TIAGO OLIVEIRA WEBER

**Síntese de CIs analógicos em nível de circuito e sistema
utilizando métodos modernos de otimização**

São Paulo
2015

TIAGO OLIVEIRA WEBER

**Síntese de CIs analógicos em nível de circuito e sistema
utilizando métodos modernos de otimização**

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Doutor em Ciências.

São Paulo
2015

TIAGO OLIVEIRA WEBER

**Síntese de CIs analógicos em nível de circuito e sistema
utilizando métodos modernos de otimização**

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Doutor em Ciências.

Área de Concentração:

Microeletrônica

Orientador:

Prof. Dr. Wilhelmus A. M. Van Noije

São Paulo
2015

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catálogo-na-publicação

Weber, Tiago Oliveira

Síntese de CIs analógicos em nível de circuito e sistema utilizando métodos modernos de otimização / T. O. Weber -- versão corr. -- São Paulo, 2015.

252 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.CAD 2.Circuitos integrados 3.Otimização global I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.

Dedico este trabalho a meu pai, Juraci, a minha mãe, Carmem, e a meu irmão, Roberto.

AGRADECIMENTOS

Nenhum caminho é percorrido individualmente. Assim, diversas são as pessoas que auxiliaram a conclusão deste trabalho e aproveito este local para ressaltar algumas delas.

Agradeço ao meu orientador, Prof. Dr. Wilhelmus A. M. Van Noije, por ter sido um inestimável guia durante a trajetória deste doutorado, compartilhando conhecimentos e discutindo tópicos que definiram os caminhos a serem explorados. Além do impacto neste trabalho, seus conhecimentos e orientações melhoraram minha visão como pesquisador.

Ao Prof. Dr. João Navarro Soares Jr. e ao Prof. Dr. Wang Jiang Chau, membros de minha banca de qualificação, agradeço pelos valiosos apontamentos e sugestões realizadas.

A minha família, agradeço por todo suporte e motivação fornecidos durante todos os momentos da elaboração desta tese. Com a forte base que ela sempre manteve, foi possível enfrentar os momentos mais difíceis, bem como multiplicar os de entusiasmo.

Aos meus colegas e amigos do Laboratório de Sistemas Integráveis, Johanny, Fabian, Sergio, Hugo, José, Lucas e Bruno, agradeço pelas inúmeras conversas, trocas de ideias e contribuições realizadas a respeito dos temas desta tese. Agradeço também a Silvana, que nunca mediu esforços para fazer do laboratório um local acolhedor a todos que lá trabalham.

Aos meus amigos João Manoel, Thomas, Stephan e Alexandre, agradeço pelo suporte durante as etapas da elaboração deste trabalho.

RESUMO

Circuitos integrados analógicos são essenciais em sistemas eletrônicos modernos, sendo responsáveis por tarefas como conversão analógica/digital e digital/analógica, comunicação por radiofrequência, filtragem, etc. O projeto deste tipo de circuito e sistema é de grande complexidade uma vez que deve atender a especificações de desempenho cada vez mais exigentes e ter um tempo de projeto reduzido a fim de não comprometer o tempo total dos projetos de sinal misto. Diversas ferramentas são propostas na literatura visando auxiliar o projetista a aumentar sua produtividade. Apesar disso, devido à forte interligação entre etapas, o fluxo de projeto de circuitos integrados analógicos ainda é, tradicionalmente, realizado utilizando-se apenas cálculos manuais e posterior ajuste fino através de *softwares* de simulação elétrica. Neste trabalho, são estudadas técnicas de síntese de circuitos analógicos utilizando métodos modernos de otimização em nível de circuito e sistema. Após este estudo, é proposto um novo algoritmo de *Simulated Annealing/Simulated Quenching*, incluindo um mecanismo para utilização do operador de *crossover* considerando informações de múltiplos objetivos. É realizada a hibridização entre o algoritmo desenvolvido e um algoritmo de *Particle Swarm Optimization* para criação de um segundo algoritmo capaz de realizar a busca pela fronteira de Pareto. As características dos algoritmos propostos foram elaboradas visando a síntese de circuitos integrados analógicos, no entanto, resultados indicam que eles também têm excelente desempenho em comparação com diversos algoritmos atuais do tipo sem derivada para determinados problemas matemáticos. A generalidade dos métodos modernos de otimização permite que variações da mesma técnica sejam utilizadas em nível de circuito (dimensionamento e polarização de componentes do circuito) e de sistema (tradução de especificações de sistema em especificações de blocos). Dessa forma, são propostas técnicas para a criação de uma ferramenta de síntese em nível de sistema e circuito utilizando métodos modernos de otimização. Uma interface através de arquivos texto de entrada foi desenvolvida para tornar a ferramenta versátil e poder ser utilizada para uma grande variedade de tipos de circuitos eletrônicos. Para validar o algoritmo e a ferramenta na síntese em nível de circuito, foram sintetizados circuitos em tecnologia 0,35 μm , 180 nm e 130 nm. Entre eles, foram sintetizados amplificadores do tipo Miller, amplificadores do tipo *folded cascode* complementar, amplificadores de baixo ruído operando em 2,45 GHz e fontes de referência. Comparações utilizando o teste não paramétrico de Mann-Whitney-Wilcoxon mostram que o algoritmo proposto tem melhor desempenho que os demais algoritmos comparados para os casos estudados. Comparações com projetos manuais e outras ferramentas confirmam a eficácia dos algoritmos e ferramenta. Para validação da ferramenta em nível de sistema, foram sintetizados filtros do tipo G_m -C.

Palavras-chave: CAD. Otimização Global. Circuitos Integrados Analógicos. *Simulated Annealing*.

ABSTRACT

Analog integrated circuits are very important in modern electronic systems, performing tasks such as analog to digital conversion, digital to analog conversion, radio frequency communication, filtering and others. The design of this type of circuit requires attending to several performance specifications as well as a time specification in order to avoid compromising the overall design time of mixed signal projects. Several tools are proposed in the literature in order to aid the designer, however the traditional design flow for analog integrated circuits is usually accomplished using only hand calculations and adjusts through the use of electrical simulators. In this work, techniques for analog design synthesis for circuit and system level are studied. An optimization algorithm is proposed based on Simulated Annealing/Simulated Quenching with a mechanism for using the crossover operator considering multiobjective information. An hybrid algorithm combining the proposed algorithm with Particle Swarm Optimization was created to properly explore the Pareto front. The characteristics of the algorithms are made to enable the synthesis of analog integrated circuits, however, tests indicate they have excellent performance in comparison with many other derivative-free algorithms when applied to certain mathematical problems. The generality of modern optimization methods allow that variations of the same techniques can be used in circuit level (sizing and biasing of circuit components) and in system level (translation of system specifications to block specifications). Therefore, techniques for the creation of a circuit-level and system-level tool are developed. An interface using spice-like text files as inputs is developed to allow the designer to use the tool for a wide range of electronic circuits. In order to validate the proposed algorithms and circuit level tool, circuits were synthesized in 0.35 μm , 180 nm and 130 nm. The synthesized circuits included Miller amplifiers, complementary folded cascode amplifiers, low noise amplifiers operating at 2.45 GHz and voltage reference circuits. Comparisons using the non-parametric Mann-Whitney-Wilcoxon test showed that the proposed algorithm has better performance than the compared algorithms for the studied cases. At the system level, syntheses of G_m -C filters were performed to validate the tool.

Keywords: CAD. Global Optimization. Analog Integrated Circuits. Simulated Annealing.

LISTA DE FIGURAS

Figura 2.1	Relação entre corrente de dreno (i_D) e tensão entre dreno e fonte (v_{DS}) para diversas tensões entre porta e fonte (v_{GS}) de um transistor NMOS sem efeito de modulação de canal.	45
Figura 2.2	Estágio fonte comum.	46
Figura 2.3	Interação da função objetivo com o otimizador e com o problema real. .	47
Figura 2.4	Exemplo de regiões de mínimos locais e mínimo global para um problema de uma variável.	48
Figura 2.5	<i>Tradeoffs</i> em projetos de amplificadores (RAZAVI, 2001).	49
Figura 2.6	Entradas e saídas do sintetizador em nível de circuito.	52
Figura 2.7	Estratégias de síntese em nível de sistema quanto à organização do espaço de projeto.	60
Figura 2.8	Uso de fronteiras de Pareto ou modelos de viabilidade do espaço de critério para síntese de projetos hierárquicos.	61
Figura 2.9	Exemplo de espaço de projeto com três variáveis e espaço de critério unidimensional.	70
Figura 2.10	Exemplo de espaço de projeto com três variáveis e espaço de critério bidimensional.	71
Figura 2.11	Exemplo de uma fronteira de Pareto para um espaço de projeto de duas dimensões com dois objetivos de minimização.	72
Figura 2.12	Pseudocódigo de um algoritmo de Estratégia Evolutiva.	74
Figura 2.13	Pseudocódigo de um Algoritmo Genético (baseado no descrito em Luke (2009)).	75
Figura 2.14	Pseudocódigo de um algoritmo de <i>Simulated Annealing</i>	76
Figura 2.15	Influência dos fatores na velocidade da partícula no PSO.	80
Figura 2.16	Pseudocódigo de um algoritmo de <i>Particle Swarm Optimization</i> com vizinhança totalmente conectada.	81
Figura 3.1	Exemplo de AOF com 3 objetivos baseada no <i>testbench</i> Rastrigin. . . .	90

Figura 3.2	Exemplo de AOF com 3 objetivos baseada no <i>testbench</i> Michalewics. . .	90
Figura 3.3	Representação de indivíduos salvos na memória para efetuar <i>crossover</i> . .	91
Figura 3.4	Ilustração do processo de fuga de mínimos locais através de crossovers. .	92
Figura 3.5	Representação de indivíduos salvos na memória para efetuar <i>crossover</i> . .	92
Figura 3.6	Exemplo de seleção do tipo “roleta” para uma síntese com cinco objetivos.	94
Figura 3.7	Fluxo do algoritmo proposto (<i>Simulated Annealing</i> com <i>crossovers</i>). . .	96
Figura 3.8	Variações de temperatura para o algoritmo proposto.	97
Figura 3.9	Pseudocódigo da versão modificada do SA.	98
Figura 3.10	Exemplo da operação de <i>crossover</i> e ajuste de peso para um problema com dois objetivos.	99
Figura 3.11	Diferentes fases de otimização para o algoritmo híbrido multiobjetivo e ilustração das soluções na fronteira de Pareto após cada fase.	101
Figura 3.12	Porcentagem de problemas mono-objetivo resolvidos por cada algoritmo em função do número de variáveis para 2.000 iterações (considerando a resposta das medianas de cada caso de <i>testbench</i> e número de variáveis).	108
Figura 3.13	Porcentagem total de problemas mono-objetivo resolvidos por cada al- goritmo para 2.000 iterações, onde $f(x_b)$ é a mediana da resposta do algoritmo e $f(x^*)$ é a melhor resposta encontrada para cada caso. Total para: (a) até 60 variáveis; (b) até 30 variáveis.	109
Figura 3.14	Porcentagem de problemas mono-objetivo resolvidos por cada algoritmo ao longo de 2.000 iterações considerando cada otimização individualmente. . .	109
Figura 3.15	Porcentagem de problemas mono-objetivo resolvidos por cada algoritmo em função do número de variáveis para 10.000 iterações (considerando a res- posta das medianas de cada caso de <i>testbench</i> e número de variáveis).	111
Figura 3.16	Porcentagem total de problemas mono-objetivo resolvidos por cada al- goritmo para 10.000 iterações, onde $f(x_b)$ é a mediana da resposta do algoritmo e $f(x^*)$ é a melhor resposta encontrada para cada caso. Total para: (a) até 60 variáveis; (b) até 30 variáveis.	112
Figura 3.17	Porcentagem de problemas mono-objetivo resolvidos por cada algoritmo para 10.000 iterações considerando cada otimização individualmente.	113

Figura 3.18 Porcentagem total de problemas mono-objetivo resolvidos por cada variação do algoritmo proposto para 2.000 iterações, onde $f(x_b)$ é a mediana da resposta do algoritmo e $f(x^*)$ é a melhor resposta encontrada para cada caso. Total para: (a) até 60 variáveis; (b) até 10 variáveis.	114
Figura 3.19 Porcentagem de problemas mono-objetivo resolvidos por cada variação do algoritmo proposto em função do número de variáveis para 2.000 iterações (considerando a resposta das medianas de cada caso de <i>testbench</i> e número de variáveis).	115
Figura 3.20 Porcentagem de problemas multiobjetivo resolvidos por cada algoritmo em função do número de objetivos para 5000 iterações.	119
Figura 3.21 Porcentagem total de problemas multiobjetivo resolvidos por cada algoritmo para 5000 iterações.	120
Figura 3.22 Porcentagem de problemas multiobjetivo resolvidos por cada algoritmo em função do número de objetivos para 10000 iterações.	120
Figura 3.23 Porcentagem total de problemas multiobjetivo resolvidos por cada algoritmo para 10000 iterações.	122
Figura 4.1 Estrutura da ferramenta de síntese em nível de circuito.	126
Figura 4.2 Relações entre arquivos de topologia, <i>testbenches</i> e especificações. . . .	127
Figura 4.3 Exemplo de situação apropriada ao uso de condicionais na geração da <i>netlist</i> : escolha do modelo de resistor em um circuito como o <i>Beta-multiplier</i> . . .	128
Figura 4.4 Descrição de circuito.	128
Figura 4.5 Descrição de parâmetros.	129
Figura 4.6 Descrição de região de operação.	131
Figura 4.7 Descrição das especificações.	131
Figura 4.8 Função custo para uma especificação superior a zero e com objetivo de maximização.	133
Figura 4.9 Função custo para uma especificação superior a zero e com objetivo de minimização.	133
Figura 4.10 Função custo para uma especificação superior a zero e com objetivo de aproximação.	134

Figura 4.11 Exemplo de informações fornecidas ao usuário durante execução da síntese para o caso de uma fonte de referência.	137
Figura 4.12 Topologia do amplificador do tipo Miller.	141
Figura 4.13 <i>Testbenches</i> dos amplificadores para extrair as medições (baseado nos <i>testbenches</i> descritos por Allen e Holberg (2002)).	143
Figura 4.14 Melhor função custo em função do número de iterações para cada algoritmo na síntese do amplificador Miller (cada valor é resultado da mediana entre todas as sínteses para cada algoritmo).	145
Figura 4.15 Topologia do amplificador <i>folded cascode</i> complementar.	146
Figura 4.16 Melhor função custo em função do número de iterações para cada algoritmo na síntese do amplificador <i>folded cascode</i> complementar (cada valor é resultado da mediana entre todas as sínteses para cada algoritmo).	148
Figura 4.17 Topologia do amplificador de baixo ruído para aplicações de banda estreita (SHAEFFER; LEE, 1997).	149
Figura 4.18 Melhor função custo em função do número de iterações para cada algoritmo na síntese do LNA (cada valor é resultado da mediana entre todas as sínteses para cada algoritmo).	152
Figura 4.19 Topologia da fonte de referência (variação da topologia de Huang et al. (2006) utilizada por Mateus et al. (2008)).	153
Figura 4.20 Melhor função custo em função do número de iterações para cada algoritmo na síntese da fonte de referência (cada valor é resultado da mediana entre todas as sínteses para cada algoritmo).	155
Figura 4.21 Curvas simuladas dos parâmetros S em função da frequência para os resultados da síntese do LNA: (a) S_{11} (b) S_{21} (c) S_{12} (d) S_{22}	161
Figura 4.22 Topologia do <i>folded cascode</i> utilizado nos testes de Barros et al. (2010) .	164
Figura 4.23 Variações da topologia de fonte de referência proposta por Huang et al. (2006) permitidas à ferramenta para seleção durante síntese.	167
Figura 4.24 Resistor ajustável digitalmente.	167
Figura 4.25 Variação da tensão de saída em função da temperatura para a fonte de referência em tecnologia 130 nm e saída de 200 mV.	170

Figura 4.26	Variação da tensão de saída em função da tensão de alimentação da fonte de referência em tecnologia 130 nm e saída de 200 mV.	170
Figura 4.27	Análise transiente da fonte de referência em tecnologia 130 nm e saída de 200 mV com alimentação variando de 0 a 1,2V em 100ns.	171
Figura 4.28	Tensão de referência para 1000 simulações de Monte Carlo para fonte de referência em tecnologia 130 nm e saída de 200 mV com e sem ajuste digital. .	171
Figura 4.29	Coeficiente de temperatura para 1000 simulações de Monte Carlo para fonte de referência em tecnologia 130 nm e saída de 200 mV com e sem ajuste digital.	172
Figura 4.30	PSRR para 1000 simulações de Monte Carlo para fonte de referência em tecnologia 130 nm e saída de 200 mV com e sem ajuste digital.	172
Figura 4.31	Topologia proposta por Banba et al. (1999) (circuito de start-up não mostrado na figura).	173
Figura 4.32	Variação de tensão de saída em função da temperatura da fonte de referência de 600 mV em tecnologia 130 nm.	175
Figura 4.33	Variação de tensão de saída em função da tensão de alimentação da fonte de referência de 600 mV em tecnologia 130 nm.	176
Figura 4.34	Análise transiente da fonte de referência de 600 mV em tecnologia 130 nm com tensão de alimentação em degrau de 0 a 1,2V em 100ns.	176
Figura 4.35	<i>Layout</i> da fonte de referência com saída de 600 mV projetada em tecnologia 130 nm.	177
Figura 4.36	Tensão de referência para 1000 simulações de Monte Carlo para fonte de referência com saída projetada para 600 mV em tecnologia 130 nm com e sem extração de <i>layout</i>	178
Figura 4.37	Coeficiente de Temperatura para 1000 simulações de Monte Carlo para fonte de referência com saída de 600 mV em tecnologia 130 nm com e sem extração de <i>layout</i>	178
Figura 4.38	PSRR para 1000 simulações de Monte Carlo para fonte de referência com saída de 600 mV em tecnologia 130 nm com e sem extração de <i>layout</i> . . .	179
Figura 4.39	Variação de tensão de saída em função da temperatura da fonte de referência de 600 mV em tecnologia 180 nm.	180

Figura 4.40	Variação de tensão de saída em função da tensão de alimentação da fonte de referência de 600 mV em tecnologia 180 nm.	180
Figura 4.41	Análise transiente da tensão de referência com tensão de alimentação em degrau de 0 a 1,8V em 100ns da fonte de referência de 600 mV em tecnologia 180 nm.	181
Figura 4.42	<i>Layout</i> da fonte de referência com saída de 600 mV projetada em tecnologia 180 nm.	181
Figura 4.43	Tensão de referência para 1000 simulações de Monte Carlo para fonte de referência com saída de 600 mV em tecnologia 180 nm.	182
Figura 4.44	Coeficiente de temperatura para 1000 simulações de Monte Carlo para fonte de referência com saída de 600 mV em tecnologia 180 nm.	182
Figura 4.45	PSRR para 1000 simulações de Monte Carlo para fonte de referência com saída de 600 mV em tecnologia 180 nm.	183
Figura 4.46	Exemplo de exposição de resultados da fronteira de Pareto para um amplificador <i>folded cascode</i> complementar.	185
Figura 4.47	Aproximações da fronteira de Pareto do Amplificador Miller.	186
Figura 4.48	Aproximações da fronteira de Pareto do amplificador <i>folded cascode</i> complementar.	187
Figura 5.1	Exemplos de possibilidades de uso da fronteira de Pareto para a síntese.	195
Figura 5.2	Descrição da ligação das fronteiras de Pareto com o nível de sistema dentro do arquivo de topologia do sistema.	198
Figura 5.3	Escolha da solução da fronteira de Pareto que servirá para calcular o custo de viabilidade da solução proposta para (a) caso em que a solução proposta é inviável e (b) caso em que a solução proposta é viável.	200
Figura 5.4	Integrador G_m -C.	201
Figura 5.5	Estrutura de uma seção de segunda ordem.	201
Figura 5.6	Estrutura de uma seção de um polo.	202
Figura 5.7	Exemplo de filtro de 6ª ordem composto por 3 seções de segunda ordem.	202
Figura 5.8	Modelo de OTA para pequenos sinais com impedâncias de entrada e saída finitas.	203

Figura 5.9	Modelo de OTA para pequenos sinais com fontes de ruído referidas à entrada e impedâncias de entrada e saída finitas.	203
Figura 5.10	Inclusão de ruído térmico e <i>flicker</i> no modelo do OTA em <i>Verilog-A</i> . . .	204
Figura 5.11	Intervalos de linearidade da corrente em função da tensão de entrada no OTA.	204
Figura 5.12	Circuito OTA e CMFB com chaves para realizar caracterização.	206
Figura 5.13	Fronteira de Pareto do OTA para o filtro com frequência de corte de 1,2 MHz (CMOS 0,35 μm).	211
Figura 5.14	Fronteira de Pareto do OTA sem variáveis compartilhadas para o filtro com frequência de corte de 20 MHz em tecnologia 0,35 μm	216
Figura 5.15	Fronteira de Pareto do OTA para o filtro com frequência de corte em 20 MHz (CMOS 180 nm).	219

LISTA DE TABELAS

Tabela 2.1	Alguns métodos de otimização.	67
Tabela 3.1	<i>Testbenches</i> matemáticos para comparação entre algoritmos mono-objetivo, onde $f(x)$ é a função principal, n é o número de variáveis, $i = 1, \dots, n$ é o índice das variáveis e p é o índice das sub funções $f_p(x)$	105
Tabela 3.2	Mediana de 50 otimizações dos algoritmos comparados para cada caso dos <i>testbenches</i> matemáticos mono-objetivo de 2.000 iterações.	106
Tabela 3.3	Mediana de 50 otimizações dos algoritmos comparados para cada caso dos <i>testbenches</i> matemáticos mono-objetivo de 10.000 iterações.	110
Tabela 3.4	Conjunto de testes ZDT (ZITZLER et al., 2000).	116
Tabela 3.5	Conjunto de testes DTLZ (DEB et al., 2002).	117
Tabela 3.6	Mediana dos resultados dos algoritmos comparados para os <i>testbenches</i> matemáticos multi-objetivo com 5000 iterações.	118
Tabela 3.7	Mediana dos resultados dos algoritmos comparados para os <i>testbenches</i> matemáticos multi-objetivo com 10000 iterações.	121
Tabela 4.1	Descrição de entradas no arquivo de configuração.	125
Tabela 4.2	Custo associado às restrições de região de operação.	136
Tabela 4.3	Melhores resultados dos algoritmos comparados para a síntese do circuito do amplificador Miller com até 2000 iterações.	142
Tabela 4.4	Dimensões dos melhores resultados obtidos pelos algoritmos comparados para a síntese do circuito do amplificador Miller com até 2000 iterações. . .	144
Tabela 4.5	Resultados dos algoritmos comparados para a síntese do Amplificador Miller.	144
Tabela 4.6	Melhores resultados dos algoritmos comparados para a síntese do circuito do amplificador <i>folded cascode</i> complementar com até 2000 iterações. . .	146
Tabela 4.7	Dimensões dos melhores resultados obtidos pelos algoritmos comparados para a síntese do circuito do amplificador <i>folded cascode</i> complementar com até 2000 iterações.	147

Tabela 4.8	Resultados dos algoritmos comparados para a síntese do amplificador <i>folded cascode</i> complementar.	147
Tabela 4.9	Melhores resultados dos algoritmos comparados para a síntese do circuito do LNA com até 2000 iterações.	150
Tabela 4.10	Dimensões dos melhores resultados obtidos pelos algoritmos comparados para a síntese do circuito do LNA com até 2000 iterações.	150
Tabela 4.11	Resultados dos algoritmos comparados para a síntese do LNA.	151
Tabela 4.12	Melhores resultados dos algoritmos comparados para a síntese do circuito da Fonte de Referência com até 2000 iterações.	153
Tabela 4.13	Dimensões dos melhores resultados obtidos pelos algoritmos comparados para a síntese do circuito da Fonte de Referência com até 2000 iterações. . .	154
Tabela 4.14	Resultados dos algoritmos comparados para a síntese do circuito fonte de referência.	154
Tabela 4.15	Resultados da síntese para o amplificador Miller em comparação com resultado manual realizado por Cortes e Bampi (2006).	156
Tabela 4.16	Dimensões dos transistores MOS e parâmetros para o circuito amplificador Miller encontrado na literatura e deste trabalho sem análise de <i>corners</i>	156
Tabela 4.17	Resultados de síntese para o amplificador Miller em diversos <i>corners</i> . .	157
Tabela 4.18	Dimensões dos transistores MOS e parâmetros para o circuito amplificador Miller encontrado na literatura e neste trabalho considerando <i>corners</i> do processo CMOS 0,35 μm	158
Tabela 4.19	Intervalo das variáveis e valores encontrados.	159
Tabela 4.20	Capacitâncias e indutâncias resultantes.	159
Tabela 4.21	Métricas de desempenho simuladas em comparação com LNAs publicados na literaturas.	160
Tabela 4.22	Métricas de desempenho do LNA considerando <i>corners</i> de processo e temperatura para as soluções encontradas.	161
Tabela 4.23	Especificações da fonte de referência e resultados encontrados neste trabalho e em trabalho utilizando programação geométrica	163
Tabela 4.24	Dimensões dos transistores MOS e resistores para o circuito fonte de referência.	163

Tabela 4.25	Dimensões finais das variáveis para o <i>folded cascode</i> para cada ferramenta após um máximo de 2500 iterações. Valores das ferramentas FRIDGE e GENOM apresentados por Barros et al. (2010).	164
Tabela 4.26	Especificações do projeto do amplificador <i>folded cascode</i> e resultados expostos por Barros et al. (2010) e os alcançados por este trabalho.	165
Tabela 4.27	Especificações da fonte de referência em tecnologia 130 nm com saída de 200 mV.	168
Tabela 4.28	Dimensões dos dispositivos para o circuito fonte de referência em tecnologia 130 nm com saída de 200 mV.	169
Tabela 4.29	Resultados nominais (TT) e para <i>corners</i> de processo para a fonte de referência em tecnologia 130 nm e saída de 200 mV (nesta etapa, o ajuste de R_3 é realizado diretamente, não através de chaves).	169
Tabela 4.30	Especificações da fonte de referência de 600 mV em 130 nm.	174
Tabela 4.31	Dimensões dos dispositivos para o circuito fonte de referência de 600 mV em tecnologia 130 nm.	174
Tabela 4.32	Resultados nominais para o circuito fonte de referência de 600 mV em tecnologia 130 nm.	175
Tabela 4.33	Dimensões dos dispositivos para o circuito fonte de referência de 600 mV em 180 nm.	179
Tabela 4.34	Resultados nominais para o circuito Fonte de Referência de 600 mV em 180 nm.	179
Tabela 4.35	Objetivos de projeto e especificações mínimas para a síntese multiobjetivo.	185
Tabela 4.36	Resultados de $1 - H_i$ (onde H_i é o volume do hipercubo) para as sínteses multiobjetivo do amplificador Miller.	187
Tabela 4.37	Resultados de $1 - H_i$ (onde H_i é o volume do hipercubo) para as sínteses multiobjetivo do amplificador <i>folded cascode</i> complementar.	188
Tabela 4.38	Tempos de síntese dos testes realizados em nível de transistores.	188
Tabela 5.1	Condições para gerar coeficientes da <i>spline</i> .	205
Tabela 5.2	Especificação de filtro G_m -C passa-baixas com frequência de corte em 1,2 MHz (CMOS 0,35 μm).	208

Tabela 5.3	Especificações mínimas do OTA para geração da fronteira de Pareto do filtro com frequência de corte de 1,2 MHz (CMOS 0,35 μm).	209
Tabela 5.4	Características dos parâmetros do OTA para criação da fronteira de Pareto do filtro com frequência de corte de 1,2 MHz (CMOS 0,35 μm).	210
Tabela 5.5	Relação entre variáveis de projeto e dimensões dos transistores do OTA para criação da fronteira de Pareto do filtro com frequência de corte de 1,2 MHz (CMOS 0,35 μm	210
Tabela 5.6	Descrição dos parâmetros em nível de sistema para síntese do filtro com frequência de corte de 1,2 MHz (CMOS 0,35 μm).	212
Tabela 5.7	Especificações dos blocos OTA após síntese do filtro G_m -C com frequência de corte de 1,2 MHz (CMOS 0,35 μm).	213
Tabela 5.8	Especificações e resultados da síntese do filtro G_m -C com frequência de corte de 1,2 MHz (CMOS 0,35 μm).	213
Tabela 5.9	Comparação dos resultados da síntese do filtro G_m -C com frequência de corte de 1,2 MHz entre a ferramenta deste trabalho e a ferramenta de Hincapié (2010) em tecnologia 0,35 μm	214
Tabela 5.10	Especificação de filtro G_m -C passa-baixas com frequência de corte de 20 MHz em tecnologia 0,35 μm	214
Tabela 5.11	Descrição dos parâmetros em nível de sistema para síntese de caráter exploratório do filtro com frequência de corte em 20 MHz (CMOS 0,35 μm).	217
Tabela 5.12	Parâmetros dos OTAs resultantes da síntese do filtro G_m -C com frequência de corte de 20 MHz (CMOS 0,35 μm).	218
Tabela 5.13	Multiplicidade e sequência dos OTAs resultantes da síntese do filtro G_m -C com frequência de corte de 20 MHz (CMOS 0,35 μm).	218
Tabela 5.14	Especificações e resultados da síntese do filtro G_m -C com frequência de corte de 20 MHz (CMOS 0,35 μm).	219
Tabela 5.15	Parâmetros dos OTAs resultante da síntese do filtro G_m -C com frequência de corte de 20 MHz (CMOS 180 nm).	220
Tabela 5.16	Multiplicidade dos OTAs resultantes das sínteses do filtro G_m -C com frequência de corte de 20 MHz (CMOS 180 nm).	220

Tabela 5.17 Especificações e resultados da síntese do filtro G_m -C com frequência de corte de 20 MHz em tecnologia 180 nm.	221
Tabela 5.18 Comparação entre resultados da síntese do filtro G_m -C passa-baixas com frequência de corte de 20 MHz e outros trabalhos (baseado na tabela 5.6 de (SAARI, 2012)).	221
Tabela 5.19 Figuras de Mérito para o resultado da síntese do filtro G_m -C e outros trabalhos.	221
Tabela 5.20 Descrição dos parâmetros em nível de sistema para síntese de caráter exploratório do filtro com frequência de corte de 20 MHz (CMOS 180 nm). . .	223
Tabela 5.21 Parâmetros dos OTAs resultantes da conversão de especificações de sistema para circuito do filtro G_m -C com frequência de corte de 20 MHz (CMOS 180 nm).	224
Tabela 5.22 Resultado da multiplicidade dos OTAs resultantes da conversão de especificações de sistema para circuito do filtro G_m -C com frequência de corte de 20 MHz (CMOS 0,35 μ m).	225
Tabela 5.23 Especificações e resultados da síntese do filtro G_m -C com frequência de corte de 20 MHz com variáveis não compartilhadas (CMOS 0,35 μ m).	225

LISTA DE ABREVIATURAS E SIGLAS

AG	Algoritmos Genéticos
AMS	Sinal Misto e Analógico (do inglês, <i>Analog Mixed-Signal</i>)
AOF	Função Objetivo Agregativa (do inglês, <i>Aggregate Objective Function</i>)
ASA	Recozimento Simulado Adaptativo (do inglês, <i>Adaptive Simulated Annealing</i>)
CAD	Projeto Auxiliado por Computador (do inglês, <i>Computer-Aided Design</i>)
CI	Circuito Integrado
CMRR	Relação de Rejeição do Modo Comum (do inglês, <i>Common Mode Rejection Ratio</i>)
CMFB	Realimentação de Modo Comum (do inglês, <i>Common Mode FeedBack</i>)
DOE	Projeto/Planejamento de Experimentos (do inglês, <i>Design of Experiments</i>)
DH	Casa de Projetos (do inglês, <i>Design House</i> ou <i>Design Center</i>)
DUT	Dispositivo Sob Teste (do inglês, <i>Device Under Test</i>)
IP	Propriedade Intelectual (do inglês, <i>Intellectual Property</i>)
LNA	Amplificador de Baixo Ruído (do inglês, <i>Low Noise Amplifier</i>)
OpAmp	Amplificador Operacional (do inglês, <i>Operational Amplifier</i>)
OTA	Amplificador Operacional de Transcondutância (do inglês, <i>Operational Transconductance Amplifier</i>)
PSO	Otimização por Enxame de Partículas (do inglês, <i>Particle Swarm Optimization</i>)
PSRR	Relação de Rejeição da Fonte de Alimentação (do inglês, <i>Power Supply Rejection Ratio</i>)
SA	Recozimento Simulado (do inglês, <i>Simulated Annealing</i>)
SOC	Sistema em Chip (do inglês, <i>System-on-Chip</i>)
SVM	Máquina de Vetores de Suporte (do inglês, <i>Support Vector Machine</i>)
SR	Taxa de Variação (do inglês, <i>Slew Rate</i>)
UGF	Frequência de Ganho Unitário (do inglês, <i>Unity Gain Frequency</i>)
PLL	Oscilador em Malha de Controle de Fase (do inglês, <i>Phase-Locked Loop</i>)
VCO	Oscilador Controlado por Tensão (do inglês, <i>Voltage Controlled Oscillator</i>)

SUMÁRIO

1	Introdução	33
1.1	Considerações Iniciais	33
1.2	Justificativa	37
1.3	Objetivo	38
1.4	Contribuições da Tese	38
1.5	Estrutura do Trabalho	39
1.6	Publicações	40
2	Revisão Teórica	43
2.1	Projeto Analógico e Estado da Arte de Síntese	43
2.1.1	Variáveis de Projeto	44
2.1.2	Medidas de Desempenho de Projeto	46
2.1.3	Função Objetivo	47
2.1.4	Problemas com Múltiplos Objetivos em Projeto de Circuitos Analógicos	48
2.1.5	Espaço de Projeto	49
2.1.6	Características de uma Ferramenta CAD de Síntese Analógica	50
2.1.7	Estado da Arte de Síntese em Nível de Circuito	51
2.1.7.1	Abordagem Baseada em Conhecimento	52
2.1.7.2	Abordagem Baseada em Otimização	53
2.1.8	Estado da Arte de Síntese em Nível de Sistema	59
2.1.8.1	Modelos de Viabilidade e Desempenho	62
2.1.8.2	Conversão de Especificações em Nível de Sistema para Nível de Circuito	66
2.2	Métodos Modernos de Otimização	67
2.2.1	Formalização de Problemas Mono-Objetivo	69

2.2.2	Formalização de Problemas Multiobjetivo	69
2.2.2.1	Fronteira de Pareto	71
2.2.3	Algoritmos Mono-Objetivo	72
2.2.3.1	Estratégia Evolutiva	73
2.2.3.2	Algoritmos Genéticos (AG)	74
2.2.3.3	Simulated Annealing	75
2.2.3.4	<i>Particle Swarm Optimization</i> (PSO)	80
2.2.4	Algoritmos Multiobjetivo	81
2.2.4.1	Técnicas Utilizando Funções Agregativas	81
2.2.4.2	Técnicas Evolutivas Multiobjetivo	82
2.2.4.3	PSO Multiobjetivo	83
2.3	Resumo do Capítulo	83

3 Algoritmos de Otimização Propostos 85

3.1	Descrição do Algoritmo Proposto para Problemas Mono-objetivo	86
3.1.1	Passo Adaptativo	87
3.1.2	Multiobjetivação	88
3.1.3	Operador de Crossover em Algoritmo Não-Populacional	89
3.1.4	Seleção de Soluções Âncoras	93
3.1.5	Ajuste Automático de Pesos	94
3.1.6	Fluxograma do Algoritmo	95
3.2	Descrição do Algoritmo Proposto para Problemas Multiobjetivo	97
3.3	Testes	101
3.3.1	Descrição do Método de Teste	102
3.3.2	Algoritmos Comparados para o Caso Mono-Objetivo	103
3.3.3	Testbenches para o Caso Mono-Objetivo	104
3.3.4	Resultados para Problemas Matemáticos Mono-Objetivo	105

3.3.4.1	Número de Iterações Máximo Igual a 2.000	105
3.3.4.2	Número de Iterações Máximo Igual a 10.000	111
3.3.5	Testes dos Recursos do Algoritmo	112
3.3.6	Algoritmos Comparados para o Caso Multiobjetivo	115
3.3.7	Testbenches para o Caso Multiobjetivo	116
3.3.8	Resultados para Problemas Matemáticos Multiobjetivo	117
3.3.8.1	Número de Iterações Máximo Igual a 5.000	117
3.3.8.2	Número de Iterações Máximo Igual a 10.000	119
3.4	Resumo do Capítulo	122
4	Síntese em Nível de Transistores	123
4.1	Estrutura da Ferramenta de Síntese em Nível de Circuito	123
4.1.1	Blocos de Leitura	124
4.1.2	Blocos de Avaliação	130
4.1.3	Cálculo da Função Custo	132
4.1.4	Bloco de Exposição de Resultados	136
4.2	Síntese Considerando Monte Carlo e <i>Corners</i> de Processo	137
4.3	Testes	138
4.3.1	Comparação entre Algoritmos	139
4.3.2	Amplificador Miller	141
4.3.3	Amplificador Folded Cascode Complementar	145
4.3.4	Amplificador de Baixo Ruído (LNA)	148
4.3.5	Fonte de Referência	152
4.4	Comparação com Projetos Manuais e Outras Técnicas de Otimização	155
4.4.1	Amplificador Miller	155
4.4.2	Síntese de Amplificador Miller considerando <i>Corners</i> de Processo	157
4.4.3	Amplificador de Baixo Ruído (LNA)	157

4.4.4	Fonte de Referência	162
4.4.5	Amplificador Folded Cascode	162
4.5	Projeto de Fonte de Referência em 130 nm com Variações de Topologia e com Ajuste Digital	166
4.6	Projeto de Fonte de Referência com saída de 600 mV em 130 nm	173
4.7	Projeto de Fonte de Referência com saída de 600 mV em 180 nm	178
4.8	Exploração da Fronteira de Pareto	182
4.8.1	Restrições para Algoritmos Multiobjetivo	183
4.8.2	Exposição dos Resultados Multiobjetivo	184
4.8.3	Testes	184
4.9	Tempos de Síntese	187
4.10	Resumo do Capítulo	188

5 Síntese em Nível de Sistema 191

5.1	Procedimento de Síntese em Nível de Sistema	191
5.1.1	Avaliação da Viabilidade de Especificações através da Fronteira de Pareto	192
5.2	Processo de Síntese em Nível de Sistema	193
5.2.1	Sistema de Organização dos Parâmetros/Variáveis	194
5.2.1.1	Tipos de Parâmetros em Nível de Sistema	195
5.2.1.2	Tipos de Parâmetros em Nível de Transistores	196
5.2.1.3	Características da Busca Usando Soluções Indexadas e Conectadas da Fronteira de Pareto	196
5.2.2	Modificação nas Informações de Entrada do Usuário	197
5.2.3	Custo de Viabilidade	198
5.3	Síntese de Filtros G_m -C	200
5.3.1	Operação e Estrutura de Filtros G_m -C	200
5.3.1.1	Modelagem do OTA	202
5.4	Utilização da ferramenta para diferentes estratégias de projeto	206

5.5	Síntese de um Filtro G_m -C Passa-Baixas com Frequência de Corte de 1,2 MHz em Tecnologia 0,35 μm	207
5.6	Síntese de um Filtro G_m -C Passa-Baixas com Frequência de Corte de 20 MHz em Tecnologia 0,35 μm	211
5.7	Síntese de um Filtro G_m -C Passa-Baixas com Frequência de Corte de 20 MHz em Tecnologia CMOS 180 nm	217
5.8	Conversão de Especificações de Sistema em Especificação de Bloco utilizando a Fronteira de Pareto apenas como Referência	222
5.9	Resumo do Capítulo	224
6	Conclusões	227
	Referências	233
	Apêndice A - Exemplos de Arquivos de Entrada	245
A.1	Exemplo de Arquivo de Configuração de um Amplificador <i>Miller</i>	245
A.2	Exemplo de Arquivo de Topologia de um Amplificador <i>Miller</i>	246
A.3	Exemplo de Arquivo de Especificações de um Amplificador <i>Miller</i>	247
A.4	Exemplo de Arquivo de <i>Testbench</i> no Domínio da Frequência de um Amplificador <i>Miller</i>	247
A.5	Exemplo de Seção de Descrição da Topologia da Fronteira de Pareto e Seção de Descrição de Variáveis em Nível de Sistema de um Filtro G_m -C	248

1 INTRODUÇÃO

1.1 Considerações Iniciais

Os circuitos integrados (CIs) tornaram-se ubíquos, estando presentes em quase todos os aparelhos/sistemas modernos e abrangendo áreas como transporte, saúde, comunicação, defesa e entretenimento. A utilização de CIs, atualmente, é tão vasta que dificilmente é encontrada uma área da atividade humana que não faça uso de um ou vários equipamentos contendo-os como peça-chave para seu funcionamento.

Diante desta vasta abrangência, a indústria de semicondutores recebeu grande avanço na tecnologia de manufatura e automação de projeto nas últimas décadas. Atualmente, projetos de alta complexidade são desenvolvidos e circuitos integrados com bilhões de transistores são produzidos. Tal avanço, no entanto, requer que um número crescente de fatores tenha que ser levado em consideração durante a fase de projeto. Consequentemente, um número maior de técnicas e tipos de ferramentas são necessários para satisfazer as demandas de projeto.

As técnicas e ferramentas de projeto estão intimamente ligadas a forma como os circuitos integrados tratam os sinais, que pode ser tanto digital quanto analógica. Na primeira forma são utilizadas apenas representações binárias dos sinais, enquanto na segunda eles são representados de forma contínua. Os circuitos integrados analógicos são responsáveis por aplicações como conversores analógico/digitais e digitais/analógicos, amplificadores, reguladores de tensão, filtros, circuitos de rádio frequência, entre outros. Ambos os tipos de circuito, em conjunto, fazem parte dos sistemas de sinal misto.

Estas diferentes formas de tratar os sinais resultam em abordagens de projeto distintas entre si. Enquanto no projeto digital é possível realizar intenso uso de abstrações dos dispositivos físicos, no projeto analógico estes dispositivos tem que ser considerados detalhadamente para que o desempenho dos circuitos seja corretamente avaliado.

Em decorrência do vasto uso de abstrações em projeto de CIs digitais, gerou-se um ambiente propício à criação de ferramentas de auxílio ao projetista. Assim, o projeto de circuitos integrados digitais teve sua produtividade acelerada, permitindo que ocorresse uma redução do

tempo de projeto e uma maior tratabilidade de problemas complexos. Por outro lado, o projeto de circuitos integrados analógicos ainda é tradicionalmente realizado utilizando como base cálculos manuais e, posteriormente, simulação e ajuste fino através de *softwares* de simulação elétrica. Devido à complexidade das relações entre as variáveis dos circuitos e suas especificações, a realização da função de projetista analógico requer vasta experiência e profundo conhecimento teórico a respeito do funcionamento de semicondutores e de técnicas de projeto.

Enquanto projetistas digitais são capazes de projetar *chips* com bilhões de transistores em menos de um ano, um time de projetistas analógicos pode requerer tempo similar para projetar um *chip* consistindo de algumas centenas de transistores (BALKIR et al., 2003). Caso o projeto possua tanto seções analógicas quanto digitais, o gargalo em tempo de projeto torna-se o analógico, comprometendo o *time to market*¹ de todo o sistema. Dessa forma, um aumento de produtividade neste tipo de projeto é refletido na indústria de semicondutores como um todo.

Em projetos analógicos, a definição de um fluxo automatizado bem estabelecido, que permita o mesmo avanço do domínio digital, ainda não foi completamente realizada. As dificuldades em automatizar o projeto analógico, ou mesmo algumas partes dele, deve-se ao fato de que este raramente ocorre através de uma progressão de passos de projeto discretos, verificados formalmente e que levam o projeto do conceito até o silício (VANDENBUSSCHE et al., 2003). Há um limiar nebuloso entre as etapas de projeto, e considerações em nível de circuito necessitam ser observadas mesmo para tomar decisões em nível de sistema. Ainda, há um grande número de graus de liberdade no projeto sem uma inter-relação clara. Isto significa que os chamados *tradeoffs*, ou compromissos de projeto, nem sempre se apresentam de forma evidente. Estas questões, somadas à forte realimentação entre as etapas, fazem com que apenas projetistas muito experientes sejam capazes de lidar com a totalidade do projeto de forma eficiente.

Neste cenário, a utilização de ferramentas de projeto assistido para incremento da produtividade dos projetistas de circuitos analógicos se torna fundamental para melhorar o fluxo de projeto. Nos trabalhos que buscam fluxos de projeto mais otimizados, ferramentas de CAD² são mais exploradas do que no fluxo tradicional, de forma a aumentar a qualidade do projeto e diminuir o esforço do projetista em reprojotos. Isso significa que o projetista poderá concentrar seus esforços em tarefas de maior abstração e em projetos desafiadores.

No *International Technology Roadmap for Semiconductors* (ITRS, 2011), a síntese de circuitos analógicos é indicada como um dos desafios a serem enfrentados para aumentar a produtividade de projeto. Este documento aponta diversas questões ainda não satisfatoriamente resolvidas para a realização da síntese, como a exploração do espaço de projeto e otimização

¹Tempo necessário entre a concepção de um produto e a sua disponibilização para a venda

²*Computed Aided Design*, do inglês, significando Projeto Auxiliado por Computador

discreta (ex.: uso de *fingers*, grade de manufatura). O ITRS também indica a realização de projeto hierárquico em nível de sistema como um ponto a ser considerado para aumento de produtividade.

Com o auxílio de ferramentas de CAD para automação e auxílio de projeto, definições de projetos que são abstraídas da tecnologia em questão podem ser realizadas e convertidas para dimensionamento de transistores apenas quando as especificações e tecnologia já estiverem definidas. Esta conversão de conceitos de projeto em dimensionamento de transistores apenas em um estágio final do fluxo permite o maior reuso de blocos. O reuso, neste contexto, é consequência da capacidade de portar um projeto que foi desenvolvido para um determinado cenário (conjunto de especificações e tecnologia) para outro sem a necessidade de reprojetar. Este recurso possibilita que o esforço despendido pela equipe (assim como o custo da empresa/organização financiadora) não seja desperdiçado. A importância desta capacidade se deve a ser comum a ocorrência de trocas de tecnologia ao longo da vida útil de um centro de projetos. Em tais trocas, todos os *hard-IPs*³ precisam ser reprojetados, uma vez que a troca de tecnologia normalmente significa que o projeto tem que recomeçar do início. Nestes casos, em um fluxo tradicional, um projetista analógico experiente (ou um grupo de projetistas) precisa ser alocado para cumprir esta tarefa.

A síntese de circuitos analógicos em nível de circuitos consiste no dimensionamento e polarização automática dos componentes de um bloco a fim de atingir as especificações necessárias. Este processo pode ser abordado de diversas formas, podendo ser dividido em abordagens baseadas em conhecimento e abordagens baseadas em otimização. A abordagem baseada em conhecimento utiliza a experiência prévia de projetistas para criar um plano de síntese. Assim, os passos que foram utilizados para criação de um bloco uma vez podem ser reaproveitados para criação de blocos similares através do compartilhamento do mesmo plano. As abordagens baseadas em otimização convertem o processo de síntese em um problema que possa ser solucionado através de métodos de otimização. Nesse caso, um algoritmo de otimização realiza alterações nas variáveis de entrada do problema a fim de gradualmente encontrar soluções que atendam o objetivo final (no caso, atender às especificações do circuito).

Em nível de sistema, a conversão/tradução de especificações de sistema para os blocos costuma ser, em um fluxo tradicional, um processo fortemente dependente da experiência do projetista na elaboração do sistema específico. Por outro lado, caso a síntese de um sistema seja realizada de forma plana (sem hierarquias) em nível de circuito, o número de variáveis envolvida torna o processo proibitivo. A fim de tornar esta síntese um processo passível de ser

³propriedades intelectuais que estão fortemente atreladas a uma determinada tecnologia, ou seja, implementadas em *hardware* até a dimensão física (*layout*). Em contraposição, *soft-IPs* são os projetos que podem ser portados a outras tecnologias por terem sido descritos em uma linguagem mais abstrata, como VHDL, Verilog ou Verilog-A.

exercido automaticamente em nível de sistema, são criados modelos de viabilidade e modelos de desempenho. Estes modelos de blocos, normalmente criados a partir de simulações em nível de circuito, são utilizados para rapidamente obter informações a respeito da viabilidade de sintetizar um bloco com determinadas especificações e seu desempenho esperado. Os modelos de viabilidade informam se um determinado conjunto de especificações é sintetizável ou não. Os modelos de desempenho mapeiam as variáveis de entrada de projeto (dimensões de componentes, tensões e correntes de polarização) para as especificações do bloco (ganho, margem de fase, área,...) sem necessidade de simulação do circuito. A existência destes modelos poupa o tempo da tentativa de síntese de um bloco em nível de circuito quando suas especificações não podem ser atingidas.

Em qualquer um dos casos, seja na síntese em nível de circuito ou sistema, as aplicações modernas tendem a favorecer o uso de abordagens baseadas em otimização devido a sua versatilidade. Ao utilizar este tipo de abordagem, o centro do processo de síntese passa a ser o algoritmo de otimização, que tentará escolher os valores das variáveis de entrada para que o circuito/sistema atenda suas especificações. O algoritmo de otimização será responsável por definir os passos na busca pelo circuito ótimo, fazendo o melhor uso possível da informação coletada durante esta busca para acelerar a síntese. Uma melhora no otimizador, por consequência, tem potencial de melhorar qualquer ferramenta de síntese baseada em otimização.

Neste trabalho, são desenvolvidos algoritmos de otimização para atender às necessidades de síntese de circuitos integrados analógicos, bem como uma ferramenta computacional para a realização da síntese de circuitos e sistemas. Um algoritmo de otimização global é desenvolvido para sínteses que necessitam apenas uma solução final e um algoritmo de exploração multiobjetivo é desenvolvido para obtenção de diversas soluções finais. Em nível de circuito, os algoritmos são implementados com foco em realizar o dimensionamento e polarização de componentes do circuito. Em nível de sistema, o algoritmo de otimização global é utilizado para conversão de especificações de sistema para especificações de blocos utilizando a fronteira de Pareto como guia de projeto.

Os algoritmos propostos são tipo livre de derivada ⁴, são baseados em *Simulated Annealing/Simulated Quenching* e utilizam o operador de *crossover* dos algoritmos genéticos para evitar mínimos locais e realizam alteração automática de pesos dos objetivos. Apesar do *Simulated Annealing* ser utilizado classicamente através da conversão de vários objetivos em apenas um usando uma função objetivo agregativa (AOF, do inglês, *Aggregate Objective Function*), neste trabalho faz-se uso da informação proveniente dos diversos objetivos para que o meca-

⁴o termo em inglês *derivative-free optimization* é utilizado para descrever algoritmos que requerem apenas a existência da função custo para operarem, não necessitando da disponibilidade das derivadas de cada solução (RIOS; SAHINIDIS, 2013)

nismo de *crossover* dos algoritmos genéticos possa ser utilizado na fuga de mínimos locais. No algoritmo com foco em apenas uma solução final, esta abordagem é utilizada diretamente para obtenção do mínimo global. No algoritmo que busca múltiplas soluções, a variação do *Simulated Annealing* proposta é utilizada de forma integrada com o algoritmo populacional *Particle Swarm Optimization*. Com essa combinação, são buscadas soluções da fronteira de Pareto, que representam os *tradeoffs* de projeto.

A ferramenta de síntese foi desenvolvida para propiciar flexibilidade tanto na síntese em nível de circuito quanto em nível de sistema. Através dela, descrições dos problemas de síntese podem ser realizadas de forma ágil e considerando as particularidades de cada nível de abstração. Em nível de circuito, o projetista pode elaborar suas *netlists*⁵ de forma similar à tradicional, porém com possibilidade de adição de informações auxiliares à otimização, como as regiões de operação ou coeficiente de inversão em que cada transistor vai operar, tipos e intervalos de variáveis, bem como tem liberdade para escolha de quaisquer medições realizáveis com o simulador elétrico. O projetista pode escolher considerar guiar a síntese à robustez do projeto através do uso de simulações de *corners* do processo ou do tipo Monte Carlo. Circuitos que são utilizados para posteriores sínteses em nível de sistema podem ter suas fronteiras de Pareto exploradas. Em nível de sistema, foi propiciada uma forma facilitada para incorporação das fronteiras de Pareto dos blocos, tipos de variáveis específicas para lidar com as diferentes configurações de uso das fronteiras e uma nova técnica de conversão de especificações de nível de sistema para nível de circuito.

1.2 Justificativa

As justificativas para o desenvolvimento de algoritmos e ferramentas de CAD para síntese de circuitos analógicos envolvem diversas esferas. Na esfera industrial, é um potencializador do rendimento dos projetistas. O desenvolvimento de métodos que abranjam a síntese desde o nível de sistema até o nível de circuito fornece aos projetistas maior capacidade de analisar questões críticas de projeto. Dessa forma, as ferramentas reduzem o *time to market*, o que permite maior competitividade dos *design centers*⁶ no mercado e aproxima os tempos de projeto das seções analógicas e digitais.

Na esfera acadêmica, o problema de síntese de CIs oferece um teste complexo para métodos de otimização, uma vez que é um problema com diversos mínimos locais, com grande espaço de projeto e multiobjetivo (diversas especificações). Assim, é um campo de estudo multidisciplinar que permite integração de técnicas otimização, técnicas de inteligência computacional e

⁵informações necessárias para descrever a conectividade de um projeto eletrônico.

⁶termo em inglês que se refere aos locais onde são projetados circuitos integrados.

métodos tradicionais de projeto.

Na esfera educacional, ferramentas de síntese podem ser utilizadas a fim de ensinar técnicas de projeto automáticas. Disciplinas que contenham ementa relacionada a projeto em nível de sistema podem ser beneficiadas em utilizar ferramentas para que os alunos possam sintetizar seus sistemas e observarem como as não idealidades do nível de circuito influenciam o desempenho geral.

1.3 Objetivo

Objetivo geral: desenvolver algoritmos e técnicas para obtenção de resultados otimizados em síntese de circuitos integrados analógicos em nível de circuito e sistema.

Objetivos específicos:

- estudo de métodos já existentes de síntese de circuitos integrados analógicos em nível de circuito e sistema;
- desenvolvimento de algoritmo de otimização do tipo livre de derivadas para síntese em nível de circuito e sistema;
- desenvolvimento de algoritmo de otimização do tipo livre de derivadas para busca da fronteira de Pareto em síntese de circuitos em nível de circuito;
- aplicação de técnicas de projeto automatizado para atender as necessidades de síntese em nível de circuito e sistema;
- interligação do nível de sistema e de circuito utilizando curvas da fronteira de Pareto;
- desenvolvimento de ferramenta contendo as técnicas propostas e interface com usuário para tornar os resultados obtidos utilizáveis por projetistas e pela comunidade científica;

1.4 Contribuições da Tese

O trabalho desenvolvido durante esta tese contém as seguintes contribuições:

- novo algoritmo de otimização aplicado à síntese analógica envolvendo *Simulated Annealing/Quenching* e considerando múltiplos objetivos para realização de *crossovers* com soluções de referência encontradas durante a síntese. O algoritmo tem foco em ser capaz de lidar com as características e tipos de variáveis de um problema de síntese analógica,

bem como oferecer uma nova forma de lidar com múltiplos objetivos durante a busca por uma única solução final;

- novo algoritmo híbrido para integrar o algoritmo proposto de *Simulated Annealing/Quenching* com o algoritmo *Particle Swarm Optimization* na busca da fronteira de Pareto de circuitos. A combinação da variação do *Simulated Annealing* proposta com um algoritmo populacional permite o uso das qualidades do primeiro com a possibilidade de superar as limitações relacionadas ao uso de funções agregativas de objetivo;
- nova técnica de extração de especificações de blocos a partir de especificações de sistema através do uso de dados da fronteira de Pareto como referência. Apesar desta técnica utilizar as fronteiras de Pareto, ela permite ao otimizador buscar soluções fora delas para que as mínimas especificações possíveis de blocos sejam encontradas.

1.5 Estrutura do Trabalho

Os primeiros capítulos desta tese apresentam o embasamento teórico que fundamenta o trabalho realizado. Cada capítulo subsequente apresenta contribuições, experimentos e resultados. Finalmente, o último capítulo contém as conclusões. O conteúdo dos capítulos é descrito com mais profundidade a seguir.

O capítulo 2 apresenta a revisão teórica, mostrando a formulação do problema de síntese de circuitos integrados analógicos pela perspectiva de otimização e o estado da arte das técnicas encontradas na literatura. São estudados os tipos de variáveis em um projeto analógico, bem como são apresentadas as características fundamentais de uma ferramenta CAD de síntese analógica. A definição do problema de múltiplos objetivos é analisada, levando à definição de fronteira de Pareto. São estudadas as abordagens para síntese analógica em nível de circuito baseada em conhecimento e baseada em otimização. Abordagens para síntese analógica em nível de sistema são classificadas e estudadas. Dentro destas áreas, os trabalhos encontrados na literatura são apresentados e os que correspondem ao estado da arte são evidenciados.

O capítulo 3 descreve os algoritmos de otimização propostos neste trabalho. As características e etapas do funcionamento deles são apresentadas e discutidas. Ainda, neste capítulo são realizadas comparações de desempenho entre os algoritmos propostos e diversos algoritmos do tipo livre de derivadas encontrados na literatura.

O capítulo 4 apresenta o procedimento de síntese em nível de circuito e os experimentos e resultados na síntese de blocos. O procedimento mostrado neste capítulo é apresentado através da estrutura da ferramenta desenvolvida para síntese. São apresentadas comparações

de resultados de síntese analógica entre algoritmos, entre projetos manuais e com ferramentas encontradas na literatura.

O capítulo 5 apresenta o procedimento de síntese em nível de sistema e mostra os resultados obtidos para sínteses de filtros G_m -C passa-baixas. A adaptação do algoritmo de síntese para a busca da fronteira de Pareto é apresentada e a modelagem do bloco OTA considerando não idealidades em *Verilog-A*.

O capítulo 6 apresenta as conclusões e considerações finais deste trabalho, bem como as sugestões de futuros trabalhos.

1.6 Publicações

A pesquisa deste trabalho resultou nas seguintes publicações:

- Artigos completos publicados em periódicos:
 - WEBER, T.O.; NOIJE, W.A.M.V. Analog Circuit Synthesis Performing Fast Pareto Frontier Exploration and Analysis Through 3D Graphs. *Analog Integrated Circuits and Signal Processing*, Springer US, v. 73, n. 3, p. 861-871, ISSN 0925-1030, 2012;
 - WEBER, T.O.; NOIJE, W.A.M.V. Multi-Objective Design of Analog Integrated Circuits using Simulated Annealing with Crossover Operator and Weight Adjusting. *JICS. Journal of Integrated Circuits and Systems*, v. 7, n. 1, p. 1-9, ISSN 1807-1953, 2012.
- Capítulo de livro:
 - WEBER, T.O.; NOIJE, W.A.M.V. Design of Analog Integrated Circuits using Simulated Annealing/Quenching with Crossovers and Particle Swarm Optimization. *Simulated Annealing - Advances, Applications and Hybridizations*, v. 1, p. 219-244, Ed. InTech, ISBN 978-953-51-0710-1, 2012.
- Trabalhos publicados em anais de eventos:
 - WEBER, T.O.; CHAPARRO, S.; NOIJE, W.A.M.V. Synthesis of a narrow-band Low Noise Amplifier in a 180 nm CMOS technology using Simulated Annealing with crossover operator. *In: Proceedings of the 26th Symposium on Integrated Circuits and Systems* p. 1-5, Curitiba, Brasil, 2013;

- WEBER, T.O.; NOIJE, W.A.M.V. Analog Design Synthesis Performing Fast Pareto Frontier Exploration. *In: Proceedings of the 2nd IEEE Latin American Symposium on Circuits and Systems*, p. 62-66, ISBN 978-1-4244-9484-2, Bogotá, Colômbia, 2011;
- WEBER, T.O.; NOIJE, W.A.M.V. Analog Design Synthesis Method Using Simulated Annealing and Particle Swarm Optimization. *In: Proceedings of the 24th Symposium on Integrated Circuits and Systems*, p. 85-90, ISBN 978-1-4503-0828-1, João Pessoa, Brasil, 2011.

2 REVISÃO TEÓRICA

O tópico de síntese de circuitos integrados analógicos abrange tanto questões relativas ao projeto analógico em si, quanto relativas a métodos de otimização. Desta forma, neste capítulo será apresentada uma revisão teórica sobre projetos analógicos e estado de arte da síntese neste campo, bem como uma introdução aos princípios de funcionamento dos métodos de otimização modernos.

2.1 Projeto Analógico e Estado da Arte de Síntese

O projeto de circuitos integrados analógicos é uma tarefa complexa que envolve conhecimento em diferentes níveis de abstração. No caso da utilização de um fluxo de projeto auxiliado por computador, é necessário o uso de ferramentas apropriadas para cada nível. Em nível de sistema, o projetista precisa escolher a arquitetura que é melhor adequada ao sistema analógico sendo desenvolvido e definir as especificações necessárias dos blocos. Este nível descreve os circuitos dos níveis inferiores como blocos funcionais. Em nível de circuito é realizada a conversão destes blocos funcionais em dimensões de componentes elétricos (transistores, resistores, capacitores, entre outros). A topologia e as dimensões dos componentes em nível de circuito determinarão as especificações possíveis do bloco. O nível de *layout* é responsável por converter os valores de dimensões presentes no nível de circuito para o desenho de máscaras que mais tarde serão utilizadas no processo de fabricação.

O foco deste trabalho está no projeto analógico em nível de circuito e em nível de sistema. As especificações finais a serem atingidas no nível de circuito são as variáveis de entrada da síntese em nível de sistema. Ou seja, as métricas de desempenho dos blocos são a interface entre estes níveis. Estas métricas de desempenho do circuito precisam ser atingidas através da manipulação das variáveis de dimensionamento e polarização dos dispositivos.

Nesta seção, será apresentado como o projeto de circuitos analógicos em nível de circuito e sistema é observado na perspectiva de ferramentas de otimização. Ainda, é feita uma revisão bibliográfica sobre os trabalhos publicados nestes campos.

2.1.1 Variáveis de Projeto

Projetos analógicos em nível de circuito possuem variáveis geométricas e elétricas. As variáveis geométricas são as dimensões dos dispositivos existentes no circuito, tais como a largura e comprimento do canal de um transistor. Elas são importantes para determinar características do dispositivo como o valor de um resistor, de um capacitor, ou a curva corrente/tensão de um transistor.

A influência destas variáveis geométricas nas medidas de desempenho do circuito são usualmente não lineares. Os efeitos das variáveis geométricas na corrente de um transistor, que irão afetar as medidas do circuito, vão depender não apenas das suas próprias dimensões mas também das tensões em seus terminais. As equações de um transistor MOS podem ser vistas na equação (2.1) para uma aproximação de canal longo, modelo de Shichman-Hodges (SHICHMAN; HODGES, 1968):

$$i_D = \begin{cases} 0, & \text{se na região de corte} \\ \mu_n C_{OX} \frac{W}{L} [(v_{GS} - v_{TH})v_{DS} - \frac{v_{DS}^2}{2}], & \text{se na região linear (de triodo)} \\ \frac{\mu_n C_{OX} W}{2L} (v_{GS} - v_{TH})^2 (1 + \lambda v_{DS}), & \text{se na região de saturação} \end{cases} \quad (2.1)$$

onde i_D é a corrente de dreno, W é a largura do canal do transistor, L é o comprimento de canal, μ_n é a mobilidade de canal, C_{OX} é a capacitância de óxido da porta por unidade de área, v_{GS} é a tensão entre a porta e a fonte, v_{TH} é a tensão de limiar, v_{DS} é o tensão entre dreno e fonte e λ é o parâmetro do efeito de modulação de canal. A Figura 2.1 mostra a relação entre corrente de dreno e tensão entre dreno e fonte de um transistor NMOS identificando as regiões de triodo e saturação.

A região de operação de um transistor é o resultado das tensões em seus terminais, como pode ser visto nas inequações (2.2), (2.3) e (2.4). A colocação de um transistor em uma região específica de operação é frequentemente uma restrição do problema, uma vez que isto afetará a funcionalidade e robustez do circuito.

$$\text{corte} \quad v_{GS} \leq v_{TH} \quad (2.2)$$

$$\text{triodo} \quad v_{DS} < v_{GS} - v_{TH} \quad e \quad v_{GS} > v_{TH} \quad (2.3)$$

$$\text{saturação} \quad v_{DS} \geq v_{GS} - v_{TH} \quad e \quad v_{GS} > v_{TH} \quad (2.4)$$

Com a redução da dimensão do canal do transistor, é possível a obter relações W/L muito

Figura 2.1: Relação entre corrente de dreno (i_D) e tensão entre dreno e fonte (v_{DS}) para diversas tensões entre porta e fonte (v_{GS}) de um transistor NMOS sem efeito de modulação de canal.

altas. Nesta condição, pequenos valores de v_{GS} , mesmo inferiores a v_{TH} , são capazes de gerar corrente de dreno (corrente sublimiar). Dessa forma, se pode realizar projeto de circuitos analógicos na região de inversão fraca. Essa região ocorre desde valores de $(v_{GS} - v_{TH})$ ligeiramente negativos até valores de aproximadamente 20mV (COMER; COMER, 2004), com corrente de dreno:

$$i_D = \frac{W}{L} i_{D0} e^{q v_{GS} / nkT} \quad (2.5)$$

onde k é a constante de Boltzmann, T é a temperatura e i_{D0} e n são dependentes da tecnologia (GRAY et al., 2001).

Em qualquer das regiões de operação, apenas W e L são as variáveis do transistor que podem ser diretamente manipuladas pelo projetista, enquanto C_{OX} , μ_n e v_{TH} , i_{D0} e n são dependentes da tecnologia e temperatura. As tensões são consequências da combinação das malhas do circuito, as quais o projetista tem controle indireto através da atribuição apropriada de corrente nos transistores do circuito.

Quanto às variáveis elétricas, as que são diretamente manipuláveis no projeto são tensões e correntes que estão usualmente na interface do dispositivo sob teste (DUT, do inglês, *Device Under Test*) e o ambiente externo, tais como correntes de polarização. Em projetos maiores, tais valores podem estar previamente limitados pelas tensões/correntes de referência existentes no sistema.

A escolha dos valores das variáveis não é arbitrária. Como elas resultam em dispositivos físicos, existem restrições que precisam ser respeitadas para que o projeto seja viável. Estas limitações são chamadas de restrições de projeto (RAO, 1996). Elas podem ser de dois tipos: comportamentais (ou funcionais) e geométricas.

Restrições geométricas são fornecidas pela empresa que fabricará o *chip* e referem-se às limitações de dimensões de largura e comprimento, como por exemplo, o menor comprimento possível para o canal dos transistores.

As restrições comportamentais são normalmente fornecidas pelo projetista para garantir que o circuito funcionará apropriadamente, tal como definição da região de operação dos transistores, tensão efetiva na porta dos transistores, ou relações específicas como a dimensão entre diferentes transistores. Este tipo de restrição permite que o projetista utilize seu conhecimento tanto para reduzir o espaço de projeto quanto para garantir robustez ao circuito sendo projetado. Se o projeto é otimizado utilizando apenas considerações nominais, o resultado pode ser um circuito muito susceptível às variações de processo, temperatura e alimentação. Restrições funcionais para aumento de robustez são apresentadas de forma sistemática por Graeb et al. (2001).

Ainda, durante o processo de projeto, existem parâmetros fixos e variáveis. Os fixos constituem aqueles sobre os quais o projetista não tem controle, ou optou por não modificar, e são pré-determinados. No caso de circuitos analógicos, um parâmetro pré-determinado pode ser, por exemplo, a tensão de alimentação ou a corrente da fonte disponível no sistema. Os demais parâmetros estão disponíveis para uso durante a otimização, e estão contidos dentro do vetor de projeto $\vec{x}$. Por exemplo, as dimensões W e L dos transistores de um circuito podem estar contidas dentro deste vetor, bem como qualquer outro parâmetro variável. Caso fossemos otimizar um estágio fonte comum como ilustrado na Figura 2.2, poderíamos ter como vetor de projeto $\vec{x} = \{W_1, L_1, R_1\}^T$ e, como parâmetro pré-determinado, v_{DD} .

Figura 2.2: Estágio fonte comum.

2.1.2 Medidas de Desempenho de Projeto

As medidas de desempenho ou especificações são resultado de funções das variáveis de projeto discutidas na seção anterior. Cada medida de desempenho representa um aspecto do circuito a ser analisado. As medidas a serem realizadas dependem do tipo de bloco sendo

sinetizado e, no caso dos resultados serem utilizados para síntese em alto nível, do modelo comportamental que será utilizado para representar o circuito.

As medidas de desempenho podem ser obtidas tanto analiticamente quanto por meio de simulações. Por meio de equações, as relações entre os parâmetros do circuito e as medidas são feitas com base na definição de cada medida (e.g., taxa de variação de saída de um amplificador $= \left| \frac{dv_{saída}(t)}{dt} \right| = \frac{I_{saída}}{C_{saída}}$) e de equações que descrevem o comportamento dos transistores. Por meio de simulações, é criado um circuito de teste em torno do circuito sendo projetado a fim de realizar as medições.

Quando uma síntese é realizada utilizando apenas medições sob condições típicas de processo, temperatura e alimentação, ela é chamada de síntese nominal. Quando ela leva em consideração variação destes parâmetros, é considerada uma síntese guiada à robustez.

2.1.3 Função Objetivo

A fim de realizar uma síntese, é necessário que exista uma forma de avaliar a qualidade de uma dada solução. Normalmente, entre as soluções que respeitam as limitações geométricas e funcionais, existem algumas que são melhores do que outras. As soluções melhores são consideradas como tal por possuírem melhor desempenho nos critérios importantes para determinado problema. A função objetivo é a responsável por fazer tal avaliação e é um dos pontos mais críticos da otimização.

O vínculo entre o método matemático de otimização e o problema real sendo abordado é realizado pela função objetivo. No caso do objetivo da otimização ser a minimização, esta função pode ser chamada de função custo. O método matemático irá buscar o menor valor possível da função custo, e a construção desta função deve ser tal que o menor resultado possível represente o melhor projeto possível. A Figura 2.3 ilustra a interação da função objetivo com o otimizador e com o problema real.

Figura 2.3: Interação da função objetivo com o otimizador e com o problema real.

Apesar do processo de otimização buscar o mínimo global, ou seja, o menor valor possível da função custo, podem existir também os chamados mínimos locais. Estes mínimos são regiões

da função custo que possuem menor valor que seus vizinhos (soluções com valores de variáveis próximas), porém não o menor possível em toda a função. Os mínimos locais frequentemente oferecem desafios aos métodos de otimização, de forma que estes podem ficar “presos” e não conseguir continuar em busca do mínimo global. A Figura 2.4 ilustra o conceito de mínimo global e local para um problema com apenas uma variável.

Figura 2.4: Exemplo de regiões de mínimos locais e mínimo global para um problema de uma variável.

2.1.4 Problemas com Múltiplos Objetivos em Projeto de Circuitos Analógicos

Existem vários objetivos a serem satisfeitos no projeto de circuitos analógicos. Cada tipo de bloco analógico possui seus tipos de especificações e, geralmente, uma boa parte deles são concorrentes entre si. Isto significa que melhorar uma característica do circuito incorre na piora de uma ou várias outras, resultando nas chamadas trocas de projeto, ou *tradeoffs*. Razavi (2001) mostra um exemplo dos *tradeoffs* envolvidos no projeto de amplificadores analógicos, incluindo especificações como potência, ganho, linearidade e outros (a Figura 2.5 ilustra alguns deles). Este tipo de problema é chamado de Problema com Múltiplos Objetivos (MOP, em inglês, *Multi-Objective Problem*).

Em problemas de otimização multiobjetivo não é sempre possível afirmar que uma solução é melhor do que a outra. Uma solução pode ser melhor que outra em uma especificação e, no entanto, ser pior em outras. Há alguns casos, porém, no qual uma solução é melhor que outra em todas as especificações. Nesses casos, dizemos que uma solução domina a outra. O conjunto das soluções do projeto que não são dominadas por nenhuma outra solução é chamado conjunto de Pareto. Este conjunto é utilizado para formar a fronteira de Pareto, que representa

Figura 2.5: *Tradeoffs* em projetos de amplificadores (RAZAVI, 2001).

todos os *tradeoffs* envolvidos na síntese/otimização. A seção 2.2 apresenta a descrição formal de problemas de múltiplos objetivos e da definição de fronteira de Pareto.

2.1.5 Espaço de Projeto

A síntese de projetos analógicos pode ser abordada tanto como um problema de variáveis discretas ou contínuas. No caso das variáveis discretas, é necessário estabelecer qual é a mínima diferença admissível entre um valor e outro, enquanto no caso das variáveis contínuas qualquer valor entre os limites inferior e superior da variável é admissível.

Quando se trabalha com variáveis discretas no problema de síntese analógica, o valor do passo da variável será usado para gerar uma nova solução a partir da anterior na fase de geração de novas soluções. Para o caso do comprimento do canal de um transistor com passo de $0,5 \mu$, por exemplo, uma variável selecionada para alteração que possuía valor $2 \mu m$ será modificada para $1,5 \mu m$ ou $2,5 \mu m$, conquanto ambos valores estejam entre os limites inferior e superior. A definição da variável discreta restrita por limites inferior e superior pode ser vista como o conjunto:

$$\{x_d \in \mathbb{R} | (\exists n \in \mathbb{N})(x_d = li + n\Delta x_d) \wedge (x_d < ls)\} \quad (2.6)$$

onde x_d é a variável discreta, li é o limite inferior, Δx_d é o passo, e ls é o limite superior. De forma similar, a variável contínua pode ser descrita como:

$$\{x_c \in \mathbb{R} | li < x_c < ls\} \quad (2.7)$$

No caso de variáveis contínuas, não existe um passo pré-definido, e a mínima variação Δx_c de uma variável x_c torna-se subjetiva. Nesse caso, utiliza-se recursos para alterar o Δx adaptativamente ao processo de otimização.

Em ambos os casos, a utilização de um Δx muito pequeno resultará em um número excessivo de iterações até que o algoritmo encontre um mínimo. Por outro lado, uma variação muito grande diminuirá a precisão com a qual o espaço de projeto será analisado e, portanto, criará dificuldade em aproximar a solução do resultado ótimo gradualmente.

2.1.6 Características de uma Ferramenta CAD de Síntese Analógica

O projeto de circuitos analógicos é um problema desafiador uma vez que possui múltiplos objetivos, possui diversos mínimos locais e o tempo requerido para cada avaliação de solução é comumente não negligenciável. Ainda, como mencionado anteriormente, os objetivos de uma síntese analógica e as restrições de desempenho usualmente são funções não lineares dos parâmetros de circuito.

Algumas características importantes a serem observadas em uma ferramenta são descritas abaixo (BALKIR et al., 2003):

- **Tempo de preparação:** é o tempo que antecede a síntese propriamente dita. É o tempo necessário para que os arquivos de configuração da síntese sejam preparados, para que o projetista converta o problema real para o formato compreensível pela ferramenta de otimização. Em avaliações baseadas em equação, este tempo de preparação é usualmente alto (de semanas a meses). Em avaliações baseadas em simulação, este tempo é reduzido uma vez que o simulador será responsável por avaliar as respostas. O projetista usualmente precisa apenas fornecer os circuitos, as medições e informações adicionais para restrição das variáveis. Uma vez que o sistema esteja configurado, o tempo de síntese das avaliações baseadas em equação é muito menor do que o tempo das baseadas em simulação. No entanto, com o rápido aumento no poder computacional, o tempo necessário às simulações torna-se mais adequado à síntese para uma grande parte dos circuitos;
- **Independência da condição inicial:** se refere à possibilidade de atingir boas soluções sem necessitar uma boa solução inicial fornecida pelo projetista. Esta característica é um requerimento fundamental para diferenciar um processo de síntese de um processo de otimização. O interesse nesta independência justifica o uso de técnicas de busca global em vez do uso de técnicas de busca local, que não possuem mecanismos para escapar de mínimos locais;
- **Precisão:** é a capacidade de obter avaliações das soluções com resultados fidedignos aos dispositivos reais. Questões relacionadas à precisão normalmente são um problema mais crítico em avaliações baseadas em equação, uma vez que o modelo utilizado precisa capturar todos os importantes efeitos secundários nos dispositivos enquanto mantém equa-

ções tratáveis pelo otimizador. Abordagens baseadas em simulação, em geral, utilizam modelos mais precisos e de uso corrente pela indústria, como o BSIM3 e o BSIM4;

- **Tolerância às variações:** é a capacidade de fornecer soluções que são boas não apenas em condições nominais, mas também considerando variações do processo de fabricação, variações de temperatura e variações de tensão de alimentação;
- **Generalidade:** refere-se à capacidade de sintetizar uma vasta gama de tipos de circuito. Esta também é uma característica na qual as abordagens baseadas em simulação têm desempenho melhor do que as baseadas em equações, visto que nelas o único requerimento para sintetizar um circuito é a capacidade do simulador de processá-lo. Isso pode gerar, por exemplo, restrição de alguns simuladores em tratar circuitos de radiofrequência;
- **Independência do processo de fabricação:** é a possibilidade de modificar a tecnologia de fabricação após o problema ser formulado. Este é um requerimento importante uma vez que a troca de tecnologia utilizada é um procedimento comum nas *design houses*. Nos casos em que esta característica não existe, faz-se necessária a realização de reprojeto manual dos blocos.

Uma ferramenta de síntese para projeto analógico necessita levar em conta a maior parte destas características para tornar-se útil para a indústria. Tanto uma ferramenta que demanda muito esforço por parte do projetista ou uma que não forneça soluções robustas, poderá ser rapidamente descartada e cair em desuso.

Na Figura 2.6, um exemplo de ferramenta de síntese em nível de circuito pode ser visto como uma caixa-preta com suas entradas e saídas. O projetista fornece a topologia a ser sintetizada (por exemplo, amplificador Miller, folded cascode,...), a tecnologia a ser utilizada (por exemplo, $0,35\ \mu\text{m}$, $0,18\ \mu\text{m}$, $90\ \text{nm}$, ...), as especificações (por exemplo, ganho, área, potência,...), os circuitos de teste (ou *testbenches*) que serão utilizados para realizar as medidas no circuito e finalmente as configurações de síntese. O sintetizador interage diversas vezes com o simulador elétrico para avaliar as soluções que são propostas durante o procedimento de otimização. As saídas do sintetizador são o circuito dimensionado e polarizado e as especificações atingidas. Caso esteja sendo realizada uma síntese buscando a fronteira de Pareto, o sintetizador deve gerar gráficos e/ou tabelas mostrando a aproximação da curva de Pareto.

2.1.7 Estado da Arte de Síntese em Nível de Circuito

O processo de síntese em nível de circuito trata do dimensionamento e polarização de circuitos analógicos dada uma topologia, tecnologia e especificações. Os blocos sintetizados

Figura 2.6: Entradas e saídas do sintetizador em nível de circuito.

neste nível são compostos por transistores, diodos, capacitores, indutores e resistores. Através da escolha apropriada de larguras e comprimentos de transistores, bem como dos demais componentes, busca-se atender às especificações mínimas de projeto e, caso a técnica permita, aproximar-se do melhor resultado possível para dada topologia.

Nye et al. (1988) apresenta um apanhado histórico dos primeiros trabalhos relacionados a esta área. Gielen e Rutenbar (2000) realizam uma classificação dos métodos de síntese em abordagens baseadas em conhecimento e abordagens baseadas em otimização.

2.1.7.1 Abordagem Baseada em Conhecimento

Esta abordagem define um plano de síntese baseado no conhecimento prévio de projetistas experientes, de forma que o desenvolvimento de um projeto possa seguir uma sequência bem definida de passos (por isso também chamada de abordagem baseada em plano). Este plano consiste de equações previamente definidas derivadas de projetos manuais. Baseado nas especificações, estas equações calculam os parâmetros de projeto (dimensionamento e polarização) (VANDENBUSSCHE et al., 2003).

Nesta abordagem, a criação do plano de síntese é a etapa mais demorada, visto que demanda experiência dos projetistas e capacidade de convertê-la para os parâmetros esperados pela ferramenta (possivelmente meses). Ainda, um plano de síntese serve apenas para uma topologia específica, tornando tal abordagem pouco genérica. Uma vez que este plano esteja pronto, no entanto, a síntese ocorre utilizando pouco tempo de processamento.

Grande parte dos estudos em síntese de circuitos analógicos nos anos 1980 utilizavam abor-

dagem baseada em conhecimento. Algumas das ferramentas que foram desenvolvidas seguindo esta abordagem são, por exemplo, IDAC (DEGRAUWE et al., 1987), OASYS (HARJANI et al., 1989) e BLADES (EL-TURKY; PERRY, 1989).

Uma iniciativa mais recente de pesquisa em síntese com abordagem baseada em conhecimento é chamada de projeto estruturado. Nesse método, uma abordagem geral baseada em cálculos simplificados (pois é desenvolvida inicialmente para cálculo manual) de modelo dos transistores somada a uma metodologia de especificação de blocos básicos (pares diferenciais, espelhos de corrente, etc) e conversão de especificações de nível de circuito para especificação em nível destes blocos é realizada (STEFANOVIC; KAYAL, 2008). Desta forma, é almejada a criação de uma forma mais genérica do uso de conhecimento para projeto de circuitos analógicos. No entanto, ainda assim são necessários planos de síntese específicos para os blocos, apesar de estes serem menos demorados de realizar do que nas abordagens anteriores.

Outra linha atual trata da incorporação do conhecimento juntamente com a abordagem baseada em otimização. Isto é realizado através do uso de projeto impulsionado por restrições (que são resultantes de conhecimento de projeto) (JERKE; LIENIG, 2009). Graeb et al. (2001) e Massier et al. (2008) propuseram formas de utilizar restrições de projeto durante a escolha das dimensões dos transistores para torná-lo mais robusto durante a otimização sem necessidade de novas simulações. O método consiste em, dada uma *netlist*, efetuar um reconhecimento automático dos blocos construtivos a fim de particionar o circuito. Uma vez realizada a classificação das partes, restrições específicas para cada um dos tipos de blocos construtivos são aplicadas a fim de aumentar a robustez do projeto final.

Apesar destes trabalhos mais atuais utilizarem conhecimento para a síntese, são mais próximos da abordagem baseada em otimização, visto que o conhecimento é utilizado apenas como auxílio para o problema de otimização.

2.1.7.2 Abordagem Baseada em Otimização

Esta abordagem converte o problema de síntese de circuitos analógicos em um problema de otimização passível de ser solucionado através de métodos matemáticos de otimização clássicos ou modernos. Esta abordagem começou a ser popularizada no final dos anos 80 com o objetivo de atender a necessidade de ferramentas de síntese flexíveis e com pouca necessidade de preparação prévia antes do uso. Assim, foram desenvolvidas para atender as falhas das abordagens baseadas em conhecimento.

Dentro do método de síntese baseado em otimização, ainda existem subdivisões referentes ao método de avaliação. O método de avaliação consiste na forma pela qual a ferramenta

avalia o quão próximo dos seus objetivos o circuito projetado está (por exemplo, avaliar um amplificador para descobrir o quão próximo do ganho de 80 dB ele está, no caso de esta ser uma especificação de projeto). Existem duas divisões para o processo de avaliação: baseada em equações e baseada em simulações.

- **Avaliação baseada em Equações:** Este método de avaliação é realizado através de equações derivadas analiticamente. A cada novo conjunto de parâmetros do circuito (solução), este é aplicado em equações que fornecem as respostas do circuito. A necessidade da criação destas equações é a problemática fundamental deste método. Elas podem ser geradas manualmente, o que demanda tempo e conhecimento por parte dos projetistas, ou automaticamente. Técnicas de simulação simbólica podem gerar funções de transferência para pequenos sinais para topologias sem a necessidade da intervenção do projetista.

No entanto, a complexidade das técnicas automáticas de análise simbólica têm crescimento exponencial em relação ao número de componentes do circuito (BALKIR et al., 2003). Alguns trabalhos nessa área são a síntese envolvendo simulação simbólica (GIELLEN et al., 1990), geração de modelos comportamentais simbólicos de circuitos não lineares (BORCHERS, 1998) e a síntese envolvendo programação geométrica (HERSHENSON et al., 2001).

- **Avaliação baseada em Simulações**

Este método de avaliação realiza simulações elétricas a fim de averiguar o quão próximo das especificações está o circuito projetado. A cada nova solução proposta pelo otimizador, uma *netlist* é gerada com o valor das variáveis de dimensionamento e polarização e é simulada. Os resultados são, então, tratados para serem convertidos em uma ou várias funções custo.

Uma vantagem desta abordagem é a baixa necessidade de preparação prévia, uma vez que normalmente a otimização usando simulações é considerada uma caixa-preta, ou seja, o otimizador não requer conhecimento sobre o circuito sendo otimizado. Em contraponto, nesse tipo de abordagem há um tempo de síntese consideravelmente maior uma vez realizadas as etapas preparatórias do que as demais abordagens. Isto deve-se a uma simulação elétrica poder ter um tempo considerável dependendo da complexidade do circuito. Ainda, o número de simulações para cada iteração deve ser suficiente para avaliar o circuito sob diversos pontos de vista, uma vez que ignorar uma especificação pode ocasionar que a síntese encaminhe-se para um circuito excelente em algumas especificações porém impossível de ser utilizado na prática.

Com o aumento da capacidade de processamento dos computadores modernos, este método tornou-se mais popular do que o baseado em equações. A seguir, serão descritos

algumas ferramentas encontradas na literatura que são fundamentais para o entendimento do estado da arte. Os trabalhos que são considerados pelo autor deste trabalho como representativos do estado da arte serão marcados com um asterisco (*) ao lado da citação. Em caso da ferramenta ter sido nomeada pelos autores, este nome será indicado antes da citação:

- **FRIDGE (MEDEIRO et al., 1994)**: este trabalho enfoca o uso de algoritmos de busca global em vez de busca local a fim de sintetizar apropriadamente circuitos analógicos. Eles combinam uma formulação especial de funções custos (com divisão entre restrições fortes, fracas e objetivos de projeto) com um algoritmo estatístico de busca. Utiliza um algoritmo muito similar ao *Simulated Annealing*, porém com uma heurística própria para controle do decréscimo da temperatura. O *testbench* utilizado para avaliar o método foi um amplificador operacional *folded cascode* completamente diferencial.
- **ASTRX/OBLX (OCHOTTA et al., 1996)**: da mesma forma que o desenvolvido por Medeiro, este trabalho também tem foco na síntese através de busca global. O processo de síntese combina o uso de uma técnica chamada AWE (Avaliação de Forma de Onda Assintótica, do inglês, *Asymtotic Waveform Evaluation*) e *Simulated Annealing*. A técnica AWE é utilizada como substituta das simulações usando SPICE, pois é capaz de prever o desempenho de pequenos sinais do circuito usando um modelo de complexidade reduzida. O *Simulated Annealing* é utilizado como algoritmo de busca de soluções. A formulação do problema de otimização utiliza soma ponderada de funções custos individuais somada a penalidades extras para representar as restrições de projeto. Os *testbenches* foram dois OTAs, um amplificador de dois estágios, um amplificador *folded cascode*, um comparador, um amplificador BICMOS de dois estágios e uma topologia nova de *folded cascode* para um processo de 1,2 μm .
- **OPTOMEGA (KERAMAT; KIELBASA, 1998)**: realiza uma otimização nominal para buscar um ótimo global e posteriormente a otimização guiada à robustez quanto às variações de processo. A otimização nominal utiliza um sistema chamado OLGA, que é o elemento que interliga a interface com o usuário, os algoritmos desenvolvidos em *MATLAB*, o banco de dados e os resultados. O algoritmo de otimização pode ser escolhido pelo usuário e pode ser *Simplex* (Nelder e Mead), *Simulated Annealing*, *Enhanced Simulated Annealing* e *SAPLEX (Simulated Annealing/simPLEX)*; A otimização para melhoramento da robustez utiliza o índice AQI (Índice de Qualidade Média, do inglês, *Average Quality Index*). Este índice é estimado através de métodos de Monte Carlo. Para avaliar a ferramenta, os autores sintetizaram um filtro

passa alta (apenas com componentes passivos) e um circuito CMOS *driver* de *clock*.

- **MAELSTROM (KRASNICKI et al., 1999)**: este trabalho afirma que as tentativas de síntese anteriores tinham a falha de tentar reduzir a qualidade das avaliações dos circuitos em prol da rapidez de síntese. Assim, o uso de simuladores elétricos comerciais é tido para eles como a forma de garantir que a síntese possa ser utilizada para casos industriais e não apenas em ambientes de pesquisa. A formulação da otimização é igual a da ferramenta OBLX e o algoritmo de busca é chamado de PRSA (do inglês, *Parallel Recombinative Simulated Annealing*). Este algoritmo combina características dos algoritmos genéticos com o *Simulated Annealing*, mantendo uma população de soluções que, a cada nova geração, podem optar por alterar seu valor através apenas de modificações típicas do *Simulated Annealing* (na nomenclatura dos algoritmos genéticos, esta modificação seria chamada de mutação) ou fazer recombinação com os outros indivíduos. Os circuitos utilizados como *testbenches* para a ferramenta foram um amplificador desenvolvido por eles, um amplificador *folded cascode* básico e um amplificador *folded cascode* completamente diferencial (o mesmo utilizado por Medeiro et al. (1994)).
- **ANACONDA (PHELPS et al., 2000)**: trabalho baseado no MAELSTROM, herdando boa parte de suas características. Seu diferencial é o uso de um algoritmo de busca chamado *Stochastic Pattern Search*. Este algoritmo combina a técnica tradicional de *pattern search* com uma técnica populacional na qual indivíduos são selecionados aleatoriamente para que a primeira técnica seja efetuada sobre ele. Este paralelismo é utilizado a fim de evitar que o aspecto local da *pattern search* prejudique a síntese. Os circuitos utilizados para avaliar a ferramenta foram um amplificador operacional de dois estágios, um amplificador *folded cascode* e um amplificador de potência.
- **HOLMES (SMEDT; GIELEN, 2003)**: apresenta um método de síntese guiado à robustez através de exploração da fronteira de Pareto e posterior criação de modelos entre as variáveis de projeto e as medidas de desempenho através de regressão multivariada. A exploração da fronteira de Pareto é realizada utilizando algoritmos evolutivos e um simulador elétrico. Após obtenção dos modelos da fronteira de Pareto, técnicas de estimativa de *yield* são utilizadas nesses modelos a fim de fornecer ao projetista relações de *tradeoff* entre *yield* e especificações de projeto. O método é testado em um amplificador Miller, mantendo uma proporção aproximadamente igual entre tempo de exploração de desempenho e tempo de exploração de *yield*.
- **(TIWARY et al., 2006)***: desenvolve um método de síntese orientado à robustez para geração de fronteiras de Pareto. A motivação é utilizar estas fronteiras de Pa-

reto para permitir uma modelagem eficiente dos *tradeoffs* do circuito sob otimização para que a síntese em um nível mais alto de abstração (como será visto na seção de síntese em nível de sistema) possa ocorrer. O processo de síntese consiste em uma otimização nominal (utilizando uma combinação de *Simulated Annealing* com algoritmos genéticos) somado a um método de Monte Carlo aproximado. Este método permite que a síntese levando em consideração variações do processo dure apenas de 5 a 6 vezes o tempo requerido para síntese nominal. A verificação da funcionalidade do método foi realizada em um circuito VCO (Oscilador Controlado por Tensão, do inglês, *Voltage Controlled Oscillator*) em anel.

- **SANGRIA (MCCONAGHY; GIELEN, 2009)***: também tem como foco a síntese orientada à robustez diante às variações de processo. Utiliza modelagem de superfície de resposta adaptativa (do inglês, *adaptive response surface modeling*) e introduz o conceito de homotopia estrutural. Realiza uma separação de níveis de avaliação, na qual simulações menos exigentes são realizadas para simulações DC e as mais exigentes envolvem variações de processo e ambiente. Há uma integração entre o processo de busca pelo ótimo nominal e pelo ótimo que garante robustez, de forma que a simultaneidade desta busca evita que a busca nominal aprisione a solução em mínimos locais quanto às variações de processo. Os circuitos nos quais o método/ferramenta foi testado foram três amplificadores operacionais e uma tensão de referência. Os amplificadores possuem 10, 30 e 50 dispositivos (a fim de testar o efeito do incremento de dimensionalidade na eficácia da ferramenta).
- **(FAKHFAKH et al., 2010)**: este trabalho utiliza a técnica de *particle swarm optimization* para sintetizar circuitos analógicos. Tem como foco provar que esta técnica é eficiente para síntese em casos mono e multiobjetivo. Como este trabalho tem maior foco na validação do conceito do que propriamente na criação de uma ferramenta, as equações de desempenho dos circuitos foram derivadas manualmente e as simulações realizadas através do *software* ADS *Agilent Advanced Design System*. Os resultados foram favoráveis para otimização mono objetivo de um LNA (do inglês, *Low Noise Amplifier*) e biobjetivo para um transportador de corrente de segunda geração.
- **GENOM (BARROS et al., 2010)***: utiliza um *kernel* modificado de algoritmos genéticos para aumentar a eficiência na síntese e a combina com o algoritmo para aprendizado de máquina *Support Vector Machine* (SVM). A modelagem com SVM auxilia o operador de seleção do algoritmo genético para evitar que soluções não promissoras sejam avaliadas através de simulação. Para testar a técnica, foi sintetizado um amplificador *folded cascode* e foram obtidos resultados superiores aos

encontrados na ferramenta FRIDGE.

- **(MUELLER-GRITSCHNEDER; GRAEB, 2010)***: cria um método integrado para busca da fronteira de Pareto focada em *yield* em uma única fase para obtenção de soluções robustas. Nele, são criados modelos lineares de dependência para análise de pior caso para cada especificação, de forma que durante a síntese tais modelos sejam gradualmente atualizados para que sempre reflitam a dependência aproximada da região atual. Estes modelos de dependência indicam, para determinada solução, quais são as direções de cada parâmetro de projeto variável e parâmetro de operação (por exemplo, temperatura e alimentação) que mais afetam negativamente cada especificação. Testes foram realizados em um amplificador de dois estágios do tipo Miller e comparados com um método de duas fases para mostrar as fraquezas do método em duas fases em relação ao método proposto;
- **(LIU et al., 2011)***: desenvolve uma nova abordagem para síntese focada em *yield*, chamada *Ordinal Optimization-based Random-Scale Differential Evolution* (ORDE). É constituído por um fluxo de dois estágios, no qual, inicialmente, a otimização ordinal é aplicada alocando um número suficiente de amostras de soluções promissoras, evitando simulações de Monte Carlo de soluções não promissoras. É utilizado um algoritmo de evolução diferencial para busca global e um operador de mutação de escala variável. O algoritmo ORDE atinge uma melhoria de desempenho de aproximadamente dez vezes em relação às outras técnicas de otimização de *yield* baseadas em Monte Carlo. Os testes foram realizados em um amplificador *folded cascode* diferencial de dois estágios.
- **(GUERRA-GÓMEZ et al., 2012)**: este trabalho faz um resgate de projeto de circuitos analógicos através de técnicas baseadas em ponto de operação. Este tipo de técnica, apesar de popular para projetos manuais, é pouco utilizada por ferramentas automáticas. Consiste no uso de correntes, tensões e, por vezes, o comprimento do canal dos transistores para selecionar o ponto de operação. Há, então, um estágio intermediário para conversão das correntes e tensões em larguras e comprimentos dos transistores através de equações de primeira ordem, tabelas e métodos de otimização. Os benefícios do uso desta técnica são a rapidez e um melhor *insight* a respeito do funcionamento dos circuitos em relação às técnicas tradicionais. O trabalho utiliza a caracterização do transistor através das curvas gm/I_D e $I_D/(W/L)$. Para exemplificar a técnica, foi projetado um espelho de corrente simples.
- **(GUERRA-GÓMEZ et al., 2013)**: utiliza o algoritmo evolutivo NSGA-II para encontrar a fronteira de Pareto e, em uma segunda fase, faz uma análise de sensibilidade com múltiplos parâmetros baseado na extrapolação de Richardson. A

análise de sensibilidade permite filtrar da fronteira de Pareto as soluções que são muito susceptíveis a variações de processo. Para testar o método, faz a síntese de um amplificador *folded cascode* OTA, um seguidor de tensão, um transportador de corrente de segunda geração e um amplificador operacional com realimentação de corrente.

- **(MALAK et al., 2015)**: realiza a exploração do espaço de projeto através de uma inicialização baseada em caminhos de Peano (curvas preenchedoras de espaço). Após, é realizado um particionamento das variáveis de projeto e realizada a otimização local utilizando o algoritmo simplex de Nelder-Mead para os melhores pontos resultantes da inicialização. As variáveis de projeto utilizadas durante a otimização não são diretamente as dimensões dos transistores, mas sim variáveis de um plano de projeto estabelecido com base na experiência do projetista e nas equações do modelo BSIM3v3. Os autores aplicaram o método ao dimensionamento e polarização de um fonte de referência constante.
- **AIDA-C (LOURENÇO et al., 2015)**: propõe o uso de restrições de projeto em nível de circuito para reduzir o impacto de efeitos indesejados de *layout* no projeto. Dessa forma, evita o tempo de execução da geração de *layout* e extração durante o laço de otimização. Utiliza o algoritmo NSGA-II para otimização multiobjetivo. A abordagem é testada em um amplificador Miller e um amplificador com estágio único e aumento de ganho utilizando combinadores de tensão.

2.1.8 Estado da Arte de Síntese em Nível de Sistema

A síntese em nível de sistema trata de projetos de maior dimensão (maior número de sub-funções sendo realizadas), nos quais abordagens diferentes das realizadas em nível de circuito podem ser necessárias. Em geral, a síntese em nível de sistema envolve a divisão do projeto em sub-blocos e, em muitos casos, a utilização de modelos comportamentais para aumentar a velocidade de simulações.

Gielen et al. (2005) apresentam uma classificação de trabalhos baseada na organização do espaço de projeto. As principais categorias são:

- **metodologia plana**: no qual não há hierarquia e todas as variáveis de projeto são abordadas simultaneamente pelo otimizador e apenas as especificações de sistema são avaliadas;
- **metodologia *bottom-up***: no qual os blocos são projetados anteriormente ao sistema, com especificações definidas com base na experiência do projetista;

- **metodologia *top-down* dirigida por restrições:** realiza o projeto começando a partir do nível de sistema. As escolhas sobre quais blocos podem ser utilizados são realizadas com base em modelos de viabilidade;
- **construção de modelos de viabilidade + metodologia *top-down* dirigida por restrições:** como fase anterior à implementação da metodologia *top-down*, faz a construção de modelos de viabilidade automaticamente. Esta construção é uma etapa do tipo *bottom-up*;
- **metodologia *bottom-up* multiobjetivo:** em vez de criar modelos de viabilidade para todo o espaço de projeto dos blocos, utiliza apenas as trocas de projetos (*tradeoffs*) capturados através da exploração da fronteira de Pareto dos mesmos. Para cada solução da fronteira de Pareto, estão contidas as especificações atingidas e as variáveis que foram utilizadas para alcançá-las. Dessa forma, não é necessária uma posterior síntese *top-down* para obter as variáveis em nível de circuito.

A Figura 2.7 ilustra estas categorias. A escolha de utilizar modelos comportamentais para acelerar as simulações não está necessariamente vinculada ao tipo de metodologia escolhida. No caso da metodologia *bottom-up* multiobjetivo, por exemplo, é possível utilizar diretamente circuitos em vez de modelos comportamentais.

Figura 2.7: Estratégias de síntese em nível de sistema quanto à organização do espaço de projeto.

A Figura 2.8 ilustra a interação entre os níveis de circuito e sistema tanto para a metodologia *bottom-up* multiobjetivo quanto para a metodologia *top-down* com construção de modelos de viabilidade. As setas apontam para o nível de sistema para indicar que a geração da fronteira de Pareto e dos modelos de viabilidade é um processo *bottom-up*. A diferença principal entre estas duas estratégias é que, no caso da construção de modelos de viabilidade, será necessário um posterior processo *top-down* para obter as dimensões dos componentes.

Figura 2.8: Uso de fronteiras de Pareto ou modelos de viabilidade do espaço de critério para síntese de projetos hierárquicos.

De forma mais genérica, algumas tarefas relacionadas à síntese em nível de sistema estão dispostas a seguir:

- **Criação de arquitetura:** dada uma descrição em nível de sistema, esta etapa busca a melhor combinação e interconexão de blocos que é capaz de realizar a função desejada;
- **Criação de modelos de viabilidade e desempenho ou criação de fronteiras de Pareto:** os modelos de viabilidade definem quais parâmetros de projeto podem ser utilizados por determinada topologia e tecnologia. Parâmetros fora da região definida pelo modelo estão em contradição com alguma restrição de projeto ou especificação mínima. Por outro lado, modelos de desempenho são mapeamentos entre as variáveis de entrada de um bloco analógico e suas especificações, de forma que o trabalho computacionalmente exigente da simulação em nível de circuito possa ser dispensado. As fronteiras de Pareto podem ser utilizadas para funções similares às realizadas por estes modelos. Nelas, não é feita a modelagem de todo o espaço de critério, apenas das soluções que representam os *tradeoffs* do projeto.
- **Conversão de especificações:** conversão das especificações de nível de sistema em especificações de nível de circuito para cada bloco. A simulação em nível de sistema é interessante, em grande parte, pois através dela é possível determinar especificações de blocos que conseguem fazer o sistema manter suas mínimas especificações. Desta forma, utilizando os modelos previamente citados juntamente com um método de otimização, pode-se determinar as especificações de blocos.

A criação de arquitetura é um tópico explorado frequentemente com enfoque em um tipo específico de circuitos, como conversores A/D (VITAL et al., 1993). Um número reduzido de trabalhos, no entanto, enfoca uma abordagem genérica para criação de arquitetura para circuitos

analógicos integrados, como as propostas por Doboli et al. (2004) e Martens e Gielen (2006). Estes utilizam um algoritmo do tipo *branch-and-bound* para desdobrar uma descrição abstrata do funcionamento do sistema progressivamente em descrições mais próximas da operação real do mesmo.

As demais áreas de síntese em nível de sistema são abordadas por diversos trabalhos ao longo das últimas décadas. O estado da arte destas áreas serão estudadas nas próximas subseções.

2.1.8.1 Modelos de Viabilidade e Desempenho

Existem algumas variações na literatura no que diz respeito ao escopo de modelos de viabilidade e desempenho aplicados a circuitos integrados analógicos. A maior parte dos métodos propostos, quando referem-se a estes modelos, dizem respeito ao mapeamento de variáveis em nível de circuito (dimensões de componentes) em especificações de blocos. Neste contexto, o espaço de projeto viável (que será traduzido no modelo de viabilidade) consiste no conjunto/região de dimensões de componentes que, quando utilizados, não violam restrições de dimensionamento ou, em alguns casos, não violam estas restrições tampouco um conjunto de especificações mínimas. Ainda, neste contexto, o modelo de desempenho realiza o mapeamento entre os parâmetros de projeto e as especificações de um determinado bloco. Assim, o modelo de desempenho substitui o papel do simulador elétrico durante as otimizações.

Outros trabalhos, no entanto, utilizam estes modelos para mapear a relação das especificações entre si. Dessa forma, estes trabalhos não utilizam as variáveis em nível de circuito como entrada. Este outro contexto estabelece relações matemáticas entre as especificações e permite que um conjunto de especificações seja avaliado quanto à viabilidade/sintetizabilidade independentemente das variáveis de entrada. Enquanto a abordagem anterior apenas aumenta a velocidade de simulação do sistema, esta outra abordagem permite a redução no número de variáveis envolvidas na simulação em nível de sistema.

Para facilitar o entendimento do estado da arte, é utilizado neste trabalho o nome “modelos com entrada em nível de circuito” para representar os modelos que usam variáveis de dimensionamento de componentes como entrada do bloco. Por outro lado, os modelos que, uma vez criados, não utilizam variáveis de dimensionamento de componentes como entrada do bloco são chamados “modelos com entrada em nível de sistema”. Apesar destas variáveis serem fundamentais para a construção dos modelos, caso sejam utilizadas durante a síntese em nível de sistema, a dimensão do problema pode se tornar proibitiva para otimização. Então, torna-se desejável a utilização de modelos comportamentais que contenham apenas as especificações dos blocos e não informações sobre sua estrutura interna. Os modelos que utilizam entrada em

nível de circuito também podem ser chamados de modelos de espaço de desempenho implícito, enquanto os que não utilizam podem ser chamados de modelos de espaço de desempenho explícito (LI et al., 2005).

A seguir, é analisado o estado da arte em trabalhos com modelos com entrada em nível de circuito e sistema.

Modelos com Entrada em Nível de Circuito

Os trabalhos iniciais na construção automática de modelos de blocos analógicos envolvem o uso de análise simbólica dos circuitos. Métodos como o proposto por Lin e Alderson (1972) e outros da época geram apenas expressões exatas e, conseqüentemente, não favorecem o incremento de velocidade na simulação para síntese. Seda et al. (1988) desenvolvem o gerador SYNAP com o intuito de fornecer equações simbólicas simplificadas e também equações exatas dos circuitos. As simplificações de equações são realizadas utilizando informações de pares diferenciais para eliminar algumas variáveis, ignorando capacitâncias e sub expressões pouco relevantes baseadas no tamanho relativo dos parâmetros de circuito. A implementação de SYNAP é feita na linguagem *Common LISP* e utiliza MACSYMA, um programa de matemática simbólica, para realização de operações algébricas. A técnica de análise DC inicialmente utiliza as correntes de polarização de entrada e as relações de espelhos de corrente e pares diferenciais para descobrir todas as correntes dos dispositivos. Após, estas correntes são utilizadas para determinar as tensões dos nós e deixá-las em função das tensões de alimentação e de porta dos transistores. A análise AC é feita através da análise nodal modificada (MNA, do inglês, *Modified Nodal Analysis*). Análise de ruído e *offset* são realizadas posteriormente sobre a matriz da MNA.

A ferramenta ISAAC (do inglês, *Interactive Symbolic Analysis of Analog Circuits*) (GIELEN et al., 1989) também cria equações simbólicas simplificadas a partir dos circuitos. Nesta ferramenta, as equações são simplificadas utilizando um critério heurístico baseado na magnitude dos elementos. Representações de termos de *mismatch* são realizadas para obtenção de especificações precisas de efeitos secundários. Utiliza-se circuitos lineares ou linearizados em uma região através de análise de pequenos sinais para obter equações no domínio da frequência. Da mesma forma que o já discutido SYNAP, o ISAAC foi construído usando a linguagem *Common LISP*. Gielen et al. (1991) apresentam a análise de distorção harmônica em circuitos fracamente não lineares.

Outro trabalho semelhante a estes é a ferramenta ASAP (do inglês, *Analog Symbolic Analysis Program*) (FERNANDEZ et al., 1990) (FERNANDEZ et al., 1991). Esta ferramenta executa o mesmo processo das anteriores, mas é implementada em linguagem C e, portanto, é considerada por seus criadores como mais portátil do que os demais trabalhos da época, bem como

permite um controle maior sobre a intensa alocação de memória necessária para o processo. Fernandez et al. (1992) constroem uma técnica de simplificação que leva em consideração potenciais faixas de variação de diferentes parâmetros simbólicos. Outra proposta para possibilitar a análise simbólica de grandes circuitos foi proposta por Hsu e Sechen (1994). Nesse trabalho, cuidadosas aproximações são realizadas antes da expansão de determinante a fim de reduzir a dimensionalidade e complexidade do mesmo.

Tan e Shi (1999) fazem o uso de Diagramas de Decisão de Determinantes (DDDs) para caracterização de pequenos sinais de circuitos analógicos. Tal método pode, além de derivar funções de rede interpretáveis como magnitude e fase, entregar expressões de polos, zeros e sensibilidade das expressões simplificadas. Daems et al. (2002) constroem um método de simplificação de circuito para análise simbólica baseado em técnicas de manipulação de gráfico de fluxo de sinal. A técnica pode ser utilizada para localizar polos e zeros no circuito e dessa forma gerar funções facilmente interpretáveis.

Apesar das técnicas de análise simbólica avaliarem diretamente uma dada topologia, elas são limitadas a circuitos lineares ou fracamente lineares. Hamad e Massara (1992) propõem uma técnica que envolve decomposição de circuitos e análise numérica para modelagem de blocos. Dado um grande circuito, este é subdividido em pequenas partes funcionais que, por sua vez, são simuladas através do simulador HSPICE. O conjunto de parâmetros de entrada usadas nas simulações é obtido através da atribuição de pequenos passos em todas as variáveis, incluindo parâmetros de carga e de *drive*, no formato de grade. Assim, todos os sub-blocos são modelados e podem servir a simulações mais rápidas posteriormente.

O conceito de modelagem de blocos baseado na utilização de dados já coletados durante a síntese de circuito é apresentado por Liu et al. (2002). Diferentemente de projetos nos quais existe um planejamento de experimentos especificamente voltado para modelagem, este conceito visa reaproveitar dados coletados durante uma síntese realizada previamente. É utilizado um comitê de regressores juntamente com um processo de combinação e votação dos mesmos. Assim, os regressores são aliviados da necessidade de modelar todas as regiões sintetizáveis de projeto, visto que esta pode ser demasiadamente grande. Após o processo de regressão, há seleções de pesos para medir a contribuição que cada regressor deve ter para cada região do espaço. A regressão ocorre através de redes neurais do tipo *feed-forward* implementadas no *software MATLAB*. A criação de modelos de desempenho para avaliação rápida de circuitos utilizando redes neurais também é abordada por Wolfe e Vemuri (2003).

A construção de modelos de desempenho através de um processo determinístico de regressão simbólica foi proposta por McConaghy (2011). Ele realiza a extração das funções dos circuitos através da técnica de aprendizado chamada *pathwise regularized learning*, a qual é

mais rápida que técnicas evolutivas e resulta em modelos *white-box*, ou seja, os quais podem ter sua estrutura analisada.

Modelos com Entrada em Nível de Sistema e Fronteiras de Pareto

Harjani e Shao (1996) realizam a modelagem de blocos analógicos através de uma técnica de aproximação usando funções de base radial. Este trabalho utiliza parte das especificações do bloco como variáveis para criação modelo de viabilidade e parte como critério de desempenho. Doboli e Vemuri (2003) propõem uma notação especial na linguagem VHDL-AMS para separação da descrição da funcionalidade de um bloco de suas especificações/limitações. Desta forma, a definição das especificações torna-se mais fácil já que o usuário ou ferramenta não corre riscos de alterar o funcionamento do bloco durante o processo de análise de desempenho.

No entanto, não bastam modelos comportamentais apenas para realizar a síntese, pois estes não estabelecem entre suas especificações de entrada nenhum tipo de relação. Um modelo comportamental em nível de sistema construído em uma linguagem como VHDL-AMS ou Verilog-A não impede a ferramenta de escolher um conjunto de especificações inalcançáveis quando este bloco necessitar ser construído em nível de circuito. A solução para esta limitação é atrelar modelos comportamentais em nível de sistema com modelos de viabilidade que restrinjam as escolhas possíveis de valores de especificações. Dessa forma, apenas valores que possam ser obtidos durante a síntese em nível de circuito são capazes de serem simulados em nível de sistema.

Bernardinis et al. (2003) propõem a criação de modelos que representem diretamente as relações entre as especificações de um bloco, e não relações entre parâmetros e especificações. Nesse trabalho, SVMs (*Support Vector Machines*) são utilizadas para modelar as especificações de um amplificador de baixo ruído. A coleta de dados é realizada através da escolha aleatória de pontos do espaço de projeto. O uso de uma camada de abstração intermediária entre nível de sistema e nível de circuito é proposto por Bernardinis e Vincentelli (2004). Nesta camada, são encapsuladas a funcionalidade e as especificações atingíveis pelos blocos.

A modelagem do espaço de critério de circuitos analógicos através aproximações de polítopos é realizada por Stehr et al. (2004). Inicialmente, o espaço de projeto é modelado através de restrições, então um mapeamento entre este espaço e espaço de critério é realizado até que, por fim, a aproximação do espaço de critério em si seja realizada. Stehr et al. (2007) também realiza a representação das especificações através da fronteira de Pareto utilizando o método de *Normal-Boundary Intersection* e a eliminação de Fourier-Motzkin.

Fronteiras de Pareto de circuitos podem ser utilizadas de forma similar aos modelos com entrada em nível de sistema. A exploração da fronteira de Pareto para circuitos analógicos tem

sido uma área ativa de pesquisa (SMEDT; GIELEN, 2003) (MUELLER-GRITSCHNEDER; GRAEB, 2010) (WEBER; NOIJE, 2011b) (WEBER; NOIJE, 2012a) (GUERRA-GÓMEZ et al., 2013) (VELASCO-JIMENEZ et al., 2014). Apesar da fronteira de Pareto não ser um modelo, ela contém as soluções correspondentes às trocas de projeto de um determinado bloco. Dependendo da abordagem adotada em síntese em nível de sistema, estas soluções são suficientes para a síntese, visto que permitem separar o espaço de projeto viável do inviável. Ainda, a fronteira de Pareto contém os parâmetros de dimensionamento para consulta, não necessitando que seja refeita a síntese em nível de sistema uma vez tendo sido concluída a conversão de especificações em nível de sistema para nível de circuito.

2.1.8.2 Conversão de Especificações em Nível de Sistema para Nível de Circuito

A conversão de especificações de sistema para os blocos trata da obtenção de um conjunto de especificações para cada bloco de forma que as especificações do sistema sejam respeitadas. Esta conversão é utilizada por trabalhos que realizam a síntese de forma hierárquica.

Nos trabalhos iniciais de tradução de especificações do nível de sistema para nível de circuito, as restrições impostas aos modelos dos blocos eram baseadas na experiência do projetista ou em equações de projeto. Chang et al. (1992) desenvolvem uma metodologia de projeto baseada na propagação de restrição de blocos através dos níveis hierárquicos. Os blocos de cada nível são expandidos até os níveis mais baixos através de um mapeamento de restrições. No caso das especificações de baixo nível não respeitarem as especificações previamente mapeadas, um novo mapeamento é realizado. Donnay et al. (1996) fazem uso de equações elaboradas manualmente para estimar o desempenho dos circuitos durante a conversão de especificações. Mas'ud et al. (1996) apresentam uma forma de conversão de especificações através da criação de modelos de superfície de resposta do sistema. Após estes modelos serem criados, métodos de intervalos são usados para determinar as regiões de projeto viável em nível de sistema. Esta abordagem leva em conta as especificações em nível de sistema, mas não as restrições presentes nos blocos. Modelos declarativos são utilizados por Plas et al. (2001) como fonte de dados para tomar as decisões sobre a viabilidade de um determinado conjunto de especificações de bloco.

O conceito de centralização de desempenho é aplicado à síntese em nível de sistema por Li et al. (2005). Nesse trabalho, o objetivo de maximizar as margens de projeto dos blocos e do sistema é adicionado ao problema de otimização. Ele faz a formulação do problema para sínteses tanto para o caso de modelos com entrada em nível de circuito quanto para o caso de modelos com entrada em nível de sistema. A aplicação deste mesmo conceito utilizando SVMs (*Support Vector Machines*) para representar o espaço de projeto viável é realizado por Bernardinis et al. (2006).

Tabela 2.1: Alguns métodos de otimização.

Método	Publicação
Multiplicadores de Lagrange	(LAGRANGE, 1806)
Programação Quadrática Sequencial	(WILSON, 1963)
Programação Linear	(DANTZIG, 1963)
Programação Geométrica	(DUFFIN BITTNER, 1967)
Algoritmos Genéticos	(HOLLAND, 1975)
<i>Simulated Annealing</i>	(KIRKPATRICK et al., 1983)
<i>Ant Colony Optimization</i>	(COLORNI et al., 1991)
<i>Particle Swarm Optimization</i>	(KENNEDY; EBERHART, 1995)

A utilização de fronteiras de Pareto para realização de síntese em nível de sistema é apresentada por Tiwary et al. (2004). O sistema sintetizado é um PLL (*Phase-Locked Loop*), considerando apenas a fronteira de Pareto do bloco VCO (*Voltage Controlled Oscillator*) e dois objetivos para construir a mesma. No trabalho de Eeckelaert et al. (2005) é proposto um *framework* de síntese utilizando algoritmo SPEA (do inglês, *Strength Pareto Evolutionary Algorithm*). Com este algoritmo multiobjetivo, as fronteiras de Pareto dos blocos são construídas e, posteriormente, os resultados são utilizados para síntese em nível de sistema. A síntese de um *Charge-Pump PLL* considerando a fronteira de Pareto do VCO e do *charge-pump* é realizada por Zou et al. (2006). Ali (2009) também realiza a síntese de um PLL, mas considerando *yield* no projeto.

Os trabalhos utilizando fronteiras de Pareto discutidos até então realizam uma indexação simples da fronteira de Pareto para uso em nível de sistema. Velasco-Jimenez et al. (2014) apresentam uma forma de indexação utilizando um índice por dimensão do projeto. Dessa forma, o otimizador pode navegar pelas soluções da fronteira de Pareto em qualquer dimensão. Esta abordagem é testada na síntese de um filtro passa-baixas contínuo utilizando amplificadores operacionais.

2.2 Métodos Modernos de Otimização

Os métodos de busca que realizam a otimização são o centro do processo de síntese. Portanto, a escolha de um método adequado é essencial para garantir eficiência no processo de otimização. A Tabela 2.1 apresenta alguns exemplos de métodos.

Os métodos clássicos são aqueles baseados em cálculo diferencial. Estes métodos foram explorados amplamente em estudos ao longo da história da matemática. No entanto, tais métodos assumem que a função é duplamente diferenciável com respeito às variáveis de projeto (parâmetros de entrada) e que as derivadas são contínuas. Ainda, são caracterizados por uma busca local, ou seja, movem as soluções sempre no sentido do menor custo possível da função.

Como pode ser visto na Figura 2.4, métodos deste tipo falham em encontrar soluções quando a solução inicial está próxima de um mínimo local. Logo, tais métodos mostram-se inadequados para síntese de circuitos analógicos.

Métodos como programação linear, geométrica e quadrática são utilizados para resolver classes particulares de problemas (RAO, 1996). Tais métodos exigem a construção de equações de formato compatível com o método, de forma que estes métodos são usualmente utilizados para síntese baseada em simulação com avaliação baseada em equações, conforme visto na seção 2.1. A construção de tais equações limita a abrangência de aplicação do método e aumenta o tempo necessário por parte do projetista para que a síntese ocorra.

Os métodos modernos ou não tradicionais são normalmente baseados em analogias físicas ou biológicas e utilizam variáveis aleatórias durante o processo de otimização. São aplicados a problemas para os quais é difícil prever um método de otimização determinístico e, consequentemente, para os quais algum nível de aleatoriedade pode auxiliar a busca por soluções. A vantagem de tais métodos reside em não fazerem suposições sobre o funcionamento do problema e possuem, em diversos casos, mecanismos para fugir de mínimos locais. Usualmente estes métodos não necessitam que a função sendo otimizada seja diferenciável, sendo categorizados como métodos de otimização livre de derivadas. Por não necessitarem formulações prévias sobre o funcionamento de um dado problema, eles têm uma maior aplicabilidade a casos distintos. Sua desvantagem é, para a maior parte dos algoritmos, não garantir a obtenção do mínimo global, apesar de serem eficientes em obter soluções suficientemente boas.

Apesar de diversos métodos de otimização poderem ser utilizados para o problema de síntese de circuitos analógicos, os métodos modernos como *Simulated Annealing* (SA), Algoritmos Genéticos (AG) e *Particle Swarm Optimization* (PSO) são especialmente adequados uma vez que fazem poucas pressuposições quanto às características do problema. Dessa forma, tais métodos podem ser utilizados em uma gama maior de circuitos sem exigir um longo tempo de adaptação do problema ao método.

Nesta seção, é formalizada a definição de problema de otimização mono-objetivo, multiobjetivo e fronteira de Pareto. Ainda, são explorados os mecanismos básicos de funcionamento de alguns métodos modernos de otimização mono e multiobjetivo.

2.2.1 Formalização de Problemas Mono-Objetivo

O problema de otimização mono-objetivo pode ser descrito como na equação 2.8 (RAO, 1996):

$$\text{encontrar } \vec{x} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} \text{ que minimiza } f(\vec{x}) \quad (2.8)$$

sujeito às restrições:

$$h_i(\vec{x}) = 0 \quad i = 1, 2, \dots, p \quad (2.9)$$

$$g_j(\vec{x}) \leq 0 \quad j = 1, 2, \dots, m \quad (2.10)$$

onde $\vec{x}$ é o vetor de projeto n -dimensional, $f(\vec{x})$ é a função custo, $h_i(\vec{x})$ com $i = 1$ até p são as restrições de igualdade e $g_j(\vec{x})$ com $j = 1$ até m são as restrições de desigualdade. Caso $g_j(\vec{x})$ e $h_i(\vec{x})$ não existam, o problema é do tipo sem restrições.

Assim, o espaço de projeto viável é definido conforme a equação 2.11 e o espaço de critério viável conforme a equação 2.12.

$$\mathbf{X} = \{\vec{x} | h_i(\vec{x}) = 0, i = 1, 2, \dots, p; \quad e \quad g_j(\vec{x}) \leq 0, j = 1, 2, \dots, m\} \quad (2.11)$$

$$\mathbf{Z} = \{f(\vec{x}) | \vec{x} \in \mathbf{X}\} \quad (2.12)$$

A Figura 2.9 ilustra o espaço de projeto viável para um problema com três variáveis e apenas restrições de limite mínimo e máximo para as variáveis. A mesma figura também mostra a relação entre o espaço de projeto viável e o projeto de critério variável de uma dimensão. Os algoritmos descritos na seção 2.2.3 são adequados para resolução deste tipo de problema.

2.2.2 Formalização de Problemas Multiobjetivo

Este tipo de problema pode ser descrito como:

$$\text{minimizar } F_{Mo}^{\rightarrow}(\vec{x}) = \{f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})\} \quad (2.13)$$

Figura 2.9: Exemplo de espaço de projeto com três variáveis e espaço de critério unidimensional.

Sujeito à:

$$h_i(\vec{x}) = 0 \quad i = 1, 2, \dots, p \quad (2.14)$$

$$g_j(\vec{x}) \leq 0 \quad j = 1, 2, \dots, m \quad (2.15)$$

onde $F_{MO}(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^k$ é a função custo multiobjetivo que contém as funções custo de cada objetivo, $\vec{x}$ o vetor contendo as variáveis de projeto, n é o número de variáveis, k é o número de objetivos, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, k$ são as funções custo para cada especificação, $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, p$ são as restrições de igualdade e $g_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $j = 1, \dots, m$ são as restrições de desigualdade.

Da mesma forma como descrito anteriormente para problemas mono-objetivo, existem o espaço de projeto viável $\mathbf{X}$ e o espaço de critério viável $\mathbf{Z}$. A Figura 2.10 ilustra-os para um caso multiobjetivo.

Existem duas formas usuais de resolver um problema multiobjetivo: métodos do tipo *scalarization*, e métodos de otimização vetorial (MARLER; ARORA, 2004). A técnica mais popular no método do primeiro tipo é a de soma ponderada das funções custo ou função objetivo agregativa (AOF, do inglês, *Aggregate Objective Function*), como pode ser visto na equação (2.16):

$$AOF = \sum_{i=1}^n f_i \cdot w_i \quad (2.16)$$

onde i é o índice de objetivo, f_i é o custo de uma função individual de uma especificação e w_i é o peso da função individual. Dessa forma, os múltiplos objetivos são transformados em

Figura 2.10: Exemplo de espaço de projeto com três variáveis e espaço de critério bidimensional.

um só, que então é resolvido com um otimizador para objetivo único.

O segundo tipo de método utiliza otimizadores desenvolvidos para múltiplos objetivos, levando em consideração cada objetivo individualmente. Estes buscam não apenas uma solução final, mas um conjunto das melhores soluções, também chamado de conjunto de Pareto. Este conjunto forma a fronteira de Pareto, separando o espaço de critério viável do espaço de critério inviável.

2.2.2.1 Fronteira de Pareto

Uma série de definições são apresentadas a fim de definir a fronteira de Pareto (REYES-SIERRA; COELLO, 2006):

- **Definição 1:** Dados dois vetores $\vec{a}, \vec{b} \in \mathbb{R}^k$ dizemos que $\vec{a} \leq \vec{b}$ se $a_i \leq b_i$ para $i = 1, \dots, k$, e que $\vec{a}$ domina $\vec{b}$ se $\vec{a} \leq \vec{b}$ e $\vec{a} \neq \vec{b}$.
- **Definição 2:** Dizemos que um vetor de variáveis $\vec{x} \in \chi \subset \mathbb{R}^k$ não é dominado com respeito a χ , se não existe outro $\vec{x}' \in \chi$ tal que $\vec{f}(\vec{x}')$ domina $\vec{f}(\vec{x})$, onde $\vec{f}$ é a função que mapeia o vetor de variáveis de projeto nas múltiplas respostas do problema multiobjetivo.
- **Definição 3:** Dizemos que um vetor de variáveis $\vec{x}^* \in \mathbf{X} \subset \mathbb{R}^k$ (onde $\mathbf{X}$ é a região viável) é um Pareto-ótimo se ele não for dominado com respeito a $\mathbf{X}$.
- **Definição 4:** O conjunto de Pareto-ótimo P^* é definido por:

$$P^* = \{\vec{x} \in \mathbf{X} | \vec{x} \text{ é Pareto-ótimo}\} \quad (2.17)$$

Figura 2.11: Exemplo de uma fronteira de Pareto para um espaço de projeto de duas dimensões com dois objetivos de minimização.

- **Definição 5:** A fronteira de Pareto PF^* é definida por:

$$PF^* = \{\vec{f}(\vec{x}) \in \mathbb{R}^k | \vec{x} \in P^*\} \quad (2.18)$$

A Figura 2.11 representa um exemplo de fronteira de Pareto para um problema de duas dimensões no qual os dois objetivos são de minimização. Os pontos âncoras são os resultados ótimos extremos para cada especificação, enquanto os outros pontos da fronteira de Pareto são compromissos entre diversas especificações (*tradeoffs*). Os pontos na área sombreada são dominados por pelo menos uma das soluções da fronteira de Pareto. Por outro lado, uma solução na fronteira de Pareto não pode ser considerada pior que alguma outra também da fronteira de Pareto. Assim, a fronteira representa os melhores resultados possíveis de um processo de otimização multiobjetivo.

É importante ressaltar que a fronteira de Pareto é uma representação dentro espaço de critério viável ($\mathbf{Z}$) e não no espaço de projeto viável ($\mathbf{X}$). Desta forma, duas soluções vizinhas na fronteira de Pareto podem ter valores de variáveis completamente diferentes, dificultando a exploração deste espaço.

2.2.3 Algoritmos Mono-Objetivo

A maior parte dos algoritmos de otimização são propostos com foco na solução de problemas com apenas um objetivo. Assim, há uma grande variedade de algoritmos na literatura para atender a este propósito. Nesta seção, serão estudadas algumas das técnicas mais utilizadas e seus princípios de funcionamento.

2.2.3.1 Estratégia Evolutiva

A Estratégia Evolutiva (ES, do inglês, *Evolution Strategy*) faz parte da classe de Algoritmos Evolutivos, que é composta por algoritmos que baseiam-se no desenvolvimento de uma população guiada através de processos baseados na evolução das espécies. O algoritmo básico de ES foi inicialmente proposto por Rechenberg (1965).

Neste tipo de algoritmo, usualmente as soluções são descritas como valores reais, diferentemente do que ocorre no caso dos Algoritmos Genéticos. Os passos em um algoritmos de ES incluem: seleção para acasalamento, recombinação, mutação e seleção ambiental.

A seleção para acasalamento é o processo de escolha das soluções da geração parental que serão recombinadas para formar soluções filhas. Nas estratégias evolutivas, a seleção pode ocorrer tanto de forma dependente da avaliação da função objetivo, quanto de forma independente. A recombinação gera uma nova solução filha a partir das características das soluções paternas. Então, ocorre a mutação, que realiza uma pequena alteração aleatória em cada indivíduo afetando todas as variáveis. Por fim, a seleção ambiental é o processo de redução do tamanho da população final (parental + gerada) para que esta retorne ao tamanho inicial. Esta redução é realizada por truncamento, de forma que apenas as soluções com melhor desempenho na função objetivo sejam preservadas (HANSEN et al., 2013).

Os algoritmos ES autoadaptáveis são capazes de otimizar características do processo evolutivo durante a própria execução do otimizador. Nesses casos, além dos parâmetros básicos que descrevem uma solução, dentro de um indivíduo em ES também existem parâmetros de estratégia endógenos, conforme descrito por Beyer e Schwefel (2002). Estes parâmetros são responsáveis por controlar características dos operadores genéticos, como o tamanho médio de cada mutação. Dessa forma, enquanto as soluções evoluem, os parâmetros de estratégia evoluem de forma conjunta.

Com estes conceitos em vista, pode-se definir os indivíduos de uma população de ES. Cada indivíduo possui informações sobre solução do problema em um vetor $\mathbf{x} \in \mathbb{R}^n$, um vetor com os parâmetros de estratégia endógenos $\mathbf{s}$ e o resultado da função objetivo $f(\mathbf{x})$.

É comum na descrição de um algoritmos ES o uso da notação $(\mu/\rho^+, \lambda)$. Nela, μ é o número de soluções na população parental, ρ refere-se ao número de pais utilizados para geração de um filho e $^+$ descreve o tipo de seleção utilizada. Caso a seleção seja do tipo “+”, a idade de um indivíduo não é levada em conta durante a etapa de seleção. Por outro lado, os indivíduos “morrem” após uma iteração na seleção do tipo “,”. Dessa forma, apenas indivíduos da prole têm chance de serem preservados para a próxima geração. O λ define o número de filhos criados em cada geração. A Figura 2.12 mostra o pseudocódigo de um algoritmo de Estratégia

```

Inicializar população  $P$ 
novos_indivíduos  $\leftarrow \emptyset$ 
enquanto critério de parada não satisfeito faça
    para  $k$  de 1 até  $\lambda$  faça
        novos_indivíduos[ $k$ ]  $\leftarrow$  recombinação(seleção para acasalamento de  $P$ )
        novos_indivíduos[ $k$ ]  $\leftarrow$  mutação(novos_indivíduos[ $k$ ])
    fim para
     $P \leftarrow P \cup$  novos_indivíduos
     $P \leftarrow$  seleção ambiental( $P$ )
fim enquanto

```

Figura 2.12: Pseudocódigo de um algoritmo de Estratégia Evolutiva.

Evolutiva.

2.2.3.2 Algoritmos Genéticos (AG)

Bem como a estratégia evolutiva (ES), os algoritmos genéticos são uma classe especial dos algoritmos evolutivos e também são baseados no processo de evolução das espécies. As variáveis em algoritmos genéticos são normalmente representadas em código binário ou em inteiros, apesar da representação em variáveis reais ser possível. Ainda, em algoritmos genéticos a seleção das soluções que farão recombinação é baseada na resposta da função custo. De acordo com Luke (2009), a diferença primária entre algoritmos genéticos e a estratégia evolutiva é que em ES, inicialmente, os pais são selecionados e então os filhos são gerados, enquanto que em AG os pais são selecionados aos poucos e vão criando filhos até que um número suficiente tenha sido gerado.

O conceito básico consiste em gerar uma população de soluções para determinado problema, avaliar quais destas são as melhores e consequentemente devem ter mais chance de reproduzirem, e então criar a nova geração da população. O processo é iterado diversas vezes de forma que gradualmente a população aperfeiçoe-se e busque o ótimo global. Uma das grandes vantagens desta técnica é a capacidade de realizar movimentos de exploração genuína (através da operação de mutação) e movimentos de exploração direcionada (através do operador de *crossover*).

As etapas básicas presentes na técnica de algoritmos genéticos são:

- **Avaliação:** processo no qual cada indivíduo da população é avaliado quanto a proximidade do objetivo desejado. Ou seja, é a aplicação da função custo em todas as soluções existentes na população;
- **Seleção (Reprodução):** consiste na escolha de quais *strings* de dados (em analogia ao

```

Inicializar população  $P$ 
 $N \leftarrow$  tamanho da população
enquanto critério de parada não satisfeito faça
  para  $i$  de 1 até  $N$  faça
    Avaliar( $P_i$ )                                      $\triangleright$  indivíduos da população  $P$ 
  fim para
   $Q \leftarrow \emptyset$ 
  para  $k$  de 1 até  $N/2$  faça
    Filhos  $A$  e  $B \leftarrow$  recombinação(seleção( $P$ ))
    Filhos  $A$  e  $B \leftarrow$  mutação(Filhos  $A$  e  $B$ )
     $Q \leftarrow Q \cup$  Filhos  $A$  e  $B$ 
  fim para
   $P \leftarrow Q$ 
fim enquanto

```

Figura 2.13: Pseudocódigo de um Algoritmo Genético (baseado no descrito em Luke (2009)).

DNA) serão utilizadas para que sobre eles ocorra crossover com outras *strings*. Cada uma destas *strings* constitui uma solução, ou seja, um indivíduo.

- **Crossover:** consiste na mistura e combinação de partes de diferentes pais (soluções pertencentes a geração anterior) para a formação de crianças (soluções pertencentes a nova geração).
- **Mutação:** apenas o processo de crossover não é capaz de fazer uma exploração adequada do espaço de projeto, pois pode resultar em convergência prematura. Desta forma, a mutação é uma componente aleatória adicionada ao algoritmo para que exista a possibilidade de novas soluções sejam geradas.

Na Figura 2.13 é possível ver o pseudocódigo dos Algoritmos Genéticos.

2.2.3.3 Simulated Annealing

O *Simulated Annealing* é um algoritmo de busca proposto por Kirkpatrick et al. (1983). Ele tem a habilidade de escapar de mínimos locais através da aceitação de soluções piores do que a presente dada uma certa probabilidade.

O nome *Simulated Annealing* (recozimento simulado) deriva do processo físico chamado recozimento no qual o algoritmo é baseado. Este processo é comumente utilizado em metalurgia e consiste no aquecimento de um sólido cristalino a uma temperatura alta e depois seu resfriamento lento até que sua configuração cristalina esteja regular. Devido a sua alta temperatura inicial, o movimento dos átomos segue um padrão aleatório. À medida que a temperatura decresce, os movimentos são reduzidos lentamente para permitir que apenas estruturas mais regulares sejam aceitas. No final do processo, obtém-se uma superfície livre de imperfeições.

```

 $t \leftarrow$  temperatura inicial
 $S \leftarrow$  solução candidata inicial
 $melhor \leftarrow S$ 
enquanto  $t \geq t_{min}$  faça
     $R \leftarrow$  varia(cópia( $S$ ))                                ▶ Função de geração
    se ( $\text{custo}(R) < \text{custo}(S)$ ) ou ( $\text{rand} < P(t, R, S)$ ) então    ▶ Critério de aceitação
         $S \leftarrow R$ 
    fim se
    resfriamento( $t$ )                                            ▶ Cronograma de resfriamento
    se  $\text{custo}(S) < \text{custo}(melhor)$  então
         $melhor \leftarrow S$ 
    fim se
fim enquanto

```

Figura 2.14: Pseudocódigo de um algoritmo de *Simulated Annealing*.

No algoritmo, a temperatura é um parâmetro utilizado para cálculo da probabilidade de aceitação de uma resposta pior que a atual. A alta temperatura inicial permite que novas soluções piores que as anteriores sejam aceitas. A medida que a temperatura decresce, o algoritmo reduz a probabilidade de aceitação destas respostas piores e começa a permitir que apenas soluções melhores sejam adotadas.

O primeiro passo do algoritmo é avaliar a solução inicial. A partir disso, o ciclo descrito a seguir é realizado:

- **Modificação (função de geração):** uma pequena modificação é realizada na solução atual para que obtenha-se uma solução candidata (nova);
- **Avaliação (função custo):** a solução candidata é avaliada através de uma função custo;
- **Decisão (função de aceitação):** caso a solução candidata seja melhor que a atual, ela é mantida. Caso seja pior, baseado na temperatura atual e na diferença de custo entre a solução atual e a candidata, é calculada a probabilidade de aceitação da solução e comparada com um valor aleatório para decidir se a solução candidata é adotada ou descartada.
- **Redução do parâmetro de controle (cronograma de resfriamento):** a temperatura do sistema é reduzida para que a probabilidade de aceitação de soluções piores seja gradualmente diminuída.

Existe um grande número de variações do *Simulated Annealing* (PEREIRA; FERNANDES, 2004). Muitas destas variações tem por objetivo acelerar a convergência do algoritmo para uma resposta suficientemente boa. O pseudocódigo 2.14 mostra o algoritmo do *Simulated Annealing* de forma simplificada. O critério de parada do algoritmo pode ser baseado na temperatura mínima, no número de iterações, no menor custo admissível, entre outros.

No critério de aceitação, a nova solução é selecionada baseada na decisão descrita na equação (2.19):

$$S_{k+1} = \begin{cases} N, & \text{se } \text{Custo}(N) < \text{Custo}(S_k) \text{ ou } \tau \leq P(t_k, N, S_k) \\ S_k, & \text{caso contrário} \end{cases} \quad (2.19)$$

onde k é o índice da solução, N é a solução candidata, S_{k+1} é a próxima solução, τ é uma variável aleatória uniformemente distribuída entre $[0, 1]$ e $P(t_k, R, S_k)$ é a probabilidade de aceitação de uma solução pior dada uma temperatura t_k e uma diferença de custo.

A probabilidade de aceitação de uma solução pior, também chamada de critério de aceitação, é a equação que define a probabilidade de uma resposta pior ser aceita através da temperatura e diferença de custo. A equação adotada é usualmente o critério de Metropolis, que pode ser visto na equação 2.20:

$$P(t_k, N, S_k) = e^{-\frac{\text{Custo}(S_k) - \text{Custo}(N)}{t_k}} \quad (2.20)$$

As variações do algoritmo de *Simulated Annealing* têm diversas diferenças entre si, principalmente na função de geração e o cronograma de resfriamento. A função de geração tem por objetivo a criação de uma nova solução com base na anterior de forma a causar uma pequena perturbação na resposta. Já o cronograma de resfriamento diminui a possibilidade de aceitação de respostas piores que a atual ao longo do tempo. Ou seja, este cronograma determina o quão rápido as respostas piores passarão a ser de difícil aceitação.

Locatelli (2002) faz uma revisão dos métodos existentes de SA para lidar com problemas de otimização global contínua. No trabalho de Bohachevsky et al. (1986), o cronograma de resfriamento determina a temperatura com base na distância da resposta atual da (suposta) resposta ideal, como pode ser visto na equação [2.21]:

$$t_k = \beta[C(S_k) - C^*]^{g_1} \quad (2.21)$$

onde t_k é a temperatura, β e g_1 são constante, $C(S_k)$ é o custo da solução atual e C^* é o custo da solução ideal. Caso não exista tal custo, uma estimativa do custo da solução ideal deve ser fornecida. Nesta equação, a temperatura é reduzida a medida que o custo da solução atual se aproxima da solução ideal. Isto permite que uma fase menos explorativa e mais voltada ao refinamento ocorra quando a solução encontra-se próxima do resultado desejado. Nesta abordagem, a função de geração pode ser vista na equação [2.22]:

$$N_k = S_k + \Delta r \Theta_k \quad (2.22)$$

onde N_k é a nova solução proposta, S_k é a solução atual, Δr é um passo de tamanho fixo e

Θ_k é um vetor de direção com tamanho igual ao número de variáveis na solução. Este vetor tem $\|\Theta_k\| = 1$ e é gerado randomicamente a cada iteração. Desta forma, a perturbação das soluções ocorre em todas as variáveis simultaneamente e com passo fixo, porém com sentidos variados. Brooks e Verдини (1988) sugerem que os valores de β e Δr sejam escolhidos com base em um pré-amostragem da função objetivo. Desta forma, β e Δr devem ser definidos de forma que a porcentagem dos movimentos de piora aceitos em relação ao total de movimentos de piora gerados seja aproximadamente 60%. Ainda, é recomendada a divisão da otimização em uma fase global, na qual o algoritmo explora o espaço de projeto, e uma fase local, na qual buscase refinar as respostas através da redução do valor do Δr , resultando em menores passos, e β , resultando em menores temperaturas.

Uma extensão do conceito de ajustar a temperatura com base na distância entre a resposta atual e a ideal é utilizado por Wang e Chen (1996), no qual a mesma ideia é aplicada também ao ajuste do passo de modificação. Ainda, é proposto que a direção das respostas geradas não seja baseada em uma distribuição uniforme (conforme apresentado por Bohachevsky et al. (1986)), mas sim baseado no resultado prévio. Caso a última modificação tenha oferecido uma resposta melhor que a anterior, a distribuição favorece a direção tomada naquele movimento. Caso tenha sido pior, ela favorece o sentido oposto.

Os trabalhos descritos até então trabalham com funções de geração com movimento isotrópico no espaço de projeto. Isto significa que, apesar de definirem o sentido do movimento de forma variada, não existe distinção no tamanho do passo para diferentes direções. No espaço de projeto podem existir situações nas quais passos pequenos em uma dimensão (uma variável) sejam interessantes, enquanto apenas passos grandes em outra dimensão possam causar efeito significativo. É preciso, então, a utilização de uma movimentação anisotrópica pelo espaço de projeto. Vanderbilt e Louie (1984) realizam esta abordagem através do uso da matriz Hessiana (matriz contém as derivadas parciais de segunda ordem de uma função) no ponto do ótimo global. Supondo o conhecimento desta matriz Hessiana H^* e definindo uma matriz P como a inversa da mesma, pode-se utilizar a decomposição de Choleski para encontrar a matriz Q definida como:

$$P = QQ^T \quad (2.23)$$

e então utilizar a matriz Q para gerar o novo valor de resposta:

$$N_k = S_K + Qu \quad (2.24)$$

onde u é um vetor com dimensão igual ao número de variáveis (da mesma forma que N e S) e com os valores distribuídos uniformemente entre $[-\sqrt{3}, \sqrt{3}]$. Naturalmente, esta matriz Hessiana no ponto ótimo não é conhecida para o problema de otimização e não pode ser

utilizada para determinar P . No entanto, uma estimativa é realizada periodicamente (após um número arbitrário de iterações) para tentar aproximar P e assim poder gerar a próxima solução. Neste mesmo trabalho, o cronograma de resfriamento é calculado através da equação a seguir:

$$t_k = t_0 \cdot (\chi_T)^{k/M} \quad (2.25)$$

onde t_0 é a temperatura inicial, χ_T é um fator de resfriamento entre 0 e 1, M é o número de iterações entre cada execução do cronograma de resfriamento e k é o índice total de iterações.

Outro estudo que também utiliza a exploração anisotrópica do espaço de projeto foi realizado por Corana et al. (1987). No entanto, diferentemente da abordagem anterior, apenas uma variável é modificada a cada iteração. Desta forma, a função de geração fica como:

$$N_k = S_K + r v_i e_i \quad (2.26)$$

onde r é um número aleatório extraído de uma distribuição uniforme no intervalo $[-1,1]$ e v_i é o máximo passo permitido na direção da variável e_i . Os valores de v_i são mantidos por um número fixo de iterações N , após o qual todos são recalculados. A regra para modificação do máximo passo permitido abaixo:

$$v_i \text{ é } \begin{cases} \text{aumentado,} & \text{se } P_A < 0,4 \\ \text{reduzido,} & \text{se } 0,4 < P_A < 0,6 \\ \text{mantido,} & \text{se } P_A > 0,6 \end{cases} \quad (2.27)$$

na qual P_A é a fração de movimentos aceitos na direção e_j em relação ao número total de movimentos gerados naquela direção. O propósito deste mecanismo é propiciar um aumento de tamanho de passo para as variáveis as quais estão tendo pouca aceitação na modificação. O cronograma de resfriamento está descrito a seguir:

$$t_k = t_0 \cdot \chi_T^{\left(\frac{k}{n N_S N_T}\right)} \quad (2.28)$$

onde n é o número de variáveis, N_S é o número de ciclos no qual o valor v (equação 2.27) é mantido fixo e N_T é o número de alterações de v até que o resfriamento ocorra.

A prova de convergência para o mínimo global no algoritmo de *Simulated Annealing* existe para um esquema de redução de parâmetro de controle logarítmico. A fim de obter soluções mais rapidamente, esquemas de redução de temperatura que não possuem prova de convergência derivados do SA original são utilizadas em diversas aplicações. Apesar da falta de prova teórica, estes algoritmos utilizam a mesma estrutura do *Simulated Annealing* e são frequentemente melhores para o subconjunto de problemas para os quais são criados. É comum chamar

Figura 2.15: Influência dos fatores na velocidade da partícula no PSO.

este tipo de variação de *Simulated Quenching*.

2.2.3.4 Particle Swarm Optimization (PSO)

PSO é um algoritmo de busca populacional que imita as habilidades de solução de problema existente em grupos de animais. Mais especificamente, ele se baseia na simulação do comportamento social de aves dentro de um bando (REYES-SIERRA; COELLO, 2006). Foi proposto pela primeira vez como um algoritmo para otimização por Kennedy e Eberhart (1995). O algoritmo funciona permitindo que múltiplos indivíduos cooperem uns com os outros como um grupo trocando informações. Os indivíduos são chamados de *partículas* e têm a capacidade de *voar* pelo espaço de projeto na busca de soluções de custo mínimo. Em cada iteração, as partículas têm a sua posição atualizada através da velocidade de voo. Esta velocidade é calculada com base na experiência de cada partícula e da experiência do grupo como um todo ou a partir da vizinhança da partícula. A velocidade pode ser calculada através da equação 2.29 (KENNEDY; EBERHART, 1995):

$$\vec{v}_i(t) = \underbrace{w\vec{v}_i(t-1)}_{\text{fator de inércia}} + \underbrace{c_1 r_1 (\vec{x}_{lbesti} - \vec{x}_i(t))}_{\text{fator do melhor local}} + \underbrace{c_2 r_2 (\vec{x}_{gbest} - \vec{x}_i(t))}_{\text{fator do melhor global}} \quad (2.29)$$

onde w , c_1 e c_2 são constantes que definem a influência de cada fator na velocidade final, r_1 e r_2 são variáveis randômicas com distribuição uniforme entre $[0,1]$, $\vec{v}_i$ é o vetor de velocidade, t é o tempo (ou o número de iterações), $\vec{x}_{lbesti}$ é o melhor resultado local da partícula i e $\vec{x}_{gbest}$ é o melhor resultado global ou da vizinhança. O primeiro termo do lado direito da equação é o fator de inércia, o segundo é a influência da melhor solução local encontrado até o momento e o terceiro termo é a influência do melhor solução global ou melhor solução da vizinhança. A Figura 2.15 ilustra o cálculo da velocidade a partir dos seus fatores. A Figura 2.16 mostra o pseudocódigo de um algoritmo *Particle Swarm Optimization*.

```

Inicializar partículas
Atualizar melhorglobal
enquanto critério de parada não alcançado faça
    para cada partícula  $k$  faça
        Atualizar velocidade[ $k$ ]
        Atualizar posição[ $k$ ]
        Avaliar posição[ $k$ ]
        Atualizar melhorlocal[ $k$ ]                                ▶ melhor da partícula
    fim para
    Atualizar melhorglobal                                ▶ melhor da vizinhança
fim enquanto

```

Figura 2.16: Pseudocódigo de um algoritmo de *Particle Swarm Optimization* com vizinhança totalmente conectada.

A topologia de conexão entre as partículas define como uma vizinhança é formada. Diversas topologias diferentes podem existir (REYES-SIERRA; COELLO, 2006). O uso de PSO para síntese de circuitos analógicos é realizado por Fakhfakh et al. (2010) em uma abordagem de avaliação baseada em equação para um problema mono-objetivo e para um biobjetivo.

2.2.4 Algoritmos Multiobjetivo

Esta seção trata dos algoritmos que foram desenvolvidos com foco na solução de problemas multiobjetivo. Os métodos principais para resolver problemas multiobjetivo fazem uso de funções agregativas ou utilizam o conceito de dominação. A seguir, estas técnicas são mostradas.

2.2.4.1 Técnicas Utilizando Funções Agregativas

Métodos para obtenção da fronteira de Pareto que utilizam funções agregativas mantêm o uso de algoritmos do tipo mono-objetivo. No entanto, estes métodos exploram o espaço de critério multidimensional através da execução destes algoritmos diversas vezes para diferentes pesos da função agregativa.

De acordo com Das e Dennis (1997), uma limitação destas técnicas é a incapacidade de convergir para soluções da fronteira de Pareto que estejam em uma região não convexa. Dessa forma, caso não seja conhecido previamente que a forma da fronteira de Pareto de um problema seja convexa, usar estas técnicas envolve o risco da não obtenção de soluções uniformemente distribuídas por toda a extensão da fronteira de Pareto. Algumas das técnicas que fazem uso de funções agregativas e ainda propõe formas de atingir soluções uniformemente distribuídas são a *Normal-Boundary Intersection* (NBI) (DAS; DENNIS, 1998), a Otimização de Pareto Sucessiva (MUELLER-GRITSCHNEDER et al., 2009) e o *Directed Search Domain* (DSD) (ERFANI; UTYUZHNIKOV, 2010).

2.2.4.2 Técnicas Evolutivas Multiobjetivo

Diversos estudos sobre o uso de métodos evolutivos para resolução de problemas multi-objetivo foram realizados durante a década de 1990. De suas análises, ficaram estabelecidas algumas características a serem buscadas por técnicas deste tipo, como a classificação de respostas baseado em sua dominação sobre as outras e a diversidade das soluções dentro de uma mesma fronteira de não dominação (DEB et al., 2002).

Para tanto, muitos dos métodos evolutivos atuais utilizam o conceito de otimalidade de Pareto durante sua busca e seleção de soluções. Isto significa que a avaliação das soluções não ocorre diretamente através da resposta de apenas uma função custo. Para definir se uma resposta é interessante, esta é avaliada em relação a todas as demais da população para saber se ela é dominada ou não. Em muitos desses métodos, os indivíduos de uma população são organizados de forma com que os não dominados tenham uma categoria inferior aos dominados. Esta categorização garante menor probabilidade de seleção aos indivíduos dominados. Dois dos algoritmos evolutivos multiobjetivo modernos mais utilizados são o *Nondominated Sorting Genetic Algorithm II* (NSGA-II) (DEB et al., 2002) e o *Strength Pareto Evolutionary Algorithm* (SPEA2) (ZITZLER et al., 2001).

No NSGA-II, para uma dada solução p , é calculado o número de soluções n_p que a dominam, e um conjunto de soluções S_p que a solução p domina. O conjunto das soluções não dominadas recebem categoria zero ($i_{categoria} = 0$). A seguir, para cada solução não dominada ($n_p = 0$), as soluções contidas em seu conjunto S_p tem seu valor de n_p reduzido em um. Durante este processo, as soluções que alcançarem $n_p = 0$ são adicionadas a um conjunto Q que contém as soluções com categoria = 1. O processo continua até que todas as fronteiras tenham sido calculadas. A fim de manter a diversidade da população, é utilizado um operador de comparação de lotação (do inglês, *crowded-comparison operator*). Para que isto seja possível, é definida a densidade em um valor $i_{distncia}$ para cada solução calculando-se a distância média de dois pontos em cada lado da solução para cada objetivo. Assim, as soluções são avaliadas baseadas tanto no seu categoria ($i_{categoria}$) quanto na sua densidade ($i_{distncia}$). Entre soluções de categorias diferentes, as com categoria inferior têm sempre preferência. Entre soluções de uma mesma categoria, as que tem menor densidade têm preferência.

O SPEA2 (e também seu predecessor SPEA) usa duas populações: uma regular e uma para arquivo externo. Inicialmente, a população regular é preenchida com soluções e a população para arquivo externo permanece vazia. A cada iteração, as soluções não dominadas da população regular são colocadas no arquivo externo. Caso o número de soluções que devem ingressar no arquivo externo seja superior a um limiar previamente definido, um processo de truncamento ocorre. A seguir, cada solução i do arquivo externo é avaliada com base no número de indivíduos

da população regular que ela domina para formar seu valor $S(i) \cup [0, 1)$. Então, cada solução j da população regular é avaliada com base na soma dos valores de $S(i)$ de todas as soluções da população do arquivo que a domina, formando assim o seu valor $F(j)$. Por fim, uma seleção por torneio binário com reposição ocorre para sua recombinação gerar a próxima população. No SPEA2 o método de truncamento não perde valores dos limites. Ainda, no SPEA2 apenas soluções do arquivo externo podem participar do processo de seleção de parceiros.

2.2.4.3 PSO Multiobjetivo

Para problemas multiobjetivo, o caso geral de PSO precisa receber algumas mudanças a fim de atender as características principais de algoritmos desta natureza.

Para manipular a chamada experiência na otimização multiobjetivo, cada partícula precisa ter uma lista com as melhores soluções encontradas por ela até o momento, bem como o grupo como um todo precisa ter uma lista global com todas as candidatas a fronteira de Pareto já visitadas por qualquer uma das partículas. Para uma nova solução entrar na lista, esta precisa não ser dominada por nenhuma das soluções já existentes na lista. Ainda, a cada iteração, as soluções antigas da lista precisam ser checadas para analisar se não foram dominadas pelas novas soluções. A seleção de um líder local de uma lista de Paretos locais de cada partícula e a seleção do líder global da lista de Paretos globais pode ser baseada em diversos critérios. Um destes critérios pode ser a densidade de soluções na lista. Nesse caso, o líder é o que possui menos soluções próximas a ele já incluídas na lista. Isto permite que a otimização busque soluções em regiões novas, em vez de rapidamente convergir para uma solução igual em todas as partículas.

2.3 Resumo do Capítulo

Neste capítulo, foram discutidas as peças-chave da síntese de circuitos integrados analógico, tais como as variáveis de projeto, medidas de desempenho, função objetivo e espaço de projeto. Ainda, uma análise das características desejáveis a uma ferramenta de síntese analógica foi realizada, bem como foi buscado o estado da arte da síntese em nível de circuitos e em nível de sistemas. Os métodos modernos de otimização foram estudados tanto para problemas mono-objetivo quanto para problemas multiobjetivo.

3 ALGORITMOS DE OTIMIZAÇÃO PROPOSTOS

Os algoritmos de otimização desenvolvidos neste trabalho têm foco na síntese eficiente de circuitos integrados analógicos. Para tanto, em sua elaboração foram levadas em consideração características próprias deste tipo de problema e características de métodos modernos de otimização.

Conforme apresentado no *International Technology Roadmap for Semiconductors* (ITRS, 2011), há muitas questões ainda não satisfatoriamente resolvidas para a realização da síntese analógica, como a exploração do espaço de projeto e otimização de variáveis discretas. Além do algoritmo proposto ser capaz de escapar de mínimos locais através de novos recursos utilizando múltiplos objetivos, ele também tem mecanismos que o fazem intrinsecamente capaz de lidar com variáveis contínuas e discretas simultaneamente.

Como pode ser visto no capítulo anterior, as técnicas de síntese de circuito integrado baseadas em otimização são muito versáteis e são as mais utilizadas em ferramentas do estado da arte em síntese. Enquanto todos os blocos de uma ferramenta de síntese têm importância fundamental, o algoritmo de otimização é o centro de decisões do processo. Cabe a ele articular os passos na busca por um circuito melhor, sendo necessário que seja capaz de lidar eficientemente com o tipo de problema sendo resolvido. Visto que pode ser utilizado de forma modular, um otimizador eficiente pode ser utilizado em diferentes ferramentas baseadas em otimização, aumentando a contribuição de trabalhos nesta área de pesquisa.

Além de poder atuar em ferramentas distintas, o algoritmo de otimização pode ser utilizado em diferentes etapas dentro de uma mesma ferramenta. No caso da síntese buscando um circuito final único ou na síntese em nível de sistema, é utilizada uma versão mono-objetivo do algoritmo. No caso de uma síntese analógica buscando a exploração das trocas de projeto, uma variação do algoritmo considerando critérios de dominação é mais apropriada.

Assim, a elaboração do algoritmo de otimização é adotada neste trabalho como a etapa principal na criação da ferramenta de síntese. Na seção 3.1 é apresentado o algoritmo proposto para problemas mono-objetivo e, na seção 3.2, o algoritmo proposto para problemas multiobjetivo. Após estas descrições, testes com problemas matemáticos são realizados a fim de comparar o

desempenho do algoritmo proposto com outros algoritmos encontrados na literatura. Os testes com circuitos são mostrados no capítulo seguinte, após os demais blocos da ferramenta de síntese terem sido apresentados.

3.1 Descrição do Algoritmo Proposto para Problemas Mono-objetivo

O mecanismo básico do algoritmo proposto é baseado no *Simulated Annealing* (SA). A variação proposta altera diversas etapas em relação ao algoritmo tradicional, desde o cronograma de resfriamento, função de geração, critério de aceitação até mecanismos adicionais de regulação. Ainda, modificações foram realizadas a fim de propiciar a fuga de regiões de mínimos locais através do uso de informações provenientes de múltiplos objetivos.

Quando se trata da obtenção de apenas uma solução final, a utilização da conversão de múltiplos objetivos em um único é característica existente na maior parte das técnicas de síntese analógica de circuitos integrados. Esta conversão ocorre, na maior parte das vezes, através da soma ponderada da resposta de cada um dos objetivos. A equação resultante pode ser chamada de função objetiva agregativa (AOF, do inglês, *Aggregate Objective Function*).

O algoritmo proposto faz uma abordagem mista entre o uso da função objetiva agregativa e o uso dos múltiplos objetivos individualmente. As informações a respeito do desempenho das soluções em cada objetivo são repassadas ao algoritmo juntamente com a resposta da AOF. Durante o processo de otimização, o algoritmo utiliza fundamentalmente a AOF para direcionar a busca. No entanto, ao encontrar uma situação de mínimo local do qual o mecanismo básico do SA não consiga escapar, o algoritmo faz uso das informações multiobjetivo. Com elas, ele escolhe a melhor combinação da solução atual com soluções prévias para libertá-lo do mínimo. Esta abordagem situa o algoritmo proposto entre a exploração mono-objetivo e a exploração multiobjetivo.

É possível relacionar essa abordagem a aquela de como um projetista desenvolve circuitos. Ele analisa as especificações individualmente e também o conjunto para tentar melhorar o projeto. Ainda assim, o projetista não explora exaustivamente todas as possíveis trocas de projeto, pois, em geral, tem interesse em apenas um único circuito final que atenda às especificações. O mesmo princípio rege o algoritmo proposto, que utiliza informações de múltiplos objetivos para escapar de mínimos locais, mas, ao conseguí-lo, continua a busca por apenas uma solução final.

Nas próximas seções serão exploradas algumas das características do algoritmo a fim de fornecer uma melhor compreensão sobre seu funcionamento.

3.1.1 Passo Adaptativo

Neste trabalho, é utilizada uma variável auxiliar para cada variável de projeto. Esta variável auxiliar define o desvio padrão do passo da variável relacionada a ela. O valor desta variável auxiliar é atualizado após cada nova solução e reflete quanto a solução melhorou a função custo em relação a solução anterior. Desta forma, um passo adaptativo é fornecido para cada variável.

Caso a solução gerada forneça um movimento de grande melhoria, o desvio padrão é mantido. Caso a solução gere um resultado pior que o anterior, o desvio padrão é reduzido. Caso a modificação ofereça apenas um pequeno benefício, o desvio padrão é aumentado. A ideia por trás destas modificações é aumentar o passo de variações que estejam causando apenas pequenas modificações na função custo, reduzir o passo de variações que estejam prejudicando a função custo e manter o passo de funções que estejam causando bom impacto na função custo.

A equação da função de geração das variáveis de projeto utilizada é mostrada na equação (3.1):

$$x'_i = x_i \cdot (1 + f(0, 1) \cdot aux_i) \quad (3.1)$$

onde x'_i é o novo valor da variável i sendo modificada, x_i é o antigo valor da variável sendo modificada, aux_i é a variável auxiliar, $f(0, 1)$ é uma variável aleatória com distribuição normal, média zero e desvio padrão unitário (que será modificado pela variável auxiliar). O desvio padrão σ_i da variável i será igual a $x_i \cdot aux_i$.

Como as alterações na variável auxiliar são efetuadas a cada iteração, e não após um conjunto de iterações, estas devem ser de pequena ordem para não alterar muito rapidamente o desvio padrão. Assim, grandes variações do desvio padrão ocorrem apenas após o efeito acumulado de diversas iterações. Permitindo a alteração de mais de uma variável simultaneamente, torna-se possível movimentos em direções diferentes das direções dos eixos.

O uso deste passo adaptativo é interessante para problemas que tenham variáveis contínuas e discretas, como pode ser o caso de circuitos analógicos. Caso a alteração de uma variável não resulte em nenhuma variação na função custo, isto pode decorrer desta variável estar sendo arredondada por ser discreta. Desta forma, o incremento do desvio padrão destas variáveis ajuda o algoritmo a não estagnar devido ao arredondamento. Como cada variável tem seu próprio desvio padrão, as variáveis contínuas continuarão com o desvio padrão apropriado para suas modificações.

3.1.2 Multiobjetivação

O processo de multiobjetivação consiste em transformar um problema mono-objetivo em um problema multiobjetivo com o intuito de facilitar sua otimização. Esse processo é realizado pela decomposição do problema original em um conjunto de subproblemas, ou pela adição de problemas secundários ao problema original. O efeito que faz com que os problemas multiobjetivo possam ser mais fáceis de solucionar é a diminuição de mínimos locais, uma vez que com múltiplos objetivos há um número maior de soluções que não são melhores ou piores que as outras devido às diversas métricas.

Knowles et al. (2001) apresenta uma relação desejada entre o problema original mono-objetivo e o problema multiobjetivo para que esta conversão seja eficiente:

$$\forall x^{opt} \in X^*$$

onde x^{opt} é o mínimo global do problema original e X^* é o conjunto de Pareto-ótimo do problema que sofreu multiobjetivação. Assim, ao algoritmo multiobjetivo buscar a fronteira de Pareto, alcançará também o ponto ótimo do problema mono-objetivo. Nos últimos anos, diversos trabalhos na literatura exploram este conceito e têm obtido resultados indicando que esta técnica é promissora, como nos trabalhos de Jensen (2004) Brockhoff et al. (2009) e Lochtefeld (2011). Segura et al. (2013) faz um revisão dos trabalhos nos quais o conceito de multiobjetivação é utilizado.

Problemas de síntese em nível de circuito já são, em sua maioria, multiobjetivo. Na maior parte das abordagens que buscam uma única solução, a informação contida nestes múltiplos objetivos é descartada em prol da simplicidade de uma única função custo que agregue todos os objetivos. Assim, é possível não descartar estas informações para utilizá-las afim de escapar de mínimos locais.

No presente trabalho, é explorada a ideia que a melhor forma de utilizar a multiobjetivação é através de um algoritmo que consulte os múltiplos objetivos apenas quando necessário, ao contrário de buscar explorar o espaço multicritério como um todo. Dessa forma, um número maior de objetivos não sobrecarrega o otimizador na busca por toda a fronteira de Pareto, mas permite que ele escape de mínimos locais de forma mais fácil. Essa fuga é possível pois o algoritmo salva soluções promissoras durante a otimização baseado nos múltiplos objetivos. Ao precisar escapar de um mínimo, é realizada a recombinação entre a melhor solução e uma das soluções prévias. Além disso, uma função custo auxiliar com uma alteração nos pesos é fornecida ao critério de aceitação do algoritmo, de forma que mínimos locais da função AOF principal não necessariamente sejam também mínimos locais da função auxiliar. Ao fazer o

algoritmo reconhecer movimentos de melhoria em qualquer das duas funções custo (a original e a auxiliar) como positivas, há uma possível redução no número de mínimos locais enquanto mantém-se a topologia da função custo original como referência.

A Figura 3.1 ilustra um caso em que as funções individuais podem auxiliar a busca pelo objetivo principal. Ela é uma adaptação da função Rastrigin (que será apresentada na Tabela 3.1) para 3 objetivos utilizando apenas uma variável para que o resultado possa ser facilmente observado. Caso um algoritmo esteja preso em algum dos mínimos locais identificados de 1 a 6, F2 é a função individual que tem maior custo neste trecho. Caso o algoritmo passar a se guiar também por melhorias nesta função, será guiado até o mínimo global. De forma semelhante, caso o algoritmo esteja preso em algum dos mínimos locais identificados de 7 a 12, F3 será a função individual com maior custo e que também guiará o algoritmo para o mínimo global caso seja utilizada. A razão deste benefício é que, nesta função, os mínimos locais da AOF estão em vales mais profundos que os vales existentes nas funções individuais.

Há situações em que esta multiobjetivação não tem benefício, como no exemplo da Figura 3.2. A função é uma adaptação da função Michalewics (Tabela 3.1). Apesar das funções individuais comporem a topologia da função AOF, neste caso elas não são capazes de fornecer informações úteis ao algoritmo para que este seja guiado ao mínimo global.

Um algoritmo que tenha acesso a informação multiobjetivo deve, portanto, utilizá-la sem depender exclusivamente dela. É uma informação com potencial de guiar para regiões melhores do espaço de projeto, mas não em todos os casos. Por este motivo, o algoritmo apresentado neste trabalho utiliza tanto a AOF original quanto uma AOF secundária (com mais peso para objetivos que tenham mal desempenho).

3.1.3 Operador de Crossover em Algoritmo Não-Populacional

Diversos algoritmos que integram *Simulated Annealing* com algoritmos genéticos existem na literatura, como nos trabalhos de Hiroyasu et al. (2000), Onbasoglu e Özdamar (2001) e Tiwary et al. (2006). Estas integrações, no entanto, normalmente aplicam o critério de aceitação e movimentos do *Simulated Annealing* junto com a natureza populacional dos algoritmos genéticos. Eles mantêm diversos indivíduos executando SA e, em algum ponto, permitem a eles realizar *crossover* entre si.

O algoritmo proposto neste trabalho une *Simulated Annealing* com o operador de *crossover* utilizando apenas um indivíduo por iteração, ou seja, de forma não-populacional. Os *crossovers* são realizados entre o melhor indivíduo e a memória de soluções prévias. Estas soluções antigas são combinadas com a melhor quando o algoritmo encontra um mínimo local no qual

Figura 3.1: Exemplo de AOF com 3 objetivos baseada no *testbench* Rastrigin.

Figura 3.2: Exemplo de AOF com 3 objetivos baseada no *testbench* Michalewicz.

Figura 3.3: Representação de indivíduos salvos na memória para efetuar *crossover*.

os movimentos de subida (aceitação de soluções piores em SA) não são capazes de auxiliar à fuga do mínimo local. Esta operação é possível diante da existência de informação de múltiplos objetivos, apesar do SA operar com base em uma função objetivo agregativa (AOF). Estas informações são utilizadas para salvar soluções que são melhores que todas as outras em algum dos objetivos durante a otimização.

Quando é utilizado uma função agregativa de objetivos, a existência de um mínimo local pode ser devido a natureza do circuito sendo otimizado ou decorrente da forma como a função agregativa foi construída. Da forma como a função é usualmente gerada, as medidas que estão longe de atingir as especificações afetam a função custo de forma mais significativa. Assim, quando o otimizador não consegue melhorar estas medidas sem prejudicar as demais, esta região torna-se um mínimo local. Mesmo quando a função custo é criada sem que os objetivos longe de serem alcançados tenham maior impacto na AOF, as informações de outros pontos podem auxiliar a retirada do mínimo local devido àquele mínimo local possivelmente não existir na otimização multiobjetivo.

Utilizando o operador de *crossover*, comumente utilizada em algoritmos genéticos, juntamente com um ajuste dos pesos da função agregativa, têm-se uma nova estratégia para auxiliar a busca a escapar de mínimos locais e explorar regiões promissoras do espaço de projeto (WEBER; NOIJE, 2012c). A Figura 3.3 mostra como um conjunto de soluções pode ser representado na memória do algoritmo para que sejam utilizadas para o operador de *crossover* posteriormente. Sob a perspectiva deste operador, cada solução é um indivíduo com um número pré-determinado de genes no cromossomo. Cada gene representa uma característica do indivíduo. Em nosso caso, os genes são variáveis reais e representam dimensões físicas de transistores, valores de componentes passivos ou parâmetros elétricos do circuito. O lugar de cada variável é comum a todos os cromossomos. Dessa forma, duas soluções diferentes possuem valores distintos em seus genes, mas estes possuem ordem idêntica dentro do cromossomo para descrever as características do circuito.

Figura 3.4: Ilustração do processo de fuga de mínimos locais através de crossovers.

Figura 3.5: Representação de indivíduos salvos na memória para efetuar *crossover*.

Em algoritmos genéticos, os *crossovers* são realizados entre as diversas soluções que são avaliadas dentro de uma mesma geração da população. Apesar de existirem diversas formas de selecionar quais soluções irão ser combinadas com as outras, a lógica principal é que soluções melhores tenham mais probabilidade de gerar filhos (novas soluções) do que as soluções piores. Para realizar *crossover* utilizando apenas um indivíduo no algoritmo proposto, cada objetivo tem um espaço de memória reservado para salvar a melhor solução encontrada para aquele objetivo até o presente momento. A cada iteração, a solução presente é comparada com as salvas nestes espaços de memória para que este conjunto seja constantemente atualizado com o melhor em cada objetivo.

Quando um mínimo local é detectado e o SA não consegue sair dele, é realizado um *crossover* entre a melhor solução atual e uma das soluções que estão nos espaços de memória, que são as chamadas soluções âncoras. As especificações que estão mais distantes de serem atingidas pela solução atual fornecem maior probabilidade para que suas respectivas soluções âncora sejam selecionadas para realizar o *crossover*. Este procedimento está ilustrado na Figura 3.4.

A Figura 3.5 mostra um exemplo de geração de solução a partir de *crossover*. Este processo produz uma solução filha das duas soluções pais, que, neste caso, são a melhor solução atual e a solução âncora selecionada.

O operador de recombinação utilizado neste trabalho baseia-se em probabilidade condicio-

nal e pode ser visto na eq. (3.2):

$$\begin{aligned} P(g_i = \text{atual} | g_{i-1} = \text{atual}) &= 0,7 \\ P(g_i = \text{âncora} | g_{i-1} = \text{âncora}) &= 0,5 \end{aligned} \quad (3.2)$$

onde g é o gene, i é o índice do gene, *atual* é um gene da solução melhor solução atual e *âncora* é um gene da solução âncora.

A detecção de mínimos locais é baseada no número de falhas consecutivas referentes ao critério de aceitação. Se o algoritmo SA não é capaz de aceitar uma solução dentro de um determinado número de tentativas, presume-se que um mínimo local foi alcançado. Como podem existir variáveis contínuas no problema, as melhorias podem ser, por vezes, muito pequenas. Isto faz com que elas sejam aprovadas pelo critério de aceitação sem, no entanto, representar uma melhoria significativa no circuito. Portanto, é importante reiniciar o contador de falhas (configurar para zero) apenas quando uma melhoria superior a 1% da função custo é alcançada.

A seleção de qual das soluções anteriores será combinada com a melhor atual baseia-se em quão longe cada um dos objetivos está de ser alcançado pela melhor solução. As piores medidas atuais garantem uma chance mais alta das respectivas soluções âncora de serem selecionadas para *crossover* com a melhor solução. A escolha é realizada através de uma seleção do tipo roleta entre as soluções âncoras. Neste tipo de seleção, uma escolha é feita aleatoriamente com base nas probabilidades de cada solução, como pode ser visto no exemplo da Figura 3.6. Assim, a chance de uma solução âncora ser selecionada se baseia em quão distante o circuito atual está de atingir o objetivo da solução âncora. Em algoritmos genéticos, existem outras formas de seleção possíveis, como seleção por torneio, classificação ou truncamento. No entanto, devido ao número baixo de soluções possíveis para serem selecionadas (igual ao número de objetivos), o método de roleta é mais adequado ao SA com *crossovers*.

3.1.4 Seleção de Soluções Âncoras

No presente trabalho, utiliza-se a definição de soluções âncoras como sendo os pontos extremos de uma aproximação de fronteira de Pareto. Eles representam a melhor solução já obtida para cada especificação dada. Como, no algoritmo proposto, as soluções âncoras são responsáveis por retirar a presente solução de regiões de mínimos locais, escolher quais as soluções devem ser salvas como soluções âncoras é um passo importante.

Selecionar diretamente os pontos extremos como pontos de ancoragem para que sejam mantidos na memória não é interessante para a otimização. A razão é que uma solução pode ser extremamente boa em uma especificação como, por exemplo, consumo de potência, mas ser ruim

Figura 3.6: Exemplo de seleção do tipo “roleta” para uma síntese com cinco objetivos.

em todas as demais. Um *crossover* entre a solução presente e uma solução âncora deste tipo resultaria, provavelmente, em uma solução não funcional. Uma abordagem mais apropriada é salvar as soluções que possuem um resultado da AOF não exageradamente pior que o resultado da AOF da melhor solução. Este *tradeoff* pode ser realizado através de uma soma ponderada do custo para a especificação e do custo total.

3.1.5 Ajuste Automático de Pesos

Alterar os pesos após o *crossover* dificulta ao *Simulated Annealing* retornar ao mesmo mínimo local atingido anteriormente. Isto é realizado através do aumento do peso da especificação sorteada na seleção do tipo roleta. É importante manter a mesma métrica quando se compara soluções memorizadas em momentos em que os pesos são diferentes. Portanto, as modificações de peso são adicionadas apenas como penalizações, mantendo os primeiros termos inalterados para permitir comparações entre soluções.

Cada vez que um mínimo local é encontrado, um vetor sp recebe um novo valor com o índice de especificação sorteado na roleta. A equação (3.3) mostra como o ajuste de peso afeta o custo após um ou mais mínimos locais serem encontrados:

$$C_P(x) = C(x) + k \sum_{i=\max(1, ms-n)}^{ms} w_{sp_i} f_{sp_i} \quad (3.3)$$

onde $C_P(x)$ é o custo total com penalizações, $C(x)$ é o custo total original, k é um fator utilizado para aumentar o peso dos ajustes em relação ao custo total, ms é o índice do último valor do vetor sp , n é o máximo número de mínimos locais que será utilizado para o ajuste,

w_{sp_i} é o peso de cada especificação salva em sp_i e f_{sp_i} é o valor da função custo para cada especificação salva no sp_i .

3.1.6 Fluxograma do Algoritmo

A Figura 3.7 mostra o fluxograma do algoritmo. O laço principal do algoritmo ocorre até que o critério de parada seja alcançado, que pode ser definido com base no número máximo de iterações, tempo limite ou no número de iterações sem melhoras na solução.

A função de geração tem dois modos de operação: o modo convencional e o de *crossover*. O modo convencional gera uma solução a partir da variação da solução atual com passo adaptativo. Quando o número de indivíduos que não é aceito é superior a um número pré-determinado, a função de geração baseada em *crossovers* é realizada.

Após a geração da nova solução, é analisado o critério de aceitação para decidir se a solução gerada é aceita ou não. Uma variável aleatória τ entre 0 e 1 é gerada e comparada com o valor de duas funções, P_{AOF1} e P_{AOF2} :

$$P_{AOF1}(t_k, N, S_k) = e^{-\frac{C(S_k) - C(N)}{|C(S_k)| \cdot t_k}} \quad (3.4)$$

$$P_{AOF2}(t_k, N, S_k) = e^{-\frac{C_P(S_k) - C_P(N)}{|C_P(S_k)| \cdot t_k}} \quad (3.5)$$

onde k é o índice da solução, N é a solução candidata, S_k é a solução atual, $C(S)$ é a função custo que avalia a solução S e $C_P(S)$ é uma função custo com penalizações apresentada na equação 3.3. Ambos critérios de aceitação são baseados no critério de Metropolis, sendo que o primeiro (eq. 3.4) considera apenas o valor da função AOF original do problema e o segundo (eq. 3.5) utiliza a AOF original com adição de penalizações para objetivos que já foram selecionados previamente para *crossover*. Essas penalizações são equivalentes ao acréscimo de peso na AOF aos objetivos as quais elas se referem.

Assim, a solução que será adotada para próxima iteração será:

$$S_{k+1} = \begin{cases} N, & \text{se } C(N) < C(S_k) \text{ ou } C_P(N) < C_P(S_k) \\ & \text{ou } \tau \leq P_{AOF1}(t_k, N, S_k) \text{ ou } \tau \leq P_{AOF2}(t_k, N, S_k) \\ S_k, & \text{caso contrário} \end{cases} \quad (3.6)$$

O cronograma de temperatura é realizado após um número arbitrário e fixo de n iterações. A cada n iterações é analisado quantas modificações de variáveis foram aceitas através do critério

Figura 3.7: Fluxo do algoritmo proposto (*Simulated Annealing* com *crossovers*).

de aceitação. A equação 3.7 descreve o processo de resfriamento:

$$t_i \text{ é } \begin{cases} \alpha \cdot t_{i-1}, & \text{se } 0,3 < P_A < 0,6 \quad (\text{resfriamento normal}) \\ \beta \cdot t_{i-1}, & \text{se } P_A < 0,3 \quad (\text{aquecimento}) \\ \gamma \cdot t_{i-1}, & \text{se } P_A > 0,6 \quad (\text{resfriamento acelerado}) \end{cases} \quad (3.7)$$

onde α é o fator de resfriamento normal (estabelecido entre 0,9 e 0,99), β é o fator de

Figura 3.8: Variações de temperatura para o algoritmo proposto.

aquecimento (entre 1,1 e 1,5), γ é o fator de resfriamento acelerado (entre 0,8 e 0,9) e P_A é a porcentagem de variações aceitas no critério de aceitação para as últimas n iterações.

Esta alteração é similar a equação 2.27, porém aplicada a temperatura em vez de à variação do passo. A Figura 3.8 mostra um exemplo de decaimento da temperatura para a situação de resfriamento normal e para casos em que ocorre aquecimento ou resfriamento acelerado. A porcentagem da temperatura inicial é mostrada em função do conjunto de iterações k (no qual k varia após n iterações). Para o decaimento normal, apenas α é utilizado. O caso ilustrado na figura para decaimento acelerado mostra uma situação na qual ocorrem diversos momentos em que P_A é superior a 0,6. Já o caso de resgate de temperatura é a consequência de diversas ocorrências de P_A inferior a 0,3. Essas variações mais abruptas de temperatura que o *annealing* tradicional são úteis para que o algoritmo adapte-se a escala e a topologia do espaço de critério do problema sendo resolvido.

De forma independente do resultado do critério de aceitação, as soluções geradas são analisadas em relação as soluções âncoras já existentes. A Figura 3.10 mostra um exemplo de situação na qual o *crossover* e ajuste de pesos auxilia o otimizador a sair de uma região de mínimo local em um problema de 2 objetivos. Pode-se observar que este mecanismo propicia uma movimentação para um ponto possivelmente mais interessante do espaço de projetos, enquanto simultaneamente diminui o tamanho do mínimo local devido a alteração dos pesos da função. A Figura 3.9 mostra o pseudocódigo do algoritmo proposto.

3.2 Descrição do Algoritmo Proposto para Problemas Multi-objetivo

Quando o objetivo do projetista ou ferramenta é a obtenção da fronteira de Pareto, o algoritmo descrito anteriormente precisa de modificações para que possa realizar a exploração dos *tradeoffs* existentes no projeto. A solução proposta nesta seção é hibridizar o algoritmo

```

t ← temperatura inicial;
S ← solução candidata inicial;
Inicializar âncoras;
Avaliar solução inicial;
para (i=1 até tam_amostra_inicial) faça                                ▶ Faz uma amostragem inicial
    N ← Gerar_aleatoria;                                              ▶ Gera solução aleatória
    Se (C(N) < C(melhor)) então melhor ← N; fim se                ▶ C() é a função custo
    Se (N melhor que algum âncora) então atualiza âncora; fim se    ▶ Salva âncoras
fim para
S ← melhor
enquanto critério de parada não satisfeito faça
    fazercrossover ← 0;
    se (contador_falhas > max_para_local) então
        fazercrossover ← 1;
        contador_falhas ← 0;
    caso contrário se (tentativas > max_para_cron) então
        T ← cronograma(T);
    fim se
    se (fazercrossover == 0) então                                    ▶ Operação normal
        Escolher variáveis a modificar;
        N ← varia(cópia(S));                                         ▶ Função de geração por passo adaptativo
    caso contrário                                                    ▶ Caso seja para realizar crossover
        S ← melhor;
        i ← roleta;                                                    ▶ Selecciona âncora
        N ← crossover(âncora(i),S);                                   ▶ Crossover entre âncora e melhor solução
        atualizar pesos da AOF2;                                         ▶ Pesos que são usados na função CP()
    fim se
    τ ← valor aleatório entre 0 e 1
    se ( (C(N) < C(S)) ou (CP(N) < CP(S)) ou ...
    ... ( τ < PAOF1 ) ou ( τ < PAOF2 ) ) então                    ▶ Critério de aceitação
        S ← N;
    caso contrário
        contador_falhas ← contador_falhas + 1;
    fim se
    se (  $\frac{C(S)-C(N)}{|C(S)|} > 0,05$  ) então                                ▶ Zera contador de falhas
        contador_falhas ← 0;
    fim se
    se (C(N) < C(melhor)) então                                    ▶ Salvar caso seja a melhor resposta
        melhor ← N;
    fim se
    Se (N melhor que algum âncora) então atualiza âncora; fim se
    Atualizar variância das modificações;
fim enquanto

```

Figura 3.9: Pseudocódigo da versão modificada do SA.

Figura 3.10: Exemplo da operação de *crossover* e ajuste de peso para um problema com dois objetivos.

desenvolvido juntamente com um algoritmo populacional (WEBER; NOIJE, 2011a) (WEBER; NOIJE, 2012b). O algoritmo MOPSO (*Multi-Objective Particle Swarm Optimization*) tem características interessantes que favorecem essa hibridização, como será visto a seguir.

Em problemas com mais de três objetivos, algoritmos baseados diretamente em dominação têm dificuldade em obter boas respostas. Com o aumento do número de objetivos, o número de soluções não dominadas também aumenta e, por consequência, a pressão da seleção baseada nesse critério perde eficiência (HUGHES, 2005). O algoritmo proposto para o caso mono-objetivo pode utilizar informação multiobjetivo sem enfrentar estes problemas. Dessa forma, ele pode ser utilizado em fases iniciais do processo de busca da fronteira de Pareto. Mais especificamente, ele pode ser usado em duas fases distintas, em um processo que tem três fases:

Fase 1: busca o mínimo global dado um vetor de pesos escolhido pelo projetista (ou um vetor unitário para tornar os pesos iguais entre os objetivos);

Fase 2: uma série de otimizações secundárias utilizando a resposta da primeira fase como semente é realizada. Cada uma tem foco em um objetivo em especial (modificando o

peso dos demais objetivos na AOF para uma fração de seu valor original). Durante cada uma destas otimizações, é construída a experiência que será transferida para uma partícula do MOPSO posteriormente. Caso seja desejado um número maior de partículas do que de objetivos, um vetor de pesos para cada uma destas novas partículas é gerado aleatoriamente com distribuição uniforme;

Fase 3: por fim, o MOPSO é executado utilizando todas as partículas já inicializadas pelas fases anteriores. Este algoritmo MOPSO é desenvolvido com foco em distribuir as soluções pela fronteira de Pareto.

A Figura 3.11 ilustra como estas diferentes fases se interconectam, bem como mostra o preenchimento da aproximação da fronteira de Pareto a medida que as fases evoluem. Durante a execução de todas as fases, as soluções não dominadas são armazenadas em uma lista. Durante as fases 1 e 2, isto é feito de forma paralela a execução do algoritmo, visto que estas fases não utilizam critério de dominação como forma de guiar a busca. A fase 1 possui apenas uma lista, que será transferida para a partícula $p_i = 1$ da fase 3. A fase 2 possui uma lista para cada uma das otimizações secundárias, de forma que estas listas serão transferidas para as partículas $p_i = 2 \dots k + 1$ na fase 3, onde k é o número de objetivos. Assim, durante a fase 3, as listas obtidas nas fases anteriores são utilizadas como experiências locais de cada partícula e continuam sendo atualizadas. No final, estas listas são utilizadas para compor a aproximação da fronteira de Pareto.

Durante a coleta e atualização das soluções não dominadas, as listas podem se tornar demasiadamente grandes. Isso aumenta o uso de memória e o tempo para realização de operações de checagem e adição de elementos. A fim de evitar este problema, foi utilizado o conceito de ϵ -dominação (MOSTAGHIM; TEICH, 2003). Nele, o conceito tradicional de dominação é modificado para permitir que apenas soluções não dominadas com alguma margem de melhoria possam ingressar na lista. Assim, dados dois vetores $\vec{a}, \vec{b} \in \mathbb{R}^k$, dizemos que $\vec{a}$ ϵ -domina $\vec{b}$ se $\vec{a}_i/(1 + \epsilon) \leq \vec{b}_i$ e $\vec{a}_i \neq \vec{b}_i \forall i = 1, \dots, k$, onde ϵ é um valor pré-determinado.

O algoritmo MOPSO desenvolvido utiliza topologia de vizinhança totalmente conectada. Dessa forma, todas as partículas tem acesso as soluções não dominadas de toda a população. A função principal do algoritmo é distribuir as soluções pela fronteira de Pareto. Por este motivo, a seleção dos líderes locais (dentro da experiência de uma mesma partícula) e globais (dentro da experiência de toda a população) é baseada na densidade de soluções nas listas. A solução que está mais distante das demais em uma lista é selecionada como líder.

Figura 3.11: Diferentes fases de otimização para o algoritmo híbrido multiobjetivo e ilustração das soluções na fronteira de Pareto após cada fase.

3.3 Testes

A fim de testar o desempenho dos algoritmos propostos em relação a outros encontrados na literatura, foram realizados diversos testes. Os testes contidos neste capítulo utilizam exclusivamente problemas matemáticos, não envolvendo circuitos integrados analógicos, que serão tratados apenas no próximo capítulo. Estes testes matemáticos são úteis para analisar características dos algoritmos que podem ser difíceis de observar utilizando circuitos diretamente, como, por exemplo, comparar o desempenho de algoritmos ao variar-se arbitrariamente o número de variáveis e objetivos do problema. Ainda, os problemas matemáticos explorados neste capítulo são amplamente utilizados na literatura de otimização, fornecendo assim um ambiente comum de análise dos algoritmos para pesquisadores de áreas além da microeletrônica.

3.3.1 Descrição do Método de Teste

O padrão seguido para coleta de dados e análise dos testes será descrito nesta subseção. Todos os testes dessa seção foram realizados utilizando um computador com processador Intel(R) Core(TM) i7-2600 CPU 3,40GHz, com 16 GB de memória RAM.

Para o caso mono-objetivo, o critério de parada do primeiro conjunto de testes foi estabelecido em 2.000 iterações. Para averiguar o efeito do aumento do número de iterações nos resultados, foi realizado também um conjunto de testes mono-objetivo com critério de parada estabelecido em 10.000 iterações. Para o caso multiobjetivo, também foram utilizados dois critério de parada, um em 5.000 iterações e outro em 10.000 iterações para investigar o efeito do número de iterações em problemas com muitos objetivos.

Como os algoritmos estudados utilizam variáveis aleatórias durante sua operação, diversos procedimentos de otimização são necessários para realizar uma comparação válida entre eles. Para os testes matemáticos mono-objetivo, cada combinação de algoritmo e *testbench* foi otimizada 50 vezes. Cada otimização recebe como solução inicial um vetor gerado aleatoriamente com distribuição uniforme entre o valor mínimo permitido para cada variável e o valor máximo. Algoritmos populacionais recebem como soluções iniciais vetores deste mesmo tipo em número igual ao tamanho de suas populações. Seguindo a abordagem de Rios e Sahinidis (2013), as comparações são realizadas utilizando as configurações padrões de cada otimizador.

Para efeito de construção de gráficos comparativos entre os algoritmos, foi adotada a mediana do algoritmo que desempenhou-se melhor em cada problema como referência. Foram considerados dois critérios distintos para avaliar se um algoritmo solucionou um problema: respostas (medianas) até 1% próximas ou com diferença de até 0,01 da melhor mediana obtida por todos algoritmos para cada caso; respostas até 5% próximas ou com diferença de até 0,05 da melhor mediana. O uso de dois critérios diferentes permite avaliar o impacto destes na ordenação dos melhores algoritmos.

Para o caso multiobjetivo, foi adotado o hipervolume H_i (ZITZLER; THIELE, 1998) como medida de referência de aproximação da fronteira de Pareto. Esta medida indica o volume do espaço de critério dominado pela aproximação da fronteira de Pareto. Através do hipervolume é possível avaliar tanto a qualidade da aproximação da fronteira de Pareto dos menores melhores valores possíveis, quanto a distribuição das soluções encontradas dentro do espaço de critério (AZEVEDO; ARAUJO, 2011). Ainda, o hipervolume é o único indicador unário conhecido que consegue capturar a característica de uma solução ser fracamente dominada em relação a outra, uma vez que avalia o volume da área dominada (ZITZLER et al., 2003).

Para manter um padrão entre os testes multiobjetivos e mono-objetivos (no qual o interesse

era a minimização do indicador), utilizou-se para comparação o cálculo $I_i = 1 - H_i$. Dessa forma, os resultados são valores no intervalo $[0, 1]$, onde 0 é a melhor resposta possível, significando que a fronteira de Pareto aproximada domina todo o hipercubo, e 1 representa um desempenho ruim, significando que a fronteira de Pareto aproximada não domina nenhuma solução do hipercubo.

Esta medida é sensível ao número de pontos amostrados do hipercubo, bem como dos limites inferiores e superiores do espaço de critério. Para cálculo do hipervolume dos testes realizados, foram amostrados 10.000 pontos. Os limites inferiores do hipercubo foram definidos como iguais às menores respostas obtidas em cada especificação entre todos os algoritmos e iterações, e os limites superiores como iguais às maiores respostas obtidas. No caso multiobjetivo foram realizadas 10 otimizações para cada síntese.

3.3.2 Algoritmos Comparados para o Caso Mono-Objetivo

Foram selecionados para comparação algoritmos do tipo livre de derivada que possuem versão disponível para uso no software MATLAB e que não são comerciais. A lista com os algoritmos utilizados para comparação está abaixo:

- **Algoritmo Proposto:** a técnica que utiliza uma variação de *Simulated Annealing* com *crossovers* e ajuste automático de pesos proposta neste trabalho;
- **CMA-ES (HANSEN; OSTERMEIER, 1996):** esta é uma técnica de estratégia evolutiva. Ela utiliza uma adaptação da matriz de covariância (CMA, do inglês, *Covariance Matrix Adaptation*) a fim de adaptar o operador de mutação ao problema. Foram utilizadas as opções *default* da implementação;
- **MCS (NEUMAIER, 2000):** é uma implementação da técnica proposta por Huyer e Neumaier (1999). O algoritmo realiza otimização global baseada em busca coordenada multinível e busca local baseada em programação quadrática sequencial. Foram utilizadas as opções *default* da implementação, sendo o número máximo de níveis $smax = 5 \cdot n + 10$, onde n é o número de variáveis do problema;
- **PSWARM (VAZ; VICENTE, 2007):** é um otimizador global que combina as técnicas de *pattern search* e *particle swarm optimization*. Foi utilizada a implementação realizada por Vaz e Vicente (2009). As opções *default* da implementação foram utilizadas, sendo o tamanho da população *popsiz* o valor máximo entre o número de variáveis n e 10.
- **ASA (INGBER; INGBER, 1989):** o *Adaptive Simulated Annealing* é um algoritmo de otimização global que busca explorar o espaço de projeto de forma mais eficiente do que

os métodos anteriores de *Simulated Annealing*. Este algoritmo foi proposto por Ingber, sendo originalmente chamado de *very fast simulated reannealing* (VFSR). Sua adaptabilidade lhe proporcionou ser a variação mais popular do *simulated annealing* atualmente. Neste trabalho foi utilizada o *gateway* do ASA para MATLAB desenvolvido por Sakata (1999);

- **GA POP50 (HOLLAND, 1975):** algoritmo genético com uma população de 50 indivíduos, seleção por torneio, 1 solução de elite e probabilidade de *crossover* de 0,8. A implementação utilizada é a existente na *toolbox* de otimização global do software MATLAB (MATHWORKS, 2013);
- **GA POP100:** A mesma implementação de algoritmos genéticos do caso anterior, porém com uma população de 100 indivíduos;
- **NSGA-II (DEB et al., 2002):** algoritmo genético multiobjetivo com população de 100 indivíduos. A implementação utilizada foi desenvolvida por Lin (2011).

Apesar do NSGA-II ser focado em problemas multiobjetivo, ele é utilizado aqui em algoritmos mono-objetivo através do uso do conceito de multiobjetivação descrito na seção 3.1.2. Dessa forma, pode-se comparar o desempenho do algoritmo proposto, que usa multiobjetivação, com um algoritmo evolutivo multiobjetivo também usando este conceito.

3.3.3 Testbenches para o Caso Mono-Objetivo

A Tabela 3.1 mostra os *testbenches* que foram utilizados para comparar os algoritmos no caso mono-objetivo. Estas funções foram selecionadas por possibilitarem um número arbitrário de variáveis e por serem *testbenches* frequentemente utilizados para comparar algoritmos de otimização (MOLGA; SMUTNICKI, 2005) (LUKE, 2009).

Na formulação original dos problemas, existe apenas a função composta $f(x)$. Para possibilitar a utilização do conceito de multiobjetivação, as funções foram divididas em sub funções $f_p(x)$. Assim, os algoritmos que usam informação multiobjetivo podem utilizar as sub funções, apesar de que todos os algoritmos sejam avaliados apenas em sua capacidade de obter o menor valor possível da função original.

Cada *testbench* da Tabela 3.1 é testado para $n = [2, 3, 5, 10, 20, 30, 40, 50, 60]$.

Tabela 3.1: *Testbenches* matemáticos para comparação entre algoritmos mono-objetivo, onde $f(x)$ é a função principal, n é o número de variáveis, $i = 1, \dots, n$ é o índice das variáveis e p é o índice das sub funções $f_p(x)$.

Nome	Função	Intervalo
Rosenbrock	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$ $f_p(x) = 100(x_{p+1} - x_p^2)^2 + (1 - x_p)^2$ com $p = 1, \dots, n-1$;	$-5, 12 \leq x_i \leq 5, 12$
Rastrigin	$f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i))$ $f_p(x) = x_p^2 - 10\cos(2\pi x_p)$ com $p = 1, \dots, n$;	$-5, 12 \leq x_i \leq 5, 12$
Michalewics	$f(x) = -\sum_{i=1}^n \sin(x_i) \cdot (\sin(\frac{ix_i^2}{\pi}))^{2m}$ $f_p(x) = -(\sin(x_p) \cdot (\sin(\frac{ix_p^2}{\pi}))^{2m})$ com $m = 10$ e $p = 1, \dots, n$;	$0 \leq x_i \leq \pi$
Ackley	$f(x) = -a \cdot e^{-b \sqrt{\frac{\sum_{i=1}^n x_i^2}{n}}} - e^{\frac{\sum_{i=1}^n \cos(c \cdot x_i)}{n}} + a + e$ $f_p(x) = x_p^2$ com $a = 20; b = 0, 2; c = 2\pi$; $p = 1, \dots, n$;	$-30 \leq x_i \leq 30$
Levy	$f(x) = \sin^2(\pi\omega) + \sum_{i=1}^{n-1} (w_i - 1)[1 + 10\sin^2(\pi\omega_i + 1) + (\omega_n - 1)^2[1 + \sin^2(2\pi\omega_n)]]$ $f_p(x) = (w_p - 1)[1 + 10\sin^2(\pi\omega_p + 1)]$ com $w_i = 1 + \frac{x_i - 1}{4}$ $p = 1, \dots, n-1$;	$-10 \leq x_i \leq 10$
Schwefel	$f(x) = 418, 9829n + \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$ $f_p(x) = -x_i \sin(\sqrt{ x_i })$ com $p = 1, \dots, n$;	$-500 \leq x_i \leq 500$
Griewank	$f(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}})$ $f_p(x) = \cos(\frac{x_p}{\sqrt{p}}) $ com $p = 1, \dots, n$;	$-600 \leq x_i \leq 600$

3.3.4 Resultados para Problemas Matemáticos Mono-Objetivo

3.3.4.1 Número de Iterações Máximo Igual a 2.000

O primeiro resultado a ser mostrado é a mediana da resposta final dos algoritmos por *test-bench* e por número de variáveis. A Tabela 3.2 mostra estes resultados para os testes com até 2.000 iterações.

Tabela 3.2: Mediana de 50 otimizações dos algoritmos comparados para cada caso dos *testbenches* matemáticos mono-objetivo de 2.000 iterações.

Testbench	Alg. Proposto	ASA	CMA-ES	GA POP100	GA POP50	MCS	NSGA	PSWARM
Número de Variáveis = 2								
rastrigin	1,49E-04	4,88E-01	3,53E-13	2,43E-01	3,88E+00	2,33E-04	4,02E+00	6,54E-14
griewank	2,47E-03	6,65E-01	5,42E-02	1,43E+00	1,43E+00	2,49E-03	1,00E-01	2,22E-02
Michalewicz	-1,80E+00	-1,74E+00	-1,80E+00	-2,14E-05	-1,91E-05	-1,80E+00	-1,80E+00	-1,80E+00
schwefel	-5,43E-07	-5,45E-07	3,57E+02	8,36E+02	8,36E+02	5,77E-04	1,33E+00	1,18E+02
levy	1,46E-14	1,71E-03	2,73E-32	1,90E-03	2,41E-03	6,73E-12	4,66E-04	1,50E-32
rosenbrock	2,04E-02	3,39E-01	1,80E-29	1,17E-02	2,17E-02	2,81E-04	8,35E-03	4,64E-04
ackley	8,88E-16	7,87E-02	4,44E-15	3,66E-01	6,75E-01	1,47E-02	5,40E-01	8,88E-16
Número de Variáveis = 3								
rastrigin	7,22E-04	1,30E+00	8,64E-13	1,68E+01	2,76E+01	2,47E-01	1,56E+01	1,91E-13
griewank	4,93E-03	2,03E+00	7,27E-02	1,38E+00	1,38E+00	2,47E-03	2,26E-01	7,89E-02
Michalewicz	-2,76E+00	-2,13E+00	-2,60E+00	-1,29E-02	-1,29E-02	-2,76E+00	-2,76E+00	-2,76E+00
schwefel	-7,97E-07	-8,17E-07	4,77E+02	1,25E+03	1,25E+03	1,20E+02	4,64E+00	1,78E+02
levy	1,27E-13	1,56E-03	4,42E-31	1,17E-02	1,36E-02	1,97E-09	2,01E-03	1,50E-32
rosenbrock	2,72E-01	9,16E-01	8,22E-28	1,49E-01	2,64E-01	1,16E-03	4,78E-01	2,48E-02
ackley	8,88E-16	5,03E-02	4,44E-15	9,08E-01	1,14E+00	3,29E-01	6,90E-01	4,44E-15
Número de Variáveis = 5								
rastrigin	4,54E-03	4,22E+00	6,68E-09	1,94E+02	2,06E+02	6,86E-01	7,49E+01	1,47E-07
griewank	9,86E-03	1,10E+01	3,45E-02	1,25E+00	1,23E+00	1,23E-02	6,22E-01	2,21E-01
Michalewicz	-4,54E+00	-2,76E+00	-3,70E+00	-8,90E-01	-9,43E-01	-4,35E+00	-4,56E+00	-4,41E+00
schwefel	7,04E-02	1,12E-03	8,91E+02	2,09E+03	2,09E+03	4,74E+02	1,40E+02	2,37E+02
levy	2,50E-12	1,01E-02	2,46E-23	5,50E-02	4,70E-02	8,17E-05	1,85E-02	3,29E-19
rosenbrock	1,61E+00	1,11E+00	2,23E-06	2,82E+00	3,79E+00	3,31E-01	3,53E+00	2,54E-01
ackley	8,88E-16	1,65E-01	2,11E-11	1,55E+00	1,34E+00	5,72E-01	1,81E+00	2,64E-09
Número de Variáveis = 10								
rastrigin	3,30E-01	1,60E+02	3,95E-02	1,65E+03	1,56E+03	2,74E+00	6,38E+02	1,53E-01
griewank	1,08E-01	6,21E+01	7,39E-03	1,19E+00	1,14E+00	1,12E-01	2,00E+00	4,19E-02
Michalewicz	-8,61E+00	-4,10E+00	-6,74E+00	-3,31E+00	-3,30E+00	-8,11E+00	-7,63E+00	-8,40E+00
schwefel	2,38E+02	2,41E+02	1,96E+03	4,18E+03	4,18E+03	1,18E+03	1,02E+03	5,92E+02
levy	1,19E-05	1,64E-01	1,23E-09	1,73E-01	1,74E-01	8,17E-05	4,33E-01	7,28E-08
rosenbrock	7,34E+00	1,28E+01	5,23E+00	3,30E+01	3,46E+01	8,48E+00	5,81E+01	6,07E+00
ackley	1,15E-14	2,11E+00	8,84E-05	2,11E+00	1,94E+00	1,15E+00	4,24E+00	9,14E-04
Número de Variáveis = 20								
rastrigin	5,59E+02	3,66E+03	2,02E+03	1,04E+04	8,63E+03	1,10E+01	6,56E+03	8,67E+02
griewank	3,50E-01	2,41E+02	6,77E-02	1,13E+00	1,14E+00	1,23E-02	2,98E+01	1,34E+00
Michalewicz	-1,49E+01	-6,06E+00	-4,86E+00	-5,83E+00	-6,66E+00	-1,39E+01	-1,11E+01	-1,54E+01
schwefel	1,14E+03	1,75E+03	4,12E+03	8,37E+03	8,37E+03	2,37E+03	3,57E+03	1,42E+03
levy	4,61E-01	3,01E+00	8,97E-02	5,76E-01	4,64E-01	8,17E-05	1,16E+01	2,40E-01
rosenbrock	1,83E+01	2,56E+02	1,85E+01	1,06E+02	1,10E+02	1,84E+01	2,34E+03	6,15E+01
ackley	7,02E-06	5,81E+00	1,13E+01	2,74E+00	2,43E+00	9,95E-01	1,11E+01	2,86E+00
Número de Variáveis = 30								
rastrigin	8,19E+03	1,51E+04	9,37E+03	2,76E+04	2,27E+04	2,47E+01	2,29E+04	1,09E+04
griewank	8,81E-01	4,65E+02	1,02E+00	1,12E+00	1,09E+00	6,87E-02	1,40E+02	1,07E+01
Michalewicz	-1,91E+01	-7,85E+00	-5,64E+00	-7,74E+00	-9,02E+00	-1,87E+01	-1,38E+01	-1,64E+01
schwefel	2,73E+03	4,35E+03	7,34E+03	1,26E+04	1,26E+04	3,56E+03	6,51E+03	2,79E+03
levy	1,63E+00	1,72E+01	1,02E+00	9,75E-01	8,29E-01	8,17E-05	4,09E+01	4,23E+00
rosenbrock	2,88E+01	1,96E+03	3,35E+01	1,78E+02	1,81E+02	2,83E+01	2,10E+04	5,20E+02
ackley	2,61E-03	9,51E+00	1,40E+01	3,01E+00	2,70E+00	1,80E+00	1,52E+01	6,78E+00
Número de Variáveis = 40								
rastrigin	2,62E+04	3,67E+04	3,44E+04	5,56E+04	4,74E+04	4,39E+01	5,00E+04	3,41E+04
griewank	9,83E-01	6,84E+02	1,49E+00	1,06E+00	1,07E+00	9,51E-02	2,90E+02	3,16E+01
Michalewicz	-2,24E+01	-9,55E+00	-6,70E+00	-9,60E+00	-1,16E+01	-2,34E+01	-1,62E+01	-1,57E+01
schwefel	5,34E+03	7,30E+03	1,11E+04	1,67E+04	1,67E+04	4,74E+03	9,59E+03	7,43E+03
levy	2,62E+00	4,29E+01	2,74E+00	1,41E+00	1,19E+00	8,17E-05	8,76E+01	1,53E+01
rosenbrock	3,96E+01	8,90E+03	1,03E+02	2,48E+02	2,52E+02	3,82E+01	5,45E+04	1,97E+03
ackley	7,71E-02	1,17E+01	1,58E+01	3,14E+00	2,86E+00	1,32E+00	1,69E+01	9,73E+00
Número de Variáveis = 50								
rastrigin	6,24E+04	6,60E+04	8,91E+04	9,15E+04	7,84E+04	6,86E+01	9,02E+04	6,62E+04
griewank	1,04E+00	9,31E+02	5,04E+00	1,11E+00	1,06E+00	9,52E-02	4,97E+02	8,55E+01
Michalewicz	-2,54E+01	-1,10E+01	-7,32E+00	-1,13E+01	-1,33E+01	-2,90E+01	-1,80E+01	-1,74E+01
schwefel	8,78E+03	1,06E+04	1,45E+04	2,09E+04	2,09E+04	5,93E+03	1,31E+04	1,12E+04
levy	3,84E+00	7,48E+01	8,06E+00	1,92E+00	1,66E+00	8,17E-05	1,49E+02	3,65E+01
rosenbrock	5,20E+01	2,15E+04	4,10E+02	3,19E+02	3,22E+02	4,81E+01	1,17E+05	6,20E+03
ackley	1,12E+00	1,35E+01	1,77E+01	3,21E+00	3,01E+00	6,41E-02	1,78E+01	1,29E+01
Número de Variáveis = 60								
rastrigin	1,17E+05	1,05E+05	1,54E+05	1,39E+05	1,21E+05	9,87E+01	1,41E+05	1,16E+05
griewank	1,29E+00	1,15E+03	3,68E+01	1,07E+00	1,08E+00	1,64E-01	6,83E+02	1,76E+02
Michalewicz	-2,60E+01	-1,22E+01	-8,05E+00	-1,29E+01	-1,52E+01	-3,42E+01	-2,00E+01	-1,85E+01
schwefel	1,20E+04	1,41E+04	1,76E+04	2,51E+04	2,51E+04	7,11E+03	1,66E+04	1,52E+04
levy	5,94E+00	1,28E+02	1,95E+01	2,36E+00	2,14E+00	8,17E-05	2,23E+02	7,06E+01
rosenbrock	8,50E+01	4,20E+04	1,32E+03	3,89E+02	3,94E+02	5,81E+01	1,83E+05	1,63E+04
ackley	3,85E+00	1,44E+01	1,87E+01	3,33E+00	3,10E+00	2,12E-02	1,83E+01	1,42E+01

Pela análise das respostas por *testbench*, é possível ver que não há um algoritmo que seja melhor absoluto em todos os casos. Isto é coerente com o *No Free Lunch Theorem* para problemas de otimização (WOLPERT; MACREADY, 1997). Deste decorre que, para qualquer algoritmo, a melhoria de desempenho para uma classe específica de problemas resulta em um pior desempenho em outra classe. No entanto, alguns algoritmos têm desempenho significativamente melhor que os outros para os casos estudados.

Analisando em detalhe o comportamento dos algoritmos entre os *testbenches*, na maior parte das instâncias da função Griewank, até 5 variáveis o MCS e o algoritmo proposto têm desempenhos similares. Após 10 variáveis, o MCS tem o melhor desempenho. Na função Schwefel, o algoritmo proposto tem o melhor desempenho, seguido pelo algoritmo ASA e PSWARM. Para a função Levy, o CMA-ES e o PSWARM têm as melhores respostas até 20 variáveis. Após 20 variáveis, o algoritmo GA POP50 e GA POP100 passam a ter melhor desempenho que os demais. Para a função Rosenbrock, o CMA-ES tem as melhores respostas até 10 variáveis, sendo que após o algoritmo MCS passa a ter as melhores respostas seguido pelo algoritmo proposto. Para a função Michalewics, o algoritmo NSGA, o CMA-ES, o MCS e PSWARM têm resultados similares até 10 variáveis, sendo que a partir de então apenas o algoritmo proposto e o MCS alternam as melhores respostas. Para a função Rastrigin, os algoritmos CMA-ES e PSWARM tem as melhores respostas, seguidos pelo algoritmo proposto. Após 10 variáveis, o algoritmo MCS passa a ter as melhores respostas. Para a função Ackley, o algoritmo proposto tem as melhores respostas até 40 variáveis. Após, o algoritmo MCS passa a ter melhor desempenho.

Apesar desta análise ter sido realizada observando os valores tais como são apresentados na Tabela 3.2, as estatísticas devem ser obtidas considerando a tolerância de proximidade da melhor resposta. A Figura 3.12 ilustra a porcentagem de problemas resolvidos (respostas até 1% ou 0,01 próximas do melhor resultado obtidos entre os algoritmos) em função do número de variáveis dos problemas. Nesta figura, foram considerados como dados para cálculo apenas as medianas das simulações de cada teste e não cada simulação individualmente.

Dos resultados apresentados na figura, é possível observar que o algoritmo proposto teve resposta melhor ou igual ao melhor dos outros algoritmos para 2, 3, 5 e 20 variáveis. Para 10 variáveis, o CMA-ES teve a melhor resposta e para casos com 30 ou mais variáveis o MCS teve a melhor resposta. Com exceção do MCS, há um decaimento geral no desempenho relativo dos algoritmos com o aumento do número de variáveis.

A Figura 3.13 sumariza os resultados obtidos para problemas mono-objetivo para 2.000 iterações. Para até 60 variáveis (Figura 3.13 a), o melhor desempenho é o do algoritmo MCS, seguido pelo algoritmo proposto, PSWARM, CMA-ES, ASA, NSGA-2, GA-POP50 e GA-POP100. Para até 30 variáveis (Figura 3.13 b), o melhor desempenho é o do algoritmo pro-

Figura 3.12: Porcentagem de problemas mono-objetivo resolvidos por cada algoritmo em função do número de variáveis para 2.000 iterações (considerando a resposta das medianas de cada caso de *testbench* e número de variáveis).

posto, seguido pelo MCS, PSWARM, CMA-ES, ASA, NSGA-2, GA-POP50 e GA-POP100. Nesta comparação fica evidenciado que para os problemas estudados com até 30 variáveis, ele tem o melhor desempenho entre os algoritmos comparados e o segundo melhor desempenho para problemas com até 60 variáveis.

Nestes gráficos, são apresentados os resultados usando duas tolerâncias distintas: $\max(1\% ; 0,01)$ e $\max(5\% ; 0,05)$. É possível observar que os resultados são muito similares, de forma que esta alteração de tolerância tem pouco efeito na ordenação dos melhores algoritmos.

A Figura 3.14 mostra a evolução da porcentagem de problemas resolvidos em função do número de iterações. As porcentagens anteriormente apresentadas eram calculadas em função da mediana da resposta do algoritmo para um dado caso (*testbench* e número de variáveis). Para a construção da Figura 3.14, no entanto, foi considerada a resposta de cada otimização individualmente. Assim, os resultados finais são diferentes quantitativamente dos apresentados até então. É possível observar que os melhores algoritmos para até 60 variáveis permanecem sendo o MCS, o algoritmo proposto, o PSWARM e o CMAES, nesta ordem.

Figura 3.13: Porcentagem total de problemas mono-objetivo resolvidos por cada algoritmo para 2.000 iterações, onde $f(x_b)$ é a mediana da resposta do algoritmo e $f(x^*)$ é a melhor resposta encontrada para cada caso. Total para: (a) até 60 variáveis; (b) até 30 variáveis.

Figura 3.14: Porcentagem de problemas mono-objetivo resolvidos por cada algoritmo ao longo de 2.000 iterações considerando cada otimização individualmente.

Tabela 3.3: Mediana de 50 otimizações dos algoritmos comparados para cada caso dos *testbenches* matemáticos mono-objetivo de 10.000 iterações.

Testbench	Alg. Proposto	ASA	CMA-ES	GA POP100	GA POP50	MCS	NSGA	PSWARM
Número de Variáveis = 2								
rastrigin	1,92E-05	9,51E-01	4,17E-13	4,62E-02	3,08E+00	2,33E-04	1,51E+00	6,54E-14
griewank	2,47E-03	6,73E-01	8,14E-02	1,42E+00	1,42E+00	2,49E-03	4,13E-02	2,22E-02
Michalewicz	-1,80E+00	-1,75E+00	-1,80E+00	-2,29E-05	-1,95E-05	-1,80E+00	-1,80E+00	-1,80E+00
schwefel	-5,45E-07	-5,45E-07	3,57E+02	8,36E+02	8,36E+02	-5,41E-07	6,52E-02	1,18E+02
levy	5,25E-16	1,75E-03	6,43E-32	8,21E-04	1,07E-03	5,48E-21	4,47E-05	1,50E-32
rosenbrock	3,09E-03	1,76E-01	5,95E-29	1,29E-02	2,03E-02	3,90E-08	9,36E-04	2,05E-06
ackley	8,88E-16	7,52E-02	4,44E-15	2,02E-01	3,56E-01	8,88E-16	1,28E-01	8,88E-16
Número de Variáveis = 3								
rastrigin	1,16E-04	3,66E-01	1,01E-12	1,06E+01	1,49E+01	2,47E-01	8,29E-01	1,91E-13
griewank	2,47E-03	2,07E+00	7,39E-02	1,35E+00	1,36E+00	2,47E-03	6,41E-02	4,44E-02
Michalewicz	-2,76E+00	-2,11E+00	-2,68E+00	-1,34E-02	-1,25E-02	-2,76E+00	-2,76E+00	-2,76E+00
schwefel	-8,17E-07	-8,17E-07	5,34E+02	1,25E+03	1,25E+03	2,09E-01	8,34E-03	1,18E+02
levy	2,54E-15	8,75E-04	4,70E-31	2,57E-03	5,75E-03	6,09E-10	3,53E-06	1,50E-32
rosenbrock	2,20E-02	5,31E-01	2,98E-28	1,42E-01	2,58E-01	1,06E-04	2,31E-01	4,62E-03
ackley	8,88E-16	2,68E-02	4,44E-15	3,01E-01	5,78E-01	3,29E-01	2,31E-02	4,44E-15
Número de Variáveis = 5								
rastrigin	4,63E-04	1,43E+00	3,31E-12	5,87E+01	1,19E+02	6,86E-01	2,79E+00	4,82E-13
griewank	3,70E-03	9,79E+00	3,69E-02	1,21E+00	1,18E+00	1,23E-02	1,46E-01	1,38E-01
Michalewicz	-4,54E+00	-2,82E+00	-3,91E+00	-1,04E+00	-9,39E-01	-4,35E+00	-4,68E+00	-4,31E+00
schwefel	-1,36E-06	-1,36E-06	9,31E+02	2,09E+03	2,09E+03	3,78E+02	1,16E-01	2,37E+02
levy	3,19E-14	7,53E-04	1,65E-30	9,69E-03	1,81E-02	8,17E-05	4,24E-05	1,50E-32
rosenbrock	2,27E-01	1,36E+00	1,72E-27	1,08E+00	2,48E+00	7,32E-03	2,09E+00	3,58E-02
ackley	8,88E-16	2,41E-02	4,44E-15	4,34E-01	6,89E-01	5,72E-01	3,67E-02	4,44E-15
Número de Variáveis = 10								
rastrigin	4,84E-03	2,78E+01	1,87E-11	3,88E+02	6,72E+02	2,74E+00	4,99E+01	1,93E-12
griewank	4,94E-03	6,25E+01	4,93E-03	1,12E+00	1,15E+00	1,12E-01	5,99E-01	3,94E-02
Michalewicz	-8,80E+00	-4,05E+00	-7,19E+00	-4,28E+00	-4,03E+00	-8,11E+00	-9,33E+00	-8,48E+00
schwefel	2,37E+02	8,93E-02	1,90E+03	4,18E+03	4,18E+03	1,18E+03	3,57E+02	5,92E+02
levy	4,94E-13	1,54E-02	7,37E-30	3,49E-02	6,46E-02	8,17E-05	3,48E-03	1,50E-32
rosenbrock	5,32E+00	5,64E+00	3,23E-27	1,75E+01	3,00E+01	8,48E+00	7,88E+00	9,20E-01
ackley	4,44E-15	2,35E-01	1,15E-14	6,26E-01	1,02E+00	1,15E+00	2,84E-01	1,51E-14
Número de Variáveis = 20								
rastrigin	5,58E-02	1,77E+03	1,80E+03	2,55E+03	3,69E+03	1,10E+01	7,34E+02	3,36E-04
griewank	1,11E-02	2,43E+02	4,93E-03	1,09E+00	1,07E+00	1,23E-02	1,21E+00	7,39E-03
Michalewicz	-1,70E+01	-6,21E+00	-1,34E+01	-9,38E+00	-8,85E+00	-1,39E+01	-1,68E+01	-1,65E+01
schwefel	6,71E+02	7,70E+02	3,83E+03	8,37E+03	8,37E+03	2,37E+03	1,46E+03	1,42E+03
levy	3,68E-08	7,64E-01	4,54E-01	1,42E-01	1,78E-01	8,17E-05	1,43E-01	5,15E-14
rosenbrock	1,63E+01	9,16E+01	7,14E+00	8,90E+01	9,84E+01	1,84E+01	5,27E+01	1,38E+01
ackley	4,44E-15	3,52E+00	1,24E+01	1,02E+00	1,40E+00	9,95E-01	2,03E+00	3,34E-07
Número de Variáveis = 30								
rastrigin	3,39E-01	1,06E+04	5,70E+03	9,21E+03	9,51E+03	2,47E+01	3,19E+03	1,41E+00
griewank	2,11E-02	4,65E+02	2,47E-03	1,08E+00	1,07E+00	6,87E-02	2,41E+00	7,50E-03
Michalewicz	-2,46E+01	-8,15E+00	-1,73E+01	-1,37E+01	-1,33E+01	-2,11E+01	-2,25E+01	-2,43E+01
schwefel	1,05E+03	3,23E+03	5,79E+03	1,25E+04	1,25E+04	3,56E+03	3,36E+03	2,22E+03
levy	9,09E-02	6,13E+00	4,54E-01	3,05E-01	3,29E-01	8,17E-05	1,78E+00	3,71E-07
rosenbrock	2,67E+01	5,42E+02	2,09E+01	1,61E+02	1,69E+02	2,83E+01	1,76E+02	2,52E+01
ackley	4,44E-15	6,67E+00	1,25E+01	1,42E+00	1,52E+00	1,80E+00	3,56E+00	8,76E-04
Número de Variáveis = 40								
rastrigin	4,08E+03	2,85E+04	1,48E+04	2,03E+04	1,93E+04	4,39E+01	8,89E+03	1,82E+02
griewank	2,09E-02	6,89E+02	2,47E-03	1,07E+00	1,09E+00	9,51E-02	7,47E+00	1,76E-01
Michalewicz	-3,13E+01	-9,56E+00	-1,57E+01	-1,78E+01	-1,77E+01	-2,53E+01	-2,77E+01	-3,16E+01
schwefel	1,64E+03	6,33E+03	7,90E+03	1,67E+04	1,67E+04	4,74E+03	5,41E+03	2,96E+03
levy	3,00E-01	2,13E+01	1,16E+00	5,15E-01	4,79E-01	8,17E-05	1,39E+01	1,20E-03
rosenbrock	3,71E+01	2,51E+03	3,31E+01	2,31E+02	2,37E+02	3,82E+01	6,11E+02	8,20E+01
ackley	7,99E-15	9,41E+00	1,40E+01	1,75E+00	1,62E+00	1,32E+00	5,41E+00	6,06E-02
Número de Variáveis = 50								
rastrigin	7,56E+03	5,82E+04	2,95E+04	3,90E+04	3,35E+04	6,86E+01	1,97E+04	2,47E+03
griewank	1,60E-02	9,36E+02	2,47E-03	1,06E+00	1,07E+00	9,52E-02	2,47E+01	1,08E+00
Michalewicz	-3,74E+01	-1,09E+01	-9,95E+00	-2,06E+01	-2,10E+01	-2,96E+01	-3,16E+01	-3,45E+01
schwefel	2,44E+03	9,19E+03	9,94E+03	2,09E+04	2,09E+04	5,93E+03	7,81E+03	3,83E+03
levy	7,21E-01	5,00E+01	2,98E+00	7,98E-01	5,85E-01	8,17E-05	3,67E+01	9,22E-02
rosenbrock	4,73E+01	1,02E+04	4,40E+01	2,97E+02	3,06E+02	4,81E+01	1,92E+03	1,10E+02
ackley	1,51E-14	1,16E+01	1,48E+01	2,04E+00	1,76E+00	6,41E-02	8,20E+00	9,45E-01
Número de Variáveis = 60								
rastrigin	1,31E+04	9,47E+04	4,83E+04	6,40E+04	5,11E+04	9,87E+01	3,90E+04	1,07E+04
griewank	1,97E-02	1,16E+03	7,42E-03	1,06E+00	1,07E+00	1,63E-01	6,90E+01	1,88E+00
Michalewicz	-4,31E+01	-1,24E+01	-9,36E+00	-2,38E+01	-2,64E+01	-3,42E+01	-3,56E+01	-3,48E+01
schwefel	3,21E+03	1,28E+04	1,22E+04	2,51E+04	2,51E+04	7,11E+03	1,04E+04	4,81E+03
levy	1,44E+00	9,73E+01	3,22E+00	1,12E+00	6,76E-01	8,17E-05	5,77E+01	1,13E+00
rosenbrock	5,79E+01	2,34E+04	5,62E+01	3,68E+02	3,72E+02	5,80E+01	7,16E+03	2,49E+02
ackley	9,86E-14	1,32E+01	1,68E+01	2,24E+00	1,83E+00	2,12E-02	1,15E+01	3,36E+00

Figura 3.15: Porcentagem de problemas mono-objetivo resolvidos por cada algoritmo em função do número de variáveis para 10.000 iterações (considerando a resposta das medianas de cada caso de *testbench* e número de variáveis).

3.3.4.2 Número de Iterações Máximo Igual a 10.000

A Tabela 3.3 mostra os resultados de todas as simulações para testes mono-objetivos realizadas com 10.000 iterações. Como esperado, nestes testes as respostas são melhores do que nos testes limitados à 2.000 iterações.

A Figura 3.15 mostra a porcentagem de problemas resolvidos (respostas até 1% ou 0,01 próximas do melhor resultado obtido entre os algoritmos) em função do número de variáveis dos problemas para 10.000 iterações. Para 10 variáveis, o melhor resultado é do algoritmo CMA-ES. Para os demais casos, o algoritmo proposto atinge os melhores resultados.

A Figura 3.16 sumariza os resultados obtidos para problemas mono-objetivo para 10000 iterações. Em comparação aos testes com 2.000 iterações, o algoritmo proposto aumenta sua porcentagem de problemas resolvidos e passa a ter o melhor desempenho para o conjunto de testes. O PSWARM aumenta sua porcentagem de problemas resolvidos, enquanto o MCS tem seu desempenho relativo prejudicado.

A Figura 3.17 mostra a evolução da porcentagem de problemas resolvidos ao longo das 10.000 iterações. Até aproximadamente 1000 iterações, o MCS resolve um número superior aos outros algoritmos. Após, o algoritmo proposto passa a ter o melhor desempenho. Há duas

Figura 3.16: Porcentagem total de problemas mono-objetivo resolvidos por cada algoritmo para 10.000 iterações, onde $f(x_b)$ é a mediana da resposta do algoritmo e $f(x^*)$ é a melhor resposta encontrada para cada caso. Total para: (a) até 60 variáveis; (b) até 30 variáveis.

causas para o valor na iteração 2.000 diferir daquele apresentado no gráfico anterior para as 2.000 iterações. A primeira diz respeito ao valor de referência do mínimo de cada problema de otimização ser adotado como o mínimo obtido por todas as simulações de todos algoritmos para o limite máximo de iterações (como apresentado na seção 3.3.1). Assim, o mínimo de referência para o primeiro caso é diferente deste. A segunda causa é que as respostas pertencerem a conjuntos de simulação diferentes.

Em seu conjunto, estes resultados indicam que as estratégias adotadas no algoritmo proposto são eficazes. Outros testes com mais algoritmos, diferentes implementações ou outros critérios de parada podem ser realizados para dar continuidade à análise do algoritmo proposto em relação a outros. Para efeito desta tese, no entanto, os resultados obtidos são suficientes para comprovar o bom desempenho do algoritmo proposto para problemas mono-objetivo e sinalizar que a técnica é promissora para ser testada em problemas de síntese analógica.

3.3.5 Testes dos Recursos do Algoritmo

Nesta seção são analisadas variações do algoritmo proposto a fim de avaliar o impacto nas respostas obtidas. Para tanto, é realizado um experimento considerando 5 implementações do algoritmo:

Figura 3.17: Porcentagem de problemas mono-objetivo resolvidos por cada algoritmo para 10.000 iterações considerando cada otimização individualmente.

- **com *crossovers* e com variação de peso automática:** versão completa do algoritmo;
- **com *crossovers* e sem variação de peso automática:** apenas os *crossovers* sendo utilizados, a AOF auxiliar permanece igual a original;
- **sem *crossovers* e sem variação de peso automática, mas com recomeços do melhor ponto:** quando um mínimo local é detectado, a solução atual é substituída pela melhor solução encontrada até então. É equivalente ao uso de *crossovers* porém sem a disponibilidade de informação de múltiplos objetivos;
- **sem *crossovers* e com variação de peso automática:** apenas a variação de peso na AOF auxiliar é realizada;
- **sem *crossovers* e sem variação de peso automática:** o algoritmo em sua versão mais simples, sem *crossovers* nem alteração de pesos.

Os testes são realizados com os mesmos *testbenches* matemáticos utilizados na seção 3.3.2 com 2000 iterações.

A Figura 3.18 mostra os resultados da comparação. O resultado das duas variações que continham o operador de *crossover* foi superior as demais. A variação com recomeço possuiu

Figura 3.18: Porcentagem total de problemas mono-objetivo resolvidos por cada variação do algoritmo proposto para 2.000 iterações, onde $f(x_b)$ é a mediana da resposta do algoritmo e $f(x^*)$ é a melhor resposta encontrada para cada caso. Total para: (a) até 60 variáveis; (b) até 10 variáveis.

desempenho superior às variações que não tinham *crossover* e inferior as que continham *crossover*. Isto indica que o benefício do *crossover* não é apenas decorrente de um recomeço próximo a melhor resposta já obtida. A alteração automática de pesos aparenta ter influência positiva no desempenho do algoritmo apenas para um número reduzido de variáveis ou objetivos.

A análise da Figura 3.19 permite aprofundar a observação. Até 5 variáveis, o algoritmo com *crossovers* e com alteração do peso foi o melhor. Após, o algoritmo com *crossovers* e sem alteração de peso passa a ser o melhor. Devido a forma como os *testbenches* foram construídos, o número de objetivos é diretamente proporcional ao número de variáveis (em geral, número de objetivos = número de variáveis - 1). Assumindo que o número de objetivos tem mais impacto na eficácia da técnica do que o número de variáveis, é possível dizer que o uso de alteração de pesos em uma AOF secundária é válida para um número de objetivos não superior a 5. Este é, em geral, o caso para síntese de circuitos integrados analógicos. No entanto, caso o objetivo seja utilizar o algoritmo genericamente, é recomendável não utilizar o ajuste automático de pesos.

Figura 3.19: Porcentagem de problemas mono-objetivo resolvidos por cada variação do algoritmo proposto em função do número de variáveis para 2.000 iterações (considerando a resposta das medianas de cada caso de *testbench* e número de variáveis).

3.3.6 Algoritmos Comparados para o Caso Multiobjetivo

Para o caso multiobjetivo, foram utilizados para comparação os seguintes algoritmos:

- **Algoritmo proposto:** a combinação entre a variação de SA proposta nesta seção juntamente com *Particle Swarm Optimization*;
- **3ª fase do algoritmo proposto isoladamente:** apenas a implementação do *Particle Swarm Optimization* multiobjetivo utilizado na implementação do algoritmo proposto mas sem a primeira fase e segunda fase utilizando SA;
- **NSGA-II (DEB et al., 2002):** algoritmo genético multiobjetivo. A implementação utilizada para comparação foi desenvolvida por Lin (2011);
- **SPEA2 (ZITZLER et al., 2001):** algoritmo evolutivo elitista multiobjetivo. Para este trabalho foi utilizada a implementação desenvolvida na plataforma PISA (BLEULER et al., 2003) e portada para MatLab por Popov (2005);
- **MOPSO (COELLO et al., 2004):** versão multiobjetivo do algoritmo Particle Swarm Optimization. A implementação utilizada deste algoritmo foi desenvolvida por Heris

Tabela 3.4: Conjunto de testes ZDT (ZITZLER et al., 2000).

Nome	Função	Intervalo
ZDT1	$f_1(x_1) = x_1$ $g(x_2, \dots, x_m) = 1 + 9 \cdot \sum_{i=2}^m x_i / (m - 1)$ $h(f_1, g) = 1 - \sqrt{f_1/g}$ com $m = 30$	$0 \leq x_i \leq 1$
ZDT2	$f_1(x_1) = x_1$ $g(x_2, \dots, x_m) = 1 + 9 \cdot \sum_{i=2}^m x_i / (m - 1)$ $h(f_1, g) = 1 - (f_1/2)^2$ com $m = 30$	$0 \leq x_i \leq 1$
ZDT3	$f_1(x_1) = x_1$ $g(x_2, \dots, x_m) = 1 + 9 \cdot \sum_{i=2}^m x_i / (m - 1)$ $h(f_1, g) = 1 - \sqrt{(f_1/g) - (f_1/g) \sin(10\pi f_1)}$ com $m = 30$	$0 \leq x_i \leq 1$
ZDT4	$f_1(x_1) = x_1$ $g(x_2, \dots, x_m) = 1 + 10(m - 1) + \sum_{i=2}^m (x_i^2 - 10(4\pi x_i))$ $h(f_1, g) = 1 - \sqrt{(f_1/g)}$ com $m = 10$	$0 \leq x_i \leq 1$
ZDT6	$f_1(x_1) = 1 - \exp(-4x_1) \sin^6(6\pi x_1)$ $g(x_2, \dots, x_m) = 1 + 9 \cdot ((\sum_{i=2}^m x_i) / (m - 1))^{0.25}$ $h(f_1, g) = 1 - (f_1/g)^2$ com $m = 10$	$0 \leq x_i \leq 1$

(2011).

3.3.7 Testbenches para o Caso Multiobjetivo

São utilizados os conjuntos de teste ZDT (ZITZLER et al., 2000) e DTLZ (DEB et al., 2002) para o caso multiobjetivo.

Os algoritmos do conjunto de funções de teste ZDT (Tabela 3.4) são vastamente utilizados na literatura para avaliar o desempenho de algoritmos multiobjetivo. Cada problema deste conjunto possui características específicas que podem existir em problemas de otimização real de forma a dificultar a obtenção da fronteira de Pareto. Todas as funções do conjunto ZDT possuem 2 objetivos. O teste número 5 (ZDT5) não foi incluído nas comparações pois utiliza como entrada apenas variáveis de entrada do tipo binária, o que foge do escopo deste trabalho.

Os algoritmos do conjunto de funções de teste DTLZ (Tabela 3.5) foram desenvolvidos com foco em problemas com mais de 2 objetivos. Todos os problemas contidos neste conjunto são escaláveis quanto ao número de variáveis e objetivos. Assim, este conjunto é ideal para comparar algoritmos que buscam a fronteira de Pareto em problemas com muitos objetivos. Para cada função do conjunto DTLZ testada neste trabalho, quatro variações de número de variáveis, objetivo e dificuldade foram utilizadas.

Tabela 3.5: Conjunto de testes DTLZ (DEB et al., 2002).

Nome	Função	Intervalo e Variações de Parâmetros
DTLZ1	$f_1 = (1 + g)0,5 \prod_{i=1}^{M-1} y_i$ $f_{2:M-1} = (1 + g)0,5 (\prod_{i=1}^{M-m} y_i)(1 - y_{M-m+1})$ $f_M = (1 + g)0,5(1 - y_1)$ $g = 100[k + \sum_{i=1}^k ((z_i - 0,5)^2 - \cos(20\pi(z_i - 0,5)))]$	$0 \leq x_i \leq 1$
		$\{ N = 13; K = 5; M = 3; \}$
		$\{ N = 13; K = 10; M = 3; \}$
		$\{ N = 10; K = 5; M = 5; \}$
DTLZ2	$f_1 = (1 + g) \prod_{i=1}^{M-1} \cos(y_i \pi / 2)$ $f_{2:M-1} = (1 + g) (\prod_{i=1}^{M-m} \cos(y_i \pi / 2)) \cdot \sin(y_{M-m+1} \pi / 2)$ $f_M = (1 + g) \sin(y_1 \pi / 2)$ $g = \sum_{i=1}^k ((z_i - 0,5)^2 - \cos(20\pi(z_i - 0,5)))]$	$0 \leq x_i \leq 1$
		$\{ N = 13; K = 10; M = 3; \}$
		$\{ N = 18; K = 15; M = 3; \}$
		$\{ N = 15; K = 10; M = 5; \}$
DTLZ3	Igual a DTLZ2, mas com g igual a da DTLZ1	$0 \leq x_i \leq 1$
		$\{ N = 8; K = 5; M = 3; \}$
		$\{ N = 13; K = 10; M = 3; \}$
		$\{ N = 10; K = 5; M = 5; \}$
DTLZ4	Igual a DTLZ2, mas com y_i substituído por y_i^α com $\alpha > 0$	$0 \leq x_i \leq 1$
		$\{ N = 13; K = 10; M = 3; \}$
		$\{ N = 18; K = 15; M = 3; \}$
		$\{ N = 15; K = 10; M = 3; \}$
DTLZ5	Igual a DTLZ2, mas com $y_2, \dots, y_{M-1}$ substituído por $\frac{1+2gy_i}{2(1+g)}$	$0 \leq x_i \leq 1$
		$\{ N = 13; K = 10; M = 3; \}$
		$\{ N = 18; K = 15; M = 3; \}$
		$\{ N = 15; K = 10; M = 5; \}$
DTLZ6	Igual a DTLZ5, mas com $g = \sum_{i=1}^k z_i^0, 1$	$0 \leq x_i \leq 1$
		$\{ N = 13; K = 10; M = 3; \}$
		$\{ N = 18; K = 15; M = 3; \}$
		$\{ N = 15; K = 10; M = 5; \}$
DTLZ7	$f_{m=1:M-1} = y_m$ $f_M = (1 + g)(M - \sum_{i=1}^{M-1} [\frac{f_i}{1+g}(1 + \sin(3\pi f_i))])$ $g = 1 + 9 \sum_{i=1}^k z_i / k$	$0 \leq x_i \leq 1$
		$\{ N = 23; K = 20; M = 3; \}$
		$\{ N = 28; K = 25; M = 3; \}$
		$\{ N = 15; K = 20; M = 5; \}$
DTLZ7		$\{ N = 28; K = 20; M = 8; \}$

3.3.8 Resultados para Problemas Matemáticos Multiobjetivo

3.3.8.1 Número de Iterações Máximo Igual a 5.000

A Tabela 3.6 mostra as medianas dos testes multiobjetivos realizadas com 5.000 iterações. Da mesma forma como nos teste mono-objetivo, não há um algoritmo que tem desempenho melhor que os outros em todos os testbenches.

Destaca-se que a fase 3 do algoritmo proposto, quando usada isoladamente (sem a etapa inicial de Simulated Annealing utilizando *crossovers* e ajuste automático de pesos), tem desempenho pior que todos os demais algoritmos. No entanto, quando composto das 3 fases, o algoritmo proposto tem desempenho melhor que os demais em grande parte dos *testbenches*. O algoritmo SPEA2 possui bom desempenho especialmente para 2 e 3 objetivos. O algoritmo

Tabela 3.6: Mediana dos resultados dos algoritmos comparados para os *testbenches* matemáticos multi-objetivo com 5000 iterações.

Testbench	Parâmetros	Fase 3 do		MOPSO	NSGA-II	SPEA2
		Alg. Proposto	Alg. Proposto Isoladamente			
Número de Objetivos = 2						
ZDT1	$n = 30$	0,062655	0,520274	0,324838	0,194366	0,060934
ZDT1	$n = 40$	0,055438	0,482860	0,344945	0,228932	0,055432
ZDT2	$n = 30$	0,113708	0,647978	0,247547	0,281312	0,064571
ZDT2	$n = 40$	0,113790	0,667215	0,368485	0,391694	0,065935
ZDT3	$n = 30$	0,133196	0,575009	0,450728	0,322325	0,161972
ZDT3	$n = 40$	0,127365	0,558639	0,452010	0,327304	0,156290
ZDT4	$n = 10$	0,004877	0,233591	0,015872	0,010382	0,016878
ZDT4	$n = 20$	0,007461	0,267917	0,018950	0,016113	0,023811
ZDT6	$n = 10$	0,075193	0,851623	0,445078	0,121262	0,106001
ZDT6	$n = 20$	0,082209	0,903480	0,737674	0,492832	0,195856
Número de Objetivos = 3						
DTLZ1	$n = 8 \text{ e } k = 5$	0,000043	0,007813	0,004248	0,000191	0,000618
DTLZ1	$n = 13 \text{ e } k = 10$	0,000073	0,023487	0,015035	0,001631	0,008875
DTLZ2	$n = 13 \text{ e } k = 10$	0,132399	0,093783	0,162776	0,052116	0,041739
DTLZ2	$n = 18 \text{ e } k = 15$	0,039745	0,077799	0,125210	0,026021	0,021501
DTLZ3	$n = 8 \text{ e } k = 5$	0,001892	0,013031	0,007821	0,000049	0,001394
DTLZ3	$n = 13 \text{ e } k = 10$	0,000933	0,044357	0,034931	0,002161	0,014376
DTLZ4	$n = 13 \text{ e } k = 10$	0,096481	0,287975	0,173313	0,123021	0,073290
DTLZ4	$n = 18 \text{ e } k = 15$	0,046035	0,171295	0,159712	0,026746	0,020225
DTLZ5	$n = 13 \text{ e } k = 10$	0,288457	0,319850	0,279829	0,281315	0,233499
DTLZ5	$n = 18 \text{ e } k = 15$	0,220873	0,249032	0,209347	0,212657	0,159943
DTLZ6	$n = 13 \text{ e } k = 10$	0,048236	0,461993	0,697037	0,184117	0,106679
DTLZ6	$n = 18 \text{ e } k = 15$	0,018983	0,468830	0,705940	0,321784	0,200360
DTLZ7	$n = 23 \text{ e } k = 20$	0,120681	0,587019	0,307917	0,268947	0,122174
DTLZ7	$n = 28 \text{ e } k = 25$	0,112664	0,577772	0,403163	0,269064	0,117731
Número de Objetivos = 5						
DTLZ1	$n = 10 \text{ e } k = 5$	0,000000	0,003291	0,004714	0,000076	0,001798
DTLZ1	$n = 15 \text{ e } k = 10$	0,000000	0,006890	0,048028	0,001301	0,003321
DTLZ2	$n = 15 \text{ e } k = 10$	0,010930	0,026582	0,336449	0,008306	0,037102
DTLZ2	$n = 20 \text{ e } k = 15$	0,003575	0,020122	0,248380	0,005271	0,026345
DTLZ3	$n = 10 \text{ e } k = 5$	0,000070	0,011456	0,013599	0,000323	0,002707
DTLZ3	$n = 15 \text{ e } k = 10$	0,000079	0,038211	0,062487	0,004776	0,019877
DTLZ4	$n = 15 \text{ e } k = 10$	0,011926	0,052371	0,025461	0,004429	0,009401
DTLZ4	$n = 20 \text{ e } k = 15$	0,003637	0,057011	0,010214	0,001575	0,010298
DTLZ5	$n = 15 \text{ e } k = 10$	0,193381	0,269427	0,370639	0,263577	0,270462
DTLZ5	$n = 20 \text{ e } k = 15$	0,127832	0,204810	0,315766	0,197418	0,165297
DTLZ6	$n = 15 \text{ e } k = 10$	0,021244	0,312926	0,803627	0,480320	0,389018
DTLZ6	$n = 20 \text{ e } k = 15$	0,010928	0,307374	0,805442	0,458399	0,354287
DTLZ7	$n = 25 \text{ e } k = 20$	0,138369	0,620770	0,616434	0,476383	0,288250
DTLZ7	$n = 30 \text{ e } k = 25$	0,146300	0,670742	0,659460	0,532714	0,454255
Número de Objetivos = 8						
DTLZ1	$n = 13 \text{ e } k = 5$	0,000000	0,003108	0,035751	0,000013	0,000841
DTLZ1	$n = 18 \text{ e } k = 10$	0,000000	0,003553	0,131577	0,000074	0,000962
DTLZ2	$n = 18 \text{ e } k = 10$	0,002754	0,024338	0,336949	0,016439	0,082214
DTLZ2	$n = 23 \text{ e } k = 15$	0,001893	0,017559	0,248203	0,010427	0,066768
DTLZ3	$n = 13 \text{ e } k = 5$	0,000000	0,010783	0,014642	0,004065	0,026246
DTLZ3	$n = 18 \text{ e } k = 10$	0,000002	0,026194	0,081732	0,014380	0,028301
DTLZ4	$n = 18 \text{ e } k = 10$	0,002955	0,024380	0,010069	0,000582	0,038712
DTLZ4	$n = 23 \text{ e } k = 15$	0,000712	0,019208	0,001263	0,000175	0,023699
DTLZ5	$n = 18 \text{ e } k = 10$	0,148023	0,229271	0,380273	0,245044	0,378359
DTLZ5	$n = 23 \text{ e } k = 15$	0,086267	0,159367	0,285860	0,189675	0,286600
DTLZ6	$n = 18 \text{ e } k = 10$	0,020035	0,312643	0,823329	0,460867	0,380436
DTLZ6	$n = 23 \text{ e } k = 15$	0,008194	0,301432	0,837656	0,492998	0,376173
DTLZ7	$n = 28 \text{ e } k = 20$	0,278288	0,740765	0,961878	0,779614	0,669075
DTLZ7	$n = 33 \text{ e } k = 25$	0,298185	0,745048	0,947542	0,719200	0,672663

NSGA-II tem desempenho similar ao SPEA2, porém inferior para até 3 objetivos e superior a partir de então. A Figura 3.20 ilustra este comportamento através da porcentagem de problemas resolvidos em função do número de objetivos. O desempenho do algoritmo proposto tem desempenho melhor em relação aos demais para 2, 5 e 8 variáveis.

Figura 3.20: Porcentagem de problemas multiobjetivo resolvidos por cada algoritmo em função do número de objetivos para 5000 iterações.

O fato da fase 3 do algoritmo proposto ter tido um desempenho não satisfatório quando utilizada sem a fase 1 e 2 é consequência do MOPSO desenvolvido ter sido projetado apenas para distribuir as soluções pela fronteira de Pareto, não sendo eficiente quando as listas iniciais não estão preenchidas satisfatoriamente.

A Figura 3.21 contabiliza a porcentagem total de problemas resolvidos por cada algoritmo para os casos estudados. O melhor desempenho é o do algoritmo proposto, seguido por SPEA2 e NSGA-II. Tanto o MOPSO quanto a fase 3 do algoritmo proposto isoladamente tiveram desempenhos inferiores aos demais.

3.3.8.2 Número de Iterações Máximo Igual a 10.000

A Tabela 3.7 mostra as medianas dos testes multiobjetivos realizadas com 10.000 iterações. O desempenho relativo dos algoritmos para este número de iterações é similar ao de 5.000 iterações. A Figura 3.22 mostra a porcentagem de problemas resolvidos em função do número de objetivos. O algoritmo SPEA2 tem o melhor desempenho para 2 e 3 objetivos. Para 5 e 8 objetivos, o algoritmo proposto tem o melhor desempenho.

A Figura 3.23 mostra a porcentagem de problemas resolvidos por cada algoritmo. De forma similar ao caso com 5.000 iterações, o melhor desempenho é o do algoritmo proposto, seguido por SPEA2, NSGA-II, MOPSO e a fase 3 do algoritmo proposto.

Figura 3.21: Porcentagem total de problemas multiobjetivo resolvidos por cada algoritmo para 5000 iterações.

Figura 3.22: Porcentagem de problemas multiobjetivo resolvidos por cada algoritmo em função do número de objetivos para 10000 iterações.

O desempenho geral dos algoritmos melhorou com o aumento do número de iterações de 5000 para 10000. O algoritmo proposto, no entanto, teve baixo incremento na melhora, fazendo com que o algoritmo SPEA2 fosse melhor que o proposto para até 3 objetivos. O bom desempenho do algoritmo proposto em relação aos demais para mais que 3 objetivos torna o seu uso promissor para exploração da fronteira de Pareto de problemas com muitos objetivos (em inglês, *many-objective optimization*). Como a síntese de CIs pode ser abordada dessa forma,

Tabela 3.7: Mediana dos resultados dos algoritmos comparados para os *testbenches* matemáticos multi-objetivo com 10000 iterações.

Testbench	Parâmetros	Alg. Proposto	Fase 3 do		MOPSO	NSGA-II	SPEA2
			Alg. Proposto Isoladamente				
Número de Objetivos = 2							
ZDT1	$n = 30$	0,055090	0,466893	0,187870	0,084374	0,054226	
ZDT1	$n = 40$	0,056779	0,502134	0,273799	0,134537	0,054833	
ZDT2	$n = 30$	0,129975	0,618850	0,150047	0,133177	0,099939	
ZDT2	$n = 40$	0,154001	0,706780	0,191728	0,218402	0,081251	
ZDT3	$n = 30$	0,132540	0,591083	0,421454	0,206412	0,132676	
ZDT3	$n = 40$	0,132981	0,586109	0,443198	0,213432	0,133306	
ZDT4	$n = 10$	0,008289	0,251864	0,016622	0,012258	0,008090	
ZDT4	$n = 20$	0,007758	0,237493	0,015669	0,013838	0,015575	
ZDT6	$n = 10$	0,067405	0,823158	0,245385	0,085993	0,062122	
ZDT6	$n = 20$	0,069267	0,872743	0,399671	0,141091	0,069666	
Número de Objetivos = 3							
DTLZ1	$n = 8 \text{ e } k = 5$	0,000049	0,008073	0,002835	0,000065	0,000504	
DTLZ1	$n = 13 \text{ e } k = 10$	0,000063	0,026383	0,018175	0,000667	0,007103	
DTLZ2	$n = 13 \text{ e } k = 10$	0,051985	0,099669	0,260849	0,055241	0,045068	
DTLZ2	$n = 18 \text{ e } k = 15$	0,041351	0,118481	0,203405	0,035106	0,032061	
DTLZ3	$n = 8 \text{ e } k = 5$	0,001916	0,013850	0,005108	0,000027	0,000915	
DTLZ3	$n = 13 \text{ e } k = 10$	0,001258	0,059220	0,037117	0,000721	0,016033	
DTLZ4	$n = 13 \text{ e } k = 10$	0,147981	0,264047	0,264085	0,068702	0,053092	
DTLZ4	$n = 18 \text{ e } k = 15$	0,151143	0,266327	0,060233	0,039347	0,037894	
DTLZ5	$n = 13 \text{ e } k = 10$	0,313383	0,303803	0,259049	0,268981	0,222937	
DTLZ5	$n = 18 \text{ e } k = 15$	0,166129	0,220957	0,179055	0,172609	0,137097	
DTLZ6	$n = 13 \text{ e } k = 10$	0,030605	0,458879	0,701522	0,055393	0,053133	
DTLZ6	$n = 18 \text{ e } k = 15$	0,034366	0,468595	0,700671	0,145154	0,075371	
DTLZ7	$n = 23 \text{ e } k = 20$	0,105882	0,530695	0,214727	0,141171	0,096204	
DTLZ7	$n = 28 \text{ e } k = 25$	0,118426	0,625431	0,289841	0,186617	0,118325	
Número de Objetivos = 5							
DTLZ1	$n = 10 \text{ e } k = 5$	0,000000	0,003990	0,004182	0,000000	0,001143	
DTLZ1	$n = 15 \text{ e } k = 10$	0,000000	0,008052	0,075531	0,000089	0,004336	
DTLZ2	$n = 15 \text{ e } k = 10$	0,005804	0,028256	0,329424	0,003875	0,049104	
DTLZ2	$n = 20 \text{ e } k = 15$	0,004792	0,022824	0,258750	0,001458	0,049920	
DTLZ3	$n = 10 \text{ e } k = 5$	0,000001	0,008793	0,007097	0,000006	0,004580	
DTLZ3	$n = 15 \text{ e } k = 10$	0,000018	0,036746	0,055661	0,000444	0,011107	
DTLZ4	$n = 15 \text{ e } k = 10$	0,006828	0,082708	0,037377	0,003285	0,024963	
DTLZ4	$n = 20 \text{ e } k = 15$	0,004805	0,052994	0,022239	0,001143	0,020434	
DTLZ5	$n = 15 \text{ e } k = 10$	0,185466	0,270304	0,358636	0,276842	0,248527	
DTLZ5	$n = 20 \text{ e } k = 15$	0,112985	0,187064	0,286897	0,188720	0,153389	
DTLZ6	$n = 15 \text{ e } k = 10$	0,024807	0,306742	0,798581	0,447474	0,405916	
DTLZ6	$n = 20 \text{ e } k = 15$	0,013910	0,296790	0,797116	0,478650	0,365350	
DTLZ7	$n = 25 \text{ e } k = 20$	0,136056	0,647467	0,498438	0,302049	0,256649	
DTLZ7	$n = 30 \text{ e } k = 25$	0,142519	0,677195	0,576424	0,370536	0,309956	
Número de Objetivos = 8							
DTLZ1	$n = 13 \text{ e } k = 5$	0,000000	0,003585	0,031717	0,000000	0,000645	
DTLZ1	$n = 18 \text{ e } k = 10$	0,000000	0,003901	0,132424	0,000002	0,000783	
DTLZ2	$n = 18 \text{ e } k = 10$	0,001222	0,022820	0,332934	0,006874	0,218468	
DTLZ2	$n = 23 \text{ e } k = 15$	0,000447	0,015738	0,260237	0,002044	0,140302	
DTLZ3	$n = 13 \text{ e } k = 5$	0,000000	0,008899	0,006796	0,001299	0,021388	
DTLZ3	$n = 18 \text{ e } k = 10$	0,000000	0,028575	0,065940	0,005733	0,042060	
DTLZ4	$n = 18 \text{ e } k = 10$	0,000696	0,017831	0,023668	0,000554	0,114144	
DTLZ4	$n = 23 \text{ e } k = 15$	0,000936	0,012832	0,015909	0,000120	0,033407	
DTLZ5	$n = 18 \text{ e } k = 10$	0,121208	0,216564	0,337718	0,237778	0,497111	
DTLZ5	$n = 23 \text{ e } k = 15$	0,072587	0,145911	0,263637	0,165634	0,410255	
DTLZ6	$n = 18 \text{ e } k = 10$	0,029050	0,298962	0,814231	0,465697	0,405246	
DTLZ6	$n = 23 \text{ e } k = 15$	0,009158	0,287138	0,820398	0,472544	0,378927	
DTLZ7	$n = 28 \text{ e } k = 20$	0,167146	0,750846	0,890202	0,481153	0,895294	
DTLZ7	$n = 33 \text{ e } k = 25$	0,164680	0,763525	0,937669	0,564823	0,706338	

os resultados também indicam que o uso do algoritmo proposto para este tipo de problema é promissor.

Figura 3.23: Porcentagem total de problemas multiobjetivo resolvidos por cada algoritmo para 10000 iterações.

3.4 Resumo do Capítulo

Neste capítulo foram apresentados os algoritmos de otimização propostos neste trabalho. Inicialmente, foi apresentado o algoritmo para problemas mono-objetivo e seus recursos para fuga de mínimos locais utilizando informações de múltiplos objetivos. Em seguida, foi descrito o algoritmo multiobjetivo que hibridiza o algoritmo mono-objetivo com o algoritmo de *Particle Swarm Optimization*. Por fim, foram realizados diversos testes com problemas matemáticos para comparar os algoritmos propostos com outros algoritmos disponíveis na literatura. Para problemas mono-objetivo, foi observado que o algoritmo proposto tem excelente desempenho em relação aos demais comparados, sendo que para testes até 2.000 iterações ele é superior aos demais para um número de variáveis/objetivos até 30, enquanto para testes até 10.000 iterações ele foi superior aos demais para a maior parte dos números de variáveis/objetivos testados. Para problemas multiobjetivo, o algoritmo proposto supera os demais nos testes realizados para mais que 3 objetivos tanto nos testes até 5.000 quando até 10.000 iterações. Estes resultados indicam que os algoritmos propostos têm desempenho favorável quando em comparação com outros na otimização de problemas matemáticos com múltiplos mínimos locais.

4 SÍNTESE EM NÍVEL DE TRANSISTORES

Os procedimentos necessários para a síntese em nível de transistores são mais complexos do que os necessários para a solução de problemas de otimização puramente matemáticos. A conversão do problema de síntese de blocos analógicos em um problema de otimização necessita a construção da função custo considerando as características dos problemas analógicos. Ainda, para tornar a ferramenta flexível, é necessário o desenvolvimento de uma interface entre usuário e a ferramenta que permita a rápida adição de novas topologias e especificações. Neste capítulo são apresentados os procedimentos de síntese através da descrição da estrutura da ferramenta de síntese desenvolvida.

Após a ferramenta ser descrita, experimentos envolvendo os algoritmos de otimização desenvolvidos no capítulo 3 são realizados com foco na síntese em nível de transistores. Os experimentos envolvem a comparação de resultados entre algoritmos, entre projetos manuais e com ferramentas encontradas na literatura. São também realizados experimentos considerando a síntese de projetos robustos às variações de processo.

4.1 Estrutura da Ferramenta de Síntese em Nível de Circuito

Nesta seção, a estrutura e o procedimento desenvolvido para a síntese de circuitos integrados analógicos em nível de circuitos são apresentados. Nele, todos os passos necessários à síntese em nível de circuito são realizados além da otimização em si.

A função básica da ferramenta em nível de circuito é dimensionar e polarizar circuitos analógicos. A partir de informações tais como tecnologia utilizada, especificações do bloco, topologia e *testbenches*, a ferramenta deve entregar ao projetista uma *netlist* com todos as dimensões ótimas dos dispositivos e das tensões/correntes de interface. Neste trabalho, o procedimento de síntese foi realizado no ambiente do *software MATLAB*, da empresa *Mathworks*, utilizando o simulador elétrico *HSPICE*, da *Synopsys*, para avaliação dos resultados. Uma versão do algoritmo de síntese foi desenvolvido em linguagem C, mas sem todas as características da versão em *MATLAB*.

A Figura 4.1 mostra a relação entre os blocos da estrutura. A ferramenta é composta por um conjunto de funções que desempenham atividades de leitura, modificação de arquivos, criação de *netlists*, atualização de variáveis, otimização, entre outras. A fim de tornar esta estrutura organizada, ela foi dividida em suas funções principais que podem ser implementadas através de uma ou mais funções de código. Nas próximas subseções, estes blocos e seus funcionamentos serão descritos.

4.1.1 Blocos de Leitura

As interfaces de entrada e saída da ferramenta determinam algumas questões fundamentais de seu uso, como a complexidade e detalhamento que o projetista poderá conferir à síntese, a facilidade com que tarefas comuns podem ser descritas pelo usuário e quão passível é a ferramenta de ser adicionada em um fluxo de projeto envolvendo outras ferramentas. A abordagem escolhida para atender estas questões foi a adoção de arquivos textuais de entrada/saída. Estes arquivos descrevem o circuito em linguagem SPICE com adição de recursos para guiar a síntese. Dessa forma, o projetista pode aproveitar sua experiência no uso da linguagem SPICE e, através de informações adicionais acrescentadas nos mesmos arquivos de descrição, compor todo o ambiente de síntese de forma a melhor atender seus objetivos. Uma vantagem de utilizar arquivos de entrada em formato textual é que eles podem ser facilmente gerados e/ou manipulados por outras ferramentas, facilitando a geração de fluxos de projeto automatizados. O uso em um fluxo de projeto automatizado será utilizado no capítulo de síntese em nível de sistemas, quando a ferramenta usa como entrada arquivos gerados após a síntese em nível de circuito. Esta abordagem baseada em texto também facilita a utilização de capturadores de esquemático no fluxo de projeto, uma vez que a opção de exportar o tipo de *netlist* do otimizador pode ser implementada ao capturador de preferência do projetista. No Apêndice A há exemplos de arquivos de entrada da ferramenta.

As entradas da ferramenta de síntese em nível de circuito são: arquivo de configuração; arquivo de topologia; arquivo(s) de *testbenches*; arquivo de especificações e arquivo(s) de tecnologia. A leitura e organização destas informações constitui a primeira fase de todo o processo a ser realizado pela ferramenta. O arquivo de configuração é o que provê esta organização, indicando à ferramenta o caminho de todos outros arquivos e configurações básicas da síntese. Desta forma, este é o único arquivo que precisa ter seu nome repassado diretamente à ferramenta durante sua chamada. A Tabela 4.1 descreve os campos existentes no arquivo de configuração.

Além da referência aos demais arquivos fornecida pelo arquivo de configuração, os outros arquivos de entrada também possuem relações entre si. Estas relações estão ilustradas na Figura 4.2. Tanto o arquivo de topologia quanto os de *testbenches* estão em contato por portas ou

Tabela 4.1: Descrição de entradas no arquivo de configuração.

Tipo	Nome	Formato de entrada	Descrição
Arquivos	Tecnologia	techfile{1} = nome_do_arquivo ⋮ techfile{n} = nome_do_arquivo	Caminhos dos arquivos de tecnologia para diferentes condições de processo
	Testbenches	testbench{1} = nome_do_arquivo ⋮ testbench{n} = nome_do_arquivo	Caminhos dos arquivos com testbenches para avaliar o circuito
Parâmetros	Largura mínima	$Wmin = valor [metros]$	Largura mínima dos transistores quando não especificado na topologia
	Comprimento mínimo	$Lmin = valor [metros]$	Comprimento mínima dos transistores quando não especificado na topologia
	Largura máxima	$Wmax = valor [metros]$	Largura máxima dos transistores quando não especificado na topologia
	Comprimento máximo	$Lmin = valor [metros]$	Comprimento máxima dos transistores quando não especificado na topologia
	Grade de transistores	$Grid = valor [metros]$	Diferença mínima permitida na variação das dimensões dos transistores
	V_{DD}	$vdd = valor [volts]$	Tensão de alimentação positiva
	V_{SS}	$vss = valor [volts]$	Tensão de alimentação negativa
	Modelo NMOS	$nmodel = nome_do_modelo$	Modelo NMOS no arquivo de tecnologia
	Modelo PMOS	$pmodel = nome_do_modelo$	Modelo PMOS no arquivo de tecnologia
	Cálculo de Coeficiente de Inversão	$ic_compute = valor_binário$	Cálculo de coeficiente de inversão dos transistores 0 (default) = desabilitado 1 = habilitado
	Simulador	$simulator = nome_do_simulador$	Simulador a ser usado (“hspice” ou “hspicerf”)

Figura 4.1: Estrutura da ferramenta de síntese em nível de circuito.

ligações (chamados na ferramenta de *links*) que permitem sua interface. Estas mesmas portas

Figura 4.2: Relações entre arquivos de topologia, *testbenches* e especificações.

podem ser utilizadas no arquivo de especificações para facilitar as análises. As especificações, por sua vez, estão relacionadas a *testbenches* e análises específicas.

A leitura da topologia constitui a interpretação de um arquivo contendo a *netlist* do circuito que o projetista tem interesse em sintetizar. A descrição da topologia é baseada em uma descrição SPICE com parâmetros adicionais para transmitir informações auxiliares à ferramenta de síntese. As partes que compõem o arquivo de topologia estão descritas a seguir:

- **Circuito:** é a *netlist* do circuito tal como se fosse escrita para uso em uma simulação tradicional. Há a possibilidade de adicionar condicionais para que determinadas regiões do circuito sejam impressas na *netlist* apenas se algum parâmetro da otimização tenha um valor específico. Isso estende a funcionalidade da descrição SPICE para incluir ou alternar blocos de circuito durante a otimização, bem como a alteração de tipos de componentes (como exemplo da Figura 4.3). Este trecho do arquivo começa com o identificador “.circuit” e é finalizado com “.close”. Uma ilustração dessa região pode ser vista na Figura 4.4;
- **Links:** este recurso serve para indicar quais são as portas (acessos externos) do circuito. Permite que estas portas possuam conectividade com os *testbenches* durante a geração

Figura 4.3: Exemplo de situação apropriada ao uso de condicionais na geração da *netlist*: escolha do modelo de resistor em um circuito como o *Beta-multiplier*.

Figura 4.4: Descrição de circuito.

das *netlists* para simulação. No uso de subcircuitos na linguagem SPICE, a ordem de descrição das portas afeta a conectividade. Assim, o uso deste recurso adicional desvincula a

ordem de enumeração da conexão em si, facilitando a utilização de múltiplos *testbenches*. Os mesmos nomes de *links* podem ser usados no arquivo de especificações. Este trecho do arquivo começa com o identificador “.link” e é finalizado com “.close”;

- **Inclusões:** determina quais arquivos externos, além dos de tecnologia, são necessários para a simulação do circuito, por exemplo, modelos *Verilog-A*. Este trecho do arquivo começa com o identificador “.include” e é finalizado com “.close”;
- **Parâmetros:** tal como na linguagem SPICE, são usados para parametrizar um elemento do circuito. Na ferramenta de síntese, no entanto, os parâmetros descritos nessa seção também descrevem as variáveis de projeto. Desta forma, a diferença ocorre no número de termos para descrever um parâmetro entre o SPICE e a descrição presente neste trabalho. Este trecho do arquivo começa com o identificador “.params” e é finalizado com “.end”. A Figura 4.5 ilustra o formato dessa região. Os campos para cada parâmetro são:

Figura 4.5: Descrição de parâmetros.

- *nome*: deve ser o mesmo nome do parâmetro presente na *netlist*;

- *valor inicial*: valor onde o processo de síntese começará;
 - *tipo de parâmetro*: os parâmetros podem ser *variáveis*, *iguais* ou *não variáveis*. Apenas os parâmetros variáveis serão repassados para o algoritmo de otimização. Parâmetros do tipo *igual* recebem o mesmo valor do parâmetro descrito acima dele no arquivo de entrada. Parâmetros *não variáveis* são úteis na descrição de capacidades de carga, fontes de tensão ou corrente externas, etc.;
 - *tipo de número*: os parâmetros podem ser *reais*, *inteiros* ou em *grade*. Os valores *inteiros* e em *grade* são arredondados durante a otimização, sendo que os em *grade* necessitam um campo extra determinando qual o espaçamento da *grade*;
 - *valor mínimo e valor máximo*: definem o intervalo dentro do qual o valor do parâmetro pode variar. Da mesma forma como o valor inicial, este valor é opcional. Caso não preenchido, a ferramenta utilizará o valor mínimo e o máximo permitido pela tecnologia (não utilizar estes valores facilita a transição de tecnologia por não necessitar reedição do arquivo de topologia para modificá-los);
- **Região de Operação**: caso seja previsto que alguns transistores estejam operando em regiões específicas ou com restrições relativas ao coeficiente de inversão, utiliza-se esta seção para indicá-las. Este recurso tentará forçar a síntese a colocar os transistores na região de operação ou coeficiente de inversão indicados através da adição de penalidades na função custo proporcionais à distância da região (será mais explicado na seção de construção da função custo). Este trecho do arquivo começa com o identificador “.operationregion” e é finalizado com “.close”. A Figura 4.6 ilustra o formato dessa região. As possibilidades são: região de corte, região linear e região de saturação ou relações em função do coeficiente de inversão, como $IC < 0,1$ ou $IC > 10$.

A Figura 4.7 ilustra o arquivo de especificações. As especificações de projeto são os critérios que a solução final precisa cumprir ao fim da síntese. Através delas, a ferramenta de síntese poderá avaliar o circuito a cada iteração de otimização a fim de aproximar-se da solução desejada. Nesta ferramenta, qualquer medição que possa ser descritas utilizando o recurso .MEASURE do *software HSPICE* pode ser realizada (SYNOPSYS, 2014). Medições elaboradas que não são facilmente realizadas através deste recurso podem ser realizadas através de blocos em *Verilog-A*.

4.1.2 Blocos de Avaliação

Os blocos de avaliação são responsáveis por gerar e avaliar o circuito baseado nos valores das variáveis entregues pelo bloco de otimização. Estes blocos são: checagem de limites de

Figura 4.6: Descrição de região de operação.

Figura 4.7: Descrição das especificações.

dimensionamento, geração de *netlists* e simulação de *testbenches*.

A checagem de limites de dimensionamento trata de garantir que os valores de variáveis entregues pelo bloco de otimização estejam dentro dos valores permitidos de projeto.

A geração de *netlists* constitui um dos elementos centrais do bloco de avaliação. Ele é responsável por gerar todas as *netlists* que serão simuladas no próximo bloco. As *netlists* são resultados da composição da topologia com cada *testbench*, com os valores das variáveis e com as medidas que devem ser feitas para avaliar o circuito. O número de *netlists* normalmente é igual ao número de *testbenches*. Para criá-las, é necessário utilizar todos os valores de parâmetros, tanto os variáveis quanto os fixos. Por este motivo, todos os valores são adicionados a vetores durante os blocos de leitura, e não apenas as variáveis, que são necessárias para o otimizador. As medidas são extraídas das descrições de especificações. Para criar estas medidas na *netlist*, os *links* descritos na topologia, especificações e *testbenches* são substituídos pelo valor dos nós da topologia.

Uma vez que as *netlists* estejam criadas, a ferramenta pode proceder com a simulação dos *testbenches*. Este bloco estabelece a relação da ferramenta com o simulador elétrico. Este passo envolve executar o simulador sem interface gráfica, gravar o relatório de saída do simulador em um arquivo e interpretá-lo. Uma busca por todos os resultados de medidas é realizada e estas são repassadas para o bloco de cálculo de função custo.

4.1.3 Cálculo da Função Custo

Reunindo todas as medidas dos *testbenches*, este bloco faz a conversão dos resultados em uma função custo. A construção da função custo para cada especificação tem importância fundamental uma vez que é através dela que o otimizador pode saber o quão próximo está do objetivo. Como visto anteriormente, quando os múltiplos objetivos de um problema são abordados com um otimizador desenvolvido para apenas um objetivo, utiliza-se uma função objetivo agregativa (AOF), que é a soma ponderada de várias funções custos individuais:

$$C = \sum_{i=1}^n c_i \cdot w_i \quad (4.1)$$

onde C é o custo total, i é o índice de objetivo, c_i é o custo de uma função individual e w_i é o peso da função custo individual. Mesmo quando utilizando uma AOF, antes de agregar todos os custos em um só, as funções precisam ser calculadas individualmente.

As funções custo individuais são projetadas de forma a fornecer uma maior variação de custo Δc_i para resultados que estão longe das especificações. Apesar de vários tipos de função poderem ser usadas para criar a função custo, funções lineares em partes (em inglês, *piece-wise*

linear functions) foram utilizadas neste trabalho para que se possa ter um controle preciso das regiões nas quais existirão um maior $\Delta c_i / \Delta m_i$ para o custo c_i e a medida m_i . O objetivo de cada especificação fornecida pelo projetista pode ser minimização, maximização ou aproximação:

- **Maximização:** medida da solução precisa ser igual ou maior que o valor da especificação. Exemplos são o ganho DC, frequência de ganho unitário, e *slew rate*. A Figura 4.8 mostra como é implementado este tipo de função neste trabalho;

Figura 4.8: Função custo para uma especificação superior a zero e com objetivo de maximização.

- **Minimização:** medida da solução precisa ser igual ou menor que o valor da especificação. Exemplos são area, potência consumida e ruído. A Figura 4.9 mostra como é implementado este tipo de função neste trabalho;

Figura 4.9: Função custo para uma especificação superior a zero e com objetivo de minimização.

- **Aproximação:** medida da solução precisa aproximar-se do valor da especificação. Exemplos são a frequência central de um oscilador e a tensão de saída de uma fonte de referência. Neste tipo de especificação, é necessário existir uma faixa de tolerância, dentro

da qual a medida é considerada igual à especificação. A Figura 4.10 mostra como é implementado este tipo de função neste trabalho.

Figura 4.10: Função custo para uma especificação superior a zero e com objetivo de aproximação.

As funções de reta de cada especificação são criadas com base nas especificações mínimas fornecidas pelo usuário. Para um determinado valor de medida, é calculado o resultado da função do segmento de reta para cada uma das regiões (distante, próxima e após atingido). O custo para uma determinada medida é o resultado do máximo custo entre cada uma das funções de segmento de reta. A utilização destas funções de reta para o cálculo da função custo é realizada de acordo com a equação 4.2:

$$C_i = \begin{cases} \max([f_{M \text{ dist } i}(x), f_{M \text{ próx } i}(x), f_{M \text{ após } i}(x)]) & \text{se } obj_i = \textbf{maximização} \\ \max([f_{m \text{ dist } i}(x), f_{m \text{ próx } i}(x), f_{m \text{ após } i}(x)]) & \text{se } obj_i = \textbf{minimização}. \\ \max([f_{M \text{ dist } i}(x), f_{M \text{ próx } i}(x), f_{m \text{ próx } i}(x), f_{m \text{ dist } i}(x)]) & \text{se } obj_i = \textbf{aproximação} \end{cases} \quad (4.2)$$

onde $\max(\text{conjunto})$ é uma função que retorna o maior resultado, $f_{M \text{ dist } i}$ é a equação do custo para objetivos de maximização na região distante, $f_{M \text{ próx } i}$ é a equação do custo para objetivos de maximização na região próxima, $f_{M \text{ após } i}$ é a equação do custo para objetivos de maximização na região após atingir o objetivo. De forma similar, $f_{m \text{ dist}}$, $f_{m \text{ próx}}$ e $f_{m \text{ após}}$ são as equações de custo para objetivos de minimização. O cálculo de cada uma destas funções é realizado através da equação da reta elaboradas de forma que os segmentos tenham intersecções nas regiões certas.

Em um projeto analógico, as diversas especificações costumam ter ordens de grandeza diferentes entre si. Enquanto, por exemplo, a corrente pode ser calculada em μA , o ganho de frequência unitário pode estar em MHz . Desta forma, antes de combinar todas as funções custos na função agregativa de objetivos (AOF), é necessário normalizá-las de acordo com os seus

valores objetivo. A equação 4.3 mostra a normalização:

$$c_{i \text{ norm}} = \frac{c_i}{\max(|spec_i|, tol_0)} \quad (4.3)$$

onde $c_{i \text{ norm}}$ é o custo normalizado para cada objetivo, c_i é o custo calculado originalmente para cada objetivo e $spec_i$ é o valor da especificação de cada objetivo. Para evitar divisões por zero, caso o objetivo seja zero, utiliza-se o valor tol_0 na divisão. Este é um valor próximo a zero e com grandeza inferior à unidade padrão do objetivo em questão.

Um indicador contendo o número de medidas que resultaram em falha é utilizado para atribuir um alto custo a estas funções. O objetivo deste indicador é fornecer um gradiente mesmo quando lidando com projetos ainda não funcionais. Em condições nas quais as medidas não podem ser visualizadas, as penalidades de restrições quanto a região de operação dos transistores oferecem importantes pistas com respeito a qual direção deve se dar a otimização.

O projetista pode especificar restrições a serem respeitadas durante a síntese. Conforme observado na seção 2.1, estas podem ser de natureza funcional ou geométrica. As funcionais são implementadas na forma de penalidades de função custo, enquanto as geométricas são implementadas como limites inferiores e superiores para as variáveis.

No caso das restrições funcionais, como é o caso da colocação de um transistor em uma região específica de operação ou uma relação específica entre variáveis, utiliza-se termos adicionais na AOF para que ocorra a conversão de um problema com restrições para um problema sem restrições. Esta conversão permite que um algoritmo como *Simulated Annealing* seja capaz de tratar das restrições. A equação 4.4 mostra como esta conversão é realizada:

$$C = \sum_{i=1}^n c_{i \text{ norm}} \cdot w_i + \sum_{j=1}^m r_j \cdot p_j \quad (4.4)$$

onde j é o índice da penalidade, r_j é o custo associado a cada restrição, p_j é o peso e m é o número total de penalidades. É possível optar por adotar as penalidades como mais do que apenas guias de projeto, através da definição de $p_j \gg w_i$ para todos objetivos e penalidades. Dessa forma, o otimizador buscará atender inicialmente às restrições e apenas posteriormente às especificações.

No caso de transistores fora da região de operação desejada, a penalização na função custo será proporcional a distância que o transistor estiver da região desejada. O cálculo é baseado na tensão entre porta e fonte V_{GS} , na tensão entre dreno e fonte V_{DS} , na tensão de limiar V_{TH} , na tensão de saturação medida V_{DSSAT} extraídos da simulação e coeficiente de inversão calculado a partir dos resultados. A Tabela 4.2 mostra a fórmula do custo relativo às restrições de região de operação, onde r é o custo para o transistor fora de região de operação, e p é o peso associado.

Tabela 4.2: Custo associado às restrições de região de operação.

Região Desejada	Região Atual	Custo da Restrição
corte	triódo ou saturação	$r = \frac{(V_{GS} - V_{TH})}{ V_{GS} } \cdot p$
triódo	corte	$r = \frac{(V_{TH} - V_{GS})}{ V_{TH} } \cdot p$
	saturação	$r = \frac{(V_{DS} - v_{DSSAT})}{ V_{DS} } \cdot p$
saturação	corte	$r = \frac{(V_{TH} - V_{GS})}{ V_{TH} } \cdot p$
	triódo	$r = \frac{(v_{DSSAT} - V_{DS})}{ v_{DSSAT} } \cdot p$
Para definições de Coeficiente de Inversão (CI)		
$CI < x$	$CI \geq x$	$r = (CI - x) \cdot p$
$CI > x$	$CI \leq x$	$r = (x - CI) \cdot p$
$CI = x$	$CI \geq x + \text{tolerância ou}$ $CI \leq x - \text{tolerância}$	$r = (\max(CI - x, 0) + \max(x - CI, 0)) \cdot p$

Caso o projetista especifique uma tensão de *overdrive* na porta, ela também será considerada nas equações.

4.1.4 Bloco de Exposição de Resultados

Após cada rodada de simulações, o custo atual e o melhor resultado atingido até então são mostrados para o projetista. Um exemplo pode ser visualizado na Figura 4.11. Isto proporciona ao projetista uma forma de checar se as especificações escolhidas são atingíveis para a dada topologia e tecnologia. Ainda, ajuda o projetista a observar em tempo real se o algoritmo escolhido para síntese está oferecendo resultados satisfatórios. Caso os resultados não estejam convergindo, o projetista pode optar por utilizar outro algoritmo ou modificar as especificações, recomeçando o processo.

No caso da exploração multiobjetivo, após todas as fases serem concluídas, o arquivo com todas as soluções da fronteira de Pareto é gerado. Além dele, também pode ser gerado o gráfico da fronteira de Pareto. Como pode haver mais que três especificações, produzir uma representação gráfica torna-se uma tarefa não trivial. Neste trabalho, uma alternativa para solucionar este problema foi permitir ao projetista selecionar duas especificações para servirem como eixos fixos e traçar múltiplos gráficos da fronteira de Pareto para as demais especificações como sendo o terceiro eixo do gráfico. Este procedimento é melhor descrito na seção 4.8.2.


```

1  Present Measurements :
2  PD = 106.56 u [W] VR25 = 599.08 m [V] IDIODE = 7.2636 u [A]
3  STD_VR25 = 4.6584m [V] TC = 8.7333 [ppm] OUT_NOISE = 3.917 u [V]
4  LR = 35.83 u [V/V] PSRR = 80.5549 [dB] AREA = 20.85 k [um]2
5  Inversion Coefficients :
6  M1 = 4.8242 M2 = 4.8243 M3 = 4.8289 M4 = 3.4815
7  M5 = 3.5689 M6 = 4.8279 M7 = 0.0134 M8 = 0.0134
8  M9 = 4.1646 M10 = 4.1674
9
10 Best Measurements :
11 PD = 124.7 u [W] VR25 = 598.77 m [V] IDIODE = 7.5117 u [A]
12 STD_VR25 = 1.2812 m [V] TC = 3.5473 [ppm] OUT_NOISE = 1.08 u [V]
13 LR = 134.5 u [V/V] PSRR = 63.8905 [dB] AREA = 12.39 k [um]2
14 Inversion Coefficients :
15 M1 = 1.8492 M2 = 1.8493 M3 = 1.8521 M4 = 0.7208
16 M5 = 0.7609 M6 = 1.8558 M7 = 0.6223 M8 = 0.6261
17 M9 = 5.8918 M10 = 5.881
18
19 Iteration: 943
20 Current Cost: 0.61
21 Best Cost: -1E-6

```

Figura 4.11: Exemplo de informações fornecidas ao usuário durante execução da síntese para o caso de uma fonte de referência.

4.2 Síntese Considerando Monte Carlo e *Corners* de Processo

Diversos trabalhos recentes na literatura abordam a questão da síntese voltada a manufatura. Estas abordagens tentam levar a robustez às variações de processo em consideração na fase de dimensionamento. No presente trabalho, há duas formas principais através das quais a ferramenta pode auxiliar o projetista a melhorar a robustez do projeto. Estas são o uso de simulações de Monte Carlo e a simulação de *corners* de processo.

Os chamados *corners* de processo representam os casos extremos destas variações e são utilizados pelos projetistas a fim de garantir a funcionalidade do circuito dentro da margem de projeto. Os *corners* de processo podem ser vários. Abaixo, uma lista com alguns modelos de transistores que podem ser encontrados nos *kits* de projeto:

- **TM**: condição média típica;
- **WO**: condição de pior um (NMOS rápido, PMOS lento);
- **WZ**: condição de pior zero (NMOS lento, PMOS rápido);
- **WP**: condição de pior potência consumida (NMOS rápido, PMOS rápido);
- **WS**: condição de pior velocidade (NMOS lento, PMOS lento);

Sendo estes os casos extremos de projeto, levá-los em consideração durante a síntese representa garantir operação mesmo em situações pessimistas. Em projeto analógico, por ve-

zes considera-se tais casos extremos menos válidos que simulações probabilísticas (de Monte Carlo). No entanto, simulações de Monte Carlo envolvem um esforço computacional elevado. Mesmo a síntese envolvendo os *corners* já significa realizar um conjunto de simulações em vez de apenas uma (caso típico) para cada avaliação do circuito. Dessa forma, cabe ao projetista definir qual abordagem é a mais interessante para o circuito sendo projetado.

Uma alteração da função custo é necessária para incluir os resultados das simulações dos casos extremos dentro da avaliação de cada solução. Ela é realizada através da média ponderada dos resultados de simulação. É dado maior peso ao resultado de simulação da condição média típica, visto que é mais representativa da operação do circuito perante o otimizador. Às demais condições, é dado peso menor e igual entre si. A função custo final é descrita pela equação 4.5:

$$C_{CORNERS} = \frac{\sum_{i=1}^n w_i C_i}{\sum_{i=1}^n w_i} \quad (4.5)$$

onde $C_{CORNERS}$ é custo total incluindo *corners*, i é o índice da condição de processo, n é o número de condições, C_i é o resultado da função custo para cada condição de processo e w_i é o peso de cada condição. Tendo a condição de média típica o índice $i = 1$ e as demais condições de pior caso os índices seguintes, o vetor de peso pode ser construído da forma expressa em 4.6.

$$\mathbf{w} = \{w_1, w_2, w_3, \dots, w_n\} = \left\{w_1, \frac{(1 - w_1)}{(n - 1)}, \frac{(1 - w_1)}{(n - 1)}, \dots, \frac{(1 - w_1)}{(n - 1)}\right\} \quad (4.6)$$

onde w_1 é um valor dentro do intervalo (0,1). Os valores w_2 até w_n são todos parametrizados em relação ao peso da condição de média típica.

Para utilizar a síntese considerando *corners* de processo na ferramenta, o projetista apenas precisa definir um caminho de tecnologia para cada *corner* no arquivo de configuração. Por outro lado, para utilizar simulações de Monte Carlo, o projetista define os *testbenches* do projeto incluindo este recurso. Para qualquer especificação que faça referência a um *testbench* utilizando simulações de Monte Carlo, a ferramenta retornará por padrão o valor médio das respostas. Caso o projetista deseje, ele pode adicionar os prefixos “min”, “max” ou “std” ao nome da especificação para que ela retorne o mínimo, máximo ou desvio padrão, respectivamente. Assim, ao projetar uma fonte de referência, por exemplo, é possível definir a tensão de saída para um valor pré-determinado e especificar que o desvio padrão deste valor seja inferior ao máximo tolerável ao projeto.

4.3 Testes

Nesta seção são realizados experimentos aplicando os algoritmos estudados e desenvolvidos à síntese de circuitos analógicos. Os circuitos utilizados para os testes foram selecionados com

base na frequência de uso em aplicações reais e na frequência de aparecimento dos mesmos em publicações relacionadas à síntese analógica. O computador utilizado para realização dos testes apresentados foi um Intel(R) Core(TM) i7-2600 CPU com uma frequência de 3,40GHz e memória RAM de 16 GB.

Os experimentos são de três tipos distintos: testes comparativos entre algoritmos, testes comparativos entre resultados da ferramenta e resultados encontrados na literatura e, por fim, experimentos visando a busca da fronteira de Pareto. Nos testes comparativos, são realizadas comparações estatísticas entre os resultados encontrados por outros algoritmos e o proposto neste trabalho a fim de averiguar sua eficácia. Nos testes com resultados encontrados na literatura, é avaliado se os resultados da ferramenta são comparáveis com projetos manuais e, portanto, podem ser aplicados à indústria. Os experimentos que exploram a fronteira de Pareto servem para testar a capacidade dos algoritmos de fornecer material suficiente ao projetista para que este escolha entre os *tradeoffs* de projeto.

Nos testes deste capítulo, por vezes será feita a distinção entre especificações do tipo “restrição” e especificações do tipo “objetivo”. Em especificações do tipo “restrição”, o valor da função custo não tem alterações após o valor mínimo da especificação ser atendida. Do ponto de vista da ferramenta, isto significa que o peso da especificação será zero após a obtenção da especificação mínima. Por outro lado, em especificações do tipo “objetivo”, o valor da função custo tem alterações mesmo após a especificação atendida. Neste último caso, o otimizador buscará melhoras além das especificações mínimas. Os testes utilizam restrições no dimensionamento de transistores de espelhos de corrente de forma que estes tenham larguras múltiplas uns dos outros a fim de melhorar o *matching*.

4.3.1 Comparação entre Algoritmos

A comparação entre algoritmos serve ao propósito de determinar qual o algoritmo mais adequado ao problema de síntese de circuitos analógicos entre os analisados. A comparação dos algoritmos em problemas de otimização puramente matemáticos, como realizada no capítulo anterior, traz indicativos favoráveis sobre o desempenho do algoritmo proposto, mas não garante extrapolação dos resultados para o caso de síntese analógica. Conforme exposto por Wolpert e Macready (1997), a excelência de um algoritmo para um tipo de problema é sempre compensada pelo seu desempenho precário em algum outro tipo.

O teste para esta comparação é a síntese de circuitos analógicos com a limitação do número de iterações, que foi definido em 2000. O número de iterações, quando aplicado à síntese de circuitos integrados, implica em um número de simulações igual a este número multiplicado pelo número de circuitos de teste (*testbenches*). Para o exemplo de síntese de um amplificador

operacional com 6 circuitos de teste, 2.000 iterações representam 12.000 simulações elétricas. Desconsiderando o tempo de processamento do algoritmo de otimização, caso cada simulação demore 300 ms para ser executada, uma síntese completa demoraria 1 hora, o que é considerado um tempo adequado. Após o número máximo de iterações, o melhor resultado encontrado até então é salvo.

Como os algoritmos utilizam variáveis aleatórias em sua operação, diversos procedimentos de síntese são realizados. As métricas utilizadas para comparação são: média, desvio padrão, mediana e também valor mínimo e máximo de cada síntese. Além de calcular estas métricas, as respostas são comparadas estatisticamente para averiguar se a diferença entre os resultados é significativa. Para tanto, foi utilizado o teste não paramétrico de Mann-Whitney-Wilcoxon (MANN; WHITNEY, 1947), que avalia se uma de duas respostas de observações independentes tende a ter uma resposta superior a outra. Supondo um conjunto de observações $\{x_1, x_2, \dots, x_n\}$ de uma população e outro $\{y_1, y_2, \dots, y_n\}$ de outra população, o teste de Mann-Whitney-Wilcoxon baseia-se na comparação de cada observação x_i de um conjunto com cada observação y_i do outro conjunto. É contado quantas vezes uma observação x_i é maior que uma observação do conjunto y_i e este valor é atribuído a uma variável U_x . Também é contado quantas vezes uma observação y_i é maior que uma observação x_i e este valor é atribuído a uma variável U_y . Sob a hipótese nula, os valores de U_x e U_y são próximos, e, consequentemente, a probabilidade de uma amostra qualquer de uma população ser maior que a outra é igual a 0,5. A formulação das hipóteses está a seguir:

$$\begin{aligned} H_0 : P(x_i > y_i) &= \frac{1}{2} \\ H_1 : P(x_i > y_i) &\neq \frac{1}{2} \end{aligned} \quad (4.7)$$

onde H_0 é a hipótese nula, H_1 é a hipótese alternativa, x_i é uma observação de um conjunto e y_i é uma observação de outro conjunto.

Como nesta seção foram comparados mais de dois métodos, foi necessário realizar diversos procedimentos de comparação utilizando este teste. Neste trabalho, cada combinação de algoritmo e circuito é sintetizado 10 vezes. No caso do teste de Mann-Whitney-Wilcoxon não mostrar diferença significativa (com nível de significância de 5%) entre os resultados para 10 amostras, um novo conjunto de 5 amostras é gerado. Diferentemente do caso dos problemas matemáticos do capítulo anterior, neste caso a solução inicial fornecida aos otimizadores é composta pelas menores dimensões possíveis disponíveis na tecnologia. Em testes realizados, este ponto de partida mostrou-se mais favorável aos otimizadores do que a escolha aleatória de um ponto no espaço de projeto.

As topologias utilizadas para esta comparação entre métodos são o amplificador Miller, o amplificador *folded cascode* complementar, um LNA e uma fonte de referência.

Figura 4.12: Topologia do amplificador do tipo Miller.

Os resultados serão apresentados da forma descrita a seguir. Inicialmente, para cada caso, uma tabela mostra as especificações do projeto e os melhores resultados obtidos por cada um dos algoritmos para a síntese. Após, é apresentada uma tabela com o resultado da função custo final para cada algoritmo em cada uma de suas sínteses. Em seguida, dados estatísticos sobre o conjunto de testes são mostrados.

4.3.2 Amplificador Miller

Este circuito é um amplificador operacional composto por dois estágios. A topologia pode ser vista na Figura 4.12. O primeiro estágio é um par diferencial M1-M2 e com carga ativa formada pelo espelho de corrente M3-M4. A corrente que flui pelo par diferencial é fornecida pelo transistor M5, que é espelho de corrente junto com M6 e M8. O segundo estágio é formado pelo estágio fonte comum com o transistor M7 com carga M6 (fonte de corrente). Há um capacitor de compensação $C1$ para introduzir um polo em baixa frequência.

O amplificador Miller sintetizado utiliza uma tensão de alimentação de +1,65 e -1,65 e tecnologia AMS 0,35 μm . A capacitância de carga foi definida em 1 pF. Os amplificadores estudados neste trabalho compartilham um conjunto de *testbenches* baseados nos descritos por Allen e Holberg (2002). Eles estão expostos na Figura 4.13, onde UGF é a frequência de ganho unitário (do inglês, *Unity Gain Frequency*), ICMR é a faixa de modo comum de entrada (do inglês, *Input Common Mode Range*), CMRR é a taxa de rejeição do modo comum (do inglês, *Common Mode Rejection Rate*) e PSRR é a taxa de rejeição da fonte de alimentação (do inglês, *Power Supply Rejection Rate*).

Tabela 4.3: Melhores resultados dos algoritmos comparados para a síntese do circuito do amplificador Miller com até 2000 iterações.

Medida	Espec.	Tipo*	Alg. Proposto	ASA	CMA-ES	GA POP 50	GA POP 100	MCS	NSGA	PSWARM
Freq. de Ganho Unitário M[Hz]	> 15	Obj.	51,35	21,05	16,36	13,76	12,02	15,39	50,89	15,90
Ganho DC dB[V]	> 80	Obj.	80,20	82,04	80,33	83,98	73,52	59,38	49,23	80,87
Margem de Fase [°]	> 60 e < 90	Restr.	60,12	62,30	71,24	54,87	48,12	60,37	29,12	67,01
Slew Rate [V/ μ s]	> 18	Obj.	38,74	26,29	21,28	10,02	10,97	20,33	10,87	17,44
Corrente consumida μ [A]	< 600	Obj.	396,6	303,10	405,60	85,64	77,53	347,80	449,40	327,00
ICMR [V]	> 1,5	Restr.	3,06	3,02	3,01	2,99	3,02	3,00	3,18	3,01
Offset m[V]	< 5	Restr.	0,10	0,04	4,00e-03	0,05	0,03	0,74	1,24	0,09
Faixa de Saída [V]	> 2	Restr.	2,79	2,79	2,80	2,77	2,76	2,79	2,80	2,78
CMRR dB[V]	> 70	Restr.	78,09	79,68	80,90	143,39	97,70	66,52	60,20	76,89
PSRR dB[V]	> 70	Restr.	95,84	89,43	88,72	88,85	108,15	63,79	73,91	87,10
Area [μ m ²]	< 5000	Obj.	1209	2410	3282	5408	1883	3345	4723	1859

* Restr. = especificação do tipo "Restrição", no qual a função custo não tem alterações após a especificação atendida;

Obj. = especificação do tipo "Objetivo", no qual a função custo tem alterações mesmo após a especificação atendida.

Para este teste, o comprimento mínimo dos transistores é $0,35 \mu\text{m}$ e a largura $0,6 \mu\text{m}$. O valor máximo do comprimento do *finger* foi arbitrado em 50 vezes o mínimo e a largura do *finger* em 300 vezes o mínimo. O número máximo de *fingers* foi arbitrado em 50. Esses intervalos permitem aos algoritmos tanto a elaboração de transistores grandes sem uso de *fingers* quanto o uso de transistores com grande multiplicidade através destes *fingers*. Assim, apesar de o espaço de projeto permitir transistores com dimensões muito grandes (caso sejam usados múltiplos *fingers* e dimensões grandes para cada um), a existência de objetivo de redução de área visa guiar os algoritmos ao uso prudente destas variáveis. O capacitor de compensação pode variar de 1 pF até 10 pF e a corrente de polarização de 100 nA até $100 \mu\text{A}$.

Inicialmente foram realizadas 10 sínteses usando cada algoritmo, mas foram necessárias 5 sínteses adicionais para que o teste de Wilcoxon-Mann-Whitney mostrasse diferença estatisticamente significativa entre as respostas. Este procedimento é realizado de acordo com a descrição dos testes apresentado na seção 4.3.1. As especificações do amplificador Miller foram baseadas no projeto manual realizado por Cortes e Bampi (2006) e se encontram na Tabela 4.3, juntamente com os melhores resultados obtidos por cada um dos algoritmos após 15 sínteses.

É possível observar que o algoritmo proposto, ASA, CMA-ES e PSWARM atenderam as especificações mínimas. Apesar disso, os algoritmos ASA e CMA-ES não deixaram todos os transistores na região de operação desejada, o que fica evidenciado pelo valor da função custo ser superior a zero na Tabela 4.5 (que será apresentada a seguir). Dessa forma, apenas o algoritmo proposto e o PSWARM foram capazes de atingir os resultados esperados em pelo menos uma das 15 sínteses. A Tabela 4.4 mostra as dimensões finais dos dispositivos nos melhores circuitos sintetizados pelos algoritmos.

Os resultados da função custo final (equação 4.4) das sínteses para cada método estão expostos na Tabela 4.5 para o amplificador Miller. A melhor mediana é do algoritmo proposto

Figura 4.13: *Testbenches* dos amplificadores para extrair as medições (baseado nos *testbenches* descritos por Allen e Holberg (2002)).

(-2E-6), seguido pelos algoritmos PSWARM, ASA, MCS, CMA-ES, GA POP50, GA POP100 e NSGA (em ordem de melhor para pior). É possível observar que mesmo o pior valor obtido pelo algoritmo proposto (3,96E-1) é melhor do que a mediana da maior parte dos algoritmos comparados. Os resultados do teste de Wilcoxon-Mann-Whitney mostram que, exceto para os pares (ASA, CMA-ES), (CMA-ES, MCS), (ASA, MCS) e (GA POP100, NSGA), todas as diferenças nos conjuntos de soluções são significativas.

Com os resultados da função custo juntamente com os resultados do teste de Wilcoxon-

Tabela 4.4: Dimensões dos melhores resultados obtidos pelos algoritmos comparados para a síntese do circuito do amplificador Miller com até 2000 iterações.

Disp.*	Algoritmo Proposto	ASA	CMA-ES	GA POP 50	GA POP 100	MCS	NSGA	PSWARM
M1	56,4(x4) / 1,1	94,7 / 2,5	71,6 / 1,5	50,7(x2) / 1,4	7,9(x39) / 1,6	0,6(x28) / 0,3	116,0(x4) / 0,4	27,4(x7) / 2,5
M2	56,4(x4) / 1,1	94,7 / 2,5	71,6 / 1,5	50,7(x2) / 1,4	7,9(x39) / 1,6	0,6(x28) / 0,3	116,0(x4) / 0,4	27,4(x7) / 2,5
M3	12,8(x2) / 0,8	28,3(x4) / 3,0	18,8(x8) / 1,5	8,4(x49) / 0,6	24,1(x2) / 1,2	80,3(x25) / 0,3	52,7(x4) / 1,6	14,8(x8) / 2,4
M4	12,8(x2) / 0,8	28,3(x4) / 3,0	18,8(x8) / 1,5	8,4(x49) / 0,6	24,1(x2) / 1,2	80,3(x25) / 0,3	52,7(x4) / 1,6	14,8(x8) / 2,4
M5	15,9(x2) / 0,4	38,0(x19) / 0,5	36,4(x16) / 0,5	30,1(x6) / 5,8	11,3(x5) / 0,9	0,6(x24) / 0,3	64,0(x47) / 0,3	24,4(x7) / 0,4
M6	15,9(x12) / 0,4	38,0(x33) / 0,5	36,4(x47) / 0,5	30,1(x14) / 5,8	11,3(x9) / 0,9	0,6(x50) / 0,3	64,0(x19) / 0,3	24,4(x6) / 0,4
M7	41,5(x14) / 1,0	120,0(x2) / 0,7	87,1(x11) / 0,8	98,2(x15) / 0,7	32,9(x8) / 1,9	109,0(x50) / 0,3	17,5(x46) / 2,1	27,7(x8) / 0,8
M8	15,9 / 0,4	38,0(x5) / 0,5	36,4(x22) / 0,5	30,1 / 5,8	11,3(x8) / 0,9	0,6(x25) / 0,3	64,0(x23) / 0,3	24,4(x7) / 0,4
C1	1,0 [pF]	2,5 [pF]	2,7 [pF]	2,0 [pF]	2,0 [pF]	2,3 [pF]	3,1 [pF]	5,4 [pF]
I1	21 [μA]	23 [μA]	91 [μA]	4 [μA]	28 [μA]	84 [μA]	91 [μA]	100 [μA]

* Dispositivo, no qual os transistores têm a largura (W) e comprimento (L) em μm . Os valores são expressos no formato $W(xf)/L$, onde f é um fator multiplicativo representando o número de *fingers*.

Tabela 4.5: Resultados dos algoritmos comparados para a síntese do Amplificador Miller.

Síntese	Alg. Proposto	ASA	CMA-ES	GA POP100	GA POP50	MCS	NSGA	PSWARM
1	-2,00E-06	2,80E+01	8,88E-01	2,74E+02	1,12E+02	5,52E+00	1,21E+02	-4,00E-06
2	-4,00E-06	5,82E+00	2,08E+01	8,48E+01	7,22E+01	5,52E+00	1,09E+02	-1,00E-06
3	-2,00E-06	2,74E+01	8,93E-01	2,17E+02	6,63E+01	5,52E+00	1,45E+02	-1,00E-06
4	-1,00E-06	1,73E+01	2,74E+01	1,32E+02	2,34E+02	5,52E+00	7,25E+01	1,56E-01
5	-1,00E-06	3,65E-01	2,74E+01	1,42E+02	2,67E+01	5,52E+00	8,15E+01	-2,00E-06
6	-1,00E-06	1,56E-01	5,04E+00	1,36E+01	1,08E+02	5,52E+00	1,82E+02	1,14E+00
7	-2,00E-06	1,64E+00	3,16E+00	1,34E+02	4,94E+01	5,52E+00	1,06E+02	5,26E-02
8	-3,00E-06	1,84E+00	2,74E+01	1,30E+02	1,54E+02	5,52E+00	1,05E+02	-2,00E-06
9	-1,00E-06	3,49E+01	3,52E+00	1,24E+02	7,68E+01	5,52E+00	2,06E+02	-2,00E-06
10	-3,00E-06	1,31E+00	2,48E+00	1,64E+02	1,27E+02	5,52E+00	4,12E+01	-1,00E-06
11	-2,00E-06	3,16E+00	1,35E-02	4,42E+02	1,24E+01	5,52E+00	8,50E+01	-1,00E-06
12	-3,00E-06	6,62E-01	2,85E+01	1,19E+02	8,82E+01	5,52E+00	8,49E+01	3,73E-03
13	-3,00E-06	5,90E-03	3,15E+01	1,99E+02	6,06E+01	5,52E+00	1,30E+02	7,07E-01
14	3,96E-01	7,26E-01	2,74E+01	4,34E+01	5,01E+01	5,52E+00	1,83E+02	7,57E-01
15	-3,00E-06	2,75E+00	2,74E+01	2,13E+02	3,29E+01	5,52E+00	1,51E+02	2,31E+00
Média	2,64E-02	8,41E+00	1,56E+01	1,62E+02	8,47E+01	5,52E+00	1,20E+02	3,42E-01
Variância	1,05E-02	1,47E+02	1,71E+02	1,05E+04	3,22E+03	0,00E+00	2,14E+03	4,27E-01
Mediana	-2,00E-06	1,84E+00	2,08E+01	1,34E+02	7,22E+01	5,52E+00	1,09E+02	-1,00E-06
Mínimo	-4,00E-06	5,90E-03	1,35E-02	1,36E+01	1,24E+01	5,52E+00	4,12E+01	-4,00E-06
Máximo	3,96E-01	3,49E+01	3,15E+01	4,42E+02	2,34E+02	5,52E+00	2,06E+02	2,31E+00
Diferentes com $p < 0,05$ no teste de Wilcoxon-Mann-Whitney								
Alg. Proposto	-	Sim	Sim	Sim	Sim	Sim	Sim	Sim
ASA	Sim	-	Não	Sim	Sim	Não	Sim	Sim
CMA-ES	Sim	Não	-	Sim	Sim	Não	Sim	Sim
GA POP100	Sim	Sim	Sim	-	Sim	Sim	Não	Sim
GA POP50	Sim	Sim	Sim	Sim	-	Sim	Sim	Sim
MCS	Sim	Não	Não	Sim	Sim	-	Sim	Sim
NSGA	Sim	Sim	Sim	Não	Sim	Sim	-	Sim
PSWARM	Sim	Sim	Sim	Sim	Sim	Sim	Sim	-

Mann-Whitney, é possível afirmar que o algoritmo proposto apresentou os melhores resultados para as sínteses do amplificador Miller do conjunto de testes realizado. Sabendo que o valor da função custo igual ou inferior a zero significa que as especificações foram atendidas, é possível observar pela Tabela 4.5 que o algoritmo proposto atingiu as especificações em 14 das 15 sínteses, o PSWARM atingiu em 8 das 15 sínteses e os demais não atingiram as especificações em nenhuma síntese. A figura 4.14 mostra o decaimento da mediana da função custo ao longo do número de iterações. Nas iterações iniciais, MCS é o que tem decaimento mais rápido, porém este logo é estabilizado. A mediana do algoritmo PSWARM decai para menos que 0 após 1500 iterações. Já a mediana do algoritmo proposto decai para menos que 0 após aproximadamente 900 iterações. Os demais algoritmos apresentam decaimentos mais lentos que os mencionados

Figura 4.14: Melhor função custo em função do número de iterações para cada algoritmo na síntese do amplificador Miller (cada valor é resultado da mediana entre todas as sínteses para cada algoritmo).

ao longo das 2000 iterações.

4.3.3 Amplificador Folded Cascode Complementar

Este circuito foi proposto por Vallee e El-Masry (1994), sendo descrito como um *folded cascode* de banda larga com *fast settling*. Sua topologia pode ser vista na Figura 4.15. Esta arquitetura é uma versão modificada da arquitetura *current steering* incorporando um estágio *cascode* de classe AB.

Para o amplificador *folded cascode* complementar foi utilizada uma tensão de alimentação de +1,65 e -1,65, a tecnologia AMS 0,35 e capacitância de saída de 10 pF. As especificações do amplificador *folded cascode* complementar foram baseadas no projeto manual realizado por Vallee e El-Masry (1994) e se encontram na Tabela 4.6, juntamente com os melhores resultados obtidos por cada um dos algoritmos entre 10 sínteses para cada um. É possível observar que o algoritmo proposto neste trabalho, o ASA, CMA-ES e o PSWARM foram capazes de atingir respostas atendendo as especificações mínimas em pelo menos uma de suas sínteses (valor da função negativa). Os demais algoritmos obtêm resultados fora das especificações em todas as sínteses.

As restrições nas variáveis dos transistores são as mesmas apresentadas para o caso do amplificador Miller e a corrente de polarização pode variar de 100 nA até 200 μ A. A Tabela 4.7

Figura 4.15: Topologia do amplificador *folded cascode* complementar.

Tabela 4.6: Melhores resultados dos algoritmos comparados para a síntese do circuito do amplificador *folded cascode* complementar com até 2000 iterações.

Medida	Espec.	Tipo*	Alg. Proposto	ASA	CMA-ES	GA POP 50	GA POP 100	MCS	NSGA	PSWARM
Frequência de Ganho Unitário M[Hz]	> 180	Obj.	199,00	186,90	229,50	154,70	163,00	229,50	59,89	188,70
Ganho DC dB[V]	> 60	Obj.	63,30	69,79	72,97	83,35	82,25	59,37	86,60	69,63
Margem de Fase [°]	> 60 e < 90	Restr.	60,03	61,45	60,69	41,89	37,83	60,33	45,40	60,18
Slew Rate [V/μs]	> 265	Obj.	494,30	345,80	335,30	182,40	198,50	265,00	56,38	462,30
Corrente Consumida m[A]	< 1,5	Obj.	0,74	1,24	1,25	1,11	1,79	0,57	0,72	0,85
ICMR [V]	> 2	Restr.	2,39	2,33	2,72	2,64	2,80	2,44	2,82	2,69
Offset m[V]	< 5	Restr.	0,22	0,01	0,07	6,33e-03	2,22e-13	0,16	0,01	0,03
Faixa de Saída [V]	> 2	Restr.	2,00	2,33	2,26	2,17	2,43	1,77	1,82	2,13
CMRR db[V]	> 70	Restr.	93,72	127,44	104,37	100,27	126,84	85,63	116,20	102,40
PSRR db[V]	> 70	Restr.	70,16	83,10	70,69	97,70	88,25	87,62	88,72	80,03
Area [μm ²]	< 16000	Obj.	740,21	9493,00	1,10e+04	1,72e+04	1,54e+04	873,25	1,78e+04	7471,00

* Restr. = especificação do tipo “Restrição”, no qual a função custo não tem alterações após a especificação atendida;

Obj. = especificação do tipo “Objetivo”, no qual a função custo tem alterações mesmo após a especificação atendida.

mostra as dimensões finais dos dispositivos nos melhores circuitos sintetizados pelos algoritmos.

Os resultados da função custo final das sínteses para cada método está exposto na Tabela 4.8. Da mesma forma como no caso do amplificador Miller, o algoritmo proposto também obtém a melhor mediana (-1E-6), seguido pelos algoritmos, PSWARM, CMA-ES, ASA, MCS, GA POP50, GA POP100 e NSGA. Novamente, o pior valor obtido pelo algoritmo proposto (1,85E-1) é melhor do que a mediana e também do mínimo (melhor valor) da maior parte dos algoritmos comparados. Os resultados do teste de Wilcoxon-Mann-Whitney mostram que, exceto para os pares (ASA, CMA-ES), (ASA, PSWARM), (CMA-ES, PSWARM), (GA POP50, GA POP100), (GA POP100, NSGA), todas as diferenças entre os conjuntos são significativas. Assim, o algoritmo proposto tem resposta diferente estatisticamente aos demais algoritmos. A partir dos resultados apresentados, é possível afirmar que o algoritmo proposto apresentou os

Tabela 4.7: Dimensões dos melhores resultados obtidos pelos algoritmos comparados para a síntese do circuito do amplificador *folded cascode* complementar com até 2000 iterações.

Disp.*	Alg. Proposto	ASA	CMA-ES	GA POP 50	GA POP 100	MCS	NSGA	PSWARM
M1	25,5 / 0,5	103(x4) / 2,1	175(x2) / 1,8	107(x3) / 0,5	71(x2) / 1,9	0,6(x50) / 0,3	0,6(x50) / 3,2	14,9(x4) / 0,3
M2	25,5 / 0,5	103(x4) / 2,1	175(x2) / 1,8	107(x3) / 0,5	71(x2) / 1,9	0,6(x50) / 0,3	0,6(x50) / 3,2	14,9(x4) / 0,3
M3	0,8(x4) / 0,4	2,4(x49) / 1,5	13,4(x15) / 0,3	6,3(x14) / 9,3	4,7(x5) / 2,5	20,3(x50) / 0,3	3,4(x13) / 9,4	180(x10) / 0,3
M4	0,8(x20) / 0,4	2,4(x12) / 1,5	13,4(x31) / 0,3	6,3 / 9,3	4,7(x3) / 2,5	20,3(x50) / 0,3	3,4(x9) / 9,4	180(x6) / 0,3
M5	10,1(x6) / 0,6	55,9 / 8	170 / 1,1	170(x30) / 1,1	4,9(x39) / 0,7	0,6(x50) / 0,3	104(x25) / 0,3	180 / 15,7
M6	10,1(x6) / 0,6	55,9 / 8	170 / 1,1	170(x30) / 1,1	4,9(x39) / 0,7	0,6(x50) / 0,3	104(x25) / 0,3	180 / 15,7
M7	10,6(x15) / 0,5	40,5(x5) / 1,6	5,1(x47) / 12,9	5,5 / 0,4	22,1(x46) / 5,9	0,6(x42) / 0,3	37,7(x41) / 3,8	180 / 0,3
M8	10,6 / 0,5	40,5(x27) / 1,6	5,1(x13) / 12,9	5,5 / 0,4	22,1(x5) / 5,9	0,6(x26) / 0,3	37,7(x16) / 3,8	180 / 0,3
M9	22,7(x2) / 0,5	98,1(x20) / 1	14,2(x11) / 1,5	10(x26) / 0,9	53,4(x21) / 2,2	0,6(x44) / 0,3	58,2(x23) / 0,3	180 / 1,5
M10	22,7(x2) / 0,5	98,1(x20) / 1	14,2(x11) / 1,5	10(x26) / 0,9	53,4(x21) / 2,2	0,6(x44) / 0,3	58,2(x23) / 0,3	180 / 1,5
M11	35,6 / 0,4	56,2(x3) / 0,5	29,5(x13) / 0,3	99,2(x19) / 0,8	113(x17) / 0,5	2(x50) / 0,3	130(x8) / 2	13,2(x4) / 0,3
M12	35,6 / 0,4	56,2(x3) / 0,5	29,5(x13) / 0,3	99,2(x19) / 0,8	113(x17) / 0,5	2(x50) / 0,3	130(x8) / 2	13,2(x4) / 0,3
M13	7,0(x5) / 0,5	24,6(x3) / 0,5	96,2(x3) / 0,3	43,6(x14) / 0,6	16,3(x4) / 2,6	0,6(x12) / 0,3	0,6(x26) / 14,1	28,3 / 0,5
M14	7,0(x5) / 0,5	24,6(x3) / 0,5	96,2(x3) / 0,3	43,6(x14) / 0,6	16,3(x4) / 2,6	0,6(x12) / 0,3	0,6(x26) / 14,1	28,3 / 0,5
M15	174 / 1,3	136(x2) / 1	130(x8) / 2,1	1,7(x32) / 1,5	42,1(x4) / 1	0,6(x30) / 0,3	76,4 / 8,9	52,4 / 0,3
M16	174 / 1,3	136(x2) / 1	130(x8) / 2,1	1,7(x32) / 1,5	42,1(x4) / 1	0,6(x30) / 0,3	76,4 / 8,9	52,4 / 0,3
II	41 [μ A]	111 [μ A]	137 [μ A]	110 [μ A]	127 [μ A]	143 [μ A]	169 [μ A]	200 [μ A]

* Dispositivo, no qual os transistores têm a largura (W) e comprimento (L) em μ m. Os valores são expressos no formato $W(x,f) / L$, onde f é um fator multiplicativo representando o número de *fingers*.

Tabela 4.8: Resultados dos algoritmos comparados para a síntese do amplificador *folded cascode* complementar.

Síntese	Alg. Proposto	ASA	CMA-ES	GA POP100	GA POP50	MCS	NSGA	PSWARM
1	-1,00E-06	5,69E-03	-0,00E+00	1,25E+02	8,60E+00	1,27E+00	2,15E+02	3,55E-01
2	-1,00E-06	-0,00E+00	3,88E+00	1,45E+01	2,16E+02	1,27E+00	1,60E+02	-1,00E-06
3	-2,00E-06	3,02E+00	3,38E-01	1,35E+02	1,59E+02	1,27E+00	1,17E+02	-1,00E-06
4	-2,00E-06	7,01E-01	9,37E-01	2,22E+02	1,59E+02	1,27E+00	2,10E+02	1,35E-01
5	8,08E-02	7,40E-01	4,51E-01	2,48E+02	5,84E+01	1,27E+00	1,59E+02	-1,00E-06
6	-1,00E-06	6,17E-01	-1,00E-06	9,72E+01	1,42E+02	1,27E+00	2,67E+02	5,66E-01
7	1,85E-01	8,76E-02	7,24E+00	2,51E+02	3,08E+01	1,27E+00	2,04E+02	1,36E+00
8	-2,00E-06	8,02E-01	5,43E-01	5,35E+01	1,29E+02	1,27E+00	2,51E+02	-1,00E-06
9	-2,00E-06	-1,00E-06	-1,00E-06	2,06E+02	5,12E+01	1,27E+00	9,19E+01	1,89E-03
10	-1,00E-06	6,42E-01	-1,00E-06	7,51E+01	1,18E+02	1,27E+00	1,51E+02	5,60E-01
Média	2,66E-02	6,62E-01	1,34E+00	1,43E+02	1,07E+02	1,27E+00	1,83E+02	2,98E-01
Variância	3,76E-03	8,05E-01	5,67E+00	7,16E+03	4,46E+03	0,00E+00	3,18E+03	1,94E-01
Mediana	-1,00E-06	6,29E-01	3,95E-01	1,30E+02	1,24E+02	1,27E+00	1,82E+02	6,84E-02
Mínimo	-2,00E-06	-1,00E-06	-1,00E-06	1,45E+01	8,60E+00	1,27E+00	9,19E+01	-1,00E-06
Máximo	1,85E-01	3,02E+00	7,24E+00	2,51E+02	2,16E+02	1,27E+00	2,67E+02	1,36E+00
Algoritmos Diferentes com $p < 0,05$ no teste de Wilcoxon-Mann-Whitney								
Alg. Proposto	-	Sim	Sim	Sim	Sim	Sim	Sim	Sim
ASA	Sim	-	Não	Sim	Sim	Sim	Sim	Não
CMA-ES	Sim	Não	-	Sim	Sim	Sim	Sim	Não
GA POP100	Sim	Sim	Sim	-	Não	Sim	Não	Sim
GA POP50	Sim	Sim	Sim	Não	-	Sim	Sim	Sim
MCS	Sim	Sim	Sim	Sim	Sim	-	Sim	Sim
NSGA	Sim	Sim	Sim	Não	Sim	Sim	-	Sim
PSWARM	Sim	Não	Não	Sim	Sim	Sim	Sim	-

melhores resultados para as sínteses do amplificador *folded cascode* complementar do conjunto de testes realizado. O algoritmo proposto atingiu as especificações em 8 das 10 sínteses, o PSWARM e o CMA-ES atingiram em 4 das 10 sínteses e os demais não atingiram as especificações em nenhuma síntese.

A Figura 4.16 mostra o decaimento da mediana da função custo ao longo do número de iterações. Similarmente ao caso do amplificador Miller, o MCS é o que tem decaimento mais rápido, porém estabiliza rapidamente. A mediana do algoritmo proposto decai para menos que 0 após aproximadamente 700 iterações. Os demais algoritmos apresentam decaimentos lentos

Figura 4.16: Melhor função custo em função do número de iterações para cada algoritmo na síntese do amplificador *folded cascode* complementar (cada valor é resultado da mediana entre todas as sínteses para cada algoritmo).

e suas medianas não chegam a alcançar o valor zero durante as 2000 iterações.

4.3.4 Amplificador de Baixo Ruído (LNA)

A topologia de amplificador de baixo ruído (LNA, do inglês, *Low Noise Amplifier*) utilizada para as sínteses é a *cascode* com degeneração indutiva, frequentemente utilizada para aplicações de banda estreita desde o trabalho de Shaeffer e Lee (1997). A Figura 4.17 mostra a topologia, na qual os indutores L_g e L_s e o transistor M_1 são responsáveis por manter o casamento da impedância de entrada na banda de interesse enquanto mantém uma baixa figura de ruído. O indutor L_d e o divisor capacitivo formados por C_1 e C_2 são usados para obter ressonância na saída e casamento de impedância. O transistor M_2 , por sua vez, é responsável pela redução do efeito Miller. O resistor R_{bias} evita que sinais RF trafeguem pelo caminho de polarização. Por último, o capacitor C_{in} realiza o desacoplamento DC.

A tecnologia CMOS utilizada para as sínteses é a IBM CMRF7SF 180 nm trabalhando com um tensão de alimentação de 1,8 V. Como este se trata de um processo melhorado para aplicações RF, há a disponibilidade de indutores parametrizados com bom fator de qualidade e capacitores MIM (*Metal-Insulator-Metal*) para serem implementados e sintetizados. Foram utilizadas impedâncias de entrada e de saída de 50 Ω . Para este teste, o comprimento mínimo

Figura 4.17: Topologia do amplificador de baixo ruído para aplicações de banda estreita (SHAFFER; LEE, 1997).

dos transistores é 180 nm e a largura 220 nm. O valor máximo do comprimento e largura dos transistores foi arbitrado em 10 μm . Nesse caso, além das larguras W_f e comprimentos L dos transistores, são adicionadas variáveis de *fingers* N_f (variando de 1 até 24) e multiplicidade Par (variando de 1 até 5) para cada transistor. Os indutores tem variáveis para a dimensão lateral LX (100 μm até 450 μm), largura das trilhas LW (5 μm até 25 μm), espaço entre as voltas LS (2,8 μm até 5 μm) e número de voltas LN (0,5 até 6,5). As capacitâncias são alteradas através de suas dimensões laterais LC (0,85 μm até 100 μm). A resistência R_{bias} foi fixada em 12 k Ω e não considerada como uma variável de otimização.

As especificações do *Low Noise Amplifier* encontram-se na Tabela 4.9, juntamente com os melhores resultados obtidos por cada um dos algoritmos entre 10 sínteses para cada um. As especificações *desloc. 11*, *desloc. 22* e *desloc. 21* são restrições auxiliares para fazerem o ponto mínimo de S_{11} e S_{22} e o máximo de S_{21} estarem próximos da frequência especificada através de sua distância relativa. Nenhum algoritmo atendeu todas as especificações em 2000 iterações. No entanto, ignorando uma restrição auxiliar, o algoritmo proposto atendeu todas as demais. Os demais algoritmos tiveram dificuldade em obter valores próximos das especificações. A Tabela 4.10 mostra as dimensões finais dos dispositivos nos melhores circuitos sintetizados pelos algoritmos.

Os resultados da função custo final das sínteses para cada método estão expostos na Tabela 4.11. A melhor mediana foi obtida pelo algoritmo proposto (41,2), seguida pelos algoritmos PSWARM, CMA-ES, NSGA, GA POP100, GA POP50, MCS e ASA. O desempenho

Tabela 4.9: Melhores resultados dos algoritmos comparados para a síntese do circuito do LNA com até 2000 iterações.

Medida	Espec.	Tipo*	Alg. Proposto	ASA	CMA-ES	GA POP 50	GA POP 100	MCS	NSGA	PSWARM
NF [dB]	< 3	Obj.	1,47	8,22	1,87	10,63	5,68	12,98	3,45	1,42
S_{12} [dB]	< -30	Restr.	-43,22	-29,98	-45,91	-53,30	-52,47	-55,06	-52,68	-37,25
S_{22} [dB]	< -10	Restr.	-21,14	-3,72	-6,11	-13,55	-5,78	-3,87	-9,32	-14,24
S_{21} [dB]	> 20	Restr.	20,15	-4,34	13,40	-5,71	-0,49	-14,13	3,40	12,64
S_{11} [dB]	< -15	Restr.	-15,37	-5,78	-10,43	-6,10	-6,14	-8,86	-7,22	-5,69
desloc. 11 [%]	< 1	Restr.	1,61	0,29	10,15	7,50	0,73	22,52	16,08	16,18
desloc. 22 [%]	< 1	Restr.	0,96	14,79	102,36	22,01	11,19	0,27	1,89	7,68
desloc. 21 [%]	< 1	Restr.	0,14	6,80	7,46	1,70	45,51	0,18	0,20	0,47
Im(Zin) [Ω]	< 5	Restr.	4,87	33,57	3,28	1,69	8,96	6,50	6,43	4,71
Im(Zout) [Ω]	< 5	Restr.	3,33	109,79	3,20	22,73	29,56	21,83	31,44	4,69
Area [mm^2]	< 0,9	Obj.	0,26 1	0,21	0,26	0,19	0,16	0,22	0,12	0,26
Potência Consumida [mW]	< 10	Obj.	9,82	0,73	7,76	0,22	5,25	0,04	7,69	11,11

* Restr. = especificação do tipo “Restrição”, no qual a função custo não tem alterações após a especificação atendida;

Obj. = especificação do tipo “Objetivo”, no qual a função custo tem alterações mesmo após a especificação atendida.

Tabela 4.10: Dimensões dos melhores resultados obtidos pelos algoritmos comparados para a síntese do circuito do LNA com até 2000 iterações.

Dispositivo	Alg. Proposto	ASA	CMA-ES	GA POP 50	GA POP 100	MCS	NSGA	PSWARM
M1 (W_f $\mu m/L\mu m$)	9,2 / 0,2	8,0 / 9,8	4,5 / 0,4	1,6 / 4,0	3,4 / 0,9	0,2 / 8,7	8,9 / 0,5	6,5 / 0,2
M2 (W_f $\mu m/L\mu m$)	3,4 / 0,4	8,2 / 4,0	2,9 / 0,8	6,6 / 4,3	3,3 / 1,2	10,0 / 9,0	9,6 / 0,9	9,0 / 0,2
Nf_1	20	10	15	13	11	15	5	24
Nf_2	9	21	9	10	9	1	5	18
$Mult_1$	1	5	2	3	3	4	4	5
$Mult_2$	4	3	4	3	3	5	2	5
LC_m [μm]	54,4	43,4	86	50,6	89,4	96,2	71,1	70,2
LC_1 [μm]	10	64,9	10,4	7,47	38,5	91,7	85	33,6
LC_2 [μm]	13,7	3,13	18,5	11,4	14,1	0,85	12,4	23,4
Bias	0,63	0,615	0,722	0,587	0,817	0,587	0,702	0,513
LX_l [μm]	410	228	316	258	181	275	149	199
LX_g [μm]	270	189	339	191	319	262	270	449
LX_d [μm]	108	344	210	298	108	236	132	100
LW_l [μm]	5,66	5,43	11,5	14,8	5,77	18,4	11,1	25
LW_g [μm]	8,32	6,85	10,8	14,7	10,6	13,7	15,9	5
LW_d [μm]	14	15,3	14,3	10,5	18	22,9	24,4	20
LS_l [μm]	4,41	4,56	3,58	3,9	3,21	4,85	2,95	2,81
LS_g [μm]	4,46	3,37	3,19	3,86	4,68	2,8	3,26	2,8
LS_d [μm]	3,63	3,04	3,79	3,49	3,35	4,44	3,01	5
LN_l [volts]	2	2,5	2,5	4,5	2,5	4	5,5	2,5
LN_g [volts]	6,5	2,5	4	4	3,5	4	6	1,5
LN_d [volts]	1,5	5	1	2	3	2,5	2,5	6,5

não satisfatório que a maioria dos algoritmos obtiveram nesta síntese deve-se, em parte, a forte dependência desse projeto de variáveis discretas. O algoritmo proposto tem seu desempenho menos afetado por essas variáveis uma vez que tem mecanismo para aumentar ou diminuir a variabilidade de cada variável independentemente. Assim, ao defrontar-se com uma variável que não afeta significativamente a resposta (possivelmente por estar sendo arredondada sempre para o mesmo valor), o desvio padrão do passo dessa variável é aumentado, assim propiciando variação apropriada do seu valor.

Os resultados do teste de Wilcoxon-Mann-Whitney mostram que, exceto para os pares (ASA, MCS), (CMA-ES, NSGA), (CMA-ES, PSWARM), (GA POP100, GA POP50), (GA POP50, PSWARM) e (NSGA, PSWARM) todas as diferenças nas respostas são significativas. A partir dos resultados apresentados, é possível afirmar que o algoritmo proposto apresentou os

Tabela 4.11: Resultados dos algoritmos comparados para a síntese do LNA.

Síntese	Alg. Proposto	ASA	CMA-ES	GA POP100	GA POP50	MCS	NSGA	PSWARM
1	4,28e+01	4,55e+03	2,12e+03	3,30e+03	4,01e+03	4,60e+03	2,33e+03	3,79e+03
2	3,95e+01	5,15e+03	1,24e+02	3,12e+03	2,99e+03	4,60e+03	1,71e+03	4,12e+03
3	1,20e+00	5,58e+03	1,67e+03	3,60e+03	3,75e+03	4,60e+03	3,55e+03	8,49e+02
4	1,40e+00	5,58e+03	2,12e+03	3,14e+03	3,48e+03	4,60e+03	1,63e+03	2,49e+02
5	5,42e+01	4,55e+03	4,69e+03	3,56e+03	3,05e+03	4,60e+03	2,79e+03	1,82e+03
6	2,36e+01	5,58e+03	2,93e+03	4,20e+03	4,81e+03	4,60e+03	1,40e+03	2,14e+03
7	1,54e+02	4,55e+03	3,02e+03	3,72e+03	3,47e+03	4,60e+03	2,64e+03	1,38e+03
8	2,51e+02	5,17e+03	7,64e+02	1,90e+03	3,81e+03	4,60e+03	1,75e+03	1,36e+03
9	3,38e-02	5,58e+03	1,38e+03	4,41e+03	3,31e+03	4,60e+03	2,24e+03	4,10e+03
10	6,03e+01	5,58e+03	7,67e+02	4,10e+03	4,01e+03	4,60e+03	1,90e+03	1,83e+03
Média	6,28e+01	5,19e+03	1,96e+03	3,50e+03	3,67e+03	4,60e+03	2,19e+03	2,16e+03
Variância	6,41e+03	2,19e+05	1,80e+06	5,12e+05	2,90e+05	0,00e+00	4,31e+05	1,90e+06
Mediana	4,12e+01	5,37e+03	1,89e+03	3,58e+03	3,62e+03	4,60e+03	2,07e+03	1,83e+03
Mínimo	3,38e-02	4,55e+03	1,24e+02	1,90e+03	2,99e+03	4,60e+03	1,40e+03	2,49e+02
Máximo	2,51e+02	5,58e+03	4,69e+03	4,41e+03	4,81e+03	4,60e+03	3,55e+03	4,12e+03
Diferentes com $p < 0,05$ no teste de Wilcoxon-Mann-Whitney								
Algoritmos								
Alg. Proposto	-	Sim	Sim	Sim	Sim	Sim	Sim	Sim
ASA	Sim	-	Sim	Sim	Sim	Não	Sim	Sim
CMA-ES	Sim	Sim	-	Sim	Sim	Sim	Não	Não
GA POP100	Sim	Sim	Sim	-	Não	Sim	Sim	Sim
GA POP50	Sim	Sim	Sim	Não	-	Sim	Sim	Não
MCS	Sim	Não	Sim	Sim	Sim	-	Sim	Sim
NSGA	Sim	Sim	Não	Sim	Sim	Sim	-	Não
PSWARM	Sim	Sim	Não	Sim	Não	Sim	Não	-

melhores resultados para as sínteses do LNA do conjunto de testes realizado.

A Figura 4.18 mostra o decaimento da mediana da função custo ao longo do número de iterações. Até aproximadamente 250 iterações, a maior parte dos algoritmos tem um rápido decaimento na função custo. Durante estas iterações, o algoritmo proposto está fazendo a exploração aleatória do espaço de projeto para obter as soluções âncoras. Assim, logo o algoritmo proposto começa a decair sua melhor resposta, atinge soluções melhores que os demais.

Na seção 4.4.3 é realizada a síntese do LNA com o algoritmo proposto considerando linearidade e variações de processo, o que não foi realizado nessa seção uma vez que apenas visava a comparação entre algoritmos.

Figura 4.18: Melhor função custo em função do número de iterações para cada algoritmo na síntese do LNA (cada valor é resultado da mediana entre todas as sínteses para cada algoritmo).

4.3.5 Fonte de Referência

A topologia de fonte de referência utilizada neste trabalho é uma variação da topologia proposta por Huang et al. (2006). Ela foi utilizada por Mateus et al. (2008) e tem como principal característica o baixo consumo de potência. Utilizando uma corrente proporcional a temperatura absoluta I_{PTAT} , é gerada uma corrente I_{CTAT} para que, somada a I_{PTAT} , cancele o coeficiente térmico de primeira ordem e gere uma tensão independente da temperatura em um resistor R_R . Sua topologia pode ser vista na Figura 4.19.

Para a fonte de referência foi utilizada a tecnologia AMS 0,35 μm com uma tensão de alimentação de 1,25 V. O comprimento mínimo dos transistores é 0,35 μm e a largura 0,6 μm . O valor máximo do comprimento foi arbitrado em 50 vezes o mínimo e a largura em 300 vezes o mínimo. O número de *fingers* dos transistores pode variar entre 1 a 10, com exceção do transistor M8, que pode variar de 4 até 20. Os valores de resistência são atingidos utilizando dimensões de largura e comprimento do resistor. A tecnologia de 0,35 μm utilizada possui um módulo de alta resistividade com 1,2 $\text{k}\Omega/\square$ de resistência de folha (R_{sh}). A resistência final é aproximadamente $R = R_{sh} \cdot \frac{L}{W}$. As resistências podem variar de 10 $\text{k}\Omega$ até 500 $\text{k}\Omega$.

Para o funcionamento apropriado do circuito, foi definido que os transistores M2, M6 e M8 devem possuir coeficiente de inversão inferior a 0,1. As demais especificações da fonte de referência se encontram na Tabela 4.14, juntamente com os melhores resultados obtidos

Figura 4.19: Topologia da fonte de referência (variação da topologia de Huang et al. (2006) utilizada por Mateus et al. (2008)).

Tabela 4.12: Melhores resultados dos algoritmos comparados para a síntese do circuito da Fonte de Referência com até 2000 iterações.

Medida	Espec.	Tipo*	Alg. Proposto	ASA	CMA-ES	GA POP 50	GA POP 100	MCS	NSGA	PSWARM
Potência μ [W]	< 3,5	Obj.	2,73	2,56	3,40	2,69	2,28	2,68	4,32	3,03
Tensão aos 27° m[V]	177 $\pm$ 10%	Restr.	175,9	176,70	175,80	176,10	176,40	176,50	176,90	176,90
TC [ppm/°C]	< 8	Obj.	4,08	5,28	5,29	5,72	7,81	13,83	8,44	5,26
LR m[V/V]	< 6	Obj.	4,50	3,48	3,60	5,28	3,53	82,40	3,79	4,03
PSRR dB[V]	> 40	Obj.	43,08	35,99	38,34	33,54	35,16	21,38	38,89	40,09
Area [μ m] ²	< 40000	Obj.	5900	23700	25200	20500	24300	19900	18200	11900

* Restr. = especificação do tipo “Restrição”, no qual a função custo não tem alterações após a especificação atendida

Obj. = especificação do tipo “Objetivo”, no qual a função custo tem alterações mesmo após a especificação atendida

por cada um dos algoritmos entre 10 sínteses para cada um. É possível observar que tanto o algoritmo proposto quanto o PSWARM encontraram soluções dentro das especificações. As especificações obtidas pelo algoritmo proposto, no entanto, alcançam as especificações com uma margem maior. Os demais algoritmos, com exceção do MCS, conseguiram boas soluções, sendo que apenas o PSRR especificado não foi alcançado por esses algoritmos.

A Tabela 4.13 mostra as dimensões finais dos dispositivos do melhor circuito da fonte de referência sintetizado por cada algoritmo.

O resultados da função custo final das sínteses para cada método estão expostos na Tabela 4.14. A melhor mediana foi obtida pelo algoritmo proposto, seguida pelos algoritmos PSWARM, ASA, CMA-ES, GA POP100, GA POP50, NSGA e MCS. Os resultados do teste de Wilcoxon-Mann-Whitney mostram que, exceto para os pares (ASA, CMA-ES), (ASA, GA POP50), (ASA, GA POP100), (ASA, NSGA), (ASA, PSWARM), (CMA-ES, GA POP100), (GA POP100, GA POP50), (GA POP100, NSGA) e (GA POP50, NSGA), todas as diferenças nas respostas são significativas. Assim, o algoritmo proposto tem resposta diferente estatisti-

Tabela 4.13: Dimensões dos melhores resultados obtidos pelos algoritmos comparados para a síntese do circuito da Fonte de Referência com até 2000 iterações.

Disp.*	Alg. Proposto	ASA	CMA-ES	GA POP 50	GA POP 100	MCS	NSGA	PSWARM
M1	12,9(x2) / 14,8	68,4(x6) / 10,7	51,0(x4) / 9,0	33,0(x5) / 16,7	25,9(x4) / 15,9	88,3(x10) / 0,3	6,2(x4) / 16,0	4,5(x3) / 11,6
M2	16,4(x2) / 1,3	44,5 / 1,3	75,2(x5) / 2,0	98,1(x5) / 5,2	53,9(x3) / 13,5	60,0(x6) / 8,0	93,2(x6) / 7,4	33,5(x3) / 5,1
M3	33,7(x3) / 5,0	103,0(x3) / 8,8	110,0(x9) / 6,5	32,4(x7) / 14,1	65,5(x8) / 3,4	53,6(x6) / 13,1	61,5(x8) / 4,3	49,0(x3) / 17,4
M4	19,3(x2) / 1,3	47,4(x3) / 1,3	14,5(x4) / 2,0	89,4(x4) / 5,2	54,3(x8) / 13,5	0,4(x9) / 8,0	25,6(x6) / 7,4	19,6(x2) / 5,1
M5	12,9 / 14,8	68,4(x5) / 10,7	51,0(x3) / 9,0	33,0 / 16,7	25,9(x4) / 15,9	88,3(x6) / 0,3	6,2(x3) / 16,0	4,5(x3) / 11,6
M6	13,9(x2) / 4,1	69,3 / 1,3	14,2 / 1,3	18,2(x5) / 4,2	23,5(x2) / 4,1	51,2(x3) / 3,8	53,4(x2) / 8,5	19,2(x3) / 4,6
M7	12,9(x2) / 14,8	68,4(x5) / 10,7	51,0(x3) / 9,0	33,0 / 16,7	25,9(x4) / 15,9	88,3(x6) / 0,3	6,2(x3) / 16,0	4,5(x3) / 11,6
M8	13,9(x7) / 4,1	69,3(x10) / 1,3	14,2(x18) / 1,3	18,2(x11) / 4,2	23,5(x14) / 4,1	51,2(x20) / 3,8	53,4(x7) / 8,5	19,2(x8) / 4,6
M9	33,7(x2) / 5,0	103,0 / 8,8	110,0(x8) / 6,5	32,4(x4) / 14,1	65,5(x3) / 3,4	53,6(x6) / 13,1	61,5(x7) / 4,3	49,0(x3) / 17,4
M10	12,9(x2) / 14,8	68,4(x3) / 10,7	51,0(x8) / 9,0	33,0(x4) / 16,7	25,9(x8) / 15,9	88,3(x6) / 0,3	6,2(x10) / 16,0	4,5(x10) / 11,6
R1	475,0 k Ω	493,0 k Ω	452,0 k Ω	446,0 k Ω	415,0 k Ω	500,0 k Ω	332,0 k Ω	5,0 k Ω
R2	231,0 k Ω	176,0 k Ω	281,0 k Ω	199,0 k Ω	357,0 k Ω	250,0 k Ω	125,0 k Ω	139,0 k Ω
R3	166,0 k Ω	357,0 k Ω	121,0 k Ω	194,0 k Ω	267,0 k Ω	155,0 k Ω	91,9 k Ω	121,0 k Ω

* Dispositivo, no qual os transistores têm a largura (W) e comprimento (L) em μm . Os valores são expressos no formato $W(xf)/L$, onde f é um fator multiplicativo representando o número de *fingers*.

Tabela 4.14: Resultados dos algoritmos comparados para a síntese do circuito fonte de referência.

Síntese	Alg. Proposto	ASA	CMA-ES	GA POP100	GA POP50	MCS	NSGA	PSWARM
1	0,00e+00	1,63e+01	1,27e+02	4,25e+01	6,08e+01	1,49e+03	5,67e+00	3,90e+00
2	0,00e+00	1,63e+01	7,07e+00	1,21e+01	2,70e+01	1,49e+03	1,29e+02	2,31e+01
3	0,00e+00	1,14e+02	3,88e+01	1,31e+03	2,14e+01	1,49e+03	4,47e+02	6,38e+01
4	0,00e+00	1,63e+01	1,59e+01	2,96e+01	1,01e+02	1,49e+03	2,07e+02	1,03e+01
5	0,00e+00	2,05e+02	1,36e+01	2,22e+01	1,64e+02	1,49e+03	1,88e+02	1,06e+01
6	0,00e+00	1,00e+01	1,83e+01	2,29e+01	1,62e+01	1,49e+03	8,30e+01	1,35e+01
7	7,14e-02	5,32e+02	2,46e+01	7,40e+01	3,58e+02	1,49e+03	4,10e+01	1,72e+01
8	0,00e+00	1,63e+01	1,63e+01	2,34e+01	5,95e+01	1,49e+03	2,97e+02	0,00e+00
9	0,00e+00	1,00e+01	2,36e+01	4,15e+01	2,65e+02	1,49e+03	2,13e+01	0,00e+00
10	6,27e+00	1,63e+01	1,98e+01	1,18e+02	2,87e+01	1,49e+03	1,64e+01	0,00e+00
Média	6,34e-01	9,53e+01	3,05e+01	1,69e+02	1,10e+02	1,49e+03	1,44e+02	1,42e+01
Variância	3,92e+00	2,77e+04	1,23e+03	1,61e+05	1,38e+04	0,00e+00	2,07e+04	3,66e+02
Mediana	0,00e+00	1,63e+01	1,90e+01	3,55e+01	6,01e+01	1,49e+03	1,06e+02	1,05e+01
Mínimo	0,00e+00	1,00e+01	7,07e+00	1,21e+01	1,62e+01	1,49e+03	5,67e+00	0,00e+00
Máximo	6,27e+00	5,32e+02	1,27e+02	1,31e+03	3,58e+02	1,49e+03	4,47e+02	6,38e+01
Diferentes com $p < 0,05$ no teste de Wilcoxon-Mann-Whitney								
Alg. Proposto	-	Sim	Sim	Sim	Sim	Sim	Sim	Sim
ASA	Sim	-	Não	Não	Não	Sim	Não	Não
CMA-ES	Sim	Não	-	Não	Sim	Sim	Sim	Sim
GA POP100	Sim	Não	Não	-	Não	Sim	Não	Sim
GA POP50	Sim	Não	Sim	Não	-	Sim	Não	Sim
MCS	Sim	Sim	Sim	Sim	Sim	-	Sim	Sim
NSGA	Sim	Não	Sim	Não	Não	Sim	-	Sim
PSWARM	Sim	Não	Sim	Sim	Sim	Sim	Sim	-

camente que todos os demais algoritmos. O algoritmo proposto atingiu as especificações em 8 das 10 sínteses, o PSWARM atingiu as especificações em 3 das 10 sínteses e os demais não atingiram as especificações em nenhuma síntese.

A partir dos resultados apresentados, é possível afirmar que o algoritmo proposto apresentou os melhores resultados para as sínteses da fonte de referência do conjunto de testes realizado.

A Figura 4.20 mostra o decaimento da mediana da função custo ao longo do número de iterações. O PSWARM tem rápido decaimento, porém a mediana da função custo estabiliza em torno do valor 10. A mediana do algoritmo proposto decai para menos que 0 após aproximadamente 1400 iterações. Os demais algoritmos apresentam decaimentos lentos e não atingem

Figura 4.20: Melhor função custo em função do número de iterações para cada algoritmo na síntese da fonte de referência (cada valor é resultado da mediana entre todas as sínteses para cada algoritmo).

mediana zero ao longo do total de iterações do teste.

4.4 Comparação com Projetos Manuais e Outras Técnicas de Otimização

A comparação de técnicas de otimização com resultados de projetos manuais é interessante para avaliar a capacidade da ferramenta em sintetizar circuitos com especificações similares às já obtidas por projetistas. Para avaliar o algoritmo proposto, também é interessante compará-lo com circuitos sintetizados por outras técnicas de otimização.

4.4.1 Amplificador Miller

Neste teste, o amplificador Miller é sintetizado e comparado com um projeto manual previamente publicado na literatura na tecnologia AMS 0,35 μm . A síntese é limitada em 5000 iterações e é realizada em um computador com processador Intel(R) Core(TM) i7-4510M com frequência de 2,0 GHz e 8 GB de memória RAM.

O projeto usado para comparação é o realizado por Cortes e Bampi (2006) e utiliza técnica de projeto g_m/I_D . A topologia utilizada é a do amplificador Miller já apresentado neste trabalho

Tabela 4.15: Resultados da síntese para o amplificador Miller em comparação com resultado manual realizado por Cortes e Bampi (2006).

Medida	Especificação ^{1 2}	Tipo ³	Algoritmo Proposto ²	Resultado de Projeto Manual ⁴
UGF M[Hz]	> 15	Restrição	17,5	15,2
Corrente μ [A]	< 325,5	Objetivo	219,4	339,4 (simulado) 259,9 (medido)
Margem de Fase [°]	> 60 e < 90	Restrição	60,2	60,1
Ganho DC dB[V]	> 80	Objetivo	80,0	85,4
Slew Rate [V/ μ s]	> 18	Restrição	18	9
ICMR [V]	> 1,5	Restrição	2,0	2,0
Excursão Máxima de Saída [V]	> 2	Restrição	2,8	2,5
CMRR dB[V]	> 89	Restrição	91,3	89,9
PSRR dB[V]	> 70	Restrição	82,0	(não disponível)
Area de Porta [μ m] ²	< 1000	Objetivo	150	929

¹ Para as especificações não explicitadas na referência, foram utilizados os valores próximos de seus resultados como especificação;

² Para capacitância de carga de 10 pF;

³ Especificação tipo “Restrição”: função custo não tem alterações após a especificação atendida; Especificação tipo “Objetivo”: função custo tem alterações mesmo após a especificação atendida;

⁴ (CORTES; BAMPI, 2006), simulado para capacitância de carga de 20 pF.

Tabela 4.16: Dimensões dos transistores MOS e parâmetros para o circuito amplificador Miller encontrado na literatura e deste trabalho sem análise de *corners*.

Transistor	Valores de projeto manual utilizando técnica g_m/I_D (CORTES; BAMPI, 2006)		Valores deste trabalho sem análise de <i>corners</i>	
	W [μ m]	L [μ m]	W [μ m]	L [μ m]
M_1	108	3	10,3 (x2)	1,06
M_2	108	3	10,3 (x2)	1,06
M_3	6	0,6	5,68 (x2)	0,91
M_4	6	0,6	5,68 (x2)	0,91
M_5	30	1	6,22	0,72
M_6	158	1	6,22 (x8)	0,72
M_7	62	0,6	13,9 (x7)	0,40
M_8	30	1,5	6,22	0,72
M_9	1	4	0,60	3,88
Parâmetro	Valor		Valor	
Tecnologia	0,35 μ m		0,35 μ m	
($V_{DD} - V_{SS}$)	3,3 V		3,3 V	
C_c	2,5 p[F]		1 p[F]	

(Figura 4.12) com a substituição da fonte de corrente I_{bias} por um transistor NMOS (M_9) em conexão diodo.

A síntese foi realizada em um total de 1 hora e 10 minutos. Na Tabela 4.15 podem ser vistas as especificações e os resultados obtidos para o algoritmo proposto e pelo trabalho da referência.

As especificações mínimas foram atingidas pelo algoritmo proposto e, quando em comparação com as especificações obtidas pelo projeto manual, os resultados mostram-se favoráveis ao uso do algoritmo. As dimensões resultantes da síntese podem ser vistas na Tabela 4.16.

4.4.2 Síntese de Amplificador Miller considerando *Corners* de Processo

Conforme descrito na seção 4.2, para garantir a robustez do projeto pode ser interessante realizar a síntese considerando não apenas as condições típicas, mas também os piores casos de processo. Assim, é realizada a síntese do amplificador Miller do caso anterior utilizando modelos com as seguintes casos: condição média típica (TM), condição de pior velocidade (WS), condição de pior um (WO), condição de pior zero (WZ) e condição de pior potência consumida (WP). Além da adição das simulações com estes modelos adicionais, o valor das especificações de área e corrente foram removidas para que o algoritmo tentasse minimizá-las mas não em detrimento de outras especificações.

A síntese foi realizada em um total de 3 horas e 49 minutos. Os resultados podem ser vistos na Tabela 4.17 e a Tabela 4.18 mostra as dimensões resultantes da síntese. As grandes dimensões dos transistores são um indicativo de que realizar esta síntese considerando todos os *corners* resulta em um provável sobre-projeto.

Em relação a solução obtida previamente para a síntese apenas nominal, é possível observar o significativo aumento na corrente consumida e na área. Através do incremento destas especificações, a solução final obteve sucesso em garantir o funcionamento do circuito dentro das especificações para as variações de processo.

Tabela 4.17: Resultados de síntese para o amplificador Miller em diversos *corners*.

Medida	Especificação	Tipo*	TM	WS	WP	WO	WZ
UGF M[Hz]	> 15	Restrição	29,27	21,05	43,93	31,75	29,53
Corrente μ [A]	minimizar	Objetivo	1120	796	1754	1337	1073
Margem de Fase [°]	> 60 e < 90	Restrição	63,6	62,3	64,8	67,0	60,0
Ganho DC dB[V]	> 80	Objetivo	86,7	86,6	81,4	80,1	87,1
<i>Slew Rate</i> [V/ μ s]	> 18	Restrição	24,5	18,7	39,3	28,8	23,7
ICMR [V]	> 1.5	Restrição	2,26	2,21	2,35	2,13	2,45
Excursão Máxima de Saída [V]	> 2	Restrição	2,8	2,8	2,8	2,8	2,8
CMRR dB[V]	> 89	Restrição	100,0	106,4	93,2	93,5	108,8
PSRR dB[V]	> 70	Restrição	87,0	91,7	80,5	78,9	92,2
Área de Portas [μ m] ²	minimizar	Objetivo	4653				

* Especificação do tipo “Restrição”: função custo não tem alterações após a especificação atendida;
Especificação do tipo “Objetivo”: função custo tem alterações mesmo após a especificação atendida.

4.4.3 Amplificador de Baixo Ruído (LNA)

Nesta seção é realizada a síntese de um LNA de banda estreita e realizada a comparação dos resultados com diversos similares encontrados na literatura. Os resultados obtidos nesta seção foram publicados em (WEBER et al., 2013).

Para este caso, duas etapas de síntese são realizadas. Inicialmente, o sintetizador não con-

Tabela 4.18: Dimensões dos transistores MOS e parâmetros para o circuito amplificador Miller encontrado na literatura e neste trabalho considerando *corners* do processo CMOS 0,35 μm .

Transistor	Valores da técnica g_m/I_d (CORTES; BAMPI, 2006)		Valores deste trabalho com consideração de <i>corners</i>	
	W [μm]	L [μm]	W [μm]	L [μm]
M_1	108	3	52,9 (x4)	1,74
M_2	108	3	52,9 (x4)	1,74
M_3	6	0,6	19,4	0,62
M_4	6	0,6	19,4	0,62
M_5	30	1	91,7	2,04
M_6	158	1	91,7 (x9)	2,04
M_7	62	0,6	39,8 (x7)	0,47
M_8	30	1,5	91,7 (x10)	2,04
M_9	1	4	91,7 (x3)	2,04
Parâmetro	Valor		Valor	
Tecnologia	0,35 μm		0,35 μm	
$(V_{DD} - V_{SS})$	3,3 V		3,3 V	
C_c	2,5 p[F]		2,23 p[F]	

sidera linearidade como um objetivo de projeto. As especificações são $NF < 3$ dB, $S_{12} < -30$ dB, $S_{22} < -10$ dB, $S_{21} > 12$ dB, $S_{11} < -15$ dB e potência < 10 mW. A segunda etapa de síntese começa a partir da melhor resposta encontrada pela primeira e adiciona a especificação de linearidade $P_{IIP3} > -10\text{dBm}$. A razão pela qual a especificação de linearidade não é utilizada durante a primeira síntese é que a análise *steady-state harmonic balance* necessita de um tempo consideravelmente maior do que todos os outros *testbenches*. As sínteses do LNA foram realizadas utilizando um processador Intel(R) Core(TM) i5-2410M com frequência de 2,3 GHz e 4 GB de memória RAM.

O LNA tem frequência central em 2,45 GHz e foi sintetizado na tecnologia IBM 180 nm com uma alimentação de 1,8 V. O dispositivo é casado com $50\ \Omega$ nas portas de entrada e saída. A Tabela 4.19 mostra os possíveis valores das variáveis do LNA e os melhores valores encontrados pelo sintetizador nas duas sínteses. Nesta Tabela, N_f é o número de *fingers* da porta, W_f é a largura de cada *finger*, Par é o número de instâncias em paralelo do transistor, L é o comprimento do canal, LC é a dimensão lateral dos capacitores quadrados, $Bias$ é a tensão de polarização do circuito, LX é a dimensão lateral do indutor quadrado, LW é a largura das trilhas do indutor, LS é o espaço entre voltas do indutor, e LN é o número de voltas do indutor. A Tabela 4.20 mostra os valores resultantes para os indutores e capacitores. A resistência R_{bias} foi fixada em 12 k Ω e não foi considerada como uma variável de otimização.

A síntese da primeira solução foi concluída após 2 horas e 58 minutos, nas quais 1074 iterações foram realizadas. Após a iteração número 812, as simulações de *corners* de processo foram habilitadas devido a proximidade da obtenção do custo zero, que representa que o cumprimento de todas as especificações mínimas. A segunda solução foi encontrada após 1 hora e 3

Tabela 4.19: Intervalo das variáveis e valores encontrados.

Variável	Intervalo	Valor Encontrado para 1ª Solução	Valor Encontrado para 2ª Solução	Unidade
Wf_1	[0,22 10]		3,90	μm
Wf_2	[0,22 10]	9,37	8,94	μm
L_1	[0,18 10]		0,32	μm
L_2	[0,18 10]	0,61	0,58	μm
LC_{in}	[0,85 100]		47,9	μm
LC_1^*	[0,85 100]	10,2	10,7	μm
LC_2^*	[0,85 100]	17,7	16,5	μm
Nf_1	[1 24]		20	adim
Nf_2	[1 24]		11	adim
$Bias$	[0,1 1]	0,62	0,60	V
Par_1	[1 5]		4	adim
Par_2	[1 5]		2	adim
LX_l	[100 450]		255	μm
LX_g	[100 450]		186	μm
LX_d	[100 450]	104	132	μm
LW_l	[5 25]		9,41	μm
LW_g	[5 25]		5,20	μm
LW_d	[5 25]	5,47	5,70	μm
LS_l	[2,8 5]		3,13	μm
LS_g	[2,8 5]	4,46	4,24	μm
LS_d	[2,8 5]	3,13	3,10	μm
LN_l	[0,5 6,5]		3	μm
LN_g	[0,5 6,5]		6	μm
LN_d	[0,5 6,5]		2	μm

* Capacitores C_1 e C_2 tem multiplicidade 4.

Tabela 4.20: Capacitâncias e indutâncias resultantes.

Elemento	Valor para 1ª Solução	Valor para 2ª Solução	Unidade
C_{in}		6,2	pF
C_1	1,12	1,24	pF
C_2	3,38	2,94	pF
L_d		3,45	nH
L_g	5,54	5,66	nH
L_s	0,65	0,89	nH

minutos. Para esta solução, foram realizadas 80 iterações incluindo o *testbench* de linearidade e simulações de *corners*.

Duas figuras de mérito são utilizadas para comparar as soluções, FoM 1 (LINTEN et al., 2004) e FoM 2 (BREDELOW et al., 2001):

$$FoM1 = \frac{Ganho[abs]}{(NF - 1)[abs] \cdot P_{dc}[mW]} \quad (4.8)$$

Tabela 4.21: Métricas de desempenho simuladas em comparação com LNAs publicados na literaturas.

Parâmetro	Trabalho 1*	Trabalho 2*	Trabalho 3*	Este Trabalho 1 ^a Solução	Este Trabalho 2 ^a Solução
Tecnologia CMOS (μm)	0,35	0,35	0,18	0,18	0,18
Frequência (GHz)	2,2	2,45	2,57	2,45	2,45
NF(dB)	1,87	4,08	3,5	1,95	2,04
S_{21} (dB)	10 (20,4**)	8,5 (23**)	16,6	13,6	12,9
S_{11} (dB)	-13	-34,6	-17,1***	-17,0	-25,4
Corrente (mA)	9	1,77	7	4,6	4,1
Tensão de Alimentação (V)	1,8	3,3	1,8	1,8	1,8
P_{IIP3} (dBm)	-2,05	0,3	-	-10,5	-7,8
P_{DC} (mW)	16,2	5,84	12,6	8,3	7,4
FoM 1	0,36	0,29	0,43	1,02	0,99
FoM 2	0,23	0,31	-	0,09	0,17

* Trabalho 1: (FAN et al., 2008); Trabalho 2: (CHAPARRO et al., 2009); Trabalho 3: (LIN et al., 2009);

** Ganho de tensão sem o casamento de 50Ω na saída;

*** Esta medida foi realizada em um dispositivo fabricado.

$$FoM2 = \frac{Ganho[abs] \cdot P_{IIP3}[mW]}{P_{dc}[mW] \cdot (NF - 1)[abs]} \quad (4.9)$$

As métricas de desempenho das melhores soluções podem ser vistas na Tabela 4.21 em comparação com outros resultados encontrados na literatura. A primeira solução, sem o *test-bench* de linearidade, obteve o melhor FoM 1 em comparação com outras soluções, mas classificou-se mal em relação a FoM 2. Após esta solução ter sido otimizada incluindo o *test-bench* de linearidade para gerar a segunda solução, ela continuou tendo bom desempenho na FoM 1 e melhorou o seu FoM 2.

Os parâmetros de desempenho para diversos *corners* de processo podem ser vistos na Tabela 4.22. Os *corners* são TT (NMOS típico, PMOS típico), FS (NMOS rápido, PMOS lento), SF (NMOS lento, PMOS rápido), SS (NMOS lento, PMOS lento) e FF (NMOS rápido, PMOS rápido). Pode-se notar que o desempenho é inferior ao especificado na solução 2 apenas para o S_{21} nos *corners* SS e SF. Ainda assim, o S_{21} nestes casos está próximo da especificação (12 dB), de forma que é possível observar o otimizador foi efetivo em aproximar as soluções das especificações desejadas.

Os parâmetros S das soluções finais podem ser vistos na Figura 4.21. Os resultados de simulação indicam eficiência do algoritmo proposto na síntese de LNAs de banda estreita considerando *corners* de processo.

Em suma, após 2 horas e 58 minutos, o primeiro LNA sintetizado atingiu 1,95 dB de Figura de Ruído, 13,6 dB de S_{21} , -17 dB de S_{11} consumindo 4,6 mA de corrente. A segunda solução

Tabela 4.22: Métricas de desempenho do LNA considerando *corners* de processo e temperatura para as soluções encontradas.

Parâmetro	TT 27°C	FF 27°C	SS 27°C	SF 27°C	FS 27°C	TT 10°C	TT 60°C
1ª Solução							
NF(dB)	1,95	1,72	2,29	2,23	1,77	1,60	2,29
S_{21} (dB)	13,59	15,25	11,64	12,00	14,88	14,62	12,62
S_{11} (dB)	-17,01	-20,26	-14,93	-15,14	-19,44	-16,46	-17,43
P_{IIP3} (dBm)	-10,49	-11,55	-8,05	-8,68	-11,35	-10,96	-9,55
Corrente (mA)	4,61	6,74	3,03	3,25	6,18	4,07	4,97
2ª Solução							
NF(dB)	2,04	1,79	2,41	2,34	1,84	1,68	2,39
S_{21} (dB)	12,89	14,51	10,96	11,31	14,16	13,80	12,03
S_{11} (dB)	-25,39	-33,89	-19,64	-20,16	-34,54	-24,66	-25,67
P_{IIP3} (dBm)	-7,8	-9,85	-3	-4,1	-9,5	-8,6	-6,5
Corrente (mA)	4,11	6,08	2,66	2,86	5,57	3,58	4,48

Figura 4.21: Curvas simuladas dos parâmetros S em função da frequência para os resultados da síntese do LNA: (a) S_{11} (b) S_{21} (c) S_{12} (d) S_{22} .

foi encontrada após a inclusão do *testbench* de linearidade e demorou 1 hora e 3 minutos. A resposta da segunda solução foi 2,04 dB de Figura de Ruído, 12,89 dB de S_{21} , -25 de S_{11} , -7.8 de P_{IIP3} consumindo 4,1 mA de corrente. As comparações com trabalhos da literatura em LNAs semelhantes mostram que as soluções geradas são comparáveis em desempenho enquanto

demandam um esforço de preparação menor.

4.4.4 Fonte de Referência

A mesma topologia de fonte de referência utilizada para os testes anteriores (Figura 4.19) é utilizada nesta seção. Diferentemente do teste anterior, no entanto, neste caso é realizada a comparação com resultados obtidos por uma técnica baseada em programação geométrica (NOVAL, 2012).

As especificações a serem atingidas e os resultados da busca global podem ser vistos na Tabela 4.23. A solução alcançada pelo presente trabalho conseguiu otimizar as diversas especificações simultaneamente, mantendo-as todas dentro das especificações utilizadas como referência. Nesta mesma tabela, encontram-se resultados apresentados por Noval (2012) para este circuito. As duas técnicas obtiveram bons resultados. Nas especificações de potência, PSSR e regulação de linha, a PG teve melhores resultados. Por outro lado, nas especificações de compensação de temperatura e área de portas a solução da ferramenta deste trabalho teve as melhores respostas. A vantagem mais evidente da utilização de uma ferramenta tal como a utilizada neste trabalho é a capacidade de atingir seus resultados sem necessidade de modelamento prévio do circuito e tampouco de um projeto inicial funcional. Por outro lado, enquanto o tempo de síntese da ferramenta deste trabalho foi de 2 horas e 26 minutos, os resultados obtidos usando programação geométrica costumam ser produzidos em minutos pois realizam avaliação baseada em equações.

As dimensões dos dispositivos alcançados após a síntese ser concluída pode ser vistas na Tabela 4.24.

4.4.5 Amplificador Folded Cascode

Na próxima comparação, é utilizada como referência a ferramenta *GENOM* (BARROS et al., 2010). Esta ferramenta foi desenvolvida para síntese de circuitos integrados analógicos através do uso de um algoritmo genético modificado. Este algoritmo tem seu operador de seleção guiado por um algoritmo de aprendizado automático baseado em *support vector machines* (SVMs).

Barros et al. (2010) fazem uma comparação entre a ferramenta *GENOM* e a *FRIDGE*. A topologia sob teste é amplificador *folded cascode* (Figura 4.22). A tensão de alimentação é +0.9 e -0.9V. A tecnologia utilizada no trabalho original foi a UMC 0,18 μm , mas para a comparação foi utilizada a IBM 0,18 μm devido a indisponibilidade da primeira. A capacitância de saída nos *testbenches* é de 3 pF. Para todas as ferramentas, foram utilizadas as mesmas variáveis e

Tabela 4.23: Especificações da fonte de referência e resultados encontrados neste trabalho e em trabalho utilizando programação geométrica .

Medida	Especificações deste trabalho	Tipo*	Resultados deste trabalho	Resultados em (NOVAL, 2012)
Potência $\mu W > \mu W$	< 4	Objetivo	3,48	2,94
Tensão de Saída à 27°C [mV]	≈ 177	Restrição	177,6	177,6
Compensação de Temperatura [ppm/°C] (de -20 °C até 80 °C)	< 8	Objetivo	3,13	6,96
Ruído de Saída [$\mu V/Hz$] (medido em 10 Hz)	< 2	Restrição	1,04	1,68
Regulação de Linha [mV/V]	< 6	Objetivo	5,37	4,39
PSSR dB[V] (medido em 5 kHz)	> 40	Objetivo	41,03	44,74
Area de Portas [mm] ²	$< 0,01$	Objetivo	0,005	0,009

* Especificação do tipo “Restrição”: função custo não tem alterações após a especificação atendida.

Especificação do tipo “Objetivo”: função custo tem alterações mesmo após a especificação atendida.

Tabela 4.24: Dimensões dos transistores MOS e resistores para o circuito fonte de referência.

Transistor/Resistor	W [μm]	L [μm]
M_1	5,92 (x5)	7,61
M_2	5,78 (x10)	7,46
M_3	14,1 (x6)	3,74
M_4	1,98 (x3)	14,9
M_5	5,92 (x2)	7,61
M_6	2,56 (x7)	5,07
M_7	5,92 (x2)	7,61
M_8	2,56 (x15)	5,07
M_9	14,1 (x3)	3,74
M_{10}	5,92 (x2)	7,61
R1	2	568
R2	2	325
R3	2	576

intervalos de variação. A Tabela 4.25 mostra estes intervalos e os valores finais para um máximo de 2500 iterações para cada ferramenta.

As especificações do projeto estão na Tabela 4.26, juntamente com os resultados atingidos por todas as ferramentas em até 2500 iterações. A ferramenta proposta levou 23 minutos para realizar todas as iterações em um computador Intel(R) Core(TM)2 CPU com uma frequência de 1,8 GHz e uma memória RAM de 3GB. As três ferramentas foram capazes de encontrar respostas que atendem as restrições. Dessa forma, compara-se os valores de area e potência, que são os objetivos da otimização. A ferramenta proposta obteve uma area final de $106,8 \mu m^2$, enquanto FRIDGE obteve $561,32 \mu m^2$ e GENOM $284,5 \mu m^2$. A area obtida pela ferramenta proposta é aproximadamente 2,7 vezes menor que a obtida pela ferramenta GENOM, e 5,3 vezes menor que a FRIDGE. Quanto a potência, a solução da ferramenta proposta consome 112

Figura 4.22: Topologia do *folded cascode* utilizado nos testes de Barros et al. (2010)

Tabela 4.25: Dimensões finais das variáveis para o *folded cascode* para cada ferramenta após um máximo de 2500 iterações. Valores das ferramentas FRIDGE e GENOM apresentados por Barros et al. (2010).

Variável	Intervalo	FRIDGE	GENOM	Ferramenta Proposta neste Trabalho
cn (V)	[-0,4 0]	-8,75E-2	-4,49E-2	0
cp (V)	[0 0,4]	6,24E-2	1,00E-3	0
l1 (m)	[0,18μ 5μ]	1,56E-6	1,38E-6	1.01E-6
l4 (m)	[0,18μ 5μ]	4,7E-7	1,94E-6	3.13E-7
l5 (m)	[0,18μ 5μ]	3,8E-7	3,70E-7	5.29E-7
l7 (m)	[0,18μ 5μ]	7,6E-7	9,10E-7	5.29E-7
l9 (m)	[0,18μ 5μ]	2,06E-6	8,90E-7	2.58E-7
l11 (m)	[0,18μ 5μ]	6,00E-7	2,19E-6	9.57E-7
ib (A)	[30μ 400μ]	4,84E-5	4,85E-5	3.2E-5
w1 (m)	[0,24μ 200μ]	1,95E-5	1,49E-5	1.26E-5
w4 (m)	[0,24μ 200μ]	3,03E-5	6,99E-6	5.3E-6
w5 (m)	[0,24μ 200μ]	7,13E-5	3,68E-5	7.98E-6
w7 (m)	[0,24μ 200μ]	1,04E-4	6,30E-5	4.74E-5
w9 (m)	[0,24μ 200μ]	6,56E-5	3,14E-5	3.53E-5
w11 (m)	[0,24μ 200μ]	3,08E-6	7,32E-6	1.59E-6

μW, enquanto tanto a solução da FRIDGE quanto da GENOM consomem aproximadamente 174 μW.

Tabela 4.26: Especificações do projeto do amplificador *folded cascode* e resultados expostos por Barros et al. (2010) e os alcançados por este trabalho.

Medida	Especificação ¹	FRIDGE	GENOM	Ferramenta Proposta neste Trabalho
<i>Gain Bandwidth</i>	> 12 MHz	16 MHz	15,3 MHz	13,2 MHz
Ganho	> 70 dB	70 dB	70,6 dB	70,0 dB
Margem de Fase	> 55 graus	80,6	79,6	86
<i>Slew Rate</i>	> 10 V/ μ s	15,3 V/ μ s	15,4 V/ μ s	10,1 V/ μ s
dm2 ²	> 1,2	9,78	9,25	10,16
dm4 ²	> 1,2	5,20	1,57	4,04
dm5 ²	> 1,2	2,21	1,84	1,58
dm7 ²	> 1,2	10,5	8,17	10,3
dm9 ²	> 1,2	3,05	2,81	4,81
dm11 ²	> 1,2	1,95	1,65	1,73
onm2 ²	> 0,1	0,104	0,11	0,10
onm4 ²	> 0,03	0,03	0,32	0,12
onm5 ²	> 0,03	0,06	0,10	0,26
onm7 ²	> 0,03	0,04	0,09	0,04
onm9 ²	> 0,03	0,05	0,04	0,03
onm11 ²	> 0,03	0,18	0,25	0,29
osp ²	> 0,5	0,63	0,57	0,52
osn ²	< 0,5	-0,50	-0,51	-0,53
Area	minimizar	561,32 [μ m] ² [³]	284,5 [μ m] ² [³]	106,8 [μm]²
Potência	minimizar	174,24 μ W [⁴]	174,60 μ W [⁴]	112,6 μW

¹ Com exceção da area e potência, todas as especificações são do tipo “restrição”;

² Restrições descritas em (BARROS et al., 2010);

³ Calculados a partir dos valores de W e L apresentados por Barros et al. (2010) sem considerar circuito de polarização;

⁴ Calculados a partir do valor de I_{bias} apresentados por Barros et al. (2010) sem considerar circuito de polarização.

A primeira solução da ferramenta GENOM que atende as restrições de projeto ocorreu na iteração 1110. A ferramenta proposta obteve a primeira solução com restrições atendidas na iteração 248 (após 137 segundos). O número de iterações reduzido que a ferramenta proposta necessita para atingir bons resultados é evidenciado nesta comparação, bem como sua capacidade de atingir respostas melhores que as ferramentas comparadas para o *testbench* em questão. Por outro lado, a ferramenta GENOM realiza uma modelagem que permite que avalie as soluções mais rapidamente (uma vez realizada a modelagem, toda a síntese usando o GENOM ocorre em poucos minutos).

4.5 Projeto de Fonte de Referência em 130 nm com Variações de Topologia e com Ajuste Digital

Fontes de referência utilizando transistores CMOS operando na região sublimiar são úteis para o desenvolvimento de sistemas com baixo consumo de potência. Esse tipo de fonte de referência tem a tensão de saída inerentemente dependente do processo de fabricação devido à variação da tensão de limiar dos transistores. Nesta seção é realizada a síntese de uma fonte de referência considerando variações de processo e ajuste digital da tensão de saída através de um *array* de resistores controlado por chaves NMOS.

A topologia base utilizada para a síntese é a proposta por Huang et al. (2006). Através de uma variável chamada *topo_var*, é permitido à ferramenta escolher entre três variações da topologia: *beta-multiplier* com espelho PMOS simples (variação A), *beta-multiplier* com espelho PMOS cascode (variação B) e alteração da posição da resistência do *beta-multiplier* (variação C) (LI et al., 2009). É permitida também a seleção entre três tipos de resistência disponíveis na tecnologia: P-Poly, N+Poly e P+Poly, implementadas através da variável *tipo_res*. Ainda, há a possibilidade utilizar ou remover o ramo para compensação do efeito de modulação do comprimento de canal (MATEUS et al., 2008) através da variável *comp_mod*. As possibilidades de topologia estão ilustradas na Figura 4.23.

Para realizar o ajuste digital da tensão na saída, a resistência R_3 pode ser variada através do acionamento do *array* de resistores (Figura 4.24). Durante a síntese, em vez de realizar o ajuste de R_3 digitalmente, é realizada uma varredura diretamente no valor de R_3 para encontrar o valor que faz a saída da fonte de referência ser 200 mV. Para fazer uso deste valor, é utilizado um recurso da ferramenta que permite ao projetista definir que o resultado de uma medida em um *testbench* seja utilizado como parâmetro para os próximos. Assim, o primeiro *testbench* encontra o R_3 que fornece a tensão correta na saída da fonte de referência e os demais utilizam este valor para realizar as demais medições. Esta busca é realizada para cada *corner* do processo, de forma que as medidas subsequentes são realizadas com o R_3 corrigido para variações de processo. Uma vez encontrada a solução final, a diferença entre a resistência máxima e a mínima entre os *corners* serve de guia para determinar qual deverá ser o valor da parte fixa e o valor da parte variável do resistor a fim de compensar as variações de processo.

Para propiciar a busca pela palavra digital que melhor corrige a tensão de saída, é criado um bloco em Verilog-A para converter números decimais de 0 a 15 em código binário de 4 bits. Cada *bit* é convertido em tensão e aplicado a porta do respectivo transistor NMOS que está atuando como chave na parte variável da resistência R_3 . É, então, realizada uma varredura do valor de entrada do bloco Verilog-A de 0 a 15 com passo 1. Assim, encontra-se o valor decimal

Figura 4.23: Variações da topologia de fonte de referência proposta por Huang et al. (2006) permitidas à ferramenta para seleção durante síntese.

Figura 4.24: Resistor ajustável digitalmente.

que minimiza o erro entre a tensão de saída e os 200 mV desejados.

As especificações do projeto podem ser vistas na Tabela 4.27. Após a síntese, a topologia da melhor solução utiliza o *beta-multiplier* com espelho PMOS cascode (variação B), não utiliza o ramo de compensação do efeito de modulação do comprimento de canal e tem seus resistores implementados com P-Poly. As dimensões finais dos dispositivos podem ser vistas na Tabela 4.28. Os resultados nominais e para *corners* de processo estão apresentados na Tabela 4.29. É possível observar que para todos os *corners* a solução está dentro das especificações. É possível utilizar os valores de $R3$ dessa tabela como referência para dividir o resistor entre a parte fixa e variável. Também é possível realizar simulações de Monte Carlo de varreduras do valor de $R3$ para encontrar o intervalo de resistência necessário para compensar as variações de processo. Esta foi a abordagem adotada e foram considerados 4 *bits* para realizar o ajuste da tensão de saída. Dessa forma, a resistência fixa de $R3$ (R_F da Figura 4.24) ficou determinada em 353 k Ω e os segmentos da parte variável (R_V da Figura 4.24) em 6,75 k Ω .

Tabela 4.27: Especificações da fonte de referência em tecnologia 130 nm com saída de 200 mV.

Medida	Especificação	Tipo*
Tensão de saída [mV]	200	Restrição
Coeficiente de temperatura [ppm V/°C] (de -20 °C até 80 °C)	< 30	Objetivo
Regulação de linha (mV/V)	< 15	Objetivo
Ruído de saída (μ V)	< 10	Restrição
PSRR (dB)	> 35	Objetivo
Consumo de potência (μ W)	< 50	Objetivo
Area (μ m) ²	< 10000	Objetivo
Coeficiente de Inversão de M3	< 0,1	Restrição
Coeficiente de Inversão de M8	< 0,1	Restrição
Coeficiente de Inversão de M9	< 0,1	Restrição

* Especificação tipo “Restrição”: função custo não tem alterações após a especificação atendida;
Especificação tipo “Objetivo”: função custo tem alterações mesmo após a especificação atendida.

O comportamento da tensão de saída em função da temperatura e em função da variação da tensão de alimentação pode ser visto nas figuras 4.25 e 4.26. A Figura 4.27 mostra a resposta transiente da fonte de referência quando a tensão de alimentação vai de 0V a 1,2V em 100 ns com uma capacitância de carga de 10 pF. O comportamento do circuito está adequado e dentro do previsto para a topologia e especificações.

Foram feitas simulações de Monte Carlo com 1000 amostras para a solução encontrada em duas situações: sem ajuste digital e com ajuste digital de 4 *bits*. Sem ajuste digital, a tensão de saída tem média 200 mV e desvio padrão de 5,15 mV, enquanto que com ajuste digital a média permanece a mesma mas o desvio padrão cai para 0,9 mV (Figura 4.28). O coeficiente de temperatura (Figura 4.29) e o PSRR (Figura 4.30) apresentam pouca variação entre o circuito sem ajuste digital e com ajuste digital. Essa semelhança é favorável ao projeto pois significa que as chaves que ajustam a tensão de saída não afetam intensamente as demais especificações.

Tabela 4.28: Dimensões dos dispositivos para o circuito fonte de referência em tecnologia 130 nm com saída de 200 mV.

Dispositivo	Dimensões ($W\mu m/L\mu m$)*
M1	8,59 / 16,3
M2	0,42 / 18,6 (x6)
M3	1,29 / 0,85 (x2)
M4	3,22 / 15,2 (x7)
M5	não utilizado**
M6	8,59 / 16,3 (x3)
M7	8,59 / 16,3 (x3)
M8	54,4 / 11,3
M9	54,4 / 11,3 (x4)
M10	1,15 / 16,3 (x10)
M11	0,42 / 18,6 (x2)
M12	0,5 / 1,4 (x10)
M13	0,5 / 1,4 (x10)
R1	2 / 1004 (477 k Ω)
R2	2 / 150 (71,4 k Ω)
R3	2 / 847 (402,7 k Ω)

* quando há múltiplos *fingers*, a quantidade é expressa em um fator multiplicativo;

** na solução final, o ramo de compensação de modulação do comprimento de canal não foi utilizado, portanto M5 não foi impresso na *netlist*.

Tabela 4.29: Resultados nominais (TT) e para *corners* de processo para a fonte de referência em tecnologia 130 nm e saída de 200 mV (nesta etapa, o ajuste de R3 é realizado diretamente, não através de chaves).

Nome	TT	FF	SS	SF	FS
R3 [Ω] para tensão de saída ser igual a 200 mV	402,7k	402,0k	402,6k	375,5k	433,6k
Coefficiente de Temperatura [ppm V/ $^{\circ}$ C] (de -20 $^{\circ}$ C até 80 $^{\circ}$ C)	18,6	23,8	19,2	21,7	21,5
Regulação de Linha [mV/V]	6,5	6,7	6,3	6,3	6,8
Ruído de Saída [nV] (medido em 10 Hz)	464,3	455,7	476,6	455,7	475,0
PSRR [dB] (medido em 5 kHz)	41,9	41,8	42,1	42,2	41,6
Consumo de Potência [μ W]	2,76	2,75	2,81	2,86	2,66
Area (portas + resistores*) [$(\mu m)^2$]	8,66 k	8,66 k	8,66 k	8,55 k	8,8 k

* nesta fase de projeto ainda não se considera R3 ajustado por chaves, e, portanto, a area está sendo modificada de acordo com a variação da resistência

Figura 4.25: Variação da tensão de saída em função da temperatura para a fonte de referência em tecnologia 130 nm e saída de 200 mV.

Figura 4.26: Variação da tensão de saída em função da tensão de alimentação da fonte de referência em tecnologia 130 nm e saída de 200 mV.

Figura 4.27: Análise transiente da fonte de referência em tecnologia 130 nm e saída de 200 mV com alimentação variando de 0 a 1,2V em 100ns.

Figura 4.28: Tensão de referência para 1000 simulações de Monte Carlo para fonte de referência em tecnologia 130 nm e saída de 200 mV com e sem ajuste digital.

Figura 4.29: Coeficiente de temperatura para 1000 simulações de Monte Carlo para fonte de referência em tecnologia 130 nm e saída de 200 mV com e sem ajuste digital.

Figura 4.30: PSRR para 1000 simulações de Monte Carlo para fonte de referência em tecnologia 130 nm e saída de 200 mV com e sem ajuste digital.

4.6 Projeto de Fonte de Referência com saída de 600 mV em 130 nm

As especificações deste projeto estão relacionadas ao projeto SAMPA ASIC, que é o *front-end* para detecção de partículas sendo desenvolvido para o experimento ALICE (do inglês, *A Large Ion Collider Experiment*) do Grande Colisor de Hádrons (LHC, do inglês, *Large Hadron Collider*) do CERN. Por especificação de sistema, o circuito será submetido à radiação, porém em níveis que não demandam a utilização de ELTs (*Enclosed-Layout Transistors*) no projeto.

A tecnologia utilizada possui comprimento de canal de 130 nm com tensão de alimentação de 1,2 V, enquanto a tensão de referência necessita ser próxima de 600 mV. Para criação de tensões abaixo de 1V em tecnologias com apenas CMOS, pode-se utilizar transistores operando em região sublimiar (como visto no projeto anterior) ou diodos e soma de correntes. Enquanto no primeiro caso há maior resistência à radiação, com o uso de diodos há a vantagem da obtenção de tensões estáveis e robustas às variações de processo sem a necessidade de *trimming* da saída.

O projeto apresentado a seguir utiliza a topologia exposta na Figura 4.31 (BANBA et al., 1999) e utiliza especificações e guias de projeto relacionados à obtenção de resultados resistentes a níveis baixos de radiação.

Figura 4.31: Topologia proposta por Banba et al. (1999) (circuito de start-up não mostrado na figura).

Para projetos resistentes à radiação em 130 nm, é recomendado o uso de comprimento de

canal superior a $0,8 \mu\text{m}$ para transistores NMOS e superior a $0,4 \mu\text{m}$ para transistores PMOS (FACCIO; GONELLA, 2006 apud SOUSA, 2009). Para a topologia utilizada, é recomendado o aumento da corrente nos diodos para redução da influência da corrente de fuga gerada pela dose total de ionização (MOREIRA et al., 2003). As especificações do projeto podem ser vistas na Tabela 4.30.

Tabela 4.30: Especificações da fonte de referência de 600 mV em 130 nm.

Medida	Especificação	Tipo*
Tensão de saída [mV]	600	Restrição
Desvio padrão da tensão de saída [mV]	< 5	Objetivo
Coeficiente de temperatura [ppm V/°C] (de -20 °C até 80 °C)	< 30	Objetivo
Regulação de linha [mV/V]	< 10	Objetivo
Ruído de saída [μV] (medido em 10 Hz)	< 5	Objetivo
PSRR [dB] (medido em 1 kHz)	> 35	Objetivo
Consumo de potência [μW]	< 5000	Objetivo
Area [μm] ²	< 10000	Objetivo
Corrente nos diodos [μA]	> 1	Objetivo

* Especificação tipo “Restrição”: função custo não tem alterações após a especificação atendida;

Especificação tipo “Objetivo”: função custo tem alterações mesmo após a especificação atendida.

As dimensões finais dos dispositivos podem ser vistas na Tabela 4.31. O número de diodos n foi arbitrado em 17 devido ao resultado de sínteses prévias. Os resultados nominais estão apresentados na Tabela 4.32.

O comportamento da tensão de saída em função da temperatura e em função da variação da tensão de alimentação pode ser visto nas figuras 4.32 e 4.33. A Figura 4.34 mostra a resposta transiente da fonte de referência quando a tensão de alimentação vai de 0V a 1,2V em 100ns. O comportamento ilustrados nas figuras está dentro do previsto para a topologia e especificações.

Tabela 4.31: Dimensões dos dispositivos para o circuito fonte de referência de 600 mV em tecnologia 130 nm.

Dispositivo	Dimensões*
M1, M2, M3	53,5(x5) / 6,63
M4	2,33(x2) / 0,92
M5	2,33(x3) / 0,92
M6	53,5(x2) / 6,63
M7, M8	45,9(x2) / 19,6
M9, M10	61,6(x5) / 7,43
D1,D2	29,7 / 29,7
R1, R2	2 / 25,2
R3	2 / 3,33
R4	2 / 12,24
C1	48,7 / 48,7
C2	51,9 / 51,9

* largura (W) e comprimento (L) em μm . Valores expressos no formato $W(xf)/L$, onde f é um fator multiplicativo representando o número de *fingers*.

Tabela 4.32: Resultados nominais para o circuito fonte de referência de 600 mV em tecnologia 130 nm.

Nome	Resultado Nominal
Tensão de Saída	600 mV
Coefficiente de Temperatura (de -20° até 80° C)	3,7 ppm V/°C
Regulação de Linha	680,4 μ V
Ruído de Saída (medido em 10 Hz)	844,6 nV
PSRR (medido em 1 kHz)	63,30 dB
Consumo de Potência	448 μ W
Area (portas + resistores + diodos)	35.290 (μ m) ²

Figura 4.32: Variação de tensão de saída em função da temperatura da fonte de referência de 600 mV em tecnologia 130 nm.

O *layout* do circuito foi realizado utilizando as ferramentas do *Virtuoso Analog Design Environment* da empresa *Cadence*. Algumas modificações foram realizadas no esquemático durante a criação do *layout*. Os transistores do par diferencial foram divididos em mais *fingers* para permitir o desenho utilizando a técnica de centroide comum. Ainda, os transistores com largura muito grandes foram divididos em mais *fingers* do que o previsto no projeto original. Essa alteração é devido a uma regra do *design kit* que determina que a distância máxima entre qualquer ponto da canal de um transistor e a conexão para polarização do substrato deve ser de 30 μ m. Assim, os transistores do par diferencial M7 e M8 foram divididos em 4 *fingers* (em vez de 2 como previsto anteriormente), os transistores M9 e M10 foram divididos em 10 (em vez de 5). Transistores *dummies* foram adicionados às extremidades dos arranjos de transistores. As resistências, que haviam sido otimizadas considerando sua largura fixa em 2 μ m, foram aumentadas para 10 μ m e os comprimentos alterados proporcionalmente a fim de reduzir o efeito das variações de processo. Outra modificação em relação ao esquemático foi o aumento das capacitâncias C_1 e C_2 , bem como a adição de capacitância entre os terminais de

Figura 4.33: Variação de tensão de saída em função da tensão de alimentação da fonte de referência de 600 mV em tecnologia 130 nm.

Figura 4.34: Análise transiente da fonte de referência de 600 mV em tecnologia 130 nm com tensão de alimentação em degrau de 0 a 1,2V em 100ns.

alimentação. Essas alterações devem-se ao melhor aproveitamento da área disponível entre as partes do circuito.

A Figura 4.35 mostra o layout total do circuito evidenciando as regiões do circuito (as partes não evidenciadas são as capacitâncias do circuito). Além do par diferencial do amplificador (M7 e M8), os transistores do espelho (M9 e M10) também foram realizados utilizando a técnica de centroide comum. Todos os demais espelhos de corrente do circuito foram desenhados utilizando transistores interdigitados. As resistências (R_1 , R_2 , R_3 , e R_4) foram desenhadas uti-

Figura 4.35: *Layout* da fonte de referência com saída de 600 mV projetada em tecnologia 130 nm.

lizando centroide comum visto a importância das relações entre elas para a geração da tensão de saída. Os diodos são do tipo $P+/N\text{-well}$ e cada um foi desenhado dentro de um *guard-ring* próprio. A disposição dos diodos é tal que D_1 é o mais próximo do centro, com os demais diodos em volta.

As formas de onda resultantes da simulação do circuito extraído seguem o mesmo comportamento do circuito simulado apenas com o esquemático. Para averiguar a operação correta do circuito diante de variações de processo, foram feitas simulações de Monte Carlo com 1000 amostras. Os resultados destas simulações podem ser vistos nas figuras 4.36, 4.37 e 4.38.

Para a versão anterior à extração de parasitas do *layout*, a tensão de saída tem média igual a 599,3 mV e desvio padrão 1,625 mV. O coeficiente de temperatura tem média igual a 3,93 e desvio padrão 1,82. O PSRR tem média igual a 63,38 e desvio padrão 0,574.

Para a versão posterior à extração de parasitas do *layout*, a tensão de saída tem média igual a 597,4 mV e desvio padrão 0,760 mV. O coeficiente de temperatura tem média igual a 7,06 e desvio padrão 1,19. Por fim, o PSRR tem média igual a 61,65 e desvio padrão 0,415.

Pelos resultados, pode-se observar a existência de diferenças entre as versões anterior e posterior à extração, indicando que as capacitâncias e resistências parasitas afetam o desempenho do circuito. No entanto, os resultados de ambas versões estão dentro das especificações e são robustos às variações de processo.

Figura 4.36: Tensão de referência para 1000 simulações de Monte Carlo para fonte de referência com saída projetada para 600 mV em tecnologia 130 nm com e sem extração de *layout*.

Figura 4.37: Coeficiente de Temperatura para 1000 simulações de Monte Carlo para fonte de referência com saída de 600 mV em tecnologia 130 nm com e sem extração de *layout*.

4.7 Projeto de Fonte de Referência com saída de 600 mV em 180 nm

A mesma topologia e especificações similares a do projeto da fonte de referência da seção anterior (realizada em 130 nm) são utilizadas nesta seção para a síntese na tecnologia IBM 180 nm. Para este projeto, a tensão de alimentação é de 1,8 V.

As dimensões finais dos dispositivos podem ser vistas na Tabela 4.33, tendo sido o número de diodos n arbitrado em 17. Os resultados nominais estão apresentados na Tabela 4.34.

O comportamento da tensão de saída em função da temperatura e em função da variação da tensão de alimentação pode ser visto nas figuras 4.39 e 4.40. A Figura 4.41 mostra a resposta

Figura 4.38: PSRR para 1000 simulações de Monte Carlo para fonte de referência com saída de 600 mV em tecnologia 130 nm com e sem extração de *layout*.

Tabela 4.33: Dimensões dos dispositivos para o circuito fonte de referência de 600 mV em 180 nm.

Dispositivo	Dimensões*
M1, M2, M3	85,6(x2) / 6,24
M4	1,64(x3) / 0,80
M5	1,64(x4) / 0,80
M6	85,6 / 6,24
M7, M8	3,54(x20) / 19,9
M9, M10	3,71(x5) / 16,8
D1,D2	15 / 15
R1, R2	2 / 140,6
R3	2 / 140,6
R4	2 / 68
C1	12,2 / 12,2
C2	4,1 / 4,1

* largura (W) e comprimento (L) em μm . Valores expressos no formato $W(xf) / L$, onde f é um fator multiplicativo representando o número de *fingers*.

Tabela 4.34: Resultados nominais para o circuito Fonte de Referência de 600 mV em 180 nm.

Nome	Resultado Nominal
Tensão de Saída	600 mV
Coeficiente de Temperatura (de -20° até 80° C)	3,54 ppm V/°C
Regulação de Linha	134,5 μV
Ruído de Saída (medido em 10 Hz)	1,08 μV
PSRR (medido em 1 kHz)	74,71 dB
Consumo de Potência	124,7 μW
Area (portas + resistores + diodos)	12.390 (μm) ²

transiente da fonte de referência quando a tensão de alimentação vai de 0V a 1,8V em 100ns. O comportamento da fonte de referência está adequado e dentro do previsto para a topologia e especificações.

Figura 4.39: Variação de tensão de saída em função da temperatura da fonte de referência de 600 mV em tecnologia 180 nm.

Figura 4.40: Variação de tensão de saída em função da tensão de alimentação da fonte de referência de 600 mV em tecnologia 180 nm.

A Figura 4.42 mostra o *layout* total do circuito evidenciando as regiões do circuito. As mesmas diretrizes usadas para desenhar o circuito em 130 nm foram utilizadas para o circuito em 180 nm, tais como uso da técnica de centroide comum e interdigitação de transistores de espelhos de corrente.

Os resultados de simulações de Monte Carlo com 1000 iterações podem ser vistos nas figuras 4.43, 4.44 e 4.45. Para a versão anterior à extração de parasitas do *layout*, a tensão de saída tem média igual a 599,9 mV e desvio padrão 1,455 mV. O coeficiente de temperatura tem

Figura 4.41: Análise transiente da tensão de referência com tensão de alimentação em degrau de 0 a 1,8V em 100ns da fonte de referência de 600 mV em tecnologia 180 nm.

Figura 4.42: Layout da fonte de referência com saída de 600 mV projetada em tecnologia 180 nm.

média igual a 3,58 e desvio padrão 0,93. O PSRR tem média igual a 74,74 e desvio padrão 0,229.

Para a versão posterior à extração de parasitas do layout, a tensão de saída tem média igual a 600,2 mV e desvio padrão 1,399 mV. O coeficiente de temperatura tem média igual a 13,2 e desvio padrão 2,03. Por fim, o PSRR tem média igual a 72,29 e desvio padrão 0,178.

De forma similar ao projeto em 130 nm, apesar das diferenças entre as versões anterior e posterior à extração, ambas respeitam as especificações de projeto e são robustas às variações de processo.

Figura 4.43: Tensão de referência para 1000 simulações de Monte Carlo para fonte de referência com saída de 600 mV em tecnologia 180 nm.

Figura 4.44: Coeficiente de temperatura para 1000 simulações de Monte Carlo para fonte de referência com saída de 600 mV em tecnologia 180 nm.

4.8 Exploração da Fronteira de Pareto

A exploração da fronteira de Pareto em circuitos integrados analógicos permite que as trocas de projeto entre as especificações sejam observadas. Em especial, em duas situações esta exploração é relevante.

A primeira situação de interesse é quando o projetista almeja entender as trocas de projeto para uma dada topologia e em uma dada tecnologia para que com elas possa escolher as especificações que melhor atendam as suas demandas. A segunda situação de interesse é quando há uma posterior síntese em nível de sistema. Nesse segundo caso, as informações contidas na fronteira de Pareto são utilizadas em nível de sistema para que sistemas com blocos sem

Figura 4.45: PSRR para 1000 simulações de Monte Carlo para fonte de referência com saída de 600 mV em tecnologia 180 nm.

correspondência sintetizável não sejam aceitos como solução. Apenas especificações de blocos que correspondem ao espaço de critério viável são consideradas sintetizáveis. Além dessa abordagem reduzir o espaço de projeto durante a síntese em nível de sistema, caso seja atrelada a modelagem dos blocos, ela também pode evitar a necessidade de demoradas simulações horizontais (envolvendo todos os blocos em nível de transistores). Este tópico é estudado com mais detalhe no próximo capítulo.

4.8.1 Restrições para Algoritmos Multiobjetivo

A utilização de algoritmos multiobjetivo aplicados a circuitos elétricos requer atenção especial na criação da função custo. Caso estes algoritmos utilizem diretamente as funções custos apresentadas na seção 4.1.3 ou apenas as medidas, grande parte dos resultados pode ser distribuída por regiões com especificações aquém das esperadas. Isso decorre do método pelo qual a maior parte deles opera, que é a relação de dominação entre as respostas. Assim, não importando quão baixo seja o desempenho de um circuito em uma dada especificação, a solução será interessante do ponto de vista do algoritmo caso ela não seja dominada. Disso, pode resultar que grande parte do esforço na exploração da fronteira de Pareto seja realizado em extremos de projeto pouco interessantes do ponto de vista prático.

Neste trabalho, quando uma solução estiver fora de uma especificação mínima, todas as medidas de desempenho deste circuito terão uma penalidade acrescida ao seu valor. Assim, a cada medida será adicionado o somatório das penalizações relativas as infrações das especificações

mínimas. A equação 4.10 descreve o efeito das penalidades em cada medida de desempenho.

$$F_i = \begin{cases} M_i(\vec{x}), & \text{se } M_i \leq M_{\min_i} \\ M_i(\vec{x}) + \sum_{i=1}^n P_i(x), & \text{se } M_i > M_{\min_i} \end{cases} \quad (4.10)$$

onde F_i é a função custo para a medida M_i , a especificação mínima é $M_{\min_i}$ e $P_i(x)$ é a penalidade proporcional às infrações das especificações mínimas e n é o número total de especificações mínimas. Deve ser observado que o sinal de M_i é invertido para casos de objetivo de maximização, de forma que o algoritmo sempre busque o menor resultado possível.

Ao longo das iterações, isso faz com que os algoritmos evitem solução fora das especificações mínimas. Estas soluções serão piores que as demais em todas as medidas e, portanto, não dominarão qualquer outra que respeite as especificações mínimas.

4.8.2 Exposição dos Resultados Multiobjetivo

Em problemas de otimização multiobjetivo, a fronteira de Pareto tem n° de dimensões igual ao n° de objetivos. Dessa forma, para casos com mais do que 3 dimensões, não há método intuitivo de representar a fronteira visualmente.

A estratégia adotada neste trabalho é a separação das informações contidas na fronteira de Pareto em múltiplos gráficos. Cada gráfico contém dois eixos e uma terceira dimensão é representada através de cores. Esta terceira dimensão é variada em cada gráfico para representar as especificações restantes. Assim, ao observar todos os gráficos lado a lado, o projetista pode ter uma visão abrangente sobre as respostas obtidas pelo algoritmo. A Figura 4.46 ilustra um exemplo de resultado de fronteira de Pareto para um amplificador Miller. Neste gráfico, as dimensões principais representam a Frequência de Ganho Unitário (UGF) e o ganho DC, enquanto as dimensões representadas por cores são o *Slew Rate*, a corrente consumida e a área. Cada ponto em um gráfico está representado também nos demais.

4.8.3 Testes

Para avaliar a exploração da Fronteira de Pareto, foram utilizados os circuitos amplificadores apresentados nas seções 4.3.2 e 4.3.3. O número de iterações máximo para cada síntese é 10.000. Para cada topologia, cada algoritmo é executado 5 vezes. O critério de comparação é o mesmo utilizado no capítulo anterior para casos multiobjetivos. Calcula-se $I_i = 1 - H_i$, onde I_i é a métrica de comparação e H_i é o volume do hipercubo (volume do espaço de critério dominado pela aproximação da fronteira de Pareto).

Figura 4.46: Exemplo de exposição de resultados da fronteira de Pareto para um amplificador *folded cascode* complementar.

Os testes são realizados com tensão de alimentação de +1,65 e -1,65 e tecnologia AMS 0,35 μm . As especificações mínimas e os objetivos de projeto são apresentados na Tabela 4.35. As especificações do tipo objetivo servirão para criar a fronteira de Pareto. Tanto nelas quanto nas especificações do tipo restrição, o valor da especificação define o critério mínimo para que uma solução seja considerada para compor a fronteira de Pareto. Os valores das especificações dos objetivos não são tão exigentes como nos testes de objetivo único. Isso se deve ao interesse em obter um número suficiente de soluções da fronteira de Pareto para gerar os gráficos da mesma.

Tabela 4.35: Objetivos de projeto e especificações mínimas para a síntese multiobjetivo.

Medida	Especificação	Tipo
UGF M[Hz]	> 10	Objetivo
Corrente m [A]	< 60	Objetivo
Margem de Fase [°]	> 60 e < 90	Restrição
Ganho DC dB[V]	> 40	Objetivo
<i>Slew Rate</i> (pos) [V/ μs]	> 1	Restrição
<i>Slew Rate</i> (neg) [V/ μs]	< -1	Restrição
<i>Slew Rate</i> médio [V/ μs]	> 1	Objetivo
Excursão Máxima de Saída [V]	> 2	Restrição
CMRR dB[V]	> 50	Restrição
PSRR dB[V]	> 50	Restrição
Area de Porta [mm] ²	< 10	Objetivo

A Figura 4.47 representa as fronteiras de Pareto obtidas por cada algoritmo para o caso do amplificador Miller, enquanto a Figura 4.48 representa a fronteira de Pareto do amplificador *folded cascode* complementar. Em ambos os casos, é possível ver que o algoritmo proposto é capaz de obter uma melhor distribuição no espaço de critério.

A Tabela 4.36 mostra os resultados da métrica baseada em hipercubo $I_i = 1 - H_i$ (explicada

na seção 3.3.1) para a síntese do amplificador Miller. Da mesma forma, a Tabela 4.37 mostra essa mesma métrica para o caso do amplificador *folded cascode* complementar. Indo de acordo com o que foi visto na figura da fronteira de Pareto da melhor resposta, é observável que as melhores respostas são do algoritmo proposto. É importante ressaltar que as disparidades entre os resultados numéricos são em parte consequência dos limites do hipercubo serem definidos pelas melhores e piores respostas.

Figura 4.47: Aproximações da fronteira de Pareto do Amplificador Miller.

No capítulo que segue, tanto o algoritmo multiobjetivo desenvolvido quanto a estratégia de representação da fronteira de Pareto serão utilizados para o desenvolvimento de transcondutores.

Figura 4.48: Aproximações da fronteira de Pareto do amplificador *folded cascode* complementar.

Tabela 4.36: Resultados de $1 - H_i$ (onde H_i é o volume do hipercubo) para as sínteses multiobjetivo do amplificador Miller.

Iteração	Alg. Proposto	Fase 3 do			
		Alg. Proposto Isoladamente	MOPSO	SPEA2	NSGA-II
1	0,000000	1	0,872940	0,687382	0,427422
2	0,887900	1	0,894002	0,687440	0,972252
3	0,292190	1	0,957260	0,688160	1,000000
4	0,000000	1	0,914124	0,686966	0,864776
5	0,079380	1	0,899388	0,687470	0,948324

4.9 Tempos de Síntese

Devido às sínteses deste capítulo terem sido executadas por processadores em diferentes situações de carga e poder computacional, os tempos de síntese necessários para os testes não pôde ser utilizado como critério de parada ou comparação. No entanto, com intuito de fornecer

Tabela 4.37: Resultados de $1 - H_i$ (onde H_i é o volume do hipercubo) para as sínteses multiobjetivo do amplificador *folded cascode* complementar.

Iteração	Alg. Proposto	Fase 3 do			
		Alg. Proposto Isoladamente	MOPSO	SPEA2	NSGA-II
1	0,346892	1	0,939838	0,473962	0,844408
2	0,515938	1	0,972954	0,474176	0,794456
3	0,494668	0,997850	0,957260	0,475200	0,689414
4	0,345234	0,990386	0,968056	0,473950	0,688668
5	0,817542	1	0,960400	0,474250	0,757726

Tabela 4.38: Tempos de síntese dos testes realizados em nível de transistores.

Síntese	Iterações	Tempo
<i>Testes de comparação com outros algoritmos</i>		
Amplificador Miller	2000	≈ 65 minutos
Amplificador <i>Folded Cascode</i> Complementar	2000	≈ 90 minutos
LNA	2000	≈ 25 minutos
Fonte de Referência	2000	≈ 10 minutos
<i>Testes de comparação com projetos manuais</i>		
Miller sem <i>corners</i>	5000	1 hora e 10 minutos
Miller com <i>corners</i>	5000	3 hora e 49 minutos
LNA (solução 1 + solução 2)	1154	4 horas
<i>Testes de comparação com outras técnicas/ferramentas</i>		
Amplificador <i>Folded Cascode</i>	2500	23 minutos
Fonte de Referência	5000	2 horas e 25 minutos
<i>Testes de fontes de referência com Monte Carlo e/ou corners</i>		
Fonte de Referência em 130nm com 600mV de saída (MC)	5000	14 horas e 6 minutos
Fonte de Referência em 180nm com 600mV de saída (MC)	5000	8 horas e 35 minutos
Fonte de Referência com variações de topologia (<i>corners</i>)	5000	2 horas e 35 minutos
<i>Testes comparativos multiobjetivo</i>		
Amplificador Miller	5000	2 horas e 16 minutos
Amplificador <i>Folded Cascode</i> Complementar	5000	2 horas e 16 minutos

uma referência sobre o tempo estimado para as sínteses, na Tabela 4.38 estão apresentados os tempos despendidos para os testes realizados. Em azul estão ilustrados os tempos aproximados de execução dos testes de comparação com outros algoritmos, que foram inferiores à 1 hora e meia. Por outro lado, em vermelho estão enfatizados os testes utilizando simulações de Monte Carlo, para os quais foram utilizados de 8 a 14 horas.

4.10 Resumo do Capítulo

Neste capítulo foi apresentada a estrutura da ferramenta em nível de circuito, descrevendo seus blocos internos e a interface com o usuário. A ferramenta tem como uma de suas características permitir flexibilidade em seu uso pelo projetista, que pode descrever qualquer circuito, *testbench* ou medição sem necessitar que eles tenham sido previstos durante a criação da ferramenta. Também foi descrita a forma de realização de alteração da topologia com o uso de

condicionais na *netlist*. Assim, o otimizador é capaz de decidir qual é a melhor topologia ou tipo de componente durante a síntese. A criação da função custo foi analisada e seu uso para síntese guiada à robustez dos projetos. Foram realizados diversos testes e comparações da ferramenta com trabalhos manuais, outras técnicas, e com outros algoritmos. Por fim, foi realizada a comparação entre diversos algoritmos para obtenção da fronteira de Pareto de amplificadores. Os resultados mostraram que a ferramenta é capaz de realizar apropriadamente a síntese de circuitos integrados analógicos, bem como os algoritmos propostos no capítulo anterior têm desempenho favorável quando comparados com os demais para os casos de síntese analisados.

5 SÍNTESE EM NÍVEL DE SISTEMA

Neste capítulo são apresentados e testados procedimentos para síntese de circuitos integrados analógicos em nível de sistema. A abordagem utilizada neste trabalho constrói aproximações de fronteiras de Pareto em nível de circuito e as utiliza em nível de sistema como guia para decidir sobre a viabilidade dos blocos. Com a definição da arquitetura por parte do usuário, a ferramenta seleciona dentro da fronteira de Pareto de cada bloco a solução que melhor satisfaz as especificações do sistema. Alternativamente, em sínteses de caráter exploratório, a ferramenta busca as especificações mínimas dos blocos que atingem as especificações de sistema utilizando a fronteira de Pareto apenas como referência.

A ferramenta utiliza uma abordagem *bottom-up* multiobjetivo (descrita na seção 2.1.8). O algoritmo proposto para a busca de apenas uma solução final apresentado no capítulo 3 é aplicado a sistemas utilizando especificações de blocos comportamentais como variáveis. A descrição das variáveis de projeto na ferramenta é revisitada para considerar as particularidades do nível de abstração de sistema. O método utilizado para geração da função custo e a estrutura da ferramenta de síntese em nível de sistema são apresentados. Ainda, o processo de síntese de filtros G_m -C é demonstrado utilizando os conceitos apresentados no capítulo.

5.1 Procedimento de Síntese em Nível de Sistema

Neste trabalho, as otimizações em nível de sistema utilizam dados previamente obtidos em nível de circuito (fronteiras de Pareto) e modelos comportamentais. Dessa forma, as simulações em nível de sistema utilizam modelos com complexidade menor e, conseqüentemente, exigem um esforço computacional reduzido. Apesar disso, o uso de dados obtidos com a simulação prévia dos blocos garante a fidedignidade das simulações.

A síntese em nível de sistema deste trabalho foi desenvolvida no *software MATLAB*. Para a descrição dos blocos do sistema é utilizada a linguagem *Verilog-A* e, para simulação, é utilizado o *software HSPICE*.

5.1.1 Avaliação da Viabilidade de Especificações através da Fronteira de Pareto

Para um sintetizador de alto nível é importante extrair de um bloco de baixo nível apenas as características fundamentais para modelá-lo apropriadamente respeitando suas limitações. Não é necessário modelá-lo de forma a reproduzir sua estrutura física e os princípios operacionais do bloco. De forma similar, não é necessário utilizar as mesmas variáveis que são utilizadas para modelar o bloco em baixo nível, tais como as dimensões de transistores. Em nível de sistema, o que é importante para o sintetizador é ser capaz de estimar se um bloco com determinadas especificações e topologia é possível de ser implementado em nível de transistores.

Conforme visto na seção 2.2.1, a região contendo as soluções realizáveis por um determinado bloco é chamada região de projeto viável ($\mathbf{X}$), enquanto a região com as especificações passíveis de serem atingidas é chamada região de critério viável ($\mathbf{Z}$). As equações 2.11 e 2.12 são reescritas aqui por conveniência:

$$\mathbf{X} = \{\vec{x} | h_i(\vec{x}) = 0, i = 1, 2, \dots, p; \quad e \quad g_j(\vec{x}) \leq 0, j = 1, 2, \dots, m\}$$

$$\mathbf{Z} = \{f(\vec{x}) | \vec{x} \in \mathbf{X}\}$$

onde $\vec{x}$ é o vetor de projeto n -dimensional, $f(\vec{x})$ é a função custo, $h_i(\vec{x})$ com $i = 1$ até p são as restrições de igualdade e $g_j(\vec{x})$ com $j = 1$ até m são as restrições de desigualdade.

Uma possível abordagem para este problema é concentrar esforços na modelagem da região $\mathbf{X}$ e utilizar modelos com parâmetros de entrada de baixo nível (dimensão de transistores) mesmo ao realizar otimização em nível de sistema. No entanto, utilizar as variáveis do vetor $\vec{x}$ durante a síntese de nível de sistema aumenta desnecessariamente o número de variáveis do problema, reduzindo grande parte da vantagem de trabalhar-se em um nível superior de abstração. Em nível de sistema, é possível trabalhar diretamente com a região $\mathbf{Z}$.

A fronteira de Pareto é suficiente como fonte de informações sobre a região $\mathbf{Z}$ uma vez que representa os compromissos de projeto que separam o espaço de critério viável do inviável. Neste trabalho, a fronteira de Pareto é utilizada para dois processos distintos. Inicialmente, ela é usada como referência para exploração do espaço de projeto. Nesta etapa, o projetista faz a definição de quais variáveis serão utilizadas e se serão independentes entre si ou conetadas com outras. Essa etapa faz basicamente a ligação entre o nível de sistema e o nível de transistores. Posteriormente, a fronteira é usada para determinar a sintetizabilidade das soluções através do critério de dominação.

Na subseção 2.1.8.2 foi estudado o estado da arte no que se refere a conversão de especificações entre nível de sistema e circuito. Os trabalhos apresentados na literatura normalmente utilizam apenas a indexação das fronteiras de Pareto como possibilidade para explorar as soluções dos blocos. Ali (2009) realiza a interpolação das respostas da fronteira de Pareto a fim de possibilitar mais liberdade na escolha das soluções. No entanto, este recurso não permite que o otimizador faça incursões ao interior da região de critério viável. No trabalho de Velasco-Jimenez et al. (2014), há um aprofundamento na questão da indexação da fronteira de Pareto. Nele, é criado um índice para cada dimensão da fronteira de Pareto afim de facilitar a navegação do otimizador pela mesma. No entanto, estas modificações ainda restringem às respostas encontradas às soluções encontradas na fronteira.

A abordagem proposta nesta tese permite o uso da fronteira de Pareto da forma tradicional para síntese. No entanto, ela permite também ao otimizador buscar soluções dentro do espaço de critério viável e inviável dos blocos, não limitando a síntese às fronteiras de Pareto encontradas. Em adição, o processo de síntese em nível de sistema deste trabalho é estruturado para ser flexível. Portanto, os arquivos de entrada garantem completo acesso às funcionalidades da ferramenta para quaisquer sistemas a serem sintetizados.

5.2 Processo de Síntese em Nível de Sistema

Tanto em nível de transistores quanto em nível de sistema, o objetivo da síntese é obter uma solução ou conjunto de soluções que atendam as especificações de projeto. A diferença fundamental entre ambas é o conteúdo das variáveis de projeto. Em nível de transistores, as variáveis são as dimensões dos dispositivos. Em nível de sistema, as variáveis são as especificações dos blocos de baixo nível. Para que estas especificações tenham um compromisso com a sintetizabilidade, um processo que restrinja ou penalize o uso de soluções não viáveis deve ser considerado. Dessa forma, apesar dos algoritmos utilizados para síntese em nível de transistores e em nível de sistema poderem ser parecidos, o último possui particularidades significativas que serão discutidas nesta seção.

A abordagem utilizada neste trabalho para manter compromisso com a sintetizabilidade é o uso de fronteiras de Pareto. Estas informam as melhores soluções viáveis em nível de transistores para um dado bloco, permitindo saber quais soluções são realizáveis mesmo enquanto modelos simplificados estão sendo utilizados para simulação durante a síntese em nível de sistema. Ao utilizar esta abordagem, as fronteiras já devem ter sido geradas para os blocos no momento de início das otimizações em sistema. Após sua geração, as fronteiras são lidas e organizadas de acordo com as escolhas de tipo de variáveis definidas pelo projetista, ponto que

será discutido em detalhe na seção 5.2.1. Para cada bloco e cada especificação de baixo nível (variável em nível de sistema), a estratégia de organização vai definir se as soluções da fronteira de Pareto serão as únicas opções válidas ou se a fronteira de Pareto servirá apenas como referência, sendo considerada principalmente através do uso da função custo de viabilidade (que será apresentada no decorrer deste capítulo).

A Figura 5.1 mostra algumas possibilidades de uso da fronteira de Pareto durante a síntese. Na Figura 5.1 (a), é ilustrado o uso de uma solução de uma fronteira de Pareto para várias instâncias de um mesmo bloco. Este arranjo é útil para casos no qual todos os blocos são iguais entre si, como, por exemplo, filtros G_m -C no qual apenas a multiplicidade do bloco é utilizada para alteração das transcondutâncias. Na Figura 5.1 (b), é ilustrado o uso de diversas soluções de uma fronteira de Pareto para vários blocos. Este arranjo pode ser utilizado quando não há necessidade que todos os blocos sejam iguais, mas quando todos devem pertencer a uma mesma topologia. O exemplo do filtro G_m -C também é válido para este caso, mas na situação em que eles possam ter dimensionamentos diferentes entre si. É importante que, neste caso, todas as soluções da fronteira de Pareto sejam compatíveis entre si em sua interface e polarização (esse tema é discutido na subseção 5.2.1.2 e em alguns exemplos deste capítulo). Na Figura 5.1 (c), é mostrada a síntese com uso de fronteiras de Pareto distintas, válida para sistemas que possuem blocos de tipos distintos. Por exemplo, para sintetizar um *Charge-Pump* PLL, são necessários um VCO, um *Charge-Pump* e um detector de fase, cada um com uma ou mais fronteira de Pareto específica para seu tipo de bloco. Na Figura 5.1 (d), a situação na qual a fronteira de Pareto é usada apenas como referência para obtenção das especificações de blocos é ilustrada. Neste caso, apesar do espaço de projeto também poder ser restringido de formas variadas pela fronteira de Pareto, a função principal desta é guiar a busca através da função custo de viabilidade. Assim, especificações de sistema que estão além das possibilidades da fronteira de Pareto podem ser obtidas, dando ao projetista informações sobre o que deve aperfeiçoar em seus blocos para conseguir obtê-las. De forma semelhante, soluções piores que as existentes na fronteira de Pareto podem ser utilizadas para atingir as especificações de sistema, mostrando ao usuário que poderá executar seu projeto com uma dada margem de segurança.

5.2.1 Sistema de Organização dos Parâmetros/Variáveis

O uso da fronteira de Pareto como referência para a síntese pode ser implementado de diversas formas. Para permitir flexibilidade nesta implementação, são criadas novas opções para descrição dos parâmetros.

Figura 5.1: Exemplos de possibilidades de uso da fronteira de Pareto para a síntese.

5.2.1.1 Tipos de Parâmetros em Nível de Sistema

Os tipos de parâmetros disponíveis em nível de sistema são: *real*, *grade*, *inteiro*, *igual* e *indexado*. Enquanto os primeiros já foram apresentadas no capítulo de síntese em nível de transistores, o parâmetro *indexado* é de uso exclusivo do nível de sistema. Este tipo de variável permite que uma solução assuma apenas valores existentes no conjunto de soluções coletados da fronteira de Pareto.

Ainda, aos tipos de variação já disponíveis (*variável*, *não variável*, *igual*), é acrescido o tipo *variável conectada*. Dois ou mais parâmetros *indexados* podem ser *conectados* para que o índice deles na fronteira de Pareto seja sempre o mesmo. Seguir o mesmo índice da fronteira de Pareto significa que respostas de uma mesma solução da fronteira serão utilizadas para as variáveis. Assim, configurar dois parâmetros para que sejam *variáveis conectadas* resulta em uma ação diferente do que configurá-los para serem *iguais*. No primeiro caso, eles compartilham o índice da solução na fronteira de Pareto. No último, eles têm o mesmo valor. Com esses tipos de parâmetros disponíveis, a utilização de indexação pode ser feita de forma independente para cada variável, ou, no caso mais típico, para um bloco inteiro (utilizando parâmetros conectados).

Como exemplo, caso um parâmetro de um bloco seja *indexado* e *variável*, sendo que os parâmetros restantes do mesmo bloco sejam *indexados* e *conectados* ao primeiro, o bloco como um todo (por exemplo, um OTA) seguirá o mesmo índice na fronteira de Pareto. Assim, durante a descrição nos arquivos de entrada, o projetista tem a liberdade de escolher qual variável/parâmetro da fronteira de Pareto será usada como índice para a busca. A variável índice será utilizada para organizar as soluções da fronteira de Pareto de forma crescente. Assim, ao algoritmo de otimização variar as soluções, a característica mais relevante do bloco (por exemplo, a transcondutância do OTA) será considerada dentro de uma lista organizada, o que é uma característica desejável dentro da otimização.

5.2.1.2 Tipos de Parâmetros em Nível de Transistores

No capítulo anterior, seção 4.1.1, os tipos de parâmetro que poderiam ser utilizadas em nível de transistor foram apresentados: *variáveis*, *iguais* ou *não-variáveis*. Neste capítulo, um novo tipo de parâmetro é acrescentado ao nível de transistores a fim de permitir que todas as soluções de uma fronteira de Pareto possuam variáveis com mesmo valor, mas que ainda assim sejam otimizadas durante o processo de síntese. Os parâmetros chamados *compartilhados* são os que precisam ser iguais em todas as soluções da fronteira de Pareto para que mais de uma solução possa ser usada simultaneamente no sistema. Assim, por exemplo, a tensão de polarização dos transistores em um sistema que use distribuição de tensões precisa ser a mesma. De modo semelhante, a mesma corrente de referência precisa ser utilizada para um sistema que use distribuição de correntes. Além de tensões e correntes, também podem ser compartilhadas dimensões de transistores para que o espelhamento de corrente seja feito apropriadamente.

Durante a criação da fronteira de Pareto utilizando o algoritmo descrito na seção 3.2, os parâmetros compartilhados serão considerados como variáveis apenas na fase 1. Assim, serão modificados apenas enquanto o algoritmo busca o mínimo global considerando os pesos das especificações atribuídos pelo projetista. Uma vez tendo encontrado a solução que melhor atenda às especificações, estes parâmetros serão fixadas para utilização durante a fase 2 e fase 3 do algoritmo. Como resultado, todas as soluções existentes na fronteira de Pareto terão os mesmos valores para os parâmetros compartilhados.

5.2.1.3 Características da Busca Usando Soluções Indexadas e Conectadas da Fronteira de Pareto

A restrição da escolha dos valores em nível de sistema aos já encontrados na fronteira de Pareto tem duas características vantajosas. A primeira é a redução do espaço de projeto e a segunda é a garantia de utilização de parâmetros de blocos coerentes entre si.

Quanto a redução do espaço de projeto, ela é dependente do número de soluções na fronteira de Pareto e da quantidade de variáveis conectadas. No caso de um conjunto de parâmetros conectados, apenas um deles é de fato uma variável de projeto, reduzindo assim o número de variáveis passadas ao otimizador.

Quanto a utilização de parâmetros do modelo coerentes entre si, isto se deve aos parâmetros terem sido extraídos diretamente de blocos dimensionados e funcionais. No caso em que a indexação não é realizada, alguns parâmetros do modelo possivelmente terão que ser arbitrados. Por exemplo, no macromodelo do OTA que será apresentado na seção 5.3.1.1, existem 5 intervalos de linearidade em função da amplitude do sinal de entrada (ΔV_1 , ΔV_2 , ΔV_3 , ΔV_4 e ΔV_5). Quando a fronteira de Pareto é utilizada para indexar as soluções, os intervalos ΔV_2 , ΔV_3 , ΔV_4 e ΔV_5 podem ser variáveis conectadas à variável ΔV_1 , garantindo que os valores destas variáveis serão coerentes. Quando não há indexação, não é vantajoso tornar todos estes intervalos variáveis livres, pois eles tem que seguir o comportamento padrão de um OTA a fim de corretamente reproduzir sua curva de transcondutância. Dessa forma, uma alternativa possível é utilizar apenas o primeiro intervalo (ΔV_1) como variável e calcular os outros arbitrariamente a partir dele (por exemplo: $\Delta V_2 = 1,2\Delta V_1$... $\Delta V_4 = 3\Delta V_1$).

Parâmetros de modelo como os discutidos no caso dos intervalos de linearidade, que não são necessariamente objetivos da síntese em nível de circuito mas que tem que ser armazenados para modelagem efetiva dos blocos, afetam o processo de aquisição da fronteira de Pareto. Restrições nas especificações de projeto, em geral, fazem parte deste tipo de parâmetro. Por exemplo, a otimização em nível de transistores de um amplificador pode ter uma restrição referente ao mínimo *slew rate* (taxa de variação da saída) para que o projeto seja interessante para a aplicação. Não sendo este um objetivo de projeto, mas sim uma restrição, a função custo multiobjetivo não retornaria o valor dela para compor o arquivo com a fronteira de Pareto. A restrição seria apenas considerada para elevar a função custo das outras medidas (conforme discutido na seção 4.8.1). Para utilizar o modelo do amplificador em um sistema, no entanto, a ausência desse tipo de informação pode limitar a capacidade das simulações de preverem o comportamento real do sistema com transistores. Para implementar essa operação, os algoritmos precisam ser adaptados a lerem um tipo de função custo com um conjunto de parâmetros adicional, que é atrelado às respostas das funções custo para compor o arquivo de saída com a fronteira de Pareto.

5.2.2 Modificação nas Informações de Entrada do Usuário

A definição da correspondência entre as variáveis de projeto e as fronteiras de Pareto é necessária para a execução da síntese em nível de sistema. Assim, o projetista precisa explicitar

Figura 5.2: Descrição da ligação das fronteiras de Pareto com o nível de sistema dentro do arquivo de topologia do sistema.

à ferramenta quais são as variáveis da fronteira de Pareto que serão consideradas e quais variáveis do projeto estarão submetidas a ela. Para tal, uma nova seção ao arquivo de topologia é adicionada no formato apresentado na Figura 5.2.

Além dessa modificação, uma descrição alternativa do circuito pode ser fornecida pelo o usuário caso ele deseje a geração automática das soluções com transistores para cada iteração. Essa descrição é a mesma *netlist* do circuito usando modelos em *Verilog-A*, apenas com a substituição destes por blocos com transistores.

5.2.3 Custo de Viabilidade

Quando otimizações exploratórias estão sendo realizadas, não é necessário utilizar apenas soluções disponíveis de antemão na fronteira de Pareto. Ainda assim, é possível utilizar as soluções encontradas na fronteira como referência sobre a sintetizabilidade do projeto. Para este fim, é criada uma função custo que mede as distâncias entre as especificações de blocos em nível de sistema e as solução existentes na fronteira de Pareto.

A viabilidade de um conjunto de especificações para determinado bloco é determinada pela dominação de alguma solução da fronteira de Pareto em relação a este conjunto. Para realizar a análise de viabilidade, uma matriz é criada com soluções da fronteira de Pareto nas linhas e objetivos nas colunas. Para os objetivos de maximização, o valor da coluna tem seu sinal invertido a fim de padronizar toda a análise para objetivos de minimização. Para esta análise ser feita apropriadamente, é recomendável separar os objetivos de aproximação em objetivos de maximização e minimização. Nesta matriz, se uma solução da fronteira de Pareto é menor ou igual a solução avaliada em todos objetivos, isso significa que esta solução da fronteira de Pareto domina a solução avaliada.

Uma solução dominada pela fronteira de Pareto é considerada viável. Essa viabilidade é decorrente do conhecimento prévio que uma solução igual ou melhor que a proposta em todos objetivos é possível de ser realizada, mesmo que eventualmente não seja possível produzir a solução com exatamente as mesmas especificações da solução proposta.

Para determinação da distância entre uma solução avaliada e a fronteira de Pareto, inicialmente é calculada a distância geométrica entre a solução proposta e cada solução da fronteira de Pareto. Caso a solução proposta não seja dominada por nenhuma das soluções da fronteira de Pareto, ela é considerada não viável. Nesse caso, é considerada para geração da função custo a menor distância entre uma solução da fronteira de Pareto e a solução proposta. Esta é solução mais próxima de atingir os objetivos requeridos pela solução proposta. A Figura 5.3 (a) mostra uma representação da escolha da menor distância geométrica para este caso. O custo de viabilidade será proporcional a esta distância. Apenas são considerados para compor esta distância geométrica os objetivos não atingidos.

Por outro lado, caso a solução proposta seja dominada por qualquer das soluções da fronteira de Pareto, ela é considerada viável. Nesse caso, a solução que possuir o maior valor da menor de suas componentes geométricas da distância é considerada. Esta solução é a que atinge os objetivos da solução proposta com maior margem de segurança. Na Figura 5.3 (b) está uma representação deste caso, no qual a solução p_1 é escolhida pois sua menor componente de distância (dy_1) é maior que a menor componente de distância de p_2 (dx_2). Caso a solução p_2 tenha uma variação de ordem similar a dx_2 , a solução proposta deixará de ser dominada por ela. O custo de viabilidade será proporcional a esta menor componente da distância geométrica, mas terá um valor negativo. Para o otimizador isso significará que é vantajoso aumentar esta distância.

Para cada topologia do sistema o processo descrito é realizado. O custo de viabilidade do sistema será a soma do custo de viabilidade de cada um dos blocos em relação a suas respectivas fronteiras de Pareto. A função custo total para o sistema será composta pelo custo das

Figura 5.3: Escolha da solução da fronteira de Pareto que servirá para calcular o custo de viabilidade da solução proposta para (a) caso em que a solução proposta é inviável e (b) caso em que a solução proposta é viável.

especificações do sistema e da viabilidade.

5.3 Síntese de Filtros G_m -C

O projeto de filtros G_m -C é uma boa aplicação para testar as funcionalidades da ferramenta de síntese visto a forte interligação existente entre as especificações do OTA e do sistema. Filtros G_m -C são utilizados em malha aberta, de forma que podem operar em frequências mais elevadas que os filtros Opamp-RC e MOSFET-C (SAARI, 2012). No trabalho de Ramachandran (2005), a criação de macromodelos de OTAs foi realizada e feita a síntese sistemática de filtros G_m -C. Hincapié (2010) elabora uma estratégia de projeto automático de filtros G_m -C utilizando programação geométrica. Ali (2009) realiza a síntese de um filtro G_m -C utilizando fronteiras de Pareto de maneira simplificada em relação a utilizada neste trabalho.

5.3.1 Operação e Estrutura de Filtros G_m -C

Filtros G_m -C são compostos de elementos transcondutores e capacitores. A Figura 5.4 mostra um integrador G_m -C. A corrente de saída do OTA é $i = g_m \cdot v_{entrada}$, onde g_m é a transcondutância. Devido a capacitância na saída, a função de transferência resultante é a exposta na equação 5.1.

Figura 5.4: Integrador G_m -C.

Figura 5.5: Estrutura de uma seção de segunda ordem.

$$\frac{v_{saída}}{v_{entrada}} = \frac{1}{s} \frac{g_m}{C} \quad (5.1)$$

A Figura 5.5 ilustra a estrutura de uma seção de segunda ordem, que tem sua função de transferência descrita na equação 5.2.

$$H_{\text{Seção 2ª Ordem Gm-C}}(s) = \frac{\frac{g_{m4}g_{m1}}{C_1C_2}}{s^2 + \frac{g_{m3}}{C_1}s + \frac{g_{m4}g_{m2}}{C_1C_2}} \quad (5.2)$$

É possível definir as relações entre transcondutâncias e capacitores para obter a função de transferência desejada para o filtro uma vez que uma função de transferência de 2ª ordem pode ser descrita pela equação 5.3:

$$H_{2^\text{ª Ordem}}(s) = \frac{K_0}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2} \quad (5.3)$$

onde K_0 ganho do filtro, ω_0 é a frequência angular de corte e Q é o fator de qualidade. Para seções de apenas um polo, a estrutura da Figura 5.6 é utilizada. A função de transferência é descrita na equação 5.4.

Figura 5.6: Estrutura de uma seção de um polo.

Figura 5.7: Exemplo de filtro de 6ª ordem composto por 3 seções de segunda ordem.

$$H_{\text{Seção de um polo}}(s) = \frac{\frac{g_{m1}}{C_1}}{s + \frac{g_{m2}}{C_1}} \quad (5.4)$$

Para criar filtros com ordens maiores, é possível utilizar o cascadeamento de seções de dois e de um polo ou fazer a emulação de filtros do tipo LC ladder. A Figura 5.7 mostra o exemplo de um filtro de 6ª ordem feito a partir do cascadeamento de 3 seções de segunda ordem. Nos exemplos que seguem, o cascadeamento de seções de segunda e primeira ordem será a estratégia adotada para composição de filtros de ordem maior que 2.

5.3.1.1 Modelagem do OTA

Um OTA ideal possui impedâncias de entrada e de saída infinitas. Ainda, sua transcondutância é constante, não sendo afetada por variações na amplitude ou frequência do sinal de entrada. No entanto, um OTA não ideal possui limitações nestas especificações, afetando, consequentemente, o sistema. Dessa forma, as especificações do OTA devem ser medidas no circuito durante a geração da fronteira de Pareto e modeladas para que possam ser utilizadas em nível de sistema.

A impedância de entrada dos OTAs é puramente capacitiva. Na saída, a impedância é capacitiva e resistiva. Assim, o modelo do OTA com estas impedâncias pode ser visto na Figura 5.8. Os procedimentos adotados para medição das impedâncias do OTA são descritos por Kar-

Figura 5.8: Modelo de OTA para pequenos sinais com impedâncias de entrada e saída finitas.

Figura 5.9: Modelo de OTA para pequenos sinais com fontes de ruído referidas à entrada e impedâncias de entrada e saída finitas.

dontchik (1992).

No filtro as capacitâncias parasitas de entrada e de saída do OTA são somadas às capacitâncias externas ao amplificador e afetam diretamente a função de transferência. Sabendo previamente quais são estas capacitâncias parasitas, é possível subtrair o seu valor das capacitâncias do filtro de forma a preservar a função de transferência desejada. Devido às variações de valor das capacitâncias parasitas em função de variações de processo, é recomendável manter a relação entre capacitância parasita e capacitância total de um nó a menor possível.

Um OTA pode ter diferentes tipos de fonte de ruído. Cada fonte de ruído pode ser adicionada referida à entrada do modelo. A Figura 5.9 mostra o modelo do OTA não ideal com adição das fontes de ruído térmico e *flicker*. Na Figura 5.10, é mostrado como é realizada a adição do ruído térmico (linha 1) e do ruído *flicker* (linha 2) no modelo do OTA em *Verilog-A*.

Outra não idealidade do OTA é a relação entre a transcondutância e a variação da amplitude de entrada. A medida que a amplitude do sinal de entrada aumenta, o valor de transcondutância é reduzido em relação ao seu valor máximo. Essa distorção do OTA pode afetar diretamente o SFDR do filtro.

Para medir a linearidade do OTA em função da amplitude de entrada, foram adotadas as medições de 5 intervalos de tensão em torno do valor de sua transcondutância máxima. Posteriormente, estes intervalos são utilizados para definir uma *spline*¹ utilizada para modelar a relação entre tensão e corrente no OTA descrito em *Verilog-A*. Assim, é possível reproduzir as distorções na corrente e consequentemente na transcondutância causadas pelo incremento da amplitude do sinal de entrada. Os segmentos da *spline* devem possuir uma ordem suficiente para garantir a aproximação da função desejada e suavidade entre as partes. Para a presente

¹função polinomial dividida em partes

```

1 V(vin_p, vin_n) <+ white_noise(irtnoise, "térmico");
2 V(vin_p, vin_n) <+ flicker_noise(irfnoise, 1, "flicker")

```

Figura 5.10: Inclusão de ruído térmico e *flicker* no modelo do OTA em Verilog-A.

Figura 5.11: Intervalos de linearidade da corrente em função da tensão de entrada no OTA.

aplicação, foi utilizada uma *spline* de 3ª ordem. A escolha de modelar a corrente com *splines* é devida ao alto grau necessário para modelá-los apenas com uma função polinomial, o que resulta em instabilidade na função. Outra alternativa poderia ser utilizar tabelas de busca, mas isto resultaria em um número de pontos excessivo a serem coletados para modelar a corrente. Assim, o arquivo contendo todas as soluções da fronteira de Pareto teria dimensão muito superior a necessária para modelar utilizando *splines*.

As medições são realizadas através da varredura da tensão de um dos terminais de entrada do OTA de V_{SS} até V_{DD} . Um bloco em Verilog-A conectado a saída do OTA mede as variações de corrente na saída e retorna os intervalos em que g_m está tem até 5%, 25%, 50%, 95% e 99,9% de diferença em relação ao valor máximo de g_m nos valores ΔV_1 , ΔV_2 , ΔV_3 , ΔV_4 e ΔV_5 respectivamente. A Figura 5.11 mostra estes intervalos.

Para construir a *spline* toma-se proveito do valor da corrente em função da tensão ser uma função ímpar. Assim, é possível modelar apenas a função a partir da tensão 0, e ao reconstruí-la no modelo, fazer o espelhamento para valores negativos. A equação da corrente de cada uma

Tabela 5.1: Condições para gerar coeficientes da *spline*.

Primeira Parte		Segunda Parte			
$I_a \Big _{v=0} = 0$		$I_b \Big _{v=\frac{\Delta V_1}{2}} = I_a \Big _{v=\frac{\Delta V_1}{2}}$			
$\frac{\partial I_a}{\partial v} \Big _{v=0} = g_m \text{ máx}$		$\frac{\partial I_b}{\partial v} \Big _{v=\frac{\Delta V_1}{2}} = 0,95 g_m \text{ máx}$			
$\frac{\partial^2 I_a}{\partial v} \Big _{v=0} = 0$		$\frac{\partial^2 I_b}{\partial v} \Big _{v=\frac{\Delta V_1}{2}} = \frac{\partial^2 I_a}{\partial v} \Big _{v=\frac{\Delta V_1}{2}}$			
$\frac{\partial I_a}{\partial v} \Big _{v=\frac{\Delta V_1}{2}} = 0,95 g_m \text{ máx}$		$\frac{\partial I_b}{\partial v} \Big _{v=\frac{\Delta V_2}{2}} = 0,75 g_m \text{ máx}$			
Terceira Parte		Quarta Parte		Quinta Parte	
$I_c \Big _{v=\frac{\Delta V_2}{2}} = I_b \Big _{v=\frac{\Delta V_2}{2}}$		$I_d \Big _{v=\frac{\Delta V_3}{2}} = I_c \Big _{v=\frac{\Delta V_3}{2}}$		$I_e \Big _{v=\frac{\Delta V_4}{2}} = I_d \Big _{v=\frac{\Delta V_4}{2}}$	
$\frac{\partial I_c}{\partial v} \Big _{v=\frac{\Delta V_2}{2}} = 0,75 g_m \text{ máx}$		$\frac{\partial I_d}{\partial v} \Big _{v=\frac{\Delta V_3}{2}} = 0,5 g_m \text{ máx}$		$\frac{\partial I_e}{\partial v} \Big _{v=\frac{\Delta V_4}{2}} = 0,05 g_m \text{ máx}$	
$\frac{\partial^2 I_c}{\partial v} \Big _{v=\frac{\Delta V_2}{2}} = \frac{\partial^2 I_b}{\partial v} \Big _{v=\frac{\Delta V_2}{2}}$		$\frac{\partial^2 I_d}{\partial v} \Big _{v=\frac{\Delta V_3}{2}} = \frac{\partial^2 I_c}{\partial v} \Big _{v=\frac{\Delta V_3}{2}}$		$\frac{\partial^2 I_e}{\partial v} \Big _{v=\frac{\Delta V_4}{2}} = \frac{\partial^2 I_d}{\partial v} \Big _{v=\frac{\Delta V_4}{2}}$	
$\frac{\partial I_c}{\partial v} \Big _{v=\frac{\Delta V_3}{2}} = 0,5 g_m \text{ máx}$		$\frac{\partial I_d}{\partial v} \Big _{v=\frac{\Delta V_4}{2}} = 0,05 g_m \text{ máx}$		$I_e \Big _{v=\frac{\Delta V_5}{2}} = I_{\text{máx}} - I_{\text{média}}$	

das 4 partes da *spline* podem ser vistas nas equações 5.5, 5.6, 5.7, 5.8 e 5.9:

$$I_a = a_0 + a_1 v + a_2 v^2 + a_3 v^3 \quad (5.5)$$

$$I_b = b_0 + b_1 v + b_2 v^2 + b_3 v^3 \quad (5.6)$$

$$I_c = c_0 + c_1 v + c_2 v^2 + c_3 v^3 \quad (5.7)$$

$$I_d = d_0 + d_1 v + d_2 v^2 + d_3 v^3 \quad (5.8)$$

$$I_e = e_0 + e_1 v + e_2 v^2 + e_3 v^3 \quad (5.9)$$

onde I_a até I_e são as correntes dos diferentes segmentos da *spline* e os parâmetros a_0 a a_3 , b_0 a b_3 , c_0 a c_3 , d_0 a d_3 e e_0 a e_3 são os coeficientes das respectivas partes da *spline*.

As condições definidas para gerar os coeficientes da *spline* estão expostas na Tabela 5.1. Nessas condições, o $g_m \text{ máx}$ é valor medido da transcondutância máxima, $I_{\text{máx}}$ é a corrente máxima medida na saída do OTA e $I_{\text{média}}$ é a corrente média. Esses critérios, juntamente com os intervalos e correntes medidas, são utilizados para criação dos coeficientes que definem a *spline* no modelo do OTA em Verilog-A.

O circuito de *common-mode feedback* é otimizado juntamente com o OTA devido ao compartilhamento de variáveis entre eles. Dessa forma, são usadas chaves ideais para refazer as conexões internas em função do *testbench* que está sendo utilizado. O circuito com as chaves ideais pode ser visto na Figura 5.12. Para as medições do OTA isoladamente, todas os parâmetros de chave (*cmfb_on*, *cmfb_loop* e *cmfb_in*) são configurados para o valor 0. Para as medições de impedância do circuito incluindo CMFB, todas os parâmetros de chaves são con-

Figura 5.12: Circuito OTA e CMFB com chaves para realizar caracterização.

figurados para o valor 1. Para as medições do CMFB em malha aberta, o parâmetro *cmfb_on* é 1, o parâmetro *cmfb_loop* é 1 e o parâmetro *cmfb_in* é 2.

Por fim, são adicionadas também as medições de corrente consumida e a área do OTA, bem como do conjunto OTA + CMFB, para que possam ser usados para estimar o consumo e área total do sistema.

5.4 Utilização da ferramenta para diferentes estratégias de projeto

Definições em nível de sistema influenciam o espaço de projeto dos blocos que compõe a fronteira de Pareto. Dessa forma, a ferramenta precisa ser flexível para permitir que estas definições sejam implementáveis facilmente no fluxo de projeto. No caso do filtros G_m -C, algumas definições podem ser, por exemplo:

- Uso de OTAs iguais ou distintos;
- Uso de distribuição de tensão ou corrente de polarização no sistema;

Para o uso de OTAs iguais no sistema, não há compromisso entre as diversas soluções da fronteira de Pareto uma vez que a solução que for escolhida para compor o sistema não terá que interagir com soluções diferentes dela. Assim, questões como tensão de polarização e de modo comum podem ser definidas individualmente para cada solução da fronteira de Pareto. Ainda que apenas um tipo de OTA seja usado para todas as instâncias do filtro, a multiplicidade de cada OTA pode ser variada para gerar transcondutâncias diferentes.

Para o uso de OTAs diferentes em um mesmo filtro, usualmente estes OTAs compartilham o circuito de polarização. Assim, após uma fase inicial de otimização na qual todas as variáveis são alteráveis, o uso de parâmetros compartilhados (descritas na seção 5.2.1) limita a variação

das soluções para que a fronteira de Pareto seja composta apenas de soluções compatíveis entre si. Os parâmetros que serão compartilhados são consequência da forma de polarização do sistema.

Para sistemas com distribuição de tensão de polarização, o circuito que gera a tensão é comum a todos os blocos. Dessa forma, para que a corrente de referência seja corretamente replicada (ou multiplicada) nos blocos, estes precisam ter os transistores dos espelhos de corrente (que recebem as tensões de polarização) iguais ou múltiplos dos transistores usados para gerar as tensões. Durante a configuração da ferramenta de síntese, todos os parâmetros do circuito de polarização são categorizados como sendo do tipo *compartilhados*, bem como o comprimento e largura dos transistores que recebem estas tensões. Assim, resta apenas a multiplicidade destes transistores como variáveis de polarização dos blocos nas fases avançadas de projeto.

Para sistemas com distribuição de corrente de polarização, cada bloco recebe uma corrente com valor comum a todos os blocos e gera suas próprias tensões de polarização a partir dela. Neste caso, a restrição de projeto para os blocos é possuir a mesma corrente de referência, sendo este o único parâmetro compartilhado entre os blocos.

5.5 Síntese de um Filtro G_m -C Passa-Baixas com Frequência de Corte de 1,2 MHz em Tecnologia 0,35 μm

Nesta seção é realizada a síntese de um filtro passa-baixas com frequência de corte em 1,2 MHz. O projeto é realizado com as especificações similares às definidas na dissertação de Hincapié (2010), que realizou a síntese utilizando Programação Geométrica, a fim de propiciar uma comparação dos resultados das técnicas. Para este projeto são usados OTAs distintos entre si mas com mesma topologia, conforme Figura 5.12.

Foi utilizada a tecnologia AMS 0,35 μm com tensão de alimentação +1,65 V e -1,65 V para esta síntese. As especificações do filtro estão na Tabela 5.2, a partir das quais é determinado um filtro *Butterworth* de ordem 3 e com a função de transferência descrita na equação 5.10.

$$FT(s) = \frac{4,286 \cdot 10^{20}}{s^3 + 1,508 \cdot 10^7 s^2 + 1,137 \cdot 10^{14} s + 4,286 \cdot 10^{20}} \quad (5.10)$$

Tabela 5.2: Especificação de filtro G_m -C passa-baixas com frequência de corte em 1,2 MHz (CMOS 0,35 μm).

Característica	Especificação
Tipo	Passa-Baixas
Protótipo	Butterworth
Frequência de Corte	1,2 MHz $\pm 5\%$
Ganho	1 $\pm 5\%$
Frequência de Rejeição	5 MHz
Rejeição Máx. na Faixa de Passagem	≤ 1 dB
Rejeição Min. na Faixa de Rejeição	≥ 30 dB
Potência	≤ 100 mW
Area	≤ 10 mm ²
Relação entre capacitância parasita e a capacitância total dos nós do filtro	$\leq 0,12$
SFDR	≥ 50 dB e maximizar

As raízes do filtro são:

$$\begin{aligned}
 & -7,5398 \cdot 10^6 \\
 & (-3,7699 + 6,5297i) \cdot 10^6 \\
 & (-3,7699 - 6,5297i) \cdot 10^6
 \end{aligned}$$

O filtro de 3ª ordem pode ser implementado com uma seção de dois polos e uma seção de um polo. As funções de transferência destes blocos são determinadas a partir da separação de polos da função de transferência total. Assim, as funções resultantes são:

$$FT_{2^a\text{Ordem}}(s) = \frac{5,684 \cdot 10^{13}}{s^2 + 7,540 \cdot 10^6 s + 5,684 \cdot 10^{13}} \quad (5.11)$$

$$FT_{1^a\text{Ordem}}(s) = \frac{7,540 \cdot 10^6}{s + 7,540 \cdot 10^6} \quad (5.12)$$

Identificando nas funções de transferência resultantes suas correspondências com a função de transferência das equações 5.2 e 5.4, as relações de transcondutância e capacitância para os blocos são obtidas. Fazendo $g_{m2} = g_{m1}$ para todos os blocos, as relações para a seção de segunda ordem são descritas nas equações 5.13 e 5.14.

$$\frac{g_{m3}}{C_1} = 7,540 \cdot 10^6 \quad (5.13)$$

$$\frac{g_{m1}g_{m4}}{C_1C_2} = 5,685 \cdot 10^{13} \quad (5.14)$$

Tabela 5.3: Especificações mínimas do OTA para geração da fronteira de Pareto do filtro com frequência de corte de 1,2 MHz (CMOS 0,35 μm).

Medida	Especificação	Tipo
g_m [μS]	$> 0,1$ e < 10000	Objetivo
corrente consumida _(OTA) [μA]	< 600	Objetivo
ruído térmico referido à entrada _(OTA) [nV]	< 100	Objetivo
ruído <i>flicker</i> referido à entrada _(OTA) [μV]	< 100	Objetivo
faixa com até 95% de linearidade [mV]	> 25	Objetivo
$C_{\text{Saída (OTA)}}$ [pF]	< 1	Objetivo
C_{Entrada} [pF]	< 1	Objetivo
$R_{\text{Saída (OTA)}}$ [$\text{M}\Omega$]	$> 0,1$	Objetivo
Area [mm^2]	< 1	Objetivo
$C_{\text{Saída (OTA+CMFB)}}$ [pF]	< 1	Restrição
$R_{\text{Saída (OTA+CMFB)}}$ [$\text{M}\Omega$]	$> 0,1$	Restrição
ruído térmico referido à entrada _(OTA+CMFB) [nV]	< 100	Restrição
ruído <i>flicker</i> referido à entrada _(OTA+CMFB) [μV]	< 100	Restrição
<i>offset</i> [μV]	< 10	Restrição
Diferença entre $I_{\text{saída pos. min}}$ e $I_{\text{saída neg. min}}$ [%]	$< 0,3$	Restrição
Frequência de ganho unitário [MHz]	> 6	Restrição
Medidas adicionais para modelagem		
area _(OTA+CMFB) [μm^2]		
corrente consumida _(OTA+CMFB) [μA]		
demais faixas de linearidade (75%, 50%, 5% e 0,01%) [mV]		
$I_{\text{saída média}}$		
$I_{\text{saída min.}}$		
$I_{\text{saída max.}}$		

A relação entre transcondutância e capacitância para o bloco de primeira ordem é descrita na equação 5.15.

$$\frac{g_{m1}}{C_1} = 7,540 \cdot 10^6 \quad (5.15)$$

A geração da fronteira de Pareto do bloco OTA utiliza as especificações da Tabela 5.3. Em adição, todos os transistores devem estar operando em saturação. Os parâmetros de projeto estão descritas na Tabela 5.4, onde é possível observar que todos os parâmetros do circuito de polarização são do tipo *compartilhado*. A forma como estas variáveis são utilizadas para determinação das dimensões dos transistores é descrita na Tabela 5.5. As relações de multiplicidade dos transistores visam manter as tensões de porta proporcionais a corrente necessária para cada ramo.

A busca da fronteira de Pareto para o bloco OTA é limitada em 5000 iterações. Após a fase inicial da otimização são obtidos os valores das variáveis compartilhadas. Os valores obtidos para estas variáveis são: $I_1 = 70,7 \mu\text{A}$; $V_{CM\text{ REF}} = 1,26 \text{ V}$; $W_3 = 13,3 \mu\text{m}$; $L_3 = 8,7 \mu\text{m}$; $W_5 = 41,9 \mu\text{m}$; $L_5 = 17 \mu\text{m}$ e $N_6 = 6$; $N_{17} = 1$. Então, essas variáveis são fixadas e o otimizador realiza a busca pela aproximação da fronteira de Pareto. A Figura 5.13 mostra as 484 soluções encontradas no espaço de critério. A forma na qual são construídos os gráficos e como é possível

Tabela 5.4: Características dos parâmetros do OTA para criação da fronteira de Pareto do filtro com frequência de corte de 1,2 MHz (CMOS 0,35 μm).

Nome	Tipo de Número	Tipo de Parâmetro	Intervalo	Passo
I_1	grade	var. compartilhada	[100 nA - 100 μA]	100 nA
$V_{CM\ REF}$	real	var. compartilhada	[1,15V 2,14V]	contínuo
W_1	real	variável	[0,6 μm - 100 μm]	contínuo
L_1	real	variável	[0,35 μm - 100 μm]	contínuo
N_1	inteiro	variável	[1 10]	1
W_3	real	var. compartilhada	[0,6 μm - 100 μm]	contínuo
L_3	real	var. compartilhada	[0,35 μm - 100 μm]	contínuo
W_5	real	var. compartilhada	[0,6 μm - 100 μm]	contínuo
L_5	real	var. compartilhada	[0,35 μm - 100 μm]	contínuo
N_{5b}	inteiro	variável	[1 - 20]	1
N_6	grade	var. compartilhada	[2 - 10]	2
N_{17}	inteiro	var. compartilhada	[1 - 10]	1
W_9	real	variável	[0,6 μm - 100 μm]	contínuo
L_9	real	variável	[0,35 μm - 100 μm]	contínuo
N_9	inteiro	variável	[1 - 10]	1

Tabela 5.5: Relação entre variáveis de projeto e dimensões dos transistores do OTA para criação da fronteira de Pareto do filtro com frequência de corte de 1,2 MHz (CMOS 0,35 μm).

Transistor	W	L	Multiplicidade
M_1 e M_2	W_1	L_1	N_1
$M_3, M_4, M_{13}, M_{14}, M_{15}$ e M_{16}	W_3	L_3	$N_{5b} \cdot N_{17}$
M_5, M_7 e M_8	W_5	L_5	$N_{5b} \cdot N_6$
M_6	W_5	L_5	N_6
M_9, M_{10}, M_{11} e M_{12}	W_9	L_9	N_9
M_{17}	W_3	L_3	N_{17}
M_{18}	W_5	L_5	$N_6/2$

interpretá-los foi discutida na seção 4.8.2.

Para a abordagem utilizada neste caso (OTAs distintos mas com mesma topologia), cada OTA é relacionado a mesma fronteira de Pareto em nível de sistema, mas soluções distintas dessa fronteira podem ser utilizadas em diferentes instâncias do OTA dentro de um mesmo filtro (caso b da Figura 5.1). Ainda, nessa síntese é realizada uma busca direta na fronteira de Pareto, sem permitir que as variáveis assumam valores diferentes dos existentes na fronteira. Para tanto, a declaração das variáveis em nível de sistema é feita conforme a Tabela 5.6. Nesta tabela, é importante observar que o único parâmetro variável para cada OTA é a transcondutância. Os demais parâmetros do OTA são conectados a este, de forma que o índice da transcondutância de um OTA na fronteira de Pareto determinará o índice dos outros parâmetros.

Os parâmetros finais dos transcondutores após a síntese podem ser vistos na Tabela 5.7. Os resultados da síntese podem ser vista na Tabela 5.8. Há uma grande similaridade entre os

Figura 5.13: Fronteira de Pareto do OTA para o filtro com frequência de corte de 1,2 MHz (CMOS 0,35 μm).

resultados da simulação *Verilog-A* e a simulação com transistores no *HSPICE*. Isso indica que a modelagem foi realizada corretamente e serve para representar o circuito real.

Na Tabela 5.9 é mostrada uma comparação entre o resultado deste trabalho e o do trabalho apresentado por Hincapié (2010), que utilizou programação geométrica. É possível perceber que ambos atingiram especificações parecidas. Apesar disso, a solução da ferramenta proposta obteve SFDR superior em decorrência de sua solução final ter obtido menor ruído. É importante ressaltar que a forma de medição das não linearidades do circuito pode ter grande impacto nos resultados.

5.6 Síntese de um Filtro G_m -C Passa-Baixas com Frequência de Corte de 20 MHz em Tecnologia 0,35 μm

Nesta seção será sintetizado um filtro com frequência de corte de 20 MHz e com as especificações descritas na Tabela 5.10. A tecnologia utilizada é a AMS 0,35 μm com tensão de

Tabela 5.6: Descrição dos parâmetros em nível de sistema para síntese do filtro com frequência de corte de 1,2 MHz (CMOS 0,35 μm).

Nome	Tipo de Número	Tipo de Parâmetro	Variável a qual está conectada	Descrição
<i>b1_gm1</i>	indexado	variável	-	Transcondutância do OTA 1 da seção de 2ª ordem
<i>b1_tnoise1</i>	indexado	conectado	<i>b1_gm1</i>	Ruído térmico do OTA 1 da seção de 2ª ordem
<i>b1_fnoise1</i>	indexado	conectado	<i>b1_gm1</i>	Ruído <i>flicker</i> do OTA 1 da seção de 2ª ordem
<i>b1_deltav1</i>	indexado	conectado	<i>b1_gm1</i>	Intervalo de linearidade 1 do OTA 1 da seção de 2ª ordem
<i>b1_deltav2</i>	indexado	conectado	<i>b1_gm1</i>	Intervalo de linearidade 2 do OTA 1 da seção de 2ª ordem
<i>b1_deltav3</i>	indexado	conectado	<i>b1_gm1</i>	Intervalo de linearidade 3 do OTA 1 da seção de 2ª ordem
<i>b1_deltav4</i>	indexado	conectado	<i>b1_gm1</i>	Intervalo de linearidade 4 do OTA 1 da seção de 2ª ordem
<i>b1_deltav5</i>	indexado	conectado	<i>b1_gm1</i>	Intervalo de linearidade 5 do OTA 1 da seção de 2ª ordem
<i>b1_cout1</i>	indexado	conectado	<i>b1_gm1</i>	Capacitância de saída do OTA 1 da seção de 2ª ordem
<i>b1_rout1</i>	indexado	conectado	<i>b1_gm1</i>	Resistência de saída do OTA 1 da seção de 2ª ordem
<i>b1_isupply1</i>	indexado	conectado	<i>b1_gm1</i>	Corrente consumida do OTA 1 da seção de 2ª ordem
<i>b1_area1</i>	indexado	conectado	<i>b1_gm1</i>	Area do OTA 1 da seção de 2ª ordem
<i>b1_cin1</i>	indexado	conectado	<i>b1_gm1</i>	Capacitância de entrada do OTA 1 da seção de 2ª ordem
<i>b1_maxc1</i>	indexado	conectado	<i>b1_gm1</i>	Corrente máxima de saída do OTA 1 da seção de 2ª ordem
<i>b1_midc1</i>	indexado	conectado	<i>b1_gm1</i>	Corrente média de saída do OTA 1 da seção de 2ª ordem
<i>b1_coutcmfb1</i>	indexado	conectado	<i>b1_gm1</i>	Capacitância de saída do OTA+CMFB 1 da seção de 2ª ordem
<i>b1_routcmfb1</i>	indexado	conectado	<i>b1_gm1</i>	Resistência de saída do OTA+CMFB 1 da seção de 2ª ordem
<i>b1_areacmfb1</i>	indexado	conectado	<i>b1_gm1</i>	Area do OTA+CMFB 1 da seção de 2ª ordem
<i>b1_gm3</i>	indexado	variável	-	Transcondutância do OTA 3 da seção de 2ª ordem
<i>b1_tnoise1</i>	indexado	conectado	<i>b1_gm3</i>	Ruído térmico do OTA 3 da seção de 2ª ordem
<i>b1_fnoise1</i>	indexado	conectado	<i>b1_gm3</i>	Ruído <i>flicker</i> do OTA 3 da seção de 2ª ordem
⋮	⋮	⋮	⋮	demais variáveis do OTA 3 da seção de 2ª ordem são conectadas ao <i>b1_gm3</i>
<i>b1_gm4</i>	indexado	variável	-	Transcondutância do OTA 4 da seção de 2ª ordem
<i>b1_tnoise4</i>	indexado	conectado	<i>b1_gm4</i>	Ruído térmico do OTA 4 da seção de 2ª ordem
<i>b1_fnoise4</i>	indexado	conectado	<i>b1_gm4</i>	Ruído <i>flicker</i> do OTA 4 da seção de 2ª ordem
⋮	⋮	⋮	⋮	demais variáveis do OTA 4 da seção de 2ª ordem são conectadas ao <i>b1_gm4</i>
<i>s1_gm1</i>	indexado	variável	-	Transcondutância do OTA 1 da seção de 1 polo
<i>s1_tnoise1</i>	indexado	conectado	<i>s1_gm1</i>	Ruído térmico do OTA 1 da seção de 1 polo
<i>s1_fnoise1</i>	indexado	conectado	<i>s1_gm1</i>	Ruído <i>flicker</i> do OTA 1 da seção de 1 polo
⋮	⋮	⋮	⋮	demais variáveis do OTA 1 da seção de 1 polo são conectadas ao <i>s1_gm1</i>

alimentação +1,65 V e -1,65 V. A partir das especificações, é escolhida para implementação do filtro a topologia *Butterworth* de ordem 5 e com a função de transferência descrita na equação 5.16.

$$FT(s) = \frac{3,134 \cdot 10^{40}}{s^5 + 4,067 \cdot 10^8 s^4 + 8,268 \cdot 10^3 s^3 + 1,039 \cdot 10^{25} s^2 + 8,07 \cdot 10^{32} s + 3,134 \cdot 10^{40}} \quad (5.16)$$

As raízes do filtro são:

$$\begin{aligned} & -1,2566 \cdot 10^8 \\ & (-0,3883 + 1,1951i) \cdot 10^8 \quad (-0,3883 - 1,1951i) \cdot 10^8 \\ & (-1,0166 + 0,7386i) \cdot 10^8 \quad (-1,0166 - 0,7386i) \cdot 10^8 \end{aligned}$$

Tabela 5.7: Especificações dos blocos OTA após síntese do filtro G_m -C com frequência de corte de 1,2 MHz (CMOS 0,35 μm).

Parâmetro *	Seção de 2 polos		Seção de 1 polo	
	OTAs 1 e 2	OTA 3	OTA 4	OTAs 1 e 2
g_m [μS]	1520	1840	1060	944
ruído térmico referido à entrada _(OTA) ** [nV]	3,23	2,79	4,33	4,33
ruído <i>flicker</i> referido à entrada _(OTA) ** [μV]	1,05	0,89	1,31	1,17
faixa de entrada *** [mV]	200	120	370	280
$C_{\text{Saída (OTA)}}$ [fF]	358	475	256	261
C_{Entrada} [fF]	818	958	612	808
$R_{\text{Saída (OTA)}}$ [$\text{k}\Omega$]	107	100	112	174
$\text{Area}_{\text{(OTA)}}$ [μm^2]	55200	46400	68600	46200
I consumida _(OTA) [μA]	848	707	1050	2110
$C_{\text{Saída (OTA+CMFB)}}$ [fF]	422	-	344	261
$\text{Area}_{\text{(OTA+CMFB)}}$ [μm^2]	164000	-	204000	136000
I consumida _(OTA+CMFB) [μA]	2540	-	3180	2110

* Dados sobre CMFB são utilizados apenas para OTA 1 e OTA 4 da seção de dois polos e OTA 1 da seção de 1 polo

** referido à entrada

*** para variação de até 5% do valor de g_m máximo

Tabela 5.8: Especificações e resultados da síntese do filtro G_m -C com frequência de corte de 1,2 MHz (CMOS 0,35 μm).

Nome da Especificação	Objetivo	Simulação Verilog-A	Simulação HSPICE
Ganho DC [V]	$1 \pm 5\%$	0,98	0,97
Frequência de -3 dB [MHz]	$1,2 \pm 5\%$	1,19	1,19
Rejeição em 5 MHz [dB]	> 30	37	37
Potência [mW]	minimizar	33,3	33,7
Area [mm^2]	minimizar	1,43	1,43
SFDR [dB]	> 50	60,5	59,7
IIP3 [dBm]	-	7,7	6,6
Densidade de ruído referido à entrada [$\text{nV}/\sqrt{\text{Hz}}$]	-	14,37	14,48
χ^* do capacitor 1 da seção de 2ª ordem	$< 0,12$	0,012	0,012
χ^* do capacitor 2 da seção de 2ª ordem	$< 0,12$	0,017	0,017
χ^* do capacitor da seção de 1 polo	$< 0,12$	0,011	0,011
Atraso de grupo médio [n s]	-	310	310
Variação de atraso de grupo [n s]	-	92	91

* χ definido como a relação entre capacitância parasita e a capacitância total

O filtro de 5ª ordem pode ser implementado com duas seções de 2ª ordem e uma seção de um polo. As funções de transferência destes blocos são determinadas a partir da separação de polos da função de transferência total. Assim, as funções resultantes são:

$$FT_{2^a \text{ Ordem } 1}(s) = \frac{1,579 \cdot 10^{16}}{s^2 + 7,766 \cdot 10^7 s + 1,579 \cdot 10^{16}} \quad (5.17)$$

Tabela 5.9: Comparação dos resultados da síntese do filtro G_m -C com frequência de corte de 1,2 MHz entre a ferramenta deste trabalho e a ferramenta de Hincapié (2010) em tecnologia 0,35 μm .

Nome da Especificação	Objetivo neste trabalho	Resultado neste trabalho	Resultado no trabalho de Hincapié (2010)
Ganho DC	$1 \pm 0,05$	0,97	0,99
Frequência de -3 dB [MHz]	$1,20 \pm 5\%$	1,19	1,3
Ruído RMS referido à entrada [μV]	-	15,7	63,3
SFDR [dB]	> 50	59,7	56

Tabela 5.10: Especificação de filtro G_m -C passa-baixas com frequência de corte de 20 MHz em tecnologia 0,35 μm .

Característica	Especificação
Tipo	Passa-Baixas
Protótipo	Butterworth
Frequência de Corte	$20 \text{ MHz} \pm 5\%$
Ganho	$1 \pm 5\%$
Frequência de Rejeição	90 MHz
Rejeição Máx. na Faixa de Passagem	$< 1 \text{ dB}$
Rejeição Min. na Faixa de Rejeição	$> 50 \text{ dB}$
Potência	$\leq 100 \text{ mW}$ e minimizar
Area	$\leq 10 \text{ mm}^2$ e minimizar
Relação entre capacitância parasita e a capacitância total dos nós do filtro	$\leq 0,3$
SFDR em banda	$\geq 50 \text{ dB}$ e maximizar

$$FT_{2^a \text{ Ordem}}(s) = \frac{1,579 \cdot 10^{16}}{s^2 + 2,033 \cdot 10^8 s + 1,579 \cdot 10^{16}} \quad (5.18)$$

$$FT_{1^a \text{ Ordem}}(s) = \frac{1,256 \cdot 10^8}{s + 1,256 \cdot 10^8} \quad (5.19)$$

Fazendo $g_{m2} = g_{m1}$ para todos os blocos, as relações para a primeira seção de 2ª ordem são descritas nas equações 5.20 e 5.21:

$$\frac{g_{m3}}{C_1} = 1,579 \cdot 10^{16} \quad (5.20)$$

$$\frac{g_{m1}g_{m4}}{C_1C_2} = 7,766 \cdot 10^7 \quad (5.21)$$

Para a seção de segunda seção de 2ª ordem, as relações são descritas nas equações 5.22 e 5.23:

$$\frac{g_{m3}}{C_1} = 1,579 \cdot 10^{16} \quad (5.22)$$

$$\frac{g_{m1}g_{m4}}{C_1C_2} = 2,033 \cdot 10^8 \quad (5.23)$$

A relação entre transcondutância e capacitância para o bloco de primeira ordem é descrita na equação 5.24:

$$\frac{g_{m1}}{C_1} = 1,256 \cdot 10^8 \quad (5.24)$$

A fronteira de Pareto do OTA utiliza especificações similares às da síntese para o filtro em 1,2 MHz. As modificações são a adição da restrição da frequência de ganho unitário ser superior a 100 MHz e a conversão das variáveis compartilhadas em variáveis comuns. A relação das variáveis do OTA com as dimensões dos transistores também são as mesmas descritas para o filtro em 1,2 MHz.

Para esta síntese, foi definido que todos os OTAs do filtro são iguais, visto que esta é uma estratégia comumente adotada para facilitar a sintonização (*tuning*) do sistema. Assim, para cada iteração durante a síntese, apenas uma solução da fronteira de Pareto é selecionada e a multiplicidade de cada OTA é a variável utilizada para possibilitar transcondutâncias diferentes para diferentes OTAs. No que diz respeito à geração da fronteira de Pareto do transcondutor, isso significa que não há necessidade do uso de variáveis compartilhadas uma vez que as soluções da fronteira de Pareto não precisarão interagir entre si no circuito final. A fronteira de Pareto obtida pode ser vista na Figura 5.14, sendo composta por 696 soluções.

Da mesma forma como na síntese para o filtro com frequência de corte em 1,2 MHz, nessa síntese foi permitida apenas uma busca direta na fronteira de Pareto sem permitir que os OTAs assumam valores diferentes dos existentes na fronteira. A Tabela 5.11 mostra a descrição dos parâmetros em nível de sistema. Em relação a Tabela 5.6, anteriormente apresentada para a síntese em 1,2 MHz, é possível observar a adição dos OTAs da segunda seção de 2ª ordem, e da variável *toposeq*. Esta última variável indica a posição das seções de 2 polos e seção de 1 polo no caminho do sinal do filtro. Assim, utilizando o recurso de impressão condicional descrito na seção 4.1.1, é possível determinar qual sequência dos blocos será utilizada entre as abaixo:

$$\begin{aligned} &\{b1, b2, s1\}, \\ &\{b1, s1, b2\}, \\ &\{b2, b1, s1\}, \\ &\{b2, s1, b1\}, \\ &\{s1, b1, b2\}, \\ &\{s1, b2, b1\} \end{aligned} \quad (5.25)$$

Figura 5.14: Fronteira de Pareto do OTA sem variáveis compartilhadas para o filtro com frequência de corte de 20 MHz em tecnologia 0,35 μm .

onde $b1$ representa a primeira seção de dois polos, $b2$ representa a segunda seção de dois polos e $s1$ representa a seção de 1 polo.

Os parâmetros finais encontrados para os OTAs do filtro com frequência de corte de 20 MHz podem ser visto na Tabela 5.12. Como o mesmo OTA é utilizado para todas as soluções, apenas a multiplicidade causa diferentes transcondutâncias nos blocos. A multiplicidade dos OTAs pode ser vista na tabela 5.13.

As medições resultantes do filtro simulado com modelos em *Verilog-A* e eletricamente no *HSPICE* podem ser vistas na Tabela 5.14. As especificações foram atendidas e, da mesma forma como no caso anterior, há uma grande similaridade entre os resultados da simulação *Verilog-A* e a simulação com transistores no *HSPICE*, indicando uma modelagem apropriada. A comparação destes resultados com os de outros filtros encontrados na literatura é realizada na seção 5.7 após a apresentação do filtro sintetizado em tecnologia 180 nm.

Tabela 5.11: Descrição dos parâmetros em nível de sistema para síntese de caráter exploratório do filtro com frequência de corte em 20 MHz (CMOS 0,35 μm).

Nome	Tipo de Número	Tipo de Parâmetro	Variável a qual está conectada	Descrição
<i>gm</i>	indexado	variável	-	Transcondutância
<i>tnoise</i>	indexado	conectado	-	Ruído térmico
<i>fnoise</i>	indexado	conectado	<i>tnoise</i>	Ruído <i>flicker</i>
<i>deltav1</i>	indexado	conectado	-	Intervalo de linearidade 1
<i>deltav2</i>	indexado	conectado	<i>deltav1</i>	Intervalo de linearidade 2
<i>deltav3</i>	indexado	conectado	<i>deltav1</i>	Intervalo de linearidade 3
<i>deltav4</i>	indexado	conectado	<i>deltav1</i>	Intervalo de linearidade 4
<i>deltav5</i>	indexado	conectado	<i>deltav1</i>	Intervalo de linearidade 5
<i>maxc</i>	indexado	conectado	<i>deltav1</i>	Corrente máxima de saída
<i>midc</i>	indexado	conectado	<i>deltav1</i>	Corrente média de saída
<i>cout</i>	indexado	conectado	-	Capacitância de saída
<i>coutcmfb</i>	indexado	conectado	<i>cout</i>	Capacitância de saída do OTA+CMFB
<i>rout</i>	indexado	conectado	-	Resistência de saída
<i>routcmfb</i>	indexado	conectado	<i>rout</i>	Resistência de saída do OTA+CMFB
<i>isupply</i>	indexado	conectado	-	Corrente consumida
<i>isupplycmfb</i>	indexado	conectado	<i>isupply</i>	Corrente consumida do OTA+CMFB
<i>area</i>	indexado	conectado	-	Area
<i>areacmfb</i>	indexado	conectado	<i>area</i>	Area do OTA+CMFB
<i>cin</i>	indexado	conectado	-	Capacitância de entrada
<i>b1_n1</i>	inteiro	variável	-	Multiplicidade do OTA 1 e 2 da 1ª seção de 2ª ordem (intervalo entre 1 e 10)
<i>b1_n3</i>	inteiro	variável	-	Multiplicidade do OTA 3 da 1ª seção de 2ª ordem (intervalo entre 1 e 10)
<i>b1_n4</i>	inteiro	variável	-	Multiplicidade do OTA 4 da 1ª seção de 2ª ordem (intervalo entre 1 e 10)
<i>b2_n1</i>	inteiro	variável	-	Multiplicidade do OTA 1 e 2 da 2ª seção de 2ª ordem (intervalo entre 1 e 10)
<i>b2_n3</i>	inteiro	variável	-	Multiplicidade do OTA 3 da 2ª seção de 2ª ordem (intervalo entre 1 e 10)
<i>b2_n4</i>	inteiro	variável	-	Multiplicidade do OTA 4 da 2ª seção de 2ª ordem (intervalo entre 1 e 10)
<i>s1_n1</i>	inteiro	variável	-	Multiplicidade do OTA 1 e 2 da seção de 1ª ordem (intervalo entre 1 e 10)
<i>toposeq</i>	inteiro	variável	-	Sequência de posicionamento das seções do filtro (intervalo entre 0 e 5)

5.7 Síntese de um Filtro G_m -C Passa-Baixas com Frequência de Corte de 20 MHz em Tecnologia CMOS 180 nm

A troca de tecnologia normalmente significa um reprojeto completo dos blocos e sistema em um fluxo de projeto tradicional. Com o uso de ferramentas de automação, no entanto, a conversão das especificações de projeto em especificações de blocos e dimensionamento dos mesmos é facilitado, propiciando reaproveitamento do esforço do projetista e redução no *time-to-market* do projeto.

Para exemplificar este processo, nesta seção será sintetizado um filtro com as mesmas especificações do projeto da seção 5.6, porém em tecnologia TSMC 180 nm e tensão de alimentação

Tabela 5.12: Parâmetros dos OTAs resultantes da síntese do filtro G_m -C com frequência de corte de 20 MHz (CMOS 0,35 μm).

Parâmetro *	Medida
g_m [μS]	184
ruído térmico (OTA) ** [$n\text{V}$]	10,8
ruído <i>flicker</i> (OTA) ** [μV]	5,09
faixa de entrada *** [mV]	220
$C_{\text{Saída (OTA)}}$ [fF]	30,6
C_{Entrada} [fF]	39,8
$R_{\text{Saída (OTA)}}$ [k Ω]	478
$\text{Area}_{\text{(OTA)}}$ [μm] ²	531
$I_{\text{consumida(OTA)}}$ [μA]	114
ruído térmico (OTA+CMFB) ** [$n\text{V}$]	11,0
ruído <i>flicker</i> (OTA+CMFB) ** [μV]	5,09
$C_{\text{Saída (OTA+CMFB)}}$ [fF]	176
$\text{Area}_{\text{(OTA+CMFB)}}$ [μm] ²	2090
$I_{\text{consumida(OTA+CMFB)}}$ [μA]	345

* Dados sobre CMFB são utilizados apenas para OTA 1 e OTA 4 das seções de 2ª ordem e OTA 1 da seção de 1 polo

** referido à entrada

*** para variação de até 5% do valor de g_m máximo

Tabela 5.13: Multiplicidade e sequência dos OTAs resultantes da síntese do filtro G_m -C com frequência de corte de 20 MHz (CMOS 0,35 μm).

	1ª Seção de 2 polos			2ª Seção de 2 polos			Seção de 1 polo
	OTAs 1 e 2	OTA 3	OTA 4	OTAs 1 e 2	OTA 3	OTA 4	OTAs 1 e 2
Multiplicidade	1	1	1	1	2	1	1
Sequência	2ª Seção de 2 polos - Seção de 1 polo - 1ª Seção de 2 polos						

+0,9 V e -0,9 V.

As soluções obtidas no espaço de critério para representar a fronteira de Pareto do OTA podem ser vistas na Figura 5.15. A aproximação da fronteira de Pareto é composta por 455 soluções.

Os parâmetros finais encontrados para os OTAs do filtro em tecnologia 180 nm podem ser visto na Tabela 5.15. A multiplicidade dos OTAs pode ser vista na tabela 5.16. As medições resultantes do filtro simulado com modelos em *Verilog-A* e eletricamente no *HSPICE* podem ser vistos na Tabela 5.17.

Os resultados apresentados nesta seção, em tecnologia 180 nm, e os da seção 5.6, em 0,35 μm , são comparados na Tabela 5.18 com outros trabalhos encontrados na literatura. Na Tabela

Tabela 5.14: Especificações e resultados da síntese do filtro G_m -C com frequência de corte de 20 MHz (CMOS 0,35 μm).

Nome da Especificação	Objetivo	Simulação Verilog-A	Simulação HSPICE
Ganho DC [V]	$1 \pm 5\%$	0,95	0,95
Frequência de -3 dB [MHz]	$20 \pm 5\%$	19,6	19,6
Rejeição em 90 MHz [dB]	> 50	57	57
Potência [mW]	minimizar	8,0	8,3
Area [mm^2]	minimizar	0,016	0,016
SFDR [dB]	≥ 40	45,5	46,2
IIP3 [dBm]	-	9,16	10,21
Densidade de ruído referido à entrada [$n\text{V}/\sqrt{\text{Hz}}$]	-	54,9	55,4
χ^* do capacitor 1 da 1ª seção de 2ª ordem	$\leq 0,3$	0,13	0,13
χ^* do capacitor 2 da 1ª seção de 2ª ordem	$\leq 0,3$	0,24	0,24
χ^* do capacitor 1 da 2ª seção de 2ª ordem	$\leq 0,3$	0,21	0,21
χ^* do capacitor 2 da 2ª seção de 2ª ordem	$\leq 0,3$	0,22	0,22
χ^* do capacitor da seção de 1 polo	$\leq 0,3$	0,20	0,20
Atraso de grupo médio [n s]	-	31	31
Variação de atraso de grupo [n s]	-	13	13

* χ definido como a relação entre capacitância parasita e a capacitância total

Figura 5.15: Fronteira de Pareto do OTA para o filtro com frequência de corte em 20 MHz (CMOS 180 nm).

Tabela 5.15: Parâmetros dos OTAs resultante da síntese do filtro G_m -C com frequência de corte de 20 MHz (CMOS 180 nm).

Parâmetro *	Medida
g_m [μS]	69,5
ruído térmico (OTA) ** [nV]	17,3
ruído <i>flicker</i> (OTA) ** [μV]	5,24
faixa de entrada *** [mV]	10
$C_{Saída}$ (OTA) [fF]	8,4
$C_{Entrada}$ [fF]	18,4
$R_{Saída}$ (OTA) [k Ω]	1460
$Area_{(OTA)}$ [μm] ²	412
I consumida(OTA) [μA]	22,9
ruído térmico (OTA+CMFB) ** [nV]	18,1
ruído <i>flicker</i> (OTA+CMFB) ** [μV]	5,24
$C_{Saída}$ (OTA+CMFB) [fF]	96,8
$Area_{(OTA+CMFB)}$ [μm] ²	1420
I consumida(OTA+CMFB) [μA]	69,3

* Dados sobre CMFB são utilizados apenas para OTA 1 e OTA 4 das seções de 2 polos e OTA 1 da seção de 1 polo

** referido à entrada

*** medição de ΔV_1

Tabela 5.16: Multiplicidade dos OTAs resultantes das sínteses do filtro G_m -C com frequência de corte de 20 MHz (CMOS 180 nm).

	1ª Seção de 2 polos			2ª Seção de 2 polos			Seção de 1 polo
	OTAs 1 e 2	OTA 3	OTA 4	OTAs 1 e 2	OTA 3	OTA 4	OTAs 1 e 2
Multiplicidade	1	1	4	2	4	2	1
Sequência	Seção de 1 polo - 2ª			Seção de 2 polos - 1ª			Seção de 2 polos

5.19 estão os resultados para as figuras de mérito apresentadas nas equações 5.26 e 5.27.

$$FoM\ 1 = \frac{\text{Potência}}{\text{Ordem} \cdot \text{Largura de Banda}} \quad (5.26)$$

$$FoM\ 2 = \frac{\text{Potência}}{\text{Ordem} \cdot \text{Largura de Banda} \cdot \text{SFDR}} \quad (5.27)$$

Dado que no presente trabalho foi utilizada uma topologia simples de OTA e sem utilizar circuito de sintonização, esta comparação tem intenção apenas de demonstrar que a solução obtida pelo otimizador tem especificações compatíveis com a literatura.

Tabela 5.17: Especificações e resultados da síntese do filtro G_m -C com frequência de corte de 20 MHz em tecnologia 180 nm.

Nome da Especificação	Objetivo	Simulação Verilog-A	Simulação HSPICE
Ganho DC [V]	$1 \pm 5\%$	0,95	0,95
Frequência de -3 dB [MHz]	$20 \pm 5\%$	19,7	20,8
Potência [mW]	minimizar	1,62	1,68
Area [mm] ²	minimizar	0,008	0,008
SFDR [dB]	≥ 40	41,0	41,3
IIP3 [dBm]	-	3,4	3,8
Densidade de ruído referido à entrada [$nV/\sqrt{Hz}$]	-	61,6	70,0
χ^* do capacitor 1 da 1ª seção de 2ª ordem	$\leq 0,3$	0,23	0,23
χ^* do capacitor 2 da 1ª seção de 2ª ordem	$\leq 0,3$	0,30	0,30
χ^* do capacitor 1 da 2ª seção de 2ª ordem	$\leq 0,3$	0,26	0,26
χ^* do capacitor 2 da 2ª seção de 2ª ordem	$\leq 0,3$	0,28	0,28
χ^* do capacitor da seção de 1ª ordem	$\leq 0,3$	0,29	0,29
Atraso de grupo médio [ns]	-	31	30
Variação de atraso de grupo [ns]	-	14	14

* χ definido como a relação entre capacitância parasita e a capacitância total

Tabela 5.18: Comparação entre resultados da síntese do filtro G_m -C passa-baixas com frequência de corte de 20 MHz e outros trabalhos (baseado na tabela 5.6 de (SAARI, 2012)).

Trabalho	Ganho DC [dB]	Freq. [MHz]	Topo.*/Ordem	Ruído [$nV/\sqrt{Hz}$]	IIP3	VDD [V] / Pot. [mW]	Area [mm^2]	SFDR [dB]
Este trabalho (0,35 μm) **	≈ 0	20	Bu./5	55,4	10,2	3,3/8,3	0,016***	46
Este trabalho (0,18 μm) **	≈ 0	20	Bu./5	70,0	3,8	1,8/1,7	0,008***	41
(HORI et al., 2003)	≈ 0	1,5-12	El./6	174	9,3	1,8/10	0,83	41
(HORI et al., 2004)	≈ 0	0,5-12	Bu./4	254	9,4	1,8/4,5	0,125	39
(LEWINSKI; SILVA-MARTINEZ, 2007)	≈ 0	30	El./5	21	nd	3,3/85	1,4	65
(CROMBEZ et al., 2008)	≈ 0	0,1-20	Bu./2	5,6	20	1,2/14,2	0,5	66
(LO et al., 2009)	≈ 0	0,5-20	Bu./3	12	19	1,2/11,1	0,23	61
(ACOSTA et al., 2009)	≈ 0	6-12	Ch./3	64	33	5/180	1,5	62

Bu. = Butterworth; El. = Elíptico; Ch. = Chebychev;

** resultados de simulação e sem circuito de sintonização;

*** área de gates de transistores e capacitores.

Tabela 5.19: Figuras de Mérito para o resultado da síntese do filtro G_m -C e outros trabalhos.

Trabalho	FoM 1* [nJ]	FoM 2** [fJ]
Este trabalho (0,35 μm) ***	0,08	1,9
Este trabalho (0,18 μm) ***	0,02	1,2
(HORI et al., 2003)	0,17	13,2
(HORI et al., 2004)	0,11	14,2
(LEWINSKI; SILVA-MARTINEZ, 2007)	0,57	0,18
(CROMBEZ et al., 2008)	0,36	0,09
(LO et al., 2009)	0,19	0,15
(ACOSTA et al., 2009)	5	3,15

* $FoM 1 = \text{Potência}/(\text{Ordem} \cdot \text{Largura de Banda})$;

** $FoM 2 = \text{Potência}/(\text{Ordem} \cdot \text{Largura de Banda} \cdot \text{SFDR})$;

*** resultados de simulação e sem circuito de sintonização.

5.8 Conversão de Especificações de Sistema em Especificação de Bloco utilizando a Fronteira de Pareto apenas como Referência

A fim de demonstrar o uso da ferramenta e da aproximação da fronteira de Pareto em situações de caráter exploratório, são realizadas sínteses usando variáveis não conectadas nesta seção. Enquanto o otimizador busca soluções para satisfazer as especificações de sistema, ele não limita as variáveis (especificações dos blocos) apenas às já encontradas na aproximação da fronteira de Pareto. Isso permite ao otimizador alcançar especificações de sistema que seriam inalcançáveis com a fronteira de Pareto obtida, permitindo ao projetista observar quais especificações de blocos precisa garantir para obtê-las. Nesse tipo de síntese, a função custo descrita na seção 5.2.3 ganha maior relevância do que nos casos anteriores. A solução final será o conjunto de especificações do OTA que atendem as especificações de sistema enquanto são o menos restritivas possíveis em nível de bloco.

Para estas sínteses foi definido que todos os OTAs do filtro são iguais. A fronteira de Pareto utilizada como referência para as sínteses desta seção é a mesma utilizada para a seção 5.6.

Na definição dos parâmetros de alto nível, enquanto na seção anterior os parâmetros de um mesmo bloco do sistema eram majoritariamente do tipo *conectados* e apenas um parâmetro por bloco era *variável*, nesta síntese os parâmetros são em grande parte *variáveis*. Os parâmetros que permanecem *conectados*, são dessa forma para facilitar a modelagem do OTA, como no caso dos parâmetros ΔV_2 , ΔV_3 , ΔV_4 e ΔV_5 em relação ao ΔV_1 , ou para reduzir o número de variáveis da síntese. A Tabela 5.20 mostra a descrição das variáveis de alto nível para este caso.

A função de transferência do filtro é a mesma descrita para os filtros em 20 MHz das seções anteriores. No entanto, enquanto os objetivos dos casos anteriores eram minimizar area, consumo e maximizar SFDR, no presente caso, esses passam a ser apenas restrições e o objetivo se torna minimizar a função custo de viabilidade. Com a finalidade de avaliar a sintetizabilidade dos circuitos para diferentes especificações foram realizadas três sínteses: com restrição de SFDR mínimo igual a 40 dB, 50 dB e 60 dB. As demais especificações permaneceram as mesmas do caso anterior (Tabela 5.10). Os três casos foram sintetizados compartilhando a mesma fronteira de Pareto descrita anteriormente (Figura 5.15).

A Tabela 5.21 mostra os parâmetros do OTA da solução final encontrada para as sínteses. A Tabela 5.22 mostra a multiplicidade e sequência final dos OTAs dentro dos filtros para cada caso. Conforme aumenta a restrição de SFDR mínimo, observa-se a redução do ruído *flicker* e térmico do OTA e aumento da faixa de entrada linear. Esse comportamento é esperado visto a dependência do SFDR no ruído e linearidade do circuito.

Tabela 5.20: Descrição dos parâmetros em nível de sistema para síntese de caráter exploratório do filtro com frequência de corte de 20 MHz (CMOS 180 nm).

Nome	Tipo de Número	Tipo de Parâmetro	Variável a qual está conectada	Descrição
<i>gm</i>	indexado	variável	-	Transcondutância
<i>tnoise</i>	indexado	variável	-	Ruído térmico
<i>fnoise</i>	indexado	conectado	<i>tnoise</i>	Ruído <i>flicker</i>
<i>deltav1</i>	indexado	variável	-	Intervalo de linearidade 1
<i>deltav2</i>	indexado	conectado	<i>deltav1</i>	Intervalo de linearidade 2
<i>deltav3</i>	indexado	conectado	<i>deltav1</i>	Intervalo de linearidade 3
<i>deltav4</i>	indexado	conectado	<i>deltav1</i>	Intervalo de linearidade 4
<i>deltav5</i>	indexado	conectado	<i>deltav1</i>	Intervalo de linearidade 5
<i>maxc</i>	indexado	conectado	<i>deltav1</i>	Corrente máxima de saída
<i>midc</i>	indexado	conectado	<i>deltav1</i>	Corrente média de saída
<i>cout</i>	indexado	variável	-	Capacitância de saída
<i>coutcmfb</i>	indexado	conectado	<i>cout</i>	Capacitância de saída do OTA+CMFB
<i>rout</i>	indexado	variável	-	Resistência de saída
<i>routcmfb</i>	indexado	conectado	<i>rout</i>	Resistência de saída do OTA+CMFB
<i>isupply</i>	indexado	variável	-	Corrente consumida
<i>isupplycmfb</i>	indexado	conectado	<i>isupply</i>	Corrente consumida do OTA+CMFB
<i>area</i>	indexado	variável	-	Area
<i>areacmfb</i>	indexado	conectado	<i>area</i>	Area do OTA+CMFB
<i>cin</i>	indexado	variável	-	Capacitância de entrada
<i>b1_n1</i>	inteiro	variável	-	Multiplicidade do OTA 1 e 2 da primeira seção de 2ª ordem (intervalo entre 1 e 10)
<i>b1_n3</i>	inteiro	variável	-	Multiplicidade do OTA 3 da primeira seção de 2ª ordem (intervalo entre 1 e 10)
<i>b1_n4</i>	inteiro	variável	-	Multiplicidade do OTA 4 da primeira seção de 2ª ordem (intervalo entre 1 e 10)
<i>b2_n1</i>	inteiro	variável	-	Multiplicidade do OTA 1 e 2 da segunda seção de 2ª ordem (intervalo entre 1 e 10)
<i>b2_n3</i>	inteiro	variável	-	Multiplicidade do OTA 3 da segunda seção de 2ª ordem (intervalo entre 1 e 10)
<i>b2_n4</i>	inteiro	variável	-	Multiplicidade do OTA 4 da segunda seção de 2ª ordem (intervalo entre 1 e 10)
<i>s1_n1</i>	inteiro	variável	-	Multiplicidade do OTA 1 e 2 da seção de 1ª ordem (intervalo entre 1 e 10)
<i>toposeq</i>	inteiro	variável	-	Sequência de posicionamento das seções do filtro (intervalo entre 0 e 5)

Para os casos de SFDR mínimo igual a 40 dB e 50 dB, a função custo de viabilidade resultou em um valor inferior a zero. Dessa forma, estas soluções são consideradas viáveis (existe solução dentro da fronteira de Pareto que domina a solução necessária para atingir as especificações de sistema). Para o caso de SFDR mínimo igual a 60 dB, nenhum OTA da fronteira de Pareto foi capaz de atingir as especificações descritas na tabela. Estas especificações servem, dessa forma, como referência para que o projetista busque alternativas de implementação. Especificamente, a solução da fronteira de Pareto mais próxima da solução final não alcançou as medidas de faixa de linearidade, corrente consumida, area, ruído flicker, térmico e resistência de saída necessárias.

As medidas de sistema resultantes das soluções apresentadas constam na Tabela 5.23. Uma

Tabela 5.21: Parâmetros dos OTAs resultantes da conversão de especificações de sistema para circuito do filtro G_m -C com frequência de corte de 20 MHz (CMOS 180 nm).

Parâmetro *	SFDR ≥ 40 dB	SFDR ≥ 50 dB	SFDR ≥ 60 dB
g_m [μS]	972	1080	1080
ruído térmico (OTA) ** [nV]	25,7	5,9	3,8
ruído <i>flicker</i> (OTA) ** [μV]	16,0	2,0	0,7
faixa de entrada *** [mV]	100	140	260
$C_{Saída}$ (OTA) [fF]	337	283	10
$C_{Entrada}$ [fF]	418	536	972
$R_{Saída}$ (OTA) [$k\Omega$]	100	100	100
$Area_{(OTA)}$ [μm] ²	4350	4970	1400
I consumida(OTA) [μA]	561	561	409
ruído térmico (OTA+CMFB) ** [nV]	25,8	5,9	3,8
ruído <i>flicker</i> (OTA+CMFB) ** [μV]	16,0	2,0	0,7
$C_{Saída}$ (OTA+CMFB) [fF]	337	324	18
$Area_{(OTA+CMFB)}$ [μm] ²	13800	13900	4270
I consumida(OTA+CMFB) [μA]	1680	1680	1220
Solução viável****	Sim	Sim	Não

* dados sobre CMFB são utilizados apenas para OTA 1 e OTA 4 das seções de 2 polos e OTA 1 da seção de 1 polo

** referido à entrada

*** para variação até 5% do valor de g_m máximo

**** para topologia e tecnologia selecionadas

observação importante refere-se às medidas de SFDR terem resultado em valores muito próximos da especificação mínima para cada caso. Isso é esperado e consequência do objetivo da otimização ser minimizar a função custo viabilidade em detrimento de melhorias além das restrições mínimas. O valor da função custo de viabilidade é reduzido conforme as especificações dos blocos se tornam menos restritivas e mais fáceis de serem implementadas fisicamente. Dessa forma, uma solução a ser fabricada não usara as especificações desta tabela, mas as terá como referência do mínimo necessário dos blocos para atingir as especificações de sistema.

Comparando os resultados viáveis (SFDR ≥ 40 dB e ≥ 50 dB) com os resultados da síntese da seção anterior na Tabela 5.18 (variáveis compartilhadas) observa-se que os da presente seção tem desempenho inferior. A razão desta diferença deriva da observação do parágrafo anterior. Enquanto nos presentes casos é buscada a minimização da função custo viabilidade, no caso anterior os objetivos eram a minimização da area, potência e maximização do SFDR utilizando soluções da fronteira de Pareto em si.

5.9 Resumo do Capítulo

Neste capítulo foi apresentado o procedimento de síntese em nível de sistema. Para tal, foi utilizada a avaliação da viabilidade dos circuitos com base na fronteira de Pareto e introduzi-

Tabela 5.22: Resultado da multiplicidade dos OTAs resultantes da conversão de especificações de sistema para circuito do filtro G_m -C com frequência de corte de 20 MHz (CMOS 0,35 μm).

Caso	1ª Seção de 2 polos			2ª Seção de 2 polos			Seção de 1 polo	Sequência
	OTAs	OTA	OTA	OTAs	OTA	OTA	OTAs	
	1 e 2	3	4	1 e 2	3	4	1 e 2	
SFDR ≥ 40 dB	2	1	2	2	5	5	1	b2,s1,b1
SFDR ≥ 50 dB	2	2	3	2	3	2	1	b2,b1,s1
SFDR ≥ 60 dB	10	4	4	9	10	6	4	b2,b1,s1

Tabela 5.23: Especificações e resultados da síntese do filtro G_m -C com frequência de corte de 20 MHz com variáveis não compartilhadas (CMOS 0,35 μm).

Nome da Especificação	SFDR ≥ 40 dB	SFDR ≥ 50 dB	SFDR ≥ 60 dB
Ganho DC [V]	0,95	0,96	0,97
Frequência de -3 dB [MHz]	19,6	19,6	19,5
Ruído referido à entrada [$nV/\sqrt{Hz}$]	69,9	13,79	3,94
Potência [mW]	47,4	40,3	99,7
Area [mm^2]	0,10	0,99	0,98
IIP3 [dBm]	3,03	4,19	8,70
SFDR [dB]	40,0	50,1	60,5
FoM 1* [nJ]	0,47	0,40	1,00
FoM 2** [fJ]	47,2	3,9	0,9

* $FoM 1 = \text{Potência}/(\text{Ordem} \cdot \text{Largura de Banda})$;

** $FoM 2 = \text{Potência}/(\text{Ordem} \cdot \text{Largura de Banda} \cdot \text{SFDR})$;

dos novos tipos de variáveis em relação às utilizadas em nível de transistor. Para exemplificar os conceitos descritos no capítulo, foram realizadas a modelagem do bloco OTA, geração da fronteira de Pareto e diversas sínteses de filtros G_m -C. Os resultados foram comparados com trabalhos apresentados na literatura realizados manualmente e com programação geométrica e indicam que a técnica proposta permite alcançar soluções otimizadas em nível de sistema. Ainda, uma nova técnica para conversão de especificações de sistema para especificações de blocos utilizando as fronteiras de Pareto apenas como referência foi proposta e sínteses exploratórias foram realizadas.

6 CONCLUSÕES

Houve um grande avanço na indústria de semicondutores nas últimas décadas, propiciando elaboração de projetos de alta complexidade e reduzido *time-to-market*. No entanto, houve maior incremento na produtividade de projetos digitais, enquanto os projetos analógicos permaneceram com seu fluxo de projeto praticamente inalterado, fazendo deles o gargalo de tempo em projetos de sinal misto. Ainda, um projeto analógico realizado de forma tradicional é fortemente dependente da tecnologia e especificações, necessitando um projetista ou uma equipe de projetistas para realizar a reelaboração do bloco caso seja necessária qualquer modificação mesmo em projetos considerados não desafiadores. Este contexto favorece à busca de alternativas que aumentem a produtividade dos projetistas analógicos bem como os auxiliem a buscar soluções de alta qualidade diante de projetos complexos.

As ferramentas mais usualmente presentes no fluxo de projeto analógico são os capturadores de esquemático, simuladores elétricos e as ferramentas relacionadas a *layout* (para desenho, DRC, LVS, extração, entre outras). A partir dos primeiros trabalhos relacionados à síntese de circuitos integrados analógicos, foram propostas abordagens que utilizavam um novo tipo de ferramenta para realizar o dimensionamento dos transistores automaticamente. Este tipo de ferramenta seria responsável pela síntese em nível de circuito e poderia tanto seguir um plano de projeto previamente estipulado, quanto converter o problema de projeto em um problema de otimização. Ferramentas em nível de sistema também foram propostas ao longo da história do estudo do campo de síntese analógica, normalmente com foco em sistemas específicos mas também existindo abordagens genéricas.

Neste trabalho foi realizado o estudo de ferramentas de **síntese em nível de circuito e sistema**, bem como de algoritmos de otimização do tipo livre de derivada para aplicação durante a síntese. Tendo em vista o papel fundamental do algoritmo de otimização neste processo, foi proposto um novo algoritmo que envolve *Simulated Annealing/Simulated Quenching* (SA/SQ) juntamente com características advindas de algoritmos genéticos utilizando informações de múltiplos objetivos. Foi também desenvolvido um algoritmo híbrido entre o primeiro e o *Particle Swarm Optimization* para a obtenção de aproximações das fronteiras de Pareto dos circuitos. Foi desenvolvida uma ferramenta de síntese capaz de sintetizar circuitos analógicos

em nível de circuito a partir de descrições fornecidas pelo usuário e fornecer circuitos dimensionados. A ferramenta também foi aperfeiçoada para permitir a síntese em nível de sistema utilizando fronteiras de Pareto obtidas para cada bloco.

As ferramentas para síntese analógica existem há mais de 30 anos, dentro dos quais estas progressivamente migraram de uma abordagem baseada em conhecimento para uma abordagem baseada em otimização. Os trabalhos encontrados na literatura visam atingir algumas ou todas as características desejadas a uma ferramenta de síntese de circuitos integrados analógicos, que incluem independência do processo de fabricação, capacidade de sintetizar diversos tipos de circuito, precisão e tempo baixo de preparação antes da síntese. A revisão bibliográfica, estudo do estado da arte e a formulação do problema de síntese de circuitos integrados analógicos pela perspectiva de otimização são apresentados no capítulo 2 desta tese.

O algoritmo proposto de SA/SQ com *crossovers* utiliza informações advindas da característica multiobjetivo do problema para selecionar algumas soluções que têm desempenho melhor que todas as demais em alguma especificação (chamadas soluções âncora) e guardá-las na memória. Uma vez que o algoritmo tenha encontrado um mínimo local, realiza-se uma operação de *crossover* entre a solução atual e uma selecionada das memorizadas para auxiliar o algoritmo a continuar a exploração do espaço de projeto. Ainda, nos momentos de *crossovers* são efetuadas alterações de peso das especificações na função custo a fim de aumentar o impacto das especificações que foram detectadas como mais distantes de serem atingidas. Esta abordagem permite que a fuga dos mínimos locais ocorra de forma direcionada aos objetivos mais críticos e possibilita que o sintetizador direcione as soluções para áreas promissoras do espaço de projeto. No capítulo 3, o algoritmo é apresentado e comparado com diversos algoritmos do tipo livre de derivada na busca pelo mínimo global de problemas matemáticos mono-objetivo e multiobjetivo. Para os teste mono-objetivo, a função custo retornava ao algoritmo, além da função custo agregativa (composição de todos resultados), as partes da função que compunham o resultado final (considerados os múltiplos objetivos). Para o caso mono-objetivo, o algoritmo proposto obteve resultados superiores aos demais para um número de variáveis/objetivos até 30 nos testes até 2.000 iterações, enquanto que para testes até 10.000 iterações ele obteve resultados melhores que os demais para a maior parte dos números de variáveis/objetivos testados. Para os problemas multiobjetivo e para um número de objetivos superior a 3, o algoritmo proposto obteve desempenho superior aos demais com os quais foi comparado, tanto nos testes até 5.000 quanto nos até 10.000 iterações. O algoritmo e os testes realizados podem ser vistos no capítulo 3.

Foi desenvolvida uma ferramenta para **síntese em nível de circuitos** que utiliza interface através de arquivos texto de entrada. Através desta interface é possível definir topologias, espe-

cificações, *testbenches* e configurações de síntese. Assim, a ferramenta permite grande flexibilidade ao projetista e pode ser utilizada para uma grande variedade de circuitos eletrônicos. A fim de validar a ferramenta e comparar os algoritmos, foi realizado uma série de testes de síntese aplicados a circuitos integrados. Os testes comparativos são realizados entre o algoritmo proposto e outros algoritmos, resultados manuais encontrados na literatura, resultados de projetos realizados com outras ferramentas. Os testes comparativos entre algoritmos são realizados tanto para o caso mono-objetivo quando para a busca da fronteira de Pareto no caso multiojetivo. Os resultados mostraram que o algoritmo teve desempenho superior aos demais comparados, sendo esta diferença comprovada estatisticamente com nível de significância (α) de 0,05 no teste não paramétrico de Wilcoxon-Mann-Whitney. Os blocos sintetizados nos testes comparativos são o amplificador do tipo Miller, um amplificador *Folded Cascode* e um *Folded Cascode* complementar, um amplificador de baixo ruído (LNA) e uma fonte de referência. Além da síntese nos testes comparativos, projetos de fontes de referência foram realizados utilizando diferentes tecnologias e topologias, sendo que para duas versões (em 130 nm e 180 nm), o fluxo de projeto foi realizado até a elaboração de *layout* e extração do circuito com parasitas. Os últimos testes realizados em nível de circuito foram referentes a busca da fronteira de Pareto, no qual foram sintetizados o amplificador *Miller* e o amplificador *Folded Cascode* complementar. Para este teste foram utilizados o algoritmo *SA/SQ* com *crossovers* integrado com o *Particle Swarm Optimization* em comparação com outros algoritmos multiobjetivo. Foi possível alcançar as especificações mínimas dos projetos e traçar gráficos em três dimensões mostrando as possíveis escolhas de projeto. Como resultado dos testes, mostrou-se que o algoritmo proposto foi capaz de encontrar uma melhor aproximação da fronteira de Pareto que os demais algoritmos para os casos estudados. A ferramenta desenvolvida para uso em nível de circuito e os testes foram apresentados no capítulo 4.

Para **síntese em nível de sistema**, a ferramenta foi modificada para utilizar fronteiras de Pareto dos blocos como referência para exploração do espaço de projeto e como guia para avaliar a sintetizabilidade de um determinado conjunto de especificações de blocos. A ferramenta seleciona dentro das fronteiras de Pareto as soluções que permitem que as especificações de sistema sejam atingidas. Nos casos em que a síntese está sendo realizada de forma exploratória, a ferramenta utiliza as fronteiras de Pareto apenas como referência, permitindo que especificações de blocos diferentes das disponíveis diretamente na fronteira sejam utilizadas. Ainda assim, a fronteira de Pareto continua sendo utilizada neste caso como guia de sintetizabilidade mesmo no caso exploratório. Testes foram realizados para a síntese de filtros G_m -C e utilizando fronteiras de Pareto de blocos OTA. Os resultados indicaram que a ferramenta foi capaz de realizar a síntese obtendo figuras de mérito interessantes quando comparados a projetos encontrados na literatura. A conversão de especificações em nível de sistema para nível de circuito foi reali-

zada com sucesso. A ferramenta desenvolvida para uso em nível de sistema foi apresentada no capítulo 5.

Com os algoritmos propostos na tese e a ferramenta desenvolvida, buscou-se melhorar o desempenho em relação ao estado da arte em síntese de circuitos integrados analógicos. A utilização de ferramentas de síntese é vista pelo autor como uma forma de aumentar a produtividade dos projetistas, de forma que estes possam projetar circuitos e sistemas com melhor desempenho e de forma ágil. Mesmo com a adoção de ferramentas de síntese, o projetista continua sendo o principal responsável pela elaboração de um circuito ou sistema confiável. Ele é quem definirá os testes a serem realizados e os critérios que a ferramenta utilizará para dimensionar os circuitos. No entanto, ao utilizar a ferramenta para realizar o projeto, o projetista será beneficiado em tempo e qualidade quando precisar fazer uma modificação das especificações ou tecnologia do projeto.

O trabalho desenvolvido pode ser utilizado em diversas áreas. Na pesquisa, pode ser utilizado como plataforma de teste e comparação entre algoritmos de otimização para o problema de circuitos analógicos. O pesquisador pode interfacear seu algoritmo com a ferramenta e testar o desempenho do mesmo com os algoritmos já implementados. Na indústria, pode ser utilizado como forma de aumentar a produtividade das equipes de projeto analógico, uma vez que pode automatizar projetos considerados mais simples e fornecer soluções iniciais para projetos complexos, ilustrando graficamente as escolhas de projeto. E, finalmente, na esfera educacional, pode-se utilizar a ferramenta de síntese como forma de ensinar técnicas de projeto automatizadas em disciplinas que contenham ementa relacionada a projeto em alto nível.

Como **considerações finais** desta tese, ressalta-se que o uso de algoritmos do tipo livre de derivada em abordagens baseadas em otimização e avaliação baseada em simulação tem se delineado como o caminho mais promissor para o futuro da síntese de circuitos integrados analógicos. Este tipo de abordagem atende de forma adequada grande parte das demandas do problema de síntese, permitindo flexibilidade e precisão. Em especial, os resultados obtidos ao longo desta tese indicam que o desenvolvimento de algoritmos que consideram múltiplos objetivos mesmo em sínteses que têm por objetivo obter apenas uma solução final é uma abordagem promissora. Quanto ao projeto hierárquico, ao utilizar a fronteira de Pareto das soluções como fonte de informações para a síntese em nível de sistema, é possível realizar explorações de projeto e desenvolver sistemas de forma mais integrada e favorecer o reuso do esforço do projetista. Este é um campo de pesquisa ativo e com muitas possibilidades a serem exploradas.

Por fim, propõem-se alguns tópicos como **recomendações para futuros trabalhos** e aperfeiçoamentos desta pesquisa. Um dos problemas no desenvolvimento de algoritmos para síntese analógica é a ausência de uma métrica comum na literatura para avaliação dos resultados. Dessa

forma, seria interessante a criação de um conjunto de testes para padronização da avaliação de algoritmos de síntese analógica utilizando tecnologias modernas. Esta padronização possivelmente proporcionaria o crescimento da área tanto por servir como desafio para os pesquisadores já envolvidos em problemas de síntese quanto por permitir que pesquisadores da área de otimização possam testar seus algoritmos. Quanto ao desenvolvimento de novos algoritmos, recomenda-se explorar o uso das características dos algoritmos propostos nesta tese em outros tipos de algoritmos além do *Simulated Annealing* e o *Particle Swarm Optimization*. Quanto ao projeto hierárquico, recomenda-se a criação de uma metodologia de exploração progressiva da fronteira de Pareto em nível de circuito enquanto a síntese ocorre em nível de sistema. O uso da fronteira de Pareto apenas como referência apresentada nesta tese serve como precedente para a realização desta exploração progressiva. Nesta nova abordagem, determinadas regiões da fronteira de Pareto seriam avaliadas apenas quando a síntese do sistema demande maior resolução nestas áreas, reduzindo a demanda sobre a criação de fronteiras de Pareto completas antes do começo da síntese em nível de sistema. Ainda, recomenda-se a criação de formas simplificadas de avaliação das respostas quanto a *yield* e efeitos parasitários de *layout* para que estas características possam ser facilmente implementadas dentro do ciclo de otimização.

REFERÊNCIAS

ACOSTA, L.; JIMENEZ, M.; CARVAJAL, R.; LOPEZ-MARTIN, A.; RAMIREZ-ANGULO, J. Highly linear tunable CMOS Gm-C low-pass filter. **Circuits and Systems I: Regular Papers, IEEE Transactions on**, v. 56, n. 10, p. 2145–2158, Oct. 2009. ISSN 1549-8328.

ALI, S. H. M. **System Level Performance and Yield Optimisation for Analogue Integrated Circuits**. Tese (Doutorado) — University of Southampton, 2009.

ALLEN, P. E.; HOLBERG, D. R. **CMOS Analog Circuit Design**. 2nd ed. Oxford: Oxford University Press, 2002. ISBN 9780195116441.

AZEVEDO, C.; ARAUJO, A. F. R. Correlation between diversity and hypervolume in evolutionary multiobjective optimization. In: **Evolutionary Computation (CEC), 2011 IEEE Congress on**. New Orleans: IEEE, 2011. p. 2743–2750.

BALKIR, S.; DUNDAR, G.; OGRENCI, S. **Analog VLSI Design Automation**. Boca Raton: CRC Press, 2003. ISBN 0-8493-1090-3.

BANBA, H.; SHIGA, H.; UMEZAWA, A.; MIYABA, T.; TANZAWA, T.; ATSUMI, S.; SAKUI, K. A CMOS bandgap reference circuit with sub-1-V operation. **Solid-State Circuits, IEEE Journal of**, v. 34, n. 5, p. 670–674, May 1999. ISSN 0018-9200.

BARROS, M.; GUILHERME, J.; HORTA, N. Analog circuits optimization based on evolutionary computation techniques. **Integration, the VLSI Journal**, v. 43, n. 1, p. 136 – 155, 2010. ISSN 0167-9260.

BERNARDINIS, F. D.; JORDAN, M.; SANGIOVANNIVINCENNELLI, A. Support vector machines for analog circuit performance representation. In: **Design Automation Conference, 2003. Proceedings**. Anaheim: IEEE, 2003. p. 964 – 969.

BERNARDINIS, F. D.; NUZZO, P.; VINCENNELLI, A. Robust system-level design with analog platforms. In: **Computer-Aided Design, 2006. ICCAD '06. IEEE/ACM International Conference on**. San Jose, CA: IEEE, 2006. p. 334–341. ISSN 1092-3152.

BERNARDINIS, F. D.; VINCENNELLI, A. A methodology for system-level analog design space exploration. In: **Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings**. Paris: IEEE, 2004. v. 1, p. 676–677 Vol.1. ISSN 1530-1591.

BEYER, H.-G.; SCHWEFEL, H.-P. Evolution strategies - a comprehensive introduction. **Natural Computing**, Kluwer Academic Publishers, v. 1, n. 1, p. 3–52, 2002. ISSN 1567-7818.

BLEULER, S.; LAUMANN, M.; THIELE, L.; ZITZLER, E. PISA — a platform and programming language independent interface for search algorithms. In: **Evolutionary Multi-Criterion Optimization (EMO 2003)**. Berlin: Springer, 2003. p. 494 – 508.

BOHACHEVSKY, I. O.; JOHNSON, M. E.; STEIN, M. L. Generalized simulated annealing for function optimization. **Technometrics**, Taylor & Francis, v. 28, n. 3, p. 209–217, Aug. 1986.

BORCHERS, C. Symbolic behavioral model generation of nonlinear analog circuits. **IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing**, v. 45, p. 1362–1371, 1998.

BREDERLOW, R.; WEBER, W.; SAUERER, J.; DONNAY, S.; WAMBACQ, P.; VERTREGT, M. A mixed-signal design roadmap. **Design Test of Computers, IEEE**, v. 18, n. 6, p. 34–46, 2001. ISSN 0740-7475.

BROCKHOFF, D.; FRIEDRICH, T.; HEBBINGHAUS, N.; KLEIN, C.; NEUMANN, F.; ZITZLER, E. On the effects of adding objectives to plateau functions. **Evolutionary Computation, IEEE Transactions on**, v. 13, n. 3, p. 591–603, June 2009. ISSN 1089-778X.

BROOKS, D. G.; VERDINI, W. A. Computational experience with generalized simulated annealing over continuous variables. **American Journal of Mathematical and Management Sciences**, American Sciences Press, Syracuse, NY, USA, v. 8, n. 3-4, p. 425–449, Jan. 1988. ISSN 0196-6324.

CHANG, H.; SANGIOVANLLI-VINCENTELLI, A.; BALARIN, F.; CHARBON, E.; CHOUDHURY, U.; JUSUF, G.; LIU, E.; MALAVASI, E.; NEFF, R.; GRAY, P. A top-down, constraint-driven design methodology for analog integrated circuits. In: **Custom Integrated Circuits Conference, 1992., Proceedings of the IEEE 1992**. Boston: IEEE, 1992. p. 8.4.1–8.4.6.

CHAPARRO, S.; PABÓN, A. A.; ROA, E.; NOIJE, W. V. A low power CMOS LNA design using geometric programming. In: **XV IBERCHIP Workshop**. Buenos Aires: [s.n.], 2009. p. 741–745.

COELLO, C.; PULIDO, G.; LECHUGA, M. Handling multiple objectives with particle swarm optimization. **Evolutionary Computation, IEEE Transactions on**, v. 8, n. 3, p. 256–279, June 2004. ISSN 1089-778X.

COLORNI, A.; DORIGO, M.; MANIEZZO, V. Distributed optimization by ant colonies. In: **Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life**. Cambridge: MIT Press, 1991. p. 134–142.

COMER, D.; COMER, D. Operation of analog MOS circuits in the weak or moderate inversion region. **Education, IEEE Transactions on**, v. 47, n. 4, p. 430–435, 2004. ISSN 0018-9359.

CORANA, A.; MARCHESI, M.; MARTINI, C.; RIDELLA, S. Minimizing multimodal functions of continuous variables with the simulated annealing algorithm. **ACM Transactions on Mathematical Software**, ACM, New York, v. 13, n. 3, p. 262–280, set. 1987. ISSN 0098-3500.

CORTES, F.; BAMPI, S. Miller OTA design using a design methodology based on the g_m/I_d and Early-Voltage characteristics: Design considerations and experimental results. In: **XII Workshop Iberchip**. San José, Costa Rica: [s.n.], 2006. p. 10–13.

CROMBEZ, P.; CRANINCKX, J.; WAMBACQ, P.; STEYAERT, M. A 100-kHz to 20-MHz reconfigurable power-linearity optimized Gm - C biquad in 0.13- μ m CMOS. **Circuits and Systems II: Express Briefs, IEEE Transactions on**, v. 55, n. 3, p. 224–228, mar. 2008. ISSN 1549-7747.

DAEMS, W.; GIELEN, G.; SANSEN, W. Circuit simplification for the symbolic analysis of analog integrated circuits. **Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on**, v. 21, n. 4, p. 395–407, apr 2002. ISSN 0278-0070.

DANTZIG, G. B. **Linear programming and extensions**. Princeton, NJ: Princeton University Press, 1963.

DAS, I.; DENNIS, J. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. **Structural optimization**, Springer-Verlag, v. 14, n. 1, p. 63–69, 1997. ISSN 0934-4373.

DAS, I.; DENNIS, J. E. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. **SIAM J. on Optimization**, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, v. 8, n. 3, p. 631–657, Mar. 1998. ISSN 1052-6234.

DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. **Evolutionary Computation, IEEE Transactions on**, IEEE Press, Piscataway, NJ, USA, v. 6, n. 2, p. 182–197, abr. 2002. ISSN 1089-778X.

DEGRAUWE, M. G. R.; NYS, O.; DIJKSTRA, E.; RIJMENANTS, J.; BITZ, S.; GOFFART, B. L. A. G.; VITTOZ, E. A.; CSERVENY, S.; MEIXENBERGER, C.; STAPPEN, G. van der; OGUEY, H. J. IDAC: An interactive design tool for analog CMOS circuits. **IEEE Journal of Solid-state Circuits**, v. 22, p. 1106–1116, 1987.

DOBOLI, A.; DHANWADA, N. R.; NUNEZ-ALDANA, A.; VEMURI, R. A two-layer library-based approach to synthesis of analog systems from VHDL-AMS specifications. **ACM Transactions on Design Automation of Electronic Systems**, v. 9, p. 238–271, 2004.

DOBOLI, A.; VEMURI, R. Behavioral modeling for high-level synthesis of analog and mixed-signal systems from VHDL-AMS. **IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems**, v. 22, p. 1504–1520, 2003.

DONNAY, S.; GIELEN, G.; SANSEN, W.; KRUIKAMP, W.; LEENAERTS, D.; BUYTAERT, S.; MARENT, K.; BUCKENS, M.; DAS, C. Using top-down CAD tools for mixed analog/digital ASICs: a practical design case. **Analog Integrated Circuits and Signal Processing**, Kluwer Academic Publishers, v. 10, n. 1-2, p. 101–117, 1996. ISSN 0925-1030.

DUFFIN BITTNER, L. R. J. Geometric programming - theory and application. **ZAMM - Journal of Applied Mathematics and Mechanics**, WILEY-VCH Verlag, v. 47, n. 8, p. 561–561, 1967.

ECKELAERT, T.; MCCONAGHY, T.; GIELEN, G. G. E. Efficient multiobjective synthesis of analog circuits using hierarchical Pareto-optimal performance hypersurfaces. In: **Design, Automation, and Test in Europe**. Munich: IEEE, 2005. p. 1070–1075.

EL-TURKY, F. M.; PERRY, E. E. BLADES: An artificial intelligence approach to analog circuit design. **IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems**, v. 8, p. 680–692, 1989.

ERFANI, T.; UTUZHNIKOV, S. V. Directed search domain: a method for even generation of Pareto frontier in multiobjective optimization. **Engineering Optimization**, p. 1–17, 2010.

FACCIO, F.; GONELLA, L. **Radiation tolerance of the 130 nm technology**. [S.l.], 2006.

FAKHFAKH, M.; COOREN, Y.; SALLEM, A.; LOULOU, M.; SIARRY, P. Analog circuit design optimization through the particle swarm optimization technique. **Analog Integrated Circuits and Signal Processing**, v. 63, p. 71–82, 2010.

FAN, X.; ZHANG, H.; SANCHEZ-SINENCIO, E. A noise reduction and linearity improvement technique for a differential cascode LNA. **Solid-State Circuits, IEEE Journal of**, v. 43, n. 3, p. 588–599, 2008. ISSN 0018-9200.

FERNANDEZ, F.; MARTIN, I.; RODRIGUEZ-VAZQUEZ, A.; HUERTAS, J. On simplification techniques for symbolic analysis of analog integrated circuits. In: **Circuits and Systems, 1992. ISCAS '92. Proceedings, 1992 IEEE International Symposium on**. San Diego: IEEE, 1992. v. 3, p. 1149–1152 vol.3.

FERNANDEZ, F.; RODRIGUEZ-VAZQUEZ, A.; HUERTAS, J. ASAP: a portable program for the symbolic analysis of analog integrated circuits. In: **Euro ASIC '90**. Paris: IEEE, 1990. p. 80–85.

FERNANDEZ, F.; RODRIGUEZ-VAZQUEZ, A.; HUERTAS, J. An advanced symbolic analyzer for the automatic generation of analog circuit design equations. In: **Circuits and Systems, 1991, IEEE International Symposium on**. Singapore: IEEE, 1991. p. 810–813 vol.2.

GIELEN, G.; MCCONAGHY, T.; EECKELAERT, T. Performance space modeling for hierarchical synthesis of analog integrated circuits. In: **Design Automation Conference, 2005. Proceedings. 42nd**. Anaheim: IEEE, 2005. p. 881–886.

GIELEN, G.; WALSHARTS, H.; SANSEN, W. ISAAC: a symbolic simulator for analog integrated circuits. **Solid-State Circuits, IEEE Journal of**, v. 24, n. 6, p. 1587–1597, Dec. 1989. ISSN 0018-9200.

GIELEN, G.; WAMBACQ, P.; SANSEN, W. Symbolic approximation strategies and the symbolic analysis of large and nonlinear circuits. In: **Circuits and Systems, 1991, IEEE International Symposium on**. Singapore: IEEE, 1991. p. 806–809 vol.2.

GIELEN, G. G. E.; RUTENBAR, R. A. Computer-aided design of analog and mixed-signal integrated circuits. **Proceedings of The IEEE**, v. 88, p. 1825–1854, 2000.

GIELEN, G. G. E.; WALSHARTS, H. C. C.; SANSEN, W. M. C. Analog circuit design optimization based on symbolic simulation and simulated annealing. **IEEE Journal of Solid-State Circuits**, v. 25, p. 707–713, 1990.

GRAEB, H.; ZIZALA, S.; ECKMUELLER, J.; ANTREICH, K. The sizing rules method for analog integrated circuit design. In: **Computer Aided Design, 2001. ICCAD 2001. IEEE/ACM International Conference on**. San Jose: IEEE, 2001. p. 343–349. ISSN 1092-3152.

GRAY, P. R.; HURST, P. J.; LEWIS, S. H.; MEYER, R. G. **Analysis and design of analog integrated circuits**. 4th ed. New York: Wiley, 2001.

GUERRA-GÓMEZ, I.; MCCONAGHY, T.; TLELO-CUAUTLE, E. Operating-point driven formulation for analog computer-aided design. **Analog Integrated Circuits and Signal Processing**, Springer US, v. 74, n. 2, p. 345–353, 2012. ISSN 0925-1030.

GUERRA-GÓMEZ, I.; TLELO-CUAUTLE, E.; FRAGA, L. G. de la. Richardson extrapolation-based sensitivity analysis in the multi-objective optimization of analog circuits. **Applied Mathematics and Computation**, v. 222, n. 0, p. 167–176, 2013. ISSN 0096-3003.

HAMAD, H.; MASSARA, R. A macromodeling approach to the synthesis of analog circuits using circuit decomposition and numerical circuit analysis. In: **Circuits and Systems, 1992, Proceedings of the 35th Midwest Symposium on**. Washington, DC: IEEE, 1992. p. 1108–1111 vol.2.

HANSEN, N.; ARNOLD, D. V.; AUGER, A. Evolution strategies. 2013. Disponível em: <https://www.lri.fr/~auger/publications.html>.

HANSEN, N.; OSTERMEIER, A. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In: **Evolutionary Computation, 1996., Proceedings of IEEE International Conference on**. Nagoya: IEEE, 1996. p. 312–317.

HARJANI, R.; RUTENBAR, R. A.; CARLEY, L. R. OASYS: A framework for analog circuit synthesis. **IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems**, v. 8, p. 1247–1266, 1989.

HARJANI, R.; SHAO, J. Feasibility and performance region modeling of analog and digital circuits. **Analog Integrated Circuits and Signal Processing**, Kluwer Academic Publishers, v. 10, n. 1-2, p. 23–43, 1996. ISSN 0925-1030.

HERIS, M. K. **MATLAB code for Multi-Objective Particle Swarm Optimization (MOPSO)**. 2011. Disponível em: <http://delta.cs.cinvestav.mx/~ccoello/EMOO/EMOOSOFTWARE.html>.

HERSHENSON, M. D. 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.

HINCAPIÉ, J. A. O. **Aplicação da Programação Geométrica no Projeto de Filtros Gm-C para Receptores RF CMOS**. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, São Paulo, 2010.

HIROYASU, T.; MIKI, M.; OGURA, M. Parallel simulated annealing using genetic crossover. In: **Parallel and Distributed Computing and Systems (PDCS 2000), 12th IASTED International Conference on**. Las Vegas: ACTA Press, 2000.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence**. Ann Arbor: University of Michigan Press, 1975.

HORI, S.; MAEDA, T.; MATSUNO, N.; YOSHIDA, N. Low-power widely tunable Gm-C filter with an adaptive DC-blocking, triode-biased MOSFET transconductor. In: **Solid-State Circuits Conference, 2004. ESSCIRC 2004. Proceeding of the 30th European**. Leuven: IEEE, 2004. p. 99–102.

HORI, S.; MAEDA, T.; YANO hitoshi; MATSUNO, N.; NUMATA, K.; YOSHIDA, N.; TAKAHASHI, Y.; YAMASE, T.; WALKINGTON, R.; HIKARU, H. A widely tunable CMOS Gm-C filter with a negative source degeneration resistor transconductor. In: **Solid-State Circuits Conference, 2003. ESSCIRC 2003. Proceedings of the 29th European**. Estoril: IEEE, 2003. p. 449–452.

HSU, J.-J.; SECHEN, C. DC small signal symbolic analysis of large analog integrated circuits. **Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on**, v. 41, n. 12, p. 817–828, Dec. 1994. ISSN 1057-7122.

HUANG, P.-H.; LIN, H.; LIN, Y.-T. A simple subthreshold CMOS voltage reference circuit with channel-length modulation compensation. **Circuits and Systems II: Express Briefs, IEEE Transactions on**, v. 53, n. 9, p. 882–885, Sept. 2006. ISSN 1549-7747.

HUGHES, E. Evolutionary many-objective optimisation: many once or one many? In: **Evolutionary Computation. The 2005 IEEE Congress on**. Edinburgh: IEEE, 2005. v. 1, p. 222–227 Vol.1.

HUYER, W.; NEUMAIER, A. Global optimization by multilevel coordinate search. **Journal of Global Optimization**, Kluwer Academic Publishers, v. 14, n. 4, p. 331–355, 1999. ISSN 0925-5001.

INGBER, A. L.; INGBER, L. **Very Fast Simulated Re-Annealing**. 1989.

JENSEN, M. T. Helper-objectives: Using multi-objective evolutionary algorithms for single-objective optimisation. **Journal of Mathematical Modelling and Algorithms**, v. 3, n. 4, p. 323–347, 2004.

JERKE, G.; LIENIG, J. Constraint-driven design: the next step towards analog design automation. In: **Proceedings of the 2009 international symposium on Physical design**. New York, NY, USA: ACM, 2009. (ISPD '09), p. 75–82. ISBN 978-1-60558-449-2.

KARDONTCHIK, J. **Introduction to the Design of Transconductor-Capacitor Filters**. Norwell: Kluwer Academic Publishers, 1992. ISBN 9780792391951.

KENNEDY, J.; EBERHART, R. C. Particle swarm optimization. In: **IEEE International Conference on Neural Networks**. Perth: IEEE, 1995. p. 1942–1948.

KERAMAT, M.; KIELBASA, R. OPTOMEGA: An environment for analog circuit optimization. In: **Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on**. Monterey: IEEE, 1998. v. 6, p. 122–125 vol.6.

KIRKPATRICK, S.; GELATT, J. C. D.; VECCHI, M. P. Optimization by simulated annealing. **Science**, v. 220, p. 671–680, 1983.

KNOWLES, J.; WATSON, R.; CORNE, D. Reducing local optima in single-objective problems by multi-objectivization. In: ZITZLER, E.; THIELE, L.; DEB, K.; COELLO, C. C.; CORNE, D. (Ed.). **Evolutionary Multi-Criterion Optimization**. Berlin: Springer Berlin Heidelberg, 2001, (Lecture Notes in Computer Science, v. 1993). p. 269–283. ISBN 978-3-540-41745-3.

KRASNICKI, M.; PHELPS, R.; RUTENBAR, R.; CARLEY, L. MAELSTROM: Efficient simulation-based synthesis for custom analog cells. In: **Design Automation Conference, 1999. Proceedings. 36th**. New Orleans: IEEE, 1999. p. 945–950.

LAGRANGE, J. de. **Lecons sur le calcul des fonctions**. Paris: Courcier, 1806.

LEWINSKI, A.; SILVA-MARTINEZ, J. A 30-Mhz fifth-order elliptic low-pass CMOS filter with 65-dB spurious-free dynamic range. **Circuits and Systems I: Regular Papers, IEEE Transactions on**, v. 54, n. 3, p. 469–480, Mar. 2007. ISSN 1549-8328.

LI, L.; XIAOWEI, C.; ZHEYING, L. The design of sub-threshold reference circuit using resistor temperature compensation. In: **Circuits and Systems, 2009. MWSCAS '09. 52nd IEEE International Midwest Symposium on**. Cancun: IEEE, 2009. p. 78–81. ISSN 1548-3746.

LI, X.; WANG, J.; PILEGGI, L.; CHEN, T.-S.; CHIANG, W. Performance-centering optimization for system-level analog design exploration. In: **Computer-Aided Design, 2005. ICCAD-2005. IEEE/ACM International Conference on**. San Jose, CA: IEEE, 2005. p. 422–429.

LIN, C.; KALKUR, T.; MORIN, M. A 2.4GHz common-gate LNA using on-chip differential inductors in a 0.18 μm CMOS technology. In: **Electrical, Communications, and Computers, 2009. CONIELECOMP 2009. International Conference on**. Cholula: IEEE, 2009. p. 183–188.

LIN, P.; ALDERSON, G. SNAP: symbolic-network-analysis program. **Electrical Engineers, Proceedings of the Institution of**, v. 119, n. 2, p. 160, feb. 1972. ISSN 0020-3270.

LIN, S. **NGPM - A NSGA-II Program in Matlab v1.4**. 2011. Disponível em: <http://www.mathworks.com/matlabcentral/fileexchange/31166-ngpm-a-nsga-ii-program-in-matlab-v1-4>.

LINTEN, D.; ASPEMYR, L.; JEAMSAKSIRI, W.; RAMOS, J.; MERCHA, A.; JENEI, S.; THIJS, S.; GARCIA, R.; JACOBSSON, H.; WAMBACQ, P.; DONNAY, S.; DECOUTERE, S. Low-power 5 GHz LNA and VCO in 90 nm RF CMOS. In: **VLSI Circuits, 2004. Digest of Technical Papers. 2004 Symposium on**. Honolulu: IEEE, 2004. p. 372–375.

LIU, B.; FERNANDEZ, F.; GIELEN, G. Efficient and accurate statistical analog yield optimization and variation-aware circuit sizing based on computational intelligence techniques. **Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on**, v. 30, n. 6, p. 793–805, june 2011. ISSN 0278-0070.

LIU, H.; SINGHEE, A.; RUTENBAR, R.; CARLEY, L. Remembrance of circuits past: macro-modeling by data mining in large analog design spaces. In: **Design Automation Conference, 2002. Proceedings. 39th**. New Orleans: IEEE, 2002. p. 437 – 442. ISSN 0738-100X.

LO, T.-Y.; HUNG, C.-C.; ISMAIL, M. A wide tuning range Gm-C filter for multi-mode CMOS direct-conversion wireless receivers. **Solid-State Circuits, IEEE Journal of**, v. 44, n. 9, p. 2515–2524, Sept. 2009. ISSN 0018-9200.

LOCATELLI, M. Simulated annealing algorithms for continuous global optimization. In: **Handbook of global optimization**. [S.l.]: Springer, 2002. p. 179–229.

LOCHTEFELD, D. **Multi-objectivization in Genetic Algorithms**. Tese (Doutorado) — Wright State University, 2011.

LOURENÇO, N.; CANELAS, A.; PÓVOA, R.; MARTINS, R.; HORTA, N. Floorplan-aware analog IC sizing and optimization based on topological constraints. **Integration, the VLSI Journal**, v. 48, n. 0, p. 183 – 197, 2015. ISSN 0167-9260.

LUKE, S. **Essentials of Metaheuristics**. Lulu, 2009. Disponível em: <http://cs.gmu.edu/~sean/book/metaheuristics/>.

MALAK, A.; LI, Y.; ISKANDER, R.; DURBIN, F.; JAVID, F.; GUEBHARD, J.-M.; LOUERAT, M.-M.; TISSOT, A. Fast multidimensional optimization of analog circuits initiated by monodimensional global Peano explorations. **Integration, the VLSI Journal**, v. 48, p. 198 – 212, 2015. ISSN 0167-9260.

MANN, H. B.; WHITNEY, D. R. On a test of whether one of two random variables is stochastically larger than the other. **The Annals of Mathematical Statistics**, Institute of Mathematical Statistics, v. 18, n. 1, p. 50–60, 1947.

MARLER, R. T.; ARORA, J. S. Survey of multi-objective optimization methods for engineering. **Structural and Multidisciplinary Optimization**, Springer, v. 26, n. 6, p. 369–395, Apr. 2004.

MARTENS, E.; GIELEN, G. Top-down heterogeneous synthesis of analog and mixed-signal systems. In: **Proceedings of the conference on Design, Automation and Test in Europe**. Munich: IEEE, 2006.

MASSIER, T.; GRAEB, H. E.; SCHLICHTMANN, U. The sizing rules method for CMOS and bipolar analog integrated circuit synthesis. **IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems**, v. 27, p. 2209–2222, 2008.

MAS'UD, A. F.; OHTSUKA, T.; KUNIEDA, H. Translation of specifications in hierarchical analog LSI design. In: **Circuits and Systems, 1996. ISCAS '96, Connecting the World, 1996 IEEE International Symposium on**. [S.l.: s.n.], 1996. v. 4, p. 735–738.

MATEUS, J.; ROA, E.; HERNANDEZ, H.; NOIJE, W. V. A $2.7\mu\text{A}$ sub 1-V voltage reference. **Proceedings of IEEE Symposium on Integrated Circuits and System Design**, p. 81–84, 2008.

MATHWORKS. **Global Optimization Toolbox: User's Guide (R2013b)**. 2013. Disponível em: <http://www.mathworks.com/products/global-optimization/>.

MCCONAGHY, T. FFX: Fast, scalable, deterministic symbolic regression technology. In: RI-OLO, R.; VLADISLAVLEVA, E.; MOORE, J. H. (Ed.). **Genetic Programming Theory and Practice IX**. New York: Springer, 2011, (Genetic and Evolutionary Computation). p. 235–260. ISBN 978-1-4614-1769-9.

MCCONAGHY, T.; GIELEN, G. G. E. Globally reliable variation-aware sizing of analog integrated circuits via response surfaces and structural homotopy. **IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems**, v. 28, p. 1627–1640, 2009.

MEDEIRO, F.; FERNANDEZ, F.; DOMINGUEZ-CASTRO, R.; RODRIGUEZ-VAZQUEZ, A. A statistical optimization-based approach for automated sizing of analog cells. In: **Computer-Aided Design, 1994., IEEE/ACM International Conference on**. San Jose, CA: IEEE, 1994. p. 594–597. ISSN 1063-6757.

MOLGA, M.; SMUTNICKI, C. **Test functions for optimization needs**. [S.l.], 2005.

MOREIRA, P.; ANELLI, G.; BALDI, S.; BONACINI, S.; CERVELLI, G.; CHRISTIANSEN, J.; DESPEISSE, M.; FACCIO, F.; HAENSLER, K.; KLOUKINAS, K.; MARCHIORO, A.; SZCZYGIEL, R. A voltage reference circuit in 130 nm CMOS. In: **Proc. 5th Int. Meeting Front End Electronics for High Energy, Nuclear, Medical, and Space Applications**. Snowmass, CO: [s.n.], 2003.

MOSTAGHIM, S.; TEICH, J. The role of ϵ -dominance in multi objective particle swarm optimization methods. In: **Evolutionary Computation, 2003. CEC '03. The 2003 Congress on**. [S.l.: s.n.], 2003. v. 3, p. 1764–1771 Vol.3.

MUELLER-GRITSCHNEDER, D.; GRAEB, H. Computation of yield-optimized Pareto fronts for analog integrated circuit specifications. In: **Design, Automation Test in Europe Conference Exhibition (DATE), 2010**. Dresden: IEEE, 2010. p. 1088–1093. ISSN 1530-1591.

MUELLER-GRITSCHNEDER, D.; GRAEB, H.; SCHLICHTMANN, U. A successive approach to compute the bounded Pareto front of practical multiobjective optimization problems. **SIAM J. on Optimization**, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, v. 20, n. 2, p. 915–934, July 2009. ISSN 1052-6234.

NEUMAIER, A. **MCS: Global Optimization by Multilevel Coordinate Search**. 2000. Disponível em: <<http://www.mat.univie.ac.at/~neum/software/mcs/>>.

NOVAL, J. J. S. **Metodologia para a Otimização do Rendimento e Desempenho dos Circuitos Analógicos usando Programação Geométrica**. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, São Paulo, 2012.

NYE, W.; RILEY, D.; SANGIOVANNI-VINCENTELLI, A.; TITS, A. DELIGHT.SPICE: an optimization-based system for the design of integrated circuits. **Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on**, v. 7, n. 4, p. 501–519, 1988. ISSN 0278-0070.

OCHOTTA, E. S.; RUTENBAR, R. A.; CARLEY, L. R. Synthesis of high-performance analog circuits in ASTRX/OBLX. **IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems**, v. 15, p. 273–294, 1996.

ONBASOGLU, E.; ÖZDAMAR, L. Parallel simulated annealing algorithms in global optimization. **Journal of Global Optimization**, Kluwer Academic Publishers, v. 19, n. 1, p. 27–50, 2001. ISSN 0925-5001.

PEREIRA, A. I.; FERNANDES, E. M. A study of simulated annealing variants. In: **XXVIII Congreso de Estadística e Investigación Operativa**. Cadiz, Spain: Sociedad de Estadística e Investigación Operativa, 2004.

PHELPS, R.; KRASNICKI, M.; RUTENBAR, R. A.; CARLEY, L. R.; HELLUMS, J. R. 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.

PLAS, G. V. der; DEBYSER, G.; LEYN, F.; LAMPAERT, K.; VANDENBUSSCHE, J.; GIELEN, G.; VESELINOVIC, P.; LEENARTS, D. AMGIE-A synthesis environment for CMOS analog integrated circuits. **IEEE TCAD**, 1037, 2001.

POPOV, A. **MOEA - User Manual**. [S.l.], 2005.

RAMACHANDRAN, A. **Nonlinearity and noise modeling of operational transconductance amplifiers for continuous time analog filters**. Dissertação (Mestrado) — Texas AM University, 2005.

RAO, S. S. **Engineering Optimization: Theory and Practice**. Hoboken: New Age Publishers, 1996.

RAZAVI, B. **Design of Analog CMOS Integrated Circuits**. 1st ed. New York: McGraw-Hill, Inc., 2001.

RECHENBERG, I. Cybernetic solution path of an experimental problem. **Royal Aircraft Establishment Library Translation**, 1965.

REYES-SIERRA, M.; COELLO, C. A. C. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. **International Journal of Computational Intelligence Research**, v. 2, p. 287–308, 2006.

RIOS, L. M.; SAHINIDIS, N. V. Derivative-free optimization: a review of algorithms and comparison of software implementations. **Journal of Global Optimization**, Springer US, v. 56, n. 3, p. 1247–1293, 2013. ISSN 0925-5001.

SAARI, V. **Continuous-time low-pass filters for integrated wideband radio receivers**. New York: Springer-Verlag, 2012.

SAKATA, S. **ASAMIN - Matlab gateway function to Lester Ingber's Adaptive Simulated Annealing (ASA)**. 1999. Disponível em: <http://ssakata.sdf.org/software/>.

SEDA, S.; DEGRAUWE, M.; FICHTNER, W. A symbolic analysis tool for analog circuit design automation. In: **Computer-Aided Design, 1988. ICCAD-88. Digest of Technical Papers, IEEE International Conference on**. Santa Clara, CA: IEEE, 1988. p. 488–491.

SEGURA, C.; COELLO, C. A. C.; MIRANDA, G.; LEÓN, C. Using multi-objective evolutionary algorithms for single-objective optimization. **4OR**, Springer Berlin Heidelberg, v. 11, n. 3, p. 201–228, 2013. ISSN 1619-4500.

SHAEFFER, D. K.; LEE, T. H. A 1.5-V, 1.5-GHz CMOS Low Noise Amplifier. **IEEE Journal of Solid-State Circuits**, v. 32, p. 745–759, 1997.

SHICHMAN, H.; HODGES, D. Modeling and simulation of insulated-gate field-effect transistor switching circuits. **Solid-State Circuits, IEEE Journal of**, v. 3, n. 3, p. 285–289, 1968. ISSN 0018-9200.

SMEDT, B. D.; GIELEN, G. HOLMES: capturing the yield-optimized design space boundaries of analog and RF integrated circuits. In: **Design, Automation and Test in Europe Conference and Exhibition, 2003**. Munich: IEEE, 2003. p. 256–261. ISSN 1530-1591.

SOUSA, F. J. P. A. de. **Radiation Tolerant Low Power 12 bit ADC in 130 nm CMOS Technology**. Dissertação (Mestrado) — Faculdade de Engenharia da Universidade do Porto, 2009.

STEFANOVIC, D.; KAYAL, M. **Structured Analog CMOS Design**. Dordrecht: Springer Netherlands, 2008. ISBN 1402085729, 9781402085727.

STEHR, G.; GRAEB, H.; ANTREICH, K. Analog performance space exploration by Fourier-Motzkin elimination with application to hierarchical sizing. In: **Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference on**. San Jose, CA: IEEE, 2004. p. 847 – 854. ISSN 1092-3152.

STEHR, G.; GRAEB, H.; ANTREICH, K. Analog performance space exploration by normal-boundary intersection and by Fourier-Motzkin elimination. **Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on**, v. 26, n. 10, p. 1733 –1748, Oct. 2007. ISSN 0278-0070.

SYNOPSYS. **HSPICE® User Guide: Basic Simulation and Analysis**. Mountain View, 2014.

TAN, X.-D.; SHI, C.-J. Interpretable symbolic small-signal characterization of large analog circuits using determinant decision diagrams. In: **Design, Automation and Test in Europe Conference and Exhibition 1999. Proceedings**. Munich: IEEE, 1999. p. 448 –453.

THE International Technology Roadmap for Semiconductors (ITRS). [S.l.], 2011. Disponível em: <<http://www.itrs.net/>>.

TIWARY, S. K.; TIWARY, P. K.; RUTENBAR, R. A. Generation of yield-aware Pareto surfaces for hierarchical circuit design space exploration. In: **Design Automation Conference**. San Francisco: IEEE, 2006. p. 31–36.

TIWARY, S. K.; VELU, S.; RUTENBAR, R. A.; MUKHERJEE, T. Pareto optimal modeling for efficient PLL optimization. In: **Technical Proceedings of the 2004 NSTI Nanotechnology Conference and Trade Show**. Boston: [s.n.], 2004.

VALLEE, R. E.; EL-Masry, E. I. A very high-frequency CMOS complementary folded cascode amplifier. **IEEE Journal of Solid-State Circuits**, v. 29, n. 2, p. 130–133, 1994.

VANDEBUSSCHE, J.; STEYAERT, M.; GIELEN, G. **Systematic Design of Analog IP Blocks**. Norwell, MA, USA: Kluwer Academic Publishers, 2003. ISBN 1402074719.

VANDERBILT, D.; LOUIE, S. G. A Monte Carlo simulated annealing approach to optimization over continuous variables. **Journal of Computational Physics**, v. 56, n. 2, p. 259 – 271, 1984.

VAZ, A.; VICENTE, L. N. A particle swarm pattern search method for bound constrained global optimization. **Journal of Global Optimization**, Springer US, v. 39, n. 2, p. 197–219, 2007. ISSN 0925-5001.

VAZ, A.; VICENTE, L. N. **PSwarm Home Page**. 2009. Disponível em: <<http://www.norg.uminho.pt/aivaz/pswarm/>>.

VELASCO-JIMENEZ, M.; CASTRO-LOPEZ, R.; ROCA, E.; FERNANDEZ, F. Implementation issues in the hierarchical composition of performance models of analog circuits. In: **Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014**. Dresden: IEEE, 2014. p. 1–6.

VITAL, J. C.; HORTA, N. C.; SILVA, N. S.; FRANCA, J. E. CATALYST: a highly flexible CAD tool for architecture-level design and analysis of data converters. In: **Design Automation, 1993, with the European Event in ASIC Design. Proceedings. [4th] European Conference on**. Paris: IEEE, 1993. p. 472 –477.

WANG, P. P.; CHEN, D.-S. Continuous optimization by a variant of simulated annealing. **Comput. Optim. Appl.**, Kluwer Academic Publishers, Norwell, v. 6, n. 1, p. 59–71, July 1996. ISSN 0926-6003.

WEBER, T. O.; CHAPARRO, S.; NOIJE, W. V. Synthesis of a narrow-band low noise amplifier in a 180 nm CMOS technology using simulated annealing with crossover operator. In: **Proceedings of the 26th Symposium on Integrated Circuits and Systems**. Curitiba: IEEE, 2013.

WEBER, T. O.; NOIJE, W. A. M. V. Analog design synthesis method using simulated annealing and particle swarm optimization. In: **Proceedings of the 24th Symposium on Integrated Circuits and Systems**. João Pessoa, Brasil: IEEE, 2011. p. 85–90. ISBN 978-1-4503-0828-1.

WEBER, T. O.; NOIJE, W. A. M. V. Analog design synthesis performing fast Pareto frontier exploration. In: **Proceedings of the 2nd IEEE Latin American Symposium on Circuits and Systems**. Bogotá: IEEE, 2011. p. 62–66. ISBN 978-1-4244-9484-2.

WEBER, T. O.; NOIJE, W. A. M. V. Analog circuit synthesis performing fast Pareto frontier exploration and analysis through 3D graphs. **Analog Integrated Circuits and Signal Processing**, Springer US, v. 73, n. 3, p. 861–871, 2012. ISSN 0925-1030.

WEBER, T. O.; NOIJE, W. A. M. V. Design of analog integrated circuits using simulated annealing/quenching with crossovers and particle swarm optimization. In: TSUZUKI, M. de S. G. (Ed.). **Simulated Annealing - Advances, Applications and Hybridizations**. [S.l.]: InTech, 2012. p. 219–244. ISBN 978-953-51-0710-1.

WEBER, T. O.; NOIJE, W. A. M. V. Multi-objective design of analog integrated circuits using simulated annealing with crossover operator and weight adjusting. **Journal of Integrated Circuits and Systems (JICS)**, v. 7, 2012.

WILSON, R. B. **A Simplicial Algorithm for Concave Programming**. Tese (Doutorado), Cambridge, 1963.

WOLFE, G.; VEMURI, R. Extraction and use of neural network models in automated synthesis of operational amplifiers. **IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems**, v. 22, p. 198–212, 2003.

WOLPERT, D. H.; MACREADY, W. G. No free lunch theorems for optimization. **IEEE Transactions on Evolutionary Computation**, v. 1, n. 1, p. 67–82, 1997.

ZITZLER, E.; DEB, K.; THIELE, L. Comparison of multiobjective evolutionary algorithms: Empirical results. **Evolutionary Computation**, v. 8, p. 173–195, 2000.

ZITZLER, E.; LAUMANN, M.; THIELE, L. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: GIANNAKOGLU, K. C.; TSAHALIS, D. T.; PÉRIAUX, J.; PAPAILIOU, K. D.; FOGARTY, T. (Ed.). **Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems**. Athens, Greece: International Center for Numerical Methods in Engineering, 2001. p. 95–100.

ZITZLER, E.; THIELE, L. Multiobjective optimization using evolutionary algorithms - a comparative case study. In: EIBEN, A.; BÄCK, T.; SCHOENAUER, M.; SCHWEFEL, H.-P. (Ed.). **Parallel Problem Solving from Nature - PPSN V**. Berlin: Springer Berlin Heidelberg, 1998, (Lecture Notes in Computer Science, v. 1498). p. 292–301. ISBN 978-3-540-65078-2.

ZITZLER, E.; THIELE, L.; LAUMANN, M.; FONSECA, C.; FONSECA, V. da. Performance assessment of multiobjective optimizers: an analysis and review. **Evolutionary Computation, IEEE Transactions on**, v. 7, n. 2, p. 117–132, Apr. 2003. ISSN 1089-778X.

ZOU, J.; MUELLER, D.; GRAEB, H.; SCHLICHTMANN, U. A CPPLL hierarchical optimization methodology considering jitter, power and locking time. In: **Design Automation Conference, 2006 43rd ACM/IEEE**. San Francisco: IEEE, 2006. p. 19–24. ISSN 0738-100X.

APÊNDICE A – EXEMPLOS DE ARQUIVOS DE ENTRADA

A ferramenta desenvolvida neste trabalho utiliza arquivos de texto para configuração do sintetizador e descrição dos circuitos. Este apêndice fornece um exemplo dos arquivos de configuração, de especificações, de topologia e de descrição de um *test bench*. Mais detalhes podem ser encontrados na seção 4.1. Os primeiros exemplos descritos nestes arquivos são referentes a um amplificador Miller em tecnologia 0,35 μm . O último exemplo refere-se a seção de fronteira de Pareto e descrição de variáveis de um filtro G_m -C em tecnologia 180 nm.

A.1 Exemplo de Arquivo de Configuração de um Amplificador *Miller*

```

1  techname{1} = c35_tm
2  techname{2} = c35_ws
3  techname{3} = c35_wp
4  techname{4} = c35_wo
5  techname{5} = c35_wz
6  Wmin = 0.6e-6
7  Lmin = 0.35e-6
8  Wmax = 100e-6
9  Lmax = 20e-6
10 nmosmodel = CMOSN
11 pmosmodel = CMOSP
12 ic_compute = 0
13 inputtoponame = topology
14 inputspecname = specs
15 cirname = miller
16 testbenchname{1} = tb_prim
17 testbenchname{2} = tb_sr
18 testbenchname{3} = tb_inputrange
19 testbenchname{4} = tb_outputswing
20 testbenchname{5} = tb_cmrr
21 testbenchname{6} = tb_psrr
22 vdd = 1.65
23 vss = -1.65
24 cl = 10e-12
25 simulator = hspice

```

A.2 Exemplo de Arquivo de Topologia de um Amplificador Miller

```

1  * Amplificador Miller
2  .circuit
3  M1 7 3 5 5 CMOSF W = "W1" L = "L1" M="N1" AD="0.85u*W1*int((N1+1)/2)/N1" AS="0.85u*W1*(int
   (N1/2+1)-0.5)/N1" PD="(1.7u+2*W1)*int((N1+1)/2)/N1" PS="(1.7u+2*W1)*(int(N1/2+1)-0.5)/
   N1"
4  M2 8 4 5 5 CMOSF W = "W1" L = "L1" M="N1" AD="0.85u*W1*int((N1+1)/2)/N1" AS="0.85u*W1*(int
   (N1/2+1)-0.5)/N1" PD="(1.7u+2*W1)*int((N1+1)/2)/N1" PS="(1.7u+2*W1)*(int(N1/2+1)-0.5)/
   N1"
5  M3 7 7 13 13 CMOSN W = "W3" L = "L3" M="N3" AD="0.85u*W3*int((N3+1)/2)/N3" AS="0.85u*W3*(
   int(N3/2+1))/N3" PD="(1.7u+2*W3)*int((N3+1)/2)/N3" PS="(1.7u+2*W3)*(int(N3/2+1))/N3"
6  M4 8 7 13 13 CMOSN W = "W3" L = "L3" M="N3" AD="0.85u*W3*int((N3+1)/2)/N3" AS="0.85u*W3*
   int(N3/2+1)/N3" PD="(1.7u+2*W3)*int((N3+1)/2)/N3" PS="(1.7u+2*W3)*int(N3/2+1)/N3"
7  M5 5 1 14 14 CMOSF W = "W8" L = "L8" M="N5" AD="0.85u*W8*int((N5+1)/2)/N5" AS="0.85u*W8*
   int(N5/2+1)/N5" PD="(1.7u+2*W8)*int((N5+1)/2)/N5" PS="(1.7u+2*W8)*int(N5/2+1)/N5"
8  M6 12 1 14 14 CMOSF W = "W8" L = "L8" M="N6" AD="0.85u*W8*int((N6+1)/2)/N6" AS="0.85u*W8*
   int(N6/2+1)/N6" PD="(1.7u+2*W8)*int((N6+1)/2)/N6" PS="(1.7u+2*W8)*int(N6/2+1)/N6"
9  M7 12 8 13 13 CMOSN W = "W7" L = "L7" M="N7" AD="0.85u*W7*int((N1+1)/2)/N1" AS="0.85u*W7*
   int(N1/2+1)/N1" PD="(1.7u+2*W7)*int((N1+1)/2)/N1" PS="(1.7u+2*W7)*int(N1/2+1)/N1"
10 M8 1 1 14 14 CMOSF W = "W8" L = "L8" M="N8" AD="0.85u*W8*int((N8+1)/2)/N8" AS="0.85u*W8*
   int(N8/2+1)/N8" PD="(1.7u+2*W8)*int((N8+1)/2)/N8" PS="(1.7u+2*W8)*int(N8/2+1)/N8"
11 M9 1 1 13 13 CMOSN W = "W9" L = "L9" M="N9" AD="0.85u*W9*int((N9+1)/2)/N9" AS="0.85u*W9*
   int(N9/2+1)/N9" PD="(1.7u+2*W9)*int((N9+1)/2)/N9" PS="(1.7u+2*W9)*int(N9/2+1)/N9"
12 Cmillar 8 12 "C1"
13 .close
14
15 .links
16 linkin VDD = 14
17 linkin VSS = 13
18 linkin INP = 4
19 linkin INN = 3
20 linkout OUT = 12
21 vdd
22 .close
23
24 .include
25 * sem modelos extras nesse exemplo
26 .close
27
28 .params
29 .param C1 = 1e-12 float var 1e-12 10e-12
30 .param W1 = 0 float var 0 0 * Quando os valores de mínimo e máximo são 0, são usados os
   valores do arquivo de configurações para facilitar a troca de tecnologia
31 .param L1 = 0 float var 0 0
32 .param W3 = 0 float var 0 0
33 .param L3 = 0 float var 0 0
34 .param W7 = 0 float var 0 0
35 .param L7 = 0 float var 0 0
36 .param W8 = 0 float var 0 0
37 .param L8 = 0 float var 0 0
38 .param W9 = 0 float var 0 0
39 .param L9 = 0 float var 0 0
40 .param N1 = 1 integer var 1 10
41 .param N3 = 1 integer var 1 10
42 .param N5 = 1 integer var 1 10
43 .param N6 = 1 integer var 1 10
44 .param N7 = 1 integer var 1 10
45 .param N8 = 1 integer var 1 10
46 .param N9 = 1 integer var 1 10
47 .close
48
49 .operationregion
50 .region M1 = saturation 0.1
51 .region M2 = saturation 0.1
52 .region M3 = saturation 0.1
53 .region M4 = saturation 0.1
54 .region M5 = saturation 0.1
55 .region M6 = saturation 0.1
56 .region M7 = saturation 0.1
57 .region M8 = saturation 0.1
58 .region M9 = saturation 0.1
59 .close

```

A.3 Exemplo de Arquivo de Especificações de um Amplificador *Miller*

```

1  UGF      AC      trig,at=1,targ,vdb(OUT),val=0,fall=1  maximize  15e6  [Hz]  tb_prim an-ac 1 0
2  OUTAC    AC      FIND,V(OUT),AT=1                    none      0      [V]  tb_prim an-ac 0 0
3  INAC     AC      FIND,V(INP,INN),AT=1                 none      0      [V]  tb_prim an-ac 0 0
4  GAINAC   AC      PARAM,('20*log10(OUTAC/INAC)')        maximize  80     dB[V] tb_prim an-ac 1 1
5  MIN_PH   AC      MIN,vp(OUT),from=1,to='UGF'          none      none  [°]  tb_prim an-ac 0 0
6  PH_MARGINmin AC  PARAM,('180+MIN_PH')                 maximize  60     [°]  tb_prim an-ac 1 0
7  PH_MARGINmax AC  PARAM,('180+MIN_PH')                 minimize  90     [°]  tb_prim an-ac 1 0
8  TH1      TRAN    WHEN,V(OUT)='0',CROSS=1              none      0      [s]  tb_sr an-tran 0 0
9  TH2      TRAN    WHEN,V(OUT)='0',CROSS=2              none      0      [s]  tb_sr an-tran 0 0
10 SRPOS    TRAN    DERIV,V(OUT),AT='TH1'                maximize  18e6  [V/s] tb_sr an-tran 10 0.1
11 SRNEG    TRAN    DERIV,V(OUT),AT='TH2'                minimize  -18e6  [V/s] tb_sr an-tran 10 0.1
12 ISUPtemp DC      FIND,I(VSUP),AT=0                   none      350e-6  [A]  tb_inputrange an-dc 0 0
13 ISUPPLY   DC      PARAM,('ISUPtemp')                  minimize  350e-6  [A]  tb_inputrange an-dc 1 1
14 SWMEAN    DC      MIN,DERIV('V(OUT)'),FROM=0,TO=,0    none      0      [V]  tb_inputrange an-dc 0 0
15 MAXDERIV  DC      MAX,DERIV('I(Xopamp.M5)')           none      0      [V]  tb_inputrange an-dc 0 0
16 VIMIN     DC      FIND,V(INP),WHEN,DERIV('V(OUT)')='SWMEAN*0.9',CROSS=1 none 0 [V]  tb_inputrange an-dc 0 0
17 VIMAX     DC      FIND,V(3),WHEN,DERIV('I(Xopamp.M5)')='0.1*MAXDERIV',RISE=LAST maximize  0.4 [V]  tb_inputrange an-dc 1 0
18 ICMR      DC      PARAM,('VIMAX-VIMIN')               maximize  1.5    [V]  tb_inputrange an-dc 0.1 0
19 OFFSET    DC      FIND,V(OUT),AT=0                   none      5e-3    [V]  tb_inputrange an-dc 1 0
20 OFFSETabs DC      PARAM,('ABS(OFFSET)')               minimize  5e-3    [V]  tb_inputrange an-dc 0 0
21 SWMNI     DC      MIN,DERIV('V(OUT)'),FROM=vssupply,TO,=,vddsupply none 0 [V]  tb_outputswing an-dc 0 0
22 SWMNA     DC      FIND,DERIV('V(OUT)'),AT=0           none      0      [V]  tb_outputswing an-dc 0 0
23 VOMAX     DC      FIND,V(OUT),WHEN,DERIV('V(OUT)')='SWMNI+ABS(SWMNI)*0.5',CROSS=1 maximize  0 [V]  tb_outputswing an-dc 0 0
24 VOMIN     DC      FIND,V(OUT),WHEN,DERIV('V(OUT)')='SWMNI+ABS(SWMNI)*0.5',CROSS=LAST none 0 [V]  tb_outputswing an-dc 0 0
25 OUTSWING  DC      PARAM,('VOMAX-VOMIN')               maximize  2      [V]  tb_outputswing an-dc 1 0
26 VOU       AC      FIND,V(OUT),AT=1                    none      0      [V]  tb_cmrr ac 0 0
27 VCM       AC      FIND,V(INN),AT=1                    none      0      [V]  tb_cmrr ac 0 0
28 CMRRDB    AC      PARAM,('20*LOG10(VCM/VOU)')          maximize  80     dB[V] tb_cmrr ac 1 0
29 VUO       AC      FIND,V(OUT),AT=1                    none      0      [V]  tb_psrr ac 0 0
30 VPSRRDD   AC      FIND,V(VDD),AT=1                    none      0      [V]  tb_psrr ac 0 0
31 VPSRRSS   AC      FIND,V(VSS),AT=1                    none      0      [V]  tb_psrr ac 0 0
32 PSRR      AC      PARAM,('20*LOG10(VPSRRDD/VUO)')      maximize  70     dB[V] tb_psrr ac 1 0
33 AREA      AC      param=('2*W1*L1*N1+2*W3*L3*N3+W8*L8*N5+W8*L8*N6+W7*L7*N7+W8*L8*N8+W9*L9*N9)/1e-12' minimize 950 [um2] tb_prim
an-ac 1 1

```

A.4 Exemplo de Arquivo de *Testbench* no Domínio da Frequência de um Amplificador *Miller*

```

1  * Testbench .AC e .OP
2  .circuit
3  VSUP 10 0 "vddsupply"
4  VSS 2 0 "vssupply"
5  Vip 3 0 0 AC 1
6  Vin 4 0 0
7  Cll 5 0 "cll"
8  Xopamp 3 4 10 2 5 opamp
9  .close
10
11 .links
12 linkin VSUPPLY = VSUP
13 linkin VDD = 10
14 linkin VSS = 2
15 linkin INP = 3
16 linkin INN = 4
17 linkout OUT = 5
18 .close
19
20 .analysis
21 an-op = OP
22 an-ac = AC oct 10 1 1e9
23 .close

```

A.5 Exemplo de Seção de Descrição da Topologia da Fronteira de Pareto e Seção de Descrição de Variáveis em Nível de Sistema de um Filtro G_m -C

```

1 .topologies
2 OTA_with_nonidealities1 = mo_ota_180nm (b1_gm1 = GMMAX; b1_gm1 = GMMIN; b1_tnoise1= TNOISE
; b1_fnoise1= FNOISE; b1_icmr1_1= icmr1; b1_icmr1_2= icmr2; b1_icmr1_3= icmr3;
b1_icmr1_4= icmr4; b1_icmr1_5= icmr5; b1_cin1 = CIN; b1_cout1 = COUT; b1_rout1 = ROUT;
b1_isupply1 = ISUPPLY; b1_areal = AREA; b1_midc1 = MIDCURRENTtemp; b1_maxc1 =
MAXCURRENT; b1_fnoisecmb1 = FNOISECMFB; b1_tnoisecmb1 = TNOISECMFB; b1_coutcmb1 =
COUTCMFB; b1_routcmb1 = ROUTCMFB; b1_isupplycmb1 = ISUPPLYCMFB; b1_areacmb1 =
AREAWCMFB)
3 .close
4
5 .params
6 .param b1_gm1 = 1.89e-6 indexed var 1e-6 250e-6
7 .param b1_icmr1_1 = 100e-3 indexed linked 50e-3 500e-3
8 .param b1_icmr1_2 = 100e-3 indexed linked 50e-3 500e-3
9 .param b1_icmr1_3 = 100e-3 indexed linked 50e-3 500e-3
10 .param b1_icmr1_4 = 100e-3 indexed linked 50e-3 500e-3
11 .param b1_icmr1_5 = 100e-3 indexed linked 50e-3 500e-3
12 .param b1_maxc1 = 10e-6 indexed linked 1e-7 1e3
13 .param b1_midc1 = 1e-7 indexed linked 1e-8 1e-3
14 .param b1_cout1 = 10e-15 indexed linked 5e-15 10e-12
15 .param b1_coutcmb1 = 1e-12 indexed linked 1e-15 10e-12
16 .param b1_fnoise1 = 100e-6 indexed linked 1e-6 100e-6
17 .param b1_tnoise1 = 100e-9 indexed linked 1e-9 100e-9
18 .param b1_tnoisecmb1 = 1e-9 indexed linked 1e-9 100e-9
19 .param b1_fnoisecmb1 = 1e-6 indexed linked 1e-6 100e-6
20 .param b1_rout1 = 500e3 indexed linked 1e3 100e6
21 .param b1_routcmb1 = 100e6 indexed linked 1e3 100e6
22 .param b1_isupply1 = 100e-6 indexed linked 1e-6 100e-6
23 .param b1_isupplycmb1 = 1e-6 indexed linked 1e-6 100e-6
24 .param b1_areal = 100e6 indexed linked maximize 1 1e-6 100e6
25 .param b1_areacmb1 = 100e6 indexed linked 1e-6 100e6
26 .param b1_cin1 = 10e-15 indexed linked 5e-15 10e-12
27 .param b1_n1 = 1 integer var 1 10
28 .param b1_n3 = 1 integer var 1 10
29 .param b1_n4 = 1 integer var 1 10
30 .param b2_n1 = 1 integer var 1 10
31 .param b2_n3 = 1 integer var 1 10
32 .param b2_n4 = 1 integer var 1 10
33 .param s1_n1 = 1 integer var 1 10
34 .param TNOISEdbm_par = 10e-6 float nvar 1e-6 100e-6
35 .param TNOISEdb_par = 10e-6 float nvar 1e-6 100e-6
36 .param TNOISErms_par = 10e-6 float nvar 1e-6 100e-6
37 .param topoorder = 0 integer var 0 5
38 .param Powercons_par = 10e-6 float nvar 1e-6 100e-6
39 .param AREA_par = 10e-6 float nvar 1e-6 100e-6
40 .close

```