

UNIVERSIDADE FEDERAL DO PAMPA

Robson André Domanski

**MÉTODOS EFICIENTES NA ESTIMATIVA DE PRODUTIVIDADE PARA O
DIMENSIONAMENTO AUTOMÁTICO DE CIRCUITOS INTEGRADOS
ANALÓGICOS**

Alegrete

2016

Robson André Domanski

**MÉTODOS EFICIENTES NA ESTIMATIVA DE PRODUTIVIDADE PARA O
DIMENSIONAMENTO AUTOMÁTICO DE CIRCUITOS INTEGRADOS
ANALÓGICOS**

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

2016

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) .

D666m Domanski, Robson André

Métodos Eficientes de Estimativa de Produtividade para o
Dimensionamento Automático de Circuitos Integrados Analógicos
/ Robson André Domanski.

92 p.

Dissertação(Mestrado)-- Universidade Federal do Pampa,
MESTRADO EM ENGENHARIA ELÉTRICA, 2016.

"Orientação: Alessandro Girardi".

1. Circuitos Integrados Analógicos. 2. Particle Swarm
Optimization. 3. Monte Carlo. I. Título.

Robson André Domanski

MÉTODOS EFICIENTES NA ESTIMATIVA DE PRODUTIVIDADE PARA O DIMENSIONAMENTO AUTOMÁTICO DE CIRCUITOS INTEGRADOS ANALÓGICOS

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, 13 de Dezembro de 2016.

Banca examinadora:



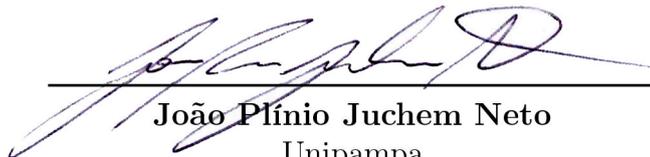
Alessandro Girardi

Orientador



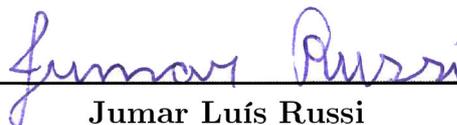
Hamilton Klimach

UFRGS



João Plínio Juchem Neto

Unipampa



Jumar Luís Russi

Unipampa

*Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.*

AGRADECIMENTOS

Ao chegarmos com os momentos das conclusões do trabalho, é de grande importância lembrar que todos os desafios que tivemos no decorrer do tempo, não os vencemos sozinhos, pois sempre há alguém para incentivarmos e ajudar-nos a atingir nossas metas e realizarmos nossos sonhos. Desta forma, em homenagem a todos que de alguma forma colaboraram com o desenvolvimento deste trabalho, eu manifesto o meu sincero agradecimento.

Agradeço ao professor Alessandro Girardi pela orientação junto ao Grupo de Arquitetura de Computadores e Microeletrônica (GAMA), aos desafios em mim impostos e a dicas de organização, os quais com certeza foram de grande importância para o desenvolvimento do trabalho e para o meu processo de aprendizagem. A Fundação de Amparo à Pesquisa do Rio Grande do Sul (FAPERGS) pelo apoio financeiro ao projeto. Agradeço aos colegas de Mestrado pelo apoio em todos os momentos. A minha mãe Nair, e meu irmão Christian, que em toda a minha vida estiveram sempre me apoiando e dando-me forças para seguir com meus sonhos. Agradeço ao professor Lucas Severo que sempre esteve disposto a ajudar e esclarecer dúvidas sobre a ferramenta UCAF. Aos colegas de Pesquisa Luiz Antônio e Alian Engroff e aos demais colegas do GAMA pela parceria ao decorrer do projeto e experiências trocadas.

*"A única maneira de fazer um bom trabalho é amando o que você faz. Se você ainda não encontrou, continue procurando. Não se desespere. Assim como no amor, você saberá quando tiver encontrado."
(Steve Jobs)*

RESUMO

O projeto de circuitos integrados analógicos, dentro da indústria da microeletrônica tem a sua evolução ditada pela grande necessidade da integração de circuitos mistos. Esta evolução faz com que os dispositivos semicondutores sejam cada vez mais miniaturizados, o que implica na complexidade cada vez maior no processo de fabricação, resultando em uma grande variabilidade de parâmetros. Esta complexidade no projeto está diretamente ligada ao dimensionamento dos dispositivos que compõem o circuito, já que o espaço de projeto é altamente não-linear. O dimensionamento de circuito analógico pode ser modelado como um problema de otimização e resolvido por heurísticas de otimização. A solução resultante é dependente da estratégia de modelagem e na estimativa de desempenho, o que é feito, em geral, por simulação elétrica. Neste contexto, foi desenvolvida a ferramenta UCAF. No entanto, a solução otimizada cai na fronteira do espaço de projeto, onde uma pequena variação nos parâmetros do dispositivo afeta o desempenho do circuito. Isso conduz à inclusão de simulação Monte Carlo no circuito de otimização, aumentando o esforço computacional. O objetivo principal deste trabalho é analisar dois métodos diferentes de amostragem, a fim de reduzir o número de rodadas Monte Carlo, e a inserção da heurística de otimização *Particle Swarm Optimization*, visando a minimização do tempo necessário para o dimensionamento do circuito. A amostragem por hipercubo latino, a qual requer um número menor de amostras para um nível de confiança razoável, é utilizado nas primeiras iterações do processo de otimização. Depois de um certo ponto, o método de amostragem é alterado para a amostragem aleatória tradicional. A heurística *Particle Swarm Optimization* foi implementada na ferramenta UCAF, devido ao seu baixo custo computacional. A metodologia é aplicada para o dimensionamento de um amplificador de transcondutância operacional OTA Miller e um amplificador Telescopic, mostrando vantagens em termos de tempo de processamento e desempenho do circuito. Pode-se demonstrar que a utilização de uma nova heurística, e diferentes métodos de amostragem para a simulação Monte Carlo no processo de otimização produz uma busca mais eficiente no espaço de projeto com um ganho em relação ao esforço computacional.

Palavras-chave: Circuitos Integrados Analógicos, Ferramenta UCAF, Particle Swarm Optimization, Monte Carlo

ABSTRACT

The analog integrated circuit design within the microelectronics industry has its evolution dictated by the great need for integration of mixed circuits. This trend makes the semiconductor devices are increasingly miniaturized, which implies the increasing complexity in the manufacturing process, resulting in a large variability of parameters. This complexity is directly linked to the design of devices that compose the circuit, since the design space is highly nonlinear. The design of analog circuit can be modeled as an optimization problem and solved by optimization heuristics. The resulting solution is dependent on modeling strategy and performance estimation, which is done generally by electrical simulation. In this context, the UCAF tool was developed. However, the optimized solution falls on the border of the design space where a small variation in device parameters affect circuit performance. This leads to the inclusion of Monte Carlo simulation on the circuit optimization, increasing the computational effort. The main objective of this study is to analyze two different methods of sampling, in order to reduce the number of Monte Carlo runs, and the inclusion of a new heuristic optimization, in order to minimize the time required for the design of the circuit. The Latin hypercube sampling, which requires a smaller number of samples for a reasonable confidence level is used in the first iteration of the optimization process. After a certain point, the sampling method is changed to the traditional random sampling. Heuristic Particle Swarm Optimization was implemented in UCAF tool, due to its low computational cost. The methodology is applied for the design of a Miller and a Telescopic operational transconductance amplifier, showing advantages in terms of processing time and circuit performance. We can demonstrate that the use of a new heuristic, and different methods of sampling for Monte Carlo simulation in the optimization process produces a more efficient search of the design space, and advantages in relation to computational effort.

Keywords: Analog Integrated Circuits, UCAF Tools, Particle Swarm Optimization, Monte Carlo

LISTA DE ILUSTRAÇÕES

Figura 1 – Fluxograma geral do dimensionamento automático de blocos analógicos.	23
Figura 2 – Fluxo de projeto de circuito integrados analógicos.	30
Figura 3 – Exemplo de 12 amostras com a amostragem aleatória simples.	35
Figura 4 – Exemplo de 12 amostras com a amostragem pelo método de hipercubo latino.	36
Figura 5 – Fluxograma proposto da ferramenta UCAF.	37
Figura 6 – Diagrama de blocos funcionais.	38
Figura 7 – Tipos de Topologia. (a) circuito de entrada e saída simples, (b) circuito de entrada diferencial e saída simples e (c) circuito de entrada e saída diferencial.	40
Figura 8 – Função para as especificações: (a) valor mínimo requerido pelas especificações (b) valor máximo requerido pelas especificações.	42
Figura 9 – Enxame de partículas, ilustrando o comportamento inicial do algoritmo PSO.	45
Figura 10 – Atualização de uma partícula x dentro do espaço de busca.	46
Figura 11 – Atualização das partículas dentro do espaço de busca, convergindo para o mínimo global.	47
Figura 12 – Esquemático OTA Miller.	52
Figura 13 – Resultados encontrados no projeto de um OTA Miller com as especificações dada pela Tabela 1 para uma população variando de 10 a 240 e o número total de iterações igual a 100. a) Função Custo; b) Potência Dissipada; c) Área de gate.	54
Figura 14 – Resultado encontrados no projeto de um OTA Miller com as especificações dada pela Tabela 1 para uma população com tamanho 100 e variando o número de iterações. a) Função Custo; b) Potência Dissipada; c) Área de gate.	55
Figura 15 – Especificações dos valores finais em relação à função custo para uma população com tamanho de 100 e número total de iterações variando de 10 a 240, com cinco valores de semente diferentes.	56
Figura 16 – Esquemático do amplificador operacional Telescopic.	58
Figura 17 – Resultados encontrados no projeto de um amplificador Telescopic com as especificações dado pela Tabela 3 para uma população variando de 10 a 240 e o número total de iterações igual a 100. a) Função Custo; b) Potência Dissipada; c) Área de gate.	60
Figura 18 – Resultado encontrados no projeto de um amplificador Telescopic com as especificações dado pela Tabela 1 para uma população com tamanho 100 e variando o número total de iterações. a) Função Custo; b) Potência Dissipada; c) Área de gate.	61

Figura 19 – Especificações dos valores finais em relação a Função Custo para uma população com tamanho de 100 e o número total de iterações variando de 10 a 240, com cinco valores de semente diferentes.	62
Figura 20 – Fluxo de otimização incluindo estimativa de produtividade apenas para melhor solução encontrada durante o processo de otimização (SEVERO, 2012).	66
Figura 21 – Gráficos da média pelo número de rodadas de simulação Monte Carlo para as especificações do OTA Miller da Tabela 5. a) Av0; b) GBW; c) PM; d) Potência; e) SR.	67
Figura 22 – Gráficos do desvio padrão pelo número de rodadas de simulação Monte Carlo para as especificações do OTA Miller da Tabela 5. a) Av0; b) GBW; c) PM; d) Potência; e) SR.	68
Figura 23 – Gráficos do erro ξ_μ em relação ao número de rodadas de simulação Monte Carlo para os valores de média das especificações. a) Av0; b) GBW; c) PM; d) Potência; e) SR.	70
Figura 24 – Gráficos do erro ξ_σ em relação ao número de rodadas de simulação Monte Carlo para os valores de desvio padrão das especificações. a) Av0; b) GBW; c) PM; d) Potência; e) SR.	71
Figura 25 – Curvas de erro ξ_μ em relação ao número de rodadas de simulação Monte Carlo com método de amostragem por hipercubo latino, referentes às médias das especificações.	73
Figura 26 – Curvas de Erro ξ_μ em relação ao número de rodadas de simulação Monte Carlo com método de amostragem aleatório, referentes às médias das especificações.	74
Figura 27 – Evolução função custo para o melhor indivíduo da população durante o processo de otimização com a técnica de amostragem híbrida e troca em 3000 execuções (LHS/RS-3000).	78
Figura 28 – Produtividade final em relação ao ponto de troca do método de amostragem.	79
Figura 29 – Valor final da função custo em relação ao ponto de troca do método de amostragem.	80
Figura 30 – Evolução função custo para o melhor indivíduo da população durante o processo de otimização.	83

LISTA DE TABELAS

Tabela 1 – Especificações para o dimensionamento do OTA Miller.	53
Tabela 2 – Resultados das especificações para o dimensionamento do OTA Miller utilizando PSO e SA.	57
Tabela 3 – Especificações para o dimensionamento do amplificador Telescopic . . .	59
Tabela 4 – Resultados das especificações para o dimensionamento do amplificador Telescopic, utilizando PSO e SA.	63
Tabela 5 – Valores das variáveis do OTA Miller utilizados para análise dos métodos de amostragem.	65
Tabela 6 – Tabela de diferença percentual dos valores estimados para a média e desvio padrão pelo método hipercubo latino com 10.000 rodadas e os valores de referência.	69
Tabela 7 – Especificações projeto automático do OTA Miller.	72
Tabela 8 – Comparação entre os projetos executados com os diferentes métodos de amostragem Monte Carlo.	76
Tabela 9 – Resultados obtidos com o projeto automático do OTA Miller com a técnica de amostragem híbrida e troca em 3000 execuções (LHS/RS-3000).	77
Tabela 10 – Valores das variáveis do OTA Miller dimensionadas a partir do projeto considerando amostragem híbrida e troca em 3000 execuções (LHS/RS-3000).	77
Tabela 11 – Resultados obtidos com o projeto automático do OTA Miller.	81
Tabela 12 – Valores das variáveis do OTA Miller dimensionadas a partir do projeto considerando amostragem híbrida (2780).	82

SUMÁRIO

1	INTRODUÇÃO	21
1.1	OBJETIVO	23
1.2	ORGANIZAÇÃO DO TRABALHO	23
2	TRABALHOS RELACIONADOS	25
2.1	CONCLUSÃO	26
3	ASPECTOS SOBRE O DIMENSIONAMENTO AUTOMÁTICO DE CIRCUITOS INTEGRADOS ANALÓGICOS	29
3.1	PROJETO DE CIRCUITOS INTEGRADOS ANALÓGICOS	29
3.2	DESIGN CENTERING	31
3.2.1	SIMULAÇÃO MONTE CARLO	33
3.2.1.1	AMOSTRAGEM ALEATÓRIA SIMPLES	33
3.2.1.2	AMOSTRAGEM POR HIPERCUBO LATINO	34
3.3	A FERRAMENTA UCAF	37
3.3.1	INTERFACE DE ENTRADA	37
3.3.2	NÚCLEO DA FERRAMENTA	38
3.3.3	TECNOLOGIA DE FABRICAÇÃO	39
3.3.4	BIBLIOTECA DE TOPOLOGIAS	39
3.3.5	OTIMIZAÇÃO	39
3.3.5.1	ALGORITMOS GENÉTICOS (AG)	40
3.3.5.2	SIMULATED ANNEALING (SA)	41
3.3.6	FUNÇÃO CUSTO	41
3.3.7	ESPECIFICAÇÕES	43
3.3.8	SIMULAÇÃO ELÉTRICA	43
3.3.9	INTERFACE DE SAÍDA	43
3.4	PARTICLE SWARM OPTIMIZATION	43
3.4.1	ALGORITMO BASE	44
3.4.2	MODIFICAÇÕES DO ALGORITMO PSO	47
3.5	CONCLUSÃO	49
4	ANÁLISE DO ALGORITMO PSO NA FERRAMENTA UCAF	51
4.1	OTA MILLER	51
4.2	TELESCOPIC	57
4.3	CONCLUSÃO	62
5	ANÁLISE DA TÉCNICA HÍBRIDA DE AMOSTRAGEM MONTE CARLO	65
5.1	MÉTODOS DE AMOSTRAGEM	65
5.2	MÉTODO HÍBRIDO DE AMOSTRAGEM	71

5.3	CONCLUSÃO	83
6	CONCLUSÃO	85
	REFERÊNCIAS	87

1 INTRODUÇÃO

A microeletrônica tem uma história longa, comparado ao seu curto período de existência. O primeiro transistor foi fabricado em 1947 pela Bell Labs, o qual propiciou a fabricação dos primeiros circuitos integrados em 1959 na Fairchild, resultando nos primeiros circuitos comercializados em 1962 (BRINKMAN; HAGGAN; TROUTMAN, 1997). Com o seu desenvolvimento, a microeletrônica vem contribuindo para o crescimento e a massificação da tecnologia da informação. Neste contexto, a indústria de semicondutores representa uma grande importância no desenvolvimento tecnológico mundial.

Os circuitos integrados são circuitos eletrônicos desenvolvidos em escalas micro-métricas ou nanométricas sobre elementos semicondutores, normalmente o silício. Eles surgiram como uma solução para o problema da miniaturização e do consumo de energia. Este desenvolvimento possibilitou que os circuitos eletrônicos se tornassem cada vez menores, assim possibilitando a integração de mais dispositivos dentro de uma mesma área, o que significa uma capacidade de processamento cada vez maior.

Em 1965 foi apresentada a Lei de Moore, a qual segundo Gordon Moore (1965), um dos fundadores da Intel, previu que o número de transistores que poderiam ser fabricados dentro de um único chip dobraria a cada ano.

Neste contexto, a tecnologia de fabricação de circuitos integrados teve um grande desenvolvimento, chegando a integrar bilhões de transistores em um único chip, hoje em dia. Além disso, a tendência e o objetivo de desenvolver sistemas em um único chip que reúna tanto circuitos digitais quanto analógicos se tornou cada vez maior.

Na atualidade, a maioria dos sistemas integrados são digitais, porém os circuitos analógicos estão constantemente presentes na composição dos circuitos integrados, seja para a interface, geração de sinais ou polarização de circuitos. No que tange ao dimensionamento dos transistores, 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, nos quais cada transistor deve ser dimensionado individualmente. Entretanto, os circuitos digitais, em relação ao número de transistores, são mais complexos, pois podem possuir até alguns bilhões a mais que os analógicos.

O projeto misto de circuitos integrados é composto por partes analógicas e partes digitais, que apresentam uma grande disparidade na sua construção, pois a parte digital é realizada de forma totalmente automática, por meio de ferramentas de CAD (*Computer Aided Design*), ao contrário da analógica que conta com ferramentas de CAD apenas para simulação (GIELEN; RUTENBAR, 2000).

Nos projetos dos circuitos analógicos, os projetistas precisam construir cada circuito e subcircuito a ser utilizado, de forma que cada componente integrado, seja ele um capacitor, resistor, transistor ou até mesmo um indutor, precisa ser dimensionado

individualmente.

A variabilidade resultante do processo de fabricação é um ponto importante dentro do projeto de circuitos analógicos. Como os dispositivos tendem a diminuir de tamanho com a evolução do processo de fabricação, a variabilidade dos parâmetros se torna cada vez mais significativa, podendo inclusive modificar completamente as características elétricas dos dispositivos. A produtividade é o número de circuitos que atendem às especificações em relação ao total de circuitos fabricados. Ela é afetada diretamente pela variabilidade dos parâmetros do processo (LIN; CHEN; GEIGER, 2006; XU et al., 2009). Sendo assim, a estimativa da produtividade do circuito durante a etapa de projeto é fundamental para maximizar a sua funcionalidade em aplicações reais. Esta previsão é estimada geralmente através de simulação Monte Carlo, a qual é lenta e acarreta um grande impacto computacional na simulação do projeto (YU; LI, 2011).

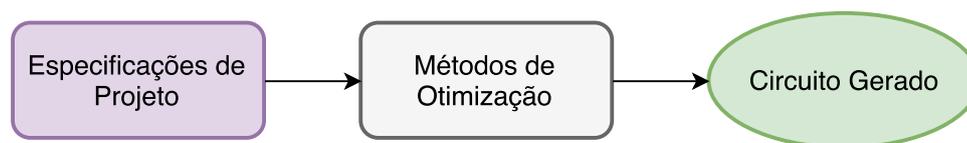
Neste contexto, torna-se necessário o estudo e o desenvolvimento de uma ferramenta para o projeto de circuitos integrados analógicos que considere a estimativa de produtividade. A ferramenta desenvolvida no Grupo de Arquitetura de Computadores e Microeletrônica (GAMA) da UNIPAMPA, denominada UCAF, tem como objetivo dimensionar automaticamente blocos analógicos integrados explorando de maneira eficiente o espaço de projeto. A Figura 1 apresenta o fluxograma geral para uma ferramenta para automação de circuitos integrados analógicos. Nesta ferramenta busca-se otimizar a área dos circuitos, a dissipação de potência e a sua produtividade (SEVERO, 2012). A ferramenta UCAF transforma o problema do dimensionamento de transistores em um problema de otimização. A otimização recai em um problema de programação não-linear e requer a formulação de uma função simples de estimativa de desempenho (função custo) a ser minimizada, sujeita a um conjunto de restrições, conforme a forma padrão demonstrada na Equação 1.1.

$$\underset{(\mathbf{p}, \mathbf{q})}{\text{minimizar}} F_m(\mathbf{p}, \mathbf{q}) = 1, \dots, M \text{ sujeito a } C_n(\mathbf{p}, \mathbf{q}) \leq C_{n(ref)}, n = 1, \dots, N \quad (1.1)$$

Onde M é o número total de F_m especificações a otimizar e N é o número de C_n funções restrição de desempenho. Os parâmetros de projetos são representado por \mathbf{p} e o parâmetros da tecnologia de fabricação por \mathbf{q} . A estimativa de desempenho e de produtividade é realizada através de simulação elétrica SPICE. Cada iteração do processo de otimização requer a repetição destas simulações, o que torna o procedimento bastante lento.

A motivação deste trabalho é diminuir o impacto no custo computacional da estimativa de desempenho e produtividade, levando em conta a implementação de uma nova heurística de otimização e o estudo de métodos alternativos de amostragem para simulação Monte Carlo.

Figura 1 – Fluxograma geral do dimensionamento automático de blocos analógicos.



Fonte: Elaborado pelo autor.

Uma heurística de otimização bastante eficiente é a PSO, do inglês *Particle Swarm Optimization*, proposto por Kennedy e Eberhart (1995), a qual foi incluída na ferramenta UCAF para explorar o espaço de projeto. Também foram realizados estudos sobre a aplicação do método de amostragem aleatória simples, comumente usada no método Monte Carlo, comparando-o à amostragem por hipercubo latino (LHS, do inglês *Latin Hypercube Sampling*) (MCKAY; BECKMAN; CONOVER, 1979).

A aplicação da heurística PSO na ferramenta UCAF, juntamente com os diferentes métodos de amostragem para simulação Monte Carlo, para o projeto de amplificadores operacionais do tipo Miller e Telescopic, demonstrou-se bastante eficiente, com resultados superiores a heurísticas tradicionais, como Simulated Annealing e Algoritmos Genéticos, em termos de desempenho final do amplificador e de produtividade do circuito gerado.

1.1 OBJETIVO

Este trabalho tem por objetivo o aprimoramento da ferramenta UCAF, que visa a automação do projeto de circuitos integrados analógicos através de heurísticas de otimização, de forma a acelerar o processo de busca de soluções otimizadas para as especificações do circuito, incluindo a análise de produtividade.

A estimativa de produtividade do circuito projetado requer o estudo dos métodos de amostragem na simulação Monte Carlo, visando diminuir o impacto computacional.

Além disso, este trabalho também tem como objetivo verificar os efeitos da implementação da heurística PSO na ferramenta UCAF e a comparação com heurísticas mais tradicionais, como Simulated Annealing.

1.2 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizada em 6 capítulos. No Capítulo 2 são apresentados os trabalhos que motivaram o estudo e abordagem desta dissertação. O Capítulo 3 apresenta a revisão bibliográfica necessária para o estudo e desenvolvimento do projeto. O Capítulo 4 está relacionado à implementação da heurística PSO na ferramenta UCAF. O Capítulo 5 trata da análise da técnica híbrida de amostragem Monte Carlo na ferramenta UCAF. Por fim, o Capítulo 6 apresenta as conclusões deste trabalho.

2 TRABALHOS RELACIONADOS

Este capítulo apresenta o estado da arte no desenvolvimento de ferramentas de automação do projeto de circuitos integrados analógicos incluindo análise de produtividade.

Jafari et al. (2010) utiliza a heurística Algoritmos Genéticos para realizar a síntese automática de circuitos integrados analógicos. Com o intuito de reduzir o tempo de desenvolvimento e otimizar o circuito, a estimativa de desempenho é realizada através de equações analíticas simplificadas. Para avaliar o algoritmo, é apresentado um projeto de um amplificador operacional de transcondutância (OTA) CMOS de dois estágios em processo de $0,18 \mu\text{m}$. Os resultados obtidos demonstraram 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 alguns aspectos do circuito são desconsiderados.

O trabalho de Canelas et al. (2016), apresenta uma técnica na redução do número de amostras na análise Monte Carlo, visando um maior ganho na produtividade utilizando o algoritmo de agrupamento K-means. Este algoritmo seleciona aleatoriamente um ponto de dados como um centro de agrupamento. Os centros seguintes são selecionados a partir do resto dos pontos de dados de acordo com uma probabilidade que é diretamente proporcional à distância entre o ponto de dados e os centros já definidos. Esta técnica utiliza simulação elétrica para avaliação do desempenho do circuito e foi integrada à ferramenta de otimização de dimensionamento de circuitos chamada AIDA-C. Os testes foram realizados em um amplificador operacional de um estágio na tecnologia $130\mu\text{m}$. A técnica de estimativa de produtividade proposta reduziu o número de simulações Monte Carlo requeridas durante o processo de otimização em 75% e obteve uma produtividade final do circuito de 90%.

Mahmoudi e Zimmermann (2015) combinaram o método Monte Carlo e a *Response Surface Modeling* para analisar a produtividade de circuitos analógicos. Visando melhorar o desempenho computacional em relação à estratégia de amostragem empregada no Monte Carlo, foi abordado o estudo da técnica de amostragem por Hipercubo Latino (LHS), reduzindo o número de amostras necessárias para atingir certa precisão. Eles propõem um método para construir projetos otimizados visando proporcionar um alto desempenho em experimentos Monte Carlo. Este método utiliza Algoritmos Genéticos de Permutação (PermGA) com base em um esquema de expansão do cromossomo (CLE). Através dos exemplos de circuitos, foi possível comprovar que a técnica CLE proposta proporciona uma grande melhoria em termos de esforço computacional necessário em comparação com os PermGA comuns. A vantagem computacional aumenta exponencialmente quando o número de variáveis de entrada aumenta.

O trabalho de Jaffari e Anis (2011) demonstra o método de amostragem por Hipercubo Latino (LHS) que tem sido utilizado como principal meio na redução de

variância na simulação Monte Carlo para uma análise na variabilidade de parâmetros no projeto de circuitos analógicos. Ele estuda uma classe para o método LHS que controle as correlações cruzadas lineares ou quadráticas de amostras para reduzir ainda mais a variância das estimativas de rendimento, pois verificou-se que no LHS os vetores de amostras uniformes são gerados permutando aleatoriamente a ordem das amostras em cada dimensão, levando a indesejadas correlações cruzadas lineares ou quadráticas entre um par de amostras. Uma classe de algoritmos de permutação foi estudada para o problema de análise de rendimento que reduz as correlações de pares de amostras. A classe desenvolvida demonstrou eficiência em relação à redução do número necessário de amostras para a estimativa da produtividade do circuito analógico.

Guerra-Gomez, Tlelo-Cuautle e Fraga (2015) abordaram em seu trabalho uma estratégia baseada em OCBA (*Optimization on Computational Budget Allocation*), com o intuito de reduzir o custo computacional para o cálculo da produtividade no projeto de circuitos analógicos quando utilizado simulações Monte Carlo. O OCBA é aplicado para realizar a distribuição das simulações entre os casos críticos de otimização do dimensionamento de circuitos analógicos. Ele também consegue limitar as simulações para os casos não críticos. Os projetos são realizados em um amplificador operacional de transcondutância usando tecnologia de 90nm, e aplicado em três heurísticas de otimização diferentes: *Non-dominated Sorting Genetic Algorithm* (NSGA-II), *Multi-Objective Evolutionary Algorithm with Decomposition* (MOEA/D) e o *Particle Swarm Optimization* (PSO). A estratégia proposta baseada na abordagem OCBA demonstrou uma melhora na produtividade de até 1,8 vezes e uma redução no número de simulações de até 85%, assim demonstrando que a análise de variação é bastante importante dentro do projeto.

O trabalho apresentado por Barros, Guilherme e Horta (2010b) descreve o desenvolvimento de uma ferramenta de automação para o dimensionamento de circuitos integrados analógicos. A ferramenta desenvolvida, denominada GENOM, utiliza a heurística Algoritmos Genéticos em seu núcleo, que por meio de simulações elétricas busca um resultado otimizado para o projeto de circuitos analógicos. A técnica de Monte Carlo é aplicada na ferramenta visando a melhoria na produtividade do circuito, como também a identificação dos pontos de pior caso a serem utilizados posteriormente no ciclo de dimensionamento. A ferramenta de otimização resultante trouxe melhorias na produtividade de projeto de amplificadores operacionais CMOS.

2.1 CONCLUSÃO

Os trabalhos citados neste capítulo evidenciaram as diferentes estratégias adotadas em relação ao dimensionamento automático de circuitos analógicos, como também a consideração da produtividade nesses projetos. Outro ponto importante de se notar é em relação as técnicas de redução da variância quando calculado a produtividade, visando

reduzir o tempo computacional do fluxo de projeto, por meio de estratégias de amostragem Monte Carlo.

3 ASPECTOS SOBRE O DIMENSIONAMENTO AUTOMÁTICO DE CIRCUITOS INTEGRADOS ANALÓGICOS

Este capítulo analisa o estado da arte do desenvolvimento da ferramenta UCAF, e uma introdução sobre projetos de circuitos integrados. Nesta análise, o projeto de circuitos analógicos será brevemente contextualizado e, em seguida, a ferramentas UCAF será descrita.

Neste capítulo, é mostrada também a técnica de *design centering* e uma descrição sobre a heurística *Particle Swarn Optimization*.

3.1 PROJETO DE CIRCUITOS INTEGRADOS ANALÓGICOS

Segundo Graeb (2007) nos circuitos integrados os blocos analógicos representam a etapa mais crítica dentro de um projeto. Este projeto é realizado, em sua grande maioria, de forma manual por um projetista, o qual utiliza de algumas ferramentas computacionais de auxílio ao projeto (CAD), como por exemplo, ambientes de simulações e editores de leiaute. Assim, o projetista necessita de um vasto conhecimento para a escolha e o projeto da topologia a ser implementada.

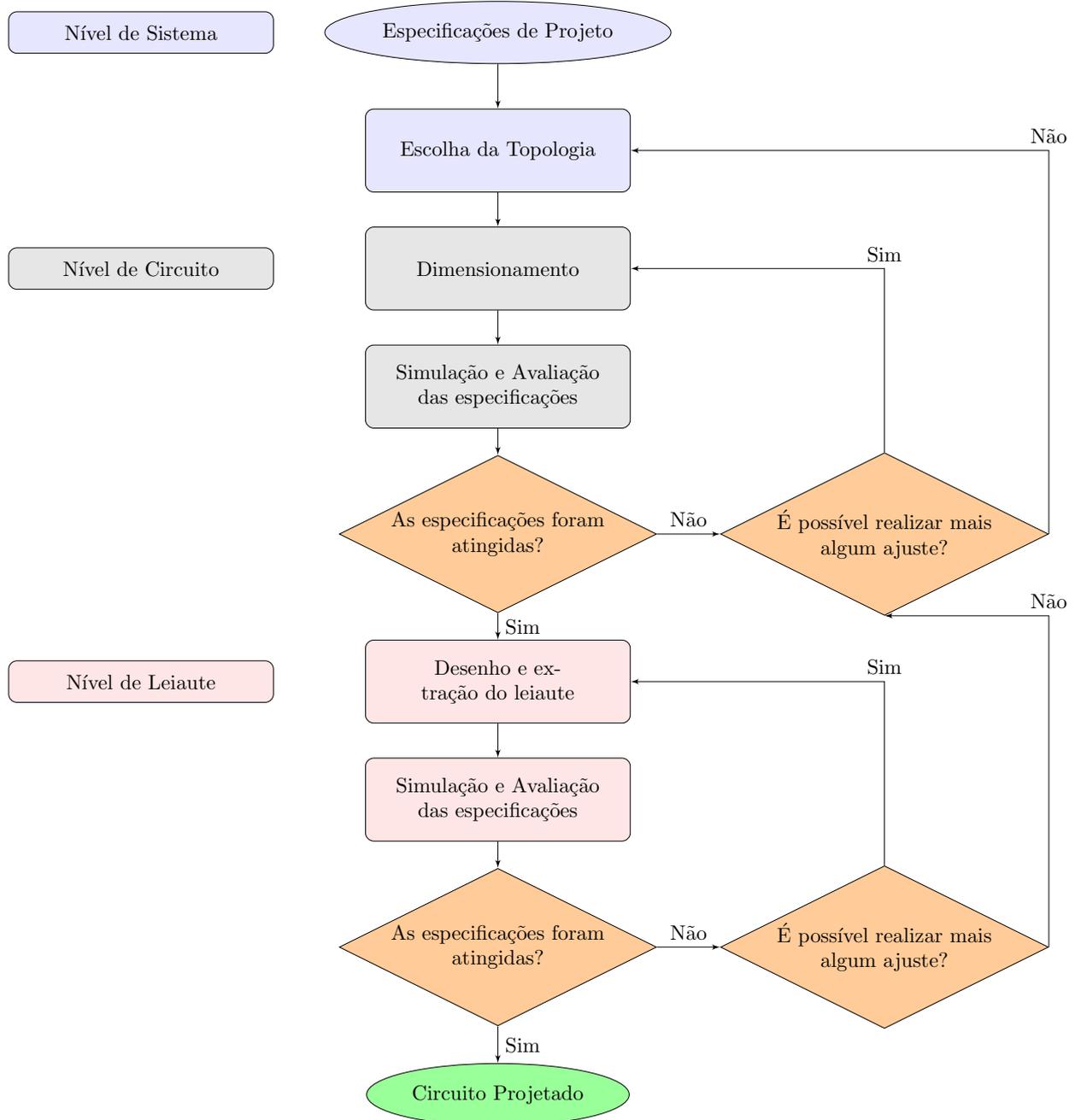
O projeto de um circuito integrado analógico pode ser dividido em três grandes níveis de abstração, que são (BALKIR; DUNGAR; OGRENCI, 2003):

- Nível de Sistema - no nível de sistema devem ser estudadas as especificações do circuito, e, a partir disto, é escolhida a topologia mais adequada para o circuito;
- Nível de Circuito - neste nível são dimensionados os transistores que compõem o circuitos e as suas fontes de polarização, tomando como base a topologia escolhida. O dimensionamento de um transistor CMOS consiste em determinar o comprimento (W) e a largura (L) do dispositivo. Nesta etapa são realizadas as simulações e a avaliação das especificações;
- Nível de leiaute - neste nível, com as especificações de projetos atingidas, o leiaute do circuito deve ser realizado. Para isso, utiliza-se de editores pré-configurados baseados nas regras de projeto fornecidas pela *foundry*. Assim, o projetista analisa a estratégia de desenho para que o circuito mantenha suas características elétricas.

Nestas etapas normalmente o projetista utiliza metodologias simplificadas de projeto, e conta com auxilio das ferramentas CAD e simuladores elétricos do tipo SPICE. Estas etapas sofrem ajustes até as especificações desejadas serem atingidas. Caso não seja possível atingi-las, volta-se à etapa inicial com a escolha de uma nova topologia. A Figura 2 mostra o fluxo completo de um projeto de circuito integrado analógico, no qual deve

haver algumas etapas como simulações do circuito sujeito a variabilidades no processo de fabricação, como análises de Monte Carlo e de pior caso.

Figura 2 – Fluxo de projeto de circuito integrados analógicos.



Fonte: Adaptado de Severo (2012).

Conforme analisado, o projeto de circuito integrado analógico depende de uma série de passos e decisões tomadas manualmente pelo projetista. Na etapa de dimensionamento, um dos pontos mais críticos do projeto devido às metodologias como um todo utilizarem equações simplificadas é que os primeiros resultados tendem a não representar os resultados

reais. Tendo isto em vista, temos que o grande problema das técnicas utilizadas nos projetos é que o resultado depende fortemente da experiência do projetista.

3.2 DESIGN CENTERING

O desempenho dos circuitos integrados é cada vez mais afetado pelos fenômenos de variabilidade dentro do seu processo de fabricação, tais como as variações de parâmetros de processo, mudanças de temperatura, ruído da fonte de alimentação, entre outros. Estes fenômenos podem levar a diferenças significativas entre circuitos concebidos e fabricados. Em virtude da redução das dimensões dos componentes fabricados, o impacto das variações tende a aumentar a cada nova tecnologia desenvolvida. Porém, as tolerâncias de processo não acompanham na mesma proporção esta redução. Nos circuitos analógicos, a variabilidade de parâmetros pode levar à perda da funcionalidade do circuito, levando o circuito a operar em um nível de polarização diferente e assim sofrendo variações no casamento de dispositivo (DRENNAN; MCANDREW, 2003).

A variabilidade de parâmetros no processo de fabricação de um circuito integrado pode ser classificada como (ORSHANSKY; NASSIF; BONING, 2010):

- *Front-end*: estas variações ocorrem nas etapas de fabricação do circuito, como oxidação, construções das linhas de polisilício, entre outras. Também podem ocorrer variações nas dimensões da largura (W) e comprimento (L) do canal dos transistores e na espessura do óxido de silício (T_{ox}), entre outras.
- *Back-end*: são caracterizadas pela variação de processo causadas pela metalização das vias e interconexões do circuito integrado, sofrendo alterações nos diversos níveis de metais.

A variabilidade de parâmetros imposta pelo circuito integrado tem como diferença os valores nominais e reais de operação de um determinado circuito. Esta variabilidade pode ser corrigida na fase do projeto do circuito, de forma a diminuir o impacto das variações no processo de fabricação sobre o comportamento elétrico do circuito.

Dentro do processo de fabricação dos dispositivos temos dois tipos de variações, que, segundo Mutlu e Rahman (2005), são:

- Variações Globais (*inter-die*): provocadas por meio de flutuações no processo de fabricação que afetam todos os dispositivos do circuito da mesma maneira. Ela é independente do comprimento L e da largura W do canal dos transistores.
- Variações Locais (*intra-die*): afeta cada transistor de um mesmo circuito individualmente. Neste caso, a variação do comprimento L depende diretamente da largura W do dispositivo.

Estas variações são influenciadas diretamente pelo processo de fabricação dos circuitos. Deste modo, é importante analisar o comportamento do circuito desde as primeiras fases do projeto, através de um modelo elétrico e de simulações elétricas do tipo SPICE. As simulações elétricas definem um modelo determinístico, de forma que para o mesmo circuito sempre serão encontrados as mesmas especificações. Por outro lado, no processo de fabricação o modelo sofre variações. Sendo assim, seus valores na prática são considerados como não determinísticos. Portanto, para a análise do circuito através de simulação elétrica é importante ter o conhecimento da variação de parâmetros do modelo, estimadas por análises estatísticas que são geradas pela caracterização do processo de fabricação. A partir disto, algumas especificações podem ser avaliadas em relação à variabilidade de parâmetros.

Neste contexto, (GRAEB, 2007) define que com a simulação elétrica é possível estimar a variação de parâmetros de um circuito utilizando a análise Monte Carlo, em que o circuito é analisado por meio de uma distribuição estatística. A análise Monte Carlo gera números aleatórios dentro de uma determinada faixa de variação conhecida para cada parâmetro através de uma função de densidade de probabilidade (FDP). O cálculo da FDP é realizado através da distribuição normal ou gaussiana, que obedece à Equação 3.1.

$$FDP = \frac{1}{\sqrt{2.\pi}.\sigma} . e^{-\frac{(x-\mu)^2}{2.\sigma^2}} \quad (3.1)$$

Aqui, σ é o desvio padrão e μ é a média do valor do parâmetro.

Esta densidade da probabilidade proporciona a análise da produtividade Y do circuito, que é calculada de acordo com a Equação 3.2.

$$Y = \frac{N^\circ \text{ de circuitos válidos}}{N^\circ \text{ total de circuitos fabricados}} . 100\% \quad (3.2)$$

Portanto, a produtividade de um circuito pode ser estimada com a análise Monte Carlo por meio de inúmeras simulações elétricas, através das quais o comportamento elétrico pode ser estimado e a produtividade é calculada pela percentagem dos circuitos que atendem ao comportamento desejado.

A produtividade de um circuito pode ser otimizada. Esta otimização recebe o nome de *design centering*. Neste caso, o dimensionamento do circuito leva em consideração a variabilidade dos parâmetros da tecnologia de fabricação e visa encontrar uma solução em que a produtividade seja maior possível. Assim, o circuito terá o comportamento elétrico esperado mesmo nas piores condições de variação dos parâmetros.

3.2.1 SIMULAÇÃO MONTE CARLO

A simulação Monte Carlo foi desenvolvida na segunda Guerra Mundial, proposta por Von Neumann e Ulam, com o propósito de solucionar problemas matemáticos nos quais o tratamento analítico não se demonstrava viável (SOBOL, 1994).

O método da simulação Monte Carlo atualmente pode ser referenciado como um método de simulação estatística, que utiliza sequências de números aleatórios para realizar suas simulações, ou seja, é um método numérico universal desenvolvido para resolver problemas por meio de amostragem (BINDER; HEERMANN, 1992).

O Monte Carlo pode resolver inúmeros problemas. Estes problemas por si necessitam estar modelados em termos de funções de densidade de distribuição da probabilidade. Através destas distribuições são gerada amostras para a simulação Monte Carlo. Este processo pode se repetir inúmeras vezes e o seu resultado é definido através de técnicas estatísticas, que são dependentes do número de realizações (amostras) determinadas em uma simulação Monte Carlo, podendo ter dezenas ou milhares de iterações até que se chegue em um resultado satisfatório (BINDER; HEERMANN, 1992).

Portanto, a função de densidade de probabilidade gera os valores de base para a simulação Monte Carlo. Estes, por sua vez, são dependentes do tipo de amostragem utilizada para a simulação. Dentro do Monte Carlo existem vários tipos de amostragem, onde a principal diferença entre eles é a forma com que os valores são gerados, pois devido a essa diferença são necessárias mais ou menos rodadas de simulações para haver uma análise com precisão.

Com isso, as técnicas de amostragem buscam obter um maior controle nas amostras necessárias para a simulação Monte Carlo. A seguir serão apresentadas a Amostragem Aleatória Simples e a Amostragem por Hipercubo Latino.

3.2.1.1 AMOSTRAGEM ALEATÓRIA SIMPLES

A amostragem aleatória simples é o tipo de amostragem utilizada como padrão na simulação Monte Carlo, que consiste em um conjunto de números sorteados ao acaso dentro de um determinado conjunto. O método de amostragem aleatória simples gera suas amostras de n pontos para cada variável de entrada de um modelo, sendo esses pontos selecionados aleatoriamente dentro de um conjunto N . Com um número suficiente de iterações, a amostragem aleatória simples recria a distribuição de entrada através da amostragem.

No método de amostragem simples um problema de agrupamento surge quando há um número pequeno de amostras. De acordo com o Teorema do Limite Central, este problema está relacionado com o conceito de erro padrão da média, que é encontrado através do desvio padrão da distribuição de entrada dividido pela raiz quadrada do número

de iterações por simulações.

O valor esperado para uma variável aleatória p é $\mu = E(\mathbf{v})$. Se forem gerados valores v_1, \dots, v_n de forma independente e aleatória na distribuição v , pode-se estimar μ como

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n v_i.$$

Uma das grandes vantagens do método Monte Carlo é que os próprios valores amostrados podem ser usados para estimar o erro $\hat{\mu}_n - \mu$. O erro médio quadrado na amostragem Monte Carlo é σ^2/n . A estimativa mais usada para o desvio padrão σ^2 é:

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (v_i - \hat{\mu}_n)^2.$$

A amostragem Monte Carlo tipicamente necessita de um número alto de amostras (n) para garantir que a estimativa seja uma boa aproximação para o valor real de variância σ^2 . A estimativa de σ^2 indica que o erro está na ordem de σ/\sqrt{n} .

Do Teorema do Limite Central, sabe-se que $\hat{\mu}_n - \mu$ possui aproximadamente uma distribuição normal com média 0 e variância σ^2/n . Este teorema pode ser usado para uma aproximação no cálculo de intervalo de confiança c de μ . De modo geral:

$$\mu_c = \hat{\mu}_n \pm \Phi^{-1}(1 - \alpha/2) \frac{\sigma}{\sqrt{n}}, \quad (3.3)$$

onde $\alpha = 1 - c/100$ e $\Phi(\cdot)^{-1}$ é a função distribuição cumulativa inversa (ICDF) da distribuição padrão normal $N(0, 1)$.

Os valores gerados pela amostragem aleatória simples tendem a ser amostrados próximos à média, gerando uma pequena tendência de valores afastados da média para um número pequeno de amostras (KHOUAS; DERIEUX, 2000).

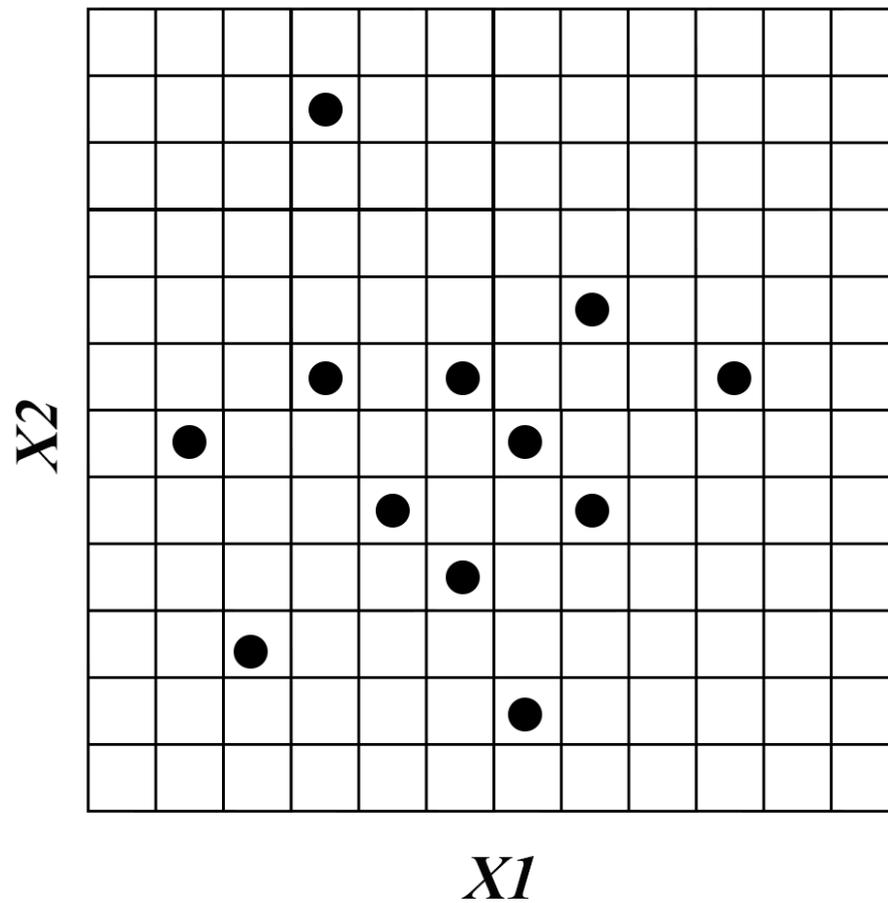
A Figura 3 mostra um exemplo de como funciona o sorteio por amostragem aleatória simples. Neste exemplo temos um conjunto qualquer, onde cada ponto representa uma amostra dentro de um conjunto de 12 amostras.

As simulações Monte Carlo baseadas no método de amostragem simples tendem a representar melhor as variações de um circuito, porém gasta um número maior de simulações, o que tende a afetar o ganho computacional de um projeto. A precisão na simulação Monte Carlo pode ser considerada proporcional à raiz quadrada do número de amostras, enquanto a sua complexidade cresce linearmente com o número de amostras (GRAEB, 2007).

3.2.1.2 AMOSTRAGEM POR HIPERCUBO LATINO

A técnica de amostragem Hipercubo Latino (LHS) impõe um controle maior no processo de amostragem comparado com as demais técnicas desenvolvidas e mantém a ideia

Figura 3 – Exemplo de 12 amostras com a amostragem aleatória simples.



Fonte: Elaborado pelo autor.

inicial de possuir um paradigma de amostras aleatórias em seus valores. Sua amostragem é extraída por uma distribuição acumulada de probabilidade das variáveis de entrada do modelo de simulação, onde a simulação é dividida em n partes de igual probabilidade, e posteriormente é realizada a escolha de um elemento dentro de cada estrato. Como há uma permutação destes valores, todos os estratos estarão representados dentro da amostra (MCKAY; BECKMAN; CONOVER, 1979).

Portanto, esta técnica consiste na distribuição de probabilidade das variáveis de entrada da simulação em n partes iguais de probabilidade. Os valores são escolhidos aleatoriamente dentro de cada parte. A Equação 3.4 é utilizada para a geração dos valores hipercúbicos a serem permutados.

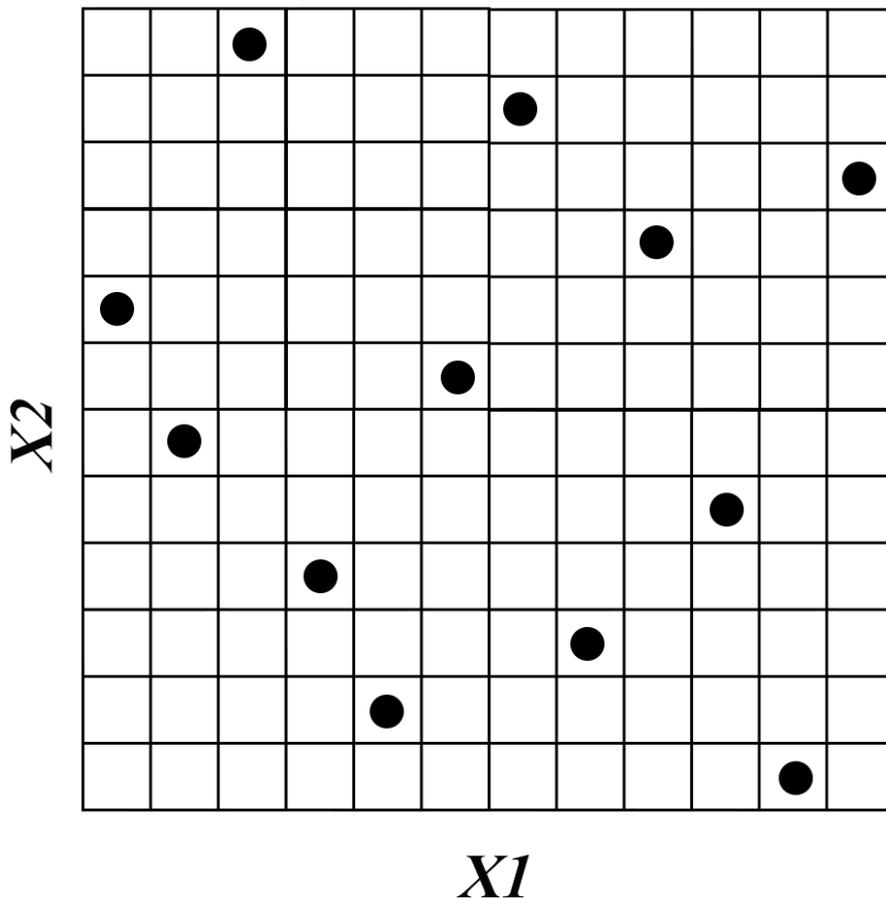
$$hc_i = f^{-1}\left(\frac{i - rand_i}{n}\right) \quad (3.4)$$

Onde:

- hc_i Valor que compõe a amostragem hipercúbica.
 i Iteração corrente.
 f^{-1} Inversa da função de distribuição.
 $rand_i$ Número aleatório no intervalo de $[0, 1]$.
 n Tamanho da amostra.

A Figura 4 mostra um exemplo de como funciona o sorteio por amostragem com a técnica hipercubo latino. Neste exemplo temos um sistema qualquer, onde cada ponto representa uma amostra, dentro de um conjunto de 12 amostras.

Figura 4 – Exemplo de 12 amostras com a amostragem pelo método de hipercubo latino.



Fonte: Elaborado pelo autor.

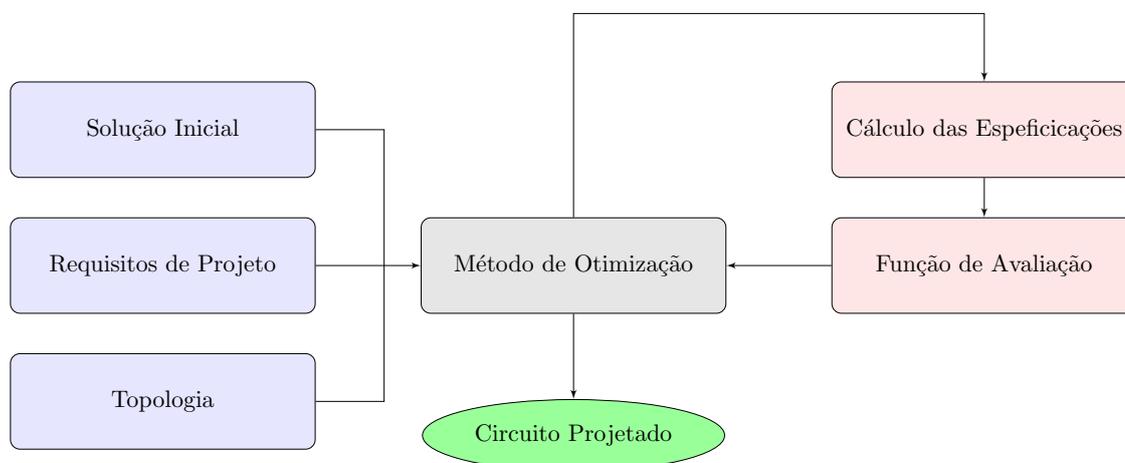
Neste exemplo pode-se observar que a amostragem estratifica a distribuição de probabilidade de entrada, na qual há uma divisão da distribuição em intervalos com probabilidades iguais de sorteio, e seleciona um valor aleatório pertencente a cada um dos intervalos.

3.3 A FERRAMENTA UCAF

Este trabalho é baseado na ferramenta UCAF, desenvolvido pelo Grupo de Arquitetura de Computadores e Microeletrônica da UNIPAMPA. Assim, esta seção tem como base o trabalho de Severo (2012), o qual descreve o desenvolvimento e comportamento básico deste sistema.

A ferramenta UCAF é baseada em uma metodologia para o dimensionamento de circuitos integrados analógicos através de otimização e utiliza simulações elétricas para estimativa de desempenho do circuito. A Figura 5 apresenta o fluxograma da ferramenta UCAF. Conforme esta figura, inicialmente é realizado o levantamento dos requisitos do projeto, escolhida a tecnologia de fabricação e gerada uma solução inicial (podendo ou não ser aleatória). Posteriormente, o método de otimização recebe os requisitos de projeto. A cada solução gerada, o método de otimização recebe estas informações e atribui valores às variáveis do circuito. Com estes valores são realizados os cálculos das especificações, e assim a análise da função de avaliação. O método de avaliação se repete até que uma solução otimizada seja encontrada e que seus requisitos de projetos sejam atendidos.

Figura 5 – Fluxograma proposto da ferramenta UCAF.



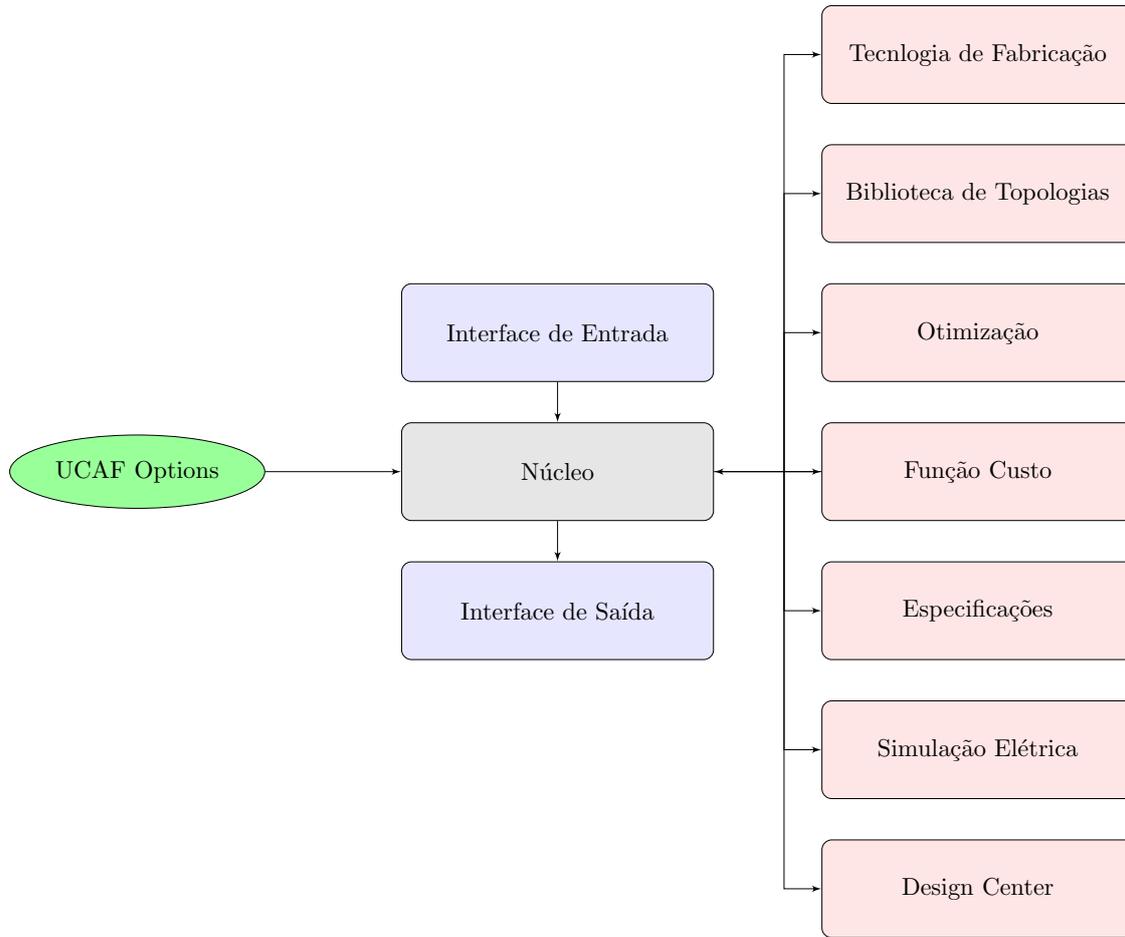
Fonte: Adaptado de Severo (2012).

A ferramenta foi implementada em blocos funcionais modulares, conforme mostra a Figura 6, sendo que cada bloco pode ser configurado para melhor ajuste de fluxo da UCAF. Esta implementação foi realizada no Matlab[®] e será detalhada nas subseções seguintes.

3.3.1 INTERFACE DE ENTRADA

A interface de entrada serve como entrada do núcleo da UCAF e é realizada por linha de comando, através de um *script*, sendo utilizada para realizar as configurações

Figura 6 – Diagrama de blocos funcionais.



Fonte: Adaptado de Severo (2012).

da ferramenta, onde são inseridos todos os dados necessários para a execução da mesma. Portanto, assim que definidas as configurações no *script*, a ferramenta pode ser executada.

Estas configurações são dadas através do ajuste da estrutura *UCAF Options* que contém todos os dados necessários para a execução da ferramenta.

3.3.2 NÚCLEO DA FERRAMENTA

O núcleo da ferramenta, como seu nome sugere, é a parte principal da UCAF, pois este bloco está ligado a todos os demais blocos funcionais. Ele é responsável pela leitura da entrada, por acionar e por configurar todos os demais blocos para a automação do projeto, baseando-se na estrutura *UCAF Options*, que representa as configurações de entrada.

3.3.3 TECNOLOGIA DE FABRICAÇÃO

Este bloco representa a tecnologia de fabricação responsável pela inserção e configuração do modelo elétrico dos transistores CMOS, das dimensões mínimas e máximas dos transistores, como também os parâmetros utilizados no modelo. A escolha da tecnologia de fabricação é de grande importância para o dimensionamento do circuito, pois ele é responsável por aproximar o comportamento elétrico dos dispositivos, por meio de simulações elétricas. A escolha da tecnologia de fabricação pode influenciar na solução encontrada pela heurística de otimização, pois dependendo de como ele efetua a modelagem dos transistores, podem ocorrer descontinuidades nas curvas características do comportamento elétrico do dispositivo.

A entrada de dados do modelo é incluído nas simulações SPICE executadas pela ferramenta. Estas entradas são dadas através de um ou mais arquivos, dependendo da tecnologia utilizada, sendo estes dados configurado na interface de entrada.

3.3.4 BIBLIOTECA DE TOPOLOGIAS

A biblioteca de topologias representa o bloco onde estão contidas as características do circuito integrado analógico que se deseja projetar automaticamente, de forma a encontrar uma solução otimizada para as especificações e restrições do circuito.

A ferramenta UCAF foi desenvolvida para aceitar qualquer tipo de circuito integrado analógico que se deseja dimensionar. As variáveis livres de projeto são consideradas conforme o número de incógnitas do circuito a ser projetado.

Dentro da ferramenta UCAF foram definidas três formas de circuito, conforme demonstrado na Figura 7 (ALLEN; HOLBERG, 2002), para a inserção das topologias.

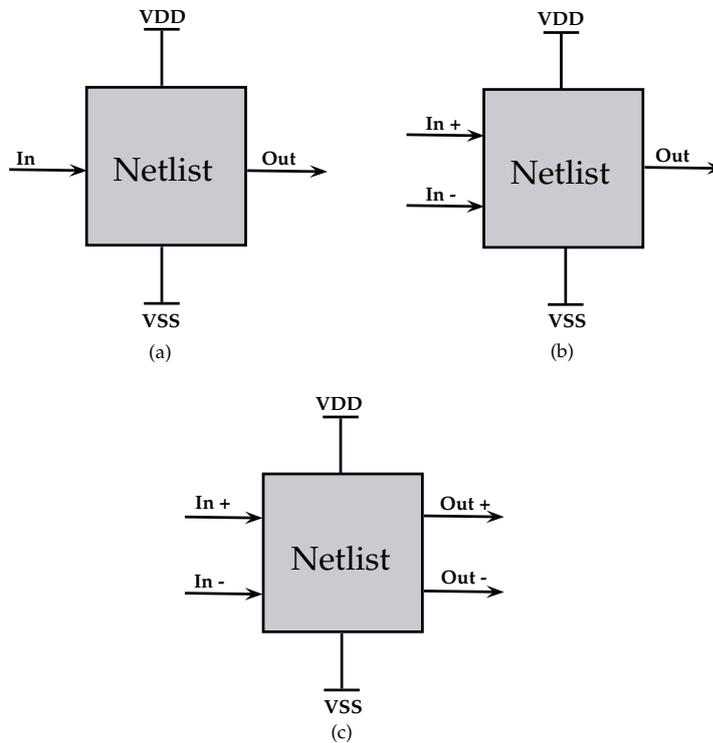
O tipo de circuito é caracterizado pelo tipo de *netlist* a ser dimensionado, que é estabelecido como um subcircuito. Este subcircuito é definido pelo padrão do bloco, que é determinado pelas entradas, saídas e fontes de alimentação.

3.3.5 OTIMIZAÇÃO

O bloco de otimização se caracteriza por conter as heurísticas responsáveis por gerar possíveis soluções otimizadas para a topologia do circuito. A partir de soluções iniciais pré-definidas, é executada a heurística até que se encontre uma solução otimizada para as variáveis livres de projeto, de forma que estas soluções possuam valores definidos dentro de um limite máximo e mínimo baseado na topologia empregada. A avaliação das soluções é realizada por meio de uma função custo (a qual será analisada a seguir).

Na ferramenta UCAF foram inseridas duas heurísticas, sendo elas Algoritmos Genéticos (AG) e Simulated Annealing (SA). Estas heurísticas são definidas na interface

Figura 7 – Tipos de Topologia. (a) circuito de entrada e saída simples, (b) circuito de entrada diferencial e saída simples e (c) circuito de entrada e saída diferencial.



Fonte: Adaptado de Severo (2012).

de entrada pelo usuário, as quais serão exemplificadas nos tópicos 3.3.5.1 e 3.3.5.2.

3.3.5.1 ALGORITMOS GENÉTICOS (AG)

Os Algoritmos Genéticos (AG) são definidos como uma classe de algoritmos de otimização que se baseiam na evolução biológica (HOLLAND, 1975). Conforme Digalakis e Margaritis (2001), o AG é uma meta-heurística de busca global que simula os processos biológicos para gerar soluções a determinados problemas. Estas soluções são denominadas de indivíduo e cada um deles, por sua vez é associado a um valor da função custo.

As iterações no AG são indivíduos de uma determinada população que são cruzados entre si, gerando novos indivíduos que herdam as características do pai, de forma que cada novo indivíduo possa ou não sofrer mutações. Sua avaliação é realizada por meio de uma função custo, que também é denominada *fitness* (MITCHELL, 1998).

Cada indivíduo na população dentro do AG é chamado de cromossomo, que pode ser representado como uma possível solução. Um cromossomo é uma sequência de codificação binária ou real, que evolui através de sucessivas iterações, denominadas geração. Nos cromossomos binários as variáveis livres de otimização são codificadas em números binários, onde o número de bits irá depender da resolução de cada uma das

variáveis. Porém, antes de realizar o cálculo para a sua avaliação é necessário efetuar a decodificação. Já os cromossomos reais não necessitam da codificação e decodificação, pois eles são vistos como um vetor real, e suas posições são definidas pelas variáveis reais do problema (MITCHELL, 1998).

A implementação dos Algoritmos Genéticos na ferramenta UCAF deu-se através de um toolbox para Matlab[®] denominado *Genetic Algorithm Optimization Toolbox* (GAOT), desenvolvido por Houck, Joines e Kay (1995). O GAOT possibilita uma vasta configuração dentro do algoritmo, como o tipo de cromossomos - que podem ser real ou binário -, o tamanho da população e o número de gerações a serem executadas. Todas as demais configurações referentes ao GAOT podem ser encontradas através de Houck, Joines e Kay (1995).

3.3.5.2 SIMULATED ANNEALING (SA)

O *Simulated Annealing* (SA) é uma meta-heurística utilizada para solucionar problemas de otimização não-linear, inspiradas na analogia de sistemas termodinâmicos, que simula o resfriamento de átomos aquecidos até o ponto mínimo de energia (OSMAN; KELLY, 1996).

A cada iteração do SA são gerados valores para de uma nova solução baseada na solução corrente. Sempre que encontrada uma solução melhor que a corrente ela é aceita, porém soluções não tão boas podem ser aceitas também com o intuito de escapar de mínimos locais. Tudo isso depende do parâmetro de temperatura, que define a probabilidade de aceitação de soluções piores. Quando sua temperatura estiver alta, existe uma chance maior de aceitar uma solução não tão boa, e ao passo que estando baixas estas soluções são rejeitadas mais facilmente. Como sua temperatura é decrementada até zero, a cada iteração realizada as soluções convergem para um conjunto de soluções locais (GLOVER; KOCHENBERGER, 2003).

O SA tem a opção de ser configurado com busca híbrida, de forma que utiliza de otimização local e global simultaneamente. Com essa opção é possível encontrar uma solução local mais próxima da solução atual (LIU et al., 2009).

A implementação do *Simulated Annealing* na ferramenta UCAF foi realizada através da função *Simulated Annealing* que está disponível no *Global Optimization Toolbox* do Matlab[®] (MATWORKS, 2012b). Esta função permite que a heurística seja totalmente configurada pelo usuário.

3.3.6 FUNÇÃO CUSTO

A função custo é a parte mais importante da otimização, pois com base em seus resultados cada solução gerada é avaliada. Na ferramenta UCAF a função custo foi

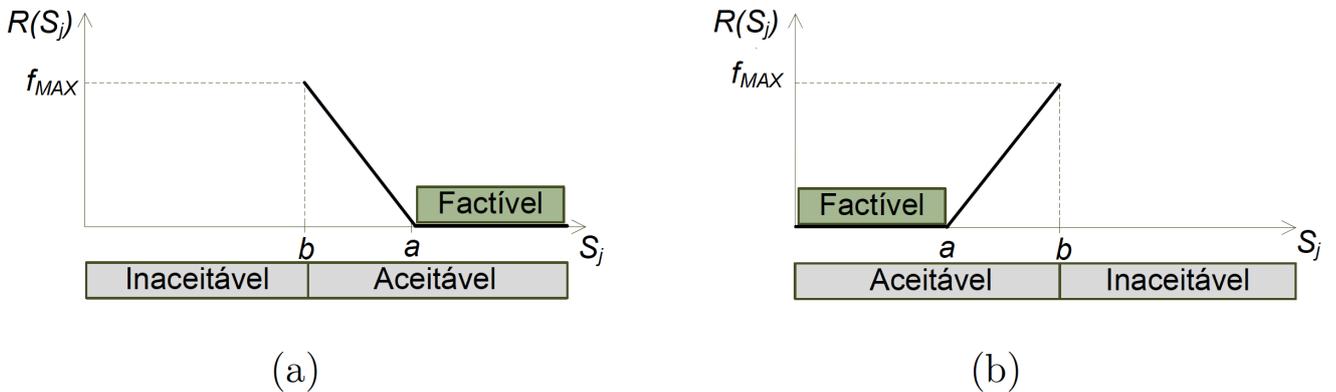
projetada de forma que o usuário deve informar as especificações que serão otimizadas e as que restringem o espaço do projeto.

A função custo utilizada na UCAF é uma função que avalia as soluções das restrições de projeto como também as especificações que se desejam minimizar ou maximizar. A função é similar à utilizada em Phelps et al. (2000), e é dada pela Equação 3.5, onde $S_n(X)$ a função da especificação, que depende do vetor X de variáveis livres, PO_n e PR_m são os pesos das especificações de objetivo e das especificações de restrição, que devem ser positivas e diferentes de zero. Estas especificações deverão ser definidas pelo usuário.

$$f_c(X) = \sum_{n=1}^N PO_n \cdot S_n(X) + \sum_{m=1}^M PR_m \cdot R(S_m(X), S_{mref}) \quad (3.5)$$

O valor de restrição da função custo $R(S_m(X), S_{mref})$ é calculado em função do valor desejado S_{mref} e do valor alcançado $S_m(X)$ de uma determinada especificação. Para o cálculo de $R(S_m(X), S_{mref})$ foram utilizadas as funções conforme a Figura 8, de acordo com Barros, Guilherme e Horta (2010a), onde o valor de a é dado pelo valor máximo ou mínimo de S_{mref} , e o valor de b representa o valor limite de aceitação (BARROS; GUILHERME; HORTA, 2010a).

Figura 8 – Função para as especificações: (a) valor mínimo requerido pelas especificações (b) valor máximo requerido pelas especificações.



Fonte: Adaptado de Severo (2012).

A função custo utiliza os dados de entrada do bloco de especificação, onde o usuário define o que são as especificações de objetivo e as especificações de restrição. Com a definição dos pesos PO_n e PR_m , as especificações relevantes do projeto são escolhidas e ponderadas. Esta ponderação é uma parte fundamental da modelagem do projeto, pois existem inúmeras combinações possíveis, tornando-se difícil a escolha dos valores de ponderação.

3.3.7 ESPECIFICAÇÕES

Dentro deste bloco estão contidas as especificações de uma determinada topologia de circuito. Estes valores servem como base para o dimensionamento dos transistores. Na ferramenta UCAF foram definidos dois tipos de especificações, que são:

- Especificações de restrição - devem satisfazer valores máximos ou mínimos definidos pelo usuário;
- Especificações de otimização - devem ser minimizadas (ou maximizadas).

Os valores das especificações são estimados através de simulação elétrica SPICE, com exceção da área de gate que é calculada em relação às dimensões do dispositivo. Porém para cada especificação ou conjunto de especificações são necessários *testbenches* de simulação, de maneira que estes dependem do tipo de circuito a ser projetado e de suas características.

3.3.8 SIMULAÇÃO ELÉTRICA

Neste bloco estão armazenados os dados obtidos através das simulações elétricas, que são do tipo SPICE, como também os *testbenches* de simulação, que posteriormente serão utilizados para a avaliação das especificações do circuito a ser projetado.

Os *testbenches* de circuito utilizam análises AC, DC e transiente para realizarem a medição das especificações, portanto todas as especificações do circuito são obtidas através de simulação elétrica.

A utilização do simulador elétrico na ferramenta UCAF é executada por linha de comando, de maneira que é possível a conexão com um simulador externo. Com as simulações executadas os dados necessários para extrair as especificações são gerados. A ferramenta UCAF utiliza o simulador elétrico Synopsys HSPICE[®].

3.3.9 INTERFACE DE SAÍDA

Com todo o fluxo da ferramenta completo, por fim temos a interface de saída. Nela estão armazenados os dados do dimensionamento de um determinado circuito, como as soluções intermediárias e finais, os valores de especificações de projeto, e todos os demais dados da execução, para serem realizadas análises por parte do usuário.

3.4 PARTICLE SWARM OPTIMIZATION

O Particle Swarm Optimization (PSO) foi proposto por Kennedy e Eberhart (1995), como um modelo para otimização de funções contínuas não-lineares, implementada

com base no comportamento social da interação entre indivíduos (partículas) de um grupo (enxame). O PSO é uma heurística fundamentada no comportamento coletivo de animais e insetos, que é baseada na coordenação de seus movimentos em tarefas como a busca e obtenção de alimentos (HEPPNER; GREANDER, 1990) (REYNOLDS, 1987). As primeiras implementações do PSO foram inspiradas em bandos de pássaros, nos quais verificou-se que o comportamento dos animais é influenciado pela experiência de cada indivíduo, tanto na busca e obtenção por alimentos quanto no deslocamento para um ponto desejado.

O enxame no PSO é constituído na interação dos seus indivíduos (partículas), e estruturado para que cada indivíduo seja capaz de se comunicar com os subconjuntos de seus pares, denominados de vizinhos, que podem ser inicializados aleatória ou estaticamente. A cada iteração a partícula se movimenta pelo espaço de solução com uma determinada velocidade, e a cada iteração é realizada uma avaliação em relação à solução correspondente à posição ocupada. Sua velocidade sofre alterações em relação à melhor posição de todas as partículas encontradas naquele instante e em relação à melhor posição já encontrada pela própria partícula até então no espaço de busca. 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.

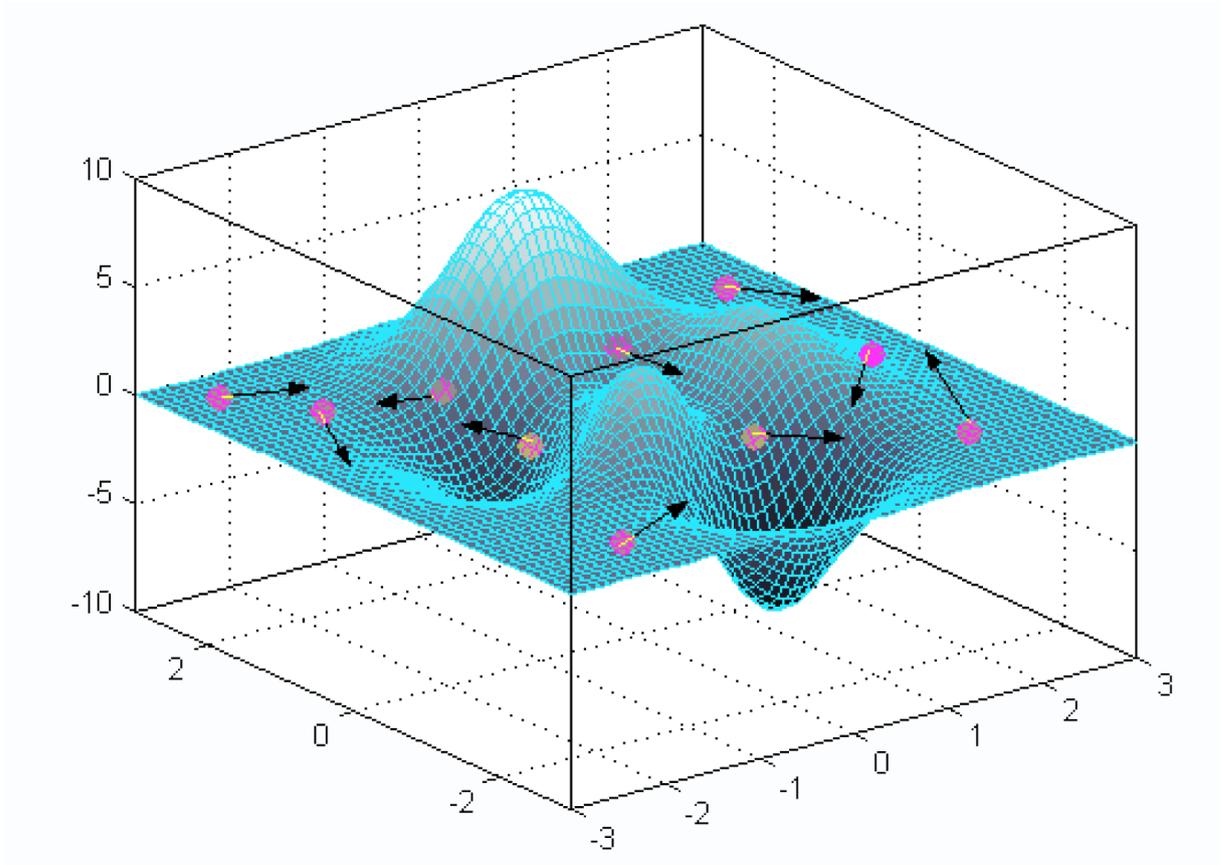
O PSO é uma meta-heurística relativamente recente. Apesar disso, ela conseguiu alcançar um relevante espaço dentre as demais meta-heurísticas devido à sua simplicidade e poder de processamento (POLI; KENNEDY; BLACKWELL, 2007). O PSO, ao contrário de outras técnicas, necessita apenas dos valores nos pontos avaliados e de operadores matemáticos para o cálculo da função objetivo. Desta forma, o PSO é considerado uma meta-heurística relativamente robusta, e este fato o torna uma técnica de fácil implementação (PARSOPOULOS; VRAHATIS, 2002).

3.4.1 ALGORITMO BASE

O PSO tem características parecidas com algumas técnicas evolucionárias, como por exemplo os Algoritmos Genéticos, visto anteriormente. Seus sistemas são inicializados como uma população aleatória ou pré-definidas, e tendem a procurar um resultado ótimo ao decorrer das gerações. O PSO encontra suas soluções por meio das partículas, que têm a sua inicialização de forma aleatória. A Figura 9 mostra a inicialização do PSO (MEDEIROS, 2005).

Portanto, o PSO inicia suas soluções aleatoriamente, e então procura por uma melhor solução a cada geração, e esta é melhorada seguindo seus dois "melhores" valores. Estes valores são pontos importantes dentro do algoritmo, e são denominados o melhor local (P_{best}), que é a melhor posição da partícula encontrada na iteração i , e o melhor

Figura 9 – Enxame de partículas, ilustrando o comportamento inicial do algoritmo PSO.



Fonte: Adaptado de MEDEIROS (2005).

global ($Gbest$), que é a melhor posição já encontrada por qualquer outra partícula da população, ou seja, melhor resultado encontrado em todo espaço do problema até então.

Ao encontrar o melhor local e global, a partícula atualiza sua velocidade e posição conforme a Equação 3.6 (KENNEDY; EBERHART, 1995).

$$vel_x(i+1) = vel_x(i) + C1 \cdot rand1 \cdot (Pbest_x - P_x(i)) + C2 \cdot rand2 \cdot (Gbest - P_x(i)) \quad (3.6)$$

A cada iteração, a posição da partícula é atualizada conforme a Equação 3.7.

$$P_x(i+1) = P_x(i) + vel_x(i+1) \quad (3.7)$$

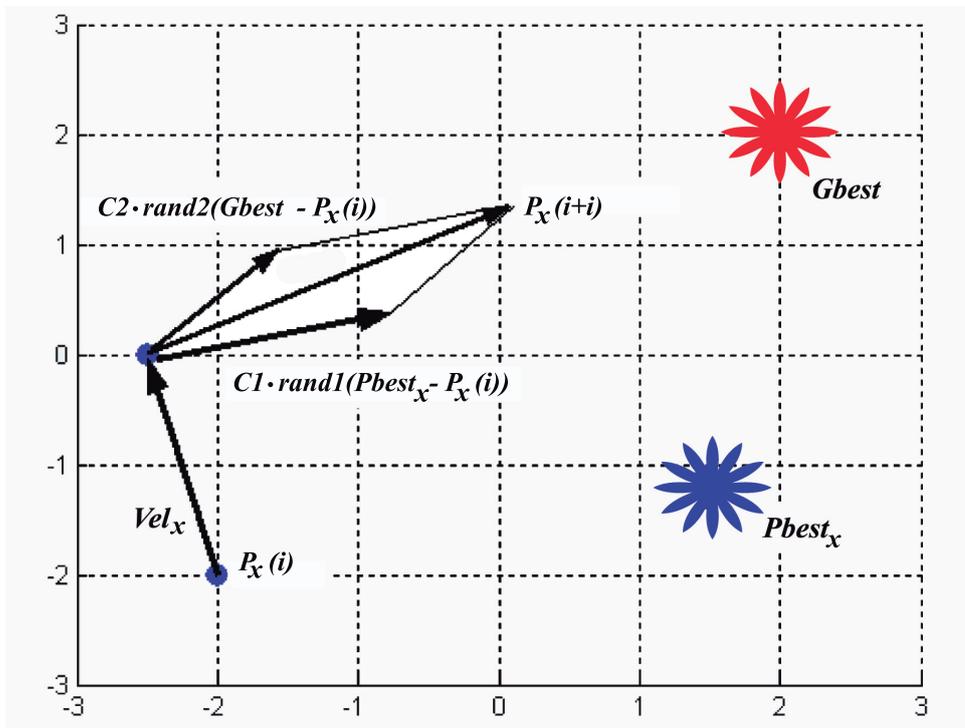
Onde:

A Figura 10 ilustra a atualização de uma determinada partícula dentro do espaço de busca, sendo que cada partícula atualiza a sua velocidade levando em conta a melhor

vel_x	Velocidade da partícula x .
i	Iteração corrente.
$C1$ e $C2$	Constantes de aceleração.
$rand1$ e $rand2$	Números aleatórios no intervalo de $[0, 1]$.
$Pbest_x$	Melhor posição local da partícula x .
$Gbest$	Melhor posição global.
P_x	Posição atual da partícula x .

posição dela mesma e do enxame.

Figura 10 – Atualização de uma partícula x dentro do espaço de busca.



Fonte: Elaborado pelo autor.

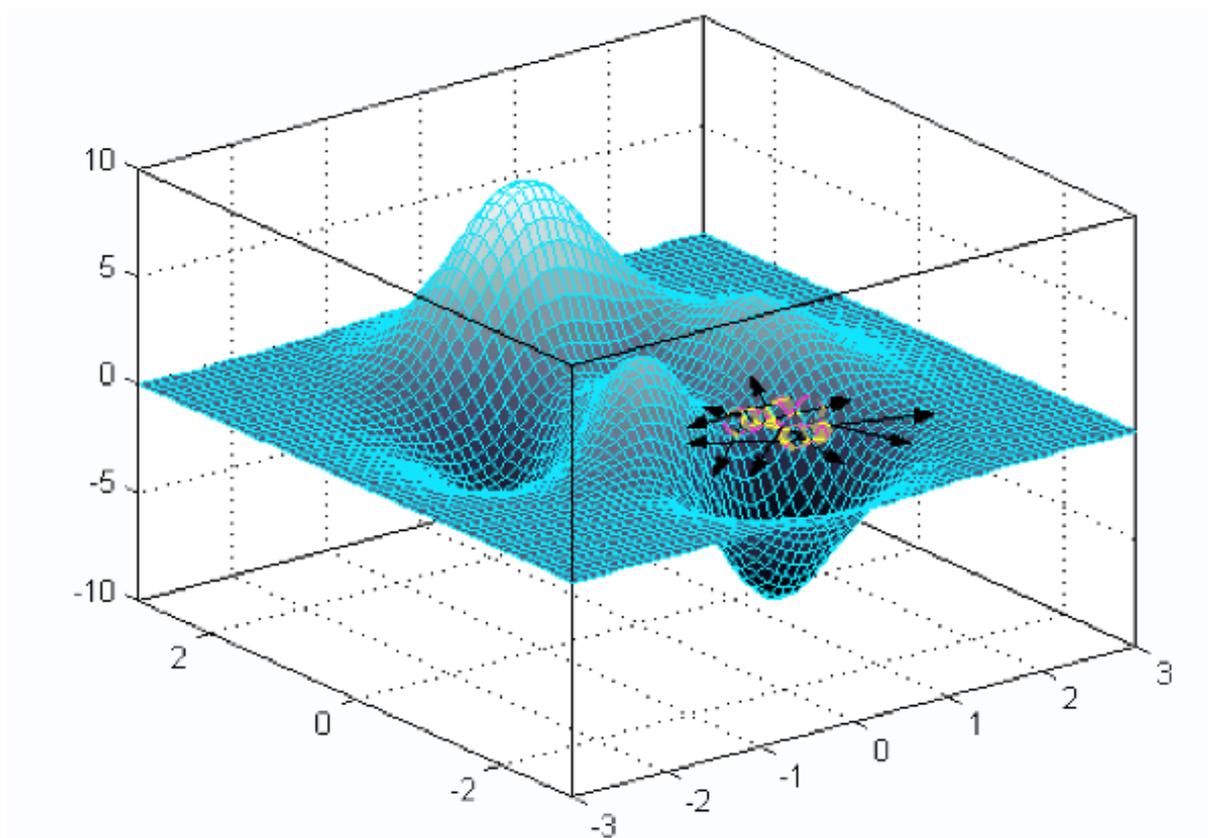
Com a Equação 3.6 a velocidade de cada partícula é calculada, e com isso suas posições são atualizadas a cada iteração movendo-se no espaço de projeto. Ao longo de sucessivas iterações, com a influência dos resultados da decisão individual e global, o enxame de partícula converge para um ponto ótimo, conforme demonstra a Figura 11.

O Algoritmo 1 demonstra o pseudocódigo do algoritmo PSO.

Existem dois passos fundamentais ao aplicar o PSO em problemas de otimização, que é a representação da solução e o cálculo da velocidade.

A busca pelo resultado ótimo dentro de um espaço de otimização é um processo de repetição, de forma que os critérios de parada se dão pelo número máximo de execuções,

Figura 11 – Atualização das partículas dentro do espaço de busca, convergindo para o mínimo global.



Fonte: Adaptado de MEDEIROS (2005).

ou se há alguma condição de parada que satisfaça o problema. No PSO há dois parâmetros para serem ajustados de forma a satisfazer o número máximo de execuções. Um deles é o número de iterações, que pode variar muito, mas que tende a ser de 20 a 40 para problemas mais simples e de 100 a 200 para problemas com uma maior complexidade. Outro é a dimensão do enxame, ou seja, quantas partículas existem dentro de uma determinada população, o que é definido pelo tamanho da população. Esta é determinada conforme o problema a ser otimizado.

As constantes de aceleração $C1$ e $C2$ são determinadas empiricamente e normalmente são iguais a 2. No entanto, outros valores podem ser utilizados na faixa entre 0 e 4.

3.4.2 MODIFICAÇÕES DO ALGORITMO PSO

O PSO apresenta algumas deficiências dentro da sua capacidade de busca. Uma delas é a tendência do enxame perder diversidade e convergir a um ótimo local de maneira prematura (POLI; KENNEDY; BLACKWELL, 2007).

Algorithm 1 Pseudocódigo do algoritmo PSO.

```

1: Gera populao inicial.
2: Gera o valor inicial para a velocidade.
3: Obtm os valores iniciais para Pbestx e Gbest.
4: function PSO
5:   for  $x < MAXITE$  do
6:     Para cada partcula calcula o valor da funo custo. ▷ Eq. 3.5
7:     Atualiza Pbestx e Gbest.
8:     Calcula a velocidade para cada partcula. ▷ Eq. 3.6
9:     Atualiza a posio de cada partcula. ▷ Eq. 3.7
10:  end for
11: end function

```

O PSO, por utilizar de busca local e global, beneficia na resolução de vários problemas, pois há uma troca de informação ao utilizar de ambas as buscas para o encontro de seu resultado. No entanto, existe um diferente equilíbrio entre as capacidades de busca local e global, acarretando que o algoritmo pode não encontrar resultados satisfatórios. Pensando nisso, Shi e Eberhart (1998) adicionaram um parâmetro de inércia w na Equação 3.6, 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. Portanto, o cálculo da velocidade pode ser atualizado conforme a Equação 3.8.

$$vel_x(i + 1) = w_i \cdot vel_x(i) + C1 \cdot rand1 \cdot (Pbest_x - P_x(i)) + C2 \cdot rand2 \cdot (Gbest - P_x(i)) \quad (3.8)$$

Shi e Eberhart (1999) definiram uma função não-linear para calcular o coeficiente de inércia w , a fim de melhorar o desempenho do PSO, conforme demonstrado na Equação 3.9.

$$w_i = (w_{inicial} - w_{final}) \cdot \frac{(MAXITER - i)}{MAXITER} + w_{final} \quad (3.9)$$

Onde:

w_i	Coeficiente de inércia.
$w_{inicial}$	Valor inicial da inércia.
w_{final}	Valor final da inércia.
$MAXITER$	Número máximo de iterações.
i	Valor corrente da iteração.

Estudos realizados, segundo Shuhua et al. (2005), tem mostrado que o peso da inércia w tem grande influência na capacidade de busca global e local no PSO, pois quanto

maior o peso da inércia, mais poderosa é a capacidade de busca global, e conseqüentemente mais fraca a capacidade de busca local. Portanto, um bom equilíbrio entre o $w_{inicial}$ e o w_{final} é essencial dentro de um problema não-linear. Assim, o estudo destes valores para determinados problemas é de grande relevância para a qualidade da solução final.

3.5 CONCLUSÃO

Neste capítulo foi apresentada a ferramenta UCAF, desenvolvida por Severo (2012), que serve como base para este trabalho. A ferramenta é dividida em blocos, o que proporciona a sua evolução, pois facilita a implementação de novas heurísticas, como a PSO. Apresentou-se também a técnica de *design centering*, onde foram abordados o tema Monte Carlo e seus dois métodos de amostragem.

4 ANÁLISE DO ALGORITMO PSO NA FERRAMENTA UCAF

Com o objetivo de implementar a heurística PSO na ferramenta UCAF, este capítulo apresenta uma bateria de testes com o intuito de encontrar o melhor ajuste da heurística, visto que ela tem alguns pontos que podem ser configurados para o seu melhor funcionamento.

O PSO, como visto anteriormente, busca soluções no espaço de projeto por meio de suas partículas, que são atualizadas através da Equação 3.8. Esta, por sua vez, tem alguns valores preestabelecidos para seu cálculo, que são as constantes $C1$ e $C2$, fixados com o valor 2. Este valor foi adotado por Kennedy e Eberhart (1995), com base em estudos realizados. Dentro da Equação 3.8 temos o cálculo do coeficiente da inércia, baseado na Equação 3.9, que por sua vez necessita de dois valores preestabelecidos pelo usuário, $w_{inicial}$ e w_{final} , valores fixados em 0,4 e 0,9 respectivamente. Estes valores são baseados no estudo desenvolvido por Shuhua et al. (2005).

A heurística PSO utiliza como critério de parada o número total de execuções definido pelo usuário, conforme a Equação 4.1. Na implementação da heurística na UCAF foram realizados inúmeros testes, com o objetivo de encontrar o melhor ajuste possível para adaptá-lo ao problema do dimensionamento automático de blocos analógicos.

$$Total\ execuções = número\ total\ de\ iterações \cdot tamanho\ da\ população \quad (4.1)$$

A seguir são apresentados os testes e resultados encontrados para definir a melhor configuração para o PSO. Foram utilizados como referência dois tipo de amplificadores operacionais, o amplificador Miller e o Telescopic (ALLEN; HOLBERG, 2002).

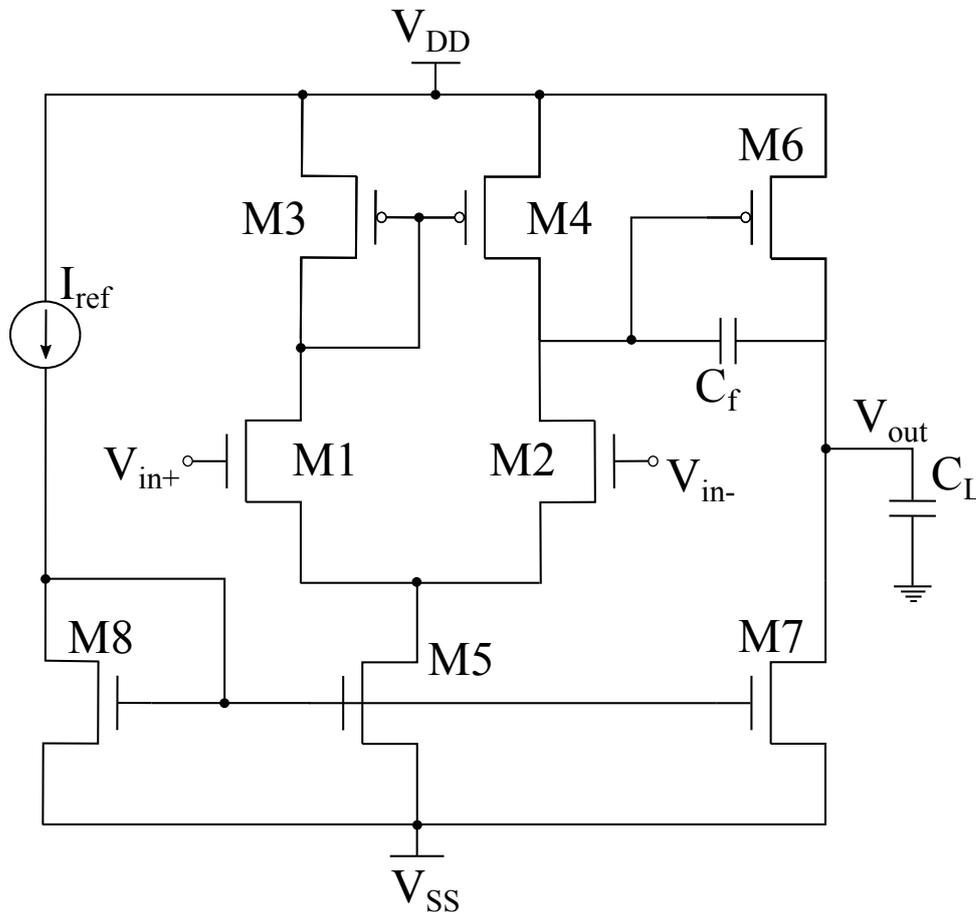
4.1 OTA MILLER

O OTA Miller é composto por dois estágios de ganho. Seu esquemático é apresentado na Figura 12. Seu primeiro estágio é um amplificador diferencial e o segundo é um amplificador inversor. A conexão entre eles é realizada através de um capacitor de compensação (C_f), visando a melhor estabilidade do circuito (ALLEN; HOLBERG, 2002). As principais especificações de desempenho do OTA Miller são:

- GBW - produto ganho-faixa;
- $Av0$ - ganho de tensão em baixas frequências;
- PM - margem de fase;
- $ICMR$ - faixa de tensão de entrada em modo comum;

- SR - velocidade de resposta - slew rate;
- OS - faixa de tensão de saída.

Figura 12 – Esquemático OTA Miller.



Fonte: Elaborado pelo autor.

No projeto do OTA Miller seus oito transistores precisam ser dimensionados de forma a obter os valores dos parâmetros de largura (W) e comprimento (L). Algumas igualdades entre os transistores podem ser atribuídas, devido ao matching. Sendo assim, os espelhos de corrente (M3-M4 e M5-M8) e o par diferencial (M1-M2) devem ser composto por transistores iguais. Portanto, serão dimensionados cinco transistores, ao invés dos oito. O dimensionamento do OTA Miller apresenta 12 variáveis livres, sendo elas: W_{M1} , L_{M1} , W_{M3} , L_{M3} , W_{M5} , L_{M5} , W_{M6} , L_{M6} , W_{M7} , L_{M7} , além da fonte de corrente (I_{ref}) e capacitância de compensação (C_f). O projeto apresenta algumas constantes, que são: fonte de alimentação $V_{DD} = |V_{SS}|$, que recebeu o valor de $0,9\text{ V}$ e uma capacitância de carga (C_L) que foi ajustada em 3 pF . A tecnologia de fabricação utilizada para o projeto, foi a $XFAB\ 0,18\ \mu\text{m}$, que define os valores mínimos para os dispositivos que são $L = 0,18\ \mu\text{m}$ e $W = 0,22\ \mu\text{m}$. Com os valores mínimos de cada variável foram definidas as faixas de

variações das variáveis livres que são: $L = 0,18 \mu\text{m}$ a $100 \mu\text{m}$, $W = 0,22 \mu\text{m}$ a $200 \mu\text{m}$, $I_B = 1 \mu\text{A}$ a $300 \mu\text{A}$ e $C_f = 1 \text{pF}$ a 100pF .

A função custo foi configurada com todos os fatores de ponderação de restrições iguais a 1 e os valores requeridos iguais aos valores mostrados na segunda coluna da Tabela 1. A área de gate e a potência dissipada são objetivos de projeto, e as demais especificações são restrições de projeto.

Tabela 1 – Especificações para o dimensionamento do OTA Miller.

Especificações	Valor Requerido
Av0 (dB)	≥ 70.00
GBW (MHz)	≥ 5.00
PM ($^\circ$)	≥ 60.00
SR (V/ μs)	≥ 5.00
ICMR+ (V)	≥ 0.70
ICMR- (V)	≤ -0.70
Pdiss (μW)	Minimizar
Área de gate (μm^2)	Minimizar
Função Custo	Minimizar

Fonte: Elaborado pelo autor.

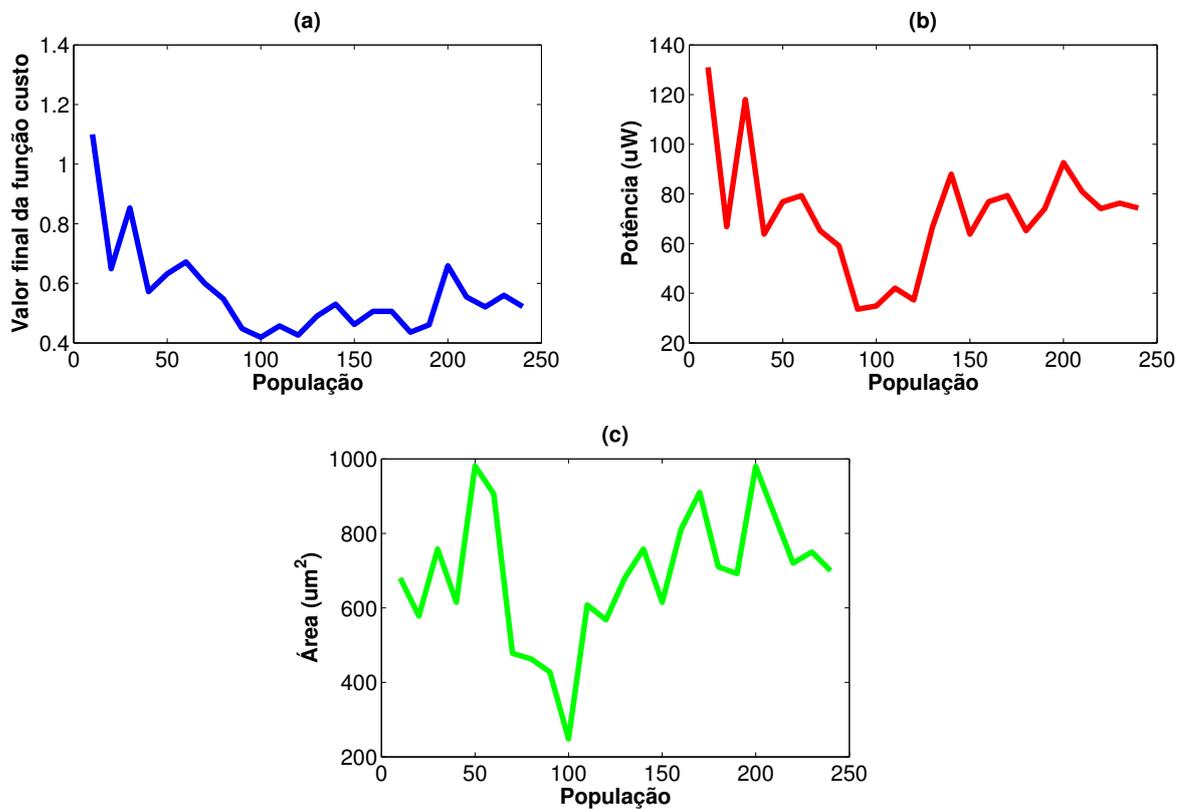
Com as configurações definidas para o início do dimensionamento do amplificador OTA Miller, o PSO precisa ser ajustado para encontrar um ponto de equilíbrio, ou seja, que encontre a melhor configuração entre o tamanho da população e o número de iterações, de modo que atinja um resultado esperado e não gaste um tempo computacional muito elevado. Para isso foram seguidos alguns passos.

Primeiramente, o número total de iterações recebeu um valor arbitrário de 100 e o tamanho da população foi variado na faixa de 10 a 240, com o passo de 10. A Figura 13 demonstra os resultados obtidos nas simulações, em relação a função custo (a), potência dissipada (b) e área de gate (c).

A Figura 13 mostra que, para a função custo e a área de gate o melhor resultado alcançado é quando o tamanho da população é de 100 partículas. Já para a potência dissipada o melhor resultado é obtido para uma população igual a 90. Portanto, foi definido que a melhor configuração para o tamanho da população é igual a 100, visando que o principal objetivo da ferramenta é minimizar ao máximo o valor final da função custo.

Fixando o tamanho da população em 100, posteriormente foi variado o valor do número total de iterações entre 10 e 240, com passo de 10. A Figura 14 apresenta os

Figura 13 – Resultados encontrados no projeto de um OTA Miller com as especificações dada pela Tabela 1 para uma população variando de 10 a 240 e o número total de iterações igual a 100. a) Função Custo; b) Potência Dissipada; c) Área de gate.



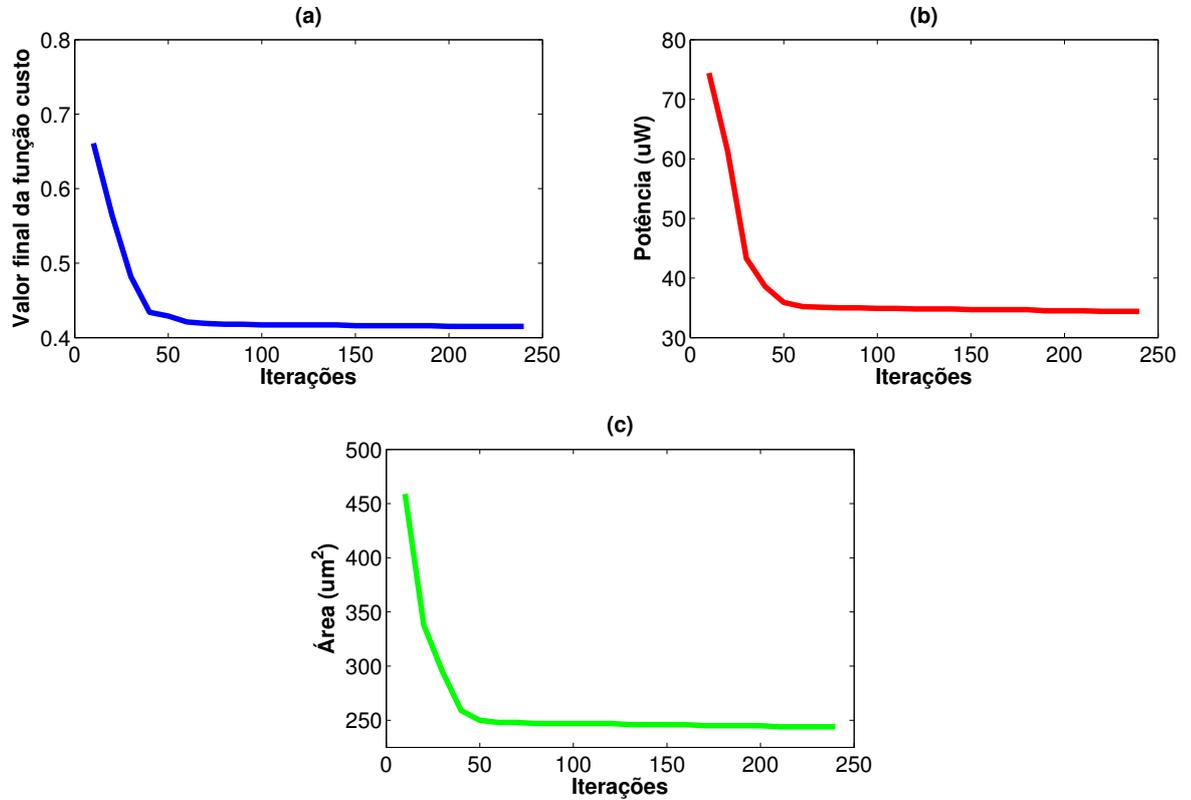
Fonte: Elaborado pelo autor.

resultados obtidos em relação à função custo (a), potência dissipada (b) e área de gate (c).

Os resultados da Figura 14 demonstram que os valores a serem otimizados decaem rapidamente em torno de 60 iterações. A execução de mais de 60 iterações exige um tempo computacional maior e não retorna uma diminuição da função custo tão significativa.

Portanto, para o dimensionamento do OTA Miller foi definido que a melhor configuração para o algoritmo PSO é atingido quando o tamanho da população alcança 100 e o número total de iterações é 60. Para estes casos de testes foi utilizado uma semente de randomização (*Seed*) fixa para a geração dos números aleatórios necessários para o cálculo das posições das partículas (Eq. 3.8), ou seja, os valores de inicializações do algoritmo foram sempre os mesmos para todos os testes. Com isso, garantiu-se que todos os resultados fossem comparados de maneira justa. A semente randômica ficou definida como o valor zero.

Figura 14 – Resultado encontrados no projeto de um OTA Miller com as especificações dada pela Tabela 1 para uma população com tamanho 100 e variando o número de iterações. a) Função Custo; b) Potência Dissipada; c) Área de gate.



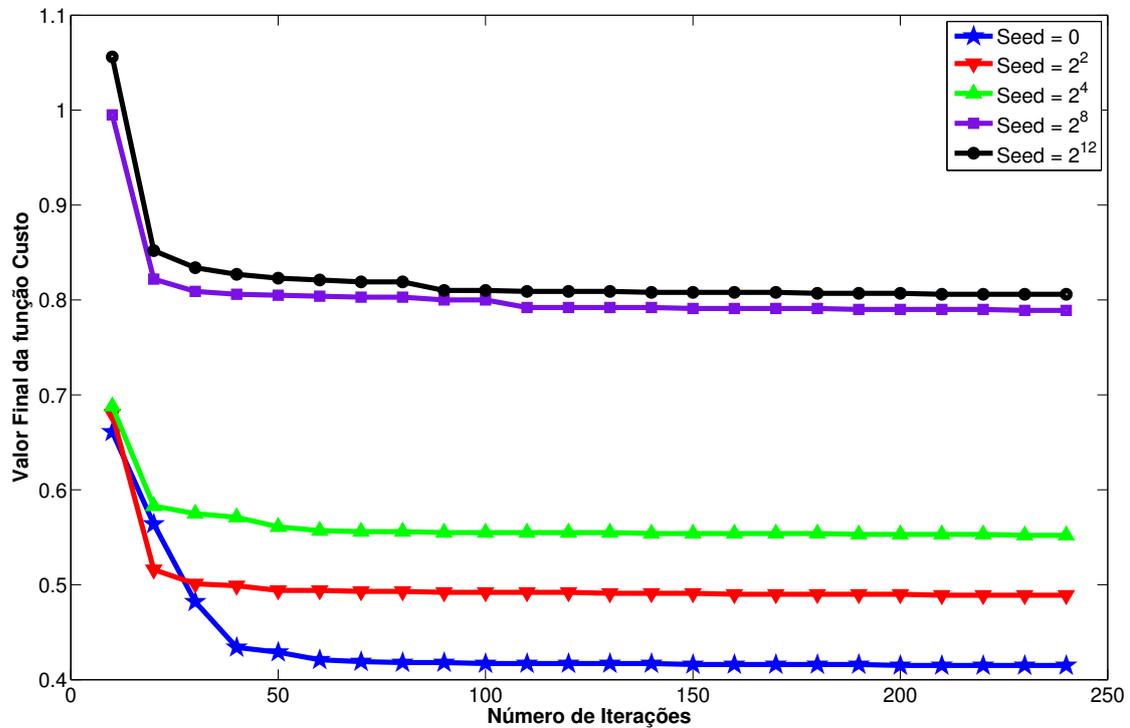
Fonte: Elaborado pelo autor.

Com o intuito de analisar se a semente randômica influencia no resultado final, foram executados cinco experimentos com sementes de randomização diferentes, com tamanho da população em 100 e número total de iterações variando de 10 a 240, com passo de 10. A Figura 15 apresenta os resultados alcançados. Foi analisado neste caso apenas a função custo, pois ela é o ponto mais importante a ser minimizado dentro da ferramenta UCAF.

A Figura 15 demonstra que a semente tem grande influência na busca por melhores resultados. Pode-se observar que, ao aumentar o valor da semente randômica, o resultado tende a convergir mais rapidamente, porém o resultado estabiliza em um patamar mais elevado. Portanto, ficou definido que se obtém o melhor resultado de otimização quando o valor da semente é definido como zero, que na Figura 15 é representado por *Seed* igual a zero.

Tendo encontrado a melhor configuração para o PSO, que os resultados otimizados

Figura 15 – Especificações dos valores finais em relação à função custo para uma população com tamanho de 100 e número total de iterações variando de 10 a 240, com cinco valores de semente diferentes.



Fonte: Elaborado pelo autor.

estão apresentados na Tabela 2. Pode-se comparar com os resultados encontrados para o mesmo problema utilizando o algoritmo Simulated Annealing, descrito em (SEVERO, 2012).

Com a análise dos resultados obtidos conclui-se que, quando a ferramenta é executada com a heurística PSO, apresenta uma redução de 63,27 % em potência dissipada e 41,65 % na área de gate quando comparados com os resultados obtidos através do SA. O SA utilizou de 11.280 iterações na sua execução, na busca do resultado ótimo, e para fins de comparação, o PSO foi configurado com o tamanho da população igual a 100 e o número de iterações igual a 60. A função custo final foi reduzida de 2,14 no SA, para 0,41 com o algoritmo PSO. Foi possível perceber também que ambos algoritmos atingiram todas as especificações exigidas.

Tabela 2 – Resultados das especificações para o dimensionamento do OTA Miller utilizando PSO e SA.

Especificações	PSO (este trabalho)	SA (SEVERO, 2012)
Av0 (dB)	79.94	81.59
GBW (MHz)	5.01	5.00
PM (°)	60.02	60.74
SR (V/ μ s)	8.38	5.33
ICMR+ (V)	0.87	0.76
ICMR- (V)	-0.71	-0.79
Pdiss (μ W)	34.98	95.13
Área de gate (μ m ²)	248	425
Função Custo	0.41	2.14

Fonte: Elaborado pelo autor.

4.2 TELESCOPIC

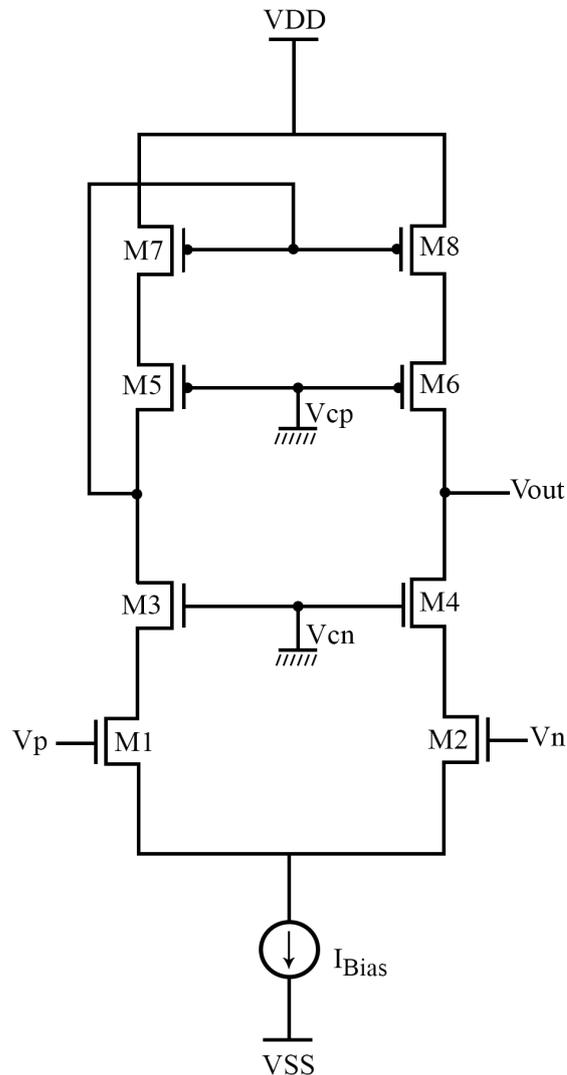
Para um segundo caso de teste foi utilizado o amplificador Telescopic. Seu esquemático é demonstrado na Figura 16. Suas correntes de entrada são espelhadas à de saída e sua arquitetura coloca os pares diferenciais de entrada e saída no mesmo ramo. As especificações elétricas do Telescopic são semelhantes às do OTA Miller.

Conforme a Figura 16, o amplificador Telescopic possui oito transistores que devem ser dimensionados, de forma a obter os valores dos parâmetros de largura (W) e comprimento (L). Algumas igualdades entre os transistores podem ser atribuídas para garantir o matching. Sendo assim, (M1 - M2), (M3 - M4), (M5 - M6) e (M7 - M8) devem possuir transistores iguais. Portanto, serão dimensionados quatro transistores, ao invés dos oito. O projeto apresenta 11 variáveis livres, sendo elas: W_{M1} , L_{M1} , W_{M3} , L_{M3} , W_{M5} , L_{M5} , W_{M7} , L_{M7} , além da fonte de corrente (I_B) e as referências de tensão V_p e V_n . Sua fonte de alimentação $V_{DD} = |V_{SS}|$, recebeu o valor de 0,9 V. A tecnologia de fabricação utilizada foi a XFAB 0,18 μ m. As faixas de variações das variáveis livres são: $L = 0,18 \mu$ m a 50 μ m, $W = 0,22 \mu$ m a 100 μ m, $I_B = 1 \mu$ A a 100 μ A e $V_p = |V_n| = 0,80$ a 0,90.

A função custo foi configurada com todos os fatores de ponderação de restrições iguais a 1 e os valores das especificações iguais aos valores mostrados na segunda coluna da Tabela 3. A área de gate e a potência dissipada são objetivos de projeto, e as demais especificações são restrições de projeto.

Com as configurações definidas para o início do dimensionamento do amplificador operacional Telescopic, foram realizados os mesmos casos de testes que no OTA Miller

Figura 16 – Esquemático do amplificador operacional Telescópico.



Fonte: Elaborado pelo autor.

para encontrar a melhor configuração possível do PSO.

O número de iterações recebeu o valor arbitrário de 100 e o tamanho da população foi variado na faixa de 10 a 240, com o passo de 10. A Figura 17 apresenta os resultados obtidos nas simulações, em relação a função custo (a), potência dissipada (b) e área de gate (c).

A Figura 17 demonstra que, para os três casos o melhor resultado encontrado é quando a população é 100 partículas.

O resultado referente ao tamanho da população se demonstrou igual ao primeiro teste, quando foi projetado o OTA Miller. O próximo passo é variar o valor do número total de iterações e analisar o comportamento da solução final. O valor do número total

Tabela 3 – Especificações para o dimensionamento do amplificador Telescopic

Especificações	Valor Requerido
Av0 (dB)	≥ 60.00
GBW (MHz)	≥ 2.00
PM ($^{\circ}$)	≥ 50.00
SR (V/ μ s)	≥ 5.00
ICMR+ (V)	≥ 0.40
ICMR- (V)	≤ -0.40
Pdiss (μ W)	Minimizar
Área de gate (μ m ²)	Minimizar
Função custo	Minimizar

Fonte: Elaborado pelo autor.

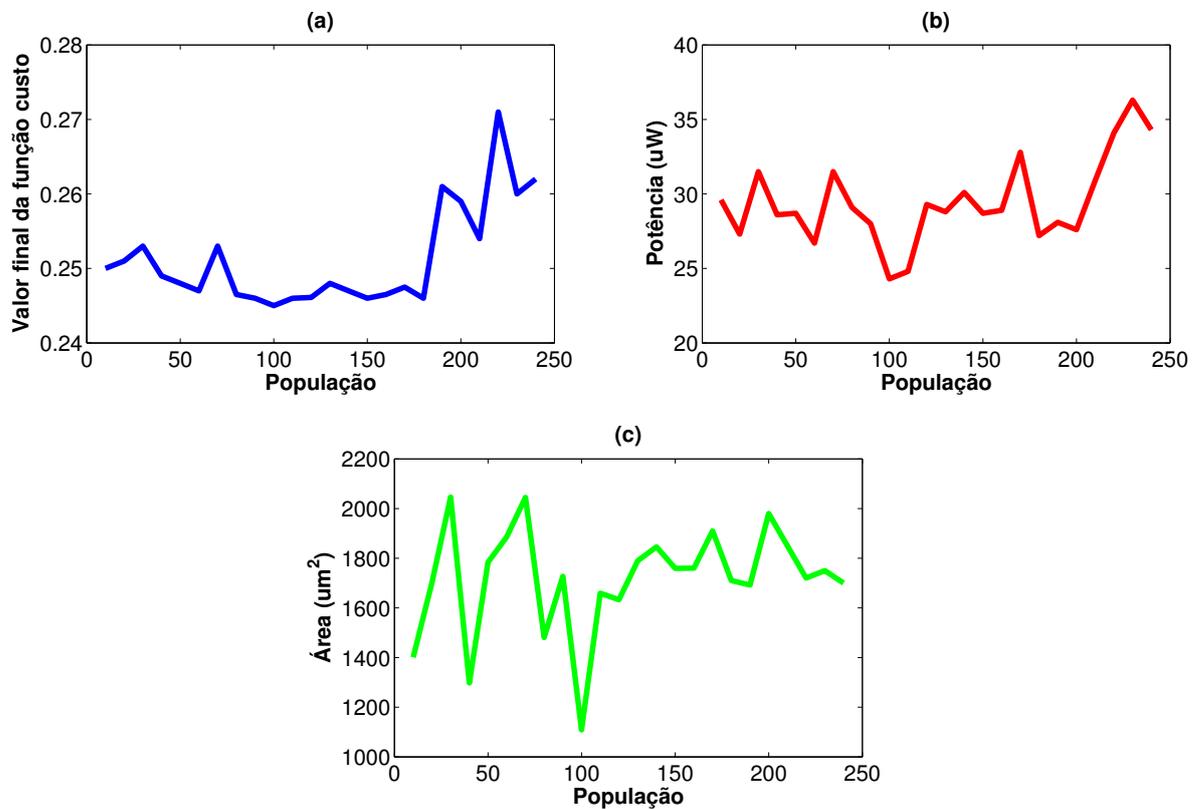
de iterações foi variado de 10 a 240, com passo de 10. A Figura 18 mostra os resultados obtidos em relação a função custo (a), potência dissipada (b) e área de gate (c).

Os resultados da Figura 18 demonstram que o algoritmo tem comportamento um pouco diferente do demonstrado para o amplificador Miller. O valor da função custo decai significativamente em torno de 30 iterações. Porém, ele mantém uma redução pequena até 110 iterações. Já a potência dissipada reduz rapidamente em 20 iterações e continua decaindo em torno de 50 iterações. Daí em diante ela estabiliza. A área, ao contrário das demais, teve um comportamento inverso. Ao aumentar o número total de iterações seu tamanho aumenta, estabilizando na casa de 130 iterações. Porém, observando o valor final da função custo nota-se que ao utilizar 60 iterações, mesmo utilizado no OTA Miller, o algoritmo alcança um resultado satisfatório.

Portanto, para o dimensionamento do amplificador Telescopic, mesmo tendo alguns resultados diferentes comparado ao OTA Miller, quando o algoritmo PSO é definido com o tamanho da população em 100 e número total de iterações igual a 60, ele alcança um resultado satisfatório. Para estes casos de testes foi utilizado uma semente de randomização fixa (*Seed*) para a geração dos números aleatórios necessários para o cálculo das posições das partículas (Eq. 3.8), ou seja, os valores de inicializações do algoritmo foi sempre a mesma para todos os tipos de caso. Com isso, garantiu-se que todos os resultados fossem comparados de maneira justa. A semente randômica ficou definida, da mesma forma que no OTA Miller, com o valor zero.

Com o intuito de analisar se a semente randômica influencia no resultado final, foram executados cinco experimentos com sementes de randomização diferentes, igualmente à do OTA Miller, com tamanho da população em 100 e valor do número total de iterações

Figura 17 – Resultados encontrados no projeto de um amplificador Telescopic com as especificações dado pela Tabela 3 para uma população variando de 10 a 240 e o número total de iterações igual a 100. a) Função Custo; b) Potência Dissipada; c) Área de gate.



Fonte: Elaborado pelo autor.

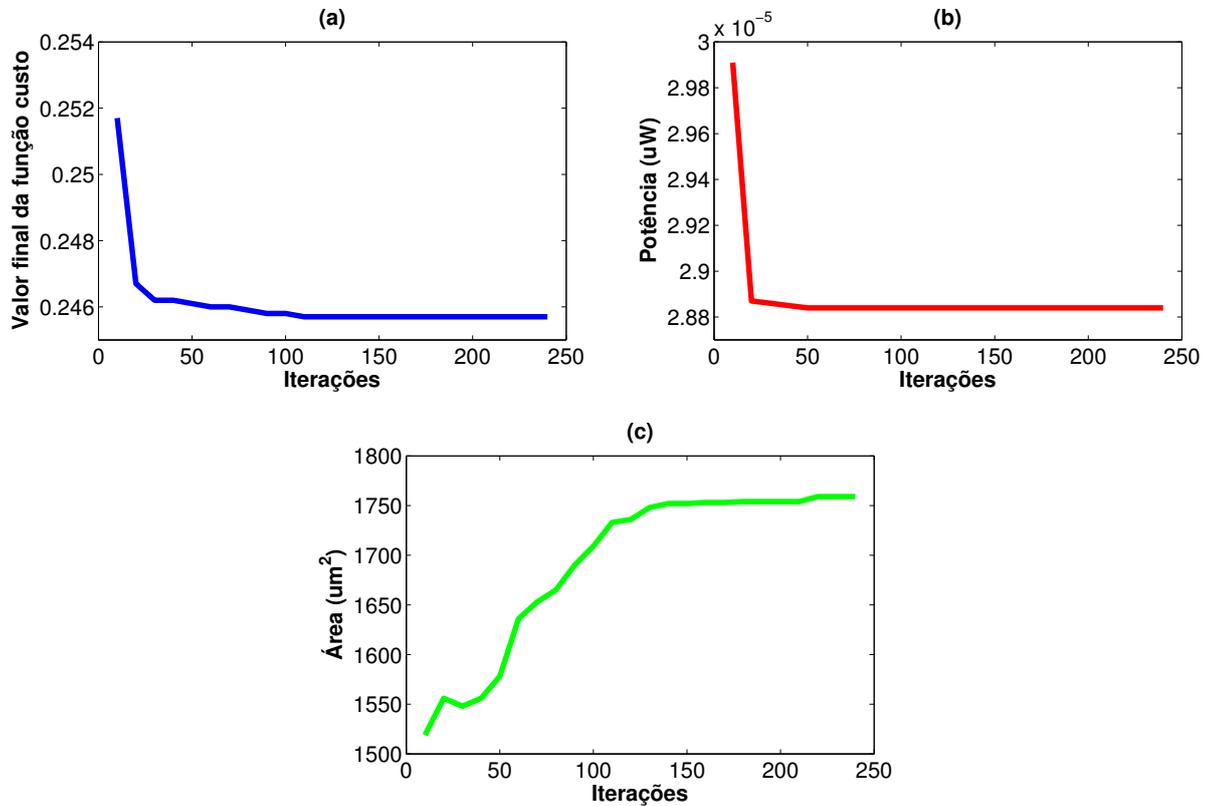
variando de 10 a 240, com passo de 10. A Figura 19 apresenta os resultados alcançados. Foi analisado neste caso apenas a função custo, pois ela é o ponto mais importante a ser minimizado dentro da ferramenta UCAF.

A Figura 19 demonstra que a semente no Telescopic tem o mesmo comportamento que no OTA Miller, porém com uma variação menor. O melhor resultado de otimização foi alcançado quando o valor da semente é definido como zero, representado na Figura 19 como *Seed* igual à zero.

Tendo encontrado a melhor configuração para o PSO, que os resultados otimizados estão apresentados na Tabela 4. Pode-se comparar com os resultados encontrados para o mesmo problema utilizando o algoritmo Simulated Annealing, descrito em (SEVERO, 2012).

Os resultados obtidos utilizando o PSO apresentaram uma redução de 41,93 % na

Figura 18 – Resultado encontrados no projeto de um amplificador Telescopic com as especificações dado pela Tabela 1 para uma população com tamanho 100 e variando o número total de iterações. a) Função Custo; b) Potência Dissipada; c) Área de gate.

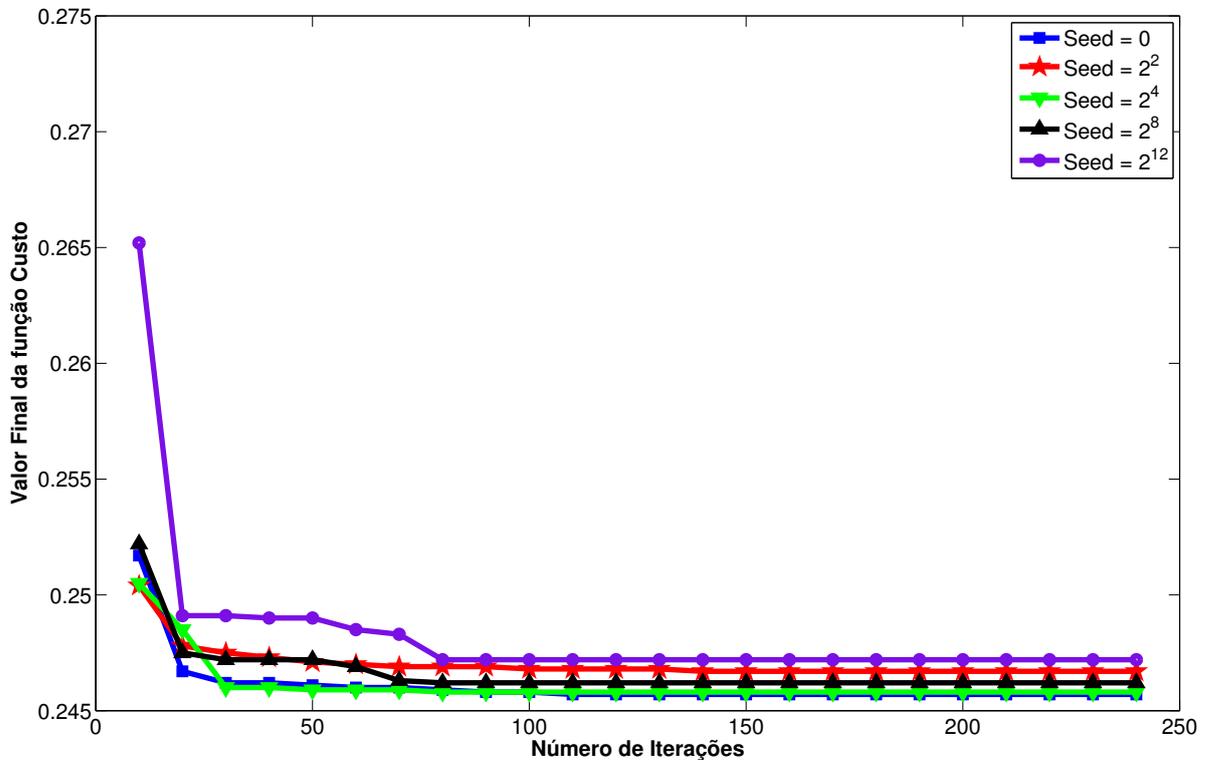


Fonte: Elaborado pelo autor.

potência dissipada e 16,23 % na área de gate, quando comparado com os resultados obtidos através do SA. O SA utilizou 8.920 iterações na sua execução, na busca do resultado ótimo. Para fins de comparação, o PSO foi configurado como tamanho da população igual a 100 e o número total de iterações igual a 60. A função custo final foi reduzida de 0,86 no SA para 0,23 com o algoritmo PSO. Em ambos os algoritmos todas as especificações foram atingidas.

Os resultados obtidos com o algoritmo Particle Swarm Optimization demonstraram um ganho superior em relação ao Simulated Annealing, implementado anteriormente na ferramenta. Também ficou definida a melhor configuração do PSO, que é quando o tamanho da população é igual a 100, o número total de iterações igual a 60 e semente randômica definida como zero.

Figura 19 – Especificações dos valores finais em relação a Função Custo para uma população com tamanho de 100 e o número total de iterações variando de 10 a 240, com cinco valores de semente diferentes.



Fonte: Elaborado pelo autor.

4.3 CONCLUSÃO

Neste capítulo foram abordados os dimensionamentos de dois amplificadores, do tipo OTA Miller e Telescopic, os quais foram explorados utilizando a heurística Particle Swarm Optimization.

Em ambos dimensionamentos a heurística Particle Swarm Optimization demonstrou comportamentos semelhantes, o qual foi verificado que os melhores resultados são encontrados quando o tamanho da população é igual a 100, o número total de iterações igual a 60 e a semente randômica definida como zero. A utilização do cálculo da inércia junto ao cálculo da velocidade da partícula permitiu encontrar um elevado número de soluções ótimas.

A heurística PSO insere melhorias ao processo de busca por um resultado otimizado na ferramenta UCAF, comparado ao SA. Obteve-se uma redução da função custo de 2,14 utilizando o SA, para 0,41 com o PSO no projeto do OTA Miller e de 0,86 utilizando o

Tabela 4 – Resultados das especificações para o dimensionamento do amplificador Telescopic, utilizando PSO e SA.

Especificações	PSO (este trabalho)	SA (SEVERO, 2012)
Av0 (dB)	63.90	62.88
GBW (MHz)	4.71	3.50
PM (°)	75.98	50.09
SR (V/ μ s)	5.01	5.76
ICMR+ (V)	0.58	0.64
ICMR- (V)	-0.71	-0.60
Pdiss (μ W)	28.84	49.66
Área de gate (μ m ²)	1640	1957
Função custo	0.23	0.86

Fonte: Elaborado pelo autor.

SA, para 0,23 com o PSO no projeto do amplificador Telescopic.

5 ANÁLISE DA TÉCNICA HÍBRIDA DE AMOSTRAGEM MONTE CARLO

A análise da produtividade no processo de automação do projeto analógico permite uma modelagem realística, contribuindo para a geração de circuitos factíveis no mundo real. Porém, ela apresenta um longo tempo de simulação no processo de otimização. Deste modo, busca-se uma estratégia que leve em consideração este problema de produtividade e diminua o impacto do custo computacional.

Na ferramenta UCAF descrita por (SEVERO, 2012) a produtividade do circuito é estimada apenas na iteração em que uma provável solução é encontrada, conforme pode-se ver no fluxograma da Figura 20. Para o cálculo da produtividade é utilizada a análise de Monte Carlo.

Para buscarmos uma estratégia que mais se adeque na diminuição do custo computacional levando-se em conta uma estimativa de produtividade do circuito, foi realizado um estudo sobre a aplicação de diferentes métodos de amostragem para a análise Monte Carlo. São comparadas duas técnicas, a amostragem aleatória simples e a amostragem por hipercubo latino.

5.1 MÉTODOS DE AMOSTRAGEM

Com o objetivo de analisar o desempenho dos métodos de amostragem aleatório e por hipercubo latino na estimativa de média e desvio padrão de algumas especificações do circuito foi utilizado um OTA Miller previamente dimensionado. A Tabela 5 mostra os valores das dimensões dos transistores, fonte de corrente e capacitância de acoplamento do amplificador.

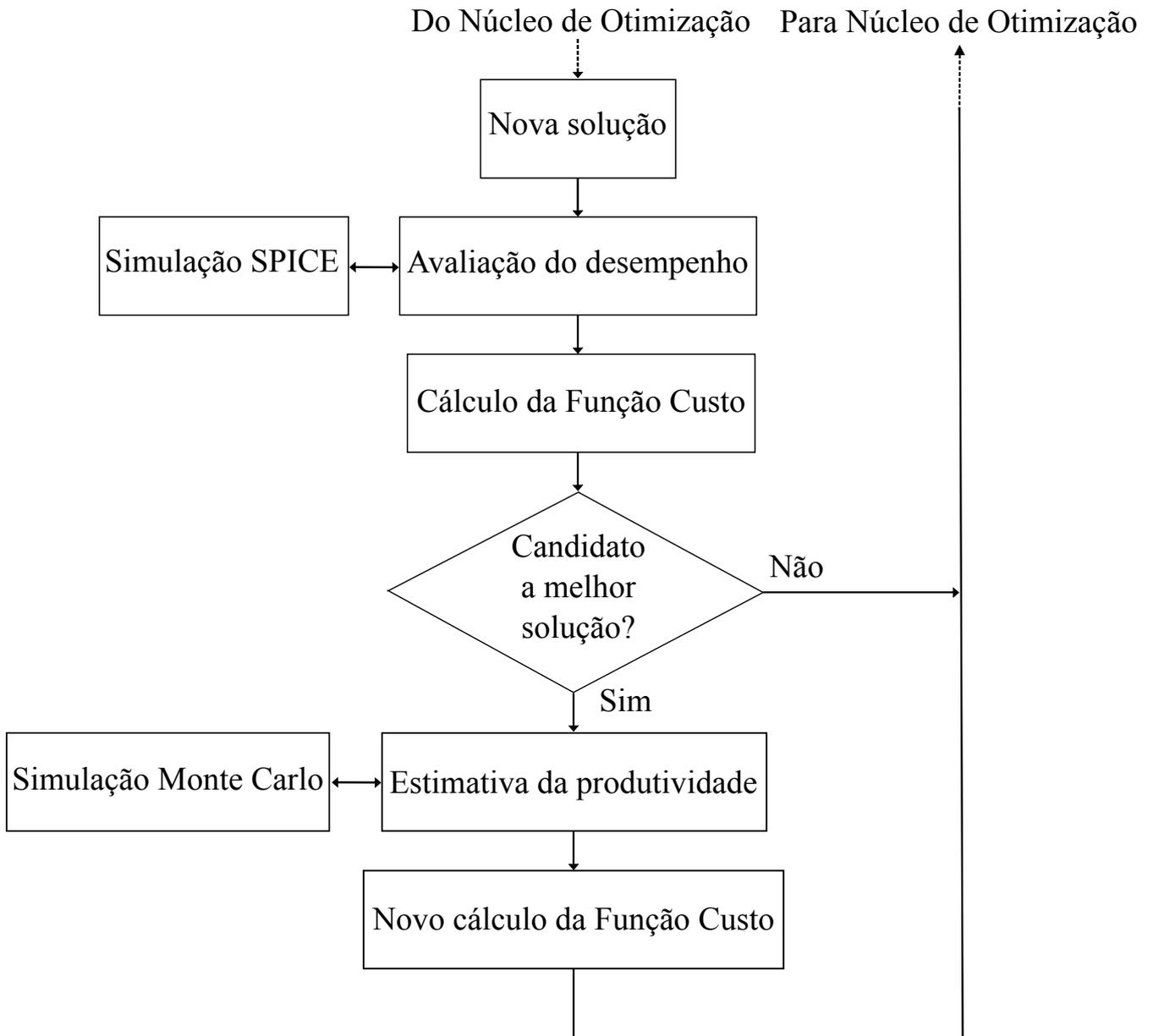
Tabela 5 – Valores das variáveis do OTA Miller utilizados para análise dos métodos de amostragem.

Variável	Valor
W_1/L_1	26,18 μm / 3,28 μm
W_3/L_3	98,78 μm / 5,00 μm
W_5/L_5	55,29 μm / 3,17 μm
W_6/L_6	90,14 μm / 0,94 μm
W_7/L_7	91,25 μm / 2,39 μm
I_{ref}	100,001 μA
C_f	5,99 pF

Fonte: Elaborado pelo autor.

Foram definidas algumas especificações para a avaliação do comportamento dos

Figura 20 – Fluxo de otimização incluindo estimativa de produtividade apenas para melhor solução encontrada durante o processo de otimização (SEVERO, 2012).



Fonte: Adaptado de Severo (2012).

métodos de amostragem. As especificações escolhidas foram Av0, GBW, PM, SR e potência dissipada.

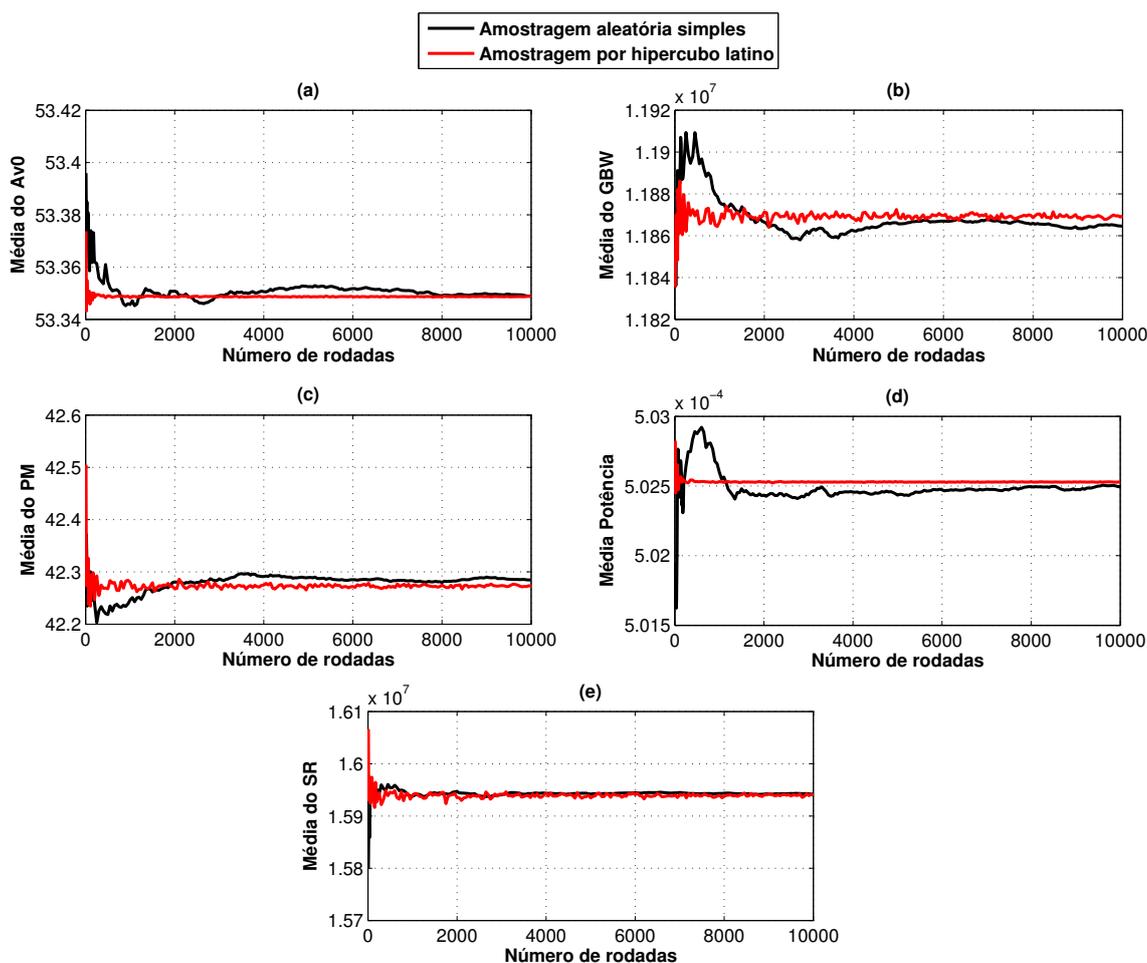
Foram executadas simulações para a estimativa de produtividade do circuito variando-se o número de rodadas de simulação Monte Carlo de 10 a 200, com um incremento de 10, e de 250 a 10.000 com incremento de 50.

A cada simulação Monte Carlo foi possível calcular os valores de média e desvio padrão para cada especificação. Com estes valores foi possível avaliar o comportamento

dos métodos de amostragem para a estimativa da produtividade do circuito.

As Figuras 21 e 22 apresentam os valores de média e desvio padrão, respectivamente, em relação ao número de rodadas para ambos os métodos de amostragem, nos quais temos os valores de Av0, GBW, PM, Potência Dissipada e SR.

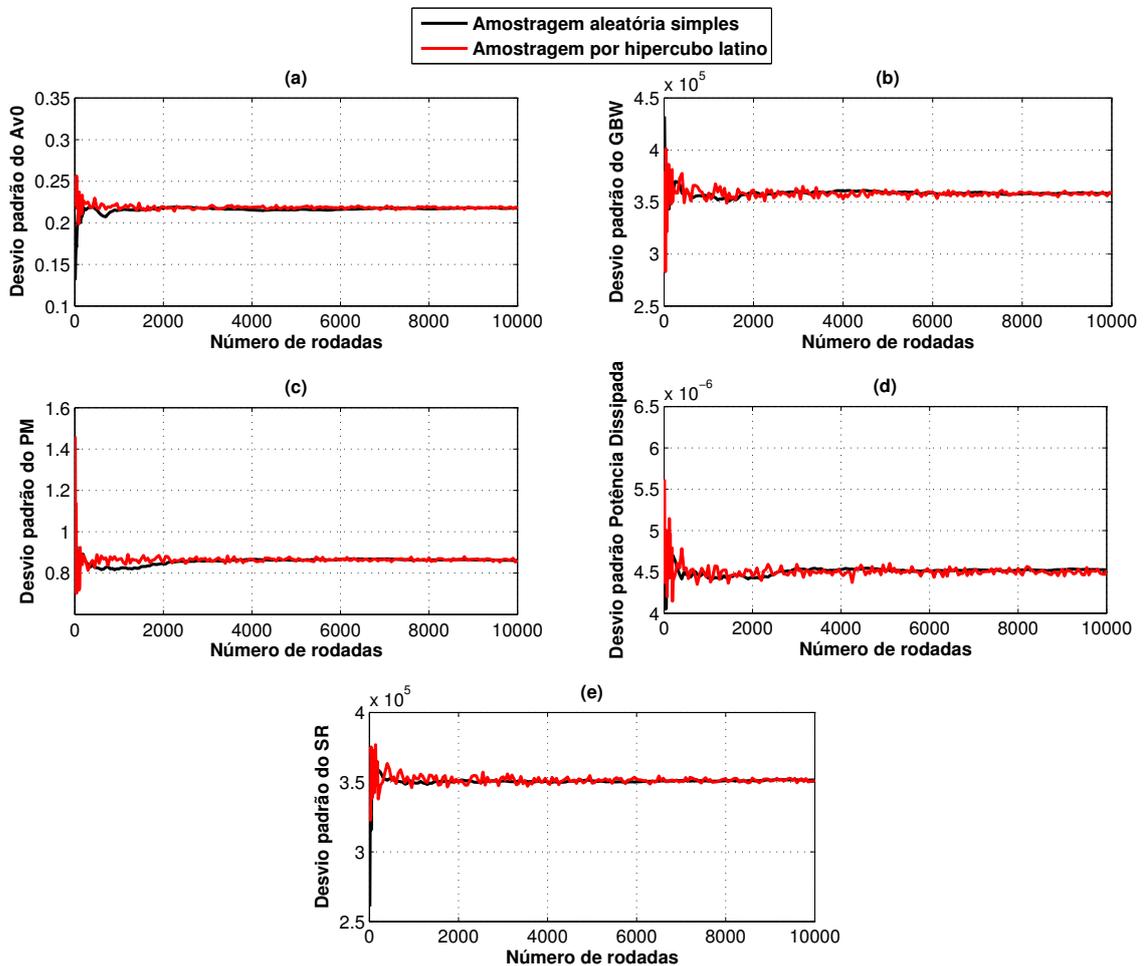
Figura 21 – Gráficos da média pelo número de rodadas de simulação Monte Carlo para as especificações do OTA Miller da Tabela 5. a) Av0; b) GBW; c) PM; d) Potência; e) SR.



Fonte: Elaborado pelo autor.

É possível verificar que os valores das especificações na amostragem por hipercubo latino convergem mais rapidamente para valores estáveis do que na amostragem aleatória simples. Ou seja, são necessárias menos rodadas de simulação no hipercubo latino para obter-se estimativas de média estáveis. Já para o desvio padrão este efeito não é tão proeminente, apesar de ambos os métodos convergirem de forma semelhante. Menos rodadas de simulação significa mais velocidade de processamento para estimativa de produtividade.

Figura 22 – Gráficos do desvio padrão pelo número de rodadas de simulação Monte Carlo para as especificações do OTA Miller da Tabela 5. a) Av0; b) GBW; c) PM; d) Potência; e) SR.



Fonte: Elaborado pelo autor.

Estimando os valores de média e desvio padrão com 10.000 rodadas e amostragem aleatória, consegue-se alcançar um valor que garante uma alta confiabilidade na estimativa. Portanto, este resultado foi considerado como valor de referência. Tendo isto como base, a Tabela 6 apresenta a diferença da estimativa da média e desvio padrão do método hipercubo latino com 10.000 rodadas de simulação e o valor de referência.

Os resultados obtidos na Tabela 6 demonstram que os valores finais estimados em ambos os métodos de amostragem estão bem próximos, ou seja, ambos os métodos têm um comportamento parecido quando o número muito grande de rodadas estipulado é alcançado (10.000 rodadas).

De modo a medir a velocidade de convergência dos métodos de amostragem, realizou-se testes com o intuito de verificar quantas rodadas de simulação eram necessárias

Tabela 6 – Tabela de diferença percentual dos valores estimados para a média e desvio padrão pelo método hipercubo latino com 10.000 rodadas e os valores de referência.

Especificação	Diferença de média (%)	Diferença de desvio padrão (%)
Av0	0,0004	0,1358
GBW	0,0409	0,0125
Margem de Fase	0,0290	0,1949
SR	0,0126	0,1979
P_{Diss}	0,0072	0,1232

Fonte: Elaborado pelo autor.

para se estimar a média e o desvio padrão com um determinado erro ξ em relação ao valor de referência. O erro ξ para média(μ) e desvio padrão(σ) são representados pelas Equações 5.1 e 5.2 respectivamente.

$$\xi\mu_E^n = \hat{\mu}_E^n - \hat{\mu}_E^* \quad (5.1)$$

$$\xi\sigma_E^n = \hat{\sigma}_E^n - \hat{\sigma}_E^* \quad (5.2)$$

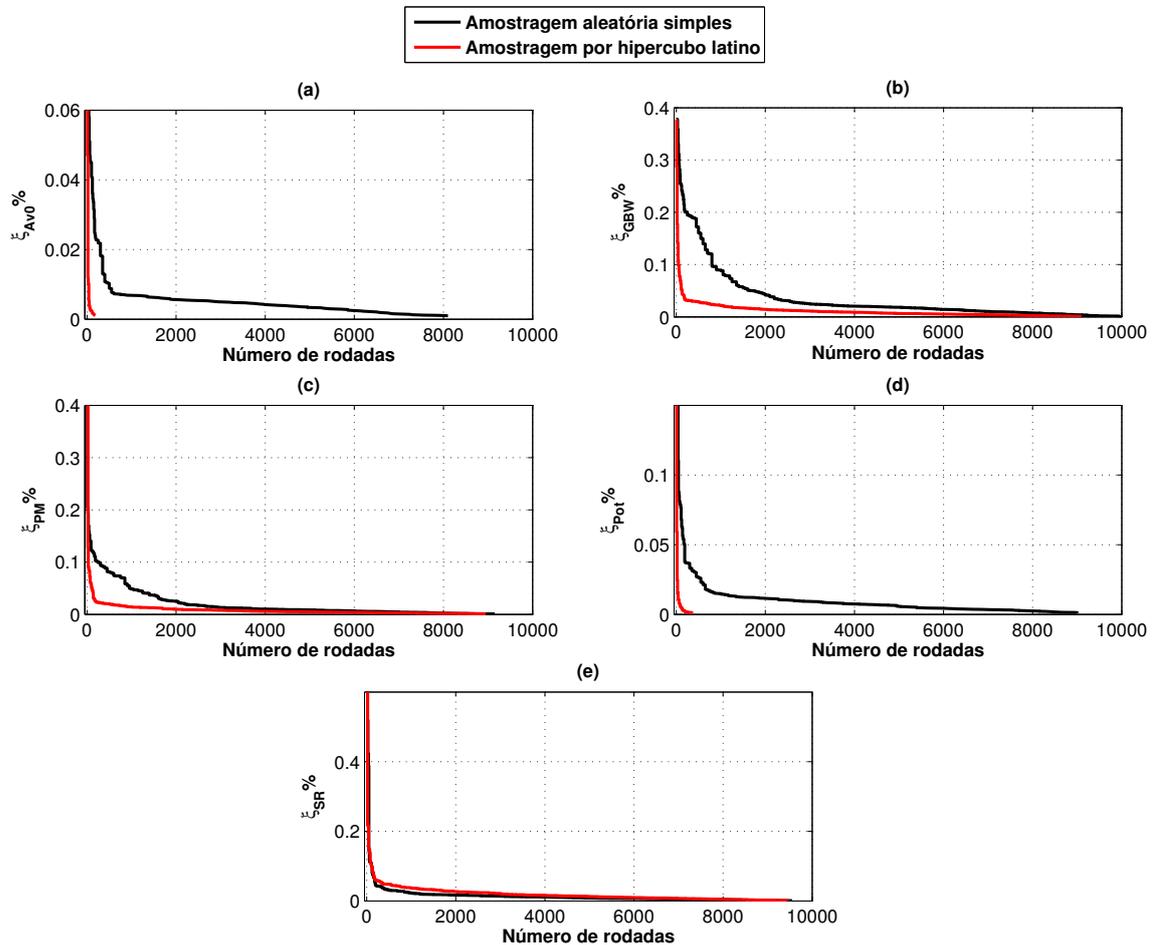
Onde:

- n valor de rodadas Monte Carlo.
- $\hat{\mu}$ valor de referência para média.
- $\hat{\mu}^*$ média estimada com 10.000 rodadas.
- $\hat{\sigma}$ valor de referência para desvio padrão.
- $\hat{\sigma}^*$ desvio padrão estimado com 10.000 rodadas.
- E especificação a ser analisada.

Para a execução destes testes foi considerado um valor de ξ igual a 0,001%, que foi sendo incrementado no passo de 0,001 até alcançar os resultados esperados. Com esta análise obteve-se gráficos que demonstram o valor do erro ξ em relação ao número de rodadas para os valores de média e desvio padrão, para ambos os métodos de amostragem, mostrados na Figura 23 .

Analisando a Figura 23, é possível notar que o método de amostragem por hipercubo latino estima os valores com um número de rodadas menor do que o método de amostragem aleatória, de modo que os valores obtidos mesmo com um número baixo de rodadas estão próximos aos valores de referência.

Figura 23 – Gráficos do erro ξ_μ em relação ao número de rodadas de simulação Monte Carlo para os valores de média das especificações. a) Av0; b) GBW; c) PM; d) Potência; e) SR.



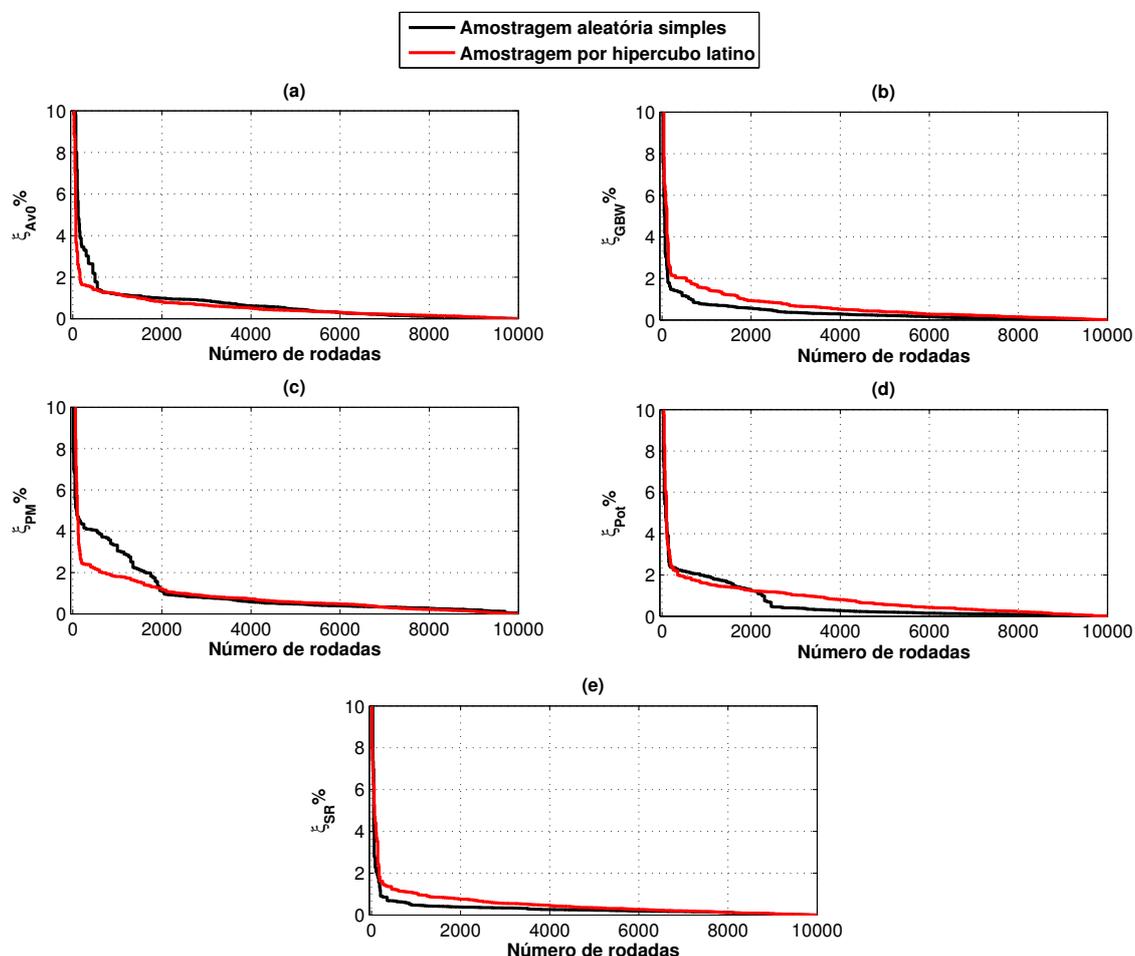
Fonte: Elaborado pelo autor.

A Figura 24 apresenta os gráficos do erro ξ em relação ao número de rodadas de simulação Monte Carlo, para ambos os métodos de amostragem, para os valores de desvio padrão das especificações de Av0, GBW, PM, Potência e SR.

Na Figura 24 é possível analisar, que a estimativa de valores de desvio padrão das especificações de Av0, GBW, PM, Potência e SR, possuem um desempenho inferior quando executados com o método de amostragem por hipercubo latino. Sendo assim, este método necessita um número maior de rodadas para obter valores próximos aos valores de referência.

Com a análise do desempenho dos métodos de amostragem, o método de amostragem por hipercubo latino demonstrou-se mais eficiente para a estimativa da média em relação ao método aleatório.

Figura 24 – Gráficos do erro ξ_σ em relação ao número de rodadas de simulação Monte Carlo para os valores de desvio padrão das especificações. a) Av0; b) GBW; c) PM; d) Potência; e) SR.



Fonte: Elaborado pelo autor.

5.2 MÉTODO HÍBRIDO DE AMOSTRAGEM

A proposta desta seção é desenvolver um método híbrido de amostragem que utilize tanto a amostragem aleatória simples como o hipercubo latino, com o intuito da redução do tempo computacional sem perder a precisão na estimativa da produtividade.

Para a realização deste estudo foram definidos alguns aspectos. A ferramenta UCAF foi configurada para executar o algoritmo PSO, em que ficou definido o tamanho da população igual a 100 e número total de iterações igual a 60, e sua semente para a geração de números aleatórios definida como zero. O amplificador utilizado foi o OTA Miller, cujo esquemático é demonstrado na Figura 12. O projeto utilizou a tecnologia IBM 0,13 μ m.

O projeto do OTA Miller segue a mesma lógica dos teste realizados até então, os

quais seus oito transistores são dimensionados, de forma a obter os valores dos parâmetros de largura (W) e comprimento (L). Devido as igualdades já citadas anteriormente, são dimensionados 5 transistores ao invés dos 8. O projeto apresenta então 12 variáveis livres.

A fonte de alimentação $V_{DD} = |V_{SS}|$, recebeu o valor de 0,6 V. Os valores mínimos de W e L foram definidos como $L = 0,50 \mu\text{m}$ e $W = 10,00 \mu\text{m}$. Com os valores mínimos de cada variável foram definidas as faixas de variações das variáveis livres que são: $L = 0,50 \mu\text{m}$ a $5 \mu\text{m}$, $W = 10,00 \mu\text{m}$ a $100 \mu\text{m}$, $I_B = 10 \mu\text{A}$ a $300 \mu\text{A}$ e $C_L = 0,80 \text{ pF}$ a $6,00 \text{ pF}$.

A função custo foi configurada com todos os fatores de ponderação de restrições iguais a 1 e os valores requeridos para as especificações iguais aos valores mostrados na segunda coluna da Tabela 7. Como mostra a Tabela 7 a área de gate, a potência dissipada, como também a produtividade foram considerados como objetivos de projeto, e as demais especificações são restrições de projeto.

Tabela 7 – Especificações projeto automático do OTA Miller.

Especificação	Valor requerido
A_{v0} (dB)	$\geq 50,00$
GBW (MHz)	$\geq 2,00$
Margem de fase ($^\circ$)	$\geq 60,00$
SR (V/ μs)	$\geq 5,00$
ICMR+ (V)	$\geq 0,40$
ICMR- (V)	$\leq -0,40$
OS+ (V)	$\geq 0,40$
OS- (V)	$\leq -0,40$
PSRR+ (dB)	$\geq 40,00$
PSRR- (dB)	$\leq -40,00$
P_{Diss} (μW)	minimizar
Área de gate (μm^2)	minimizar
Yield (%)	maximizar

Fonte: Elaborado pelo autor.

Ambas as técnicas de amostragem possuem características diferentes. A por hipercubo latino demonstrou convergir para um resultado esperado com um menor número de amostras, e a amostragem aleatória permite refinar o resultado com maior precisão.

Para isso, é necessário a definição de alguns pontos, que são:

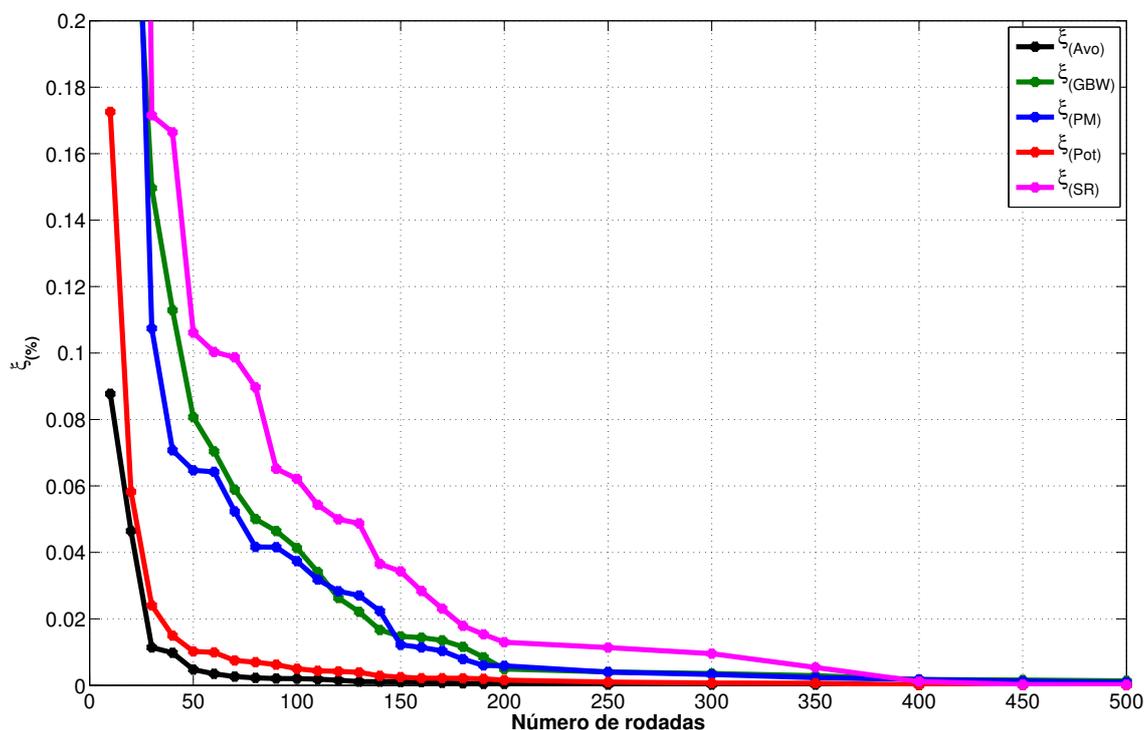
- Número de rodadas usando o método de amostragem por hipercubo latino (LHS);

- Número de rodadas usando o método de amostragem aleatória (RS);
- Para uma técnica híbrida, o ponto ideal de troca do método, e qual técnica seria utilizada inicialmente.

Foi realizado um estudo de caso para definir o número ideal de rodadas para ambas os métodos de amostragem. Neste trabalho ficou definido que a média tem um peso maior em relação ao desvio padrão para a estimativa da produtividade final. Com estas informações, ficou definido que um valor de ξ de 0,1% referente à média seria um valor aceitável. Assim, pode-se definir o número de rodadas para cada método de amostragem.

A Figura 25 apresenta as curvas mais detalhadas do erro ξ pelo número de rodadas de simulação Monte Carlo com o método de amostragem por hipercubo latino, referentes às médias das especificações.

Figura 25 – Curvas de erro ξ_μ em relação ao número de rodadas de simulação Monte Carlo com método de amostragem por hipercubo latino, referentes às médias das especificações.

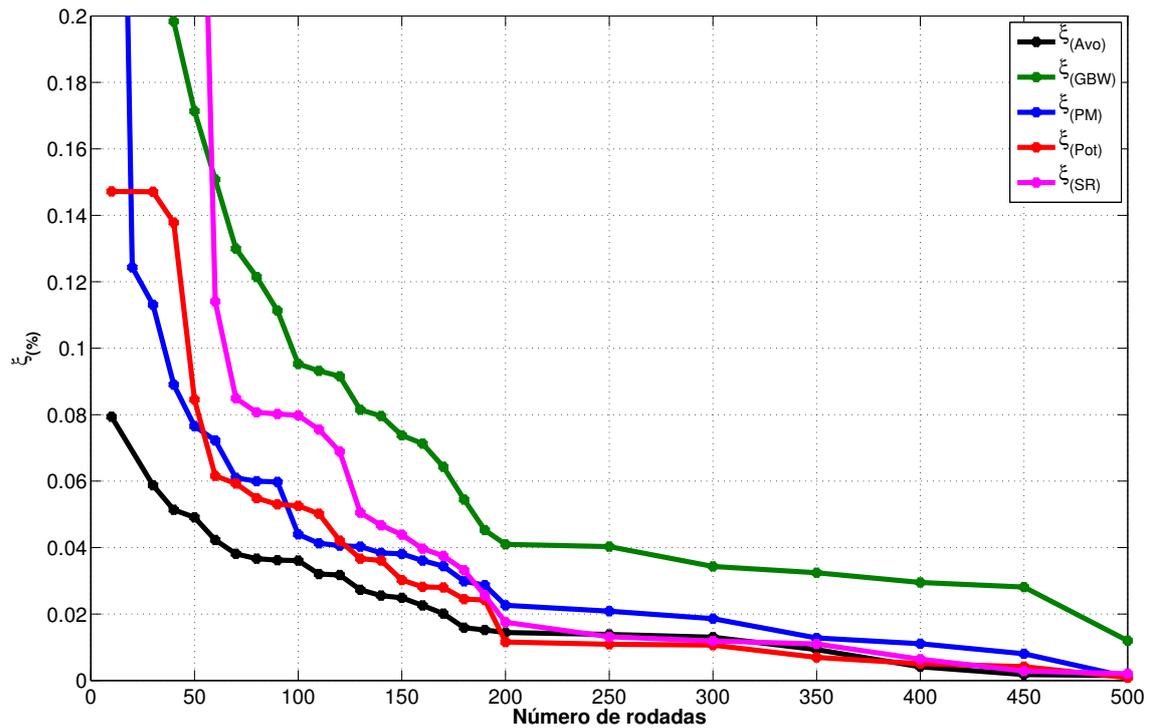


Fonte: Elaborado pelo autor.

Observa-se que, com 50 rodadas de simulação, todas as especificações atingem um erro máximo ξ de 0,1% em relação ao valor de referência.

A Figura 26 apresenta as curvas mais detalhadas do erro ξ pelo número de rodadas de simulação Monte Carlo com o método de amostragem aleatório, referentes às médias das especificações.

Figura 26 – Curvas de Erro ξ_μ em relação ao número de rodadas de simulação Monte Carlo com método de amostragem aleatório, referentes às médias das especificações.



Fonte: Elaborado pelo autor.

Observa-se que o método de amostragem aleatório tem um comportamento um pouco diferente, pois necessita de 100 rodadas de simulação, para que todas as especificações tenham um erro máximo ξ de 0,1% em relação ao valor de referência.

O método de amostragem por hipercubo latino consegue gerar uma estimativa de valores de média com um número menor de rodadas quando comparado com o método de amostragem aleatória para todas as especificações - quando se considera $\xi < 0,1\%$. Em alguns casos, como na estimativa do valor médio de GBW, o número de rodadas de simulação da amostragem por hipercubo latino chega a ser 40 vezes menor.

Porém, para os valores de desvio padrão, o método de amostragem por hipercubo latino possui um desempenho inferior. É preciso um número maior de rodadas de simulação para se obter valores mais próximos aos valores de referência.

Esta análise indica que o uso de amostragem por hipercubo latino é viável de

ser utilizada para o cálculo da produtividade de um circuito analógico com um número reduzido de rodadas Monte Carlo (em torno de 50 a 100), como uma alternativa para estimativa rápida e com erros pequenos para as especificações de desempenho do circuito. A perda de precisão do resultado é compensada pela grande velocidade do procedimento de cálculo. Isto torna a amostragem por hipercubo latino especialmente aplicável em um processo de otimização com número grande de rodadas, principalmente no início do processo, quando a exploração do espaço de projeto ainda não é suficientemente refinada e valores precisos de produtividade ainda não são necessários.

Com o número de rodadas previamente fixados, para ambos os métodos de amostragem, foi definido que o método de amostragem por hipercubo latino seria utilizado primeiramente com um número menor de rodadas, conforme já visto, com o intuito de ganhar velocidade na hora da convergência e processamento. A partir de certo ponto troca-se o método de amostragem para o aleatório, que segue até o final da execução da ferramenta, buscando um resultado mais refinado.

Inicialmente considerou-se um valor para a troca em 1000 execuções, que foi aumentando no passo de 500, até o limite de 5000, visando que todo o fluxo da ferramenta foi fixada em 6000 execuções. Além deste teste em relação ao melhor ponto de troca de amostragem, executou-se também, para fins de comparação, um projeto que utiliza apenas o método de amostragem por hipercubo latino com número de rodadas igual a 50, e um projeto somente com o método aleatório e número de rodadas igual a 100.

É importante ressaltar que os valores de 50 e 100 rodadas para os métodos de amostragem por hipercubo latino e aleatório, respectivamente, foram utilizados para estimar um valor de produtividade para otimizar o tempo no fluxo de projeto. No melhor resultado de cada fluxo de projeto foi executada uma nova análise Monte Carlo, que foi configurada com o método de amostragem aleatório simples e número de rodadas igual a 10.000, mesmo valor adotado como referência anteriormente, com o intuito de determinar a produtividade de forma mais precisa de cada projeto.

A Tabela 8 apresenta as principais características em relação ao fluxo de projeto, com os valores de tempo de execução, função custo final, produtividade final, potência dissipada e área do gate, em relação aos testes previamente estabelecidos.

Todos os resultados foram executados em uma máquina com processador Intel® Core™i7-3770 com 8GB de memória RAM de 1333 MHz, placa de vídeo GeForce GTX 550 Ti, 1TB de HD e utilizando o sistema Ubuntu, versão 14.04 LTS.

Conforme os resultados da Tabela 8, o projeto com amostragem por hipercubo latino (LHS) foi o mais eficiente em relação ao custo computacional, porém a sua produtividade e função custo final obtiveram o pior resultado. O projeto com amostragem aleatória simples (RS) foi o mais custoso computacionalmente e não obteve o melhor resultado entre

Tabela 8 – Comparação entre os projetos executados com os diferentes métodos de amostragem Monte Carlo.

Amostragem	Tempo (h)	F. Custo	Prod. (%)	P_{Diss} (μ W)	Área (μ m ²)
LHS	4,47	2,98	78,26	549,40	259,5
RS	8,09	2,09	87,35	519,60	311,9
LHS/RS-1000	8,00	2,19	81,58	547,50	328,4
LHS/RS-1500	7,80	2,12	83,58	552,30	269,8
LHS/RS-2000	7,42	2,34	84,14	534,90	228,5
LHS/RS-2500	7,13	2,31	84,21	509,90	260,9
LHS/RS-3000	6,88	1,77	90,90	569,90	302,2
LHS/RS-3500	6,73	2,14	86,99	549,40	298,3
LHS/RS-4000	6,76	2,21	86,30	553,40	291,6
LHS/RS-4500	6,65	2,35	84,01	532,60	302,7
LHS/RS-5000	5,99	2,50	83,20	542,50	296,8

Fonte: Elaborado pelo autor.

os demais testes. O projeto realizado com a técnica de amostragem híbrida, com troca em 3000 execuções (LHS/RS-3000), foi o que obteve os melhores resultados, que foi em torno de 1 hora e 12 minutos mais rápido que o projeto executado com a amostragem aleatória simples, e ainda obteve se uma produtividade final de 90,90%.

A Tabela 9 mostra os resultados obtidos para o projeto realizado com a técnica de amostragem híbrida, com troca em 3000 execuções (LHS/RS-3000), o qual nota-se que todos os valores das especificações requeridas foram alcançadas. O valores dos objetivos de otimização foram encontrados como 56,59 μ W para potência dissipada, 302,2 μ m² para área de gate e 90,90% para produtividade (estimado com 10.000 rodadas de Monte Carlo).

Os valores para o dimensionados das variáveis $W_1, L_1, W_3, L_3, W_5, L_5, W_6, L_6, W_7, L_7, I_{ref}$ e C_f são mostrados na Tabela 10.

A Figura 27 mostra a evolução da função custo do dimensionamento do OTA Miller, verificando-se que a heurística efetuou uma otimização no problema, uma vez que a melhor solução da população inicia em um valor alto e é reduzido ao longo das execuções.

Outro ponto importante ressaltar é que os resultados demonstraram que existe um ponto ótimo para a troca dos tipos de amostragem, neste caso em torno da execução de número 3000.

Para comprovar que existe um momento ótimo de troca, foram executadas simulações mais refinadas em torno de 3000 execuções. Com base nos dados da Tabela 8, o

Tabela 9 – Resultados obtidos com o projeto automático do OTA Miller com a técnica de amostragem híbrida e troca em 3000 execuções (LHS/RS-3000).

Especificação	Valor requerido	Projeto UCAF
Av0 (dB)	$\geq 50,00$	51,19
GBW (MHz)	$\geq 2,00$	16,59
Margem de fase ($^{\circ}$)	$\geq 60,00$	62,91
SR (V/ μ s)	$\geq 5,00$	19,33
ICMR+ (V)	$\geq 0,40$	0,43
ICMR- (V)	$\leq -0,40$	-0,58
OS+ (V)	$\geq 0,40$	0,54
OS- (V)	$\leq -0,40$	0,59
PSRR+ (dB)	$\geq 40,00$	61,52
PSRR- (dB)	$\leq -40,00$	61,71
P_{Diss} (μ W)	minimizar	56,59
Área de gate (μ m ²)	minimizar	302,2
Produtividade (%)	maximizar	90,90

Fonte: Elaborado pelo autor.

Tabela 10 – Valores das variáveis do OTA Miller dimensionadas a partir do projeto considerando amostragem híbrida e troca em 3000 execuções (LHS/RS-3000).

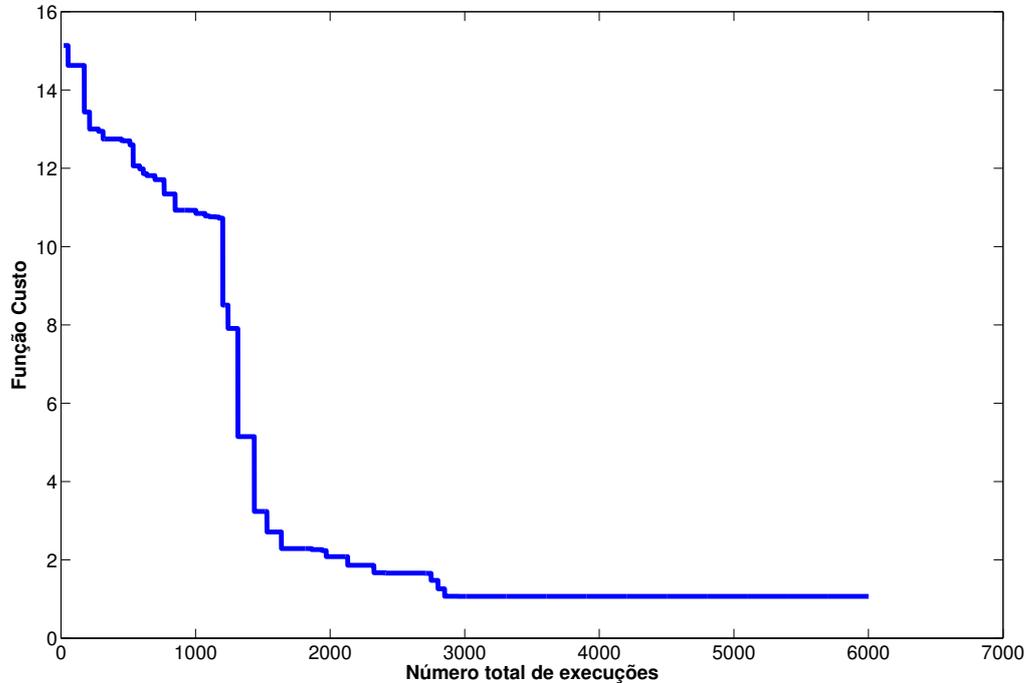
Variável	Valor
W_1/L_1	10,00 μ m / 2,01 μ m
W_3/L_3	16,57 μ m / 0,50 μ m
W_5/L_5	22,15 μ m / 7,38 μ m
W_6/L_6	94,30 μ m / 0,51 μ m
W_7/L_7	77,26 μ m / 1,98 μ m
I_{ref}	101,02 μ A
C_f	4,37p F

Fonte: Elaborado pelo autor.

momento de troca foi variado de 2500 execuções até 3500, com passo de 10. As Figuras 28 e 29, demonstram os resultados em relação à produtividade final (estimada com 10.000 rodadas) e função custo final, respectivamente.

É possível verificar que existe um ponto ótimo no processo de otimização no qual o método de amostragem deve ser alterado do hipercubo latino para o método aleatório.

Figura 27 – Evolução função custo para o melhor indivíduo da população durante o processo de otimização com a técnica de amostragem híbrida e troca em 3000 execuções (LHS/RS-3000).



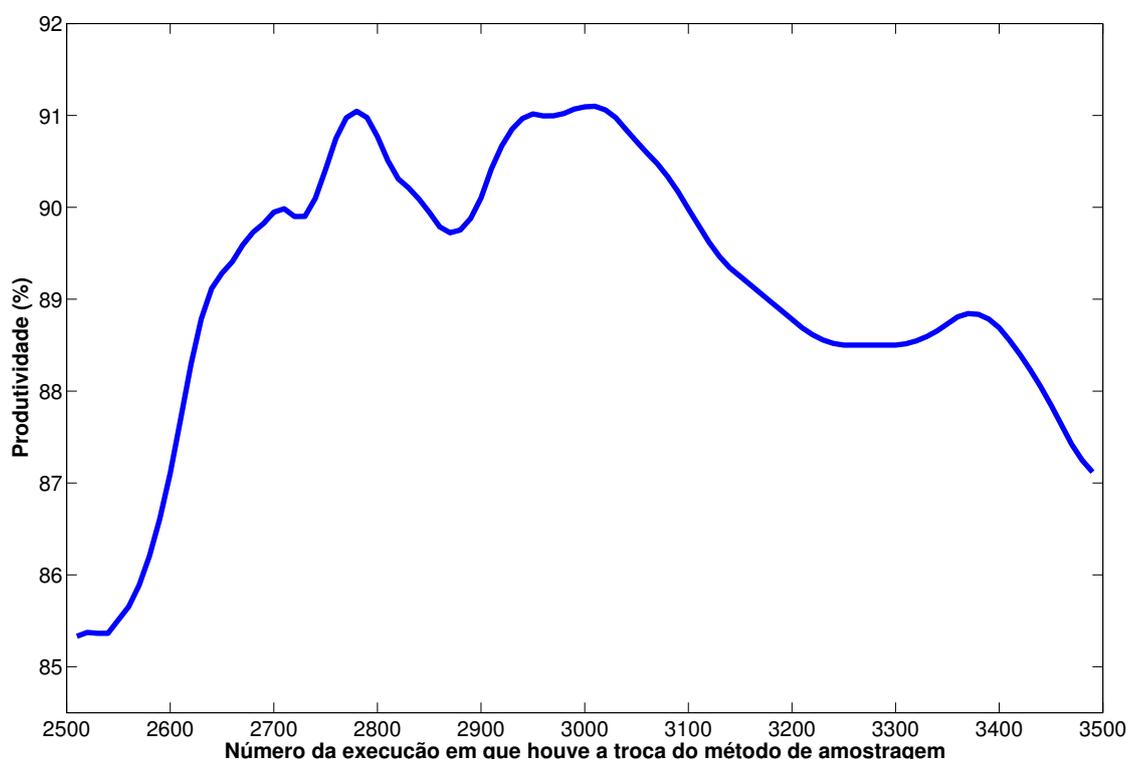
Fonte: Elaborado pelo autor.

Este ponto de troca ocorre entre 2700 e 2800 execuções. Observamos na Figura 28 que a produtividade inicia na faixa de 85%, tem uma ligeira suba até em torno de 91%, tem umas pequenas alterações e depois um declive nas demais execuções. Na Figura 29 a curva da função custo, que assemelha-se a uma parábola, tem o mesmo comportamento, porém inverso, visto que a função custo é dependente do valor da produtividade - quanto maior a produtividade, menor o valor da função custo. Novamente, o ponto ideal de troca ocorre entre 2700 e 2800 execuções. Portanto, conclui-se que o momento em que o método de amostragem é alterado influencia no resultado final do projeto. O momento ideal ficou definido em 2780 execuções, conforme analisado nas Figuras 28 e 29.

Para inserir a pesquisa deste trabalho na ferramenta UCAF foram executados três projetos do OTA Miller. O primeiro projeto foi baseado no trabalho de Severo (2012), o qual foi executado um fluxo de projeto utilizando a versão desenvolvida pelo mesmo. O segundo e o terceiro projeto foi executado utilizando a versão atual da ferramenta.

O primeiro projeto (SA) foi utilizado com a heurística SA, técnica implementada anteriormente na ferramenta. O resultado do projeto referente ao SA encontra-se na

Figura 28 – Produtividade final em relação ao ponto de troca do método de amostragem.



Fonte: Elaborado pelo autor.

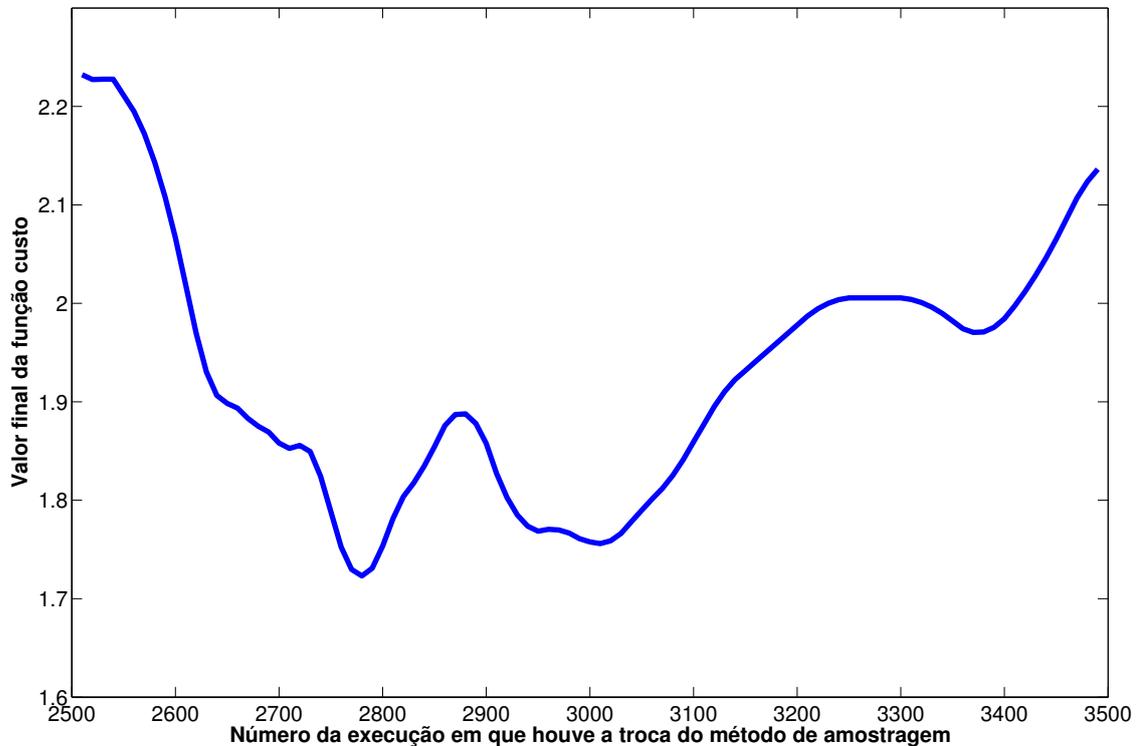
quinta coluna da Tabela 11. Todo fluxo de projeto foi executado novamente visando que a tecnologia e especificações eram diferentes no trabalho de Severo (2012). Este projeto utilizou de 100 rodadas (amostra), utilizando amostragem aleatória durante o processo do circuito, e após a otimização a execução de 10.000 rodadas utilizando o mesmo método, seguindo a metodologia proposta por Severo (2012).

O segundo projeto (PSO (YF)) foi executado com a heurística PSO e sem nenhuma técnica de *design centering* durante o processo de síntese, porém ao final da otimização foi aplicado o método de Monte Carlo com 10.000 rodadas, os quais seus resultados encontra-se na terceira coluna da Tabela 11.

Um terceiro projeto (PSO (AH-2780)) foi executado com a heurística PSO, este por fim utilizando a técnica de *design centering* utilizando o método de Monte Carlo, tendo como base os melhores resultados alcançados no trabalho, utilizou o método híbrido de amostragem com troca em 2780 execuções.

Os resultados das especificações do projeto executado utilizando o método de amostragem híbrida no ponto de troca ideal (2780), estão apresentados na quarta coluna

Figura 29 – Valor final da função custo em relação ao ponto de troca do método de amostragem.



Fonte: Elaborado pelo autor.

da Tabela 11.

Já os valores dimensionados para as variáveis do circuito são mostrados na Tabela 12.

A Figura 30 mostra a evolução da função custo do dimensionamento do OTA Miller, para os três projetos, verificando-se que a heurística efetuou uma otimização no problema, uma vez que a melhor solução da população inicia em um valor alto e é reduzido ao longo das execuções.

Com os resultados obtidos utilizando o método implementado por Severo (2012), que estão na quinta coluna da Tabela 11, é possível verificar que a especificação $Av\theta$ obteve um valor de $46,74dB$, assim não acalçando o mínimo imposto que foi de $50,00dB$. Os demais valores impostos para as especificações foram alcançados. Os resultados utilizando o PSO que estão na terceira e quarta coluna atingiram todas as especificações impostas no projeto.

Em relação aos valores de potência e área de gate, objetivos de otimização de

Tabela 11 – Resultados obtidos com o projeto automático do OTA Miller.

Especificação	Valor requerido	PSO (YF)	PSO (AH-2780)	SA
Av0 (dB)	$\geq 50,00$	51,46	51,05	46,74
GBW (MHz)	$\geq 2,00$	9,71	16,42	10,56
PM ($^{\circ}$)	$\geq 60,00$	60,01	63,91	60,16
SR (V/ μ s)	$\geq 5,00$	8,54	18,94	14,91
ICMR+ (V)	$\geq 0,40$	0,48	0,42	0,42
ICMR- (V)	$\leq -0,40$	-0,58	-0,58	-0,57
OS+ (V)	$\geq 0,40$	0,56	0,54	0,53
OS- (V)	$\leq -0,40$	-0,59	-0,59	-0,58
PSRR+ (dB)	$\geq 40,00$	60,74	61,02	56,10
PSRR- (dB)	$\leq -40,00$	-58,95	-62,91	-44,55
P_{Diss} (μ W)	minimizar	34,38	56,59	63,50
Área de gate (μ m ²)	minimizar	147,3	311,5	469,4
Função Custo	minimizar	0,52	1,22	1,32
Função Custo Final(10.000)	minimizar	8,47	1,52	14,8
Produtividade (%)	maximizar	0,28	91,10	12,13
Tempo de execução (min)	minimizar	97	380,8	431,9

Fonte: Elaborado pelo autor.

projeto, demonstraram melhores resultados comparados com os resultados obtidos em Severo (2012). A potência dissipada, comparando os resultados da execução com o cálculo da produtividade apenas no melhor resultado e com o cálculo da produtividade em todo escopo do projeto obteve uma redução de 45,85% e 10,88% respectivamente, e área de gate obteve uma redução de 68,61% e 33,63% respectivamente.

Os resultados da terceira e quarta coluna da Tabela 11, que foram executados com a heurística PSO, demonstram um projeto que não considera *design centering* em todo fluxo de projeto (PSO-YF), e um que considera o uso de *design centering* em todo fluxo de projeto (PSO AH-2780), e ambos após a otimização executam uma simulação Monte Carlo com amostragem aleatória simples de 10.000. Nestes resultados é possível analisar a melhoria imposta pelo *design centering*. A produtividade do circuito passou de 0,28% para 91,10%, a qual acarretou em uma melhoria de aproximadamente 325,3 vezes. O tempo computacional gasto pelo projeto que utiliza *design centering* é cerca de 3,9 vezes maior comparado ao que não utiliza.

Mesmo utilizando um tempo computacional maior, o resultado que considera

Tabela 12 – Valores das variáveis do OTA Miller dimensionadas a partir do projeto considerando amostragem híbrida (2780).

Variável	PSO (YF)	PSO (AH-2780)	SA
W_1	10,00 μm	10,00 μm	12,08 μm
L_1	3,11 μm	2,01 μm	4,62 μm
W_3	31,35 μm	14,81 μm	35,24 μm
L_3	0,50 μm	0,50 μm	0,95 μm
W_5	10,00 μm	24,03 μm	61,54 μm
L_5	0,50 μm	1,18 μm	0,55 μm
W_6	78,55 μm	90,04 μm	65,37 μm
L_6	0,50 μm	0,53 μm	0,64 μm
W_7	10,00 μm	79,34 μm	90,73 μm
L_7	0,50 μm	1,90 μm	1,98 μm
I_{ref}	100,00 μA	100,00 μA	175,59 μA
C_f	5,99 pF	4,44 pF	5,16 pF

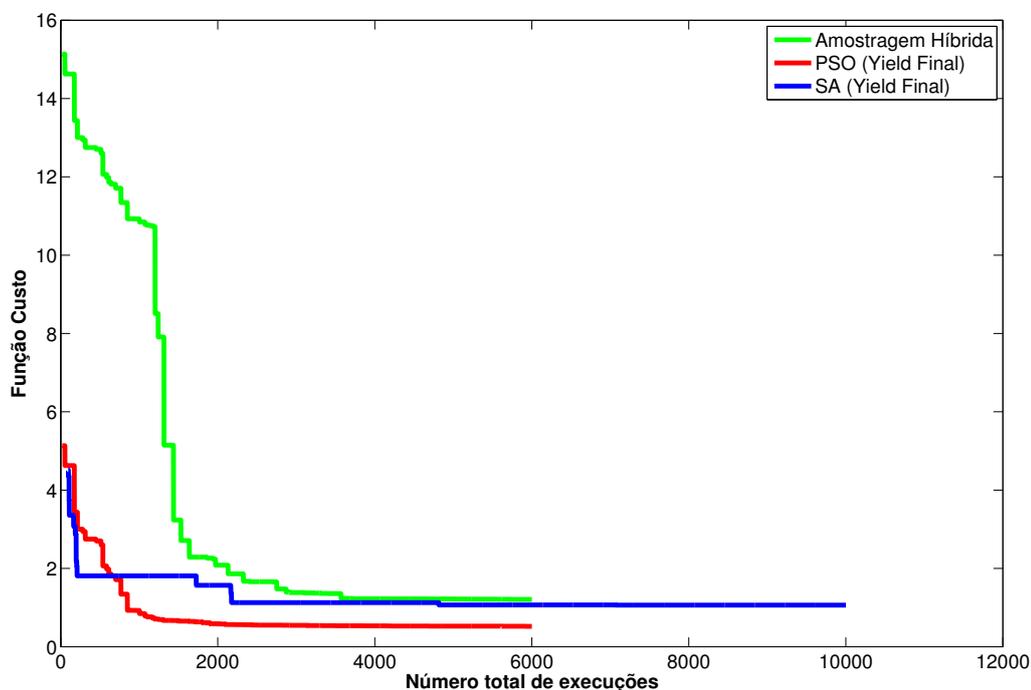
Fonte: Elaborado pelo autor.

design centering (PSO AH-2780) se mostra superior ao que não utiliza *design centering* (PSO-YF), pois pouco adianta ter um tempo computacional baixo se não atinge um valor satisfatório para a produtividade.

Comparando os projetos nos quais ambos consideram o *design centering* em todo fluxo de projeto, seus valores encontram-se na quarta e quinta coluna da Tabela 11. Nestes resultados foi possível verificar a melhoria imposta pelas novas técnicas implementadas na ferramenta UCAF (PSO(AH-2780)), na qual temos uma produtividade de 12,13% no trabalho de Severo (2012) (SA), e uma produtividade de 91,10% com as técnicas apresentadas neste trabalho. Desta forma, houve uma melhora na produtividade em cerca de 86,68%, lembrando que ambos os projetos utilizam da técnica de *design centering* durante o processo do circuito, e após a otimização a execução de 10.000 rodadas, com amostragem aleatória simples, para verificação da produtividade final. Consequentemente, temos um ganho em relação à função custo. O projeto proposto neste trabalho apresentou um valor para função custo final (considerando a execução das 10.000 rodadas sobre o melhor valor encontrado) de 1,52, enquanto que o projeto com SA obteve 14,8, o que acarreta em uma redução de 89,72%. Porém, os valores para função custo antes do cálculo da produtividade final foram praticamente equivalentes.

Com relação ao tempo, o método proposto por Severo (2012) teve um tempo superior de execução, que foi de 51,1 minutos. Nota-se que o projeto com SA e amostragem

Figura 30 – Evolução função custo para o melhor indivíduo da população durante o processo de otimização.



Fonte: Elaborado pelo autor.

aleatória simples utilizou mais iterações comparado o projeto executado com o PSO junto com a amostragem híbrida, deste modo justificando o aumento no tempo. Porém, não obteve uma produtividade equivalente ao do trabalho proposto, assim demonstrando uma menor eficiência em relação ao conjunto de tempo computacional e produtividade.

Este ganho computacional pode ser comprovado analisando a Tabela 8, na qual foi executado um projeto com *design centering* considerando apenas a amostragem aleatória simples juntamente com o algoritmo PSO. Este teste gastou 485,4 minutos em toda sua execução e obteve uma produtividade final abaixo do executado com amostragem híbrida, que foi de 87,35%.

5.3 CONCLUSÃO

Neste capítulo foi abordada a análise dos métodos de amostragem aleatório e por hipercubo latino, com base em um OTA Miller.

Foi demonstrado que uma técnica de amostragem híbrida se torna eficiente em relação à estimativa de produtividade e tempo computacional. Os melhores resultados foram encontrados quando a amostragem por hipercubo latino utilizou 50 amostras e o

aleatório simples 100. Além disso, demonstrou-se que existe um ponto ótimo de troca entre estes dois métodos de amostragem, que ficou definido em 2780 execuções.

Este capítulo demonstrou também a aplicação da ferramenta UCAF nos projetos de um OTA Miller. Os projetos foram executados com diferentes configurações da ferramenta. Na execução sem *design centering* (PSO (YF)) foi verificado que as especificações finais do circuito encontram-se próximas à fronteira do espaço de projeto, assim demonstrando um valor baixa de produtividade que foi em torno de 0,28%. Com a utilização de *design centering* a produtividade dos circuitos aumenta. O projeto SA teve uma produtividade de 12,13% e um tempo computacional mais elevado em relação ao projeto PSO (AH-2780), que teve uma produtividade de 91,10%. Portanto, por mais que ambos os projetos utilizem *design centering* o projeto PSO (AH-2780) se demonstrou superior ao SA, obtendo uma maior produtividade em um menor tempo computacional.

6 CONCLUSÃO

Este trabalho apresentou a implementação de novas técnicas de projeto na ferramenta UCAF, que visa o dimensionamento automático de projeto de circuitos integrados analógicos através de heurísticas de otimização, as quais aceleraram o processo de busca por soluções otimizadas para as especificações do circuito, considerando a análise da produtividade. Foi realizado um estudo sobre a ferramenta UCAF, técnicas de *design centering* e a heurística PSO, para que assim a ferramenta UCAF pudesse ser aprimorada.

Foram abordados os dimensionamentos de dois amplificadores, do tipo OTA Miller e Telescopic, os quais foram explorados utilizando a heurística PSO. A heurística obteve um bom comportamento para ambos os casos quando comparada ao SA. Obteve-se uma redução da função custo de 2,14 utilizando o SA, para 0,41 com o PSO no projeto do OTA Miller e de 0,86 utilizando o SA, para 0,23 com o PSO no projeto do amplificador Telescopic.

Os resultados obtidos indicam que o algoritmo PSO atinge os melhores resultados quando o tamanho da população é igual a 100, o número total de iterações é igual a 60 e a semente randômica é definida como zero. A utilização do cálculo da inércia junto ao cálculo da velocidade da partícula permitiu um balanço entre a busca por uma solução local e global.

Para o cálculo da produtividade foi desenvolvida uma técnica híbrida de amostragem para a análise Monte Carlo. A metodologia proposta foi baseada nos métodos de amostragem por hipercubo latino e amostragem aleatória simples. Para a análise dos métodos de amostragem, foi utilizado o OTA Miller, no qual a amostragem híbrida se demonstrou eficiente em relação à estimativa de produtividade e tempo computacional. Os melhores resultados foram encontrados quando a amostragem por hipercubo latino utilizou 50 amostras e o aleatório simples 100. Além disso, demonstrou-se que existe um ponto ótimo de troca entre estes dois métodos de amostragem, que ficou definido na execução de número 2780, de um total de 6000 execuções.

Os projetos foram executados com diferentes configurações da ferramenta. Na execução sem *design centering* (PSO (YF)) foi verificado que as especificações finais do circuito encontram-se próximas à fronteira do espaço de projeto, assim demonstrando um valor de baixa de produtividade que foi em torno de 0,28%. Com a utilização de *design centering* a produtividade dos circuitos aumenta. O projeto SA teve uma produtividade de 12,13% e um tempo computacional mais elevado em relação ao projeto PSO (AH-2780), que teve uma produtividade de 91,10%. Portanto, por mais que ambos os projetos utilizem *design centering* o projeto PSO (AH-2780) se demonstrou superior ao SA, obtendo uma maior produtividade em um menor tempo computacional.

Como trabalhos futuros, propõe-se avaliar o desempenho dos métodos de amostra-

gem em outras topologias, assim demonstrando a sua eficiência. Também é necessário um estudo sobre as técnicas de definição dos valores iniciais das variáveis livres, investigando se a obtenção de uma boa solução esta ligada à solução inicial.

REFERÊNCIAS

- ALLEN, P.; HOLBERG, D. *CMOS Analog Circuit Design*. [S.l.]: Oxford University Press, 2002. (Oxford series in electrical and computer engineering). ISBN 9780195116441. Citado 2 vezes nas páginas 39 e 51.
- BALKIR, S.; DUNGAR, G.; OGRENCI, A. S. *Analog VLSI Design Automation*. [S.l.]: CRC Press, 2003. Citado na página 29.
- BARROS, M.; GUILHERME, J.; HORTA, N. *Analog Circuits and Systems Optimization Based on Evolutionary Computation Techniques*. [S.l.]: Springer, 2010. Citado na página 42.
- BARROS, M.; GUILHERME, J.; HORTA, N. Analog circuits optimization based on evolutionary computation techniques. *Integration, the {VLSI} Journal*, v. 43, n. 1, p. 136 – 155, 2010. ISSN 0167-9260. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0167926009000406>>. Citado na página 26.
- BINDER, K.; HEERMANN, D. *Monte Carlo Simulation in Statistical Physics: An Introduction*. [S.l.]: Springer Berlin Heidelberg, 1992. (Graduate Texts in Physics). Citado na página 33.
- BRINKMAN, W.; HAGGAN, D.; TROUTMAN, W. A history of the invention of the transistor and where it will lead us. *Solid-State Circuits, IEEE Journal of*, v. 32, n. 12, p. 1858–1865, Dec 1997. Citado na página 21.
- CANELAS, A. et al. Yield optimization using k-means clustering algorithm to reduce monte carlo simulations. In: *2016 13th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. [S.l.: s.n.], 2016. p. 1–4. Citado na página 25.
- DIGALAKIS, J.; MARGARITIS, K. On benchmarking functions for genetic algorithms. *International Journal of Computer Mathematics*, v. 77, n. 4, p. 481–506, 2001. Citado na página 40.
- DRENNAN, P. G.; MCANDREW, C. C. Understanding mosfet mismatch for analog design. *IEEE Journal of Solid-State Circuits*, v. 38, n. 3, p. 450–456, Mar 2003. ISSN 0018-9200. Citado na página 31.
- GIELEN, G.; RUTENBAR, R. Computer-aided design of analog and mixed-signal integrated circuits. *Proceedings of the IEEE*, v. 88, n. 12, p. 1825–1854, Dec 2000. Citado na página 21.
- GLOVER, F.; KOCHENBERGER, G. A. *Handbook of Metaheuristics*. [S.l.]: Kluwer Academic Publishers, 2003. Citado na página 41.
- GRAEB, H. E. *Analog Design Centering and Sizing*. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2007. ISBN 1402060033, 9781402060038. Citado 3 vezes nas páginas 29, 32 e 34.
- GUERRA-GOMEZ, I.; TLELO-CUAUTLE, E.; FRAGA, L. G. de la. Ocba in the yield optimization of analog integrated circuits by evolutionary algorithms. In: *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*. [S.l.: s.n.], 2015. p. 1933–1936. ISSN 0271-4302. Citado na página 26.

- HEPPNER, F.; GREANDER, U. A stochastic nonlinear model for coordinated bird flocks. In: KRASNER, E. (Ed.). *The ubiquity of chaos*. [S.l.]: AAAS Publications, 1990. p. 233–238. Citado na página 44.
- HOLLAND, J. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. [S.l.]: University of Michigan Press, 1975. Citado na página 40.
- HOUCK, C. R.; JOINES, J. A.; KAY, M. G. *A Genetic Algorithm for Function Optimization: A Matlab Implementation*. [S.l.], 1995. Citado na página 41.
- JAFARI, A. et al. Design of analog integrated circuits by using genetic algorithm. In: *Computer Engineering and Applications (ICCEA), 2010 Second International Conference on*. [S.l.: s.n.], 2010. v. 1, p. 578–581. Citado na página 25.
- JAFFARI, J.; ANIS, M. On efficient lhs-based yield analysis of analog circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 30, n. 1, p. 159–163, Jan 2011. ISSN 0278-0070. Citado na página 25.
- KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: *Neural Networks, 1995. Proceedings., IEEE International Conference on*. [S.l.: s.n.], 1995. v. 4, p. 1942–1948 vol.4. Citado 4 vezes nas páginas 23, 43, 45 e 51.
- KHOUEAS, A.; DERIEUX, A. Fault simulation for analog circuits under parameter variations. *Journal of Electronic Testing*, v. 16, n. 3, p. 269–278, 2000. Disponível em: <<http://dx.doi.org/10.1023/A:1008351601024>>. Citado na página 34.
- LIN, Y.; CHEN, D.; GEIGER, R. Yield enhancement with optimal area allocation for ratio-critical analog circuits. *IEEE Transactions on Circuits and Systems I: Regular Papers*, v. 53, n. 3, p. 534–553, mar. 2006. Citado na página 22.
- LIU, B. et al. Analog circuit optimization system based on hybrid evolutionary algorithms. *Integration, the {VLSI} Journal*, v. 42, n. 2, p. 137 – 148, 2009. Citado na página 41.
- MAHMOUDI, H.; ZIMMERMANN, H. On optimal latin hypercube design for yield analysis of analog circuits. In: *Microelectronics (Austrochip), 2015 Austrian Workshop on*. [S.l.: s.n.], 2015. p. 46–49. Citado na página 25.
- MATWORKS. *A Genetic Algorithm for Function Optimization: A Matlab Implementation*. [S.l.], 2012b. Disponível em: <<http://www.mathworks.com/products/global-optimization/description7.html>>. Citado na página 41.
- MCKAY, M. D.; BECKMAN, R. J.; CONOVER, W. J. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, v. 21, n. 2, p. pp. 239–245, 1979. Citado 2 vezes nas páginas 23 e 35.
- MEDEIROS, J. A. C. C. *Enxame de Partículas como Ferramenta de Otimização em Problemas Complexos da Engenharia Nuclear*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005. Citado 3 vezes nas páginas 44, 45 e 47.
- MITCHELL, M. *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998. ISBN 0262631857. Citado 2 vezes nas páginas 40 e 41.

- MOORE, G. E. Cramming More Components onto Integrated Circuits. *Electronics*, IEEE, v. 38, n. 8, p. 114–117, abr. 1965. ISSN 0018-9219. Citado na página 21.
- MUTLU, A. A.; RAHMAN, M. Statistical methods for the estimation of process variation effects on circuit operation. *IEEE Transactions on Electronics Packaging Manufacturing*, v. 28, n. 4, p. 364–375, Oct 2005. ISSN 1521-334X. Citado na página 31.
- ORSHANSKY, M.; NASSIF, S.; BONING, D. *Design for Manufacturability and Statistical Design: A Constructive Approach*. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2010. ISBN 1441940448, 9781441940445. Citado na página 31.
- OSMAN, I.; KELLY, J. *Meta-Heuristics: Theory and Applications*. [S.l.]: Springer, 1996. Citado na página 41.
- PARSOPOULOS, K.; VRAHATIS, M. Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing*, Kluwer Academic Publishers, v. 1, n. 2-3, p. 235–306, 2002. Citado na página 44.
- PHELPS, R. et al. Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, v. 19, n. 6, p. 703–717, Jun 2000. Citado na página 42.
- POLI, R.; KENNEDY, J.; BLACKWELL, T. Particle swarm optimization. *Swarm Intelligence*, Springer US, v. 1, n. 1, p. 33–57, 2007. Citado 2 vezes nas páginas 44 e 47.
- REYNOLDS, C. W. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, ACM, New York, NY, USA, v. 21, n. 4, p. 25–34, 1987. Citado na página 44.
- SEVERO, L. C. *Uma Ferramenta para o Dimensionamento Automático de Circuitos Integrados Analógicos Considerando Análise de Produtividade*. 2012. Citado 19 vezes nas páginas 16, 22, 30, 37, 38, 40, 42, 49, 56, 57, 60, 63, 65, 66, 78, 79, 80, 81 e 82.
- SHI, Y.; EBERHART, R. A modified particle swarm optimizer. In: *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*. [S.l.: s.n.], 1998. p. 69–73. Citado na página 48.
- SHI, Y.; EBERHART, R. Empirical study of particle swarm optimization. In: *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*. [S.l.: s.n.], 1999. v. 3. Citado na página 48.
- SHUHUA, W. et al. A modified particle swarm optimization algorithm. In: *Neural Networks and Brain, 2005. ICNN B '05. International Conference on*. [S.l.: s.n.], 2005. v. 1, p. 318–321. Citado 2 vezes nas páginas 48 e 51.
- SOBOL, I. M. *A primer for the Monte Carlo method; 2nd ed*. Boca Raton, FL: CRC Press, 1994. Transl. from the Russian : Metod Monte-Karlo. Disponível em: <<https://cds.cern.ch/record/320446>>. Citado na página 33.
- XU, Y. et al. Regular Analog/RF Integrated Circuits Design Using Optimization With Recourse Including Ellipsoidal Uncertainty. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 28, n. 5, p. 623–637, 2009. Citado na página 22.

YU, G.; LI, P. Hierarchical Analog/Mixed-Signal Circuit Optimization Under Process Variations and Tuning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 30, n. 2, p. 313–317, fev. 2011. Citado na página 22.