

UNIVERSIDADE FEDERAL DO PAMPA

Anderson de Paula Fortes

**PROJETO AUTOMÁTICO DE CIRCUITOS INTEGRADOS ANALÓGICOS
UTILIZANDO METAHEURÍSTICAS BIO-INSPIRADAS**

**Alegrete
24/08/2018**

Anderson de Paula Fortes

**PROJETO AUTOMÁTICO DE CIRCUITOS INTEGRADOS ANALÓGICOS
UTILIZANDO METAHEURÍSTICAS BIO-INSPIRADAS**

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

Orientador: Alessandro Girardi

Alegrete
24/08/2018

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

F738p Fortes, Anderson de Paula

Projeto automático de circuitos integrados analógicos
utilizando metaheurísticas bio-inspiradas / Anderson de Paula
Fortes.

99 p.

Dissertação(Mestrado)-- Universidade Federal do Pampa,
MESTRADO EM ENGENHARIA ELÉTRICA, 2018.

"Orientação: Alessandro Gonçalves Girardi".

1. Circuitos Analógicos. 2. Otimização. 3. Amplificadores.
4. OTA Miller. 5. Telescopic. I. Título.

Anderson de Paula Fortes

**PROJETO AUTOMÁTICO DE CIRCUITOS INTEGRADOS
ANALÓGICOS UTILIZANDO METAHEURÍSTICAS
BIO-INSPIRADAS**

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

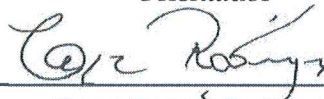
Área de concentração: Sistemas de Energia

Dissertação defendida e aprovada em: Alegrete, 24 de Agosto de 2018.

Banca examinadora:



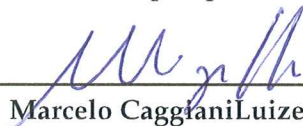
Alessandro Girardi
Orientador



Cesar Ramos Rodrigues
UFSM



Jumar Russi
Unipampa



Marcelo Caggiani Luizelli
Unipampa

Dedico este trabalho à memória da minha querida professora Márcia e sua filha Aurora.

AGRADECIMENTOS

Agradeço a Deus por me privilegiar, me abençoar com todas as coisas necessárias para que eu possa ir atrás dos meus objetivos e por estar comigo, tanto nas horas boas quanto nas mais difíceis.

Aos meus pais, Francisco e Iara, por sempre fazerem tudo que estivesse ao alcance, abrindo mão de muitas coisas para que eu e minhas irmãs pudéssemos estudar. Pelos bons exemplos de luta e por me incentivarem a me tornar alguém digno.

Às minhas irmãs, Margiani, Luciani e Carolini, pela amizade e companheirismo durante a vida toda.

Aos meus mestres, professores do ensino fundamental à pós-graduação, pelo ensinamento que me deram, presente mais valioso e que nunca alguém poderá me tirar. Em especial à memória da prof. Márcia Cera, exemplo de grande profissional, amiga e fonte de inspiração como docente.

Ao meu orientador, prof. Alessandro Girardi, pela oportunidade dada a mim de realizar este trabalho ao seu lado. Agradeço pela paciência, por sempre ajudar, incentivar, mostrar o caminho correto a ser seguido durante o desenvolvimento deste trabalho e pelo exemplo de grande profissional.

Aos meus colegas e amigos do grupo GAMA e demais amigos que fiz na universidade pelas parcerias nas madrugadas de estudo, ajudas nos trabalhos, churrascos e cervejas. Levarei a amizade de vocês para a vida toda. Agradeço de coração a todas as pessoas que trabalham na universidade, por nos darem segurança, ambiente limpo e agradável, sem os quais o ensino não seria possível.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, portanto, deixo aqui meus agradecimentos também à CAPES.

*“Vinde a mim todos que estais cansados e sobrecarregados,
e eu vos aliviarei.
Tomai sobre vós o meu jugo
e aprendei de mim,
porque sou manso e humilde de coração;
e achareis descanso para vossa alma.
Porque o meu jugo é suave,
E meu fardo é leve .”
(Jesus de Nazaré)*

RESUMO

A contínua redução das dimensões e tensões de alimentação dos circuitos integrados torna os processos de otimização e fabricação dos circuitos mais complexos. A parte analógica de um circuito integrado misto representa uma grande parcela no esforço de dimensionamento do circuito. É necessário dimensionar cada dispositivo do circuito separadamente de forma a atingir as especificações desejadas para o circuito. Cada componente de um circuito analógico, seja ele um transistor, resistor ou capacitor, representa uma ou mais variáveis, como, por exemplo, o comprimento e largura do canal de um transistor. Em circuitos com muitas variáveis, a combinação de dimensões dos diferentes componentes é muito grande, o que torna o processo de dimensionamento e ajuste dos valores muito complexo. O projetista precisa dimensionar cada um dos componentes e verificar se o circuito atende ao desempenho desejado. Esse processo de dimensionamento de circuito integrado analógico pode ser modelado como um problema de otimização e resolvido por heurísticas de otimização. No entanto, a capacidade das heurísticas para explorar o espaço de busca tem uma grande influência no resultado do dimensionamento do circuito. Neste trabalho, implementou-se duas metaheurísticas bio-inspiradas para dimensionar amplificadores operacionais em tecnologia CMOS: Cuckoo Search (CS) e Firefly. Os resultados são comparados com o dimensionamento utilizando a heurística Particle Swarm Optimization (PSO). Ademais, realizou-se um estudo acerca do critério de parada para os algoritmos propostos e também sobre os valores iniciais para as variáveis livres. Os resultados mostram que PSO e CS são mais adequados para alcançar o desempenho requerido que atenda às especificações do circuito, enquanto o FA apresenta dificuldade em pesquisar eficientemente o espaço de projeto. Além disso, os resultados mostram que heurísticas são apropriadas para o dimensionamento automático de circuitos analógicos e podem levar a resultados melhores do que no processo manual.

Palavras-chave: Circuitos Analógicos, Otimização, Ota Miller, Telescopic, Amplificadores, Cuckoo Search, Firefly Algorithm, Particle Swarm Optimization.

ABSTRACT

The continuous reduction of the dimensions and supply voltages of the integrated circuits makes the optimization and manufacturing processes of the circuits more complex. The analog part of a mixed integrated circuit represents a large part of the circuit sizing effort. It is necessary to dimension each device of the circuit separately in order to reach the specifications desired for the circuit. Each component of an analog circuit, whether it is a transistor, resistor, capacitor or etc., represents one or more variables, such as the length and width of the channel of a transistor. In circuits with many variables, the combination of dimensions of the different components is very large, which makes the process of sizing and adjusting the values very complex. The designer needs to scale each of the components and check if the circuit meets the desired performance aspects. This analog integrated circuit sizing process can be modeled as an optimization problem and solved by optimization heuristics. However, the ability of heuristics to explore the search space has a great influence on the result of the circuit sizing. In this work, two bio-inspired metaheuristics were implemented to size three operational amplifiers in CMOS technology: Cuckoo Search (CS) and Firefly and compared with the results of sizing with Particle Swarm Optimization (PSO). In addition, a study was carried out on the stopping criterion for the proposed algorithms and also on the initial values for the free variables. The results show that PSO and CS are more adequate to achieve the required performance that meets the specifications of the circuit, while the FA presents difficulty in efficiently searching the design space. In addition, the results show that heuristic algorithms are suitable for the automatic sizing of analog circuits and can lead to better results than in the manual process.

Keywords: Analog Circuits, Optimization, Miller OTA, Telescopic, Amplifiers, Cuckoo Search, Firefly Algorithm, Particle Swarm Optimization.

LISTA DE ILUSTRAÇÕES

Figura 1 – Níveis de projeto de um circuito analógico.	28
Figura 2 – Estrutura física de um transistor em tecnologia CMOS.	29
Figura 3 – Fluxo básico de um modelo baseado em otimização.	31
Figura 4 – Fluxo de projeto da ferramenta UCAF.	39
Figura 5 – Fluxograma de execução do núcleo da ferramenta.	40
Figura 6 – Esquemático de um OTA Miller em tecnologia CMOS	41
Figura 7 – Netlist representando o OTA Miller da figura 6.	42
Figura 8 – Exemplo de arquivo de parâmetro gerado pela ferramenta UCAF para o circuito da figura 6.	43
Figura 9 – Configuração de circuito para medição de características AC.	44
Figura 10 – Netlist gerado pela ferramenta representando o circuito da figura 9. .	45
Figura 11 – Configuração de circuito para medição do slew rate.	45
Figura 12 – Resposta de saída para medição de taxa de variação.	46
Figura 13 – $V_{in} \times V_{out}$ para a medição do ICMR.	46
Figura 14 – Configuração de circuito para medição do output swing.	47
Figura 15 – Medição $V_{in} \times V_{out}$ para extração do output swing.	47
Figura 16 – Métricas de avaliação das restrições de projeto: (a) especificação de valor mínimo exigido e (b) especificação de valor máximo exigido. .	50
Figura 17 – Gráfico que mostra o impacto das especificações não atendidas na função custo e sua evolução.	51
Figura 18 – Algumas das heurísticas propostas nas últimas décadas.	55
Figura 19 – (a): Movimento browniano. (b): Vôo de Lévy.	58
Figura 20 – Ilustração do funcionamento do processo de atratividade dos vagalu- mes.	61
Figura 21 – Evolução da função custo para o FA, PSO e CS utilizando $h=30$	68
Figura 22 – Representação da mesma execução da figura 21 mas com $h=20$	69
Figura 23 – Distribuição das partículas iniciais para W1	71
Figura 24 – Distância das posições iniciais das partículas para o melhor valor final de W1.	72
Figura 25 – Distância das posições iniciais das partículas para o pior valor final de W1.	73
Figura 26 – Esquemático de um OTA Miller em tecnologia CMOS	75
Figura 27 – Esquemático do OTA de 0.25V proposto por Ferreira e Sonkusale (2014)	84
Figura 28 – Uma matriz retangular de transistores nMOS com implantes de halo (à esquerda) e seu transistor equivalente (à direita)	85
Figura 29 – Netlist representando o sub-circuito M1a.	85
Figura 30 – Netlist que faz a chamada dos sub-circuitos de transistores.	86
Figura 31 – Esquemático de um amp-op telescopic em tecnologia CMOS	89

LISTA DE TABELAS

Tabela 1 – Trabalhos sobre otimização automática de CIs analógicos	33
Tabela 2 – Resultados para as 3 diferentes formas de inicialização das variáveis livres.	70
Tabela 3 – Valor médio da função custo variando o número de ninhos.	77
Tabela 4 – Valor médio da função custo variando o parâmetro pa	77
Tabela 5 – Resultados variando α	78
Tabela 6 – Resultados variando β	79
Tabela 7 – Valor médio da função custo variando o número de vagalumes	79
Tabela 8 – Número de iterações e valor da função custo obtida.	80
Tabela 9 – Especificações para os melhores resultados de cada algoritmo de otimização avaliado.	80
Tabela 10 – OTA Miller: Análise típica e análises de Corners	82
Tabela 11 – Indicadores de desempenho obtidos para o OTA Miller alimentado pelo substrato.	87
Tabela 12 – Valores das variáveis após a otimização.	88
Tabela 13 – <i>Bulk-driven</i> OTA Miller: Análise típica e análises de Corners	88
Tabela 14 – Especificações para os melhores resultados de cada algoritmo de otimização avaliado.	90
Tabela 15 – Valores das variáveis após a otimização.	90
Tabela 16 – Telescopic: Análise típica e análises de Corners	90

SUMÁRIO

1	INTRODUÇÃO	21
1.1	Objetivo	24
1.2	Organização do Trabalho	25
2	PROJETO DE CIRCUITOS ELETRÔNICOS ANALÓGICOS	27
2.1	Projeto Automático de ICs Analógicos	29
2.2	Trabalhos Relacionados	31
2.2.1	Métodos baseados no conhecimento	32
2.2.2	Métodos baseados em otimização	32
2.2.3	Conclusão	37
3	A FERRAMENTA UCAF	39
3.1	Núcleo Principal	40
3.2	Especificações e Testbenches	42
3.3	Bloco de Otimização	47
3.4	Função Custo	48
3.5	Conclusão	51
4	HEURÍSTICAS DE OTIMIZAÇÃO	53
4.1	Algoritmos Bio-Inspirados	54
4.2	Cuckoo Search via vôos de Lévy	56
4.2.1	Vôos de Lévy	56
4.2.2	Funcionamento do Algoritmo Cuckoo Search	58
4.3	Algoritmo FireFly	60
4.3.1	Atratividade	61
4.3.2	Distância e movimento	62
4.4	Particle Swarm Optimization	63
4.5	Critério de Parada	65
4.6	Estudo sobre os valores iniciais	68
4.7	Conclusão	72
5	RESULTADOS	75
5.1	Amplificador Operacional de Transcondutância do tipo Miller	75
5.1.1	Calibragem dos Parâmetros no CS	76
5.1.2	calibragem dos Parâmetros no FA	78
5.1.3	Comparação entre os algoritmos	79
5.1.4	Análise de Corners	81
5.2	OTA Miller de Baixa Tensão Alimentado pelo Substrato	83
5.3	Amplificador Operacional Telescópico	87

5.4	Conclusão	89
6	CONCLUSÃO	91
	REFERÊNCIAS	93

1 INTRODUÇÃO

A demanda por sistemas integrados de baixo custo, baixa potência e baixa tensão tem aumentado em todas as áreas de engenharia. Além disso, existe uma constante redução na dimensão dos componentes elétricos enquanto busca-se melhorar seus desempenhos.

O desenvolvimento da microeletrônica nas últimas décadas possibilitou que circuitos eletrônicos sejam cada vez mais integrados. Esses circuitos, atualmente, são construídos em escalas micrométricas ou nanométricas sobre elementos semicondutores como o silício.

A contínua redução das dimensões e tensões de alimentação dos circuitos integrados (CI) torna a otimização e fabricação dos circuitos mais complexos, lentos e mais dependentes dos processos de fabricação e variações ambientais, tais como temperatura, radiação, etc (MORETO et al., 2012). Além disso, existe também uma forte tendência em desenvolver equipamentos em um único chip que reúna tanto os circuitos digitais quanto os analógicos. Isto tem gerado uma grande necessidade de projetistas desses CIs.

Embora a maioria das funções dos sistemas integrados sejam implementadas para o processamento de sinais digitais, os circuitos analógicos são essenciais em muitas aplicações. Existem algumas funções que necessariamente serão analógicas, como por exemplo a entrada de um circuito proveniente de um sensor, microfone, antena, rede, entre outros. Esses sinais devem ser recebidos ou detectados e, em seguida, amplificados e filtrados até um nível que permita a digitalização desse sinal.

A parte analógica, mesmo representando uma pequena fração de um circuito eletrônico (cerca de 20%), é a parte mais difícil de projetar devido à natureza complexa e a alta necessidade de conhecimento de circuitos analógicos por parte do projetista (VURAL, 2012). Esta área, mesmo sendo pequena, representa cerca de 40% de todo o esforço de projeto. Isso se deve ao fato de que o projeto geralmente é realizado de forma manual, sem o uso de ferramentas de automação.

Atualmente existem diversas ferramentas de projeto assistidas por computador (CAD) para auxiliar o fluxo de projeto dos CIs digitais. Por outro lado, poucas ferramentas CAD profissionais são dedicadas a auxiliar no projeto de CIs analógicos, como as ferramentas proprietárias fornecidas pelas empresas CADENCE[®], Synopsys[®] ou Mentor Graphics[®].

No dimensionamento dos transistores, por exemplo, os circuitos analógicos são mais complexos se comparados aos circuitos digitais. Nos circuitos digitais todos os transistores possuem tamanhos pré-definidos, ao contrário dos analógicos, em que cada transistor deve ser dimensionado individualmente (SEVERO, 2012).

O projeto de um CI analógico fundamentalmente depende do conhecimento e da experiência do projetista, visto que eles são sistemas que possuem muitas variáveis

livres, isto é, possuem muitos parâmetros que podem ser modificados, como dimensões de cada componente do CI (comprimento e largura do gate dos transistores, resistores, capacitores e etc).

Em um circuito analógico simples, com 6 transistores, por exemplo, um capacitor e uma fonte de corrente, existem variáveis representando o comprimento e largura do gate dos 6 transistores, uma representando a capacitância e uma que define a corrente, totalizando 14 variáveis livres. Existe uma grande combinação de valores para as variáveis livres que devem ser investigadas a fim de atender às especificações de projeto. Os projetistas precisam atribuir um valor a cada variável, ou seja, dimensionar cada componente integrado, sejam eles transistores, resistores, capacitores e até mesmo indutores (BARROS; GUILHERME; HORTA, 2007; ALPAYDIN; BALKIR; DUNDAR, 2003; GIELEN; RUTENBAR, 2000). Alguns exemplos de especificações de projeto são o ganho de tensão, a margem de fase, o consumo de potência, a velocidade de resposta, área do gate, entre outros.

O projeto de CIs analógicos é geralmente constituído de três principais passos. Primeiro é definida uma dentre as várias arquiteturas e topologias possíveis para o circuito. O segundo passo é determinar os valores de parâmetros do circuito (resistência e valores de capacitores, geometrias de dispositivos, tais como comprimento e largura de transistores MOS) (JAFARI et al., 2010; TAHERZADEH-SANI et al., 2003). O terceiro passo consiste no desenho do leiaute representando o CI fisicamente.

O processo de dimensionamento pode ser realizado através de duas abordagens diferentes: baseado no conhecimento ou baseado em otimização. A idéia básica do projeto baseado no conhecimento é formular equações de projeto de tal forma que, dadas as características de desempenho desejadas para o CI, os parâmetros de projeto possam ser calculados.

Na síntese baseada no conhecimento utiliza-se uma metodologia de projeto que geralmente emprega equações de primeira ordem, com o objetivo de definir inicialmente as dimensões dos transistores e atender a alguns poucos objetivos de projeto, tais como o ganho de tensão e a frequência de ganho de tensão unitário (GIELEN; RUTENBAR, 2000). A partir disso, o projetista faz uso de simuladores do tipo SPICE (*Simulated Program with Integrated Circuits Emphasis*) para fazer pequenos ajustes nas dimensões e parâmetros dos transistores a fim de alcançar as especificações do projeto.

Esse procedimento de ajuste dos parâmetros é realizado geralmente através da mudança de um único parâmetro ou mesmo apenas uma das dimensões de um dos transistores e, a partir disso, o projetista avalia os novos resultados obtidos verificando se um determinado objetivo de projeto foi alcançado e também não foram degradados os demais objetivos (MORETO et al., 2012). Esse processo iterativo e repetitivo entre o projetista e o simulador é muito trabalhoso, lento e por isso é totalmente dependente da experiência do projetista. Outras desvantagens são o grande tempo de preparo

necessário para desenvolver equações, a dificuldade de usá-las em uma tecnologia diferente e a limitação a um conjunto de circuitos.

Muitos circuitos a serem dimensionados apresentam um grande número de variáveis livres a serem determinadas. Nesses casos, métodos baseados em equações de primeira ordem e processo manual interativo usando o simulador SPICE e busca aleatória, por exemplo, geralmente são ineficientes quando muitas especificações de projeto são consideradas, devido ao espaço de busca ser extremamente grande. Em um circuito com 24 variáveis livres, por exemplo, o número de combinações possíveis no espaço de projeto pode ultrapassar a ordem de 10^{40} , o que é muito grande para avaliar com uma técnica exaustiva. Nesse caso, o tempo e o custo do projeto do circuito se tornam altos, além de ser bastante difícil se atender a todas as especificações desejadas.

Além disso, alguns amplificadores operacionais de baixa tensão foram propostos nos últimos anos utilizando matrizes de transistores pequenos ao invés de um único transistor grande para reduzir os chamados efeitos de halo, que são causados por uma alteração na dopagem do material de fabricação dos transistores com o intuito de reduzir os efeitos de canal curto dos transistores (FERREIRA; SONKUSALE, 2014; ZHAO et al., 2015; GRASSO et al., 2015). Isso pode levar o projeto a ter mais variáveis livres, como o número de linhas e colunas nessas matrizes de transistores.

Nos casos de circuitos mais complexos, uma possibilidade é o uso de algoritmos baseados em alguma heurística para a escolha dos valores para as variáveis. Esses algoritmos podem processar muitas especificações de projeto simultaneamente (DEY et al., 2014). Dessa forma, o uso de algoritmos heurísticos são considerados uma boa solução para projeto dos CIs analógicos.

Heurísticas são utilizadas para resolver problemas de grande tamanho com muitos critérios a serem considerados (TALBI—ANTONIO; ALBA, 2006). Elas podem ser adaptadas para atender aos requisitos específicos do problema. Mesmo que não garantam encontrar a solução ótima, elas fornecem boa capacidade de encontrar uma solução aceitável (CHAN; TIWARI, 2007).

Na síntese baseada em otimização, o dimensionamento dos componentes é transformado em um problema de minimização de função que pode ser calculado através de métodos numéricos ou por simulação. Eles são baseados na introdução de um avaliador de desempenho dentro de um laço de otimização iterativa. A cada modificação dos valores das variáveis livres, a avaliação é realizada.

Sistemas de avaliação baseados em métodos numéricos ou equações muitas vezes consomem muito mais tempo de preparo, além de que as simplificações requeridas nas equações causam baixa precisão e incompletude. Os métodos baseados em simulação não se baseiam em equações analíticas, mas em simulações do tipo SPICE para avaliar os desempenhos do circuito no processo de otimização, o que resulta em uma maior precisão.

Algumas técnicas de otimização são mais diretas, preocupadas em achar apenas uma solução, caindo com frequência em ótimos locais, como os algoritmos gulosos. Outras se destinam a pesquisar com mais cuidado o espaço de soluções, na tentativa de obter resultados mais próximos do ótimo. Algumas heurísticas apresentadas na literatura são Pesquisa Local (AARTS; LENSTRA, 2003), Simulated Annealing (SA) (SIARRY et al., 1997), Tabu Search (TS) (GLOVER, 1989), etc.

A maioria dos problemas de otimização de projeto de circuito exigem simultaneamente diferentes tipos de variáveis, funções objetivo e de restrição na sua formulação. Os procedimentos de otimização acima mencionados requerem um longo tempo de computação quando o problema apresenta um grande espaço de busca e um grande número de objetivos a serem alcançados. Nos últimos anos, foram propostos novos algoritmos de otimização heurística, muitos deles inspirados na natureza. A ideia por trás desses algoritmos é se inspirar no comportamento coletivo dos sistemas descentralizados e auto-organizados (VURAL, 2012). A técnica de se inspirar em comportamento coletivo das espécies é conhecida como Swarm Intelligence (SI) (BONABEAU; DORIGO; THERAULAZ, 1999). Em muitos trabalhos eles demonstraram alcançar melhores resultados comparados às estratégias que utilizam um único agente na exploração do espaço de busca. Algumas heurísticas bio-inspiradas ainda não foram amplamente utilizadas na otimização de CIs analógicos.

1.1 OBJETIVO

Este trabalho tem por objetivo utilizar heurísticas bio-inspiradas na otimização de amplificadores operacionais (amp-ops) de forma automática. Para isso, são utilizadas três heurísticas: Particle Swarm Optimization (PSO), Firefly Algorithm (FA) e Cuckoo Search (CS). Estas heurísticas são utilizadas pelo fato de apresentarem bons resultados em otimizações de uma série de problemas e não serem, até então, amplamente utilizadas no processo de otimização de CIs analógicos. A ideia é analisar o comportamento dessas heurísticas no projeto de CIs analógicos. Além disso, outro objetivo do trabalho é propor uma estratégia que permita utilizar como variáveis livres as dimensões das matrizes de transistores utilizadas em alguns amplificadores de baixa tensão, desenvolvendo um algoritmo capaz de gerar sub-circuitos de forma automática em cada iteração da heurística de acordo com o número de transistores em série e em paralelo de cada matriz. Técnicas automáticas de design de amp-ops que considerem essas dimensões como variáveis livres foram pouco exploradas na literatura.

Para que os objetivos sejam atendidos, é necessário, a partir da determinação das especificações a serem atendidas pelo circuito, explorar o espaço de projeto a fim de encontrar uma solução que atenda as restrições a fim de gerar um circuito otimizado, isto é, de tamanho reduzido, com baixa dissipação de potência e que satisfaça as demais especificações desejadas.

São comparadas três heurísticas, PSO, FA e CS, no projeto de um OTA (Operational Transconductance Amplifier) Miller. Em seguida, a heurística CS é utilizada para dimensionar outros dois amplificadores, um do tipo telescopic e um de baixa tensão alimentado pelo substrato com capacitor Miller.

1.2 ORGANIZAÇÃO DO TRABALHO

A sequência desse trabalho está organizada como segue. No segundo capítulo são apresentadas as etapas necessárias para a concepção de um CI analógico funcional, focando nas características do processo de dimensionamento e apresenta também os modelos elétricos dos CIs dimensionados. Nesse capítulo também é realizada uma revisão da literatura acerca de métodos para o design de CIs analógicos. No capítulo 3 é demonstrado o fluxo básico da ferramenta utilizada, como é realizada a otimização, principais funções, *testbenches* para determinar os valores das especificações e como é realizado o cálculo da função custo. No capítulo 4 são apresentadas as heurísticas utilizadas para otimizar os circuitos, o critério de parada utilizado e um estudo sobre os valores iniciais para as variáveis livres a serem trabalhadas nas heurísticas. O capítulo 5 apresenta os circuitos dimensionados neste trabalho e os resultados obtidos para cada um deles, como valor das especificações. Por fim, no capítulo 6 é apresentada a conclusão sobre o trabalho desenvolvido.

2 PROJETO DE CIRCUITOS INTEGRADOS ANALÓGICOS

Neste capítulo é apresentado o fluxo necessário para a concepção de um CI analógico, desde o início do projeto até a fase de prototipação. São apresentadas as sub-divisões do projeto, isto é, todas as fases que devem ser realizadas para a criação de um CI funcional. Também é descrito como o projeto de um CI pode ser transformado em um projeto automático utilizando otimização. Por fim, é realizada uma revisão bibliográfica acerca de trabalhos cujo foco é o dimensionamento de CIs analógicos de forma automática.

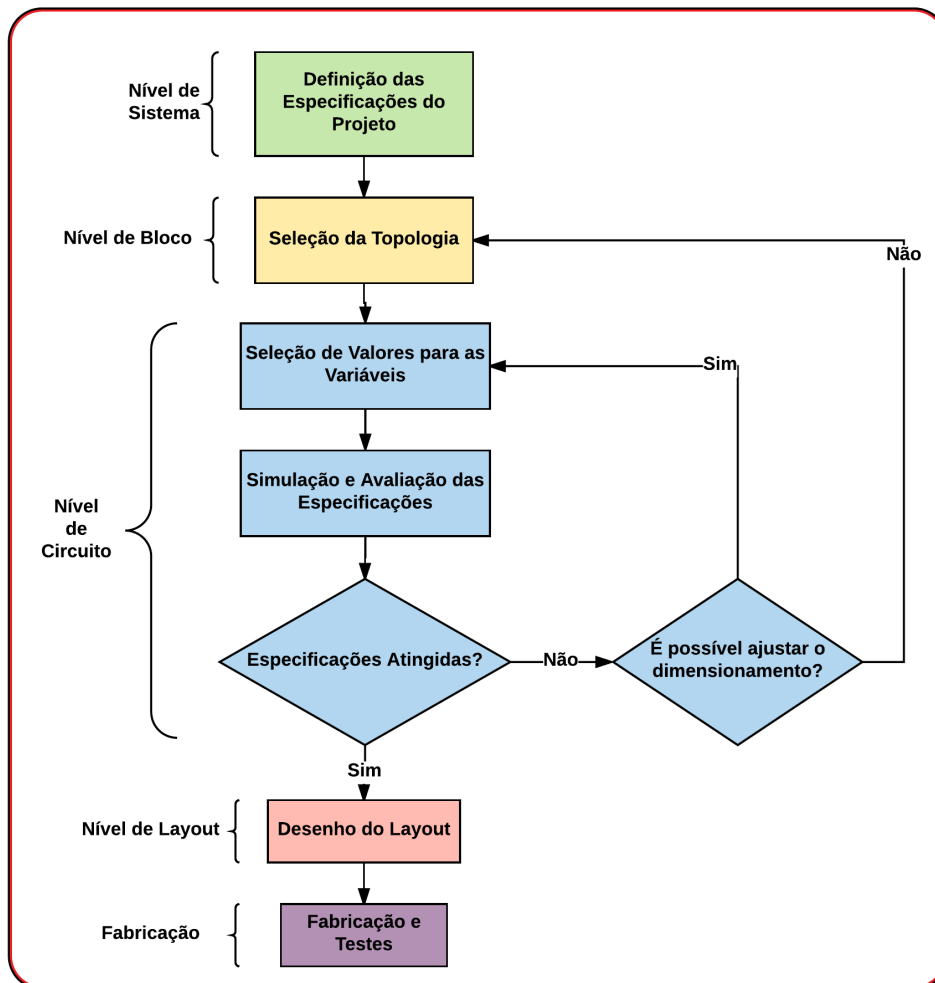
De acordo com Barros, Guilherme e Horta (2007), para assegurar o funcionamento dos circuitos integrados analógicos, deve-se seguir cinco níveis hierárquicos de projeto:

1. Nível de sistema: Neste estágio de desenvolvimento, as especificações requeridas pelo circuito e a tecnologia do processo de fabricação são definidos;
2. Nível de bloco: Neste estágio, descrições em alto nível dos blocos básicos são efetivamente traduzidos para uma arquitetura de blocos funcionais requeridos para realizar o comportamento especificado pelo sistema;
3. Nível de circuito: Nesta fase realiza-se um processo de otimização para cada bloco básico analógico, onde a otimização é um processo iterativo para determinar as dimensões físicas em nível de dispositivo, por exemplo, as dimensões dos transistores. A tolerância do projeto deverá ser obtida levando em conta as variações ambientais e do processo de fabricação para garantir que o CI atenderá às especificações após o processo de fabricação. Para isso pode-se utilizar análise de Monte Carlo ou de Corners (pior caso). As especificações elétricas e de desempenho requeridas do circuito final são verificadas usando um simulador SPICE;
4. Nível de leiaute: Nesse estágio, os blocos básicos otimizados no passo anterior são representados de forma física. Leiaute é um conjunto de formas geométricas que devem obedecer às regras de projeto especificadas pelo processo de fabricação;
5. Fabricação e teste: Nesse estágio, o projeto do leiaute produzido na etapa anterior é enviado para fabricação e o CI é finalmente produzido. O teste e validação são passos fundamentais para assegurar a operação correta do circuito integrado e, assim, um sistema de caracterização elétrica adequado deve ser definido.

Na figura 1 pode ser observada uma abstração dos níveis hierárquicos de projeto proposto por Barros, Guilherme e Horta (2007).

No nível de sistema uma topologia de circuito é escolhida de acordo com as especificações desejadas. A topologia é escolhida com base no conhecimento do projetista. A partir da escolha da topologia, é realizada a definição da arquitetura dos

Figura 1 – Níveis de projeto de um circuito analógico.



Adaptado de Barros, Guilherme e Horta (2007)

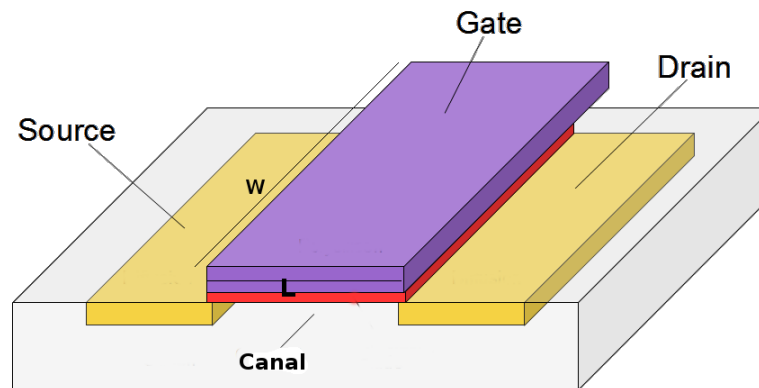
blocos que irão compor o circuito. Já no nível de circuito, os transistores do circuito devem ser dimensionados. Neste dimensionamento determina-se o comprimento e largura do gate do transistor, como mostrado na figura 2.

Para um primeiro dimensionamento, normalmente o projetista faz uso de equações aproximadas, como equação da corrente I_D do modelo quadrático para transistores CMOS (HOLBERG; ALLEN, 2002; SEVERO; GIRARDI, 2011).

Após o dimensionamento inicial, o circuito deve ser simulado e os resultados comparados com as especificações desejadas. Isto é realizado através de simulações do tipo SPICE. Caso as especificações desejadas não sejam atingidas, deve-se refazer o processo de dimensionamento. Se não for possível atingir as especificações redimensionando o circuito, deve-se escolher outra topologia de circuito. Se as especificações forem atingidas, então pode-se progredir ao nível de leiaute.

Na fase de leiaute o projetista deve adotar estratégias que permitam desenhar o CI de forma que ele tenha baixa variabilidade de parâmetros após a fabricação, ou

Figura 2 – Estrutura física de um transistor em tecnologia CMOS.



Adaptado de (LEE et al., 2009)

seja, se comporte como o esperado (SEVERO; GIRARDI, 2011). Após a fase de leiaute, o circuito deve novamente ser simulado, uma vez que agora ele possui elementos como capacitâncias e resistências.

O projeto de um CI analógico possui uma série de etapas. Em todas essas etapas é necessário tomar decisões e validar o CI. Normalmente todas as decisões são tomadas de acordo com a experiência do projetista. No dimensionamento dos componentes, considerada a parte mais crítica, geralmente o projetista realiza a mudança de apenas um parâmetro ou apenas uma das dimensões de um dos transistores do circuito por vez.

Encontrar um ponto de polarização dos circuitos que atinja as especificações do projeto requer muitas iterações de redimensionamento e simulações. Cada vez que um componente é redimensionado, deve-se realizar a simulação elétrica a fim de verificar se as especificações de projeto foram atingidas e também se não foi degradado nenhum dos demais objetivos do projeto (MORETO et al., 2012).

No dimensionamento dos transistores, o uso de equações simplificadas na maioria das vezes leva a resultados que não retratam o real funcionamento do circuito. Outro problema na etapa de dimensionamento do circuito, é que muitas vezes redução de área e potência consumida acabam sendo considerados objetivos secundários. O objetivo principal é produzir um circuito que atinja as especificações do projeto.

Para que o circuito dimensionado por meio de simulação fique o mais próximo possível de um circuito fabricado, deve-se considerar no dimensionamento as variações do processo de fabricação.

2.1 PROJETO AUTOMÁTICO DE ICs ANALÓGICOS

O processo de dimensionamento do circuito baseado apenas na experiência e conhecimento do projetista possui algumas desvantagens, pois é necessário que para

cada diferente topologia seja criado um novo plano de projeto (SEVERO; GIRARDI, 2011).

Outra possibilidade para dimensionar um CI analógico é transformar o dimensionamento em um problema de otimização combinatória. Nessa estratégia algum método de otimização é utilizado para explorar o espaço de projeto, isto é, a variação da dimensões dos componentes é realizada de forma automática de acordo com os valores obtidos para as especificações a cada iteração. Os métodos baseados em otimização podem ser baseados em equações ou em simulação elétrica (BARROS; GUILHERME; HORTA, 2007).

Os modelos baseados em equações tendem a ser mais rápidos, uma vez que não é necessária uma simulação elétrica completa, no entanto, as simplificações necessárias nas equações podem levar a uma menor precisão. Além disso, não é possível obter todas as especificações.

Os métodos baseados em simulação elétrica utilizam simuladores elétricos do tipo SPICE para avaliação do circuito. Para essa simulação, são utilizados modelos elétricos fornecidos pelas empresas de fabricação dos CIs. Apesar desse modelo consumir um maior tempo computacional, uma vez que é necessário realizar diferentes simulações para medir as várias especificações, este método é mais preciso, pois é baseado em modelos complexos para a modelagem dos dispositivos.

A utilização de simulação elétrica do tipo SPICE permite que sejam analisadas de forma confiável qualquer especificação do circuito. Isto dá ao modelo de otimização uma maior flexibilidade. Além disso, este modelo de projeto permite realizar a análise da produtividade do circuito através da análise de Monte Carlo, por exemplo (SEVERO, 2012). A análise da produtividade pode ser considerada já no processo de otimização, e não realizada apenas no final do dimensionamento (DOMANSKI, 2016).

Uma etapa também muito importante no processo de dimensionamento automático de um CI é a escolha do método de otimização. Existe um grande número de métodos que podem ser utilizados.

A escolha do método de otimização deve ser estudada a fim de que o modelo escolhido consiga atingir uma solução que atenda as especificações desejadas e permita considerar outros objetivos como redução da área e da potência. Existem diferentes tipos de métodos de otimização, como métodos exatos, métodos heurísticos entre outros.

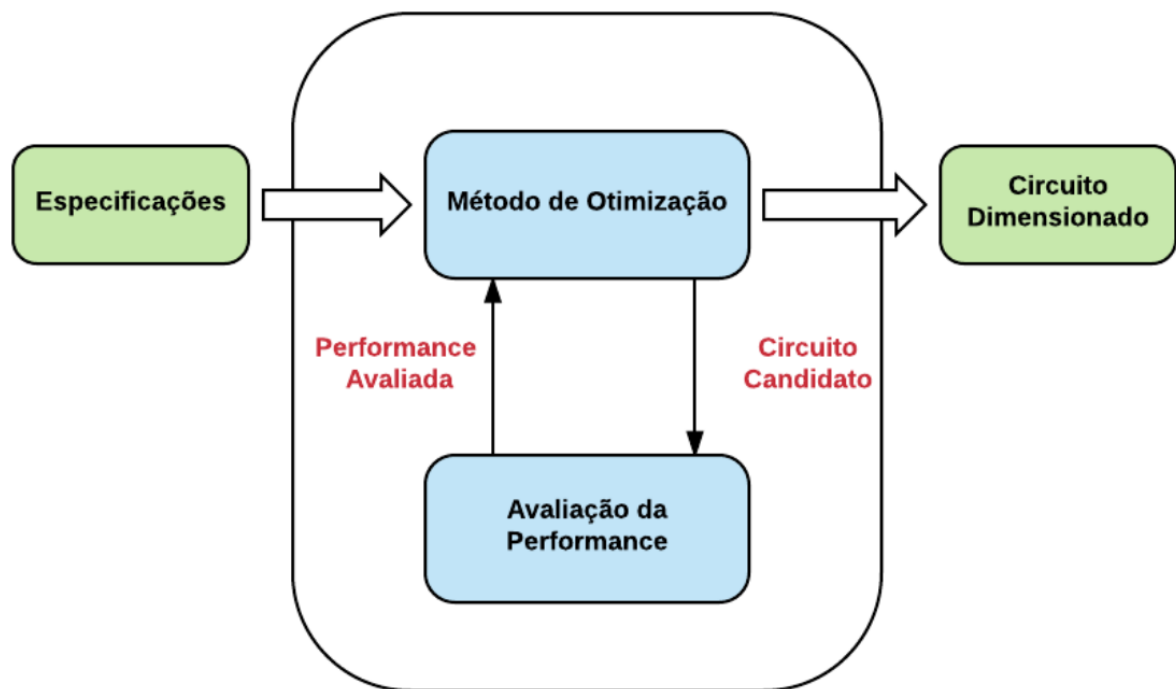
Os métodos de otimização heurísticos nem sempre garantem a solução ótima, pois são métodos estocásticos. No entanto, quando o espaço de busca é não linear e muito grande, eles são uma boa estratégia para a busca de uma solução (GIUNCHIGLIA et al., 2001).

Uma vantagem dos métodos heurísticos é que muitos deles não dependem da solução inicial, dessa forma, não é necessário um plano de projeto para determinar um dimensionamento inicial do circuito. No capítulo 4 são apresentados diferentes métodos

de otimização inspirados em inteligência das espécies.

Para o dimensionamento automático de um circuito analógico utiliza-se geralmente ferramentas que recebem como entrada informações a respeito do circuito e das especificações desejadas e então é realizada a otimização através de um ciclo de avaliação e mudança de parâmetros. O fluxo de uma ferramenta iterativa de otimização com a avaliação de cada solução pode ser resumido como o mostrado na figura 3.

Figura 3 – Fluxo básico de um modelo baseado em otimização.



Fonte: Autor.

2.2 TRABALHOS RELACIONADOS

Diferentes trabalhos foram realizados com o intuito de propor técnicas que auxiliem no projeto e na redução do tempo de dimensionamento dos CIs analógicos. Alguns trabalhos buscam utilizar algoritmos baseados em heurísticas de otimização para a definição dos parâmetros do circuito, outros trabalhos tem como foco propor sistemas de representação das especificações do circuito como equações para tornar o processo de otimização mais rápido. Diversos trabalhos são encontrados na literatura a respeito dessas linhas. Neste capítulo é realizada uma revisão bibliográfica sobre alguns trabalhos que propõem a aplicação de métodos heurísticos da área de inteligência artificial (IA) na otimização de circuitos integrados analógicos.

Vários sistemas de projeto foram propostos para automatizar o projeto de blocos analógicos. Os trabalhos encontrados podem ser classificados em duas categorias: baseados no conhecimento e baseados em otimização.

2.2.1 MÉTODOS BASEADOS NO CONHECIMENTO

As primeiras estratégias de automação, propostas há cerca de 30 anos, tentaram sistematizar o projeto usando um plano derivado do conhecimento (LOURENÇO; HORTA, 2012).

A ideia básica da abordagem baseada no conhecimento é traduzir a experiência do projetista em regras de síntese que utilizam equações simplificadas. O objetivo principal é representar as especificações do circuito através de equações que permitam calcular as dimensões dos componentes de forma a atender aos requisitos.

Alguns dos sistemas relatados na literatura que seguem esta abordagem podem ser vistos em Degrauwe et al. (1987), Onodera, Kanbara e Tamaru (1990) e El-Turky e Perry (1989).

A principal vantagem dessa abordagem é o curto tempo de execução. Por outro lado, criar um plano de projeto é difícil e demorado, além de que é necessário realizar frequentes mudanças nas equações para mantê-las atualizadas conforme a tecnologia evolui. Segundo Lourenço e Horta (2012), a abordagem baseada no conhecimento foi aplicada com sucesso moderado. Além disso, a precisão da solução muitas vezes não é satisfatória, uma vez que força as equações analíticas do projeto a serem simples (JAFARI et al., 2010).

2.2.2 MÉTODOS BASEADOS EM OTIMIZAÇÃO

A abordagem de projeto baseada em otimização utiliza um algoritmo para definir as dimensões do circuito. A otimização baseia-se na avaliação do circuito a cada iteração, que pode ser baseado em equações ou baseada em simulação elétrica.

Nos métodos de otimização baseados em equações, os sistemas usam modelos simplificados para prever o comportamento e desempenho do dispositivo dentro do ciclo de otimização (JAFARI; BIJAMI; ZEKRI, 2010). Segundo Lourenço e Horta (2012), o ponto forte dos métodos baseados em equações é o curto tempo de avaliação, tornando-os, assim como as abordagens baseadas em conhecimento, extremamente adequadas para derivar projetos simples em que não sejam consideradas muitas especificações.

Em Hershenson, Boyd e Lee (1998) alguns circuitos amp-op são otimizados usando Programação Geométrica (GP). O tempo de execução é na ordem de poucos segundos. Os resultados atingidos são comparados com os do simulador HSPICE e mostram que o sistema de equações atinge resultados muito próximos ao do simulador. No entanto, a aplicação do modelo é bastante difícil de derivar para novos circuitos. Essa

abordagem possui baixa precisão e limita as soluções em algumas regiões de operação do circuito (SEVERO et al., 2012).

A outra maneira é usar um simulador elétrico do tipo SPICE para avaliar o circuito com um modelo completo. Essa alternativa fornece melhor precisão, mas exige maior poder computacional.

Na Tabela 1 é mostrada uma relação de alguns trabalhos relacionados baseados em otimização, contendo de forma breve algumas informações como método de otimização utilizado, circuito dimensionado e sistema de avaliação das especificações do circuito.

Tabela 1 – Trabalhos sobre otimização automática de CIs analógicos

Autor - ano	Circuito Otimizado	Método de Otimização	Sistema Avaliador	Tecnol.
(JAFARI et al., 2010)	amp-op de dois estágios	GA	Equações	0.18 μm
(JAFARI et al., 2012)	amp-op de 3 estágios	NHSFL	Equações	0.18 μm
(YENGUI et al., 2012)	amp de transimpedância e driver óptico	GA e SQP	Spectre	0.18 μm
(BARRA et al., 2012)	amp-op de dois estágios Miller	MOGA	Equações	0.18 μm
(SEVERO et al., 2012)	amp-op diferencial e <i>folded cascode</i>	SA	HSPICE	0.18 μm
(DE et al., 2016)	amp-op de dois estágios e comparador	CBC	Equações	0.35 μm
(DEHBASHIAN, 2017)	amp-op de dois estágios	GSA-PSO	HSPICE	0.25 μm
(SASIKUMAR, 2017)	amp-op de dois estágios	PSO-SA	Equações	0.35 μm
(MALLICK et al., 2017)	amp-op diferencial e de dois estágios	GSA e PSO	Spectre	0.35 μm
(VURAL, 2012)	amp-op diferencial e de dois estágios	PSO	Equações	0.35 μm
(HORTA, 2012)	amp-op <i>folded cascode</i>	Otimização evolutiva	HSPICE	0.18 μm

Jafari et al. (2010) utiliza algoritmos genéticos para realizar o dimensionamento automático de circuitos integrados analógicos. A principal parte do trabalho é o dimensionamento da largura e comprimento do gate dos transistores que compõem o circuito, valores de resistores, capacitores e tensão e corrente de polarização. O Algoritmo Genético é utilizado juntamente com uma avaliação baseada em equações para produzir uma ferramenta que determine os tamanhos de dispositivos em um circuito analógico. O software pode otimizar os tamanhos do dispositivo considerando os objetivos e restrições impostas. Após o dimensionamento de um amp-op de dois estágios, o circuito é simulado usando o programa HSPICE com processo de 0.18 μm . O trabalho demonstra a eficiência dos algoritmos genéticos na determinação dos tamanhos dos dispositivos

em um circuito analógico. Porém, o uso de equações simplificadas, apesar de acelerar a análise de desempenho do circuito em cada iteração, acarreta a perda de precisão, pois o uso de equações dificilmente reflete o funcionamento do dispositivo na prática.

Em Jafari et al. (2012) é proposto um algoritmo denominado NHSFL (New Hybrid Suffled Frog Leaping) para dimensionar dois amplificadores operacionais em tecnologia $0.18 \mu m$, um Miller com capacitores aninhados de três estágios de ganho denominado NMC e outro chamado DPZCC (*Double Pole-Zero Cancellation compensated amp-op*). Esses amplificadores apresentam 11 e 18 transistores, respectivamente. Os circuitos são avaliados através de equações. As variáveis livres do algoritmo representam o comprimento e largura do gate dos transistores, além dos valores de capacitância do circuito. Os resultados do dimensionamento de ambos os amplificadores atingem as especificações desejadas para os circuitos. Os resultados são comparados para os mesmos circuitos com os algoritmos genéticos e algoritmo imperialista competitivo. A comparação mostra que o algoritmo proposto atinge melhores valores considerando desempenho e tempo de design.

No trabalho de Yengui et al. (2012) é proposto um algoritmo híbrido obtido pela combinação do algoritmo genético (GA) com programação sequencial quadrática (SQP). O GA é o principal otimizador, enquanto o SQP é usado para refinar mais os resultados. O algoritmo é aplicado para otimizar dois circuitos: um amplificador de transimpedância (TIA) e um driver óptico em tecnologia $0.18 \mu m$, que são parte de uma rede ótica on-chip (ONoC). Os circuitos apresentam 12 e 8 variáveis livres, respectivamente. O artigo não demonstra os valores de cada especificação dos circuitos. Os resultados são comparados através de uma figura de mérito com o projetos dos mesmos circuitos utilizando o algoritmo SQP, em que o desempenho é determinado através do simulador Spectre.

Em Barra et al. (2012) é proposta uma metodologia de otimização de amp-ops utilizando MOGA (Algoritmo genético multiobjetivo). O algoritmo é utilizado para realizar a variação dos parâmetros do circuito, como dimensões do gate dos transistores, valor de capacitância e corrente de polarização. Em cada iteração do algoritmo é realizada uma avaliação do circuito através do uso de equações. Para avaliar a abordagem proposta, é realizada a otimização de um amp-op de dois estágios do tipo Miller em tecnologia $0.18 \mu m$. Os objetivos no projeto são maximizar o GBW (ganho por largura de banda) e minimizar a potência dissipada. A otimização considera 14 variáveis livres. Após a execução, é realizada uma simulação elétrica no simulador Spectre e os resultados são comparados entre os obtidos pelas equações, pela simulação elétrica e apresentados em Jafari et al. (2010).

O trabalho publicado por Severo et al. (2012) apresenta o dimensionamento automático de um amplificador diferencial e um amplificador *folded cascode* utilizando o algoritmo Simulated Annealing para ajustar os parâmetros dos transistores do circuito.

Para o amplificador diferencial deseja-se maximizar o ganho e o intervalo em modo comum de entrada (ICMR) considerando outras especificações como restrições, como tamanho máximo de área, margem de fase e ganho por largura de faixa (GBW). O projeto do amplificador *folded cascode* considera 15 variáveis livres representando as dimensões do gate de 6 transistores, duas tensões de referências e a corrente de polarização do circuito. Para esse amplificador deseja-se minimizar a potência consumida e a área do circuito enquanto as restrições são ganho em baixas frequências (A_{v0}), GBW, margem de fase e tempo de resposta (*slew rate* - SR). Os circuitos são avaliados através de simulação elétrica do tipo SPICE. Uma função custo é apresentada para representar o desempenho do circuito, no qual deseja-se minimizá-la. Os circuitos são implementados em tecnologia CMOS $0.18 \mu m$ com uma tensão de alimentação de 1.8 V. A metodologia de projeto é implementada em Matlab e o simulador Synopsys HSPICE é utilizado para simulação do circuito.

Em Mallick et al. (2017) propõe-se um algoritmo híbrido baseado em população denominado algoritmo de busca gravitacional (GSA) combinado com otimização por enxame de partículas (PSO) (GSA-PSO) para projetar dois circuitos analógicos, um amplificador diferencial e um amplificador operacional de dois estágios em tecnologia CMOS de $0.35 \mu m$. Os circuitos são otimizados utilizando-se 14 e 19 variáveis livres, respectivamente, representando as dimensões do gate dos transistores, corrente de polarização, capacitor de carga e um capacitor de acoplamento (apenas no amplificador operacional). O objetivo da otimização é minimizar a área do gate dos transistores e o consumo, além de maximizar o ganho em baixas frequências. São utilizadas equações para a avaliação do circuito durante o processo de otimização. Após o dimensionamento é realizada simulação SPICE em software Cadence IC 5.1. Os resultados obtidos no trabalho através de simulação elétrica mostram que o uso da estratégia híbrida levou a melhores resultados em relação a outros algoritmos da literatura em termos de velocidade de convergência, especificações de projeto e parâmetros de desempenho dos amplificadores. Os resultados obtidos mostram que foi possível obter um circuito de menor área e melhor desempenho como ganho e dissipação de potência em comparação com outros algoritmos. Segundo os autores, a maior dificuldade nesse trabalho é o ajuste dos parâmetros dos algoritmos GSA e PSO.

No trabalho de Vural (2012) foi utilizada a heurística PSO para o dimensionamento de um amplificador diferencial com espelho de corrente e um amplificador operacional de dois estágios. Os circuitos foram projetados utilizando tecnologia TSMC de $0.35 \mu m$. Para a otimização são utilizadas equações e o objetivo é apenas reduzir a área. Após o processo de dimensionamento, os parâmetros de projeto resultantes são utilizados em simulador SPICE para validar os valores das especificações de projeto. São necessários alguns ajustes manuais para que se atenda a todas as especificações dos projetos. Os resultados alcançados satisfazem todas as especificações de projeto e a área

de ambos os circuitos é minimizada em comparação com outros métodos como GA e Busca Tabu.

Em Lourenço e Horta (2012) é proposta uma ferramenta para realizar a síntese automática de circuitos analógicos utilizando otimização evolutiva. Como caso de estudo é realizado o dimensionamento de um amp-op *folded cascode* com 15 variáveis livres e os objetivos são minimizar área e potência. A avaliação da performance é realizada através de simulação SPICE no simulador HSPICE. O projeto também leva em consideração a produtividade, realizada através de análises de corners. Nessa análise o circuito é simulado considerando as piores hipóteses em que o circuito pode funcionar de acordo com as variações de parâmetros de processo, como por exemplo tensão de alimentação, temperatura, etc.

Em De et al. (2016) são dimensionados dois CIs, um comparador de dois estágios e um amp-op de dois estágios *folded-cascode* utilizando um algoritmo proposto recentemente denominado Otimização de Corpos Colidentes (*Colliding Bodies Optimization - CBO*). O CBO é um algoritmo multiagente que não depende de nenhum parâmetro de controle interno. O principal objetivo desse trabalho é otimizar o dimensionamento dos gates dos transistores para reduzir as áreas ocupadas pelos circuitos e obter melhores parâmetros de desempenho dos circuitos. Algumas das especificações consideradas no projeto são A_v0 , margem de fase, ICMR, potência dissipada, tempo de resposta e etc. A tecnologia utilizada é $0.35 \mu m$ com uma tensão de alimentação de 5 V. No comparador são utilizadas 7 variáveis livres, representando a largura e comprimento do gate dos transistores e um resistor. Para o amp-op a otimização é realizada utilizando 17 variáveis livres representando as dimensões dos gates dos transistores, dois resistores e um capacitor. A otimização é realizada utilizando equações e após o dimensionamento o circuito é simulado para validar seu funcionamento.

No trabalho de Dehbashian (2017) é proposto um algoritmo híbrido utilizando o Algoritmo de Busca Gravitacional (GSA) e PSO para dimensionar um amp-op de dois estágios em tecnologia de $0.25 \mu m$. O objetivo da otimização é reduzir a área e consumo do CI. A otimização é realizada considerando 12 variáveis livres representando as dimensões do gate de 5 transistores, corrente de polarização e tamanho de um capacitor. O circuito é alimentado com 1.8 V e as especificações consideradas são ganho em baixas tensões, margem de fase, consumo de potência, *output swing* entre outros. Os resultados são comparados com outros trabalhos e atingem melhores resultados principalmente para potência e área do gate.

Em Sasikumar (2017) é usada uma implementação híbrida com PSO e GA para o dimensionamento de um amp-op de dois estágios em tecnologia $0.35 \mu m$. O objetivo é reduzir a área e potência dissipada pelo circuito. O sistema avaliador no ciclo de otimização é baseado em equações e após o circuito dimensionado é realizada simulação no simulador Cadence IC 5.1. O sistema é modelado como um problema de otimização

com nove variáveis livres representando as dimensões do gate dos transistores e uma capacitância. Os resultados são comparados com os obtidos no dimensionamento do mesmo amp-op utilizando apenas o PSO e demonstram que os algoritmos utilizados atingiram melhores resultados para a maioria das especificações do amplificador.

2.2.3 CONCLUSÃO

Neste capítulo foram apresentados os níveis hierárquicos para assegurar o funcionamento dos circuitos integrados analógicos, como nível de sistema, nível de bloco, nível de circuito, nível de leiaute e nível de fabricação e teste. Também foi apresentado o fluxo básico para o dimensionamento automático de CIs analógicos, transformando o processo de dimensionamento em um problema de otimização combinatória e fazendo o uso de um algoritmo heurístico para tal tarefa.

Também foi apresentada uma revisão bibliográfica sobre o projeto automático de circuitos integrados analógicos. Através deste estudo bibliográfico observou-se que diversos trabalhos buscam estudar métodos que auxiliem no projeto de CIs analógicos visando diminuir o tempo de design e aumentar a qualidade dos CIs projetados. Na maioria dos trabalhos descritos afirma-se que um projeto dependente apenas do conhecimento do projetista pode ser muito demorado e é bastante complexo quando o circuito possui um número grande de variáveis livres e várias especificações a serem atendidas. Um grande número de trabalhos recentes busca explorar o uso de inteligência artificial para realizar o dimensionamento do circuito. Alguns dos trabalhos verificam o comportamento de novos algoritmos para circuitos considerados básicos, enquanto em outros o foco é realizar o dimensionamento de circuitos mais complexos, com maior número de variáveis livres e número de restrições. Existe uma dificuldade de comparação entre os resultados de boa parte dos trabalhos pelo fato de que eles utilizam diferentes topologias de circuitos ou diferentes tecnologias de fabricação para a simulação ou cálculo das especificações.

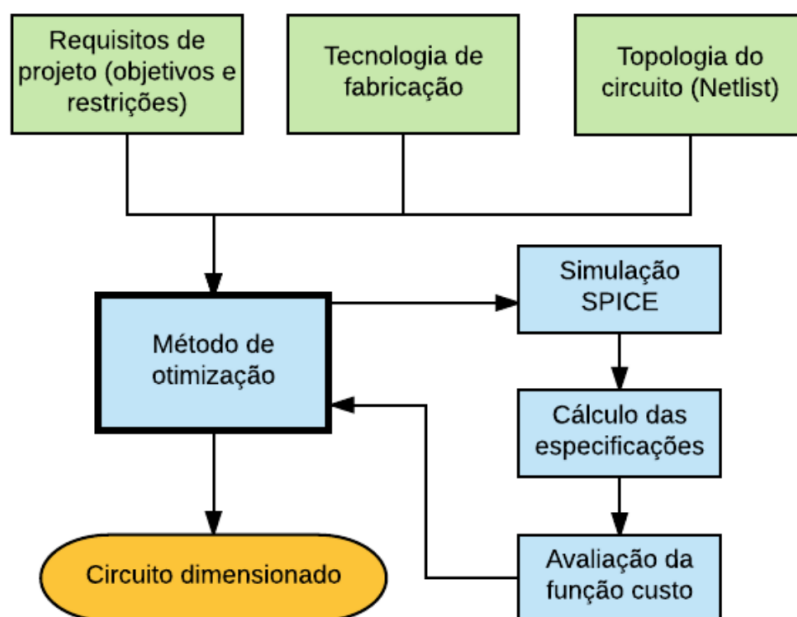
3 A FERRAMENTA UCAF

Este capítulo busca detalhar o funcionamento das principais partes da ferramenta utilizada para o dimensionamento automático de blocos analógicos deste trabalho. São listadas as informações que a ferramenta deve receber do usuário, como ela realiza a alteração dos valores para cada variável do circuito e como é realizada a simulação e avaliação das especificações.

Neste trabalho foi utilizada a ferramenta UCAF, desenvolvida em linguagem Matlab® por Severo (2012), com funcionalidades acrescentadas por Domanski (2016), no Grupo de Arquitetura de Computadores e Microeletrônica da Unipampa (GAMA).

A ferramenta UCAF baseia-se em uma metodologia de otimização utilizando heurísticas para pesquisar o espaço de projeto e simulação elétrica SPICE para avaliação do comportamento dos circuitos. Na figura 4 ilustra-se o fluxo básico dessa ferramenta.

Figura 4 – Fluxo de projeto da ferramenta UCAF.



Adaptado de Severo (2012)

O método de otimização recebe como entrada as especificações a serem atendidas pelo CI, as restrições do projeto, a tecnologia de fabricação e a descrição elétrica do circuito, chamada de *netlist*. Com essas informações, o método de otimização define valores para as variáveis livres do circuito, que representam as dimensões dos dispositivos, valores de capacitores, resistores e etc, e então realiza a simulação elétrica a fim de obter os valores das especificações e calcular a função custo, com base na equação 3.4. Com base na avaliação da função custo, o método de otimização atribui novos valores às variáveis do circuito com o intuito de buscar uma solução melhor. Esse ciclo se repete até que se

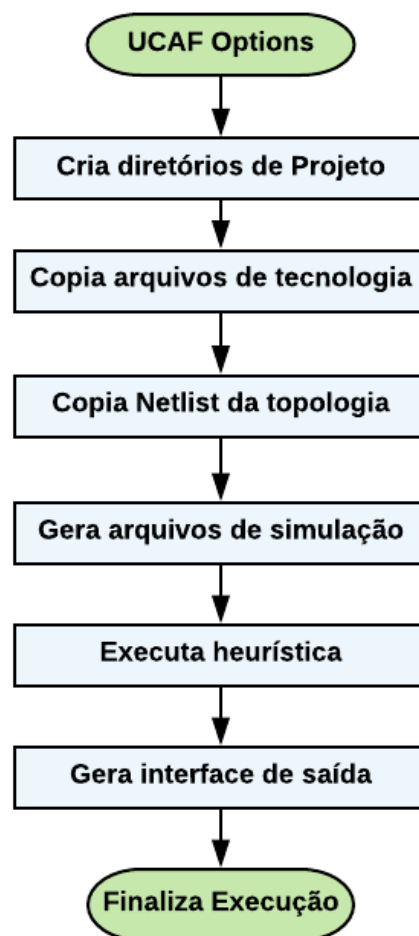
atinja uma solução considerada satisfatória, isto é, que atenda aos requisitos do projeto ou algum critério de parada definido pelo usuário.

A ferramenta UCAF é composta por vários módulos, como módulo principal, módulo de otimização, módulo de simulação elétrica, entre outros. Dessa forma, a manutenção e escalabilidade da ferramenta pode ser realizada de forma simples. Caso seja desejado utilizar uma nova heurística para realizar a otimização do circuito, é necessário alterar apenas o módulo de otimização.

3.1 NÚCLEO PRINCIPAL

A ferramenta possui um núcleo onde o usuário ajusta as variáveis de configuração. O núcleo baseia-se na estrutura UCAF Options, que representa as configurações ajustadas na interface de entrada. A figura 5 demonstra de forma resumida o funcionamento do núcleo da ferramenta.

Figura 5 – Fluxograma de execução do núcleo da ferramenta.



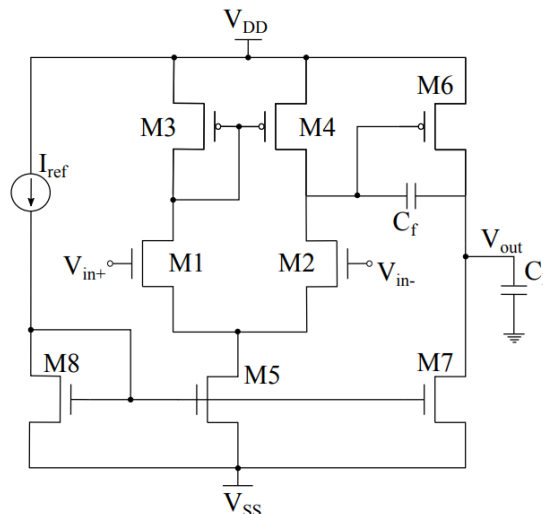
Adaptado de Severo (2012)

As duas primeiras etapas apresentadas no fluxograma representam a preparação

dos diretórios e arquivos necessários para a execução. É criado um diretório de projeto e neste diretório são copiados todos os arquivos necessários para a simulação do CI, como netlist, arquivos referentes a tecnologia e arquivos de simulação elétrica gerados. A partir disso, o algoritmo de otimização altera os valores dos parâmetros do circuito a fim de atingir as especificações. Com o circuito dimensionado, são gerados relatórios para o usuário contendo informações como tamanhos e valores para os parâmetros de cada dispositivo, valores atingidos para cada especificação e demais informações pertinentes.

Por exemplo, para o dimensionamento do OTA Miller representado na figura 6, o usuário deve criar um netlist, como o demonstrado na figura 7, e informar à ferramenta o caminho desse arquivo representando o circuito.

Figura 6 – Esquemático de um OTA Miller em tecnologia CMOS



Fonte: Adaptado de Severo (2012)

As linhas iniciadas com um asterisco são interpretadas como comentário e, portanto, não são consideradas na ferramenta. Os parâmetros 'lr' e 'wr' representam o comprimento e a largura do gate dos transistores, respectivamente. Por exemplo, na linha 10 da figura, 'lr=L1*1e-6' significa que comprimento do gate do transistor chamado 'x1' será de L1 micrômetros, sendo que L1 é uma variável alterada a cada iteração pela heurística de otimização e escrito em outro arquivo pela ferramenta, este arquivo se encontra no mesmo diretório e é importado para esse netlist. A mesma lógica é seguida para todos os outros 'lr', 'wr', IB e Cc. Dessa forma, a cada iteração do algoritmo é necessário alterar apenas o arquivo de parâmetros (mostrado na figura 8). Cada parâmetro é uma variável livre no processo de otimização.

Os valores das dimensões dos transistores são limitados pelo modelo do transistor disponível para simulação elétrica. A capacitância e a corrente devem possuir uma faixa limitada pela viabilidade desse valor para a posterior fabricação e bom

Figura 7 – Netlist representando o OTA Miller da figura 6.

```

1  *netlist representando o OTA Miller
2
3  *inclusão das bibliotecas com os modelos elétricos
4  .lib 'xh018/lp3mos/cr018gpii_v1d0.l' TT_RFMOS
5  .lib 'xh018/lp3mos/cr018gpii_v1d0.l' stat_noise
6
7  *Descricao elétrica
8  .subckt OTA_miller vp vn vdd vss vout
9
10 x1 2 vn 3 3 nmos_rf lr='L1*1e-6' wr='W1*1e-6'
11 x2 5 vp 3 3 nmos_rf lr='L1*1e-6' wr='W1*1e-6'
12 x3 2 2 vdd vdd pmos_rf lr='L3*1e-6' wr='W3*1e-6'
13 x4 5 2 vdd vdd pmos_rf lr='L3*1e-6' wr='W3*1e-6'
14 x5 3 4 vss vss nmos_rf lr='L5*1e-6' wr='W5*1e-6'
15 x6 vout 5 vdd vdd pmos_rf lr='L6*1e-6' wr='W6*1e-6'
16 x7 vout 4 vss vss nmos_rf lr='L7*1e-6' wr='W7*1e-6'
17 x8 4 4 vss vss nmos_rf lr='L5*1e-6' wr='W5*1e-6'
18 Ibias vdd 4 'IB*1e-6'
19 Cc 5 vout 'Cc*1e-12' IC=0
20 Cl vout 0 3p IC=0
21
22 .ends

```

Fonte: Autor.

funcionamento do circuito.

Uma vez gerados os valores para as variáveis livres do circuito, ele então é simulado e suas especificações são avaliadas através de uma série de *testbenches*, mostrados na seção 3.2. O simulador HSPICE é invocado pela ferramenta através de uma chamada de sistema contendo os arquivos de entrada e saída para a simulação. Na seção 3.2 é demonstrado um exemplo de arquivo de entrada representando um *testbench*.

3.2 ESPECIFICAÇÕES E TESTBENCHES

As especificações de um circuito representam o comportamento que se deseja que o circuito apresente. Nesta aplicação, é necessário que a otimização automática leve em conta todas elas. Para isso, o usuário deve informar quais são as especificações, os valores desejados para cada uma (limites mínimos e máximos aceitáveis) e quais delas são principais focos da otimização, ou seja, quais deseja-se minimizar o seu valor (ou maximizar).

As principais especificações para os amplificadores operacionais dimensionados no trabalho são as seguintes (SEVERO; GIRARDI, 2011):

- Avo - representa o ganho em baixas frequências. Obtido através de análise no domínio da frequência em malha aberta, através de simulação CA.

Figura 8 – Exemplo de arquivo de parâmetro gerado pela ferramenta UCAF para o circuito da figura 6.

```
* Projeto_Miller Parameters File
.PARAM W1=6.900956e+00
.PARAM L1=3.062347e-01
.PARAM W3=8.000000e+00
.PARAM L3=2.855465e-01
.PARAM W5=7.098635e+00
.PARAM L5=4.939068e-01
.PARAM W6=8.000000e+00
.PARAM L6=1.800000e-01
.PARAM W7=3.278318e+00
.PARAM L7=1.965620e-01
.PARAM IB=1.265602e+01
.PARAM Cc=9.633470e+00
```

Fonte: Autor.

- GBW - produto ganho-largura de faixa. Obtido pela análise no domínio da frequência.
- PM - é a margem de fase. Assim como o Avo e GBW, a margem de fase é obtida pela análise no domínio da frequência.
- SR - velocidade de resposta. Para se obter esse valor é necessária uma análise no domínio do tempo, obtida por uma simulação transiente.
- ICMR - tensão no modo comum de entrada. É realizada uma simulação CC (variação linear no nível de tensão ou corrente).
- OS - Faixa de tensão de saída. É realizada uma simulação da faixa de tensão de saída do circuito durante a sua operação, obtida também através de simulação CC.
- Potência dissipada - Calculada através de uma simulação do ponto de operação do circuito.
- Área de gate - é calculada pela multiplicação da largura pelo comprimento do canal de cada transistor.

As especificações são divididas em dois tipos: as de restrição e as de otimização. Todas as especificações listadas podem ser de qualquer um dos dois tipos, conforme a configuração dada pelo usuário. As especificações de restrição são as que apresentam um intervalo de valores aceitáveis. Caso o valor obtido esteja fora desse intervalo, a solução é descartada. As de otimização não invalidam a solução alcançada e deseja-se

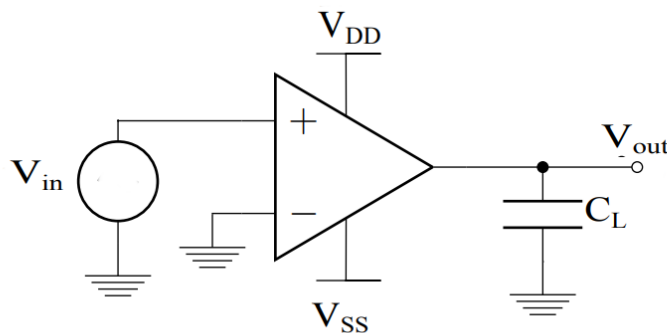
sempre minimizá-las (ou maximizar), como por exemplo o consumo de um CI, quanto menor, melhor. Esse conceito é detalhado na seção 3.4.

Para obtenção dos valores das especificações, são necessárias simulações elétricas utilizando diferentes análises: CA, CC e transiente. Os *testbenches* de medição, implementados com base em Allen et al. (1987), são mostrados nas seguintes seções.

1. Análise no domínio da frequência em malha aberta e potência dissipada

Este *testbench* é capaz de medir as características em malha aberta do circuito, como A_{v0} , GBW, margem de fase e potência dissipada. O circuito utilizado para esta análise é mostrado na figura 9. A frequência da tensão senoidal de entrada é variada dentro de uma faixa, e a tensão de saída é calculada no domínio da frequência.

Figura 9 – Configuração de circuito para medição de características AC.



Fonte: Autor.

A ferramenta gera de forma automática os arquivos de netlist representando os *testbenches*, como demonstrado na figura 10.

2. Velocidade de resposta

A velocidade de resposta (slew rate) do amplificador é medida através de uma simulação transiente para obter a taxa de variação da frequência no decorrer do tempo (dV/dt). O circuito da figura 11 pode ser utilizado para esse tipo de análise (SEVERO; GIRARDI, 2011). Deve-se gerar uma fonte quadrada na entrada do circuito e verificar a alteração na tensão de saída. O objetivo da medição é verificar a taxa de alteração do nível de saída do amplificador operacional

Para obter a taxa de variação do op-amp, a taxa de alteração do sinal de saída deve ser medida nas bordas de subida e descida. A inclinação da curva (subida ou descida) é calculada na região mais linear do gráfico. A figura 12 mostra um exemplo dessa medição realizada de forma elétrica.

Figura 10 – Netlist gerado pela ferramenta representando o circuito da figura 9.

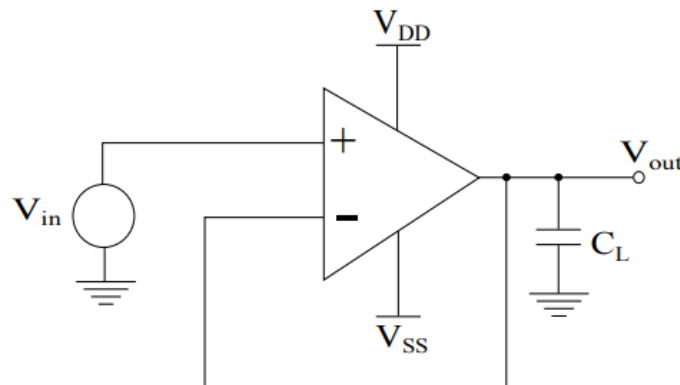
```

1  Title: Projeto_Miller_OpenLoopAC_type1
2  *Netlist Generated by UCAF Framework
3
4
5  .OPTIONS POST=1
6  .OPTIONS POST_VERSION=9601
7
8  *Include Parameters
9  .include Projeto_Miller.par
10
11 *Include Netlist
12 .include Miller_Netlist.txt
13
14 *Circuit Simulations
15 VDD VDD 0 9.000000e-01
16 VSS VSS 0 -9.000000e-01
17 X1 VINP 0 VDD VSS OUT OTA_miller
18 VINP VINP 0 DC 0 AC 1
19 .AC DEC 100 1 1G
20 .DC VINP 0.000000e+00 9.000000e-01 9.000000e-01
21 .END

```

Fonte: Autor.

Figura 11 – Configuração de circuito para medição do slew rate.



Fonte: Autor.

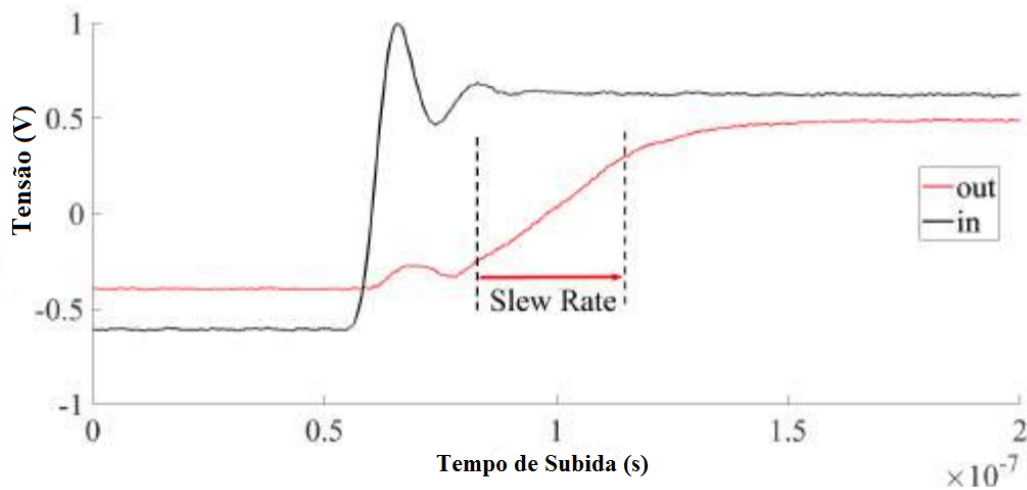
3. Tensão de entrada em modo comum

Para medir a tensão de entrada em modo comum (ICMR) utiliza-se a mesma configuração de circuito para medir o slew rate, mostrada na figura 11. Para esta medição, uma onda de formato triangular é usada como entrada no circuito e observa-se a região linear da relação $V_{in} \times V_{out}$ do circuito, como mostra a figura 13.

4. Faixa de tensão de saída

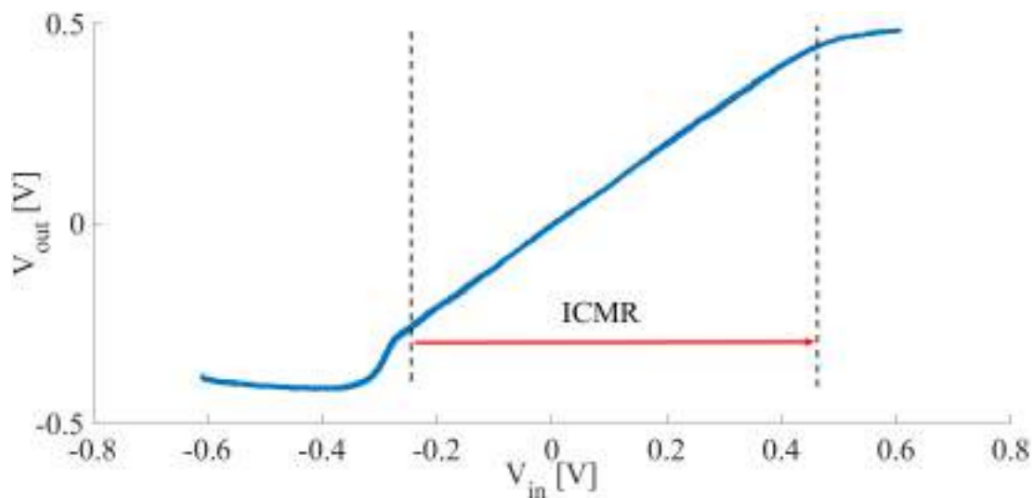
A faixa de tensão de saída, ou output swing (OS), pode ser obtida através de uma simulação de amplificação de forma que seja possível saturar o sinal de saída.

Figura 12 – Resposta de saída para medição de taxa de variação.



Fonte: Autor.

Figura 13 – $V_{in} \times V_{out}$ para a medição do ICMR.



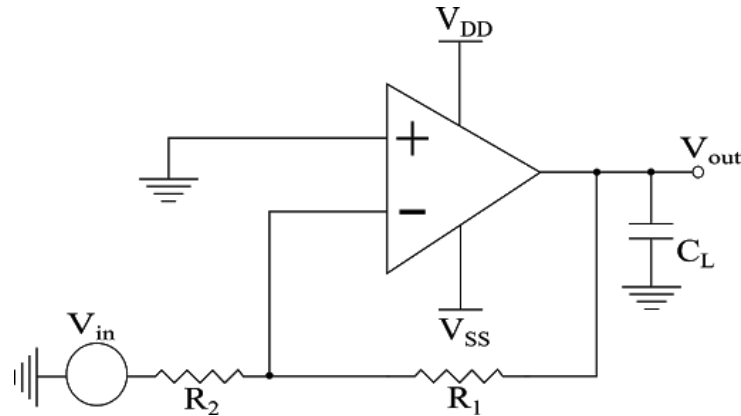
Fonte: Autor.

Através da faixa linear e de saturação do sinal de saída, os valores máximos e mínimos da excursão do sinal de saída são obtidos. A figura XXX mostra o gráfico $V_{in} \times V_{out}$ onde o OS é medido nos pontos máximo e mínimo onde a saída ainda é linear.

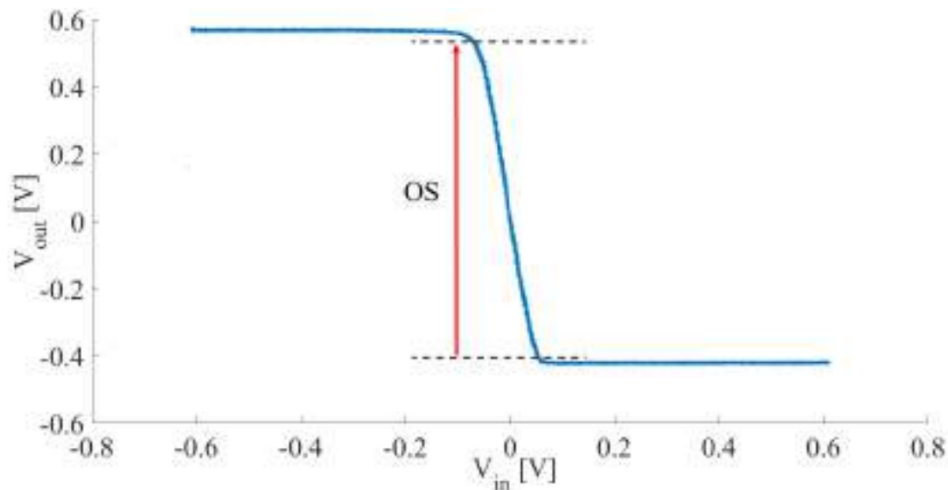
A configuração utilizada na ferramenta para a medição do OS é mostrada na figura 15 (SEVERO, 2012).

Para que a ferramenta possa obter os valores de cada especificação na simulação, é utilizado um *toolbox* Hspice para Matlab, que é um conjunto de funções fornecidas

Figura 14 – Configuração de circuito para medição do output swing.



Fonte: Autor.

Figura 15 – Medição $V_{in} \times V_{out}$ para extração do output swing.

Fonte: Autor.

que permitem visualizar e manipular os resultados gerados na simulação.

3.3 BLOCO DE OTIMIZAÇÃO

O bloco de otimização é um dos principais blocos da ferramenta. Ele é o bloco responsável por definir os valores e dimensões para os componentes do circuito. Para isso, ele utiliza um algoritmo baseado em heurística que gera valores para as variáveis do circuito.

As soluções geradas para o circuito devem sempre estar dentro das dimensões máximas e mínimas permitidas pela tecnologia utilizada. A partir dos valores gerados é realizada a simulação e o cálculo da função custo com base nos valores obtidos para as especificações. Baseado na função custo, o algoritmo de otimização altera os valores

para as variáveis. A função custo utilizada para avaliação do circuito pelas heurísticas da ferramenta é apresentada na seção 3.4

Na ferramenta UCAF em versões anteriores foram adicionadas as heurísticas dos Algoritmos Genéticos (GA) e Simulated Annealing (SA), ambas por Severo (2012), e PSO, desenvolvida por Domanski (2016), utilizada nesse trabalho e apresentada na seção 4.4.

3.4 FUNÇÃO CUSTO

A função custo (ou função de aptidão) tem grande importância no processo de otimização. Ela deve ser formulada de forma a refletir a qualidade da solução.

É muito importante que a função custo consiga diferenciar na proporção correta as melhores das piores soluções. Se houver pouca precisão em diferenciar as soluções boas das más, todo o processo de otimização será comprometido, uma vez que uma boa solução pode ser colocada de lado durante a execução do algoritmo, podendo levar o mesmo a explorar regiões não promissoras e a gastar mais tempo de computação (RAZAVI, 2001).

Neste trabalho a função custo leva em consideração tanto as especificações que devem ser otimizadas quanto as de restrição.

Dado um conjunto de funções de desempenho do circuito (especificações de projeto) $Y(\mathbf{r}, \mathbf{q}) = \{S_1, S_2, \dots, S_s\}$ que depende de um conjunto de valores de parâmetros de projeto \mathbf{r} e de um conjunto de valores de parâmetros de tecnologia \mathbf{q} . S é uma especificação individual e s é o número de especificações de projeto. As funções de desempenho para um OTA são as apresentadas na seção 3.2 (Av0, GBW, SR, potência dissipada, etc). Os parâmetros de projeto são as variáveis livres que o designer pode manipular para projetar o circuito, como dimensões do gate dos transistores, número de transistores unitários em série e em paralelo, correntes de referência, valores de capacitores, etc. Os parâmetros da tecnologia incluem informações sobre o modelo elétrico (como espessura do óxido, tensão de limiar, etc.), tensão de alimentação e temperatura de operação.

A aceitação de um circuito pode ser expressa da seguinte forma:

$$Y(\mathbf{r}, \mathbf{q}) \in \Phi. \quad (3.1)$$

O conjunto Φ contém a região de especificações aceitáveis no espaço de desempenho do circuito. A região de aceitação Ψ no espaço de projeto é definida como:

$$Y(\mathbf{r}, \mathbf{q}) \in \Phi \rightarrow \mathbf{r} \in \Psi. \quad (3.2)$$

Um circuito fabricado será considerado aceitável se todos os seus desempenhos reais estiverem dentro dos limites aceitáveis, isto é, se $S_i^L \leq S_i \leq S_i^U$, onde os índices L e

U correspondem aos limites inferiores e superiores das especificações, respectivamente.

A abordagem proposta para a otimização do desempenho do circuito é um problema de programação não linear e requer a formulação de uma única função custo a ser minimizada, sujeito a um conjunto de restrições de desigualdade, como na seguinte forma padrão:

$$\begin{aligned} & \underset{\mathbf{r}, \mathbf{q}}{\text{minimize}} && F_i(\mathbf{r}, \mathbf{q}), \quad i = 1, \dots, I \\ & \text{sujeito a} && C_j(\mathbf{r}, \mathbf{q}) \leq C_{j(\text{ref})}, \quad j = 1, \dots, J \end{aligned} \quad (3.3)$$

onde I é o número total de F_i especificações a serem otimizadas e J é o número de C_j funções de restrições de performance. Aqui, \mathbf{Y} pode ser reescrita como um conjunto de objetivos de projeto e restrições de performance como $\mathbf{Y}(\mathbf{r}, \mathbf{q}) = \{F_1, \dots, F_I, C_1, \dots, C_J\}$.

$C_j(\mathbf{r}, \mathbf{q})$ é uma função dependente do tipo de especificação: valor mínimo requerido ($C_{\min}(\mathbf{r}, \mathbf{q})$) ou valor máximo requerido ($C_{\max}(\mathbf{r}, \mathbf{q})$) (BARROS; GUILHERME; HORTA, 2010). Essas funções são mostradas na figura 16, onde a é o máximo ou mínimo valor requerido e b é o valor limite entre a performance aceitável e inaceitável. Os valores entre os pontos entre a e b retornam valores intermediários para as funções de restrições. Esses valores levam a um custo adicional para a função custo conforma a distância do valor obtido estiver do valor desejado. Caso o valor obtido esteja à esquerda do valor desejado, em um caso de minimização, o custo adicional é zero. De forma análoga, caso o valor obtido esteja à direita do valor desejado, em caso de maximização, o custo atribuído a função custo é zero. Ou seja, as restrições possuem um valor desejado e um intervalo considerado aceitável. O valor atribuído à função custo é calculado de forma linear, conforme mostrado na figura 16.

A função custo utilizada é calculada de acordo com um sistema de penalidades, onde o valor é obtido através de um somatório de penalidades calculadas de acordo com a diferença entre o valor esperado e o valor simulado para cada especificação, como mostra a equação:

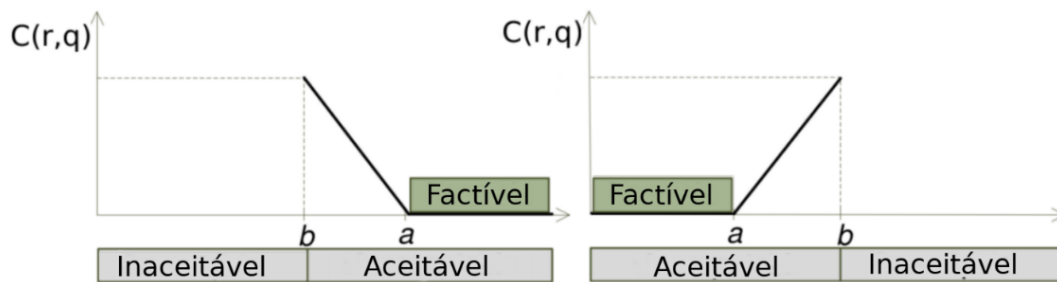
$$f_c(\mathbf{r}, \mathbf{q}) = \sum_{i=1}^I w_i \cdot \hat{F}_i(\mathbf{r}, \mathbf{q}) + \sum_{j=1}^J v_j \cdot \hat{C}_j(\mathbf{r}, \mathbf{q}). \quad (3.4)$$

Aqui, w_i e v_j são pesos que indicam a importância relativa de objetivos e restrições de projeto, respectivamente. \hat{F} and \hat{C} são funções de restrições e objetivos de projeto normalizadas, a fim de manter todos os fatores de soma na mesma ordem de grandeza.

O valor calculado para a função custo representa uma figura de mérito para a otimização. Utilizando a estratégia apresentada, uma função custo com valor zero representa um circuito em que todas as especificações foram atingidas.

Como exemplo, consideremos uma especificação, como o ganho ($Av0$) do amp-op, onde o projetista informa à ferramenta que deseja-se atingir 70 dB e que o mínimo

Figura 16 – Métricas de avaliação das restrições de projeto: (a) especificação de valor mínimo exigido e (b) especificação de valor máximo exigido.



Adaptado de Severo e Girardi (2011)

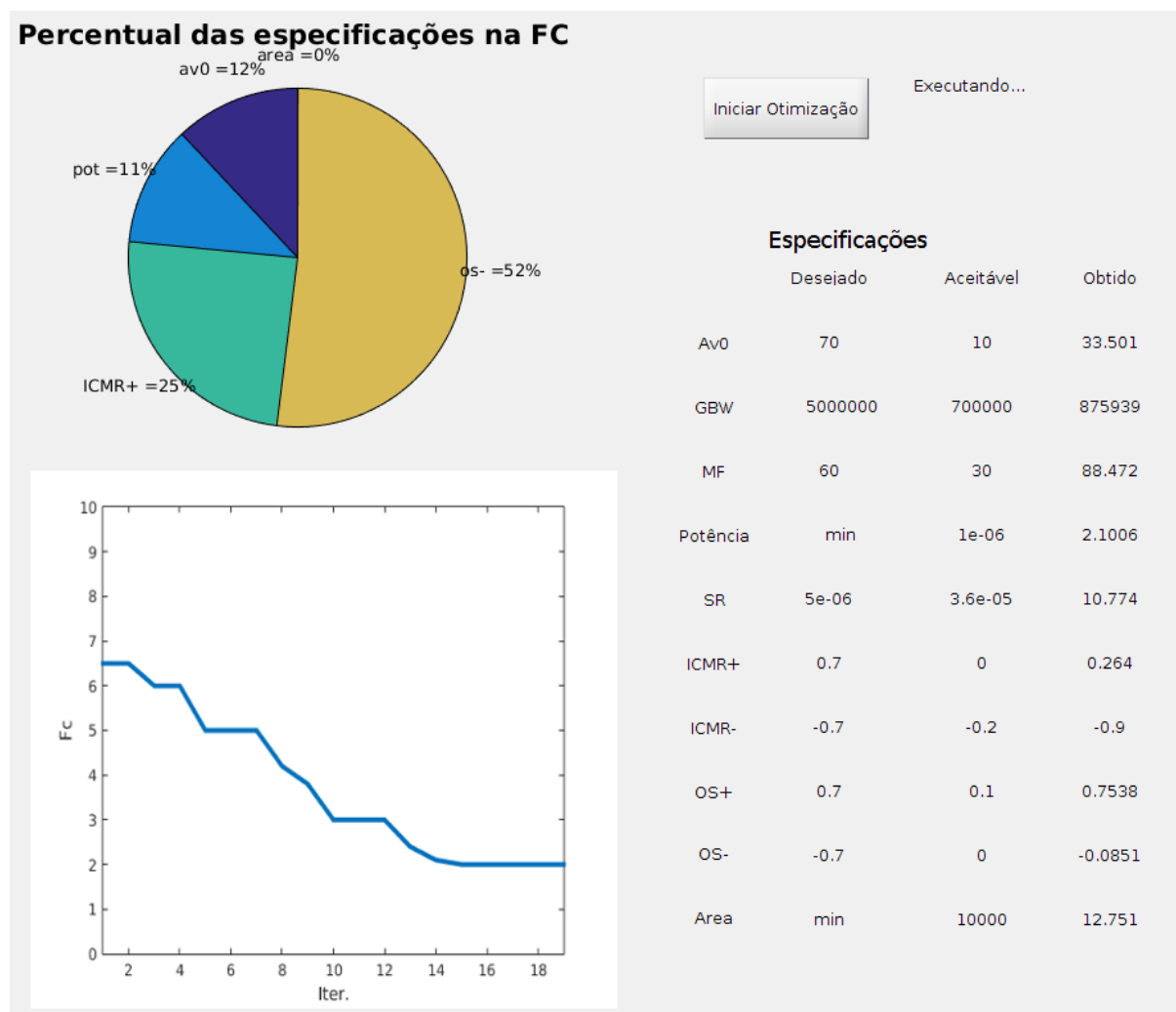
aceitável como uma solução válida é 50 dB. Se o valor obtido na simulação for maior ou igual a 70 dB, será atribuído zero à função custo. Caso contrário, se o valor for menor que 70 e maior ou igual a 50 dB, será atribuído um valor maior que zero e menor que 1. Por exemplo, se ganho for igual a 60 dB, o valor adicionado à função custo será de 0.5 multiplicado pelo peso de importância dado à essa especificação. Ou ainda, caso o valor obtido seja menor que 50 dB, essa solução é considerada fora da região factível e a solução é descartada.

Para uma especificação de otimização é necessário um valor de referência. Por exemplo, considerando a potência como uma especificação de otimização, para que seja possível atribuir um valor à função custo de acordo com a potência medida, é necessário uma referência de grandeza que se deseja atingir. Atribui-se à função custo uma penalidade dada de acordo com a potência medida dividido pela potência de referência multiplicado pelo peso dado à essa especificação. Considerando que deseja-se minimizar a potência e o projetista informa como referência um valor de 10 μW (grandeza que ele considera satisfatória) e na simulação foi obtido um consumo de 20 μW , será atribuído à função custo um valor de 2 multiplicado pelo peso dado para essa especificação pelo projetista. Se o valor da simulação obtido foi de 10 μW , ainda assim será atribuído o valor de 1 à função custo, o que leva o algoritmo a continuar tentando reduzir essa especificação.

A ferramenta permite também observar, através de uma interface gráfica, quais especificações não estão atingindo o valor desejado e qual o impacto delas no valor da função custo a cada iteração do algoritmo. O gráfico é gerado apenas nos casos em que a solução seja considerada válida, ou seja, aquelas soluções que apresentam todas as especificações dentro do intervalo aceitável. Isso permite ao projetista observar quais especificações estão mais distante do valor desejado para que, se desejável, ele aumente o peso dessa especificação. A evolução da função custo também pode ser observada nessa GUI (*Graphical User Interface*), como pode ser observado na figura 17.

Pode-se observar que as especificações OS-, ICMR+, potência dissipada e ganho

Figura 17 – Gráfico que mostra o impacto das especificações não atendidas na função custo e sua evolução.



Fonte: Autor.

não estão atingindo os valores que foram estipulados como ideais, ainda que estejam dentro dos limites aceitáveis. Como as demais especificações não acrescentaram nenhum valor à função custo, pois atingiram o valor esperado, elas não são apresentadas neste gráfico.

3.5 CONCLUSÃO

Neste capítulo foi demonstrado o fluxo básico de funcionamento da ferramenta UCAF. Foi demonstrado como funciona o seu núcleo principal, que recebe algumas informações do projetista, como netlist do circuito e tecnologia utilizada. Apresentou-se também como os valores a serem simulados são variados e quais os arquivos necessários para essa etapa. Também foi demonstrado como são feitas as análises das

especificações dos circuitos, isto é, quais os *testbenches* são utilizados para as diferentes especificações. Também foi apresentada a ferramenta UCAF e seu funcionamento básico, como principais módulos e função custo utilizada pelo processo de otimização. Por fim, foi apresentado como é realizada a aceitação de uma solução e como é calculada a função custo utilizada pelos algoritmos.

4 HEURÍSTICAS DE OTIMIZAÇÃO

Este capítulo traz uma contextualização sobre alguns métodos de otimização, apresentando de forma mais detalhada três heurísticas bio-inspiradas que são utilizadas neste trabalho. Também é realizado um estudo sobre critérios de parada para os algoritmos.

A otimização ou programação matemática é o estudo de problemas de planejamento e projeto usando ferramentas matemáticas.

Segundo Yang (2010c), é possível escrever a maioria dos problemas de otimização na forma matemática genérica

$$\begin{aligned} \min_{x \in R} \quad & f_i(x), \quad (i = 1, 2, \dots, M), \\ \text{sujeito a} \quad & h_j(x) = 0, \quad (j = 1, 2, \dots, J), \\ & g_k(x) \leq 0, \quad (k = 1, 2, \dots, K), \end{aligned} \quad (4.1)$$

Onde $f_i(x), h_j(x), g_k(x)$ são funções do vetor $x = (x_1, x_2, \dots, x_n)^T$.

Para a resolução desses problemas existem diversos métodos de otimização. Deve-se escolher um conjunto de parâmetros que permita a obtenção de uma solução que atenda aos requisitos dentro do problema (FRANÇA et al., 2005).

Existem diferentes métodos para otimizar uma função, como programação matemática e métodos heurísticos. Vários métodos de programação matemática, como programação linear, programação inteira e programação dinâmica podem ser usados para resolver problemas de otimização. Esses métodos muitas vezes usam informações de gradiente para pesquisar o espaço de solução a partir de um ponto de partida inicial. Em geral, os métodos baseados em gradientes convergem mais rápido e podem obter soluções com maior precisão em comparação com abordagens estocásticas na execução da tarefa de busca local (KAVEH; TALATAHARI, 2010). Para a implementação efetiva desses métodos, as variáveis e a função custo precisam ser contínuas. No entanto, em muitos problemas não são possíveis de ser aplicados, devido a não convexidade, zonas restritas e etc.

Os algoritmos de otimização podem ser classificados de forma simples em algoritmos determinísticos e algoritmos estocásticos. Algoritmos determinísticos são aqueles cujo comportamento pode ser completamente previsto, pois eles são apresentados sempre com o mesmo conjunto de entradas, passos, e realizam os mesmos cálculos. Esses algoritmos garantem a solução ótima em um tempo finito.

Os algoritmos estocásticos encontram soluções em um período de tempo razoável, mas não há garantia de que as soluções ótimas sejam alcançadas e nem sabe-se com certeza se um algoritmo funcionará, pois muitas vezes a complexidade do problema torna impossível pesquisar todas as soluções ou combinações possíveis.

A idéia dos algoritmos estocásticos é ter um método eficiente que funcionará

a maior parte do tempo e será capaz de produzir soluções de boa qualidade. Entre as soluções de qualidade encontradas, espera-se que algumas delas sejam quase ótimas, embora não haja garantia de tal otimização (YANG; DEB, 2009).

Boa parte dos algoritmos estocásticos usam aleatorização. A aleatorização fornece uma boa maneira de realizar uma pesquisa mais global, uma vez que ajuda a fugir da pesquisa apenas local. Isso significa que os algoritmos metaheurísticos se mostram mais adequados para a otimização global. Vale ressaltar que na literatura não há definições acordadas de heurísticas e metaheurísticas. Os dois termos muitas vezes são utilizados como sinônimos por vários autores.

Dois componentes importantes de qualquer algoritmo metaheurístico são: intensificação e diversificação. Diversificação significa gerar soluções diversas para explorar o espaço de busca na escala global, enquanto a intensificação significa focar a busca em uma região local, explorando a informação que existe uma boa solução atual naquela região (GANDOMI; YANG; ALAVI, 2013).

A intensificação procura garantir que as soluções converjam para o resultado ótimo ou próximo disso, enquanto a diversificação via aleatoriedade evita que as soluções sejam restritas a um ótimo local. A boa combinação desses dois principais componentes procura garantir que a otimização global seja alcançável.

Existem diversos algoritmos metaheurísticos. Alguns são baseados em trajetórias de um único agente, como por exemplo o Simulated Annealing (SA). Outros se baseiam em um conjunto de agentes ou população, como algoritmos genéticos (GA) e otimização por enxames de partículas (PSO).

Na figura 18 é possível observar alguns dos vários métodos de otimização baseados em heurísticas que surgiram nas últimas décadas.

Muitos algoritmos metaheurísticos modernos foram desenvolvidos com base em inteligência de enxame na natureza, como o algoritmo dos vaga-lumes (Firefly Algorithm - FA), algoritmo de Busca Cuckoo (Cuckoo Search - CS) e a Otimização por Enxame de partículas (Particle Swarm Optimization - PSO). Os três algoritmos são implementados para otimizar amplificadores operacionais neste trabalho. Nas próximas seções são apresentadas explicações mais detalhadas sobre algoritmos bio-inspirados bem como o funcionamento dos três algoritmos citados no parágrafo.

4.1 ALGORITMOS BIO-INSPIRADOS

Uma classe de metaheurísticas que tem sido muito explorada é a dos algoritmos bio-inspirados (YANG; DEB, 2009; YANG, 2010b; KENNEDY, 2011; KRISHNANAND; GHOSE, 2009). Eles são baseados no comportamento de certos animais ou em processos e modelos de sistemas e fenômenos biológicos (CASTRO; ZUBEN, 2005).

A eficiência destes algoritmos metaheurísticos pode ser atribuída ao fato de que eles imitam as melhores características da natureza, especialmente a seleção do

Figura 18 – Algumas das heurísticas propostas nas últimas décadas.



Adaptado de Gandomi, Yang e Alavi (2013)

mais adequado em sistemas biológicos que evoluíram pela seleção natural ao longo de milhões de anos. Por exemplo, o PSO foi inspirado pela inteligência de peixes e pássaros, enquanto o Algoritmo Firefly foi inspirado pelo padrão piscante de vaga-lumes.

Os algoritmos metaheurísticos inspirados na natureza foram usados em uma ampla gama de problemas de otimização, incluindo problemas NP-difíceis, como o problema do caixeiro viajante (BONABEAU; DORIGO; THERAULAZ, 1999; BLUM; ROLI, 2003; YANG, 2010a; GOLDBERG; HOLLAND, 1988).

4.2 CUCKOO SEARCH VIA VÔOS DE LÉVY

Cuckoo Search (CS) é um algoritmo de pesquisa metaheurística. Este algoritmo baseia-se no comportamento parasita de algumas espécies de cuco em combinação com o comportamento dos vôos de Lévy de algumas aves e moscas. Esse algoritmo foi proposto por Yang e Deb (2009) e os estudos preliminares mostram que pode superar algoritmos muito utilizados como GA e PSO (GANDOMI; YANG; ALAVI, 2013).

Cucos são pássaros que possuem uma característica de reprodução que chama a atenção pela estratégia agressiva. Algumas espécies colocam seus ovos em ninhos de outras espécies e também muitas vezes podem remover os ovos dos outros para aumentar a probabilidade de incubação de seus próprios ovos (PAYNE; SORENSEN, 2005).

Se um pássaro hospedeiro descobre que os ovos não são seus, ele joga esses ovos intrusos fora ou abandona seu ninho e constrói um novo ninho em outro lugar.

Algumas espécies de cucos evoluíram de tal forma que as aves femininas são capazes de reproduzir as cores padrões dos ovos da espécie hospedeira. Isso reduz a probabilidade de os ovos serem abandonados e, assim, aumenta sua reprodutividade. Além disso, o momento da colocação de ovos de algumas espécies no ninho da ave hospedeira também é realizado no momento certo, pois geralmente os ovos de cuco reproduzem um pouco mais cedo do que os ovos do hospedeiro (GANDOMI; YANG; ALAVI, 2013).

Uma vez que o primeiro cuco é chocado, a primeira ação de instinto do filhote é expulsar os ovos do hospedeiro para fora do ninho, o que aumenta a quantidade de alimento fornecido a ele pelo pássaro hospedeiro. Estudos também mostram que um filhote de cuco também pode imitar o chamado de filhotes do pássaro que o acolheu para obter acesso a mais oportunidades de alimentação (PAYNE; SORENSEN, 2005) *apud* (YANG; DEB, 2009).

4.2.1 VÔOS DE LÉVY

Segundo Barthelemy, Bertolotti e Wiersma (2008), uma caminhada aleatória (ou passeio aleatório) é um processo estocástico em que partículas ou ondas viajam ao longo de trajetórias aleatórias. A primeira aplicação de uma caminhada aleatória foi na descrição do movimento das partículas em um fluido (movimento browniano).

O movimento browniano é provavelmente o mais conhecido de todos os passeios aleatórios. É o caso especial de uma caminhada aleatória para a qual os comprimentos dos passos são distribuídos de acordo com uma distribuição de probabilidade gaussiana.

Nos vôos de Lévy, os comprimentos dos passos são distribuídos de uma forma distinta, em que o padrão pode ser descrito resumidamente como aglomerados de pequenos passos separados por grandes saltos que diferem do padrão de movimentação.

Esse padrão contrasta visualmente com os padrões mais homogêneos do movimento browniano (BARTUMEUS et al., 2003). Eles são compostos por sequências aleatórias de segmentos de movimento (como voar, nadar ou andar), com comprimentos l , extraídos de uma função de distribuição de probabilidade $p(l) = l^{-\mu}$ onde $1 < \mu < 3$ (lei de potência).

Os vôos de Lévy são uma classe particular de caminhada aleatória em que os comprimentos dos passos durante a caminhada são descritos por uma distribuição de probabilidade conhecida como "de cauda pesada". Essa distribuição é dito ter uma cauda "pesada" porque os valores de grande comprimento são mais prevalentes do que em outras distribuições aleatórias como Poisson ou Gaussiana. As distribuições com valores mais extremos do que uma distribuição gaussiana são chamadas de distribuições "cauda pesada".

Vários estudos mostraram que o comportamento de deslocamento de muitos animais e insetos demonstrou as características típicas dos vôos de Lévy (BROWN; LIEBOVITCH; GLENDON, 2007; REYNOLDS; FRYE, 2007; PAVLYUKEVICH, 2007). Esses estudos demonstraram que os vôos de Lévy são padrões de pesquisa ótimos para caçadores, animais ou humanos, que procuram alvos escassos que são colocados aleatoriamente. A caça ocorre buscando-se bastante em lugares próximos, com saltos distantes para outros lugares.

A estratégia é ideal se o pesquisador estiver exclusivamente envolvido na busca, não tem conhecimento prévio dos locais de destino e se o espaçamento médio entre alvos sucessivos excede a faixa perceptual do pesquisador. Pesquisas ótimas ocorrem especificamente quando os comprimentos dos passos são distribuídos de acordo com uma lei de potência com um expoente de -2 , ou seja, quando $p(l) = l^{-2}$ (LUZ et al., 2001).

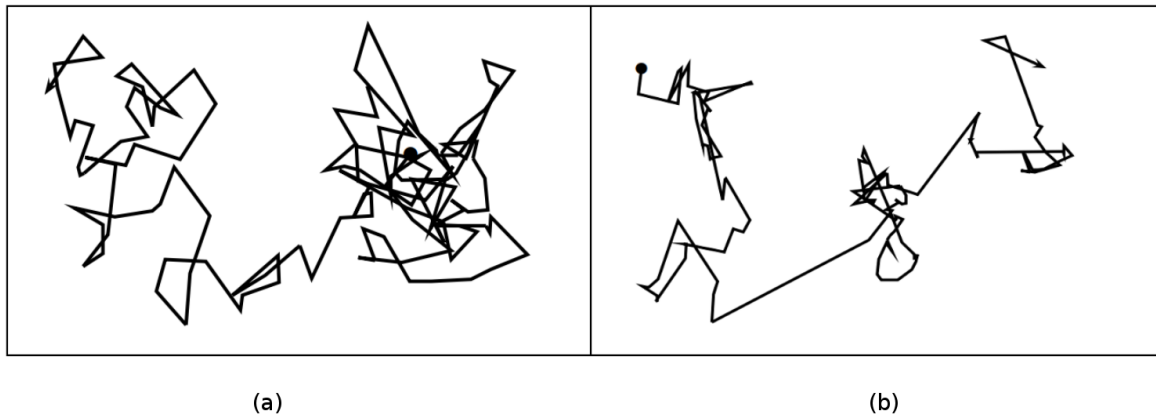
Segundo Viswanathan et al. (1996), na biologia, os vôos de Levy são observados, por exemplo, nos padrões comportamentais de uma gama diversificada de animais, incluindo veados, zangões, macacos, albatrozes e etc. Eles foram nomeados em homenagem ao matemático francês Paul Pierre Lévy.

Na figura 19 é mostrado um exemplo no padrão geral do movimento browniano comparado ao vôo de Lévy. A figura 19(a) apresenta o movimento browniano em 2D. É retratada uma caminhada aleatória com um tamanho de passo que segue uma distribuição Gaussiana. O caminho é percorrido com 50 passos partindo da origem (0, 0) (marcado com um ponto preto). A fig. 19(b) retrata um vôo de Lévy com 50 passos consecutivos a partir da origem (0, 0) (marcado com um ponto preto).

Localmente, parece um movimento browniano simples, mas, de tempos em tempos, há um grande salto e o processo (ou o animal) começa a explorar outra parte do plano.

Durante várias iterações, um vôo de Lévy será distribuído numa distância muito mais longa com relação à sua posição inicial do que uma caminhada aleatória Gaussiana

Figura 19 – (a): Movimento browniano. (b): Vôo de Lévy.



Fonte: Yang (2010c)

do mesmo comprimento (REYNOLDS; FRYE, 2007).

Do ponto de vista da implementação, a geração de números aleatórios com vôos Lévy consiste em duas etapas: a escolha de uma direção aleatória e a geração de etapas que obedecem à distribuição escolhida de Lévy.

A geração de uma direção deve ser desenhada a partir de uma distribuição uniforme, enquanto a geração de etapas é mais complexa. Há algumas maneiras de conseguir isso, mas uma das maneiras mais eficientes e diretas é usar o chamado algoritmo Mantegna (MANTEGNA, 1994) para uma distribuição de Lévy estável (YANG, 2010c).

4.2.2 FUNCIONAMENTO DO ALGORITMO CUCKO SEARCH

De acordo com Yang e Deb (2009), existem três regras no algoritmo CS. São elas:

1. Cada cuco coloca um ovo por vez e deposita em um ninho de uma ave hospedeira escolhido aleatoriamente;
2. Os melhores ninhos com alta qualidade de ovos (soluções) serão transferidos para as próximas gerações;
3. O número de ninhos hospedeiros disponíveis é fixo e o ovo colocado por um cuco pode ser descoberto pelo pássaro hospedeiro com uma probabilidade $pa \in [0, 1]$.

Manter as melhores soluções para as próximas iterações busca garantir a intensificação das buscas naquelas regiões mais promissoras.

Cada ovo em um ninho representa uma solução e um ovo de cuco representa uma nova solução. O objetivo é usar as novas e potencialmente melhores soluções para

substituir uma solução pior nos ninhos. Neste trabalho, cada ovo de cuco representa um vetor contendo valores, um para cada variável livre. A avaliação de cada solução, ou ovo de cuco, é realizada através da simulação elétrica utilizando os valores do vetor e a qualidade do ovo é dada de acordo com o valor da função custo calculada.

Caso o pássaro hospedeiro descubra a existência de um ovo intruso, pode destruir o ovo ou abandonar o ninho e construir um ninho completamente novo em outro lugar. Por simplicidade, esta última hipótese pode ser aproximada pela fração pa . Os ninhos que são substituídos recebem novas soluções aleatórias. Esta estratégia procura garantir a diversidade das soluções, pois uma nova solução é gerada aleatoriamente possivelmente em um lugar diferente da solução descartada.

Com base nestas três regras, os passos básicos do Cuckoo Search (CS) podem ser resumidos como o algoritmo 1 (YANG; DEB, 2009).

Ao gerar novas soluções $x(t + 1)$ para um cuco i , um vôo Lévy é realizado

$$X_i(t + 1) = X_i(t) + \alpha \otimes Levy(\lambda) \quad (4.2)$$

onde $\alpha > 0$ é o tamanho do passo que deve estar relacionado às escalas do problema de interesse e λ representa o expoente da lei de potência que define o tamanho dos passos no voo de Lévy. Segundo Yang (2010c), na maioria dos casos, pode-se usar $\alpha = 1$. A equação acima é essencialmente a equação estocástica para caminhada aleatória.

Em geral, uma caminhada aleatória depende apenas da localização atual (o primeiro termo na equação acima), e não da sequência de eventos que precederam, e a probabilidade de transição (o segundo termo).

O produto \otimes significa multiplicação realizada independentemente entre os vetores. Este produto é semelhante ao usado no PSO, mas a caminhada aleatória através do vôo Lévy é mais eficiente na exploração do espaço de pesquisa, pois seu comprimento de etapa é muito maior a longo prazo (YANG; DEB, 2009).

Algumas das novas soluções devem ser geradas em torno da melhor solução obtida até agora, isso intensifica a busca nas áreas mais promissoras. No entanto, uma fração das novas soluções deve ser gerada por randomização em um local distante o suficiente da melhor solução atual, isso assegurará que o sistema não seja preso em um ótimo local.

Segundo Yang e Deb (2009), existe uma aparente semelhança entre CS e o algoritmo subida de encosta em combinação com alguma randomização em grande escala. Mas existem algumas diferenças significativas. Em primeiro lugar, o CS é um algoritmo baseado na população, de forma semelhante ao GA e PSO, mas usa uma randomização diferente à medida que o comprimento do passo é "cauda-pesada", o que permite passos grandes.

Apesar da semelhança com alguns outros algoritmos, o CS possui um menor

número de parâmetros a serem determinados comparados ao PSO e GA, por exemplo.

Algoritmo 1: Cuckoo Search por Vôos de Lévy

```

Função Custo  $f(x)$ ,  $x = (x_1, \dots, x_d)^T$ ;
Gerar população inicial dos  $n$  ninhos hospedeiros  $X_i (i = 1, 2, \dots, n)$ ;
while ( $t < \text{MaxIterações}$ ) ou (critério de parada) do
    Obter novas soluções aleatoriamente por vôos de Levy; Avaliar a qualidade  $F_i$ ;
    Escolher um ninho entre  $n$  (digamos,  $j$ ) aleatoriamente;
    if  $F_i > F_j$  then
        | Substituir  $j$  pela nova solução;
    end
    Substituir uma fração ( $pa$ ) de piores ninhos;
    Manter as melhores soluções;
    Classificar as soluções e encontre a melhor atual;
end
mostrar resultados;

```

4.3 ALGORITMO FIREFLY

O algoritmo Firefly (FA) trata-se de um algoritmo metaheurístico estocástico, baseado no comportamento dos vagalumes.

Na natureza os vagalumes usam seu brilho para atrair possíveis parceiros, sendo essa atração mais forte quanto maior for a intensidade da luz. A ideia do algoritmo FA é calcular o valor da função objetivo em diversos pontos do domínio, escolhidos inicialmente de forma aleatória, considerando que cada um desses pontos é considerado um vagalume. Assim como no algoritmo CS, cada vagalume representa um vetor contendo um valor para cada uma das variáveis livres e possui um valor de função custo. A intensidade de luz desses vagalumes é relacionada com o valor da função custo calculada para o vetor de valores que cada vagalume corresponde (ponto no domínio).

Em seguida são feitas iterações, seguindo certas regras, com o objetivo de fazer com que os valores convirjam para o ponto que gere o maior brilho, obtido no ponto onde a função apresenta o valor ótimo (ou o melhor conhecido).

A atratividade de cada vagalume está ligada à intensidade da luz (I) que ele emite (que é proporcional à qualidade daquela solução). A intensidade da luz diminui com o aumento da distância (YANG, 2010b). A partir das informações dadas acima são formuladas algumas regras para o algoritmo:

1. Um vagalume pode ser atraído por qualquer outro.
2. A atratividade é proporcional ao brilho. O vagalume de menor brilho sempre irá se mover em direção ao de maior brilho.

3. A atratividade é proporcional à intensidade da luz, que é inversamente proporcional à distância, portanto a atratividade diminuirá quanto maior for a distância entre os vaga-lumes.
4. Quando não houver nenhum outro com brilho maior que o seu, o vaga-lume se moverá de forma aleatória.
5. O brilho de um vaga-lume é determinado pela forma da função. Por exemplo: em um problema de minimização o brilho será inversamente proporcional ao valor da função, pois o algoritmo tem o objetivo de encontrar o ponto que a intensidade da luz seja máxima.

A figura 20 ilustra de forma resumida o funcionamento do algoritmo com relação à atratividade que os vaga-lumes exercem uns sobre os outros. O vaga-lume com maior brilho (melhor solução) atrai os demais. No entanto, alguns vaga-lumes mais distantes são atraídos por algum vaga-lume com o brilho menor do que o de melhor solução, pois a distância reduz a atratividade do melhor vaga-lume.

Figura 20 – Ilustração do funcionamento do processo de atratividade dos vaga-lumes.



Fonte: Zhou et al. (2015)

4.3.1 ATRATIVIDADE

No FA existem dois pontos importantes: a intensidade da luz (ou brilho), que é um parâmetro individual de cada vaga-lume, e a atratividade, que depende da distância

que o vagalume está sendo observado. Apesar de parecidos, são conceitos diferentes. Pode-se assumir que a atratividade de um vagalume é proporcional ao seu brilho que, por sua vez, está associado à função a ser avaliada. A intensidade da luz de um vagalume, que pode ser escrita como $I(r)$, diminui quanto mais longe estiver a fonte geradora, portanto a atratividade entre dois vagalumes também diminuirá se a distância entre eles aumentar.

O decremento da intensidade obedece à lei do quadrado do inverso. Ou seja, a intensidade da luz I diminui à medida que a distância r aumenta em termos de $I = 1/r^2$. Além disso, o ar absorve a luz que se torna mais fraca à medida que a distância aumenta.

Considerando as reduções causadas pela distância e pela absorção do ar, tem-se a seguinte fórmula para a intensidade do brilho enxergada por um vagalume, em função da distância:

$$I(r) = I_0 e^{-\gamma r^2} \quad (4.3)$$

onde r é a distância entre os vagalumes, γ é o coeficiente de absorção de luz, que determina quanto a intensidade diminui com a distância e I_0 é a intensidade original da luz, em $r = 0$.

Como a atratividade de um vagalume é proporcional à luz percebida pelos vagalumes adjacentes, a atratividade (β) pode ser definida como

$$\beta(r) = \beta_0 e^{-\gamma r^2} \quad (4.4)$$

onde β_0 é a atratividade em $r = 0$.

4.3.2 DISTÂNCIA E MOVIMENTO

A distância entre dois vagalumes (i e j) é dada pela fórmula

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^N (x_{i,k} - x_{j,k})^2} \quad (4.5)$$

onde N é o número de dimensões. A equação do movimento de um vagalume i , atraído por outro vagalume j é dado por

$$x(t+1) = x(t) + \beta_0 e^{-\gamma r_{ij}^2} (x_j(t) - x_i(t)) + \alpha(\text{rand} - 1/2) \quad (4.6)$$

onde $x_i(t)$ é a posição inicial do vagalume i . $\beta_0 e^{-\gamma r_{ij}^2} (x_j(t) - x_i(t))$ é a parcela do movimento devido à atratividade gerada pelo vagalume j e $\alpha(\text{rand} - 1/2)$ é a parcela aleatória do movimento, onde rand é um número aleatório que pode variar entre 0 e 1.

O parâmetro γ é de extrema importância na determinação da velocidade de convergência do algoritmo. Teoricamente ele pode assumir qualquer valor positivo, mas na prática ele é determinado com base na distância característica do sistema a ser otimizado. Seu valor geralmente varia entre 0,01 e 100. (YANG, 2010b)

Existem dois casos específicos importantes: quando $\gamma \rightarrow \infty$ e quando $\gamma \rightarrow 0$. Para $\gamma \rightarrow 0$, a atratividade será sempre constante, o que seria equivalente a vagalumes espalhados num céu ideal, onde todos podem ser observados, de qualquer distância e, portanto, sejam sempre atraídos em direção ao que apresenta a maior intensidade de luz. Com $\gamma \rightarrow \infty$, tem-se uma situação completamente oposta: nenhum vagalume pode ser observado por outro, fazendo com que eles se movam de forma completamente aleatória. Este caso corresponde a um método de busca aleatória (RIBEIRO et al., 2014).

Como o parâmetro pode ser ajustado para que fique entre estes dois extremos, o Firefly pode encontrar ótimos globais e locais de maneira muito efetiva.

O pseudocódigo do FA, mostrado em Yang (2010b), pode ser visto no algoritmo 2.

Algoritmo 2: Algoritmo Firefly

```

Função Custo  $f(x)$ ,  $x = (x_1, \dots, x_d)^T$ ;
Gerar população inicial dos vagalumes  $X_i (i = 1, 2, \dots, n)$ ;
Determinar a intensidade de luz  $l_i$  em  $x_i$  com base em  $f(x_i)$ ;
Definir o coeficiente de absorção de luz  $\gamma$ ;
while ( $t < \text{MaxIterações}$ ) ou (critério de parada) do
    for  $i=1: n$  all  $n$  fireflies do
        for  $j=1: n$  all  $n$  fireflies do
            if  $I_j > I_i$  then
                | Mover o vagalume  $i$  para  $j$  na dimensão  $d$ ;
            end
            A atratividade varia com a distância  $r$  ;
            Avaliar a nova solução e atualizar a intensidade de luz;
        end
    end
    Classificar os vagalumes com base em  $f(x)$  e escolher o melhor;
end
Mostrar resultados;

```

4.4 PARTICLE SWARM OPTIMIZATION

O algoritmo de otimização por enxame de partículas (PSO) é uma técnica evolutiva, inspirada no comportamento social dos pássaros (BARROS; GUILHERME; HORTA, 2010). É uma heurística de otimização global e local de sistemas não-lineares. Ela usa uma população de indivíduos, referidos como partículas, que se movem através do espaço de busca com uma determinada velocidade em direção à partícula com a melhor posição (KROHLING, 2004).

As posições iniciais das partículas são definidas de forma aleatória e cada

partícula mantém a informação da melhor solução que encontrou até o momento. Esse valor é chamado $Pbest$. O algoritmo também armazena informações sobre a melhor solução global e sua localização. Esse valor é o melhor valor obtido entre todas as partículas e é chamado de $Gbest$.

A cada iteração, a posição de cada partícula é atualizada em direção a seu $Pbest$ e ao $Gbest$. As novas posições são determinadas de acordo com uma função de aptidão pré definida. As partículas movem-se em direção à solução ótima ou à melhor solução encontrada até o momento. Esse movimento ocorre com uma determinada velocidade e a cada iteração é realizada uma avaliação da função custo correspondente à posição ocupada. O pseudocódigo do PSO pode ser visto no algoritmo 3 (BARROS; GUILHERME; HORTA, 2010).

Desta maneira, o PSO é capaz de refinar boas soluções, pois o enxame, inicialmente disperso pelo espaço de busca, tende a focar em determinadas regiões que apresentam resultados mais propícios para a solução esperada, de maneira que, ao decorrer do tempo, este enxame alcança um ponto de alinhamento. No entanto, existe um diferente equilíbrio entre as capacidades de busca local e global, fazendo com que o algoritmo possa não encontrar resultados satisfatórios. Em Shi e Eberhart (1998) os autores adicionaram um parâmetro de inércia w na equação do cálculo da velocidade, com o intuito de equilibrar a capacidade entre a procura global e a busca local. Este parâmetro de inércia foi definido principalmente como uma constante positiva. A cada iteração $t+1$, cada partícula i atualiza sua velocidade v conforme a seguir:

$$V_i(t + 1) = v_i(t) + C_1 * rand_1 * (Pbest_i - P_i(t)) + C_2 * rand_2 * (Gbest - P_i(t)) \quad (4.7)$$

onde C_1 e C_2 são as constantes de aceleração, $P_i(t)$ é a posição atual da partícula e $Pbest$ e $Gbest$ representam a melhor solução da partícula i e a melhor solução global, respectivamente. A posição P da partícula é atualizada da seguinte forma:

$$P_i(t + 1) = P_i(t) + V_i(t) \quad (4.8)$$

A tendência é que as partículas converjam para um ponto comum que representa

o ponto ótimo no processo de otimização.

Algoritmo 3: Particle Swarm Optimization

```
Função Custo  $f(x)$ ,  $x = (x_1, \dots, x_d)^T$ ;  
Gerar população inicial das  $n$  partículas  $X_i (i = 1, 2, \dots, n)$ ;  
Gerar valor para a velocidade;  
Obter os valores iniciais para  $Pbest(x_i)$  e  $Gbest$ ;  
while ( $t < MaxIterações$ ) ou (critério de parada) do  
    Determinar o valor de  $f(x_i)$  para cada partícula;  
    Atualizar  $Pbest(x_i)$  e  $Gbest$ ;  
    Calcular a velocidade para cada partícula;  
    Atualizar a posição de cada partícula;  
end  
Mostrar resultados;
```

4.5 CRITÉRIO DE PARADA

Segundo Zielinski e Laur (2007), para os aspectos teóricos dos algoritmos evolutivos, os critérios de parada geralmente não são importantes. No entanto, para aplicações práticas, a escolha dos critérios de parada influencia no tempo de execução da otimização.

Devido a diferentes critérios de parada, a otimização pode ser encerrada antes da convergência da população, ou os recursos computacionais podem ser desperdiçados caso a execução da otimização seja encerrada de forma tardia.

Para Bergh (2007), embora seja relativamente simples projetar um critério de parada para um algoritmo de pesquisa local, é bastante difícil de fazer para um algoritmo de busca global, a menos que o valor da função objetivo no minimizador global seja conhecido antecipadamente.

Na maioria dos trabalhos da literatura são utilizados um dos critérios de paradas seguintes (JAFARI et al., 2012; MALLICK et al., 2017; SEVERO et al., 2012; VURAL, 2012):

- Número de iterações pré-estabelecido pelo usuário ou tempo máximo de execução é atingido;
- Resultado considerado satisfatório é atingido;

Em Bergh (2007) é citado que a convergência de um algoritmo baseado em enxame pode ser detectada calculando um raio máximo entre as partículas do enxame, fazendo uma análise do agrupamento das mesmas ou calculando a taxa de mudança na função objetivo.

Em Babu e Angira (2003) o critério de parada adotado é encerrar o processo de busca quando uma das seguintes condições for satisfeita:

- o número máximo de iterações é atingido.
- $|f_{max}^k - f_{min}^k| < v$ onde f é o valor da função custo para a k -ésima iteração e v é um valor determinado como diferença mínima para o algoritmo continuar .

Os critérios baseados em melhoria terminam uma execução de otimização se apenas pequenas melhorias forem feitas, porque, geralmente, no início de uma operação de otimização grandes melhorias são alcançadas, enquanto em etapas posteriores a melhoria se torna pequena.

Em Zielinski e Laur (2007) são apresentados alguns diferentes critérios parada. Os primeiros são baseados em melhorias na função custo. São denominados pelo autor de *ImpBest* e *ImpAvg*:

- *ImpBest*: é monitorada a melhoria do melhor valor da função custo. Se cair abaixo de um determinado limite t a execução de otimização é encerrada.
- *ImpAv*: semelhante ao *ImpBest*, mas em vez de observar o melhor valor da função custo, o valor médio calculado de toda a população é verificado.

Outra possibilidade de parada do algoritmo é utilizar como base a distribuição dos agentes. Esses critérios consideram a diversidade da população. Se a diversidade é baixa, os indivíduos estão próximos um do outro, então assume-se que a convergência foi obtida.

Outros critérios de parada, segundo Zielinski e Laur (2007), são:

- *MaxDist*: a distância entre cada membro da população e o melhor indivíduo é observada. A execução de otimização é interrompida se a distância máxima for inferior a um limiar m .
- *MaxDistQuick*: É uma generalização do *MaxDist*. Em vez de usar toda a população para o cálculo da distância máxima para o melhor membro da população, um algoritmo de *quicksort* é empregado para classificar os indivíduos devido ao seu valor de função objetivo, e apenas os melhores p indivíduos são considerados.

A justificativa para esse último critério é que existem execuções onde a maioria da população convergiu para o melhor, mas por causa dos demais indivíduos que ainda estão explorando o espaço, a otimização não é interrompida, embora não contribuam com novas informações.

A utilização de um número máximo de iterações como critério de parada é adequado quando se deseja comparar diferentes algoritmos, por exemplo. Muitas vezes

não se tem conhecimento sobre o valor ótimo de um problema para interromper a execução quando se atinja um valor considerado bom. Assim, torna-se mais vantajoso utilizar um critério de parada que considera o conhecimento sobre o comportamento da função custo até o momento na execução.

Neste trabalho foram implementados dois critérios de parada, além do número máximo de iterações. As implementações foram realizadas com base no descrito:

- O melhor valor de função custo em cada iteração é comparado com o melhor valor da função custo na iteração anterior. Se o valor de variação v entre as duas iterações for menor que um erro e , um contador $cont$ é incrementado. Caso o valor de $cont$ atinja um número pré-determinado n , a execução é interrompida.
- O terceiro critério de parada implementado é semelhante ao anterior, mas utiliza o conceito de janela móvel. O melhor valor de função custo em uma iteração t é comparado com o melhor valor da função custo na iteração $t - h$, onde h é o tamanho da janela a ser considerada. Se a diferença entre os valores for menor que um limiar e , a execução do algoritmo é interrompida.

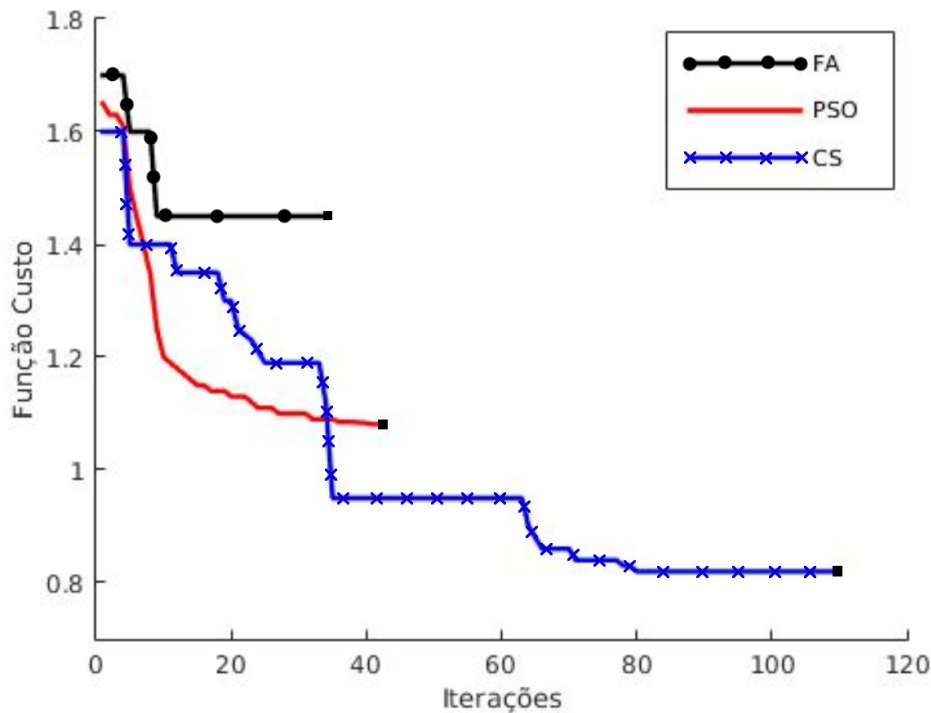
A diferença entre estes dois últimos basicamente é que no primeiro compara-se os resultados de duas iterações subsequentes e permite-se que a diferença entre os valores comparados sejam menores que o erro por n vezes até interromper o algoritmo. Para as demais execuções nesse trabalho foi adotada a segunda estratégia, que utiliza a janela móvel. Os critérios de parada baseados em distribuição dos agentes possuem implementação mais complexa e requerem um número maior de computação para cálculo das distâncias.

A figura 21 demonstra um exemplo da evolução da função custo em uma execução dos três algoritmos descritos no trabalho, utilizando critério de parada com tamanho de janela $h=30$ e limiar de erro definido como 5% do valor na iteração $t - h$. Os algoritmos foram executados utilizando o mesmo tamanho da população ($Pop=50$). Essas execuções foram realizadas utilizando o circuito OTA Miller.

Como pode-se observar, a utilização desse critério de parada faz com que os três algoritmos convirjam de forma diferente e parem a execução em iterações distintas. Em diferentes execuções dos algoritmos o número de iterações e o valor da função custo ao final da execução tende a variar. Em uma execução com os mesmo valores iniciais para as variáveis, mas utilizando um tamanho de janela $h = 20$, por exemplo, faria com que a execução encerrasse antes, o que não permitiria que se chegasse a um resultado melhor, como mostra a figura 22.

Apesar do aspecto aleatório dos algoritmos, percebe-se que o critério de parada tem grande influência na obtenção dos resultados finais. O algoritmo CS apresentou maior capacidade de escapar de mínimos locais, fato que pode ser observado pela diminuição do valor da função custo nas iterações próximas a iteração 65.

Figura 21 – Evolução da função custo para o FA, PSO e CS utilizando $h=30$.



Fonte: Autor.

Já o algoritmo FA finalizou precocemente e não conseguiu fugir de um mínimo local, fornecendo uma solução final bem pior que os demais algoritmos testados.

4.6 ESTUDO SOBRE OS VALORES INICIAIS

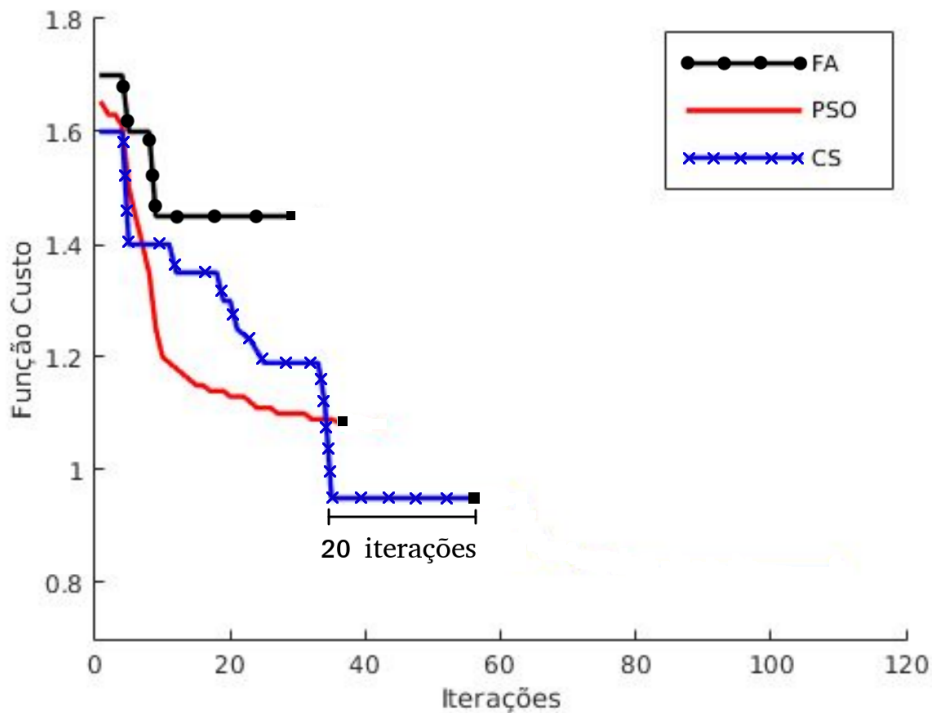
Na maioria dos algoritmos iterativos utilizados em otimização a função custo obtida no final do algoritmo varia de acordo com os valores assumidos como ponto de partida.

Segundo Stehr et al. (2003), valores iniciais para as variáveis podem ser determinados de forma aleatória, com base na intuição dos projetistas ou em equações que requerem uma elaboração complexa.

Nas heurísticas utilizadas neste trabalho, o dimensionamento inicial não deve necessitar de intervenção do usuário e deve ser independente do conhecimento do projetista. Para isso, decidiu-se observar se as diferentes formas de inicialização das variáveis livres afetam a função custo obtida no final da otimização e o quanto esses valores iniciais interferem no resultado final.

Segundo alguns autores, em muitos problemas a convergência do algoritmo melhora quando o espaço de busca é examinado em uma etapa separada antes que se realize a otimização (STEHR et al., 2003; KROHLING, 2004).

A maneira mais tradicional de definir os valores iniciais das variáveis e que a

Figura 22 – Representação da mesma execução da figura 21 mas com $h=20$.

Fonte: Autor.

maioria dos autores utilizam é distribuição aleatória em todo o espaço de busca. De acordo com esses trabalhos, o algoritmo deve ser capaz de convergir rapidamente para uma área de pesquisa promissora, mesmo começando aleatoriamente (KROHLING, 2004).

Para fazer isso, inicia-se os valores aleatoriamente entre os limites de intervalo permitido para cada variável, como exemplifica a equação 4.9.

$$\text{valor}_{inicial}(i) = Pos_{min} + (Pos_{max} - Pos_{min}) * rand \quad (4.9)$$

De acordo com Stehr et al. (2003), a inicialização dos valores mais próximos do centro do espaço de busca aumenta a chance de encontrar as melhores soluções nos algoritmos que utilizam um único agente explorando o espaço de busca. Obtém-se um ganho no desempenho do algoritmo, pois, em geral, pode-se observar que o centro do espaço de busca é o ponto mais próximo de todos os outros pontos. De acordo com o autor, esta estratégia pode ser utilizada em vários tipos de algoritmos de otimização.

Nos algoritmos baseados em população, de acordo com (TALUKDER, 2011), a diversidade inicial do enxame é importante para o desempenho, pois leva a uma melhor cobertura do espaço de busca.

Além disso, quando o enxame inicial não abrange todo o espaço de busca, os algoritmos podem não conseguir atingir os melhores resultados, uma vez que os melhores valores para as variáveis podem estar localizados fora da área coberta pelas

partículas ou indivíduos.

Para demonstrar isso, limitou-se a área de inicialização a 1/3 e 1/8 central do intervalo permitido para cada variável do amp-op Miller, a fim de observar a influência no desempenho do algoritmo PSO. As variáveis desse circuito são W1 e L1 (gate dos transistores M1 e M2), W3 e L3 (gate dos transistores M3 e M4), W5 e L5 (gate dos transistores M5 e M8), W6 e L6 (gate do transistor M6), W7 e L7 (gate do transistor M7), corrente I_{ref} e a capacitância C_f . Na Tabela 2 pode-se observar os valores para a função custo obtidos após 50 execuções de cada estratégia.

Tabela 2 – Resultados para as 3 diferentes formas de inicialização das variáveis livres.

Valor	Distribuição		
	Todo Espaço de Busca	1/3 Central	1/8 Central
Média da Função Custo	0.8620	0.9336	0.9630
Melhor função Custo	0.5150	0.6111	0.6981

Como pode-se observar, a estratégia em que os valores iniciais das partículas estão espalhadas por todo o espaço de busca apresentou melhores resultados tanto na média quanto no melhor valor para a função custo a ser minimizada. Isso deixa claro que em algoritmos baseados em população devem ter os indivíduos espalhados por todo espaço de busca.

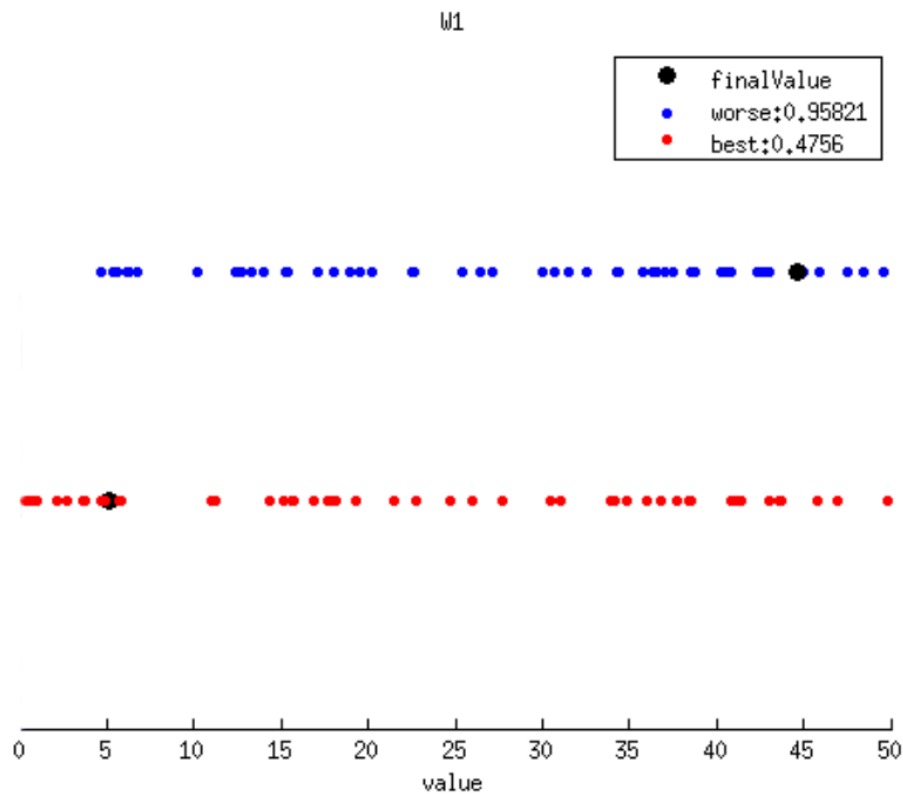
Após isso, foi observada a relação dos valores iniciais com a função custo final neste problema. Para isso, executou-se novamente 100 vezes o processo de otimização espalhando as partículas (ou agentes da população) de forma aleatória em todo o espaço de busca e armazenando os valores iniciais de cada uma das variáveis, isto é, os valores gerados de forma aleatória antes da primeira iteração do processo de otimização.

Após as 100 execuções, comparou-se os valores iniciais da melhor e da pior solução entre as 100 execuções a fim de observar a existência de um padrão na inicialização que levou a um melhor resultado. Para isso, foram observadas as distâncias entre cada uma das partículas iniciais de cada variável livre para o valor final para aquela variável ao final da otimização (valor daquela variável que levou a um melhor resultado). A ideia é observar se na melhor solução as partículas iniciaram em um intervalo mais próximo ao seu respectivo valor no final da execução, para assim, em cada nova execução, iniciar as partículas considerando um histórico de locais que levaram a melhores soluções.

Na figura 23 é mostrada a distribuição de cada uma das partículas para a variável W1, que representa o w do transistor M1 da figura 6, no momento da inicialização para o melhor caso, em vermelho, e o pior resultado, em azul. O ponto preto demonstra o valor final para a variável, em ambos os casos.

Observa-se na figura que não existe grande diferença na distribuição de par-

Figura 23 – Distribuição das partículas iniciais para W1



Fonte: Autor.

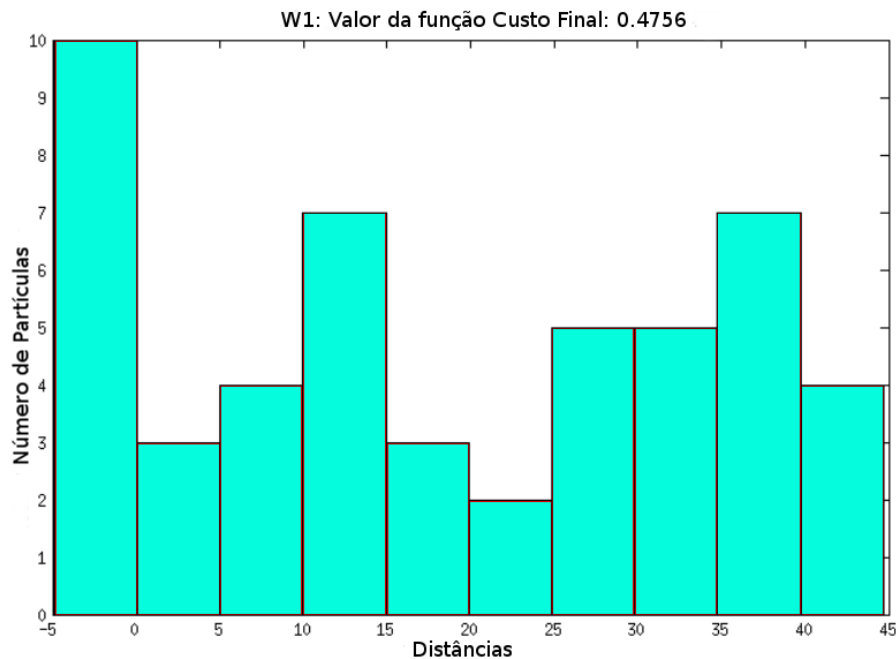
tículas entre o melhor e o pior resultado. Além disso, não há uma grande diferença no número de partículas iniciadas em um raio próximo do valor final (10% do espaço possível) entre o melhor e o pior caso.

Para observar a relação entre as partículas e o resultado final para cada variável também foi criado o histograma entre as distâncias e o resultado final. Para isso, dividiu-se o espaço de busca de cada variável em 10 partes. Um exemplo é mostrado nas Fig. 21 e 22 que representam a variável W1 na melhor e a pior solução, respectivamente. Pode-se observar que, para o melhor caso (figura 24), existem 13 partículas que começaram em raio próximo (5μ para mais e 5μ para menos) do resultado final, pois existem 10 partículas para a esquerda e 3 partículas para a direita. Os números negativos foram utilizados para diferenciar a distância para a esquerda e a distância para a direita entre os valores iniciais e o valor final para a variável. Para o pior caso (figura 25), 10 partículas começaram no mesmo raio da solução final.

Assim como o demonstrado para a variável W1, foi criado o histograma de distâncias para cada uma das 12 variáveis livres do OTA Miller. Com base nisso, observou-se que apenas em 7 casos a melhor solução apresentou mais variáveis iniciadas no raio de 10% do valor final.

A partir dos resultados obtidos, pode-se constatar que a inicialização de forma

Figura 24 – Distância das posições iniciais das partículas para o melhor valor final de W1.



Fonte: Autor.

aleatória em todo o espaço de busca leva a melhores resultados, mas não garante resultados ótimos, uma vez que utilizou-se heurísticas não-determinísticas.

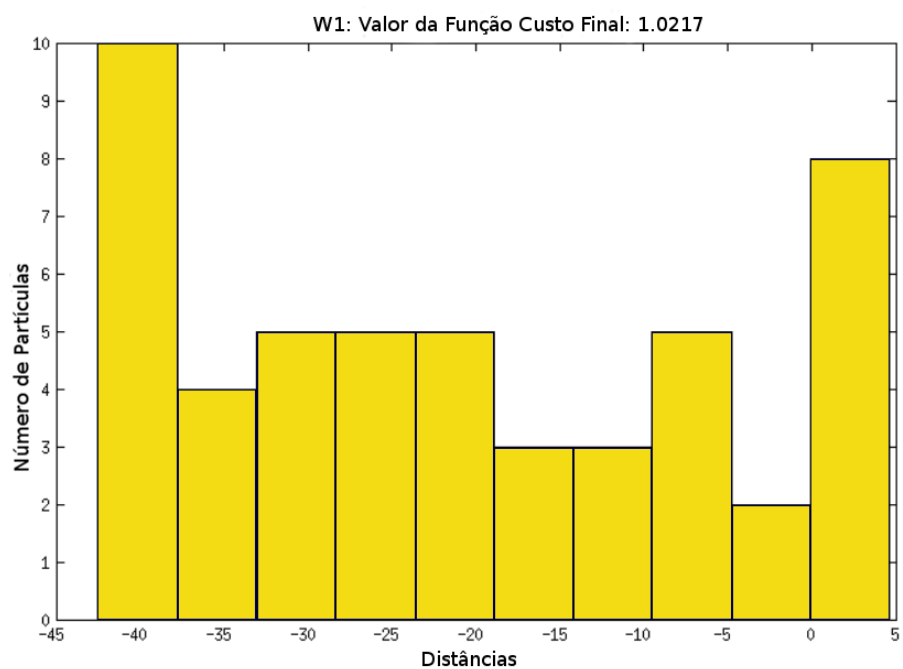
4.7 CONCLUSÃO

Neste capítulo foram apresentados alguns métodos de otimização para diferentes tipos de problemas. Existe diferentes tipos de métodos de otimização, geralmente divididos em determinísticos e estocásticos. Dentre os algoritmos estocásticos existem diversos métodos, alguns baseados em um único agente e outros baseados em população ou bio-inspirados.

Como neste trabalho a otimização a ser realizada é não-linear e apresenta um grande número de variáveis, foram apresentadas as metaheurísticas bio-inspiradas CS, FA e PSO que podem ser utilizadas para o dimensionamento de CIs analógicos. Essas metaheurísticas são baseadas em diversos agentes explorando o espaço de busca. O funcionamento dos algoritmos FA e PSO é bastante parecido. Além disso, foi apresentado um estudo sobre a inicialização dos valores para cada variável livre onde é possível constatar que inicializar as variáveis de forma aleatória em todo o espaço de busca apresenta melhores resultados do que limitando a área de inicialização.

Foi apresentado também um estudo sobre critérios de parada que podem ser utilizados pelas heurísticas e, por fim, foi realizado um estudo sobre os valores iniciais para as variáveis a serem ajustadas pela heurística.

Figura 25 – Distância das posições iniciais das partículas para o pior valor final de W1.



Fonte: Autor.

5 RESULTADOS

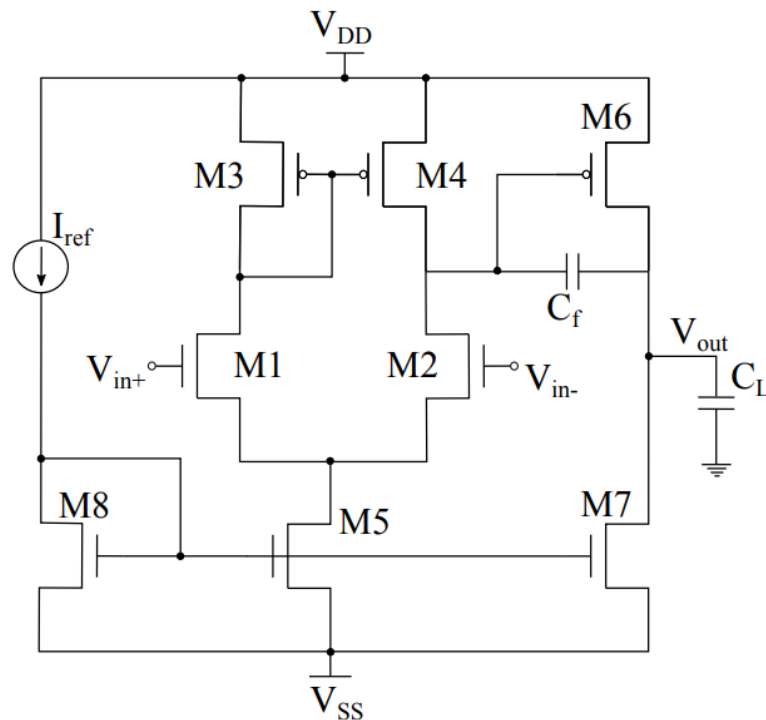
Neste capítulo são apresentados os três amplificadores dimensionados neste trabalho. São demonstrados também os resultados obtidos para as especificações de cada um. Além disso, são realizadas comparações com os resultados de alguns outros trabalhos em que os mesmos circuitos são dimensionados.

Foram utilizados os algoritmos FA, CS e PSO para o dimensionamento do circuito OTA Miller para comparação entre os resultados obtidos. Após isso, realizou-se o dimensionamento de outros dois amplificadores. As próximas seções apresentam a estrutura desses circuitos e como eles são transformados em um problema de otimização.

5.1 AMPLIFICADOR OPERACIONAL DE TRANSCONDUTÂNCIA DO TIPO MILLER

Os amplificadores operacionais de transcondutância (OTAs) usualmente apresentam alta impedância de saída, podendo funcionar como uma fonte de corrente controlada por tensão, daí o nome transcondutância dado a esse tipo de amp-op (RAZAVI, 2001). Um tipo de amplificador operacional de transcondutância é o Miller. Ele é composto de dois estágios de ganho com um capacitor de compensação Miller. Uma implementação deste circuito em tecnologia CMOS é mostrada na figura 26.

Figura 26 – Esquemático de um OTA Miller em tecnologia CMOS



Fonte: Adaptado de Severo (2012)

O primeiro estágio desse circuito é um amplificador diferencial e o segundo

estágio é um amplificador inversor. A ligação entre esses estágios é realizada através de um capacitor de acoplamento com o intuito de dar mais estabilidade ao amplificador (HOLBERG; ALLEN, 2002). Esse capacitor é a chamada compensação Miller.

As principais características desses circuitos são o ganho em baixas frequências (Avo), o produto ganho-largura de faixa (GBW), a margem de fase (PM), a velocidade de resposta (SR), a faixa de tensão de entrada em modo comum ($ICMR$) e a faixa de tensão de saída (OS).

O circuito OTA Miller é composto por 12 variáveis livres, representando a largura (W) e o comprimento (L) do gate de 5 transistores, bem como a fonte de corrente I_{bias} e uma capacitância de acoplamento C_f . A tecnologia utilizada para simulação é TSMC 0.18 μm com uma tensão de alimentação de 1.8 V. Os valores de W variam de 0.22 μm a 50 μm e de L de 0.18 μm a 10 μm . Os valores para C_f podem variar de 0.1 pF a 10 pF e I_{bias} de 0.1 μA a 100 μA . Esses intervalos de valores são definidos pela tecnologia de fabricação. As variáveis são denominadas $W1$, $L1$ (transistores $M1$ e $M2$), $W3$, $L3$ (transistores $M3$ e $M4$), $W5$, $L5$ (transistores $M5$ e $M8$), $W6$, $L6$ (transistor $M6$), $W7$, $L7$ (transistor $M7$), I_{bias} e C_c .

Os algoritmos foram calibrados a fim de encontrar os melhores valores para os seus respectivos parâmetros. Para o PSO foram utilizados os parâmetros definidos no trabalho de Domanski (2016), onde foi realizado um estudo para o dimensionamento do OTA Miller. Para a calibragem dos outros algoritmos foi utilizado o mesmo circuito. Os valores definidos para cada parâmetro dos algoritmos na calibragem foram replicados para o dimensionamento dos demais circuitos do trabalho. Isso se dá pelo fato de que a calibragem é realizada neste trabalho para que se tenha noção da grandeza que deve ser utilizada em cada parâmetro, uma vez que, apesar dos circuitos divergirem no número de variáveis livres, a grandeza dos valores dessas variáveis é a mesma. Além disso, deseja-se que não seja necessária uma calibragem para cada vez que a topologia de circuito a ser dimensionado seja alterada.

5.1.1 CALIBRAGEM DOS PARÂMETROS NO CS

A fim de estabelecer valores para os parâmetros do algoritmo, primeiramente foi fixado o número de iterações do algoritmo em 50 e foi variado o número de ninhos de 10 a 60 com variação de 10 ninhos. Esse intervalo foi definido de forma empírica, pois o aumento do número de ninhos impacta no tempo de execução e muitas vezes já não garante uma melhora significativa nos resultados obtidos. Para cada número de ninhos diferente foram realizadas 10 execuções. A Tabela 3 demonstra o valor médio obtido para cada número diferente de ninhos.

É importante salientar que o aumento no número de ninhos leva a uma cobertura maior do espaço de busca mas também a um aumento no tempo total de execução do algoritmo. No entanto, como pode-se observar na Tabela 3, o aumento de 50 para

Tabela 3 – Valor médio da função custo variando o número de ninhos.

número de ninhos	função custo média
10	1.112
20	1.098
30	0.932
40	0.847
50	0.801
60	0.832

60 ninhos não acarretou em uma melhoria na média da função custo, uma vez que deseja-se minimizar este valor.

O algoritmo também foi executado sendo variado o parâmetro pa , que diz respeito a probabilidade de uma ave hospedeira descobrir um ovo intruso e descartá-lo. Essa probabilidade foi variada entre 15% e 70%. Para cada valor de pa o algoritmo foi executado 10 vezes com 50 iterações em cada execução e 50 ninhos. A Tabela 4 mostra o valor médio da função custo nas 10 execuções para cada valor de pa (onde 0 = 0% e 1 = 100%).

Tabela 4 – Valor médio da função custo variando o parâmetro pa

valor de pa	função custo média
0.15	1.087
0.25	0.843
0.40	0.921
0.50	1.044
0.70	1.225

Como pode ser observado na Tabela 4, o valor que se mostrou com melhor média entre os resultados foi o de pa igual a 0.25. Um valor muito grande de pa leva a uma taxa grande de descartes dos ninhos, o que faz muitas vezes com que boas soluções sejam descartadas. Um valor muito pequeno faz com que aconteçam muito poucos descartes, o que pode levar o algoritmo a explorar menos o espaço de soluções.

5.1.2 CALIBRAGEM DOS PARÂMETROS NO FA

Em relação ao algoritmo CS, o FA possui um maior número de parâmetros a serem definidos. Para observar o seu comportamento no dimensionamento do circuito OTA Miller com diferentes valores, realizou-se algumas execuções.

Primeiramente variou-se o valor do α , que está ligado à parcela aleatória do movimento. Esse valor pode ir de zero à 1, segundo Yang (2010b). Os valores testados foram $\alpha = 0.3$, $\alpha = 0.5$ e $\alpha = 0.7$ e $\alpha = 0.9$. Foram realizadas 10 execuções para cada um dos valores com um número de iterações igual a 50 e 50 vagalumes. Os resultados são mostrados na Tabela 5.

Tabela 5 – Resultados variando α

valor de α	função custo média
0.3	1.801
0.5	1.594
0.7	1.457
0.9	1.683

Observou-se que um número baixo para α leva o algoritmo a convergir mais rápido para o melhor resultado e já estagnar o valor para a melhor função custo nas primeiras iterações. Já com um α muito grande, o movimento tende a ser muito aleatório, o que prejudica o processo de intensificação nas melhores áreas. Esse valor representa o valor inicial para o α , em cada iteração esse valor tende a diminuir de forma à chegar a zero nas iterações finais da execução.

O valor de γ deve levar em consideração as dimensões do problema considerando-se a maior distância possível entre os vagalumes (soluções). Realizou-se execuções variando o valor de γ de 0 a 10 (com variações de 1), e, após, de 0.1 a 2 (com variações de 0.2) para um ajuste fino. As execuções tiveram 50 iterações, 50 vagalumes e $\alpha = 0.25$. Dessa forma, constatou-se que o valor de γ que apresentou melhor resultado foi $\gamma = 1.5$.

Segundo Yang (2010b) o valor de β na maioria das aplicações deve ser zero, mas em alguns problemas atinge-se melhores soluções se esse valor for maior que zero. Os valores de β simulados foram de 0 a 0.3, com variação de 0.1 e novamente com 50 iterações e 50 vagalumes, além de $\gamma = 1.5$ e $\alpha = 0.7$. Para cada valor de β realizou-se 10 execuções. A Tabela 6 mostra os resultados.

Também foi variado o número de vagalumes, de 10 a 60. O número de iterações foi fixado em 50 e utilizando $\alpha = 0.7$, $\gamma = 1.5$ e $\beta = 0.2$. Para cada número diferente de vagalumes o algoritmo foi executado 10 vezes. Os resultados médios são mostrados na Tabela 7.

Tabela 6 – Resultados variando β

valor de β	função custo média
0	1.521
0.1	1.710
0.2	1.406
0.3	1.801

Tabela 7 – Valor médio da função custo variando o número de vagalumes

número de vagalumes	função custo média
10	2.732
20	2.521
30	1.871
40	1.551
50	1.498
60	1.369

5.1.3 COMPARAÇÃO ENTRE OS ALGORITMOS

Para analisarmos melhor o critério de parada adotado neste trabalho, foi realizado um conjunto de testes com 10 execuções para cada algoritmo. Em cada execução foram utilizados os mesmos valores iniciais (mesmo vetor de números aleatórios) para as três heurísticas, assim garantindo uma comparação justa. Todos os testes foram realizados para o mesmo amplificador OTA Miller descrito na seção 5.1. Na Tabela 8 são demonstrados os valores da função custo e o número de iterações em cada execução.

O algoritmo FA encerrou a execução em um número menor de iterações em 8 dos casos. No entanto, ele apresentou um valor médio de função custo acima dos outros dois algoritmos. Isto leva a observar que ele converge de forma mais rápida, o que faz com que os agentes, ou vagalumes, se agrupem de forma mais rápida ao redor do que apresenta melhores resultados no início da execução, prejudicando uma exploração mais ampla do espaço de busca e caindo em mínimos locais.

Considerando o valor da função custo, pode-se observar que em 6 casos o algoritmo PSO apresentou melhor resultado, enquanto o CS se mostrou melhor em 4 casos. Comparando apenas o CS e o PSO, pode ser observado que o PSO converge mais rapidamente, o que leva ao fim da execução de forma mais rápida e com um valor

Tabela 8 – Número de iterações e valor da função custo obtida.

Execução	FA		PSO		CS	
	Num. de Iter.	f(x)	Num. de Iter.	f(x)	Num. de Iter.	f(x)
1	34	1.473	42	1.087	109	0.822
2	33	1.314	87	0.914	52	1.005
3	78	1.114	42	0.645	98	0.698
4	72	1.315	48	0.883	140	0.623
5	36	1.152	49	0.662	78	0.841
6	37	1.211	52	0.897	41	0.855
7	48	1.185	64	0.783	89	0.954
8	69	1.243	91	0.782	77	0.850
9	41	1.132	49	1.112	64	1.108
10	52	1.201	82	0.745	104	0.997
Média	44.5	1.206	50.5	0.833	83.5	0.855
Desvio Padrão	16.21	0.10	18.12	0.16	27.79	0.13

médio melhor. Pode-se observar também, através do desvio padrão, que o algoritmo FA apresenta valores mais próximos à média, tanto na função custo quando no número de iterações. Já o algoritmo CS apresentou o maior desvio padrão para o número de iterações enquanto o PSO apresentou um desvio padrão maior para a função custo.

No trabalho de Barra et al. (2012) também é realizado o dimensionamento de um amp-op Miller de dois estágios em tecnologia $0.18 \mu\text{m}$ e os resultados são obtidos através de uma simulação SPICE no software CADENCE Spectre. A partir dos valores obtidos para as variáveis livres pelos autores, realizou-se a uma nova simulação no software HSPICE no presente trabalho para comparação. Na Tabela 9 são demonstrados os valores atingidos para as especificações no melhor resultado de cada algoritmo e também os obtidos a partir dos valores de Barra et al. (2012).

Tabela 9 – Especificações para os melhores resultados de cada algoritmo de otimização avaliado.

Especificação	Valor Requerido	FA	PSO	CS	Barra et al. (2012)
Av0 (dB)	≥ 70.00	55.08	70.93	71.01	56.76
GBW (MHz)	≥ 5.00	5.24	5.83	5.00	1.30
MF (°)	≥ 60.00	65.20	60.00	66.57	76.24
SR (V/ μs)	≥ 5.00	6.73	7.14	7.80	3.26
ICMR+ (V)	≥ 0.70	0.63	0.74	0.72	0.81
ICMR- (V)	≤ -0.70	-0.68	-0.80	-0.86	-0.69
OS+ (V)	≥ 1.00	0.77	0.83	0.84	0.83
OS- (V)	≤ -1.00	-0.83	-0.84	-0.77	-0.83
Pdiss (μW)	minimizar	152.53	49.74	58.13	120.56
Área do gate (μm^2)	minimizar	16.21	14.32	11.23	13.21
Tempo Exec. (min)	-	19.08	28.50	94.41	-

Em termos de desempenho do circuito, o amplificador OTA Miller projetado com a heurística FA não atingiu os requisitos mínimos de Av_0 , ICMR e OS. Também foi o circuito que apresentou a maior dissipação de potência e maior área de gate dos três projetos. Apesar de os projetos utilizando PSO e CS também não terem alcançado os requisitos mínimos de OS + e OS -, os circuitos tiveram um desempenho geral melhor. Com o PSO, o amplificador foi melhor otimizado em termos de dissipação de potência, enquanto a heurística CS levou a um projeto com menor área de gate.

O projeto considerando a heurística CS apresentou um circuito com um desempenho bem próximo do obtido com a heurística PSO. Em termos de otimização, a dissipação de potência ficou em torno de 12% maior, enquanto que a área de gate ficou em torno de 22% menor. A grande diferença de desempenho entre as heurísticas de otimização PSO e CS, as quais apresentaram os melhores resultados, está no tempo de execução. O projeto com a heurística CS demandou um tempo de execução de aproximadamente 1 hora e 34 minutos em um PC Intel Core i7-3770 de 3.40GHz e 8GB de memória principal, o que é mais que o triplo do tempo de execução do projeto com a heurística PSO. Isto pode ser explicado pelo fato de que o cálculo dos movimentos por voos de Lévy é computacionalmente mais intenso. Além disso, a convergência para a solução ótima é mais lenta, o que faz com que o algoritmo necessite de um número maior de iterações.

O principal objetivo no projeto automático de Barra et al. (2012) é reduzir a potência e maximizar o GBW e os resultados mostram que tanto o PSO quanto o CS atingiram melhores valores em ambos os casos.

Algumas das especificações não atingiram o valor desejado pelo fato de os algoritmos considerarem como aceitável qualquer solução que, além de possuir os valores para as variáveis livres entre as faixas máximas e mínimas de cada variável, atingirem valores para as especificações dentro do intervalo considerado aceitável, como mostrado na figura 16. As especificações do projeto funcionam com um sistema de penalidade. Quanto mais longe o valor obtido ficar do valor requerido para determinada especificação, maior será a penalidade aplicada à função custo. Caso a especificação fique fora da faixa aceitável, é somado um valor que tende ao infinito à função custo. Esse tipo de restrição é chamado de *soft restrictions*, ou seja, não restringe a região de busca, somente o espaço de soluções.

5.1.4 ANÁLISE DE CORNERS

Um dos problemas críticos no projeto de CI analógico é a variabilidade do processo, ou seja, os dispositivos projetados para terem determinadas dimensões e comportamentos, além de serem todos iguais, apresentam diferenças e comportamentos diferentes dos calculados antes da fase de concepção.

Segundo Lourenço e Horta (2012), esse fenômeno afeta dispositivos em chips

diferentes, mas também dispositivos dentro do mesmo chip, e deve ser resolvido por um projeto de circuito robusto com várias técnicas compensatórias. Para verificar se a grande maioria dos circuitos fabricados funcionará de acordo com as especificações, algumas técnicas de análise de produtividade podem ser empregadas. As técnicas mais comuns nos projetos analógicos são Simulação de Monte Carlo e Análise de Corners (análise de canto).

A simulação de Monte Carlo executa muitas simulações aplicando variações aleatórias nos parâmetros dos circuitos e processos. Essa variação é dada de acordo com uma função de densidade de probabilidade gera os valores de base para a simulação Monte Carlo (BINDER et al., 1993). Então é gerada uma amostragem estocástica dos comportamentos do circuito nas condições do mundo real.

A análise de corners considera as piores hipóteses de funcionamento do circuito. Isso é feito através de simulações com combinações de variações de parâmetros de processo, por exemplo, fonte de alimentação, temperatura, etc (LOURENÇO; HORTA, 2012).

Para os circuitos dimensionados neste trabalho foram realizadas análises de corners de 4 casos: Transistores nmos rápidos e pmos rápidos (FF), nmos lentos e pmos rápidos (SF), nmos rápidos e pmos lentos (FS), nmos lentos e pmos lentos (SS).

A análise de corners é feita de forma bastante simples por parte da ferramenta de dimensionamento. Basta alterar a informação a respeito do tipo de análise no netlist do circuito. Em geral basta alterar o "TT"(como mostra a linha 3 da figura 30) para FF, FS, SS ou SF.

A tabela 5.1.4 mostra as especificações obtidas na simulação típica utilizando o CS em comparação com os valores obtidos nas análises de corners.

Tabela 10 – OTA Miller: Análise típica e análises de Corners

Especificação	Típica	FF	FS	SF	SS
Av0 (dB)	70.01	68.32	70.09	65.19	48.23
GBW (MHz)	5.00	4.89	4.62	4.67	4.01
MF (°)	62.45	62.84	60.68	61.23	66.85
SR (V/ μ s)	6.11	6.18	6.27	6.00	6.02
ICMR+ (V)	0.71	0.72	0.72	0.69	0.73
ICMR- (V)	-0.86	-0.83	-0.79	-0.82	-0.87
OS+ (V)	0.84	0.87	0.79	0.80	0.82
OS- (V)	-0.71	-0.70	-0.71	-0.69	-0.74
Pdiss (μ W)	83.18	83.24	82.01	84.23	79.69

Os valores destacados em verde representam o melhor valor para cada especificação, enquanto os de cor vermelha representam o pior valor para cada especificação. É possível observar na tabela que existe mudanças para os valores de todas as especificações. O Av0 da simulação SS (*slow slow*), por exemplo, apresentou uma queda de em torno de 65%, o que pode comprometer o funcionamento de todo o sistema. No

entanto, é importante ressaltar que a análise de corners representa um comportamento subestimado do CI.

5.2 OTA MILLER DE BAIXA TENSÃO ALIMENTADO PELO SUBSTRATO

Outro circuito dimensionado de forma automática neste trabalho é um OTA Miller de baixa tensão alimentado pelo substrato (*bulk-driven*) proposto por Ferreira e Sonkusale (2014). Ele foi proposto devido a crescente demanda por circuitos de baixa potência e baixa tensão para aplicações biomédicas e internet das coisas, por exemplo. Isso exige a exploração da região sub-limiar do transistor para alcançar uma maior razão gm/I_D , proporcionando alta eficiência energética (RAJPUT; JAMUAR, 2002). As aplicações biomédicas, por exemplo, devem operar com baixa tensão de alimentação devido a limitações do ambiente, aliadas ao fato de que uma bateria implantada deve durar vários anos (YADAV; PRASAD, 2017).

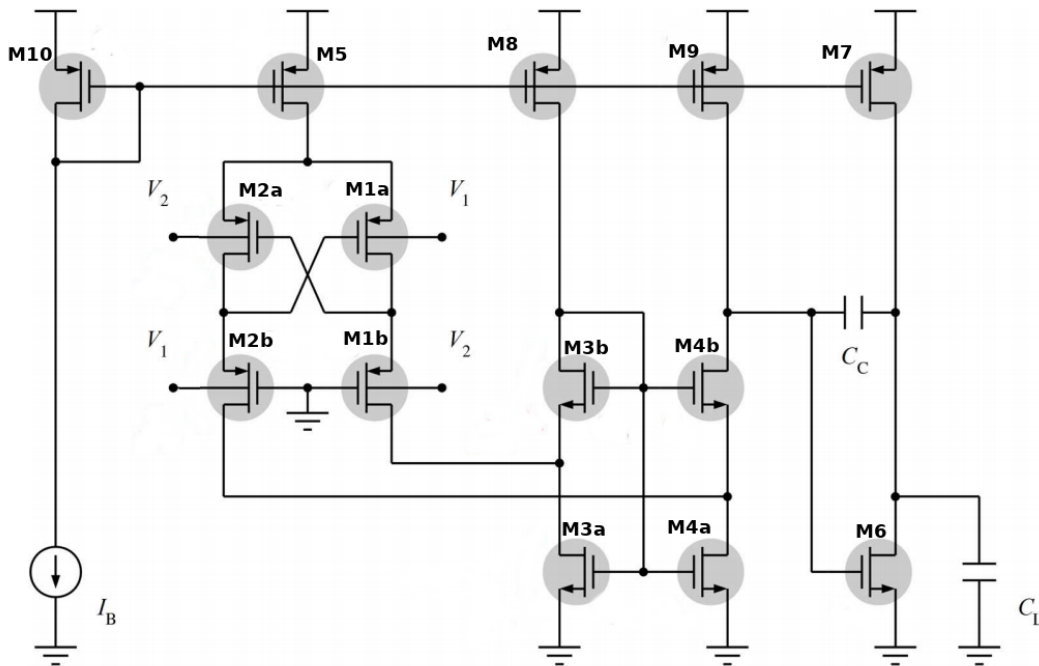
Esse circuito apresenta uma topologia mais desafiadora que os outros dois circuitos pelo fato de apresentar um maior número de variáveis livres. Seu acionamento, ao invés de ser realizado pelo gate, é realizado pelo substrato, devido a baixa tensão de alimentação que é de de 0.25 V. Essa baixa tensão faz com que o circuito opere na região de inversão fraca.

Para reduzir a tensão de limiar dos transistores são realizados implantes tipo *halo*, que são uma alteração na dopagem do material dos transistores (ANNEMA et al., 2005). Ferreira e Sonkusale (2014) propuseram uma nova topologia de OTA para combater o baixo ganho em corrente contínua com uma alimentação de 0.25 V. Essa topologia é demonstrada na figura 27.

Esses implantes nas regiões de dreno e source dos transistores são utilizados para mitigar os efeitos de canal curto. No entanto, isso causa deslocamento da tensão de limiar e uma impedância de saída que não aumenta linearmente com o aumento do comprimento do canal (CHAKRABORTY et al., 2007). Para reduzir o efeito do *halo*, é adotada uma configuração de leiaute distribuído para todos os transistores, isto é, ao invés de utilizar um único transistor grande, é adotado um conjunto de transistores menores para reduzir a impedância de saída, como mostra a figura 28 (FERREIRA; SONKUSALE, 2014).

Para esse circuito é necessário um passo adicional na ferramenta para gerar as matrizes de transistores. No modelo elétrico, como mostrado na figura 7, é possível informar o número de transistores em paralelo. Na linha 10 da figura 7, por exemplo, onde contém "`x1 2 vn 3 3 nmos_rf lr='L1*1e-6' wr=W1*1e-6`", se for adicionada a informação **m=10**, a simulação levará em conta 10 transistores em paralelo. No entanto, o modelo elétrico não permite informar o número de transistores em série. Para isso, é necessária a criação de um sub-circuito representando essas matrizes, como mostrado na figura 28.

Figura 27 – Esquemático do OTA de 0.25V proposto por Ferreira e Sonkusale (2014)



Fonte: Ferreira e Sonkusale (2014)

Para isso, a ferramenta gera um sub-circuito de forma automática a cada iteração da heurística. Quando o número de transistores em série e em paralelo das matrizes de transistores são considerados como variáveis livres no processo de otimização, a mudança desses valores em cada iteração leva à necessidade de que seja gerado um novo netlist para cada uma das células de transistores. Por exemplo, na figura 29 é mostrado o netlist representando uma matriz de transistores onde existem m transistores em paralelo e n transistores em série. O valor de m é gerado no arquivo de parâmetros, como o mostrado na figura 8, e o valor de n define o número de linhas que devem ter nesse arquivo de netlist. Na figura 29 o valor de n é igual a 7.

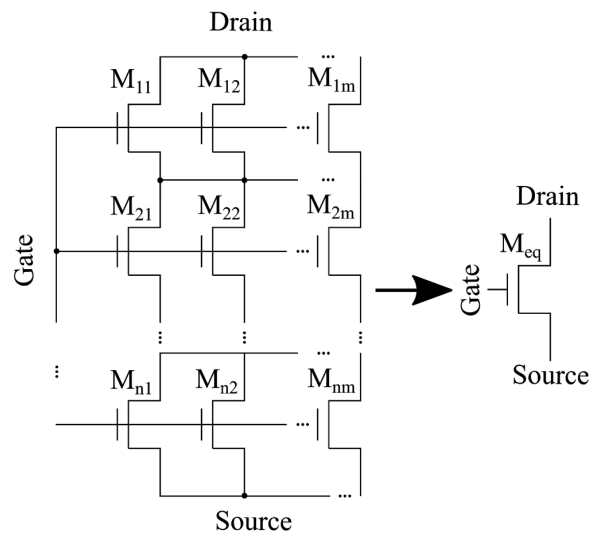
No arquivo netlist representando o OTA, ao invés de chamar o transistor com suas dimensões, como mostrado na figura 7, são chamados os sub-circuitos representando as matrizes de transistores, como demonstrado na figura 30.

Na figura 30, os três pontos na linha 7 representam a inclusão dos arquivos das outras matrizes, omitidos na figura para reduzir espaço.

Para o dimensionamento desse circuito utilizou-se o algoritmo CS, pois, além de apresentar bons resultados como o PSO e o melhor resultado de função custo dentre todos os comparados, trata-se de um algoritmo proposto mais recentemente e com menor número de trabalhos encontrados na literatura.

As especificações consideradas no projeto automático foram as mesmas apresentadas por Ferreira e Sonkusale (2014), para que seja possível uma comparação dos

Figura 28 – Uma matriz retangular de transistores nMOS com implantes de halo (à esquerda) e seu transistor equivalente (à direita)



Fonte: Ferreira e Sonkusale (2014)

Figura 29 – Netlist representando o sub-circuito M1a.

```
.subckt M1a dreno gate source subs
x1 dreno gate 1 subs pfet lr='La*1e-6' wr='Wa*1e-6' m='Ma'
x2 1 gate 2 subs pfet lr='La*1e-6' wr='Wa*1e-6' m='Ma'
x3 2 gate 3 subs pfet lr='La*1e-6' wr='Wa*1e-6' m='Ma'
x4 3 gate 4 subs pfet lr='La*1e-6' wr='Wa*1e-6' m='Ma'
x5 4 gate 5 subs pfet lr='La*1e-6' wr='Wa*1e-6' m='Ma'
x6 5 gate 6 subs pfet lr='La*1e-6' wr='Wa*1e-6' m='Ma'
x7 6 gate source subs pfet lr='La*1e-6' wr='Wa*1e-6' m='Ma'
.ends
```

Fonte: Autor.

resultados. O GBW, a potência dissipada e a área do gate foram considerados como especificação de otimização. Deseja-se maximizar o GBW e minimizar a potência e a área. As especificações de restrição do projeto incluem ganho de tensão de baixa frequência $A_{v0} \geq 60$ dB, margem de fase $MF \geq 60^\circ$ e taxa de variação $SR \geq 0.64$ V/ms. Esses valores são os obtidos em Ferreira e Sonkusale (2014).

A capacitância de carga CL foi definida em 15 pF. O circuito foi convertido em um problema de otimização contendo 24 variáveis livres. Cada transistor é composto de uma matriz retangular de transistores unitários. Como algumas matrizes de transistores devem ser iguais, o problema de otimização é reduzido para 5 matrizes a serem dimensionadas: $M1a = M2a$, $M1b = M2b$, $M3a = M4a$, $M3b = M4b$ e M6.

Figura 30 – Netlist que faz a chamada dos sub-circuitos de transistores.

```

1  * Miller_013
2  .inc 'IBM013/models/design.inc'
3  .lib 'IBM013/models/allModels.inc' tt
4
5  *inclusao do arquivo que contém as matrizes de transistores
6  .include celulas/M1a.txt
7  ...
8  .subckt OTA_enhanced in1 in2 vdd vss out
9  x2a 6 7 2 in2 M1a
10 x1a 7 6 2 in1 M1a
11 x2b 8 vss 6 in1 M1b
12 x1b 9 vss 7 in2 M1b
13 x3b 3 3 9 9 M3b
14 x4b 4 3 8 8 M3b
15 x3a 9 3 vss vss M3a
16 x4a 8 3 vss vss M3a
17 x6 out 4 vss vss M6
18 i5 vdd 2 'I5*1e-9'
19 i7 vdd out 'I7*1e-9'
20 i8 vdd 3 'I8*1e-9'
21 i9 vdd 4 'I8*1e-9'
22 Cc 4 out 'Cc*1e-12' IC=0
23 Cl out vss 15p IC=0
24
25 .ends

```

Fonte: Autor.

Cada array tem como variáveis livres a largura W_{un} e o comprimento L_{un} dos transistores unitários, o número de transistores unitários em série (n) e o número de transistores unitários em paralelo (m). As fontes de correntes denominadas I5, I7 e I8 e a capacitância de acoplamento Cc também são consideradas como variáveis livres no projeto do circuito.

As variáveis livres podem variar nas seguintes faixas: 0.13 a 2.50 μm para L, 0.35 a 2.50 μm para W, 8 a 128 para m, 2 a 16 para n, 1 a 40 nA para as correntes I e 2 a 7 pF para Cc. O intervalo de valores para W e L são definidos pela tecnologia, enquanto os valores para n e m foram definidos com base nos valores mínimos e máximos utilizados em Ferreira e Sonkusale (2014). A tecnologia utilizada é IBM 0.13 μm com uma tensão de alimentação de 0.25 V.

A otimização desse circuito exigiu um tempo de execução de aproximadamente 1 hora e 53 minutos em um PC IntelCore i7-3770 de 3.40 GHz com 8 GB de memória principal. A Tabela 12 mostra os valores obtidos para cada uma das variáveis livres. Uma figura de mérito (FoM) é proposta por Ferreira e Sonkusale (2014) para quantificar a eficiência energética do circuito. Esta FoM é definida como

$$FoM = 100 \cdot \frac{GBW \cdot C_L}{I_{DD}} \quad (5.1)$$

Onde I_{DD} é a corrente proveniente da tensão de alimentação V_{DD} .

A tabela 11 mostra o desempenho elétrico do circuito obtido através da otimização automática e uma comparação com os resultados descritos em Ferreira e Sonkusale (2014).

Tabela 11 – Indicadores de desempenho obtidos para o OTA Miller alimentado pelo substrato.

Especificação	Este trabalho	(FERREIRA; SONKUSALE, 2014)
Av0 (dB)	67.64	60.00
GBW (kHz)	7.54	1.88
MF (°)	88.00	52.50
Slew-rate (V/ms)	0.65	0.77
P_{diss} (nW)	10.12	18.00
Área do gate (μm^2)	1719	1372
FoM (V^{-1}) (eq. 5.1)	275.85	67.40

Pode-se observar que o dimensionamento automático usando a heurística CS é capaz de fornecer melhores valores para especificações como Av0, GBW, PM e consumo de energia comparados aos valores obtidos por Ferreira e Sonkusale (2014).

O Av0 obtido é cerca de 11% melhor, enquanto o GBW é 4 vezes maior. O consumo de energia necessário para a operação do circuito é cerca de 44% menor, o que é muito importante para aplicações onde o consumo de energia é um fator crucial, como em aplicações biomédicas. Os valores das demais especificações ficaram próximos do valor desejado. Além disso, o projeto automático apresentou maior eficiência energética, indicado pelo FoM da equação 5.1, atingindo $275,85V^{-1}$, que é cerca de 4 vezes maior que o da referência.

A ferramenta UCAF foi capaz de dimensionar os transistores operando na região sublimiar, como é possível verificar pela razão g_m/I_D na Tabela 12. Todos os transistores são polarizados com g_m/I_D maior que $27 V^{-1}$, o que indica a operação na região de inversão fraca. A Tabela 12 mostra ainda os valores das variáveis livres obtidos nesse projeto.

Também foram realizadas análises de corners para esse amplificador. A tabela 13 mostra os resultados obtidos nas quatro análises (em vermelho os piores e em verde os melhores).

5.3 AMPLIFICADOR OPERACIONAL TELESCÓPICO

Outro circuito dimensionado neste trabalho é o amplificador operacional Telescópico. Suas correntes de entrada são espelhadas à de saída. As especificações elétricas do Telescópico são semelhantes às do OTA Miller. Na figura 31 é demonstrado o esquemático do amp-op telescópico.

Tabela 12 – Valores das variáveis após a otimização.

Componente	Valor	g_m/I_D (V^{-1})
$M_{1a} = M_{2a}$	$\frac{8 \cdot 2.50 \mu m}{2 \cdot 2.50 \mu m} = \frac{20 \mu m}{5 \mu m}$	32.05
$M_{1b} = M_{2b}$	$\frac{128 \cdot 0.35 \mu m}{16 \cdot 0.13 \mu m} = \frac{44.8 \mu m}{2.08 \mu m}$	27.67
$M_{3b} = M_{4b}$	$\frac{37 \cdot 0.50 \mu m}{2 \cdot 1.55 \mu m} = \frac{18.58 \mu m}{3.10 \mu m}$	29.90
$M_{3a} = M_{4a}$	$\frac{128 \cdot 2.49 \mu m}{14 \cdot 0.13 \mu m} = \frac{318.72 \mu m}{1.82 \mu m}$	31.75
M_6	$\frac{32 \cdot 0.47 \mu m}{12 \cdot 0.32 \mu m} = \frac{15.04 \mu m}{3.84 \mu m}$	31.89
I_5	32.91 nA	-
I_7	5.59 nA	-
I_8	1.00 nA	-
C_C	2.00 pF	-

Tabela 13 – Bulk-driven OTA Miller: Análise típica e análises de Corners

Especificação	Típica	FF	FS	SF	SS
Av0 (dB)	67.64	64.36	62.47	65.27	74.65
GBW (kHz)	7.54	5.86	8.83	5.21	8.43
MF (°)	88.00	66.32	66.06	74.00	81.00
Slew-rate (V/ms)	0.65	0.69	0.72	0.65	0.59
P_{diss} (nW)	10.12	10.24	10.12	10.31	10.10

Para o dimensionamento desse amplificador também foi utilizado o algoritmo CS. Os valores para os parâmetros do algoritmo são os mesmos definidos na calibragem para o dimensionamento do OTA Miller. O circuito é transformado em um problema contendo 11 variáveis livres. Dos oito transistores que compõem o circuito, alguns possuem o mesmo tamanho (M_1 e M_2 , M_3 e M_4 , M_5 e M_6 , M_7 e M_8), totalizando comprimento (L) e largura (W) de 4 diferentes transistores. As outras variáveis livres representam a corrente I_{bias} e as tensões de referência V_{cp} e V_{cn} .

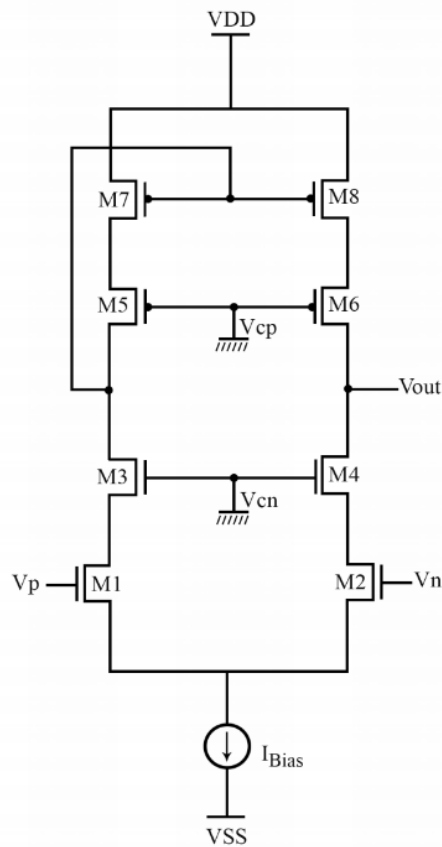
As especificações consideradas como de otimização são área do gate e potência dissipada, em que deseja-se minimizar ambas. As demais especificações e os valores desejados são mostrados na segunda coluna da Tabela 14. A tecnologia utilizada para simulação é TSMC 0.18 μm com uma tensão de alimentação de 1.8 V. Os valores de W variam de 0,22 μm a 50 μm e de L de 0,18 μm a 10 μm . Os valores para V_{cn} e V_{cp} podem variar de 0 a -0.9 V e 0 a 0.9 V, respectivamente. A corrente I_{bias} pode variar de 1 a 20 μA .

As variáveis são denominadas W_1 , L_1 (transistores M_1 e M_2), W_3 , L_3 (transistores M_3 e M_4), W_5 , L_5 (transistores M_5 e M_6), W_7 , L_7 (transistores M_7 e M_8), V_n , V_p e I_b . A Tabela 15 apresenta os valores obtidos para cada uma das variáveis ao final da execução.

É possível observar na Tabela 14 que o uso do algoritmo CS apresentou resultados melhores para 5 das 6 especificações que foram possíveis comparar.

Nas tabelas 16 e 15 são demonstradas uma comparação entre a simulação típica

Figura 31 – Esquemático de um amp-op telescopic em tecnologia CMOS



Fonte: Adaptado de Domanski (2016)

e as simulações de piores casos e os valores para cada uma das variáveis livres após o dimensionamento, respectivamente, para o amp-op Telescopic.

5.4 CONCLUSÃO

Neste capítulo foram demonstrados os valores definidos através de uma calibragem para cada um dos parâmetros das heurísticas CS e FA. Esta calibragem foi realizada através da comparação dos resultados no dimensionamento do CI OTA Miller. Também foi demonstrado os resultados obtidos no projeto automático dos amplificadores Telescopic e *bulk-driven* OTA Miller utilizando a heurística CS.

Após o dimensionamento dos três amplificadores foram realizadas análises de Corners e os resultados foram comparados com outros trabalhos da literatura.

Tabela 14 – Especificações para os melhores resultados de cada algoritmo de otimização avaliado.

Especificação	Valor Requerido	CS (este trabalho)	Domanski (2016)	Severo (2012)
Av0 (dB)	≥ 60.00	65.21	63.90	62.88
GBW (MHz)	≥ 2.00	15.27	4.71	3.50
MF (°)	≥ 60.00	84.70	75.98	50.09
SR (V/ μ s)	≥ 5	11.11	7.14	5.76
ICMR+ (V)	≥ 0.4	0.44	0.58	0.64
ICMR- (V)	≤ -0.4	-0.89	-0.71	-0.60
OS+ (V)	≥ 0.4	0.75	-	-
OS- (V)	≤ -0.4	-0.40	-	-
Pdiss (μ W)	minimizar	20.99	28.84	49.66
Área do gate (μ m ²)	minimizar	12.44	16.40	19.47

Tabela 15 – Valores das variáveis após a otimização.

Componente	Valor
$M_1 = M_2$ (W/L)	$\frac{1.5\mu m}{0.49\mu m}$
$M_3 = M_4$ (W/L)	$\frac{7.97\mu m}{0.33\mu m}$
$M_5 = M_6$ (W/L)	$\frac{3.23\mu m}{0.50\mu m}$
$M_7 = M_8$ (W/L)	$\frac{2.34\mu m}{0.48\mu m}$
V_n	-0.12 V
V_p	0.14 V
I_B	11.66 nA

Tabela 16 – Telescopic: Análise típica e análises de Corners

Especificação	Típica	FF	FS	SF	SS
Av0 (dB)	65.21	52.48	65.03	57.71	66.68
GBW (MHz)	15.27	15.41	15.42	15.03	14.96
MF (°)	84.70	84.65	84.59	84.79	84.69
SR (V/ μ s)	11.11	11.02	11.27	10.85	10.70
ICMR+ (V)	0.44	0.35	0.32	0.36	0.33
ICMR- (V)	-0.89	-0.90	-0.90	-0.90	-0.90
OS+ (V)	0.75	0.76	0.76	0.76	0.75
OS- (V)	-0.40	-0.36	-0.36	-0.41	-0.42
Pdiss (μ W)	20.99	20.99	20.99	20.99	20.99

6 CONCLUSÃO

Neste trabalho foi realizado um estudo a fim de apresentar métodos capazes de realizar o dimensionamento automático de CIs de modo a atingir as especificações. Para isso, foi realizado um estudo sobre as principais técnicas apresentadas na literatura.

Após esse levantamento bibliográfico, foi constatado que o projeto de CIs analógicos na grande maioria das vezes é realizado de forma manual, utilizando a equações para determinar valores iniciais e experiência do projetista para um ajuste a fim de atingir as especificações. O dimensionamento de CIs analógicos de forma automática utilizando heurísticas ganhou bastante destaque nos últimos anos, uma vez que tem demonstrado capacidade de atingir resultados melhores e em menor tempo do que o projeto manual.

Com base no estudo realizado acerca de heurísticas, foram selecionadas duas metaheurísticas bio-inspiradas, Cuckoo Search e Firefly Algorithm, a fim de observar o desempenho de ambas no processo de dimensionamento de um OTA Miller de dois estágios utilizando um modelo de critério de parada baseado em melhorias da função custo utilizando o conceito de janela móvel. O desempenho das heurísticas foi comparado com o do PSO.

Outros dois amplificadores também foram dimensionados. Um amp-op Telescopic e um *Bulk-driven* Miller de baixa tensão. Para esse segundo CI, foi necessária uma série de modificações na ferramenta a fim de permitir a criação automática das matrizes de transistores. O fluxo básico para essa tarefa, assim como das principais partes da ferramenta UCAF, foi demonstrado neste trabalho.

Percebe-se que é possível modelar o circuito como um problema de otimização e dimensioná-lo através do uso de heurísticas de forma a atender aos valores estipulados como aceitáveis para as especificações do circuito.

Os algoritmos PSO e CS apresentaram resultados próximos considerando as restrições do circuito Miller e ambos tiveram melhor desempenho que o algoritmo FA.

A heurística CS foi utilizada para dimensionamento dos outros dois circuitos. No dimensionamento do OTA de baixa tensão acionado pelo substrato, nota-se que a ferramenta UCAF foi eficaz no dimensionamento de matrizes retangulares de transistores unitários nos quais o número de elementos paralelos e em série da matriz, além do tamanho dos transistores unitários, eram variáveis livres. O circuito resultante, que opera na região de inversão fraca, demonstra a busca eficiente no espaço de projeto. Os resultados de desempenho mostraram que as especificações mínimas foram atendidas, como ganho de tensão de baixa frequência e margem de fase. Além disso, foi possível otimizar o GBW e o consumo de energia. Esse amplificador dimensionado resultou em um consumo de energia de 10,12 nW, que é 44% menor do que o mesmo circuito projetado manualmente. Foi demonstrado neste trabalho que uma pesquisa eficiente no espaço de design com uma ferramenta CAD analógica pode resultar em otimização

Para o circuito Telescopic o dimensionamento com o CS também demonstrou bons resultados, inclusive melhores que em outros dois trabalhos que também utilizam heurísticas para o dimensionamento automático desse CI.

Embora não possa ser afirmado que o CS e os valores de parâmetros definidos para essa heurística sejam adequados para dimensionar outros tipos de circuitos analógicos, é possível notar que a abordagem está correta e, com alguns ajustes na ferramenta e no algoritmo, o CS é capaz de obter bons resultados levando a um CI otimizado que atende às especificações desejadas.

Além disso, com esse trabalho constata-se que a utilização de heurísticas bioinspiradas se mostra adequada para problemas de dimensionamento otimizado de circuitos integrados analógicos. Os resultados obtidos trazem contribuição à ferramenta.

Como trabalhos futuros, deseja-se realizar a descrição a nível de leiaute dos circuitos para que eles possam ser prototipados e seus desempenhos avaliados através de medidas elétricas.

REFERÊNCIAS

- AARTS, E. H.; LENSTRA, J. K. *Local search in combinatorial optimization*. [S.l.]: Princeton University Press, 2003. Citado na página 24.
- ALLEN, P. E. et al. *CMOS analog circuit design*. [S.l.]: Holt, Rinehart and Winston New York, 1987. Citado na página 44.
- ALPAYDIN, G.; BALKIR, S.; DUNDAR, G. An evolutionary approach to automatic synthesis of high-performance analog integrated circuits. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 7, n. 3, p. 240–252, 2003. Citado na página 22.
- ANNEMA, A.-J. et al. Analog circuits in ultra-deep-submicron cmos. *IEEE Journal of Solid-State Circuits*, IEEE, v. 40, n. 1, p. 132–143, 2005. Citado na página 83.
- BABU, B.; ANGIRA, R. New strategies of differential evolution for optimization of extraction process. In: *Proceedings of International Symposium & 56th Annual Session of IChE (CHEMCON 2003)*. [S.l.: s.n.], 2003. Citado na página 66.
- BARRA, S. et al. Multi-objective genetic algorithm optimization of cmos operational amplifiers. In: IEEE. *Microelectronics (ICM), 2012 24th International Conference on*. [S.l.], 2012. p. 1–4. Citado 4 vezes nas páginas 33, 34, 80 e 81.
- BARROS, M.; GUILHERME, J.; HORTA, N. An evolutionary optimization kernel using a dynamic ga-svm model applied to analog ic design. In: IEEE. *Circuit Theory and Design, 2007. ECCTD 2007. 18th European Conference on*. [S.l.], 2007. p. 32–35. Citado 4 vezes nas páginas 22, 27, 28 e 30.
- BARROS, M.; GUILHERME, J.; HORTA, N. Analog circuits optimization based on evolutionary computation techniques. *INTEGRATION, the VLSI journal*, Elsevier, v. 43, n. 1, p. 136–155, 2010. Citado 3 vezes nas páginas 49, 63 e 64.
- BARTHELEMY, P.; BERTOLOTTI, J.; WIERSMA, D. S. A lévy flight for light. *Nature*, Nature Publishing Group, v. 453, n. 7194, p. 495, 2008. Citado na página 56.
- BARTUMEUS, F. et al. Helical lévy walks: adjusting searching statistics to resource availability in microzooplankton. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 100, n. 22, p. 12771–12775, 2003. Citado na página 57.
- BERGH, F. V. D. *An analysis of particle swarm optimizers*. Tese (Doutorado) — University of Pretoria, 2007. Citado na página 65.
- BINDER, K. et al. Monte carlo simulation in statistical physics. *Computers in Physics*, AIP, v. 7, n. 2, p. 156–157, 1993. Citado na página 82.
- BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, ACM, v. 35, n. 3, p. 268–308, 2003. Citado na página 55.
- BONABEAU, E.; DORIGO, M.; THERAULAZ, G. *Swarm intelligence: from natural to artificial systems*. [S.l.]: Oxford university press, 1999. Citado 2 vezes nas páginas 24 e 55.
- BROWN, C. T.; LIEBOVITCH, L. S.; GLENDON, R. Lévy flights in dove juv/hoansi foraging patterns. *Human Ecology*, Springer, v. 35, n. 1, p. 129–138, 2007. Citado na página 57.

CASTRO, L. N. D.; ZUBEN, F. J. V. *Recent developments in biologically inspired computing*. [S.l.]: Igi Global, 2005. Citado na página 54.

CHAKRABORTY, S. et al. Impact of halo doping on the subthreshold performance of deep-submicrometer cmos devices and circuits for ultralow power analog/mixed-signal applications. *IEEE Transactions on Electron Devices*, IEEE, v. 54, n. 2, p. 241–248, 2007. Citado na página 83.

CHAN, F. T.; TIWARI, M. K. Preface: Swarm intelligence, focus on ant and particle swarm optimization. In: *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization*. [S.l.]: InTech, 2007. Citado na página 23.

DE, B. P. et al. Soft computing-based approach for optimal design of on-chip comparator and folded-cascode op-amp using colliding bodies optimization. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, Wiley Online Library, v. 29, n. 5, p. 873–896, 2016. Citado 2 vezes nas páginas 33 e 36.

DEGRAUWE, M. G. et al. Idac: An interactive design tool for analog cmos circuits. *IEEE Journal of solid-state circuits*, IEEE, v. 22, n. 6, p. 1106–1116, 1987. Citado na página 32.

DEHBASHIAN, M. A new hybrid algorithm for analog ics optimization based on the shrinking circles technique. *Integration, the VLSI Journal*, Elsevier, v. 56, p. 148–166, 2017. Citado 2 vezes nas páginas 33 e 36.

DEY, N. et al. Firefly algorithm for optimization of scaling factors during embedding of manifold medical information: an application in ophthalmology imaging. *Journal of Medical Imaging and Health Informatics*, American Scientific Publishers, v. 4, n. 3, p. 384–394, 2014. Citado na página 23.

DOMANSKI, R. A. *Métodos Eficientes na Estimativa de Produtividade para o Dimensionamento Automático de Circuitos Integrados Analógicos*. Dissertação (Mestrado) — Universidade Federal do Pampa, 2016. Citado 6 vezes nas páginas 30, 39, 48, 76, 89 e 90.

EL-TURKY, F.; PERRY, E. E. Blades: An artificial intelligence approach to analog circuit design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, IEEE, v. 8, n. 6, p. 680–692, 1989. Citado na página 32.

FERREIRA, L. H.; SONKUSALE, S. R. A 60-db gain ota operating at 0.25-v power supply in 130-nm digital cmos process. *IEEE Transactions on Circuits and Systems I: Regular Papers*, IEEE, v. 61, n. 6, p. 1609–1617, 2014. Citado 7 vezes nas páginas 15, 23, 83, 84, 85, 86 e 87.

FRANÇA, F. O. d. et al. Algoritmos bio-inspirados aplicados à otimização dinâmica. [sn], 2005. Citado na página 53.

GANDOMI, A. H.; YANG, X.-S.; ALAVI, A. H. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with computers*, Springer, v. 29, n. 1, p. 17–35, 2013. Citado 3 vezes nas páginas 54, 55 e 56.

GIELEN, G. G.; RUTENBAR, R. A. Computer-aided design of analog and mixed-signal integrated circuits. *Proceedings of the IEEE*, IEEE, v. 88, n. 12, p. 1825–1854, 2000. Citado na página 22.

GIUNCHIGLIA, E. et al. Evaluating search heuristics and optimization techniques in propositional satisfiability. In: SPRINGER. *International Joint Conference on Automated Reasoning*. [S.l.], 2001. p. 347–363. Citado na página 30.

GLOVER, F. Tabu search—part i. *ORSA Journal on computing, INFORMS*, v. 1, n. 3, p. 190–206, 1989. Citado na página 24.

GOLDBERG, D. E.; HOLLAND, J. H. Genetic algorithms and machine learning. *Machine learning*, Springer, v. 3, n. 2, p. 95–99, 1988. Citado na página 55.

GRASSO, A. D. et al. Design methodology of subthreshold three-stage cmos otas suitable for ultra-low-power low-area and high driving capability. *IEEE Transactions on Circuits and Systems I: Regular Papers, IEEE*, v. 62, n. 6, p. 1453–1462, 2015. Citado na página 23.

HERSHENSON, M. del M.; BOYD, S. P.; LEE, T. H. Gpcad: A tool for cmos op-amp synthesis. In: IEEE. *Computer-Aided Design, 1998. ICCAD 98. Digest of Technical Papers. 1998 IEEE/ACM International Conference on*. [S.l.], 1998. p. 296–303. Citado na página 32.

HOLBERG, D. R.; ALLEN, P. E. Cmos analog circuit design. <http://www.cicmaa.com>, 2002. Citado 2 vezes nas páginas 28 e 76.

JAFARI, A. et al. A design automation system for cmos analog integrated circuits using new hybrid shuffled frog leaping algorithm. *Microelectronics Journal, Elsevier*, v. 43, n. 11, p. 908–915, 2012. Citado 3 vezes nas páginas 33, 34 e 65.

JAFARI, A.; BIJAMI, E.; ZEKRI, M. Parametric optimization of multistage operational amplifier by using imperialist competitive algorithm. In: *Proceedings of the 3rd International Conference on Computer and Electrical Engineering*. [S.l.: s.n.], 2010. p. V2–339. Citado na página 32.

JAFARI, A. et al. Design of analog integrated circuits by using genetic algorithm. In: IEEE. *Computer Engineering and Applications (ICCEA), 2010 Second International Conference on*. [S.l.], 2010. v. 1, p. 578–581. Citado 4 vezes nas páginas 22, 32, 33 e 34.

KAVEH, A.; TALATAHARI, S. A novel heuristic optimization method: charged system search. *Acta Mechanica, Springer*, v. 213, n. 3, p. 267–289, 2010. Citado na página 53.

KENNEDY, J. Particle swarm optimization. In: *Encyclopedia of machine learning*. [S.l.]: Springer, 2011. p. 760–766. Citado na página 54.

KRISHNANAND, K.; GHOSE, D. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *Swarm intelligence, Springer*, v. 3, n. 2, p. 87–124, 2009. Citado na página 54.

KROHLING, R. A. Gaussian swarm: a novel particle swarm optimization algorithm. In: IEEE. *Cybernetics and Intelligent Systems, 2004 IEEE Conference on*. [S.l.], 2004. v. 1, p. 372–376. Citado 3 vezes nas páginas 63, 68 e 69.

LEE, C.-W. et al. Junctionless multigate field-effect transistor. *Applied Physics Letters, AIP*, v. 94, n. 5, p. 053511, 2009. Citado na página 29.

- LOURENÇO, N.; HORTA, N. Genom-pof: multi-objective evolutionary synthesis of analog ics with corners validation. In: ACM. *Proceedings of the 14th annual conference on Genetic and evolutionary computation*. [S.l.], 2012. p. 1119–1126. Citado 4 vezes nas páginas 32, 36, 81 e 82.
- LUZ, M. D. et al. Improvements in the statistical approach to random lévy flight searches. *Physica A: Statistical Mechanics and its Applications*, Elsevier, v. 295, n. 1, p. 89–92, 2001. Citado na página 57.
- MALLICK, S. et al. Optimal sizing of cmos analog circuits using gravitational search algorithm with particle swarm optimization. *International Journal of Machine Learning and Cybernetics*, Springer, v. 8, n. 1, p. 309–331, 2017. Citado 3 vezes nas páginas 33, 35 e 65.
- MANTEGNA, R. N. Fast, accurate algorithm for numerical simulation of levy stable stochastic processes. *Physical Review E*, APS, v. 49, n. 5, p. 4677, 1994. Citado na página 58.
- MORETO, R. A. d. L. et al. Ota design by means of an evolutionary system integrated to spice. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, SciELO Brasil, v. 23, n. 6, p. 694–710, 2012. Citado 3 vezes nas páginas 21, 22 e 29.
- ONODERA, H.; KANBARA, H.; TAMARU, K. Operational-amplifier compilation with performance optimization. *IEEE Journal of solid-state circuits*, IEEE, v. 25, n. 2, p. 466–473, 1990. Citado na página 32.
- PAVLYUKEVICH, I. Lévy flights, non-local search and simulated annealing. *Journal of Computational Physics*, Elsevier, v. 226, n. 2, p. 1830–1844, 2007. Citado na página 57.
- PAYNE, R. B.; SORENSEN, M. D. *The cuckoos*. [S.l.]: Oxford University Press, 2005. v. 15. Citado na página 56.
- RAJPUT, S.; JAMUAR, S. S. Low voltage analog circuit design techniques. *IEEE Circuits and Systems Magazine*, IEEE, v. 2, n. 1, p. 24–42, 2002. Citado na página 83.
- RAZAVI, B. *Design of analog CMOS integrated circuits*. [S.l.]: Tsinghua University Press Co., Ltd, 2001. Citado 2 vezes nas páginas 48 e 75.
- REYNOLDS, A. M.; FRYE, M. A. Free-flight odor tracking in drosophila is consistent with an optimal intermittent scale-free search. *PloS one*, Public Library of Science, v. 2, n. 4, p. e354, 2007. Citado 2 vezes nas páginas 57 e 58.
- RIBEIRO, L. A. D. et al. Otimização estrutural de treliças utilizando o algoritmo firefly. Florianópolis, SC, 2014. Citado na página 63.
- SASIKUMAR, A. Operational amplifier circuit sizing based on nsga-ii and particle swarm optimization. In: IEEE. *Networks & Advances in Computational Technologies (NetACT), 2017 International Conference on*. [S.l.], 2017. p. 64–68. Citado 2 vezes nas páginas 33 e 36.
- SEVERO, L.; GIRARDI, A. Ucaf a framework for analog integrated circuit analysis and design. In: *XI Microelectronics Students Forum (SForum 2011)-Chip on the Cliffs*. [S.l.: s.n.], 2011. Citado 6 vezes nas páginas 28, 29, 30, 42, 44 e 50.

SEVERO, L. C. Uma ferramenta para o dimensionamento automático de circuitos integrados analógicos considerando análise de produtividade. Universidade Federal do Pampa, 2012. Citado 9 vezes nas páginas 21, 30, 39, 40, 41, 46, 48, 75 e 90.

SEVERO, L. C. et al. Simulated annealing to improve analog integrated circuit design: Trade-offs and implementation issues. In: *Simulated Annealing-Single and Multiple Objective Problems*. [S.l.]: InTech, 2012. Citado 3 vezes nas páginas 33, 34 e 65.

SHI, Y.; EBERHART, R. A modified particle swarm optimizer. In: IEEE. *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*. [S.l.], 1998. p. 69–73. Citado na página 64.

SIARRY, P. et al. Enhanced simulated annealing for globally minimizing functions of many-continuous variables. *ACM Transactions on Mathematical Software (TOMS)*, ACM, v. 23, n. 2, p. 209–228, 1997. Citado na página 24.

STEHR, G. et al. Initial sizing of analog integrated circuits by centering within topology-given implicit specification. In: IEEE COMPUTER SOCIETY. *Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design*. [S.l.], 2003. p. 241. Citado 2 vezes nas páginas 68 e 69.

TAHERZADEH-SANI, M. et al. Design optimization of analog integrated circuits using simulation-based genetic algorithm. In: IEEE. *Signals, Circuits and Systems, 2003. SCS 2003. International Symposium on*. [S.l.], 2003. v. 1, p. 73–76. Citado na página 22.

TALBI—ANTONIO, M. B.-G.; ALBA, N. Metaheuristics for multiobjective combinatorial optimization problems: Review and recent issues. 2006. Citado na página 23.

TALUKDER, S. *Mathematical Modelling and Applications of Particle Swarm Optimization*. 2011. Citado na página 69.

VISWANATHAN, G. M. et al. Lévy flight search patterns of wandering albatrosses. *Nature*, Nature Publishing Group, v. 381, n. 6581, p. 413, 1996. Citado na página 57.

VURAL, R. Analog circuit sizing via swarm intelligence. *AEU-International Journal of Electronics and Communications*, Elsevier, v. 66, n. 9, p. 732–740, 2012. Citado 5 vezes nas páginas 21, 24, 33, 35 e 65.

YADAV, C.; PRASAD, S. Low voltage low power sub-threshold operational amplifier in 180nm cmos. In: IEEE. *Sensing, Signal Processing and Security (ICSSS), 2017 Third International Conference on*. [S.l.], 2017. p. 35–38. Citado na página 83.

YANG, X.-S. Biology-derived algorithms in engineering optimization. *arXiv preprint arXiv:1003.1888*, 2010. Citado na página 55.

YANG, X.-S. Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, Inderscience Publishers, v. 2, n. 2, p. 78–84, 2010. Citado 5 vezes nas páginas 54, 60, 62, 63 e 78.

YANG, X.-S. *Nature-inspired metaheuristic algorithms*. [S.l.]: Luniver press, 2010. Citado 3 vezes nas páginas 53, 58 e 59.

YANG, X.-S.; DEB, S. Cuckoo search via lévy flights. In: IEEE. *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on.* [S.l.], 2009. p. 210–214. Citado 4 vezes nas páginas 54, 56, 58 e 59.

YENGUI, F. et al. A hybrid ga-sqp algorithm for analog circuits sizing. *Circuits and Systems*, Scientific Research Publishing, v. 3, n. 02, p. 146, 2012. Citado 2 vezes nas páginas 33 e 34.

ZHAO, X. et al. Transconductance improvement method for low-voltage bulk-driven input stage. *Integration, the VLSI journal*, Elsevier, v. 49, p. 98–103, 2015. Citado na página 23.

ZHOU, G.-D. et al. Energy-aware wireless sensor placement in structural health monitoring using hybrid discrete firefly algorithm. *Structural Control and Health Monitoring*, Wiley Online Library, v. 22, n. 4, p. 648–666, 2015. Citado na página 61.

ZIELINSKI, K.; LAUR, R. Stopping criteria for a constrained single-objective particle swarm optimization algorithm. *Informatica*, v. 31, n. 1, 2007. Citado 2 vezes nas páginas 65 e 66.