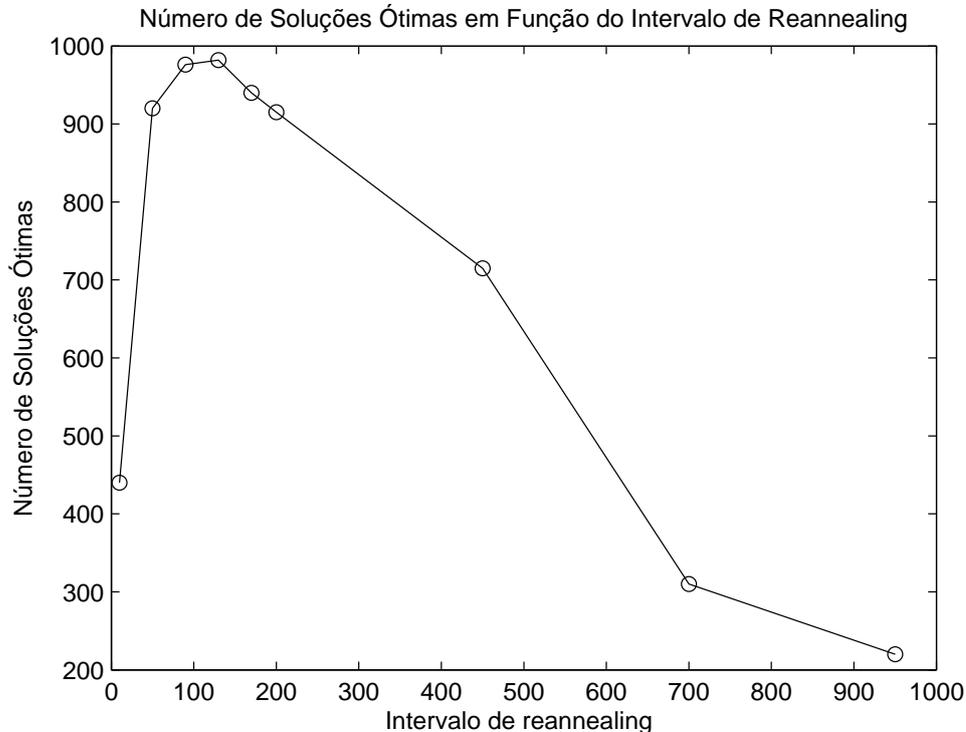
No GA, cada indivíduo da população é chamado de cromossomo. Os cromossomos de um algoritmo genético podem ser representados (codificados) de diversas formas. As formas de codificação mais utilizadas são codificação Binária e Real (SIVANANDAM; DEEPA, 2008). Em um cromossomo binário as variáveis de otimização são codificadas em números binários, cujo o número de bits depende da resolução de cada uma das variáveis. O número de bits necessários para cada variável do cromossomo é definido pela seguinte equação:

$$n_i = \log_2 \left(\frac{b_i - a_i}{\epsilon_i} \right) \quad (3.23)$$

em que a_i e b_i são os limites inferior e superior da variável i , respectivamente, e ϵ_i representa a resolução da variável i . Devido à codificação, antes de efetuar o cálculo da função custo é necessário realizar a decodificação do cromossomo binário. Para cromossomos reais não são necessárias codificações e decodificações, pois o cromossomo é visto como um vetor real em que suas posições são ocupadas pelas variáveis reais do problema.

O fluxograma do GA aplicado ao dimensionamento de circuitos integrados analógicos é mostrado na Figura 25. Neste fluxograma, o núcleo de otimização com algoritmos genéticos recebe como entradas os parâmetros de configuração da heurística, as especificações de projeto (neste caso, os dados da Tabela 2) e os dados da tecnologia de fabricação

Figura 23: Máximo número de soluções ótimas obtidas com a otimização global com relação ao intervalo de *reannealing* utilizado com função de decaimento de temperatura T_{FAST} .

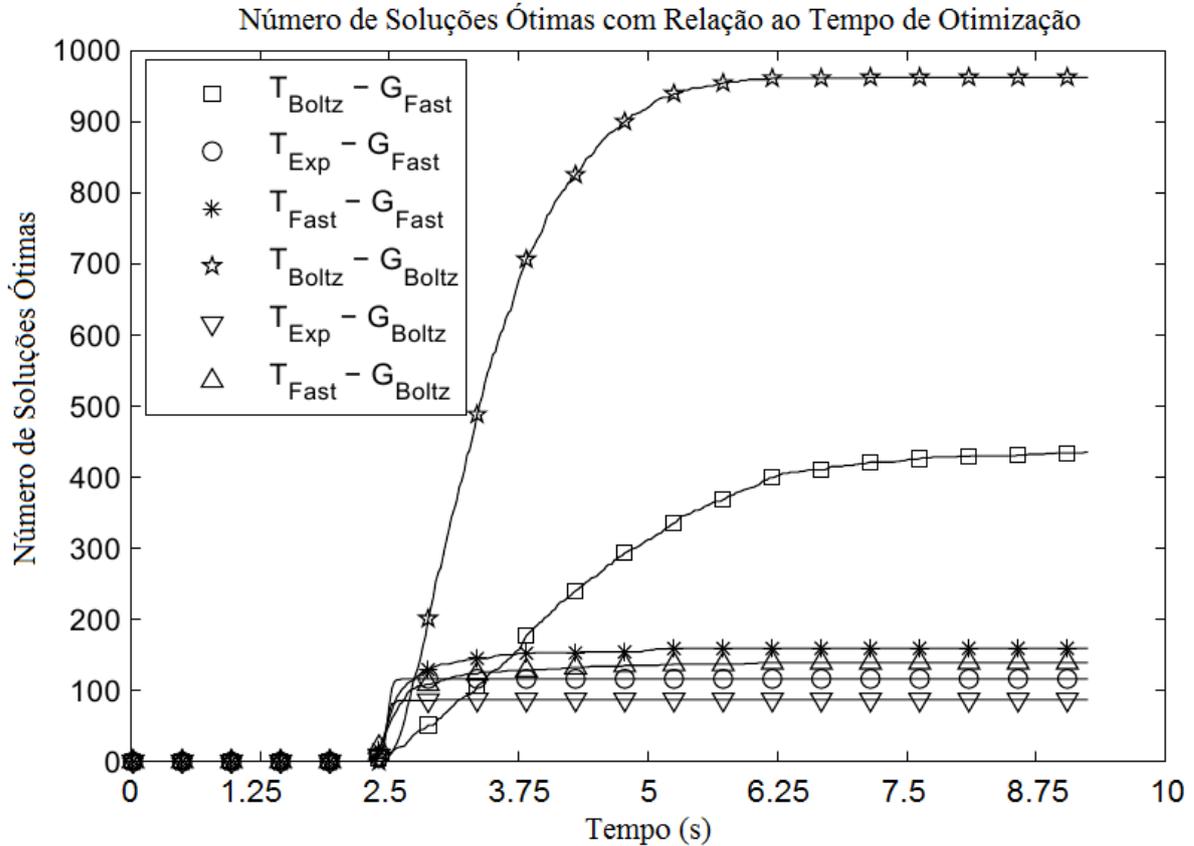


(neste caso os parâmetros de simulação da tecnologia XFAB 0,18 μm).

O primeiro passo é a criação da população de soluções. A criação da população é dada por uma função de inicialização que, em geral, gera aleatoriamente valores dentro da faixa de limites de cada variável (LINDEN, 2006). Para cada solução criada, é necessário efetuar a avaliação da solução através da função custo. Neste problema de dimensionamento de circuitos integrados analógicos a função custo é definida pela Equação 3.18, e as especificações do circuito são estimadas por simulação elétrica.

Após a criação da população, deve-se selecionar soluções na população (soluções pais) para que estas sejam utilizadas para a geração de novas soluções. A seleção das soluções pais é feita por uma função de seleção. A função de seleção é um dos principais pontos no fluxo de execução de um GA, uma vez que as novas soluções geradas dependem das soluções selecionadas da população (SIVANANDAM; DEEPA, 2008). A implementação para Matlab dos algoritmos genéticos GAOT (*Genetic Algorithm Optimization Toolbox*), desenvolvida por Houck, Joines e Kay (1995), apresenta as seguintes funções de seleção: *roulette wheel*, seleção por ranqueamento geométrico normalizado e seleção por torneio. A função de seleção *roulette wheel* baseia-se em uma probabilidade de seleção associada a cada cromossomo da população. Para GA de minimização, esta probabilidade é calculada

Figura 24: Número de resultados ótimos com relação ao tempo de otimização global do amplificador diferencial para as diferentes funções de geração de novas soluções.



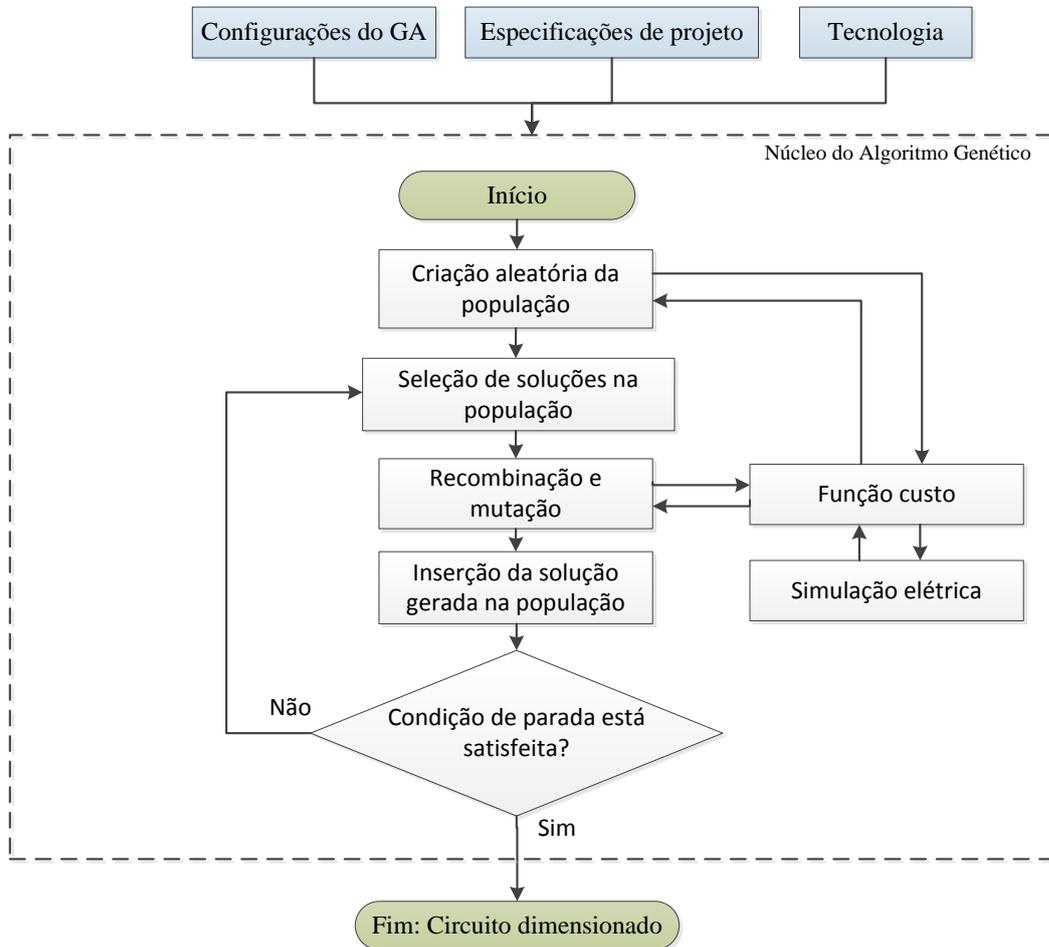
com a seguinte equação:

$$P_i = \frac{1}{f_c(X_i) \cdot \sum_{j=1}^{NPOP} \left(\frac{1}{f_c(X_j)} \right)} \quad (3.24)$$

em que f_c é a função custo, X_i ou X_j são cromossomos da população e $NPOP$ representa o número de cromossomos (indivíduos) que constituem a população. Esta probabilidade é normalizada em função do somatório do inverso da função custo, de maneira que a soma de todas as probabilidades seja igual à unidade. A probabilidade de seleção é inversamente proporcional ao valor da função custo, de forma que soluções boas (menor valor de f_c) apresentam maior chance de seleção. Neste método, após o cálculo da probabilidade de seleção de todos os cromossomos da população, a roleta é formada, fazendo com que cada solução ocupe uma faixa relativa à sua probabilidade. Para selecionar a solução, um número aleatório é gerado e comparado com as faixas da roleta. A solução que possui a faixa de probabilidade do número gerado é escolhida. Normalmente, a seleção envolve a seleção de duas soluções. Assim, o procedimento pode ser repetido com a exclusão da solução já selecionada.

O segundo método de seleção, baseado no ranqueamento geométrico normalizado, tem como base a ordenação das soluções de acordo com o valor da função custo. Com

Figura 25: Fluxograma de dimensionamento automático utilizando algoritmos genéticos (GA).



base nesta ordenação, a probabilidade de seleção de cada cromossomo é calculada com a seguinte equação:

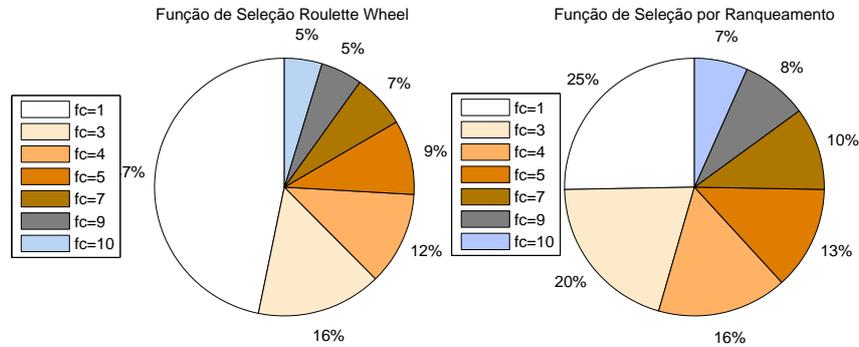
$$P_i = \frac{q \cdot (1 - q)^{r_i - 1}}{1 - (1 - q)^{NPOP}} \quad (3.25)$$

onde q é o parâmetro de ranqueamento e r_i é a posição no ranking da solução i .

Para analisar um exemplo de comparação entre as duas funções de seleção, a Figura 26 mostra a probabilidade de escolha para uma população cujos cromossomos possuem valores de função custo iguais a 1, 3, 4, 5, 7, 9 e 10. Nesta figura, verifica-se que o método de seleção por *roulette wheel* tende a apresentar maior probabilidade de seleção para as soluções melhores. Este fato faz com que a melhor solução tenha uma grande probabilidade de seleção quando o seu valor de função for muito menor que as demais soluções. Assim, a solução com $f_c(X) = 1$ possui uma alta probabilidade de seleção (47%). Já no método de ranqueamento geométrico normalizado, este efeito tende a ser reduzido, uma vez que a probabilidade não depende do valor da função custo e sim da

sua posição no ranking.

Figura 26: Comparação da probabilidade de seleção com as funções *roulette wheel* e ranqueamento geométrico normalizado.



Outro método que pode ser utilizado para a seleção de soluções da população é o modelo baseado em torneio. Este método não tem uma probabilidade de escolha associada a cada solução, pois baseia-se em torneios aleatórios. Um torneio refere-se à seleção de indivíduos em posições aleatórias na população. O indivíduo com menor valor de função custo vence o torneio. Vários torneios podem ser gerados e, ao fim, a melhor solução de todas é selecionada. Como parâmetro, este método possui somente o número de torneios (k). Com a utilização da seleção por torneio as piores soluções apresentam probabilidade de aceitação iguais às melhores soluções, fato este que causa uma maior aleatoriedade no processo de otimização.

Na sequência do fluxograma da Figura 25, são aplicados os operadores de recombinação (*crossover*) e mutação. Os operadores são responsáveis pelo processo de busca com o GA, pois, a partir das soluções selecionadas, as funções de recombinação e mutação geram novas soluções. As funções de recombinação e mutação dependem do tipo de representação utilizada para as soluções. No caso de cromossomos binários, utiliza-se as técnicas chamadas de mutação binária e recombinação simples. Assumindo as soluções X e Y como sendo dois cromossomos selecionados, formados pelas variáveis x_i e y_i , a recombinação simples é definida como

$$x'_i = \begin{cases} x_i & \text{se } i < r; \\ y_i & \text{Outro caso .} \end{cases} \quad (3.26)$$

$$y'_i = \begin{cases} y_i & \text{se } i < r; \\ x_i & \text{Outro caso .} \end{cases} \quad (3.27)$$

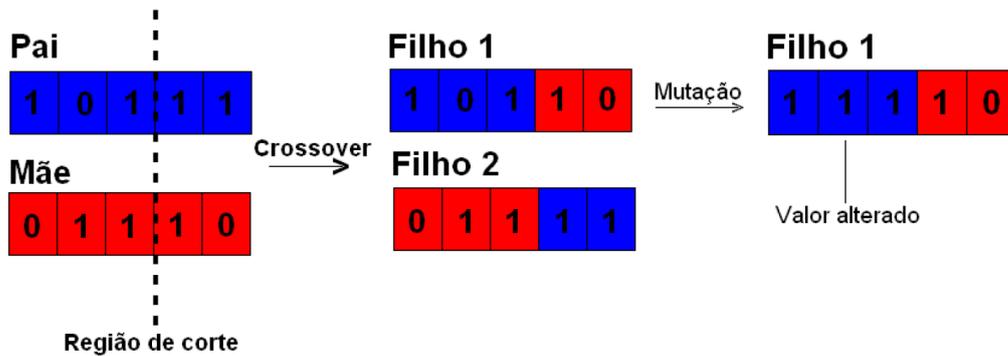
em que r é um número gerado aleatoriamente. Este método simplesmente faz a quebra de um cromossomo em uma região de corte definida por r e gera dois cromossomos (X' e Y')

contendo uma parte de cada cromossomo pai. Após a recombinação, ocorre a mutação. A mutação binária é definida pela seguinte equação:

$$x'_i = \begin{cases} 1 - x_i & \text{se } U(0, 1) < p_m; \\ x_i & \text{Outro caso .} \end{cases} \quad (3.28)$$

onde p_m é um parâmetro de probabilidade de mutação e $U(0, 1)$ é um número aleatório normalmente distribuído entre 0 e 1. O funcionamento deste método é simplesmente a inversão de um bit do cromossomo, cuja posição é definida aleatoriamente e possui uma probabilidade de ocorrência dada por p_m . A Figura 27 ilustra um exemplo de recombinação e mutação simples com cromossomos binários, onde as Equações 3.26, 3.27 e 3.28 são utilizadas para dar origem a duas novas soluções.

Figura 27: Recombinação e mutação em cromossomos binários.



Na utilização de cromossomos reais existem inúmeras formas de desempenhar a recombinação e mutação dos cromossomos. Em Houck, Joines e Kay (1995) são implementadas as seguintes funções de recombinação e mutação: recombinação simples, recombinação aritmética, recombinação heurística, mutação uniforme, mutação não-uniforme e mutação de fronteira. A recombinação simples é idêntica à utilizada em cromossomos binários dados pelas Equações 3.26 e 3.27, porém cada parte do cromossomo possui um número real ao invés de um bit. A recombinação aritmética é definida pelas seguintes equações:

$$X' = rX + (1 - r)Y \quad (3.29)$$

$$Y' = (1 - r)X + rY \quad (3.30)$$

em que r é um valor aleatório. Com a recombinação aritmética, os novos cromossomos (X' e Y') são gerados com duas combinações lineares complementares utilizando as soluções pais (X e Y). A recombinação heurística utiliza uma extrapolação linear para gerar as novas soluções. Para o cálculo, as informações da função custo são utilizadas para determinar a melhor solução entre X e Y . Sendo X melhor que Y , as soluções são

geradas com a seguinte função:

$$X' = X + r(X - Y) \quad (3.31)$$

$$Y' = X \quad (3.32)$$

em que r é um número aleatório entre 0 e 1. Após o cálculo de X' e Y' , como trata-se de uma extrapolação, os valores gerados são verificados com a faixa de limites de cada variável. Caso a solução gerada não seja válida, é gerada uma nova solução com a equação anterior. Caso sejam executados um número k de vezes a equação e uma solução válida não foi gerada, as soluções pais são retornadas como soluções geradas.

A função de mutação mais simples é a mutação uniforme, baseada na seguinte equação:

$$x'_i = \begin{cases} U(a_i, b_i) & \text{se } i = j; \\ x_i & \text{Outro caso.} \end{cases} \quad (3.33)$$

em que o cromossomo gerado (X') pode ter a variável x_i substituída por um número aleatório entre os limites da a e b desta variável. O segundo tipo de mutação, mutação não-uniforme, é definida como

$$x'_i = \begin{cases} x_i + (b_i - x_i)f(G) & \text{se } r_1 < 0,5; \\ x_i - (x_i - a_i)f(G) & \text{se } r_1 \geq 0,5. \end{cases} \quad (3.34)$$

em que a_i e b_i são os limites inferior e superior da variável i , r_1 é um número aleatório entre 0 e 1 e $f(G)$ é dado por

$$f(G) = \left[r_2 \left(1 - \frac{G}{G_{max}} \right) \right]^b \quad (3.35)$$

onde r_2 é um número aleatório com a mesma faixa de r_1 , G é a geração corrente do GA, G_{max} é o número máximo de gerações admitido e b é um parâmetro de forma da equação. Com a mutação não-uniforme a variação é alta nas gerações iniciais e tende a zero nas gerações finais. Além disso, este método apresenta a mesma probabilidade de alterar o valor da variável para cima (se $r_1 < 0,5$) ou para baixo (se $r_1 \geq 0,5$).

O terceiro tipo de mutação é a mutação de fronteira. Neste método, uma variável pertencente ao cromossomo é escolhida aleatoriamente e substituída pelo máximo (b) ou mínimo (a) valor permitido. A definição de substituição por máximo ou mínimo é feita por um número aleatório r entre 0 e 1, conforme mostra a seguinte equação:

$$x'_i = \begin{cases} a_i & \text{se } r < 0,5; \\ b_i & \text{se } r \geq 0,5. \end{cases} \quad (3.36)$$

Após a aplicação das funções de recombinação e mutação, as soluções geradas são inseridas na população.

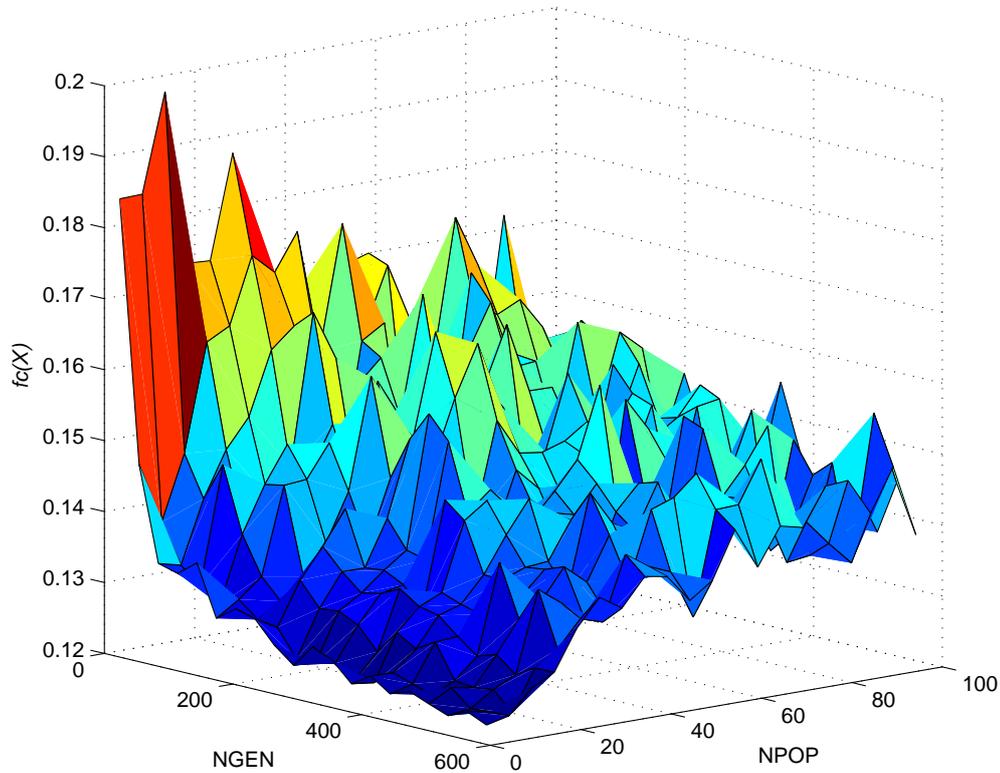
Em sequência ao fluxograma da Figura 25, a condição de parada do algoritmo é testada e, se satisfeita, o circuito está dimensionado. Caso contrário, novas soluções são selecionadas na população. As funções de recombinação e mutação são aplicadas até que a condição de parada seja satisfeita. Como condição de parada normalmente é utilizado um número máximo de gerações ($NGEN$) do GA ou uma variação mínima na função custo.

Para a análise do fluxo de dimensionamento de circuitos analógicos utilizando GA, algumas combinações de configurações foram executadas de forma que o desempenho possa ser comparado no dimensionamento do amplificador diferencial. Para isso, foi utilizada a implementação de GA em Matlab proposta por Houck, Joines e Kay (1995). Nestas análises, foi utilizado o espaço de projeto discreto do amplificador diferencial utilizado para traçar o gráfico da Figura 15. Esta escolha deve-se a possibilidade de eliminar a simulação elétrica a cada iteração, de forma a reduzir o tempo de dimensionamento e tornar possível inúmeras execuções para cada configuração.

A primeira análise tem por objetivo verificar a importância do tamanho da população ($NPOP$) e do número de gerações executadas ($NGEN$). Para esta análise, foi utilizado GA com cromossomos reais, recombinação simples, mutação uniforme e seleção por *roulette wheel*. O espaço de projeto do amplificador diferencial simplificado foi explorado 10 vezes para cada configuração executada. Cada configuração refere-se à combinação dos valores $5 \leq NPOP \leq 200$ com passos de 10 unidades e $10 \leq NGEN \leq 600$ com passos de 50 unidades. A Figura 28 mostra os resultados desta análise. Nesta figura, pode-se perceber que um aumento em $NPOP$ causa um aumento do efeito aleatório, fazendo com que a solução final encontrada pelo GA tenha a tendência de ser o melhor valor gerado aleatoriamente na população inicial. Também pode-se perceber que a combinação de valores pequenos de $NPOP$ e $NGEN$ resultam nos piores resultados (maior valor de $fc(X)$). Isso deve-se ao fato de que, com uma população muito pequena, a diversidade de soluções é pequena e com $NGEN$ pequeno são geradas poucas novas soluções. O gráfico mostra também que as melhores soluções (pequenos valores de $fc(X)$) são encontradas com uma população pequena (em torno de 10 indivíduos), porém com um grande número de gerações (valores superiores a 100 gerações).

A próxima análise refere-se à comparação entre cromossomos reais e binários. O primeiro parâmetro a ser definido é a resolução da representação (ϵ), que define a variação mínima para considerar duas soluções diferentes. Nos cromossomos binários, ϵ define também o número de bits da representação, como mostrado na Equação 3.23. Nesta análise foi utilizado um valor de ϵ igual a $0,01\mu m$, dado pela grade de variação geométrica mínima da tecnologia XFAB 0, $18\mu m$. Nesta análise foram utilizadas recombinação simples, mutação binária (para cromossomos binários), mutação uniforme (para cromossomos reais) e seleção por *roulette wheel*. Na mutação binária foram utilizados dois valores dis-

Figura 28: Função custo final obtida com a execução do GA com a combinação dos valores $5 \leq NPOP \leq 200$ com passos de 10 unidades e $10 \leq NGEN \leq 600$ com passos de 50 unidades.



tintos para a taxa de mutação (p_m): 0,5% e 5%. Foram feitas 1000 execuções para cada configuração e os valores médios encontrados nestas execuções estão mostrados na Tabela 8.

Nestes resultados, pode-se verificar que, com a utilização de GA, tanto com cromossomos binários como com cromossomos reais, boas soluções podem ser obtidas, pois são encontradas soluções próximas à solução ótima. Para isso, as configurações dos parâmetros devem ser ajustadas experimentalmente. Como exemplo, os melhores resultados para os cromossomos binários são obtidos com uma taxa de mutação maior (5%). Porém, esta evidência só foi possível após a experimentação de diferentes valores para p_m (aqui mostrados somente para 0,5% e 5%). Neste sentido, a utilização de cromossomos reais com mutação uniforme facilita a análise, pois não é necessária a configuração de parâmetros do algoritmo. Além disso, quando analisado o número de soluções ótimas encontradas, o resultado com cromossomos reais é melhor. Com relação ao tempo de execução, o GA com cromossomos reais é executado em menor tempo. Esta diferença de tempo deve-se, principalmente, à codificação e decodificação dos cromossomos binários que devem ser

Tabela 8: Resultados de execução do GA com cromossomos binários e reais

Cromossomo	p_m	NPOP	NGEN	\bar{f}_c	\bar{W}	\bar{L}	\overline{Tempo} (s)	NSO ¹
Binário	0,5%	10	10	0,221360	2,334	0,551	0,0417	0
Binário	5%	10	10	0,193722	2,050	0,552	0,0424	0
Binário	0,5%	10	100	0,174182	1,997	0,532	0,4203	0
Binário	5%	10	100	0,131788	1,651	0,529	0,4189	0
Binário	0,5%	10	1000	0,134777	1,680	0,525	4,2047	0
Binário	5%	10	1000	0,122786	1,609	0,538	4,2181	0
Real	-	10	10	0,188288	2,036	0,551	0,0144	3
Real	-	10	100	0,136151	1,703	0,534	0,1333	153
Real	-	10	1000	0,122486	1,624	0,542	1,3272	308
Solução ótima	-	-	-	0,119580	1,62	0,55	-	-

¹ NSO: Número de Soluções Ótimas encontradas

feitas a cada iteração. Nestes resultados, verifica-se que, com valores maiores de *NGEN*, um maior número de soluções ótimas é encontrado, porém às custas de um maior tempo de execução.

A análise seguinte tem como objetivo comparar o resultado de execução do GA com as diferentes funções de seleção de soluções da população. Para isso, foi utilizado o GA configurado com cromossomos reais e binários, recombinação simples, mutação uniforme (para cromossomos reais) e mutação binária (para cromossomos binários). Da mesma forma que a análise anterior, para cada configuração o GA foi executado 1000 vezes. A Tabela 9 mostra os resultados obtidos.

Nestes resultados, verifica-se que, com os três métodos de seleção, são obtidos bons resultados. Com relação aos parâmetros, pode-se verificar que, se for utilizado seleção por torneio, um pequeno número de torneios propicia os melhores resultados. Esta razão foi discutida anteriormente, pois com um número pequeno de torneios a probabilidade de seleção é praticamente igual para qualquer solução. Na medida que o número de torneios é aumentado, as melhores soluções têm uma maior chance de serem selecionadas e a diversidade de seleção é reduzida. Para a seleção com ranqueamento geométrico normalizado, os melhores resultados são obtidos com parâmetros percentuais menores.

Com a utilização dos métodos de seleção por ranqueamento geométrico normalizado com cromossomos reais, são encontrados resultados similares de função custo utilizando 100 ou 1000 gerações. Esta característica não é encontrada no método de seleção por *roulette wheel*. O maior número de soluções ótimas encontradas ainda é obtido com 1000 gerações, tendo um máximo de 86% de melhor soluções encontradas com o método de ranqueamento geométrico normalizado.

Com a utilização de cromossomos reais é possível gerar novas soluções com inúmeras formas de recombinação e mutação. Com o objetivo de verificar a importância da

Tabela 9: Resultados de execução do GA para as diferentes funções de seleção

Cromossomo	Seleção	Par.	NGEN	$\overline{f_c}$	\overline{W}	\overline{L}	\overline{Tempo} (s)	NSO ¹
Binário	<i>R. wheel</i>	-	100	0,131788	1,651	0,529	0,4189	0
Binário	<i>R. wheel</i>	-	1000	0,122786	1,609	0,538	4,2181	0
Binário	Ranking	5%	100	0,123565	1,604	0,537	0,4290	0
Binário	Ranking	50%	100	0,130938	1,625	0,534	0,4240	0
Binário	Ranking	5%	1000	0,122185	1,616	0,542	4,3662	0
Binário	Ranking	50%	1000	0,124025	1,590	0,535	4,6880	0
Binário	Torneio	2	100	0,125765	1,606	0,540	0,4253	0
Binário	Torneio	10	100	0,135734	1,676	0,541	0,4285	0
Binário	Torneio	100	100	0,135777	1,649	0,546	0,4313	0
Binário	Torneio	2	1000	0,122204	1,617	0,542	4,3057	0
Binário	Torneio	10	1000	0,124796	1,588	0,533	4,4775	0
Binário	Torneio	100	1000	0,123524	1,612	0,541	4,4966	0
Real	<i>R. wheel</i>	-	100	0,136151	1,703	0,534	0,1333	153
Real	<i>R. wheel</i>	-	1000	0,122486	1,624	0,542	1,3272	308
Real	Torneio	2	100	0,123999	1,600	0,535	0,1186	291
Real	Torneio	10	100	0,125417	1,589	0,533	0,1168	267
Real	Torneio	100	100	0,124956	1,594	0,533	0,1218	233
Real	Torneio	2	1000	0,120822	1,600	0,543	1,1745	678
Real	Torneio	10	1000	0,121421	1,598	0,542	1,1901	654
Real	Torneio	100	1000	0,121062	1,604	0,543	1,2497	726
Real	Ranking	50%	100	0,125685	1,600	0,533	0,1198	247
Real	Ranking	5%	100	0,124716	1,629	0,539	0,1303	208
Real	Ranking	0,5%	100	0,125878	1,626	0,533	0,1368	104
Real	Ranking	50%	1000	0,121737	1,599	0,540	1,1857	636
Real	Ranking	5%	1000	0,120503	1,610	0,545	1,2874	752
Real	Ranking	0,5%	1000	0,120107	1,615	0,548	1,3635	860
Solução ótima	-	-	-	0,119580	1,62	0,55	-	-

¹ NSO: Número de Soluções Ótimas encontradas

recombinação e mutação foram feitas execuções do GA sob diferentes formas de reprodução de soluções. Para isso, o GA foi configurado com seleção dada por *roulette wheel*, mutação uniforme (para o teste de recombinação) e recombinação simples (para o teste de mutação). Os resultados desta análise estão mostrados nas Tabelas 10 e 11.

Uma análise na Tabela 10 permite perceber que os melhores resultados, em relação ao valor de $f_c(X)$, são obtidos com as funções de recombinação aritmética e recombinação heurística. O menor valor de $f_c(x)$ justifica-se pelo maior número de soluções ótimas encontradas (NSO), pois para estas heurísticas o NSO foi no mínimo 32% maior que o encontrado com recombinação simples. Dentre os métodos, o único que apresenta parâmetros de configuração é o método de recombinação heurística. O parâmetro refere-se ao número de tentativas de recombinação antes de retornar. Em outras palavras, significa o número máximo de vezes que a Equação 3.31 é executada. Nos resultados com recombinação heurística, verifica-se que o parâmetro com valor igual a 10 apresenta

Tabela 10: Resultados de execução do GA com as diferentes formas de recombinação

Cromossomo	Recomb.	Par.	NGEN	\bar{f}_c	\bar{W}	\bar{L}	\overline{Tempo} (s)	NSO ¹
Binário	simples	-	100	0,174182	1,997	0,532	0,4203	0
Binário	simples	-	1000	0,122786	1,609	0,538	4,2181	0
Real	simples	-	100	0,163118	1,854	0,522	0,0611	0
Real	simples	-	1000	0,123630	1,587	0,536	0,5513	410
Real	aritmética	-	100	0,147758	1,847	0,542	0,0714	21
Real	aritmética	-	1000	0,121359	1,610	0,542	0,5813	671
Real	heurística	1	100	0,146346	1,681	0,528	0,0638	10
Real	heurística	10	100	0,141623	1,704	0,529	0,0672	42
Real	heurística	100	100	0,141553	1,709	0,535	0,0687	33
Real	heurística	1	1000	0,122105	1,600	0,541	0,5946	541
Real	heurística	10	1000	0,121335	1,605	0,543	0,6035	686
Real	heurística	100	1000	0,121648	1,599	0,542	0,5968	634
Solução ótima	-	-	-	0,119580	1,62	0,55	-	-

¹ NSO: Número de Soluções Ótimas encontradas

melhores resultados que os valores 1 e 100. Com relação à facilidade de utilização e melhores resultados encontrados, a recombinação aritmética é indicada, uma vez que não necessita de configuração de parâmetros e apresenta bons resultados.

Nos resultados de execução com os diferentes métodos de mutação (Tabela 11) verifica-se que, ao contrário da recombinação, os resultados dependem muito da técnica escolhida. Os melhores resultados foram encontrados utilizando mutação não-uniforme e os piores resultados foram obtidos com a utilização da mutação de fronteira. Com relação ao parâmetro da mutação não-uniforme - parâmetro de forma (b) - os melhores resultados foram obtidos com seu valor igual a 2. No qual, 60% das soluções encontradas equivalem à solução ótima.

De acordo com Liu et al. (2009) e Houck, Joines e Kay (1995), a heurística dos Algoritmos Genéticos apresenta uma enorme capacidade de encontrar valores ótimos globais, porém muitas vezes o algoritmo finaliza seu processo um pouco antes de atingi-los. Desta forma, uma importante melhoria nos resultados é obtida utilizando uma técnica de busca local após a otimização global com GA. Para a análise deste efeito, a solução obtida usando GA foi utilizada como solução inicial para a busca local. Para a busca local, foi utilizada a função *fmincon* (MATHEWORKS, 2012a) com o algoritmo de pontos interiores (PRESS et al., 2007). O GA foi configurado com seleção por *roulette wheel*, recombinação simples, mutação uniforme (para cromossomos reais) e mutação binária (para cromossomos binários).

A Tabela 12 mostra os resultados obtidos com a busca global seguida de local. Nesta tabela fica evidente a melhoria provocada pela busca local aos resultados obtidos com GA, pois, após a busca local, no mínimo 92,1% das execuções retornaram a solução

Tabela 11: Resultados de execução do GA com a utilização dos diferentes métodos de mutação

Cromossomo	Mutação	Par.	NGEN	\bar{f}_c	\bar{W}	\bar{L}	\overline{Tempo} (s)	NSO ¹
Binário	Binária	-	100	0,174182	1,997	0,532	0,4203	0
Binário	Binária	-	1000	0,122786	1,609	0,538	4,2181	0
Real	Uniforme	-	100	0,163118	1,854	0,522	0,0611	0
Real	Uniforme	-	1000	0,123630	1,587	0,536	0,5513	410
Real	Fronteira	-	100	0,243615	2,596	0,564	0,0549	0
Real	Fronteira	-	1000	0,240140	2,543	0,552	0,5040	0
Real	Não-Unif.	1	100	0,162833	1,858	0,527	0,0635	11
Real	Não-Unif.	2	100	0,183453	2,176	0,531	0,0640	42
Real	Não-Unif.	3	100	0,194191	2,190	0,545	0,0648	1
Real	Não-Unif.	10	100	0,234837	2,485	0,559	0,0625	0
Real	Não-Unif.	1	1000	0,124272	1,561	0,531	0,5542	461
Real	Não-Unif.	2	1000	0,122627	1,584	0,537	0,5733	600
Real	Não-Unif.	3	1000	0,125339	1,581	0,531	0,6077	491
Real	Não-Unif.	10	1000	0,168432	2,061	0,537	0,6179	60
Solução ótima	-	-	-	0,119580	1,62	0,55	-	-

¹ NSO: Número de Soluções Ótimas encontradas

Tabela 12: Resultados obtidos com a otimização global seguida de local

Cromossomo	Busca	NGEN	\bar{f}_c	\bar{W}	\bar{L}	\overline{Tempo} (s)	NSO ¹
Binário	Global	100	0,174182	1,997	0,532	0,4203	0
Binário	Global+Local	100	0,119878	1,616	0,549	0,6367	932
Binário	Global	1000	0,122786	1,609	0,538	4,2181	0
Binário	Global+Local	1000	0,119580	1,620	0,550	4,8029	1000
Real	Global	100	0,163118	1,854	0,522	0,0611	0
Real	Global+Local	100	0,119844	1,617	0,549	0,2745	921
Real	Global	1000	0,123630	1,587	0,536	0,5513	410
Real	Global+Local	1000	0,120094	1,614	0,548	0,7548	904
Solução ótima	-	-	0,119580	1,62	0,55	-	-

¹ NSO: Número de Soluções Ótimas encontradas

ótima. O melhor resultado foi obtido com cromossomos binários com 1000 gerações, no qual o número de soluções ótimas encontradas passou de 0 para 1000, ou seja, 100% das execuções retornaram a solução ótima. Este resultado deve-se ao fato de as soluções obtidas com a otimização global estarem próximas da solução ótima, mas nenhuma igual à solução ótima. O tempo de execução adicionando-se a busca local não representou grande aumento em relação ao tempo da execução global. Pode-se dizer ainda que a busca local provoca uma redução no tempo total de execução, uma vez que pode-se utilizar um número menor de gerações e mesmo assim obter melhores resultados que com um número grande de gerações somente com busca global.

As heurísticas analisadas neste trabalho possuem características distinta de imple-

mentação. Desta forma, uma comparação direta de resultados não pode ser feita. Neste contexto, o trabalho (SEVERO; GIRARDI, 2011) traz uma comparação entre as técnicas de otimização global e local baseados em várias execuções com diferentes configurações.

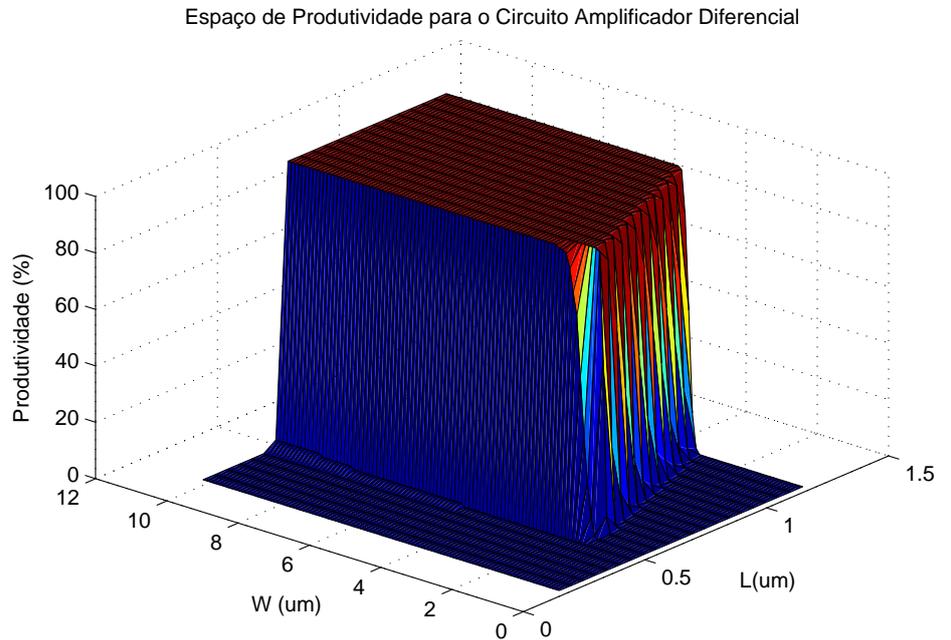
3.4 Design Centering

As técnicas de *design centering* surgiram devido ao fato de que as variações no processo de fabricação de circuitos integrados em tecnologias sub-micrométrica influenciam cada vez mais nas especificações dos circuitos analógicos. Como visto na seção 2.3, a otimização da produtividade pode ser feita de duas formas: maximização do *yield* do circuito (análise estatística) ou a maximização da distância entre o valor requerido e o valor encontrado para cada especificação (análise geométrica).

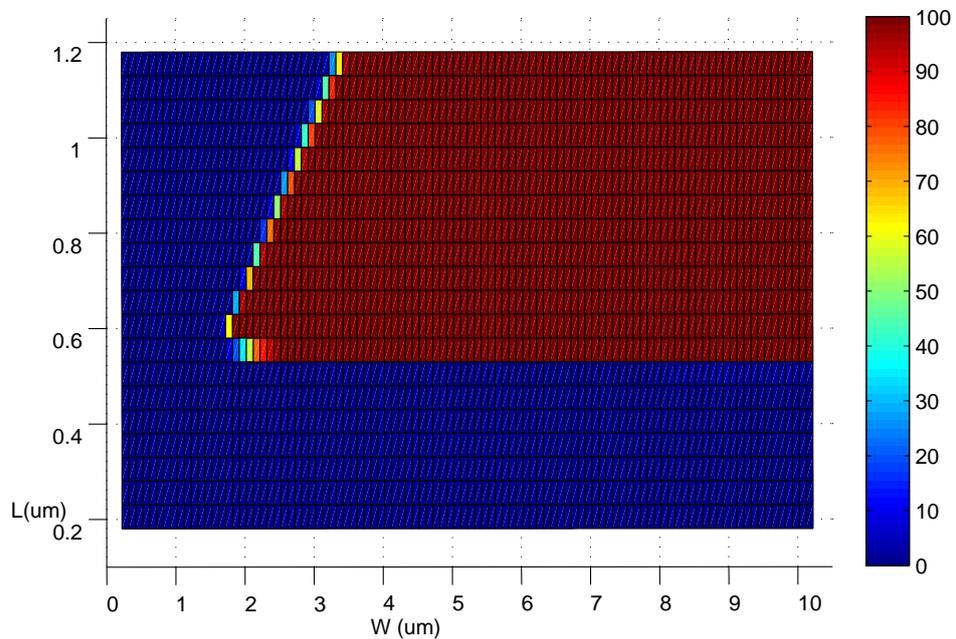
Para analisar a produtividade no espaço de projeto do amplificador diferencial simplificado utilizado neste capítulo, o espaço de projeto da Figura 15 foi simulado novamente, de forma que a produtividade do circuito fosse medida. Para a análise da produtividade foram realizadas simulações de Monte Carlo com 100 rodadas utilizando o modelo com variações locais e globais da tecnologia XFAB 0,18 μm , disponibilizado pela *foundry*. A Figura 29 mostra o espaço de projeto de produtividade do circuito. Para a obtenção do espaço de projeto de produtividade, a cada solução, as especificações medidas foram comparadas com as especificações requeridas. E assim, o *yield* equivalente ao número percentual de análises de Monte Carlo que atingem as especificações requeridas foi obtido. Neste espaço é possível verificar que o *yield* varia bruscamente em uma região de fronteira. Antes e depois desta região de fronteira existe uma vasta região do espaço com produtividade mínima (0%) ou produtividade máxima (100%). Além disso, a região de alta produtividade está localizada na região de W e L grandes, mostrando assim um *trade-off* entre *yield* e área do circuito.

Com o espaço de produtividade da Figura 29 pode-se verificar que as técnicas de *design centering* são muito necessárias para afastar a solução da região de fronteira. Sem elas, uma pequena variação nos parâmetros pode levar a uma enorme perda de produtividade. Para analisar o efeito das variações de parâmetros, a solução ótima do espaço de projeto do amplificador diferencial ($W = 1,62 \mu m$ e $L = 0,55 \mu m$), analisada nas seções anteriores, foi simulada através da análise de Monte Carlo. Para isso, foram utilizados os dados de variações de parâmetros da tecnologia XFAB de 0,18 μm . As análises de Monte Carlo foram executadas com 1000 rodadas de simulação, utilizando distribuição gaussiana. Para analisar o comportamento das variações, foram feitas simulações com variação global de parâmetros, variação local e ambas variações locais e globais, simultaneamente. Os resultados das análises de Monte Carlo e o valor nominal para a solução ótima do espaço de projeto estão mostrados na Tabela 13. Nesta Tabela, pode-se perceber que,

Figura 29: Espaço de produtividade para o amplificador diferencial de duas variáveis. (a) espaço tridimensional e (b) espaço bidimensional.



(a)



(b)

embora os valores nominais do circuitos atendam às restrições impostas (especificações requeridas), quando ocorrem variações nos parâmetros nem todas as especificações atendem aos valores requeridos. Com variação global, a produtividade do circuito é igual a 51,8%. Para variações locais o *yield* cai significativamente, resultando em um valor igual a 8,8%. Este resultado é justificado pelo fato de que blocos básicos como espelhos de corrente e

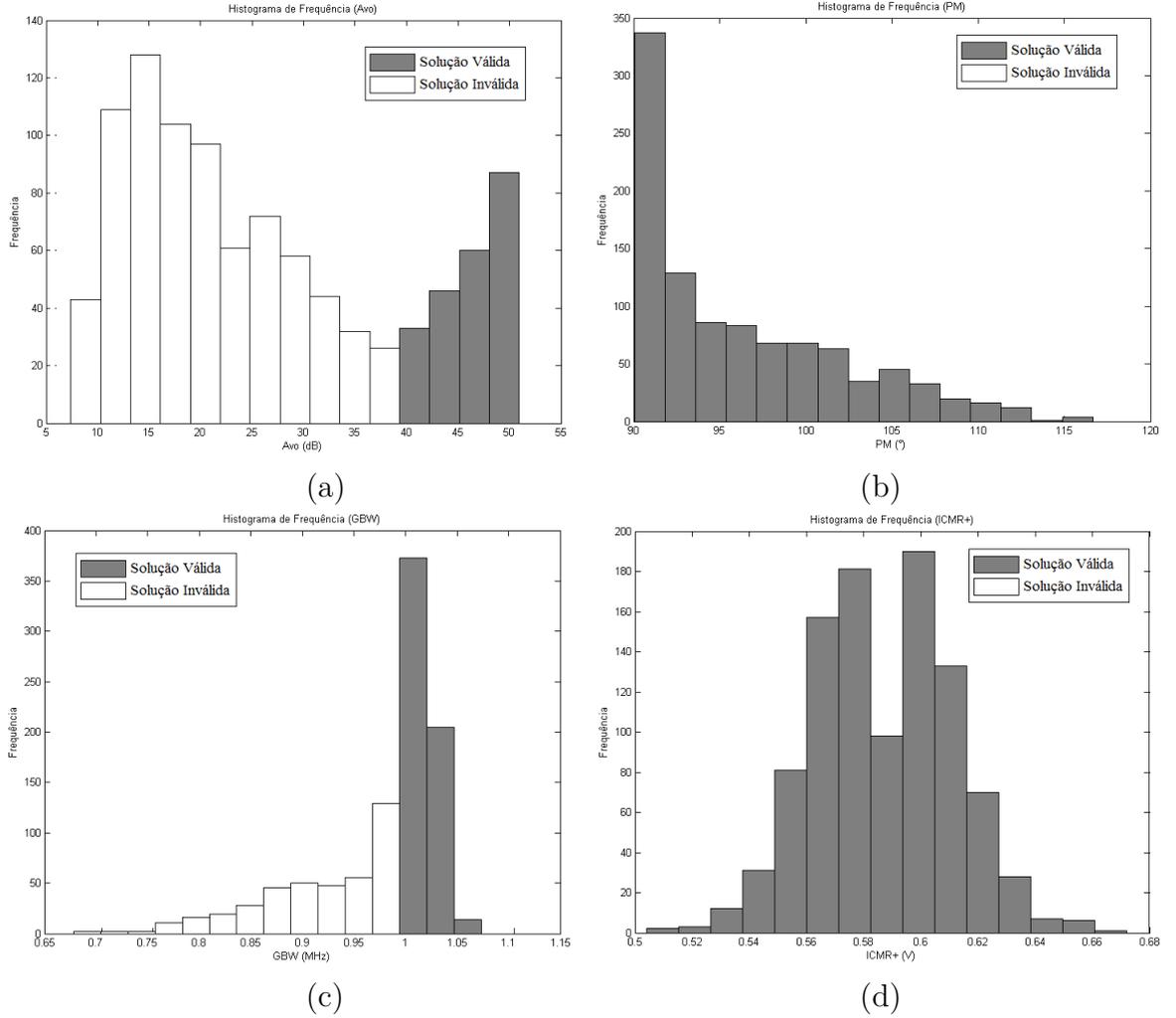
Tabela 13: Análise do resultados de variações de parâmetros do circuito amplificador diferencial através da simulação Monte Carlo

Especificação		Avo (dB)	PM ($^{\circ}$)	GBW (Hz)	ICMR+ (V)
Nominal		49,969	90,073	$1,001 \times 10^6$	0,589
Monte Carlo Variação Global <i>yield</i> = 51,8%	Média	49,969	90,073	$0,999 \times 10^6$	0,589
	Mínimo	49,513	90,059	$0,976 \times 10^6$	0,528
	Máximo	50,441	90,089	$1,031 \times 10^6$	0,648
	σ	0,189	$5,552 \times 10^{-3}$	$11,184 \times 10^3$	$23,924 \times 10^{-3}$
Monte Carlo Variação Local <i>yield</i> = 8,8%	Média	25,915	96,141	$0,971 \times 10^6$	0,588
	Mínimo	7,010	90,069	$0,675 \times 10^6$	0,570
	Máximo	50,102	116,441	$1,046 \times 10^6$	0,606
	σ	12,897	6,016	$64,596 \times 10^3$	$6,609 \times 10^{-3}$
Monte Carlo Variação Global+Local <i>yield</i> = 9,3%	Média	25,905	96,140	$0,971 \times 10^6$	0,587
	Mínimo	6,944	90,060	$0,672 \times 10^6$	0,504
	Máximo	50,488	116,658	$1,068 \times 10^6$	0,672
	σ	12,89	6,010	$56,221 \times 10^3$	$23,606 \times 10^{-3}$

pares diferenciais, utilizados no circuito amplificador diferencial, necessitem de um bom casamento (*matching*) entre os dispositivos. Como a variação considera esta variabilidade entre dois dispositivos, este casamento é afetado. Para variações locais e globais, simultaneamente, o valor do *yield* é igual a 9,3%. O aumento em relação à variação local deve-se ao fato de que a variação global pode compensar o efeito de alguma variação local (DRENNAN; MCANDREW, 2003). Em linhas gerais, pode-se afirmar que a solução ótima do problema de dimensionamento apresenta um *yield* de 9,3%, considerando o efeito das variações locais e globais, por serem mais realistas (SAXENA et al., 2008). Para esta variação, a Figura 30 mostra o histograma de frequência para as especificações do circuito amplificador diferencial. Nestes histogramas, verifica-se que as especificações de margem de fase (PM) e ICMR+ apresentam valores válidos para qualquer variação, ou seja, mesmo com variações nos parâmetros o valor requerido foi mantido. Já as especificações de ganho (Avo) e GBW apresentam grande parte das especificações classificadas como inválidas, ou seja, estão abaixo dos 40 dB mínimos requerido para o ganho e abaixo de 1 MHz requerido para o GBW. Com isso, as especificações de ganho e GBW são responsáveis por afetar o *yield* do circuito. Especificações como o ganho são muito dependentes do casamento entre o par diferencial de entrada de um amplificador diferencial. Para mostrar que o *mismatch* é extremamente importante em circuitos analógicos, a Figura 31 mostra os histogramas de frequência das especificações para a análise de Monte Carlo considerando somente variações globais. Nesta figura, percebe-se que a especificação de ganho junta-se às especificações PM e ICMR⁺ que não afetam o *yield* do circuito. Isto demonstra que a análise global não pode ser utilizada como única métrica de variação em circuitos analógicos (MUTLU; RAHMAN, 2005).

Para inserir a estratégia de *design centering* no fluxo de dimensionamento auto-

Figura 30: Histograma de frequência para a análise de Monte Carlo com variações globais e locais, simultaneamente. (a) Histograma do ganho (Avo), (b) Histograma da Margem de fase (PM), (c) Histograma do GBW e (d) Histograma do ICMR+.



mático foram propostas algumas estratégias, que são discutidas a seguir.

3.4.1 Estratégias de Design Centering Adotadas

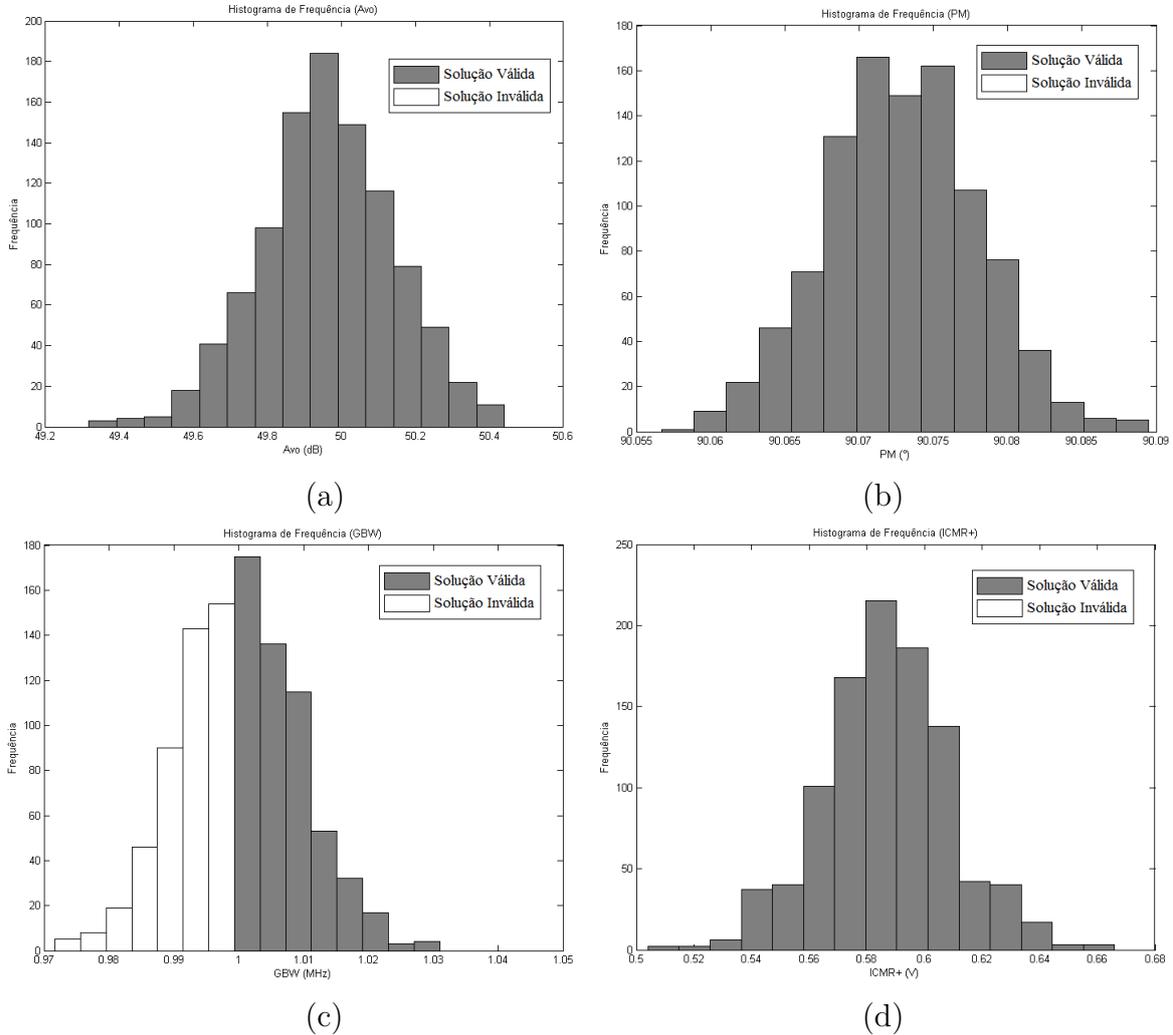
Neste trabalho foram propostas duas estratégias de *design centering* baseadas em simulação Monte Carlo. A inserção do *design centering* foi dada através da alteração da função custo ($f_c(X)$) do problema de dimensionamento, dada na Equação 3.18. A nova função custo ($f_c^*(X)$) utilizada para o *design centering* é dada na seguinte equação:

$$f_c^*(X) = f_c(X) + P_Y \cdot Y(X) \quad (3.37)$$

onde $f_c(X)$ é dada pela Equação 3.18, P_Y representa o parâmetro de ponderação do *design centering* e $Y(X)$ é dado pela seguinte equação:

$$Y(X) = 1 - yield(X) \quad (3.38)$$

Figura 31: Histograma de frequência para a análise de Monte Carlo com variação local de parâmetros. (a) Histograma do ganho (Avo), (b) Histograma da Margem de fase (PM), (c) Histograma do GBW e (d) Histograma do ICMR+.



em que $yield(X)$ assume valores entre 0 e 1 e representa a faixa de produtividade do circuito de 0% a 100%. Assim, $Y(X)$ é igual a zero para uma produtividade de 100% e é igual a 1 para uma produtividade igual a zero. Com isso, uma minimização de $Y(X)$ provoca uma otimização do $yield$.

Com o *desing centering* no projeto do amplificador diferencial simplificado, a função custo $f_c^*(X)$ passa a ser multiobjetivo, de forma que tanto área como $yield$ são objetivos de projeto. Para analisar o efeito da mudança da função custo de $f_c(X)$ para $f_c^*(X)$, o espaço de $f_c^*(X)$ foi analisado, tendo como base o espaço mostrado nas Figuras 15 e 29. A Figura 32 mostra o espaço de projeto de $f_c^*(X)$. Nesta figura verifica-se que o mínimo global de $f_c^*(X)$ é diferente do mínimo global de $f_c(X)$. Com isso, é verificado que a solução ótima com $f_c(X)$ não apresenta uma boa produtividade (cerca de 9%, como discutido anteriormente) mas está bastante próxima da solução ótima multiobjetivo.

O valor ótimo do projeto utilizando como função custo $f_c^*(X)$ é igual a 0,13658, para W e L iguais a $1,83 \mu m$ e $0,56 \mu m$, respectivamente. A solução ótima apresenta uma produtividade de 99,8%. Nos pontos em que $f_c(X)$ é mínimo ($W=1,62\mu m$ e $L=0,55 \mu m$) a função $f_c^*(X)$ apresenta um valor igual a 1,02658, que refere-se à $f_c(X)$ igual a 0,119580 acrescido de 0,907 devido à solução ter apenas 9,3% de *yield*.

Nas estratégias 1 e 2, propostas neste trabalho, o *yield* é calculado com base na simulação de Monte Carlo.

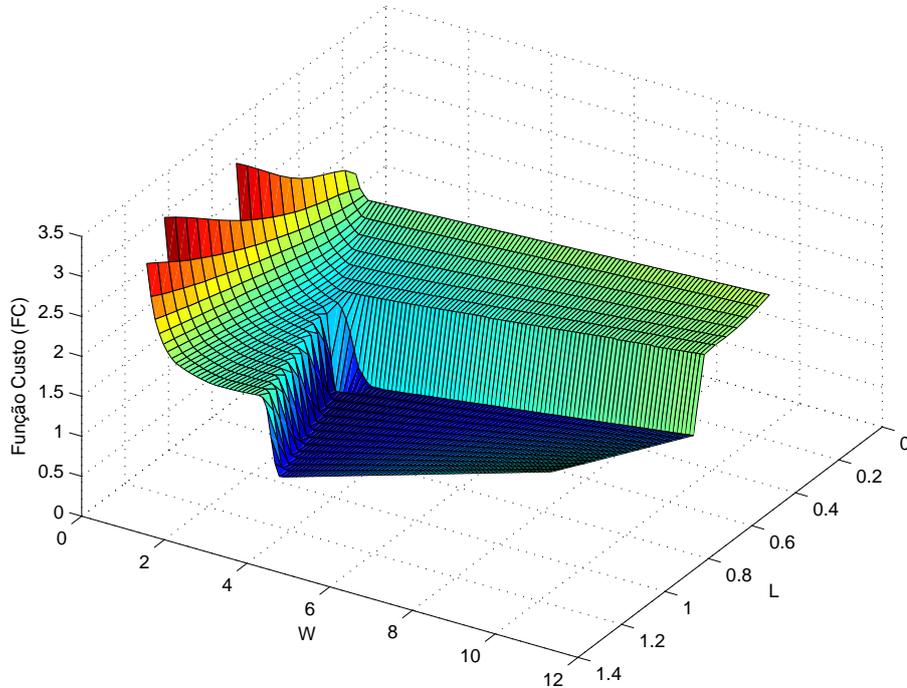
O fluxograma da estratégia 1 é mostrado na Figura 33. Neste fluxograma, o primeiro passo deve-se ao fato de verificar se as especificações nominais requeridas são atingidas. Caso estas especificações não sejam atingidas, o *yield* é tido como zero ($Y(X)$ igual a 1) e o algoritmo retorna ao cálculo da função custo ($f_c^*(X)$). Caso todas as especificações nominais sejam aceitas, a simulação de Monte Carlo é executada através de um simulador elétrico tipo SPICE. Após a finalização da simulação, o valor do *yield* é calculado (percentual de soluções que atingem as especificações requeridas) de forma que o fluxo retorne $Y(X)$ para o cálculo de $f_c^*(X)$.

Esta estratégia apresenta uma grande simplicidade, pois para cada solução gerada, que atinge as especificações requeridas, uma simulação de Monte Carlo é executada para a estimativa de *yield* seja obtido. Um dos parâmetros desta estratégia refere-se à escolha do número de rodadas da simulação de Monte Carlo. Em tese, um maior número de rodadas de simulação propicia uma melhor aproximação de medida da produtividade. Porém, o aumento do número de rodadas provoca um aumento no tempo de simulação.

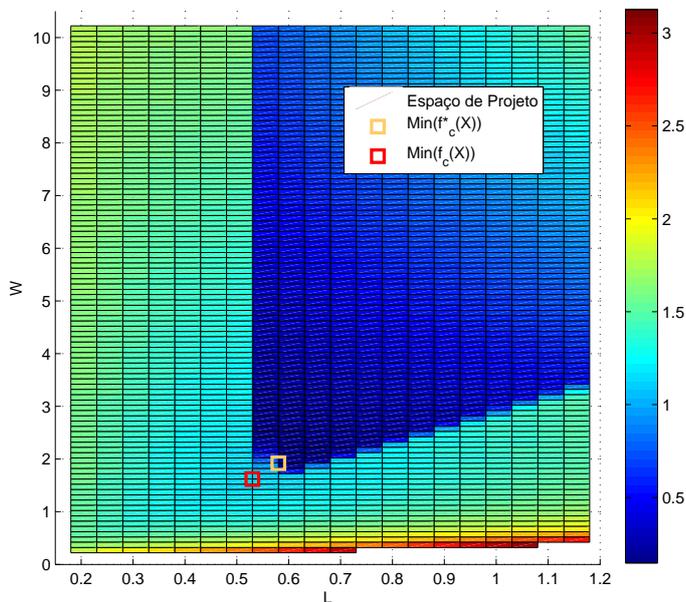
Neste trabalho, o fluxo de otimização foi executado utilizando 100 rodadas de análise de Monte Carlo a cada iteração. Ao final do processo de dimensionamento, uma simulação com 2000 rodadas é executada a fim de mostrar o resultado final com maior aproximação. O processo de síntese foi executado 12 vezes utilizando para isso a heurística dos algoritmos genéticos com as melhores configurações analisadas na seção 3.3.2. A Tabela 14 mostra os resultados obtidos para o circuito amplificador diferencial simplificado. Nestes resultados, verifica-se que a produtividade mínima encontrada foi igual a 98,6%. Assim, a produtividade final foi maximizada, uma vez que no projeto com a otimização de $f_c(X)$ (sem *design centering*) a produtividade encontrada era em torno de 9,3%. Analisando esta tabela verifica-se que somente a execução de número 2 retornou a solução ótima ($W = 1,82\mu m$ e $L = 0,56\mu m$). Este resultado foi obtido utilizando a melhor configuração de GA mostrada na Tabela 12. Além disso, algumas soluções encontradas estão bastante próximas da solução ótima, como as execuções 2 e 3. O tempo de dimensionamento ficou em torno de nove horas de execução, tempo este que é alto, visto a baixa complexidade deste projeto.

Para a redução do tempo de síntese automática utilizando a simulação de Monte Carlo para o cálculo do *yield*, a estratégia 2 é proposta. Esta estratégia é simular a

Figura 32: Espaço de Projeto do amplificador diferencial de duas variáveis para a função f_c^* . (a) espaço tridimensional e (b) espaço bidimensional com especificação da solução ótima.



(a)



(b)

estratégia 1, porém com a análise do *yield* somente se a solução atual for melhor que a melhor solução encontrada. O objetivo desta estratégia é reduzir o número de vezes que a análise de Monte Carlo é executada. No início do processo de dimensionamento

Figura 33: Fluxograma da estratégia 1 de *design centering* baseado na simulação de Monte Carlo.

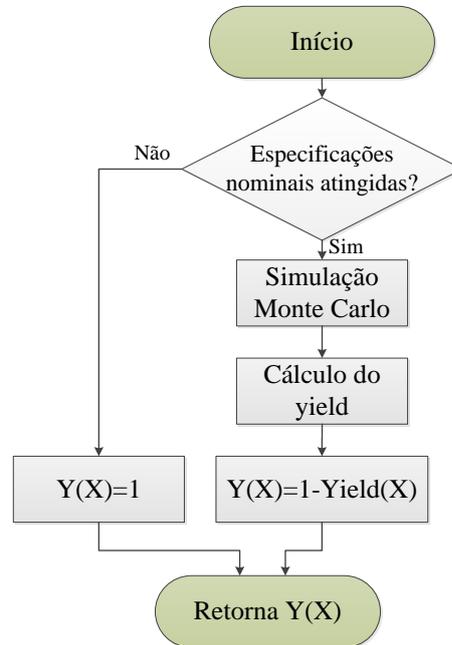


Tabela 14: Resultados de execução da estratégia 1 de *design centering*

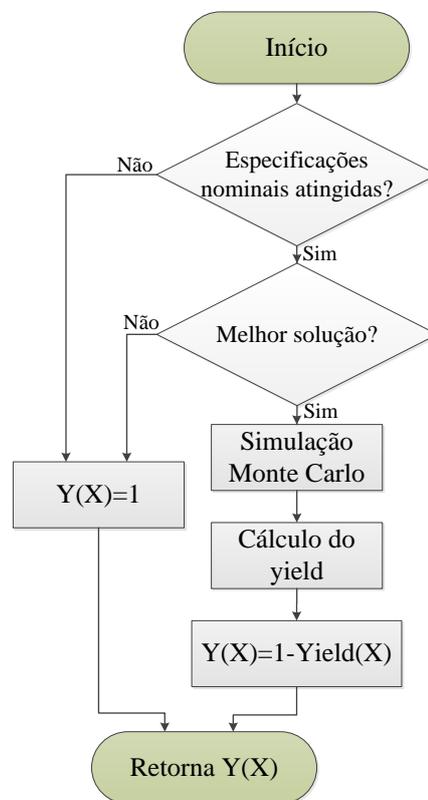
Execução	W (μm)	L (μm)	f_c^*	yield (%)	Tempo (horas)
1	1,869	0,561	0,140	99,2	17,02
2	1,824	0,562	0,137	99,8	9,22
3	1,807	0,562	0,136	90,1	9,10
4	1,858	0,567	0,141	99,9	8,77
5	1,821	0,567	0,138	99,8	8,75
6	1,847	0,565	0,139	99,8	9,03
7	1,789	0,563	0,134	98,8	9,00
8	1,941	0,553	1,433	98,9	9,45
9	1,785	0,562	1,338	98,6	8,48
10	1,785	0,562	1,338	98,6	8,86
11	1,785	0,562	1,338	98,6	8,50
12	1,785	0,562	1,338	98,6	8,40
Solução ótima	1,83	0,56	0,13658	99,8	-

automático a melhor solução é atualizada com enorme frequência, porém esta frequência de atualização é reduzida ao longo das iterações. Com isso, o número de análises de Monte Carlo é diminuída à medida que as iterações evoluem. Com a redução do número de análises, o tempo total de síntese também tende a ser reduzido, uma vez que a simulação de Monte Carlo é responsável pela maior parte do tempo de síntese. Como o valor de $f_c^*(X)$ não é conhecido antes da simulação Monte Carlo, o teste de melhor solução foi feito com base em $f_c(X)$. Neste caso, é executada a análise de Monte Carlo somente se

$f_c(X) < \min(f_c^*(X))$, sendo $\min(f_c^*(X))$ o menor valor encontrado para $f_c^*(X)$. Não é possível afirmar que a solução encontrada será melhor que a melhor solução, mas possui chances de apresentar um valor de $f_c^*(X)$ melhor.

O fluxograma desta estratégia é mostrado na Figura 34 e refere-se à inserção do teste de melhor solução ao fluxograma da 33.

Figura 34: Fluxograma da estratégia 2 de *design centering* baseado na simulação Monte Carlo.



As mesmas execuções feitas utilizando a Estratégia 1 foram repetidas utilizando a Estratégia 2. Estes resultados estão mostrados na Tabela 15, onde verifica-se que o tempo de otimização não ultrapassou 3,62 horas, tempo este que representa uma redução mínima de 56,75% em relação às execuções com a Estratégia 1. Com relação ao resultado de produtividade, o pior resultado encontrado foi de 97,1%. Nestas execuções, somente a execução de número 11 retornou a resposta ótima do problema. Desta forma, os resultados com a estratégia 2 foram similares aos obtidos com a estratégia 1, porém com uma grande redução no tempo de execução.

Tabela 15: Resultados de execução da estratégia 2 de *design centering*

Execução	W (μm)	L (μm)	f_c^*	yield (%)	Tempo (horas)
1	1,753	0,563	0,142	97,1	2,90
2	1,885	0,562	0,141	99,7	2,40
3	1,828	0,581	0,152	99,2	2,38
4	1,813	0,565	0,137	99,5	2,48
5	1,921	0,554	0,142	98,8	2,57
6	1,869	0,557	0,139	98,7	2,70
7	1,877	0,559	1,400	99,2	2,57
8	1,891	0,559	1,410	99,2	2,55
9	1,785	0,562	1,338	98,6	2,50
10	1,785	0,562	1,338	98,6	3,40
11	1,826	0,561	1,337	99,8	3,81
Solução ótima	1,83	0,56	0,13658	99,8	-

3.5 Conclusão

Neste Capítulo, o dimensionamento de um amplificador diferencial foi explorado utilizando as técnicas *Simulated Annealing* e Algoritmos Genéticos.

Com relação à técnica *Simulated Annealing*, foi verificado que os melhores resultados são encontrados com funções de geração e de decaimento de temperatura de Boltzmann, pois um maior número de soluções ótimas foram encontradas. A utilização da função de decaimento de temperatura Fast com utilização de *reannealing* também permitiu encontrar um elevado número de soluções ótimas.

A aplicação do GA possibilitou a verificação de que o número de indivíduos da população (NPOP) e o número de gerações do algoritmo (NGEN) representam as principais configurações. A utilização de cromossomos reais tende a encontrar um número maior de soluções ótimas. Com as técnicas de recombinação heurística e mutação não-uniforme são encontradas os melhores resultados.

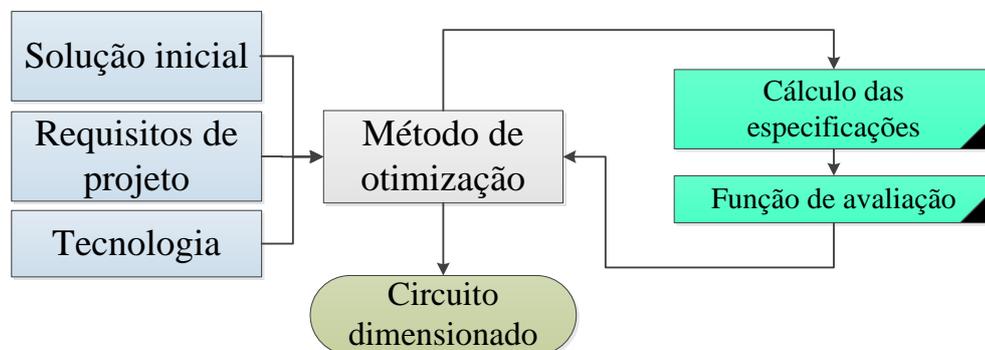
A estratégia de otimização global seguida de otimização local insere melhorias ao processo de exploração, uma vez que um maior número de soluções ótimas é encontrado.

A análise de Monte Carlo pode ser utilizado no processo de exploração do espaço de projeto de circuitos integrados analógicos utilizando *design centering*. Devido ao tempo da análise Monte Carlo ser alto, pode-se executar esta análise somente nas melhores soluções sem perder a qualidade da solução final encontrada. Nos resultados foram encontrados índices de produtividade maiores que 97,1%, sendo que, para o mesmo projeto, foi encontrada uma produtividade de apenas 9,3% sem a utilização de *design centering*.

4 A Ferramenta UCAF

A ferramenta proposta neste trabalho está baseada em uma metodologia para o dimensionamento de circuitos integrados analógicos através de otimização utilizando simulação elétrica. Esta metodologia está de acordo com a revisão bibliográfica realizada neste trabalho e segue o fluxograma mostrado na Figura 35. Nesta figura, o método de otimização recebe como entradas uma solução inicial (a qual pode ser aleatória), os requisitos de projeto e a tecnologia alvo de fabricação. Com base nestas informações, a heurística de otimização é executada de forma a atribuir valores para as variáveis do circuito e, a cada solução gerada, analisar uma função de avaliação do circuito. Para a avaliação da solução, as especificações do circuito são obtidas e, com base nestes valores, a função de avaliação é calculada. O método de otimização repete o processo até que seja encontrada uma solução otimizada que atenda às especificações de projeto.

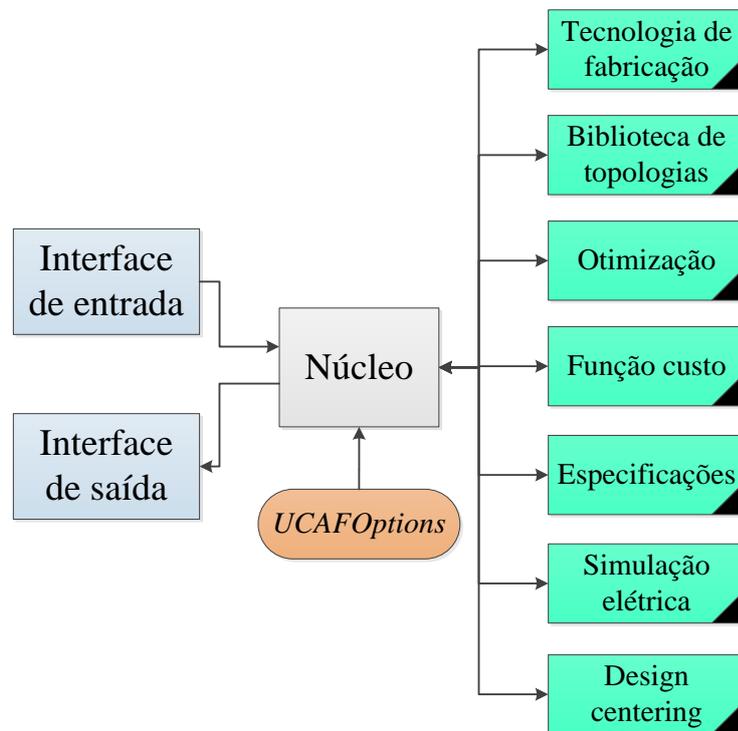
Figura 35: Fluxograma da metodologia proposta.



Neste trabalho, a metodologia mostrada na Figura 35 foi implementada através de funções modulares. A organização através de funções modulares faz com que a ferramenta possa ser configurada de diversas formas para um dado problema. Desta forma, as metodologias, as técnicas de otimização, os tipos de função custo, entre outros, podem ser comparados no projeto de uma dada topologia de circuito.

A ferramenta implementada foi chamada de UCAF e é composta por blocos funcionais modulares. Cada bloco funcional pode ser configurado para ajustar o fluxo de projeto. Os blocos que compõem a ferramenta são mostrados na Figura 36. Suas implementações foram feitas em ambiente Matlab® e serão detalhados nas seções seguintes.

Figura 36: Diagrama de blocos da ferramenta proposta.



4.1 Interfaces de Entrada e Saída

A interface de entrada é utilizada para configurar a ferramenta e inserir os dados necessários para a sua execução. A entrada é realizada por linha de comando através de um *script*. No *script*, cada um dos blocos funcionais é configurado, de forma que o fluxo de execução da ferramenta seja definido. Após a configuração, a ferramenta pode ser executada. Estas configurações são dadas através do ajuste da estrutura *UCAFOptions* que contém todos os dados da ferramenta. Esta estrutura serve como entrada do núcleo do UCAF. Um típico *script* da ferramenta proposta é mostrado no Anexo A deste trabalho.

Durante o processo de dimensionamento de um dado circuito, os dados de execução, como soluções intermediárias e valores de especificações, são armazenados. Com isso, estes dados podem ser analisados ao final da execução. Para isso, a ferramenta apresenta uma interface de saída que tem como objetivo a análise dos resultados de execução. A ferramenta apresenta como saída um relatório contendo os dados iniciais e finais de execução, bem como os valores de especificações de projeto e de produtividade. Junto ao relatório são salvos em texto os dados de execuções intermediárias para análises futuras por parte do usuário. O Anexo B mostra um típico relatório de saída da ferramenta proposta.

Além dos dados de saída, o UCAF apresenta uma interface de e-mail, de forma que o relatório de saída e os dados de execução são enviados a um e-mail configurado no início de execução. Esta interface permite que o usuário possa acompanhar o resultado de execução remotamente.

4.2 Núcleo da Ferramenta

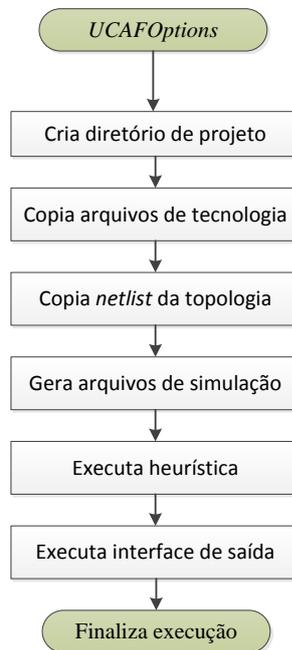
O núcleo da ferramenta é a parte funcional principal, pois este bloco tem acesso a todos os demais blocos funcionais. Este bloco é responsável por ler as informações de entrada, acionar e configurar os blocos necessários para a automação do projeto. O núcleo da ferramenta baseia-se na estrutura *UCAFOptions*, que representa as configurações ajustadas na interface de entrada. No início da síntese automática, o usuário ajusta as variáveis da estrutura *UCAFOptions* com os parâmetros desejados. Os parâmetros não ajustados recebem um valor padrão (*default*).

A Figura 37 apresenta o fluxograma de execução simplificado do núcleo da ferramenta. As primeiras partes deste fluxograma referem-se à geração e à preparação dos arquivos necessários para a execução. Primeiramente, a estrutura *UCAFOptions* é lida e um diretório de projeto é criado. Para este diretório, são copiados os arquivos de tecnologia, o *netlist* do circuito a ser dimensionado e os arquivos de simulação SPICE gerados pela ferramenta. Após isso, a heurística de otimização, principal parte da ferramenta, é executada para fazer a exploração do espaço de projeto do circuito. Ao final, a interface de saída é executada para informar ao usuário os resultados do processo de dimensionamento do circuito integrado analógico.

4.3 Tecnologia de Fabricação

O bloco de tecnologia é responsável pela inserção e configuração do modelo para os transistores CMOS, dos parâmetros do modelo e também das dimensões mínimas e máximas dos transistores. Com isso, este bloco funcional representa a entrada de dados que referem-se à tecnologia em que o circuito será fabricado. Estes dados podem ser disponibilizados diretamente pela *foundry* ou obtidos através da caracterização de transistores de testes. A escolha do modelo elétrico para os transistores é de grande importância na exploração do espaço de projeto, pois ele é responsável por aproximar, via simulações, o comportamento elétrico dos dispositivos. Outra característica quanto ao modelo escolhido é a forma como ele efetua a modelagem dos transistores, uma vez que descontinuidades ou curvas de ajustes podem afetar a busca de soluções com a heurística de otimização. Assim, modelos formados por equações contínuas como ACM (MONTORO; SCHNEIDER; CUNHA, 2007) e EKV (ENZ, 2008) tendem a funcionar melhor neste tipo de ferramenta. A

Figura 37: Fluxograma de execução do núcleo da ferramenta.



ferramenta desenvolvida é capaz de trabalhar com qualquer modelo de transistores, desde que este modelo seja compatível com o simulador elétrico utilizado.

A entrada de dados do modelo se dá através da informação do diretório em que esteja o arquivo de parâmetros do modelo a ser utilizado. Este arquivo será incluído nas simulações SPICE realizadas pela ferramenta. Caso uma dada tecnologia necessite de mais de um arquivo a ser incluído, estes arquivos devem estar no diretório especificado. Para isso as variáveis *UCAF.Technology.Dir* e *UCAF.Technology.FolderName* devem ser ajustadas com o diretório e o nome da pasta que contém os arquivos de modelo, respectivamente. Neste diretório devem estar também os modelos de Monte Carlo, caso seja utilizado *design centering* no processo de dimensionamento.

4.4 Biblioteca de Circuitos

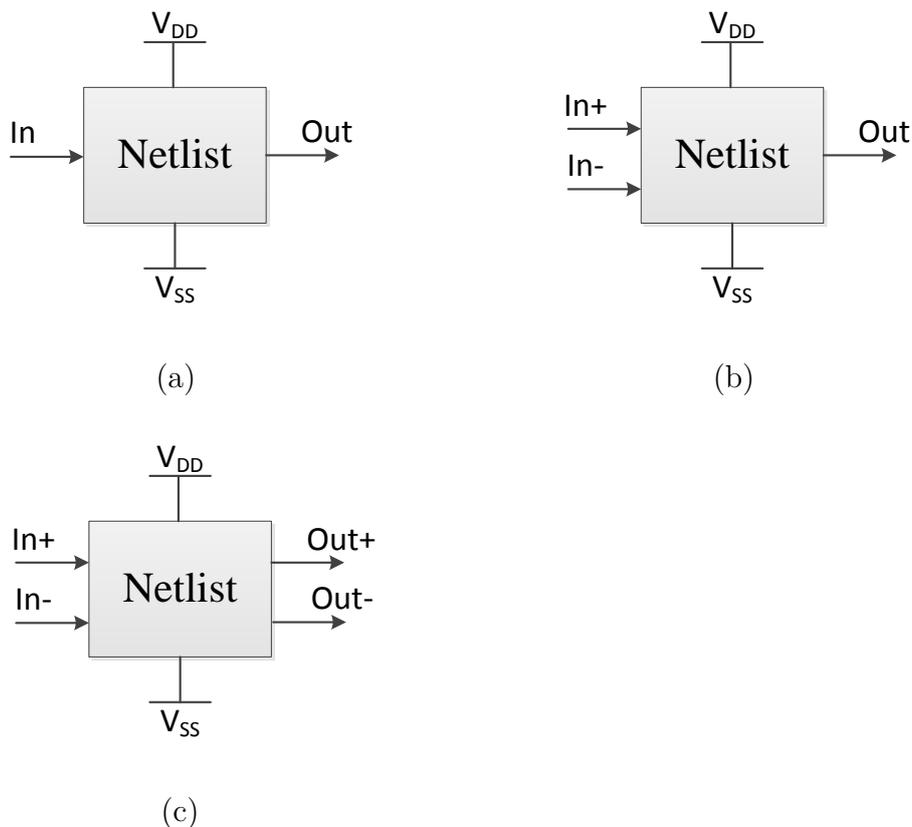
A biblioteca de circuitos refere-se aos circuitos integrados analógicos que se deseja projetar automaticamente, de forma a encontrar uma solução (dimensão dos transistores e fontes de polarização) otimizada que satisfaça às especificações mínimas requeridas no projeto.

A forma de implementação do UCAF é compatível com qualquer circuito integrado que se deseje dimensionar. O que restringe a execução de um circuito é o poder de busca das heurísticas de otimização dentro do espaço não-linear de projeto. Em uma topologia,

entende-se por variáveis livres de projeto o número de incógnitas que se deve conhecer para que o circuito seja totalmente projetado. Nesta versão do UCAF o usuário pode inserir qualquer circuito através do seu *netlist* e da informação das variáveis livres de projeto.

Para inserção de topologias foram definidas três formas de circuitos, conforme mostrado na Figura 38. A forma do circuito é caracterizada pelo tipo de *netlist* a ser dimensionado. O primeiro caso refere-se a circuitos com entrada e saída em terminação simples (*single-ended*) como, por exemplo, um amplificador de fonte comum (ALLEN; HOLBERG, 2002). Já a segunda e terceira forma referem-se a circuitos com entradas diferenciais e saídas em terminação simples ou diferencial, respectivamente. Estes circuitos são comumente topologias de amplificadores operacionais.

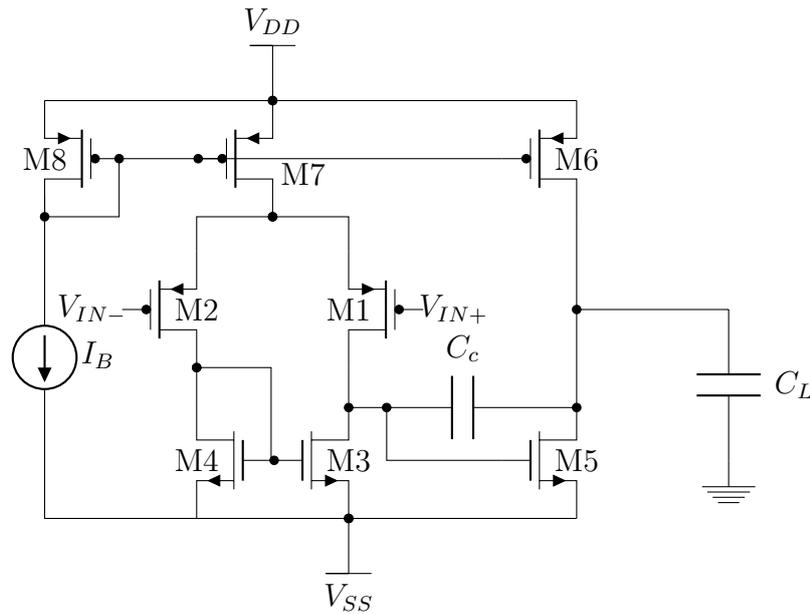
Figura 38: Tipos de topologias de circuitos presentes na ferramenta. (a) circuitos de entrada e saída em terminação simples (*single-ended*), (b) circuitos de entrada diferencial e saída em terminação simples e (c) circuitos de entrada e saída diferencial (*fully differential*).



A inserção de um circuito se dá através de um *netlist* definido como subcircuito. Para a definição do subcircuito é utilizado o padrão do bloco, dado pelas entradas, saídas e fontes de alimentação. A Figura 39 mostra um exemplo de inserção de topologia. Na parte (a) é mostrado o esquemático de um circuito amplificador de transcondutância (OTA) do tipo Miller, que será detalhado na seção 5.1. Para este esquemático é representado o

netlist na parte (b). No *netlist*, o subcircuito é definido na linha 3, através das entradas *vp* e *vn*, saída *vout* e as fontes de alimentação *vdd* e *vss*. Este circuito apresenta 11 variáveis de projeto, definidas como *W1*, *L1*, *W3*, *L3*, *W5*, *L5*, *W6*, *L6*, *W7*, *L7* e *IB*. Os valores para estas variáveis são gerados pelo bloco de otimização. Na linha 2 deste *netlist* é especificado o arquivo de tecnologia.

Figura 39: Inserção de uma topologia de circuito através do seu *netlist*. (a) esquemático de um OTA Miller CMOS e (b) *netlist* referente ao esquemático mostrado em (a).



(a)

```

1.
2. *NETLIST OTA MILLER
3. .lib 'xh018/lp3mos/xh018.lib' tm
4. .subckt OTA_miller vp vn vdd vss vout
5. X1 2 vn 3 3 ne w='W1*1e-6' l='L1*1e-6'
6. X2 5 vp 3 3 ne w='W1*1e-6' l='L1*1e-6'
7. X3 2 2 vdd vdd pe w='W3*1e-6' l='L3*1e-6'
8. X4 5 2 vdd vdd pe w='W3*1e-6' l='L3*1e-6'
9. X5 3 4 vss vss ne w='W5*1e-6' l='L5*1e-6'
10. X6 vout 5 vdd vdd pe w='W6*1e-6' l='L6*1e-6'
11. X7 vout 4 vss vss ne w='W7*1e-6' l='L7*1e-6'
12. X8 4 4 vss vss ne w='W5*1e-6' l='L5*1e-6'
13. Ibias vdd 4 'IB*1e-6'
14. Cc 5 vout 0.3p IC=0
15. Cl vout 0 3p IC=0
16. .ends

```

(b)

A parte do *script* de execução da ferramenta que refere-se à configuração da topologia de circuito a ser projetada está mostrada na Figura 40. Neste *script*, a primeira configuração refere-se ao tipo de circuito, se será utilizado o circuito de uma biblioteca

ou se será inserido um novo circuito na ferramenta. No primeiro caso, deve-se informar o arquivo da biblioteca e o nome da topologia a ser utilizada. No segundo caso, deve-se informar o diretório que encontra-se a topologia de circuito e o nome do arquivo de *netlist* utilizado. Na definição de nome são necessários dois arquivos, um para simulação típica e outro para a simulação Monte Carlo. Esta divisão permite com que os modelos e configurações de simulações possam ser definidas em *netlists* separados. Após, deve-se definir o tipo de entrada e saída que o circuito é formado e a tensão de alimentação do bloco.

Figura 40: *Script* necessário para a configuração da topologia de circuito a ser dimensionado.

```

1.  UCAF.Circuit.Use=0; %1-Usar biblioteca 0-inserir diretamente
2.  UCAF.Circuit.Lib.LibFile='mylib'; %Library file
3.  UCAF.Circuit.Lib.Topology='mylib'; %LibName
4.  UCAF.Circuit.User.Dir='D:\UCAF\UCAF_2_0\UCAF \Projetos\Projeto_OTA_Miller\';
5.  UCAF.Circuit.User.Netlist='Miller_Netlist.txt';
6.  UCAF.Circuit.User.NetlistMC='Miller_NetlistMC.txt';
7.  UCAF.Circuit.User.NetlistScktName='OTA_miller';
8.  UCAF.Circuit.User.NetlistType='1';%1-DIn_SOut, 2-DIn_DOut, 3-SIn_SOut
9.  UCAF.Circuit.User.VDD=0.9; %VDD=-VSS

```

4.5 Otimização

O bloco de otimização é um dos principais blocos da ferramenta, visto que é este bloco que gera cada uma das possíveis soluções para a topologia de circuito. É devido à sua heurística associada que soluções otimizadas podem ser encontradas. De maneira geral, o método de otimização (heurística) recebe uma ou mais soluções iniciais e, com base nestas soluções, são geradas novas soluções para as variáveis livres de otimização. As soluções geradas devem possuir valores definidos dentro de um limite mínimo e máximo e também estarem dentro de um passo ou variação mínima para considerar duas soluções diferentes. As heurísticas avaliam as soluções com base no valor de função custo ou função de avaliação, cujo valor depende de algumas características elétricas (especificações) do circuito que sejam de interesse no projeto. Na ferramenta proposta o valor de função custo é calculado utilizando uma função custo multi-objetivo (conforme será descrito na seção 4.6).

Neste projeto foram inseridas as heurísticas dos Algoritmos Genéticos (GA) e Simulated Annealing (SA) na ferramenta implementada. Além disso, a ferramenta possui interface para otimização local com a função *fmincon* do Matlab®. Esta interface, permite um refinamento na solução final, pois, conforme visto na seção 3.3, a otimização global seguida de local fornece uma melhoria no processo de otimização. A seleção da

heurística de otimização é escolhida pela variável *UCAF.Optimization.Heuristic*, definida como 1 para GA e 0 para SA. Estas heurísticas tiveram o seu funcionamento e características analisadas na seção 3.3. As subseções seguintes apresentam os parâmetros que podem ser ajustados na execução do *script*.

4.5.1 Algoritmos Genéticos (GA)

Como o objetivo deste trabalho não é a implementação da meta-heurística e sim a sua aplicação no projeto de circuitos integrados analógicos, a otimização através de algoritmos genéticos foi implementada pelo toolbox para Matlab® GAOT (do inglês, Genetic Algorithm Optimization Toolbox) desenvolvido por Houck, Joines e Kay (1995). O GAOT possui inúmeras configurações disponíveis ao usuário e pode ser utilizado de diversas formas. Dentre as configurações, destacam-se as funções de recombinação, mutação e seleção que podem ser aplicadas de diversas maneiras em uma execução. Outros parâmetros importantes são a definição do tipo de cromossomo (que, nesta aplicação, pode ser do tipo real ou binário), tamanho da população e o número de gerações a ser executado.

A parte (a) da Figura 41 ilustra as variáveis do *script* que devem ser ajustadas para a configuração do GA. As primeiras configurações referem-se à escolha da população e à definição da precisão. É possível inserir uma população previamente gerada ou utilizar uma população gerada aleatoriamente. A configuração da precisão é muito importante, pois esta define o número de soluções possíveis do problema e também, em caso de cromossomos binário, o número de bits da representação. As demais configurações referem-se às configurações do GAOT, e podem ser consultadas através de Houck, Joines e Kay (1995).

4.5.2 Simulated Annealing (SA)

Para a implementação da ferramenta, foi utilizada a função de *Simulated Annealing* disponível no *Global Optimization Toolbox* do Matlab®(MATHWORKS, 2012b). Esta função permite a configuração de vários parâmetros do algoritmo. Entre eles, destacam-se a função de decaimento de temperatura e a função de geração de novos estados.

A parte (b) da Figura 41 ilustra as variáveis necessárias para a configuração do *script* de SA. A primeira parte refere-se à solução inicial, pois esta pode ser especificada pelo usuário ou gerada aleatoriamente pela ferramenta. As configurações do SA são feitas pela função *saoptimset*, cujas opções são mostradas em MathWorks (2012c).

Figura 41: *Script* necessário para a configuração da heurística de otimização. (a) algoritmos genéticos e (b) *Simulated Annealing*.

```

1. %Genetic Algorithms Options
2. addpath('gaot');
3. UCAF.Optimization.GA.PopInitOrRandom=0; %1-User 0-Random
4. UCAF.Optimization.GA.PopInit=[]; %Matrix with the population
5. UCAF.Optimization.GA.XVar=0.01; %Solution Resolution
6. UCAF.Optimization.GA.a=1;
7. %GAOT Settings:
8.   UCAF.Optimization.GA.CromType=0; %chromossome: 0- Bin or 1- Float
9.   UCAF.Optimization.GA.N=100; %Pop size
10. % Crossover Operators:
11.   UCAF.Optimization.GA.xFns = ['simpleXover'];
12.   UCAF.Optimization.GA.xOpts = [.4];
13. % Mutation Operators:
14.   UCAF.Optimization.GA.mFns = ['binaryMutation'];
15.   UCAF.Optimization.GA.mOpts = [0.05];
16. % Selection Function:
17.   UCAF.Optimization.GA.selectFn = ['roulette'];
18.   UCAF.Optimization.GA.selectOps = [100];

```

(a)

```

1. %Simulated Annealing Options
2. UCAF.Optimization.SA.XInitOrRandom=0; %1-User 0-Random
3. UCAF.Optimization.SA.XInit=[];
4. UCAF.Optimization.SA.OPTIONS = saoptimset('AnnealingFcn', @annealingboltz,
5.   ...'TemperatureFcn', @temperatureboltz,'ReannealInterval', inf);

```

(b)

4.6 Função Custo

A função custo é a especificação mais importante de uma otimização, pois é com base no seu valor que cada uma das soluções é avaliada. Como na ferramenta proposta o usuário deve informar as especificações que são alvos de otimização e as especificações que restringem o espaço de projeto, a função custo deve levar em conta todas as especificações selecionadas pelo usuário.

Desta forma, neste trabalho é utilizada uma função custo multiobjetivo de forma que, na avaliação de uma solução, sejam avaliadas as restrições de projeto e também as especificações que se deseja maximizar ou minimizar.

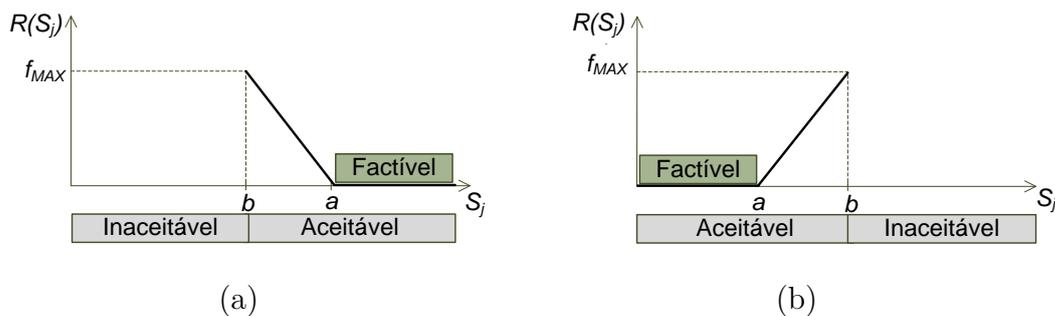
A função custo utilizada no UCAF é similar à utilizada em Phelps et al. (2000) dada pela Equação 4.1. Nesta equação, $S_i(X)$ é a especificação i do circuito, PO_i e PR_j são os pesos (fator de ponderação) das especificações objetivos e das especificações restritivas, respectivamente. Para escolher quais especificações são objetivos de otimização

e quais especificações são restrições, a ferramenta recebe a configuração de cada uma das especificações, conforme será visto na próxima seção. Os pesos PO_i e PR_j devem possuir valores positivos e diferentes de zero e são escolhidos experimentalmente.

$$f_c(X) = \sum_{i=1}^N PO_i \cdot S_i(X) + \sum_{j=1}^M PR_j \cdot R(S_j(X), S_{jref}) \quad (4.1)$$

Com relação ao valor de restrição da função custo ($R(S_j(X), S_{jref})$), o seu valor é calculado em função do valor desejado (S_{jref}) e do valor alcançado ($S_j(X)$) de uma especificação. Com isso se a especificação $S_j(X)$ está dentro do limite de restrição, $R(S_j(X), S_{jref})$ deve ser igual a zero. Caso contrário, se $S_j(X)$ está fora do valor mínimo ou máximo requerido, $R(S_j(X), S_{jref})$ deve ser proporcional à distância do valor requerido. Assim, para o cálculo de $R(S_j(X), S_{jref})$ foram utilizadas as funções mostradas na Figura 42 de acordo com Barros, Guilherme e Horta (2010). Nesta figura, o valor de a é dado pelo valor mínimo ou máximo requerido (S_{jref}) para a especificação e o valor b representa o valor limite de aceitação da solução, o qual deve ser especificado pelo usuário.

Figura 42: Função $R(S_j)$. (a) Função para especificação que requer valor superior ao valor mínimo e (b) Função para especificação que requer valor inferior ao valor máximo.



Como entrada da ferramenta, o bloco de função custo utiliza os dados de entrada do bloco especificação. Através do valor dos pesos, as especificações relevantes ao projeto são escolhidas e podem ser ponderadas. A escolha dos valores de ponderação utilizados na função custo é difícil de ser feita, pois existem inúmeras combinações possíveis e, também, muitas vezes, a faixa de variação das especificações não é conhecida. Além disso, muitas vezes o processo de otimização se detém em encontrar soluções que atinjam as especificações impostas (soluções factíveis) do que otimizar as grandezas alvo de otimização. Assim, o procedimento de escolha dos parâmetros de ponderação é experimental, no qual o usuário ajusta estes valores até as especificações serem atingidas.

4.7 Especificações

As especificações de uma topologia de circuito referem-se ao comportamento do circuito desejado em um projeto. Estes valores, na ferramenta projetada, servem de base para o dimensionamento dos transistores presentes na topologia de circuito. No UCAF deve-se trabalhar com duas formas de especificações: especificações restritivas e especificações de otimização (objetivos). As especificações restritivas são as especificações que devem satisfazer valores mínimos ou máximos definidos pelo usuário. Uma solução só é factível (*feasible solution*) se estes valores mínimos ou máximos forem atingidos. As especificações de otimização são escolhidas pelo usuário como características do projeto que se deseja otimizar, ou seja, estas especificações serão os objetivos da função custo do problema de otimização.

Os valores das especificações utilizados durante o fluxo de otimização da ferramenta, com exceção da área de gate que é calculada diretamente como a relação entre as dimensões do dispositivo, são obtidos através de simulação elétrica SPICE. Para isso, a cada especificação ou conjunto de especificações são necessários *testbenches* de simulação. Estes *testbenches* dependem do tipo de circuito e de suas características. Nesta versão da ferramenta os *testbenches* disponíveis refere-se a circuitos amplificadores operacionais de entrada diferencial e saída em terminação comum. Foram definidos seis conjuntos de especificações, dados pela seguinte nomenclatura:

- Análise no domínio da frequência em malha aberta (SP001AC_type1): com este *testbench* são obtidas as especificações Avo (ganho em baixas frequências), GBW (produto ganho-largura de faixa) e PM (margem de fase), através de simulação AC (análise no domínio da frequência).
- Velocidade de resposta (SP001TR_type1): obtenção da especificação SR (*slew rate*), como base em simulação transiente (domínio do tempo).
- Tensão de entrada em modo comum (SP001DC_type1): neste *testbench* a faixa de tensão no modo comum de entrada (ICMR) é medida com simulação DC (variação linear no nível de tensão ou corrente).
- Faixa de tensão saída (SP002DC_type1): simulação da faixa de tensão de saída do circuito durante a sua operação, obtida com simulação DC.
- Potência dissipada (SP001PowerBias): especificação de potência dissipada pelo circuito.
- Área de gate (SP001GateArea): especificação de área de gate, dada pela multiplicação da largura pelo comprimento do canal de cada transistor.

Cada especificação apresenta uma sintaxe de configuração, dada como:

$$SP = [Valor Tipo Aceitavel Ponderacao] \quad (4.2)$$

em que *Valor* é o valor requerido para restrições ou valor de referência para objetivos. *Tipo* refere-se ao tipo de especificação no qual é utilizado 1 ou 2 para restrições de mínimo ou máximo valor requerido, respectivamente. Pode-se utilizar 3 ou 4 para definir a especificação como objetivo de minimização ou de maximização. O valor *Aceitável* refere-se a um valor de aceitação de especificação. Por exemplo, é requerido um valor de GBW maior do que 1 MHz, porém são aceitáveis soluções com GBW maior que 0,1 MHz. Soluções inferiores a este valor são descartadas (soluções inaceitáveis). O parâmetros *Valor* e *Aceitável* referem-se aos parâmetros *a* e *b*, respectivamente, utilizados para o cálculo das $R(S_j(X), S_{jref})$ da função custo (*fc*), mostrados na Figura 42. O parâmetro *Ponderação* refere-se aos valores PO_i ou PR_j da função custo.

Com exceção do último tipo de especificação, os demais são obtidos com simulação elétrica. A especificação de área de gate necessita da forma com que o valor da área de gate do circuito é calculada. Para isso devem ser especificadas as variáveis que representam W e L de cada transistor, bem como um fator multiplicativo. A área de gate calculada é obtida por:

$$A = VAR_i.VAR_j.MFAT_z + VAR_k.VAR_t.MFAT_y + \dots \quad (4.3)$$

Um exemplo de configuração das especificações da ferramenta é mostrado na Figura 43. Nesta figura, primeiramente o nome das variáveis foi configurado, neste caso como W1, L1, W3, L3 e IB. Após isso, o intervalo de variação de cada variável deve ser configurado, neste caso W entre 0,22 e 100, L entre 0,18 e 50 e IB entre 0,1 e 200. Como especificações foram selecionados cinco grupos de especificações. O grupo de especificações SP001AC_type1 tem suas especificações configuradas como restrições. Como exemplo o ganho em baixas frequências foi configurado com valor mínimo requerido igual a 70 dB e a aceitação de soluções está condicionada a valores superiores a 10 dB. O Avo apresenta um valor de ponderação igual a 1. Da mesma forma, as demais especificações são configuradas, sendo que somente a área de gate foi configurada como objetivo.

4.8 Simulação Elétrica

Na parte de simulação elétrica da ferramenta estão cadastrados os simuladores elétricos tipo SPICE e os *testbenches* de simulação. Estes simuladores são utilizados para a avaliação das especificações dos circuitos através da simulação de um arquivo de texto que contém o circuito descrito. Para a medição das especificações utilizam-se *testbenches* de circuito, de forma que, com a utilização das análises AC, DC e transiente, todas as especificações do circuito sejam obtidas através da simulação elétrica.

Figura 43: *Script* de configuração das especificações.

```

1. UCAF.Specifications.VarNames=['W1','L1','W3','L3','IB'];
2. UCAF.Specifications.VarBounds=[0.22 100;0.18 50;0.22 100;0.18 50;0.1 200];
3. UCAF.Specifications.SpecMeasure.S=['SP001AC_type1','SP001TR_type1';
4.   ...'SP001DC_type1','SP002DC_type1','SP001GateArea'];
5. UCAF.Specifications.SpecMeasure.Required.SP001AC_type1=[
6.     70 1 10 1;%Avo
7.     5e6 1 1e3 1;%GBW
8.     60 1 20 1;%MF
9. UCAF.Specifications.SpecMeasure.Required.SP001AC_type1_MF=0;
10.   %1 - Phase Margin=|Phase+180| 0 - Phase Margin=|Phase-180|
11. UCAF.Specifications.SpecMeasure.Required.SP001TR_type1=[5 1 0.1 1.5]; %SR
12. UCAF.Specifications.SpecMeasure.Required.SP001DC_type1=[
13.     0.4 1 0 1;%ICMR+
14.     -0.4 1 0 0];%ICMR-
15. UCAF.Specifications.SpecMeasure.Required.SP002DC_type1=[
16.     0.6 1 0 1;%OS+
17.     -0.6 1 0 1];%OS-
18. UCAF.Specifications.SpecMeasure.Required.SP001GateArea.S=[0 4 30 1];
19. UCAF.Specifications.SpecMeasure.Required.SP001GateArea.Eq=[
20.     1 2 2; %var1*var2*multiple
21.     3 4 3];

```

A escolha do simulador elétrico muitas vezes está relacionada aos parâmetros do modelo que se tem disponível e também com relação à convergência de determinadas simulações. A ferramenta utiliza o simulador através da execução por linha de comando, onde são passados os dados de simulação através do circuito descrito. Após a simulação são gerados, em texto, os dados de resultados da simulação. Destes dados são extraídas as especificações do circuito. Como um circuito possui inúmeras especificações que devem ser mensuradas, são necessárias várias simulações para a análise de uma solução. Com isso, o maior tempo gasto na ferramenta se dá na tarefa de simulação elétrica. Nesta versão da ferramenta, foi utilizado o simulador elétrico Synopsys HSPICE®.

Como visto na seção anterior, foram implementados seis *testbenches* de medição de especificações. Destes seis, cinco são obtidos através da simulação elétrica. As subseções seguintes mostram detalhes sobre os *testbenches* de medição implementados com base em Allen e Holberg (2002).

4.8.1 Análise no domínio da frequência em malha aberta (SP001AC_type1) e potência dissipada (SP001PowerBias)

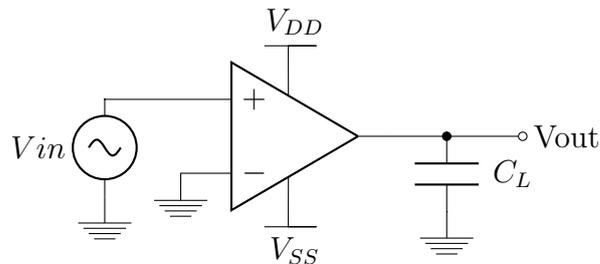
Este *testbench* de simulação tem por objetivo a medição das características de malha aberta de um amplificador operacional. Para isso, o circuito mostrado na Figura

44 é utilizado. Neste circuito a fonte de tensão de entrada senoidal tem a sua frequência variada em uma faixa de análise, calculando-se a tensão de saída no domínio da frequência. Assumindo que a amplitude do sinal senoidal de entrada seja igual à unidade, então o ganho do circuito é igual ao sinal de tensão de saída.

Uma resposta de simulação elétrica deste circuito é mostrada na Figura 45, cujo gráfico relaciona o ganho do circuito em decibéis e a fase em graus. Este diagrama é conhecido como diagrama de Bode. Na curva de ganho para baixas frequências é extraída a especificação *Avo*, neste caso igual a 50 dB. Nesta mesma curva, para ganho unitário (0 dB) a frequência refere-se à especificação Produto Ganho-Largura de faixa (*GBW*), que neste caso é igual a 9,7 MHz. Para o valor de frequência igual ao *GBW*, analisando o gráfico de fase obtêm-se a fase para uma frequência igual à *GBW*. A especificação de margem de fase (*PM*) se refere a $180^\circ - \textit{fase}$, resultando em uma margem de fase de 49° . A margem de fase do circuito é sempre referenciada a 180° . Desta forma deve-se especificar se o valor de *PM* será calculado como $|\textit{fase} + 180|$ ou $|\textit{fase} - 180|$, como configurado na linha 10 do *script* mostrado na Figura 43.

Para o mesmo circuito utilizado na medição das características de malha aberta, através de uma análise do ponto de operação (*.OP*), pode-se obter o valor da potência dissipada na polarização do circuito.

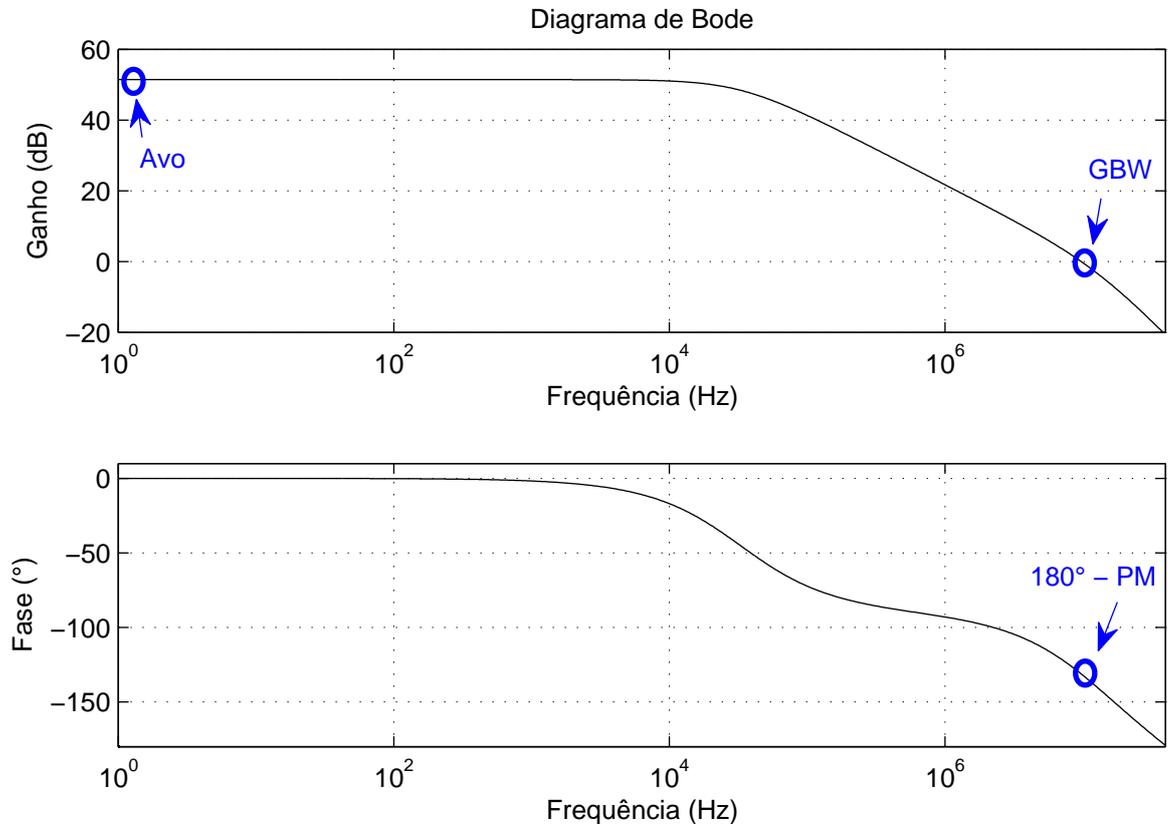
Figura 44: *Testbench* para medição das características AC em malha aberta.



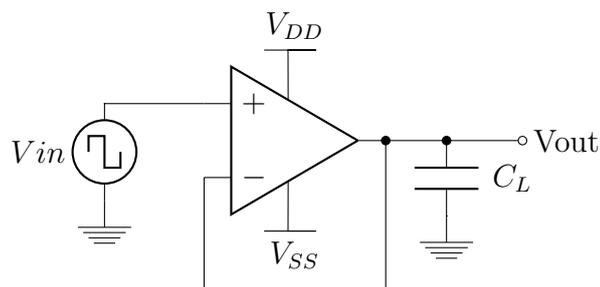
4.8.2 Velocidade de resposta (SP001TR_type1)

A medição da velocidade de resposta do amplificador operacional (*slew rate*) é feita através de uma análise transiente (*.TRAN*). Para isso, pode-se utilizar o circuito mostrado na Figura 46. Este circuito trata-se de uma configuração em ganho unitário. O objetivo da medição é verificar a taxa de alteração do nível de saída do amplificador operacional. Para isso, uma fonte com forma de onda quadrada é inserida na entrada do circuito. De acordo com a transição do nível de tensão de saída, o *slew rate* é medido.

A Figura 47 mostra um exemplo de sinal de saída para um pulso de entrada. Neste caso, tem-se duas medidas de *slew rate*, na subida e na descida. Normalmente, o *SR* é

Figura 45: Diagrama de Bode gerado com a simulação do *testbench* da Figura 44.

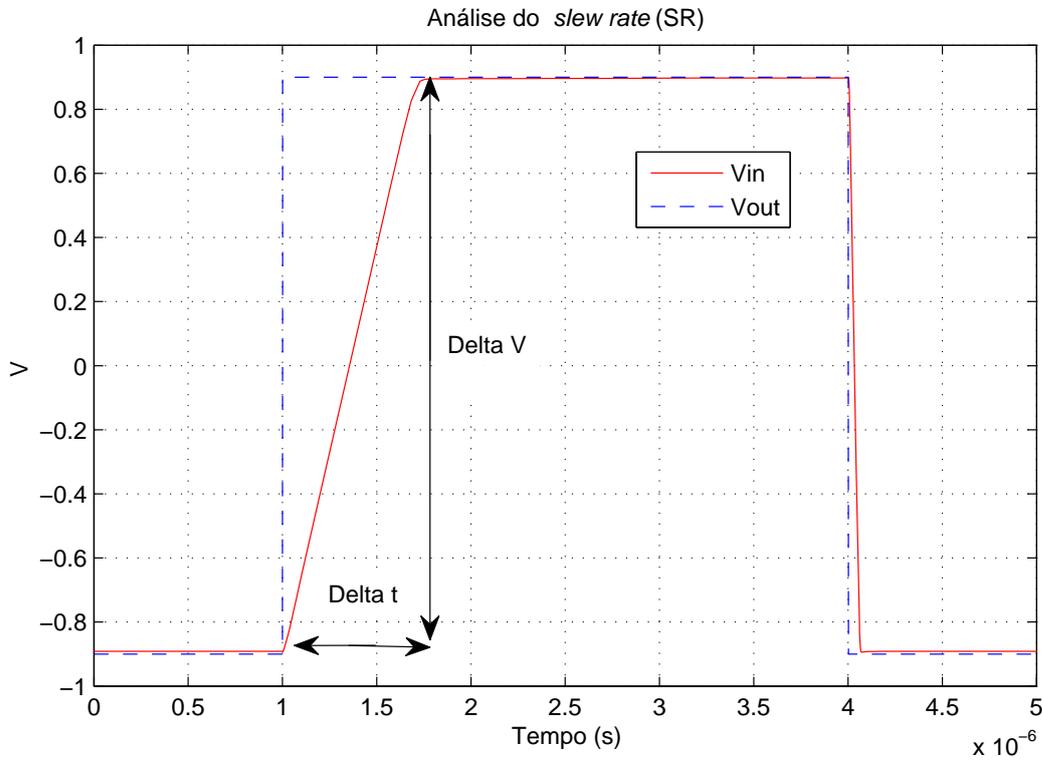
assumido como o menor valor entre a taxa de subida e descida. No caso desta figura, o SR de subida é menor que o de descida e seu valor é aproximadamente igual a $2,5 \text{ V}/\mu\text{s}$.

Figura 46: *Testbench* para medição do *slew rate* (SR).

4.8.3 Tensão de entrada em modo comum (SP001DC_type1)

A medição da faixa de tensão de entrada em modo comum ($ICMR$), pode ser obtida com a simulação do circuito da Figura 48, que trata-se de um amplificador de ganho unitário. Para esta simulação o valor DC da fonte de entrada é variado em uma faixa de valores e a linearidade do sinal de saída é observada. Os limites da linearidade

Figura 47: Sinal de tensão de entrada e saída gerado com a simulação do *testbench* da Figura 46 para a medição do *slew rate* (*SR*).



são assumidos como valores mínimos ou máximos de $ICMR$, representados por $ICMR^-$ e $ICMR^+$, respectivamente.

A Figura 49 mostra um exemplo de sinal de simulação desta configuração de circuito. Neste caso é visto que a linearidade é mantida desde o início (entrada igual a $-0,9\text{ V}$) até um valor de entrada de aproximadamente $0,70\text{ V}$. Assim, o valor de $ICMR^-$ é $-0,9\text{ V}$ e $ICMR^+$ é $0,7\text{ V}$, de forma que a faixa de entrada em modo comum seja de aproximadamente $1,6\text{ V}$.

Figura 48: *Testbench* para medição da faixa de tensão de entrada em modo comum ($ICMR$).

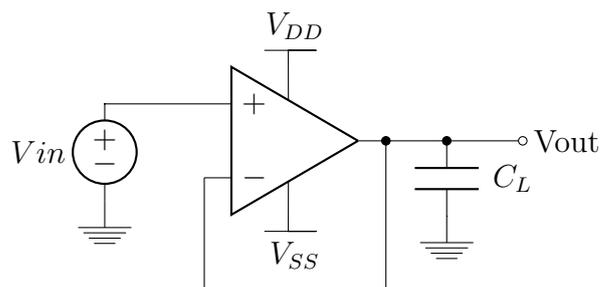
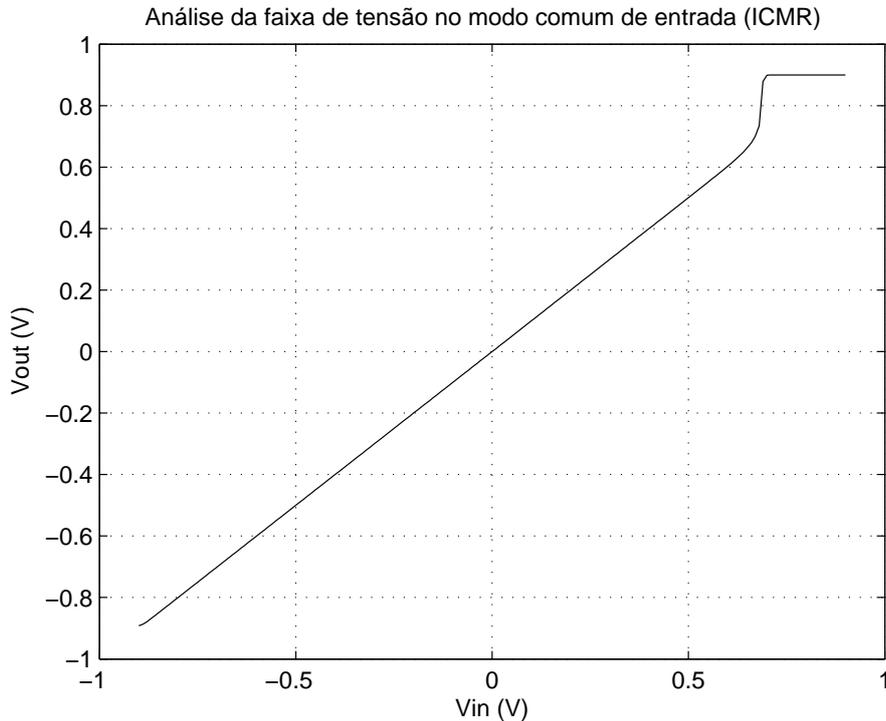


Figura 49: Sinal de tensão de saída gerado com a simulação do *testbench* da Figura 48.

4.8.4 Faixa de tensão de saída (SP002DC_type1)

A faixa de tensão de saída *output swing* (*OS*) pode ser obtida com a simulação de uma configuração de amplificação, de forma que seja possível a saturação do sinal de saída. Com base na faixa linear e de saturação do sinal de saída, os valores máximos e mínimos da excursão do sinal de saída são obtidos. A configuração utilizada na ferramenta para a medição do *OS* é mostrada na Figura 50. Este circuito é um amplificador inversor de ganho igual a -10 .

A Figura 51 mostra um gráfico resultado da simulação do circuito da Figura 50. Nesta figura, pode-se verificar que a faixa linear de inclinação igual a -10 é mantida para entradas entre $-0,3\text{ V}$ e $0,9\text{ V}$. Após isso, ocorre um comportamento não-linear do circuito. Nesta figura, a tensão de saída variou na faixa de aproximadamente $-0,9\text{ V}$ a $0,27\text{ V}$. Assim, a faixa de tensão de saída é igual a aproximadamente $1,17\text{ V}$.

4.9 Design Centering

Como discutido na Seção 3.4, neste trabalho as técnicas de *design centering* implementadas são realizadas através de simulação elétrica da análise de Monte Carlo. As duas estratégias de *design centering* propostas na Seção 3.4 foram implementadas com os seus respectivos fluxos de execução.

Figura 50: *Testbench* para medição da faixa de tensão de saída (OS).

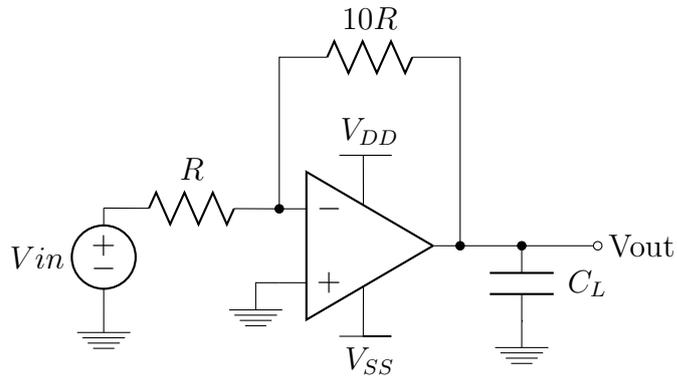
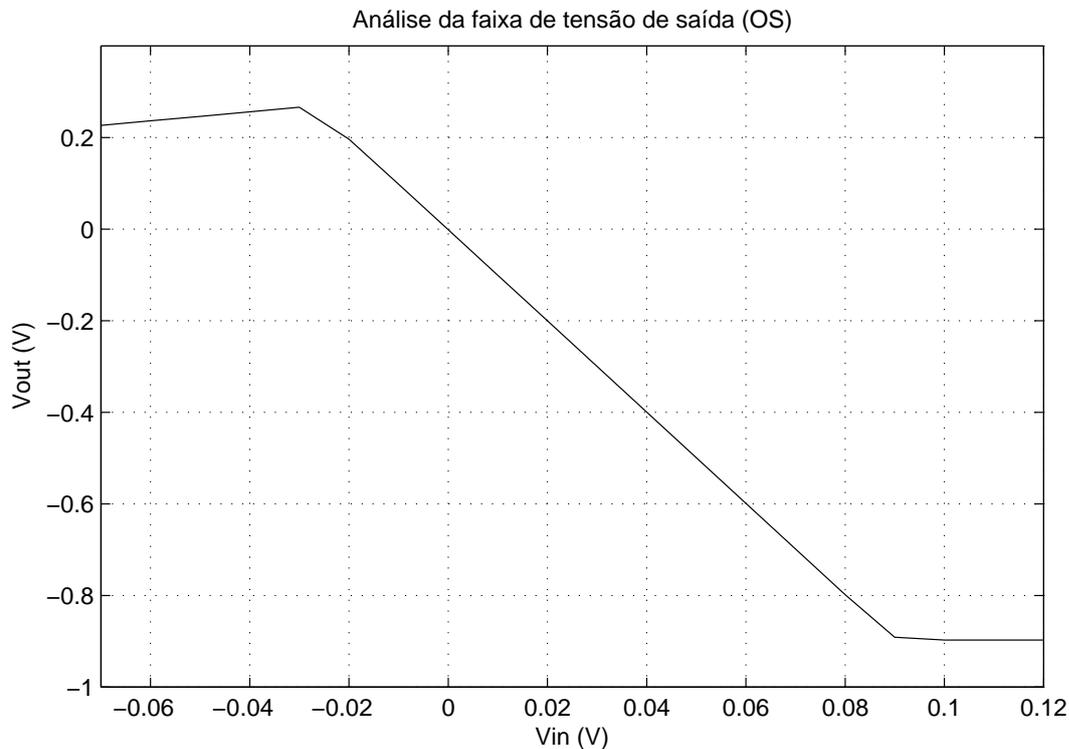


Figura 51: Sinal de tensão de saída gerado com a simulação do *testbench* da Figura 50 para a medição da faixa de tensão de saída.



As configurações do *design centering* referem-se à escolha da estratégia (1 ou 2) e do número de rodadas da análise de Monte Carlo desempenhadas a cada execução e, ao final, para a caracterização da solução final. As demais configurações são ajustadas nos outros blocos funcionais da ferramenta, como, por exemplo, o modelo de tecnologia para a simulação Monte Carlo que é inserido junto à configuração da tecnologia. As configurações referentes a simulação Monte Carlo, como tipo de simulação, são configuradas na descrição da topologia de circuito.

O parâmetro P_Y referente à ponderação de produtividade da Equação 3.37 deve

ser também ajustado experimentalmente, assim como os parâmetros PO_i e PR_j de $fc(X)$.

4.10 Conclusões

Neste capítulo a ferramenta desenvolvida neste trabalho foi apresentada, bem como os blocos que a compõem. Como mostrado, a ferramenta UCAF possui uma série de parâmetros de configurações, que mostram o número de possibilidades de comparações e análises que são possíveis. Estas comparações podem ser utilizadas como auxílio na escolha da topologia que pode ser utilizada em um dado projeto ou até mesmo na escolha de uma tecnologia de fabricação. Outra característica interessante é o fato de a ferramenta possuir interfaces de entradas de configurações e saídas de análise de resultados, de forma que possa ser facilmente executada e os resultados gerados também sejam facilmente analisados.

A forma modular de implementação da ferramenta permite que seja possível a inserção de diferentes topologias por parte do usuário, de forma que a aplicação do UCAF não se restrinja somente a um único circuito. Além disso, a modularidade facilita a inserção de novos algoritmos de otimização e *testbenches* de medidas de especificações.

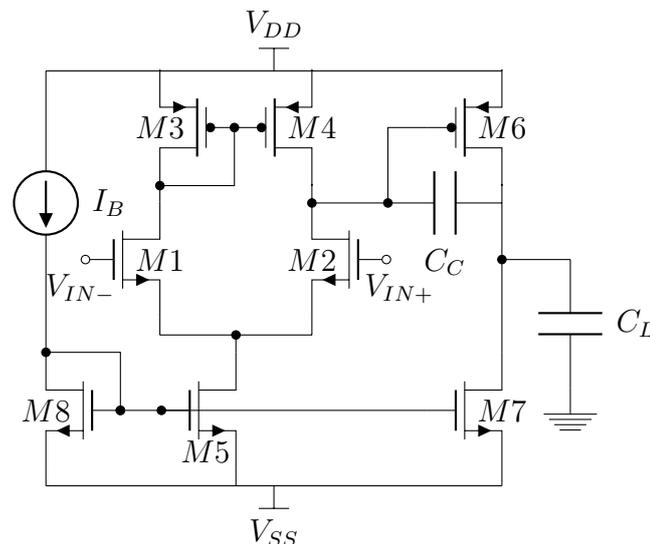
5 Resultados

Esta seção apresenta os resultados obtidos com a ferramenta UCAF. Os resultados tratam do dimensionamento de circuitos amplificadores operacionais em tecnologia CMOS.

5.1 Projeto Automático de um OTA Miller

O OTA Miller é um amplificador operacional de transcondutância composto por dois estágios de ganho. Uma implementação deste circuito em tecnologia CMOS é mostrada na Figura 52. Neste circuito, o primeiro estágio é um amplificador diferencial e o segundo estágio é um amplificador inversor. A conexão entre os estágios é feita por um capacitor de acoplamento (C_c) de forma a melhorar a estabilidade do circuito (ALLEN; HOLBERG, 2002). Este circuito tem como principais especificações o ganho em baixas frequências (A_{vo}), o produto ganho-largura de faixa (GBW), a margem de fase (PM), a velocidade de resposta - slew rate - (SR), a faixa de tensão de entrada em modo comum (ICMR) e a faixa de tensão de saída (OS).

Figura 52: Esquemático do OTA Miller em tecnologia CMOS.



O ganho para baixas frequência é definido como

$$A_{vo} = \frac{g_{m1}}{g_{ds2} + g_{ds4}} \cdot \frac{g_{m6}}{g_{ds6} + g_{ds7}} \quad (5.1)$$

em que g_{m1} e g_{m6} são as transcondutâncias dos transistores M1 e M6, respectivamente, e g_{ds2} , g_{ds4} , g_{ds6} e g_{ds7} são as condutâncias dos transistores M2, M4, M6 e M7. Nesta

equação, a primeira fração equivale ao ganho do primeiro estágio, amplificador diferencial, e a segunda fração ao ganho do segundo estágio, amplificador inversor.

O GBW é calculado com a equação

$$GBW = \frac{g_{m1}}{C_c} \quad (5.2)$$

em que C_c é a capacitância de acoplamento entre os estágios do amplificador.

O *slew rate*, pode ser calculado com a seguinte equação:

$$SR = \frac{I_5}{C_c} \quad (5.3)$$

onde I_5 refere-se à corrente de dreno do transistor M5. Esta equação sugere que SR seja diretamente proporcional a I_5 . Desta forma, grandes valores de SR refletem em uma alta potência dissipada pelo circuito.

O ICMR pode ser calculado com as equação seguintes:

$$ICMR^+ = V_{DD} - \sqrt{\frac{I_5}{\beta_3}} - |V_{T3}|(max) + V_{T1}(min) \quad (5.4)$$

$$ICMR^- = V_{SS} + \sqrt{\frac{I_5}{\beta_1}} + V_{T1}(max) + \sqrt{\frac{2I_5}{\beta_5}} \quad (5.5)$$

em que β é o parâmetro de transcondutância, proporcional à relação W/L, e V_T é a tensão de limiar.

A faixa de tensão de saída (OS) pode ser obtida como

$$OS^+ = V_{DD} + V_{DS6}(sat) \quad (5.6)$$

$$OS^- = V_{SS} + V_{DS7}(sat) \quad (5.7)$$

em que $V_{DS6}(sat)$ e $V_{DS7}(sat)$, são as tensões entre dreno e source de saturação dos transistores M6 e M7, respectivamente.

Para o projeto do OTA Miller da Figura 52, os oito transistores do circuito devem ser dimensionados, de forma que os parâmetros de largura (W) e o comprimento (L) do canal sejam obtidos. Devido à forma construtiva do circuito, algumas igualdades entre os transistores podem ser atribuídas. Assim, o par diferencial (M1-M2) e os espelhos de corrente (M3-M4 e M7-M8) devem possuir transistores iguais, fazendo com que seja necessário dimensionar somente 5 transistores. Além do dimensionamento dos transistores, para o projeto do circuito, a fonte de corrente I_B deve ser dimensionada. De maneira geral, o projeto do circuito da Figura 52 apresenta 11 variáveis livres de projeto: W_{M1} , L_{M1} , W_{M3} , L_{M3} , W_{M5} , L_{M5} , W_{M6} , L_{M6} , W_{M7} , L_{M8} e I_B . Além das variáveis livres de

projeto, o circuito apresenta algumas constantes, as quais são: fontes de alimentação (V_{DD} e V_{SS}), capacitância de carga (C_L) e capacitância de acoplamento (C_c). A capacitância C_c pode ser utilizada como variável de projeto. Porém, neste caso foi utilizada como uma constante igual a 1 pF . A capacitância C_L foi ajustada em 3 pF . As fontes de tensão V_{DD} e V_{SS} utilizadas são $0,9\text{ V}$ e $-0,9\text{ V}$, respectivamente. Este valor deve-se à tecnologia de fabricação do projeto. Neste caso, foi utilizada a tecnologia XFAB $0,18\text{ }\mu\text{m}$. A tecnologia também define os valores mínimos das dimensões do circuito que são $L = 0,18\text{ }\mu\text{m}$ e $W = 0,22\text{ }\mu\text{m}$.

O projeto do OTA Miller da Figura 52 foi realizado de duas formas. A primeira forma se deu de maneira manual e a segunda forma de maneira automática, utilizando a ferramenta desenvolvida neste trabalho. As subseções seguintes detalham os projetos e analisam os resultados.

5.1.1 Projeto Manual

O primeiro projeto do OTA Miller da Figura 52 foi feito manualmente. Para isso, foram utilizadas as Equações 5.1 a 5.7 e as equações simplificadas de corrente de dreno, condutância e transcondutância, conforme mostrado em Allen e Holberg (2002).

Para o projeto são requeridas as especificações mostradas na segunda coluna da Tabela 16. Os objetivos do projeto são de minimizar a potência dissipada e a área de gate. Com as equações simplificadas são encontradas as relações W/L para os transistores do circuito e o valor da fonte de corrente I_B iniciais. Após isso, com simulação elétrica SPICE, as especificações são medidas. Após inúmeros ajustes no dimensionamento, utilizando a estratégia de projeto mostrada em (ALLEN; HOLBERG, 2002), foi encontrada uma solução que atinge as especificações impostas ao projeto. As especificações encontradas para o circuito projetado manualmente estão mostradas na terceira coluna da Tabela 16. Os valores para as variáveis do circuito, bem como o nível de inversão dos transistores, estão mostradas na segunda coluna da Tabela 17.

Nos resultados, pode-se verificar que todas as especificações impostas foram atingidas, inclusive com uma certa margem. A área de gate do circuito é igual a $34,86\text{ }\mu\text{m}^2$, enquanto que a potência dissipada ficou em $77,08\text{ }\mu\text{W}$. O nível de inversão dos transistores se manteve em inversão moderada, com exceção do transistor M5 que tem um nível de inversão forte.

5.1.2 Projeto Automático

Para inserir o projeto na ferramenta UCAF, proposta neste trabalho, é necessário definir alguns parâmetros de configuração. O primeiro parâmetro diz respeito à tecnologia de fabricação, onde para este exemplo de projeto foram utilizados os parâmetros do modelo

BSIM 3v3 para a tecnologia XFAB 0,18 μm . Esta tecnologia define como tamanhos mínimos dos transistores $L = 0,18 \mu m$ e $W = 0,22 \mu m$. Tendo os valores mínimos possíveis para as variáveis W e L, foi definida a faixa de variação das variáveis livres do projeto como L entre 0,18 μm e 100 μm e W entre 0,22 μm e 200 μm . Para a fonte de corrente de polarização I_B foi definida a faixa de 1 μA a 300 μA . Como resolução das variáveis foi utilizada a grade da tecnologia, que neste caso é igual a 0,01 μm . Com esta resolução o número de soluções possíveis do problema é igual a $9,43 \times 10^{45}$. As constantes do circuito foram ajustadas para $V_{DD} = 0,9 V$, $V_{SS} = -0,9 V$, $C_c = 1 pF$ e $C_L = 3 pF$.

Após a configuração da tecnologia, o *netlist* foi inserido ao ambiente da ferramenta. A otimização foi configurada para Algoritmos Genéticos (GA), tendo como base os melhores resultados encontrados na seção 3.3.2: cromossomos binários, mutação binária, recombinação simples e seleção por *roulette wheel*. A inicialização da população foi feita aleatoriamente por uma função de inicialização. O tamanho da população utilizado foi igual a 100 indivíduos e o número de gerações executadas foi igual a 1000. A função custo foi configurada com todos os fatores de ponderação de restrições iguais a 1 e os valores requeridos iguais aos valores mostrados na segunda coluna da Tabela 16. Como mostra a Tabela 16 a área de gate e a potência dissipada são objetivos de projeto. A ponderação dos objetivos da função custo foi ajustada como $\frac{1}{40 \times 10^{-6}}$ para a potência dissipada e $\frac{1}{30 \times 10^{-12}}$ para a área de gate. Desta forma, a função custo é definida pela seguinte equação:

$$f_c = \frac{Pot.}{40 \times 10^{-6}} + \frac{Area}{30 \times 10^{-12}} + \sum_{j=1}^8 R(S_j(X), S_{jref}) \quad (5.8)$$

Nesta análise não foi utilizada nenhuma técnica de *design centering*, de forma que as variações de parâmetros não são levadas em conta durante o dimensionamento.

Os resultados obtidos estão mostrados na quarta coluna da Tabela 16. Nestes resultados, verifica-se que todas as especificações requeridas foram atingidas, sendo que alguns valores de especificações estão bastante próximos dos valores requeridos. Com relação aos objetivos de otimização, foram encontrados 12,91 μW de potência dissipada e 16,88 μm^2 para a área de gate. O tempo de execução da ferramenta desde a geração da população aleatoriamente até a finalização da heurística de otimização foi de 102 minutos. A Figura 53 mostra a evolução da função custo do dimensionamento do OTA Miller. Nesta figura, pode-se verificar que a heurística efetuou uma otimização no problema, uma vez que a melhor solução da população inicia em valor alto e é reduzido ao longo das iterações.

A terceira coluna da Tabela 17 mostra os valores para as variáveis do circuito e o nível de inversão dos transistores. Nesta tabela verifica-se que o nível de inversão predominante nos transistores foi o de inversão fraca, porém os transistores M5 e M7 encontram-se em inversão moderada.

Tabela 16: Especificações requeridas e encontradas na automação do projeto do circuito OTA Miller da Figura 52.

Especificação	Valor Requerido	Projeto Manual	Projeto Automático	Jafari et al. (2010)	Liu et al. (2009)*
Av0 (dB)	$\geq 70,00$	78,98	74,62	82,40	80,66
GBW (MHz)	$\geq 5,00$	8,53	5,23	9,77	2,04 ^a
PM (°)	$\geq 60,00$	62,45	61,30	60,00	55,60 ^b
SR (V/ μ s)	$\geq 5,00$	8,54	6,38	5,07	1,50 ^c
ICMR+ (V)	$\geq 0,70$	0,80	0,76	-	-
ICMR- (V)	$\leq -0,70$	-0,73	-0,72	-	-
OS (V)	$\geq 1,00$	1,15	1,14	1,17	1,91
Potência	<i>Minimizar</i>	77,08	12,91	52,00	1114,40
Dissipada (μ W)					
Área de Gate (μ m ²)	<i>Minimizar</i>	34,86	16,88	236,25	1407,78
Tempo de Execução (min)	-	-	102,01	-	164,42

* Projeto em tecnologia 0,25 μ m, utilizando $C_L = 30$ pF e $C_c = 40$ pF.

^a Requerido $GBW \geq 2$ MHz.

^b Requerido $PM \geq 50^\circ$.

^c Requerido $SR \geq 1,5$ V/ μ s.

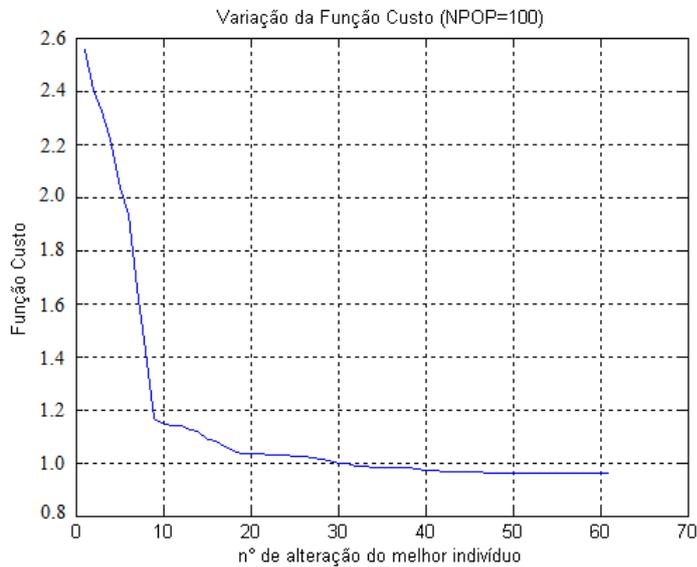
Tabela 17: Parâmetros encontrados na automação do projeto do circuito OTA Miller da Figura 52.

Especificação	Projeto Manual	Projeto Automático
W1/L1(μ m/ μ m)	1,27/0,40	2,88/0,27
Nível de Inversão	Moderada	Fraca
W3/L3(μ m/ μ m)	2,97/0,27	5,84/0,21
Nível de Inversão	Moderada	Fraca
W5/L5(μ m/ μ m)	1,60/0,40	1,42/0,90
Nível de Inversão	Forte	Moderada
W6/L6(μ m/ μ m)	48,80/0,44	31,90/0,26
Nível de Inversão	Moderada	Fraca
W7/L7(μ m/ μ m)	10,80/0,90	2,36/0,82
Nível de Inversão	Moderada	Moderada
$I_B(\mu$ A)	10,31	2,01

5.1.3 Comparação dos Resultados

As subseções anteriores demonstram os resultados obtidos para o projeto do OTA Miller da Figura 52 em tecnologia XFAB 0,18 μ m CMOS. Estes resultados foram comparados aos resultados obtidos por Jafari et al. (2010) e Liu et al. (2009). O projeto desenvolvido em Jafari et al. (2010) refere-se ao mesmo projeto apresentado nas seções anteriores, com as mesmas especificações impostas. O projeto desenvolvido por Liu et al. (2009) foi feito em tecnologia 0,25 μ m e apresenta algumas diferenças com relação aos

Figura 53: Evolução da função custo para o melhor indivíduo da população durante o processo de otimização.



parâmetros e valores requeridos. Os resultados obtidos com Jafari et al. (2010) e Liu et al. (2009) estão mostrados nas colunas 5 e 6 da Tabela 16, respectivamente. Analisando estes resultados, verifica-se que os valores das especificações encontradas estão bastante próximos. Em algumas especificações verifica-se que seus valores atingem com certa folga os valores requeridos. O processo de projeto teve um tempo de processamento de 102 minutos, que é relativamente pequeno, visto a complexidade do projeto e também o fato de o algoritmo ter executado 1552 iterações. Este tempo, em sua maior parte, é dispendido no processo de simulação, feita várias vezes a cada iteração de execução. Analisando o gráfico de evolução da função custo multiobjetivo, mostrado na Figura 53, é possível verificar que f_c sofreu um processo de otimização, uma vez que seu valor foi reduzido ao longo das iterações. No início, a solução encontrada apresenta um valor alto de função custo, o que reflete uma solução ruim (especificações requeridas não são atingidas, valores de área de gate e potência dissipada grandes). Ao final, a função custo tende a se estabilizar no valor de parada que satisfaz as restrições impostas e apresenta valores mínimos de potência e área de gate.

Com relação aos valores de potência e área de gate, objetivos de otimização, verifica-se que os resultados obtidos neste trabalho são melhores que os resultados obtidos em Jafari et al. (2010) e Liu et al. (2009). Comparando os resultados com Jafari et al. (2010), que utiliza as mesmas especificações e tecnologia, a ferramenta proposta neste trabalho encontrou uma área de gate e uma potência dissipada cerca de 92,86% e 75,17 % menores, respectivamente. Quando compara-se o projeto manual e automático, verifica-se que o projeto automático foi cerca de 83,25% e 51,58% melhor em termos de área de gate e potência dissipada. Uma justificativa dos melhores resultados encontrados com a

ferramenta proposta é o fato de esta ferramenta poder explorar de maneira eficiente todo o espaço de projeto, incluindo os níveis de inversão fraca e moderada. Esta exploração não é possível de ser encontrado em Jafari et al. (2010) devido à utilização de equações simplificadas de projeto, restringindo o projeto ao nível de inversão forte.

Para analisar o efeito das variações de parâmetros, foram feitas simulações Monte Carlo no circuito projetado. Para estas simulações foi utilizado o modelo com variações locais e globais, disponibilizados no kit de projeto da tecnologia XFAB 0,18 μm . Foi utilizado o modelo de distribuição de probabilidade gaussiana, uma vez que apresenta uma análise mais realística do projeto, como visto na Seção 3.4. Foram realizadas 1000 rodadas de simulação para a análise de Monte Carlo.

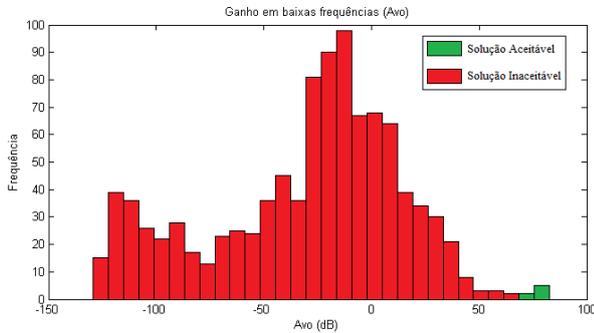
Após as simulações, os dados foram processados em Matlab[®] de forma que os histogramas de frequências para cada uma das especificações fossem gerados. Alguns dos histogramas obtidos são mostrados na Figura 54. É possível perceber que, nas especificações de análise no domínio da frequência (Avo, GBW e PM), a maior parte das soluções são inaceitáveis, de forma que a maior parte dos valores está abaixo do valor de especificação requerida. As demais especificações apresentam a maior parte das especificações dentro da faixa aceitável. Nestes resultados, apenas 7 execuções das 1000 rodadas atingem todas as especificações impostas, resultando em um *yield* de apenas 0,7%. Neste contexto, verifica-se que a variação de parâmetros afeta fortemente o circuito. Como analisado anteriormente, as especificações mais afetadas são as especificações Avo, PM e GBW. Esta variação deve-se ao fato de estas especificações estarem diretamente relacionadas ao par diferencial de entrada, que por sua vez é muito sensível ao *mismatch* entre os transistores. Desta forma, variações locais de parâmetros afetam o *mismatch*, resultando em uma perda de produtividade. O principal motivo da perda de produtividade se deve às especificações impostas ao projeto serem difíceis de serem atendidas pelo circuito. Em Deyati e Mandal (2011) o mesmo projeto é apresentado, porém com especificações mais fáceis de serem atingidas, como, por exemplo, um valor mínimo de Avo igual a 40 dB. Neste tipo de projeto (otimização de área de gate e potência dissipada) são atingidas produtividades na faixa de 19% sem a utilização de *design centering*.

5.1.4 Aplicando o Design Centering ao Projeto

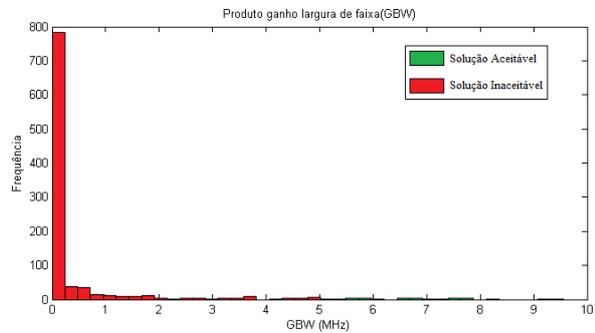
De acordo com a seção anterior, o projeto do OTA Miller da Figura 52 apresenta uma baixa produtividade sob variações locais de parâmetros. No intuito de aumentar a produtividade do circuito, esta seção apresenta os resultados do projeto do circuito com a estratégia de *design centering* propostas neste trabalho.

Para este projeto a ferramenta UCAF foi configurada para o *design centering* utilizando simulação Monte Carlo com 100 rodadas de simulação durante o processo de síntese e 1000 rodadas após a otimização. O fator de ponderação de *design centering*

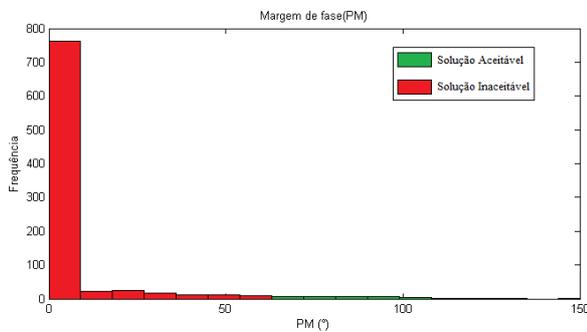
Figura 54: Histogramas de frequência para a simulação Monte Carlo no circuito OTA Miller projetado sem a utilização de *design centering*. (a) Avo, (b) GBW, (c) PM, (d) ICMR-, (e) ICMR+, (f) OS e (g) SR.



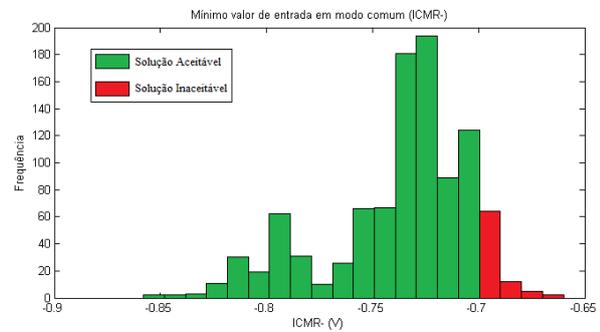
(a)



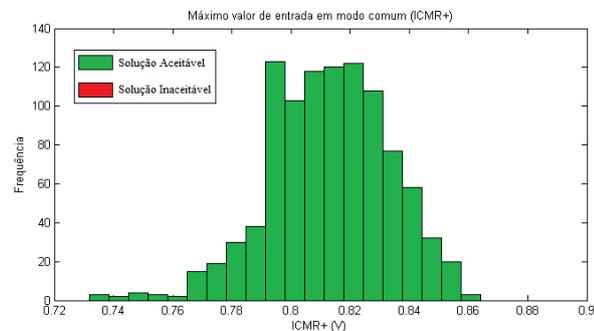
(b)



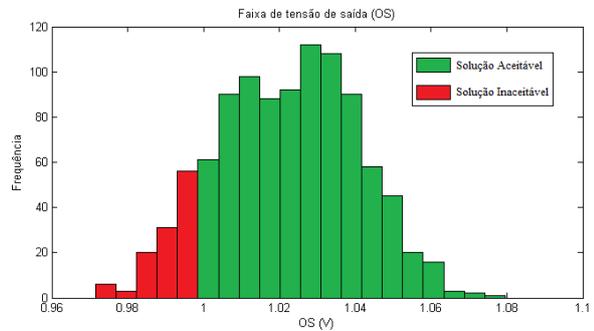
(c)



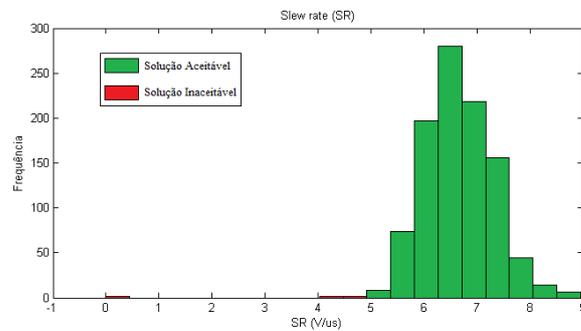
(d)



(e)



(f)



(g)

(P_Y) foi ajustado igual à unidade.

A ferramenta foi executada e os valores para as especificações encontradas estão mostrados na Tabela 18. O valor das variáveis do circuito estão mostrados na Tabela 19. Nestes resultados é possível verificar a melhoria imposta pelo *design centering*, uma vez que a produtividade do circuito passou de 0,7% para 20,08%. Desta forma, houve uma melhoria de aproximadamente 28,6 vezes na produtividade. Este aumento na produtividade deu-se às custas de um aumento na área de gate e na potência dissipada do circuito. O maior aumento foi evidenciado na potência dissipada, que passou de 12,91 μW sem *design centering* para 86,34 μW com *design centering*. Analisando as especificações, verifica-se que para o aumento do *yield*, em geral, o valor encontrado para a especificação ficou mais longe da faixa de fronteira requerida. Este fato está de acordo com o que foi analisado na Seção 2.3 deste trabalho, comprovando que um afastamento geométrico da fronteira de especificação requerida aumenta a produtividade do circuito. Este é um grande problema deste tipo de estratégia de dimensionamento, pois, como se tem um problema de otimização com objetivos e restrições e há uma relação entre as especificações e objetivos, sempre haverão especificações próximas à fronteira. Por exemplo, utilizando a Equação 5.3, obtemos a relação entre a corrente I_B e o valor de *slew rate*. Um aumento de SR requer um aumento na corrente I_B , o que, por consequência, gera um aumento na potência dissipada pelo circuito. No caso da otimização sem *design centering*, somente a área de gate e a potência dissipada devem ser minimizadas. Assim, o valor de SR é mantido muito próximo ao mínimo valor requerido, de forma que a potência dissipada seja pequena. Com o *design centering*, a especificação SR foi afastada do mínimo valor requerido de forma que o *yield* pudesse ser aumentado.

Com relação ao tempo de execução, houve um aumento de 147 minutos em relação à execução sem *design centering*. Pode-se considerar este aumento aceitável, uma vez que houve um grande incremento na produtividade.

Após a execução foram traçados os histogramas de frequências para todas as especificações do circuito. Estes histogramas estão mostrados na Figura 55. Comparando estes histogramas com os gerados na execução sem *design centering*, mostrados na Figura 54, verifica-se que a região de soluções aceitáveis foi aumentada. Desta forma, analisando as especificações separadamente, em geral houve um aumento do número de soluções que atendem às especificações impostas sob variações locais de parâmetros. Esta evidência justifica o uso de técnicas de *design centering* no projeto.

5.2 Projeto de um Amplificador Folded Cascode

Esta seção tem como objetivo apresentar o dimensionamento automático de um amplificador operacional *folded cascode* em tecnologia CMOS IBM 0,18 μm , com VT

Tabela 18: Especificações requeridas e encontradas na automação do projeto do circuito OTA Miller da Figura 52 com a utilização de *design centering*.

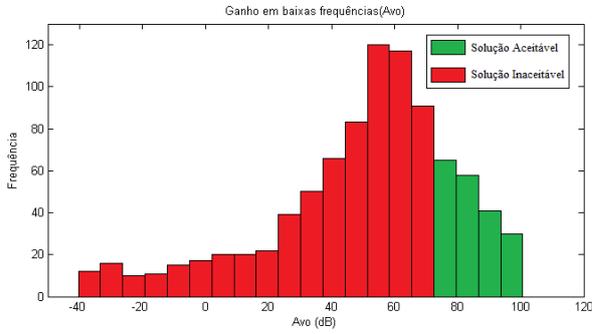
Especificação	Valor Requerido	Projeto sem <i>design centering</i>	Projeto com <i>design centering</i>
Av0 (dB)	$\geq 70,00$	74,62	84,81
GBW (MHz)	$\geq 5,00$	5,23	9,17
PM ($^{\circ}$)	$\geq 60,00$	61,30	65,13
SR (V/ μ s)	$\geq 5,00$	6,38	13,76
ICMR+ (V)	$\geq 0,70$	0,77	-
ICMR- (V)	$\leq -0,70$	-0,72	-
OS (V)	$\geq 1,00$	1,14	1,42
Potência	<i>Minimizar</i>	12,91	86,34
Dissipada (μ W)			
Área de Gate (μ m ²)	<i>Minimizar</i>	16,88	39,21
<i>yield</i>	<i>Maximizar</i>	0,70%	20,08%
Tempo de Execução (min)	-	102,01	249,02

Tabela 19: Parâmetros encontrados na automação do projeto do circuito OTA Miller da Figura 52 utilizando *design centering*.

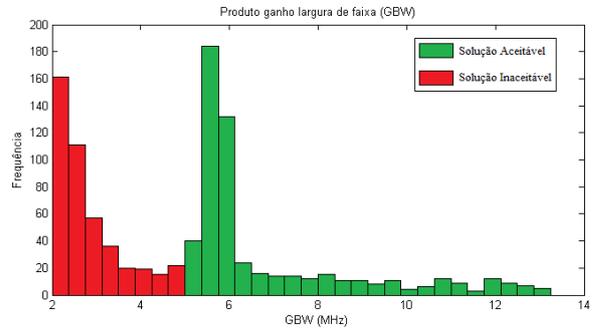
Especificação	Projeto sem <i>design centering</i>	Projeto com <i>design centering</i>
$W1/L1(\mu$ m/ μ m)	2,88/0,27	15,48/0,33
$W3/L3(\mu$ m/ μ m)	5,84/0,21	8,18/0,52
$W5/L5(\mu$ m/ μ m)	1,42/0,90	1,37/2,57
$W6/L6(\mu$ m/ μ m)	31,90/0,26	19,76/0,42
$W7/L7(\mu$ m/ μ m)	2,36/0,82	1,76/2,85
$I_B(\mu$ A)	2,01	21,19

regular e tensão de alimentação de 1,8 V. O esquemático do amplificador *folded cascode* utilizado neste projeto está mostrado na Figura 56. Este circuito é formado por um amplificador diferencial com carga ativa em configuração *cascode*. Como este projeto se dará somente de forma automática, utilizando a ferramenta UCAF com simulação elétrica, a modelagem matemática do circuito não se faz necessária. As tensões de polarização v_b e v_{bp} do esquemático da Figura 56 são obtidas com os circuitos mostrados na Figura 57. Para o projeto do amplificador operacional *folded cascode* é necessário dimensionar os seus 13 transistores (incluindo Mb e Mbp), além da fonte de corrente i_b e das fontes de tensão v_{bpc} e v_{bnc} . Assumindo as igualdades $M1 = M2$, $M4 = Mb$, $M5 = M6 = Mbp$, $M7 = M8$, $M9 = M10$ e $M11 = M12$, o circuito apresenta 15 variáveis livres de projeto ($W1, L1, W4, L4, W5, L5, W7, L7, W9, L9, W11, L11, i_b, v_{bpc}$ e v_{bnc}). Assim, para o projeto do circuito é necessário encontrar o valor para as 15 variáveis livres de projeto de forma a satisfazer as especificações requeridas.

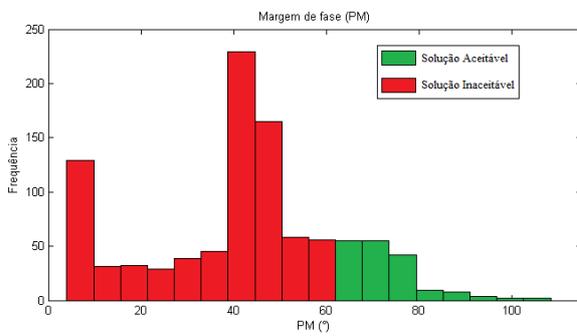
Figura 55: Histogramas de frequência para a simulação Monte Carlo no circuito OTA Miller projetado com a utilização de *design centering*. (a) Avo, (b) GBW, (c) PM, (d) ICMR-, (e) ICMR+, (f) OS e (g) SR.



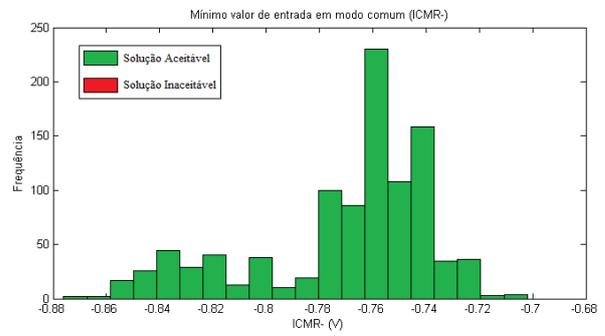
(a)



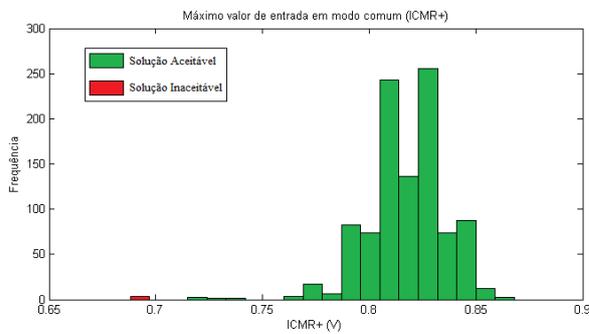
(b)



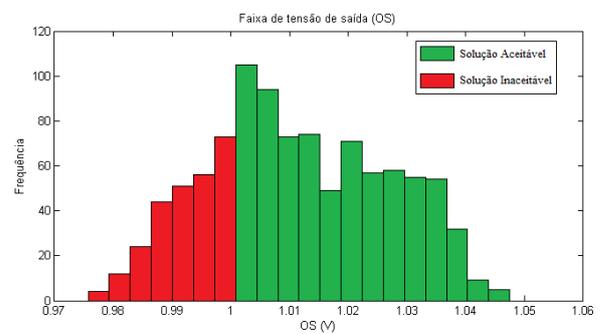
(c)



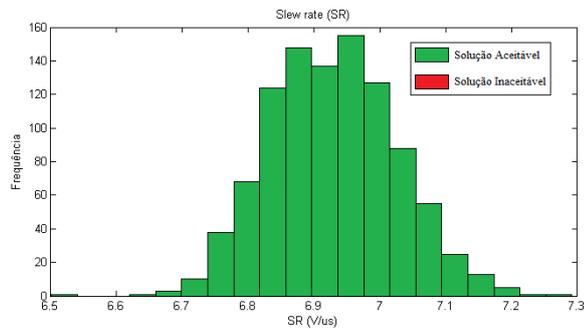
(d)



(e)



(f)



(g)

Figura 56: Esquemático do amplificador *folded cascode* em tecnologia CMOS.

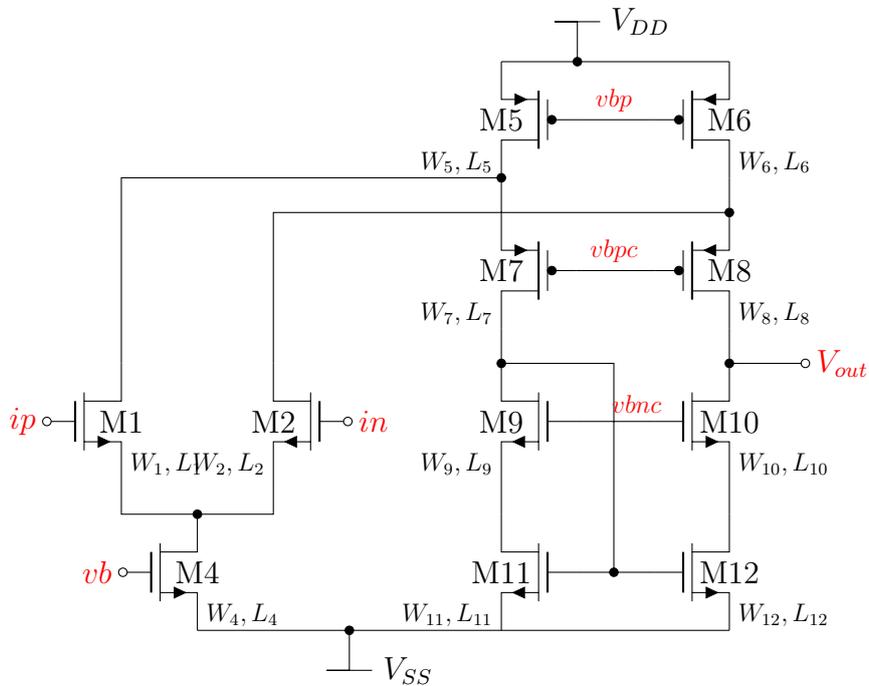
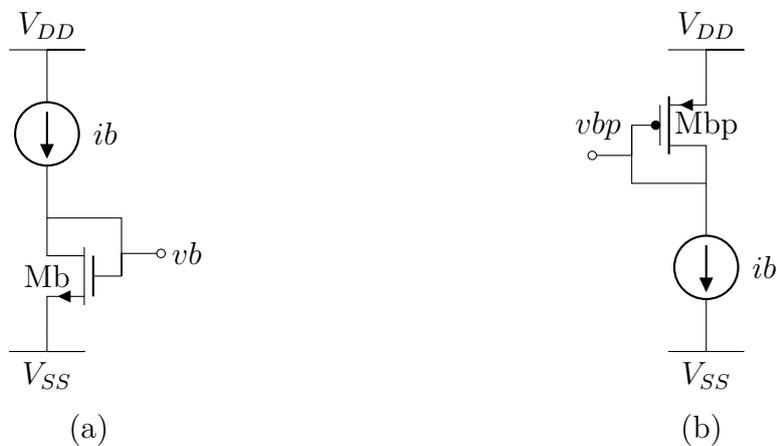


Figura 57: Fontes de corrente de polarização do circuito amplificador *folded cascode* da Figura 56. (a) fonte de polarização *vb* e (b) fonte de polarização *vbp*.



Para a inserção deste projeto na ferramenta UCAF é necessário primeiramente inserir os dados da tecnologia através dos parâmetros do modelo elétrico da tecnologia IBM 0,18 μm . Esta tecnologia define valores mínimos de W e L iguais a 0,24 μm e 0,18 μm , respectivamente. Após isso, o *netlist* SPICE do circuito é inserido ao UCAF. Neste passo são definidas as variáveis do circuito e definidos os seus limites. Neste projeto os valores de W podem assumir a faixa de 0,24 μm a 400 μm e L na faixa de 0,18 μm a 50 μm . I_b foi definido na faixa de 1 μA a 400 μA . As fontes de tensão de polarização v_{bpc} e v_{bnc} foram definidas na faixa de $-0,5$ a 0 e 0 a 0,5, respectivamente. A tensão de alimentação do circuito foi ajustada para $V_{DD} = 0,9 V$ e $V_{SS} = -0,9 V$. A capacitância de carga foi ajustada em 3 pF .

Para a otimização foi utilizada a heurística *Simulated Annealing* configurada com função do tipo Boltzmann para ambas as funções de geração de novos estados e de decaimento de temperatura. Como critério de parada foi utilizado um valor mínimo de temperatura. Além disso, foi configurada a busca global seguida de local. Para isso, foi selecionado o algoritmo de pontos interiores. Nesta otimização deseja-se dimensionar o circuito de forma a otimizar a área de gate e a potência dissipada e, ao mesmo tempo, manter as restrições de GBW, ganho em baixa frequência, margem de fase e *slew rate* dentro das faixas requeridas. Os valores requeridos para as especificações deste circuito estão mostrados na segunda coluna da Tabela 20.

A função custo foi ajustada de forma que o fator de ponderação de área de gate seja igual a $\frac{1}{100 \times 10^{-12}}$ e o fator de ponderação de potência seja igual a $\frac{1}{100 \times 10^{-6}}$. Como fatores de ponderação das restrições foram utilizados valores iguais à unidade para todas as especificações, com exceção do ganho em baixas frequências, onde o fator de ponderação utilizado foi igual a 2.

Para o *design centering* foi executada a segunda estratégia proposta neste trabalho, na qual a análise de Monte Carlo é feita somente nas melhores soluções. Para a simulação Monte Carlo foram utilizadas 100 rodadas de simulação durante o processo de dimensionamento e 2000 rodadas ao final de execução para a caracterização da solução final. Como modelo de Monte Carlo, foi utilizada variação de parâmetros locais.

Foram executadas três tipos de configuração com a ferramenta UCAF. A primeira foi a otimização global, sem a utilização de *design centering*. Na segunda execução foi feita uma otimização local após a global. Na terceira execução foi utilizada otimização global seguida de local com *design centering*. Os resultados destas análises estão mostrados na terceira, quarta e quinta colunas da Tabela 20. Nestes resultados pode-se notar que em todas as execuções as especificações requeridas foram satisfeitas.

Analisando os resultados de otimização global e otimização global seguida de local, sem *design centering*, verifica-se que, com a otimização local, houve uma redução na área de gate do circuito e um aumento na potência dissipada. Para verificar a melhoria, o valor

Tabela 20: Especificações obtidas com o dimensionamento automático do circuito amplificador *folded cascode* da Figura 56.

Especificação	Valor Requerido	UCAF	UCAF	UCAF	GENON
		Otimização Global	Otimização Global+Local	<i>design centering</i>	Barros et al. (2010)
GBW (MHz)	≥ 12	14,86	14,98	25,37	15,35
Av0 (dB)	≥ 70	73,04	70,02	71,43	70,61
PM ($^{\circ}$)	≥ 55	76,87	78,76	76,96	79,6
SR ($V/\mu s$)	≥ 10	10,98	11,37	16,43	15,36
Area de gate (μm^2)	minimizar	188,25	145,13	274,37	284,70
Potência dissipada (μW)	minimizar	129,9	133,2	219,7	244,6
<i>yield</i> (%)	maximizar	18,35	12,21	41,24	-
Tempo (min)	-	163,7	192,2	427,1	-

de função custo foi calculado, resultando em um valor igual a 3,181 para a otimização global e 2,782 para a otimização global seguida de local. Desta forma, mais uma vez, a otimização local apresenta uma melhoria no processo de otimização, sem comprometer as especificações requeridas. Com relação ao valor de produtividade, não levado em conta durante as duas primeiras análises, verifica-se que houve uma redução após a otimização local. Uma explicação para esta redução pode ser o fato de que o ganho para baixas frequências foi reduzido de 73,04 para 70,02, ficando assim muito próximo da fronteira do valor requerido.

O dimensionamento utilizando a técnica de *design centering* apresentou, ao final da síntese, uma produtividade de 41,24%. Este valor, quando comparado com os valores obtidos anteriormente (18,35% e 12,21%), apresenta uma melhoria mínima de 22,89 pontos percentuais. O aumento na produtividade do circuito é verificado com relação aos valores de especificações obtidos, uma vez que, em geral, os valores estão mais afastados da fronteira de valores requeridos. O aumento na produtividade foi compensado por um aumento na área de gate e na potência dissipada do circuito. Este aumento, em relação à otimização global seguida de local, foi de 65% para a potência dissipada e de 89% para a área de gate, respectivamente. Isto demonstra um *trade-off* entre os objetivos de projeto e a produtividade do circuito. Com relação ao tempo de síntese, a otimização local foi executada em 163,7 minutos. Já a otimização local ao final da otimização global levou 28,5 minutos, resultando em um tempo total de 192,2 minutos. A inserção do *design centering* levou o tempo de execução para 427,1 minutos, resultando em um aumento de 2,22 vezes. Este aumento no tempo de execução é aceitável, uma vez que a produtividade aumentou consideravelmente.

Com os resultados obtidos é possível comparar o desempenho da UCAF com outras ferramentas descritas na literatura. Um dos maiores problemas, porém, é o fato de que em muitos trabalhos os resultados demonstrados não podem ser reproduzidos devido à falta de informações. Neste sentido, os resultados apresentados por (BARROS; GUILHERME; HORTA, 2010) com a ferramenta GENOM são passíveis de comparação, já que a mesma configuração experimental pode ser reproduzida. Porém, os parâmetros do modelo de simulação não são divulgados. A ferramenta GENON é baseada em uma variação dos algoritmos genéticos como heurística principal de otimização. O amplificador *folded cascode* foi implementado na tecnologia UMC 0,18 μ m. Os resultados finais obtidos com GENOM para o mesmo circuito sintetizado neste trabalho estão sumarizados na sexta coluna da Tabela 20. Com base nestes resultados verifica-se que ambas as metodologias apresentam resultados similares para as restrições de projeto. Por outro lado, para a potência dissipada e a área de gate os valores obtidos com a ferramenta UCAF são aproximadamente a metade dos valores obtidos com a GENON para as execuções sem *design centering*, e um pouco menores com a utilização do *design centering*. Como em Barros, Guilherme e Horta (2010) não foi utilizado *design centering*, pode-se afirmar que os resultados obtidos neste trabalho são cerca de 54% e 51% melhores em termos de potência dissipada e área de gate, respectivamente. Assim, os resultados obtidos mostram que a utilização da ferramenta UCAF é uma boa metodologia para o projeto de amplificadores operacionais operando em micropotência. Novamente, é importante notar que a comparação entre os resultados pode não ser exata porque os parâmetros do modelo não são exatamente iguais, uma vez que os processos podem variar de *foundry* para *foundry*.

Para os resultados mostrados anteriormente, os valores finais para as variáveis do circuito são mostrados na Tabela 21. Nesta tabela, pode-se verificar que a largura do canal dos transistores tendem a ser maiores que o comprimento do canal e que a magnitude de ambos é similares em todos os processos de otimização.

5.3 Conclusão

Esta seção demonstrou a aplicação da ferramenta UCAF no projeto de duas topologias de amplificadores operacionais: OTA Miller e *folded cascode*. Os projetos foram executados com diferentes configurações da ferramenta. Nas execuções a ferramenta partiu de uma solução inicial aleatória e, ao final, uma solução que atende às especificações impostas é encontrada. A solução encontrada é ainda otimizada em termos de potência e área de gate. Nos resultados verifica-se que a otimização global seguida de otimização local provoca uma melhoria no resultado de dimensionamento.

Nas execuções sem *design centering* foi verificado que as especificações finais do circuito encontram-se próximas à fronteira do valor requerido para as especificações.

Tabela 21: Resultado final para as variáveis livres de projeto obtidas com o dimensionamento automático do circuito amplificador *folded cascode* da Figura 56.

Variável	UCAF Otimização Global+Local	UCAF <i>design centering</i>	GENON Barros et al. (2010)
W_1 (μm)	11,58	163,59	14,91
W_4 (μm)	22,39	156,80	6,99
W_5 (μm)	14,13	66,97	36,78
W_7 (μm)	30,72	110,31	63,04
W_9 (μm)	7,16	27,31	31,45
W_{11} (μm)	6,58	52,47	7,32
L_1 (μm)	0,73	0,22	1,38
L_4 (μm)	0,71	0,23	1,94
L_5 (μm)	0,29	0,36	0,37
L_7 (μm)	0,52	0,33	0,91
L_9 (μm)	0,87	0,91	0,89
L_{11} (μm)	4,54	0,18	2,19
v_{bnc} (V)	0,0579	0,018	0,001
v_{bpc} (V)	-0,0408	-0,019	-0,0449
i_b (μA)	36,78	40,22	48,51

Isso ocorre devido a algumas relações conflitantes entre as especificações restritivas e os objetivos de otimização. Assim, as especificações cujo aumento não colaboram para a otimização do objetivo são mantidas nos valores mínimos aceitáveis. Este fato faz com que, na ocorrência de variações de parâmetros o circuito fique polarizado em uma região que não atende às especificações impostas. Com a utilização de *design centering* esta característica é alterada, uma vez que para otimizar o *yield* é necessário manter uma certa margem entre o valor alcançado e o valor requerido para uma especificação.

Com a utilização de *design centering*, a produtividade dos circuitos é aumentada significativamente, em um tempo de execução razoável.

Considerações Finais

Este trabalho apresentou o desenvolvimento da ferramenta UCAF, voltada para o dimensionamento automático de circuitos integrados analógicos com análise de produtividade. Primeiramente, uma breve revisão bibliográfica sobre o projeto de circuitos analógicos foi apresentada, na qual é verificado que, atualmente, ainda existe uma necessidade no desenvolvimento de ferramentas que automatizam totalmente o projeto de um circuito integrado analógico.

Foi utilizado um amplificador diferencial simplificado para que as heurísticas *Simulated Annealing* e Algoritmos Genéticos fossem analisadas. Nestas análises foi constatado que com qualquer uma das heurísticas é possível explorar o espaço de projeto de um circuito integrado analógico eficientemente. Para isso, basta configurar os parâmetros adequadamente. Foi constatado também que a otimização global seguida de busca local fornece melhorias no processo de otimização. Para o amplificador diferencial simplificado foram analisadas duas técnicas de *design centering* baseadas na análise Monte Carlo para estimar a produtividade do circuito. Foi verificado que a simulação Monte Carlo demanda um enorme tempo de processamento, porém quando utilizada somente nas melhores soluções durante o dimensionamento, o tempo de processamento é reduzido consideravelmente sem reduzir a qualidade da solução.

A ferramenta UCAF foi implementada em Matlab® com objetivo de automatizar a etapa de dimensionamento do circuito integrado analógico e ainda efetuar uma otimização neste circuito. A construção da ferramenta foi realizada através de funções modulares. Com a configuração das funções através de um *script* é possível ajustar o fluxo de dimensionamento automático, fazendo com que um dado projeto possa ser feito de diversas maneiras. A ferramenta apresenta interfaces de entrada e saída, de modo a ser facilmente utilizada. A inserção de topologias de amplificadores operacionais com saída em terminação comum é aberta ao usuário, bastando a inserção do *netlist* e a especificação das variáveis livres.

Para demonstrar a ferramenta desenvolvida, foi desenvolvido o projeto de duas topologias de circuitos amplificadores operacionais: OTA Miller e *folded cascode*. As soluções encontradas com a execução da ferramenta satisfizeram todas as restrições impostas, além da otimização das especificações de potência dissipada e área de gate. Além disso, foram encontradas soluções com maximização da produtividade. Nas execuções sem a utilização do *design centering* foi verificado que as especificações finais do circuito encontram-se próximas à fronteira do valor requerido para as especificações. Com isso,

na ocorrência de variações de parâmetros, o circuito fica susceptível a não atender às especificações impostas. Com a utilização de *design centering* esta característica se altera, uma vez que para otimizar o *yield* é necessário manter uma certa margem entre o valor alcançado e o valor requerido para uma especificação.

Na análise de variações de parâmetros foi verificado que as variações locais representam um enorme problema para circuitos integrados analógicos, como amplificadores operacionais, devido à alta dependência do *mismatch*.

Ao comparar os resultados encontrados com a UCAF com outros trabalhos da literatura verifica-se que a ferramenta apresentou bom desempenho. Nestes resultados foram encontrados circuitos com melhor otimização de potência e área de gate, para as mesmas especificações mostradas na literatura. Além disso, os circuitos apresentaram uma produtividade razoável, sob variações locais e globais de parâmetros. Neste contexto, este trabalho apresentou uma ferramenta com grande potencial para a exploração do espaço de dimensionamento de circuitos integrados analógicos.

Como sequência deste trabalho, propõe-se que sejam utilizadas outras técnicas de *design centering* de forma que os resultados possam ser comparados. Propõe-se, também, a inserção de outras técnicas de otimização para a exploração do espaço de projeto. Para a simulação elétrica, é indicado que sejam inseridos mais *testbenches* de simulação, de forma que o circuito possa ser caracterizado utilizando mais especificações, como, por exemplo, ruído, faixa de rejeição em modo comum (CMRR), entre outros.

Referências

ALLEN, P. E.; HOLBERG, D. R. *CMOS Analog Circuit Design*. 2. ed. [S.l.]: Oxford University Press, 2002. Citado 7 vezes nas páginas 27, 32, 46, 89, 97, 105 e 107.

ALPAYDIN, G.; BALKIR, S.; DUNDAR, G. An evolutionary approach to automatic synthesis of high-performance analog integrated circuits. *IEEE Transactions on Evolutionary Computation*, v. 7, n. 3, p. 240–252, 2003. Citado na página 33.

BALKIR, S.; DUNDAR, G.; OGRENCI, A. S. *Analog VLSI Design Automation*. [S.l.]: CRC Press, 2003. Citado 3 vezes nas páginas 24, 27 e 31.

BARROS, M. F. M.; GUILHERME, J. M. C.; HORTA, N. C. G. *Analog Circuits and Systems Optimization Based on Evolutionary Computation Techniques*. [S.l.]: Springer, 2010. Citado 6 vezes nas páginas 31, 32, 33, 35, 94 e 119.

BUEHLER, M. et al. Dfm/dfy design for manufacturability and yield - influence of process variations in digital, analog and mixed-signal circuit design. *Proceedings European Conference on Design and Test*, 2006. Citado na página 36.

CORTES, F. P. et al. Análise e projeto de módulos amplificadores e comparadores em tecnologia cmos 0.35 um. *IX Workshop Iberchip*, 2003. Citado 2 vezes nas páginas 24 e 27.

DEYATI, S.; MANDAL, P. An automated design methodology for yield aware analog circuit synthesis in submicron technology. *12th International Symposium on Quality Electronic Design*, 2011. Citado 5 vezes nas páginas 32, 33, 42, 43 e 111.

DRENNAN, P. G.; MCANDREW, C. C. Understanding mosfet mismatch for analog design. *IEEE Journal of Solid-State Circuits*, v. 38, n. 3, p. 450–456, MARCH 2003. Citado 5 vezes nas páginas 15, 36, 37, 39 e 76.

ENZ, C. C. A short story of the ekv mos transistor model. *IEEE Solid State Circuit Newsletter*, v. 13, n. 32, p. 24, 2008. Citado 2 vezes nas páginas 31 e 87.

FLOUDAS, C. A.; PARDALOS, P. M. *Encyclopedia of Optimization*. 2. ed. [S.l.]: Springer, 2008. Citado 2 vezes nas páginas 54 e 60.

GIELEN, G.; RUTENBAR, R. Computer-aided design of analog and mixed-signal integrated circuits. *Proceedings of the IEEE*, v. 88, p. 1825–1852, 2000. Citado na página 23.

GIRARDI, A. *Automação do Projeto de Módulos CMOS Analógicos Usando Associações Trapezoidais de Transistores*. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, 2007. Citado na página 30.

GRAEB, H. E. *Analog Design Centering and Sizing*. [S.l.]: Springer, 2007. Citado 4 vezes nas páginas 23, 27, 39 e 41.

HAGGLUND, R. *Studies on Design Automation of Analog Circuits Performance Metrics*. Tese (Doutorado) — Department of Electrical Engineering, Linkpings university, 2003. Citado na página 24.

HERSHENSON, M.; BOYD, S.; LEE, T. Gpcad: A tool for cmos op-amp synthesis. *Proceedings IEEE/ACM International Conference Computer-Aided Design*, p. 296–303, 1998. Citado na página 35.

HERSHENSON, M. M.; BOYD, S. P.; LEE, T. H. Optimal design of a cmos op-amp via geometric programming. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 20, p. 1–21, 2001. Citado na página 32.

HOUCK, C.; JOINES, J.; KAY, M. A genetic algorithm for function optimization: A matlab implementation. *NCSU-IE TR 95-09*, 1995. Citado 5 vezes nas páginas 62, 66, 68, 72 e 92.

INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS - ITRS. *Design. Design*. [S.l.], Outubro 2011. Disponível em: <<http://www.itrs.net/Links/2011ITRS/2011Chapters/2011Design.pdf>>. Citado na página 24.

JAFARI, A. et al. Design of analog integrated circuits by using genetic algorithm. *Second International Conference on Computer Engineering and Applications*, 2010. Citado 6 vezes nas páginas 24, 32, 33, 109, 110 e 111.

KHOUAS, A.; DERIEUX, A. Fault simulation for analog circuits under parameter variations. *Journal of electronic testing: Theory and Applications*, v. 166, p. 269–278, 2000. Citado na página 40.

LABRAK, L. et al. Automated cost function formulation for analog design optimization. *European Conference on Circuit Theory and Design*, p. 559 – 562, 2007. Citado na página 35.

LI, X. et al. Performance-centering optimization for system-level analog design exploration. *IEEE International Conference on Computer Aided Design*, p. 422–429, 2005. Citado na página 42.

LINDEN, R. *Algoritmos Genéticos: Uma importante ferramenta da inteligência artificial*. [S.l.]: Brasport, 2006. Citado na página 62.

LIU, B.; MESSAOUDI, J.; GIELEN, G. A fast analog circuit yield estimation method for medium and high dimensional problems. *Design, Automation and Test in Europe (DATE)*, 2012. Citado na página 41.

LIU, B. et al. Analog circuit optimization system based on hybrid evolutionary algorithms. *Integration, The VLSI journal*, v. 42, p. 137–148, 2009. Citado 6 vezes nas páginas 24, 34, 54, 72, 109 e 110.

MARTENS, E.; GIELEN, G. Classification of analog synthesis tools based on their architecture selection mechanisms. *Integration, The VLSI journal*, v. 41, p. 238–252, 2008. Citado 2 vezes nas páginas 31 e 33.

MATHWORKS. *Fmincon - Otimização Não-linear Multiobjetivo*. Setembro 2012. Disponível em: <<http://www.mathworks.com/help/optim/ug/fmincon.html>>. Citado na página 72.

MATHWORKS. *Global Optimization Toolbox*. Maio 2012. Disponível em: <<http://www.mathworks.com/products/global-optimization/description7.html>>. Citado 2 vezes nas páginas 52 e 92.

MATHWORKS. *Simulated Annealing Solver*. Outubro 2012. Disponível em: <<http://www.mathworks.com/discovery/simulated-annealing.html>>. Citado na página 92.

MICHALEWICZ, Z.; FOGEL, D. B. *How to Solve it: Modern Heuristic*. 2. ed. [S.l.]: Springer, 2000. Citado 2 vezes nas páginas 54 e 59.

MONTORO, C. G.; SCHNEIDER, M. C.; CUNHA, A. I. The advanced compact mosfet (acm) model for circuit analysis and design. *IEEE Custom Integrated Circuits Conference*, p. 519–526, 2007. Citado na página 87.

MUTLU, A. A.; RAHMAN, M. Statistical methods for the estimation of process variation effects on circuit operation. *IEEE Transactions on Electronics Packaging Manufacturing*, v. 28, n. 4, p. 364–375, October 2005. Citado 3 vezes nas páginas 38, 39 e 76.

ORSHANSKY, M.; NASSIF, S.; BONING, D. *Design for Manufacturability and Statistical Design: A Constructive Approach*. [S.l.]: Springer, 2008. Citado 3 vezes nas páginas 36, 37 e 38.

PHELPS, R. et al. Anaconda: Simulation-based synthesis of analog circuits via stochastic pattern search. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 19, p. 703–717, 2000. Citado 3 vezes nas páginas 33, 35 e 93.

PRESS, W. et al. *Numerical Recipes: The Art of Scientific Computing*. 3. ed. New York: Cambridge University Press, 2007. Citado 2 vezes nas páginas 56 e 72.

RAZAVI, B. *Design of Analog CMOS Integrated Circuits*. [S.l.]: McGraw-Hill, 2000. Citado 2 vezes nas páginas 32 e 45.

SAXENA, S. et al. Variation in transistor performance and leakage in nanometer-scale technologies. *IEEE Transactions on Electron Devices*, v. 55, n. 1, p. 131–144, Janeiro 2008. Citado 2 vezes nas páginas 38 e 76.

SCHNEIDER, M. C.; GALUP-MONTORO, C. *CMOS Analog Design Using All-Region MOSFET Modeling*. [S.l.]: Cambridge University Press, 2010. Citado na página 38.

SEVERO, L. C.; GIRARDI, A. G. Ucaf a framework for analog integrated circuit analysis and design. *11th Microelectronics Student Forum - SFORUM*, 2011. Citado na página 74.

SEVERO, L. C. et al. Simulated annealing to improve analog integrated circuit design: Trade-offs and implementation issues. In: *Simulated Annealing - Single and Multiple Objective Problems*. [S.l.]: Rijeka : Intech, 2012. p. 261–284. Citado na página 60.

SILVEIRA, F.; FLANDRE, D.; JESPERS, P. A gm/id based methodology for the design of cmos analog circuits and its application to the synthesis of a silicon-on-insulator micropower ota. *IEEE Journal of Solid-State Circuits*, v. 31, p. 1314–1319, 1996. Citado na página 27.

- SIVANANDAM, S. N.; DEEPA, S. N. *Introduction to Genetic Algorithms*. [S.l.]: Springer, 2008. Citado 2 vezes nas páginas 61 e 62.
- SMEDT, B. D.; GIELEN, G. Watson: Design space boundary exploration and model generation for analog and rfc design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 22, p. 213–224, 2002. Citado 2 vezes nas páginas 33 e 35.
- SOBE, U. et al. Analog ip porting by topology conversion and optimization. *IP-ESC conference*, 2009. Citado na página 36.
- STEFANOVIC, D.; KAYAL, M. *Structured Analog CMOS Design*. [S.l.]: Springer, 2009. Citado na página 31.
- XI, X. J. et al. *BSIM4.3.0 MOSFET Model - User Manual*. CA 94720, 2003. Citado na página 37.
- YE, Y. et al. Statistical modeling and simulation of threshold variation under dopant fluctuations and line-edge roughness. *Design Automation Conference, 2008. DAC 2008. 45th ACM/IEEE*, p. 900 – 905, 2008. Citado 3 vezes nas páginas 15, 37 e 38.

ANEXO A – Script de Interface de Entrada

03/10/12 23:02 D:\Es...\Script Diff Amp simples PCGAMA.m 1 of 2

```

clc
clear UCAF

global UCAF

%Script Model

%Dir
UCAF.Dir='D:\UCAF\UCAF_2_0\UCAF\Projetos\Projeto_AmpDiff_Simples';
UCAF.Name='Projeto_DiffAmpSimples_v4';

%Simulation
%Electrical Simulator Selection
UCAF.Simulation.Simulator=1; %(1-HSPICE)
%Simulator Dir (Folder that is the simulator.exe)
UCAF.Simulation.SimulatorDir='C:\synopsys\Hspice_F-2011.09-SP1\BIN\';

%Technology
UCAF.Technology.Dir='D:\UCAF\UCAF_2_0\UCAF\Projetos\Projeto_AmpDiff_Simples'; %
Pasta aonde está a biblioteca
UCAF.Technology.FolderName='xh018'; %Nome da pasta em que está a biblioteca
%Circuit Library
UCAF.Circuit.Use=0; %1-Usar biblioteca 0-inserir diretamente
UCAF.Circuit.Lib.LibFile='mylib'; %Library file
UCAF.Circuit.Lib.Topology='mylib'; %LibName
UCAF.Circuit.User.Dir='D:\UCAF\UCAF_2_0\UCAF\Projetos\Projeto_AmpDiff_Simples';
UCAF.Circuit.User.Netlist='AmpDiffSimples_Netlist.txt';
UCAF.Circuit.User.NetlistMC='AmpDiffSimples_MCNetlist.txt';
UCAF.Circuit.User.NetlistScktName='AmpDiff';
UCAF.Circuit.User.NetlistType='1'; %1-DiffIn_SingleOut, 2-DiffIn_DiffOut, 3-
SingleIn_SingleOut
UCAF.Circuit.User.Specifications=[];
UCAF.Circuit.User.VDD=0.9; %VDD=-VSS
%Specifications
UCAF.Specifications.VarNames=['W1';'L1']; %var1;var2.. %Elementos devem ter o mesmo
número de caracteres
UCAF.Specifications.VarBounds=[0.22 10;0.18 1];
UCAF.Specifications.SpecMeasure.S=
['SP001AC_type1';'SP001GateArea';'SP001DC_type1']; %Elementos devem ter o mesmo
número de caracteres

UCAF.Specifications.SpecMeasure.Required.SP001AC_type1=[50 1 10 1; %av0 %value type
aceitável weighing: type=1-min constraint, 2-max constraint, 4-optimization min 5-
optimization max
                                1e6 1 1e3 1;%GBW
                                60 1 20 1; %MF
                                inf 4 40e-6 0]; %pot
UCAF.Specifications.SpecMeasure.Required.SP001AC_type1_MF=1; %1 - Phase
Margin=|Phase+180| 0 - Phase Margin=|Phase-180|

UCAF.Specifications.SpecMeasure.Required.SP001TR_type1=[5 1 0.1 1.5]; %SR %value
type aceitável/reference weighing: type=1-max/min, 4-optimization min 5-optimization max
UCAF.Specifications.SpecMeasure.Required.SP001DC_type1=[0.5 1 -0.9 1; %ICMR
                                -0.5 1 0.9 0];

```

03/10/12 23:02 D:\Es...\Script Diff Amp simples PCGAMA.m 2 of 2

```

    UCAF.Specifications.SpecMeasure.Required.SP002DC_type1=[0.5 1 -0.9 1; %OS
                                                         -0.5 1 0.9 1];
    UCAF.Specifications.SpecMeasure.Required.SP001GateArea.S=[0 4 30 1];
    UCAF.Specifications.SpecMeasure.Required.SP001GateArea.Eq=[1 2 4]; %✓
var1*var2*multiple

    %Adiciona HSPICE Toolbox
    addpath('HspiceToolbox'); %Compilar pela primeira vez na máquina

%Optimization
    UCAF.Optimization.Heuristic=2; %1-GA, 2-SA
    UCAF.Optimization.GA.a=1;
    %Simulated Annealing Options
    UCAF.Optimization.SA.OPTIONS = saoptimset('AnnealingFcn', ✓
@annealingboltz, 'TemperatureFcn', @temperatureboltz, 'ReannealInterval', inf, 'TolFun', ✓
0);
    UCAF.Optimization.SA.XInitOrRandom=0; %1-User 0-Random
    UCAF.Optimization.SA.XInit=[];

%Cost Function
    UCAF.CostFunction.Normalization=1; %1-reta normalizada

%Design Centering

    UCAF.DesignCentering.Strategy=0; %0-Non Used; 1-Monte Carlo in all Generated Solutions
    UCAF.DesignCentering.Options.Monte=100;
    UCAF.DesignCentering.Options.MonteEnd=2000; %Monte Carlo Evaluations in Final ✓
Solution

    UCAF.DesignCentering.Options.Py=1;
    %Read MC Specifications
    %SP001AC_type1
    UCAF.DesignCentering.SP001AC_type1.Read.L1avo=155;
    UCAF.DesignCentering.SP001AC_type1.Read.L2avo=217;
    UCAF.DesignCentering.SP001AC_type1.Read.L1avoP=3;
    UCAF.DesignCentering.SP001AC_type1.Read.L1P=154
    UCAF.DesignCentering.SP001AC_type1.Read.L2P=215
    UCAF.DesignCentering.SP001AC_type1.Read.L1PC=3;
    %SP001TR_type1
    UCAF.DesignCentering.SP001TR_type1.Read.L1sr=154;
    UCAF.DesignCentering.SP001TR_type1.Read.L2sr=215;
    UCAF.DesignCentering.SP001TR_type1.Read.L1srC=3;

%Call UCAF Core

UCAFCore

```

ANEXO B – Relatório de Saída

*Computer Architecture and Microelectronic Group - GAMA

*Federal University of Pampa - UNIPAMPA

*Developed by Lucas C. Severo (lucas.severo(at)unipampa.edu.br)

Results:

Xend:

W1=1.504340e+000

L1=5.407907e-001

W3=5.592979e+000

L3=6.790103e-001

W5=3.484016e+000

L5=6.348696e-001

W6=1.004958e+001

L6=2.349766e-001

W7=1.108594e+001

L7=5.556319e-001

IB=1.632276e+001

CostFunction=-7.389120e-001

Execution Time:

Start at 21-Sep-2012 17:16:34

End at 22-Sep-2012 02:03:33

Av0=8.877720e+001

GBW=1.090000e+007

MF=3.055377e+002

Pot=1.507000e-004

SR=1.606000e+001

ICMR+=7.380000e-001

ICMR-=-6.720000e-001

OS+=8.327973e-001

OS-=-8.860034e-001

GateArea=2.216736e+001

Yield=2.900000e-001