



DISSERTAÇÃO DE MESTRADO EM ENGENHARIA
ELÉTRICA

**PROPOSTA DE UMA METODOLOGIA DE
OTIMIZAÇÃO EVOLUCIONÁRIA ROBUSTA
UTILIZANDO A TRANSFORMADA UNSCENTED
APLICÁVEL A CIRCUITOS DE RF/MICROONDAS**

Maisa Netto Leidemer

Brasília, março de 2009

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA
ELÉTRICA

**PROPOSTA DE UMA METODOLOGIA DE
OTIMIZAÇÃO EVOLUCIONÁRIA ROBUSTA
UTILIZANDO A TRANSFORMADA UNSCENTED
APLICÁVEL A CIRCUITOS DE RF/MICROONDAS**

Maisa Netto Leidemer

Dissertação de mestrado submetida ao Departamento de Engenharia Elétrica da Faculdade de Tecnologia da Universidade de Brasília, como parte dos requisitos necessários para a obtenção do grau de mestre.

Banca Examinadora

Prof. Dr. Paulo Henrique Portela de Carvalho, _____
UnB/ ENE (Orientador)

Prof. Dr. Leonardo R. A. X. de Menezes, UnB/
ENE (Examinador Interno) _____

Prof. Dr. Leonardo da Cunha Brito, UFG
(Examinador Externo) _____

FICHA CATALOGRÁFICA

LEIDEMER, MAISA NETTO

Proposta de uma Metodologia de Otimização Evolucionária

Robusta Utilizando a Transformada Unscented

Aplicável a Circuitos de RF/Microondas. [Distrito Federal] 2009.

xii, 108p., 297 mm (ENE/FT/UnB, Mestre. Dissertação de Mestrado.

Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Elétrica.

- | | |
|---------------------------|------------------------------|
| 1. Otimização Robusta | 2. Análise de Sensibilidade |
| 3. Transformada Unscented | 4. Algoritmos Evolucionários |
| 5. Amplificador classe E | 6. NSGA-II |
| I. ENE/FT/UnB | II. Título (série) |

REFERÊNCIA BIBLIOGRÁFICA

LEIDEMER, M. N. (2009). Proposta de uma Metodologia de Otimização Evolucionária Robusta utilizando a Transformada Unscented Aplicável a Circuitos de RF/Microondas. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGENE.DM - 377/09, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 108p.

CESSÃO DE DIREITOS

NOME DO AUTOR: Maisa Netto Leidemer.

TÍTULO DA DISSERTAÇÃO DE MESTRADO: Proposta de uma Metodologia de Otimização Evolucionária Robusta utilizando a Transformada Unscented Aplicável a Circuitos de RF/Microondas.

GRAU: Mestre Ano: 2009

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta dissertação de mestrado pode ser reproduzida sem a autorização por escrito do autor.

Maisa Netto Leidemer
Departamento de Engenharia Elétrica
Campus Universitário Darcy Ribeiro
Caixa Postal 4386
CEP: 70919-970
Brasília - DF

Dedicatória

Dedico, com todo meu carinho, aos meus pais, tão especiais!

Maisa Netto Leidemer

Agradecimentos

A Deus, princípio eterno de nossos breves dias. Aos cuidados de Nossa Senhora em cada momento de dificuldade.

Aos meus pais por todo cuidado que têm comigo, pelo apoio e confiança, e sobretudo pelo ambiente familiar aconchegante. À minha irmã amada, por estar sempre do meu lado! Aos meus avós queridos por entenderem meus momentos de estudo e minhas ausências.

Ao Prof. Paulo, paciente e dedicado, pela oportunidade, pela orientação e pelos valiosos ensinamentos, por me ajudar a tornar real este momento que era um sonho desde a graduação.

A todos os professores do Departamento de Engenharia Elétrica que me formaram e especialmente ao Prof. Leonardo, prestativo, pela ajuda neste trabalho.

Aos amigos que tiveram papel importante nesta caminhada: Charles, Filipe, Eduardo e Carol, Mintsu, Tiago, Jacir, Bruno e Gabriela; e a todos os amigos e amigas que me incentivaram e me acompanharam nestes tempos.

Ao Dr. Heliomar e ao Fernando, pelo apoio essencial e compreensão nestes últimos meses.

À comunidade de software livre e aos que me apresentaram as poderosas ferramentas livres.

Muito obrigada!

Maisa Netto Leidemer

RESUMO

Problemas de otimização multiobjetivo na presença de incertezas são muito frequentes na engenharia. A fabricação de dispositivos, como os circuitos de RF/microondas, muitas vezes são afetadas por incertezas associadas aos componentes de fabricação. Desta forma, é desejável uma metodologia de projeto robusto que considere estas incertezas durante o processo de otimização. Esta dissertação propõe tal metodologia de projeto robusto a partir dos conceitos de otimização multiobjetivo evolucionária e análise de sensibilidade. A Transformada *Unscented* é utilizada para análise da sensibilidade e como ferramenta no algoritmo de otimização robusta. O projeto robusto é definido em três etapas: a caracterização da incerteza associada ao problema, a definição dos parâmetros sensíveis de projeto e a otimização robusta. A metodologia proposta é testada em funções analíticas e é realizado o projeto robusto de um amplificador de potência classe E, que opera na faixa de UHF.

ABSTRACT

Multi-objective optimization problems in the presence of uncertainties occur very often in engineering. The manufacture of devices, for example RF/microwaves circuits, is surrounded by uncertainties associated to the manufacturing components. So it's desirable to establish a method that renders a robust design in which uncertainties are considered during the optimization process. This dissertation presents this method utilizing the concepts of evolutionary multi-objective optimization and sensitivity analysis. The Unscented Transformation is used in the sensitivity analysis and as an auxiliary tool of robust optimization algorithm. The robust design is a three-part project: the characterization of the uncertainties associated with the problem, the identification of the sensitivity parameters and the robust optimization. The method presented is tested in analytical functions and the robust project of a class-E power amplifier operating in UHF band is elaborated.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	OBJETIVOS DA DISSERTAÇÃO	2
1.3	APRESENTAÇÃO DO MANUSCRITO	2
2	OTIMIZAÇÃO ROBUSTA	5
2.1	FORMULAÇÃO CLÁSSICA DO PROBLEMA DE OTIMIZAÇÃO MULTIOBJETIVO	6
2.2	ALGORITMOS EVOLUCIONÁRIOS	8
2.2.1	ALGORITMOS GENÉTICOS	11
2.2.2	NSGA-II	12
2.3	OTIMIZAÇÃO ROBUSTA	18
2.3.1	ANÁLISE DE SENSIBILIDADE	19
2.3.2	PROJETO ROBUSTO	23
2.3.3	ANÁLISE DO PIOR CASO	25
2.3.4	ANÁLISE ESTATÍSTICA	27
2.3.5	CONSIDERAÇÕES FINAIS	29
3	PROPOSTA DE UMA METODOLOGIA DE OTIMIZAÇÃO EVOLUCIONÁRIA ROBUSTA	31
3.1	A TRANSFORMADA UNSCENTED	31
3.1.1	DETERMINAÇÃO DOS PONTOS SIGMA	34
3.1.2	COMPARAÇÃO DA UT COM O MÉTODO DE MONTE CARLO	36
3.2	METODOLOGIA DE PROJETO ROBUSTO	46
3.2.1	MODELAGEM DAS INCERTEZAS ASSOCIADAS ÀS VARIÁVEIS DE ENTRADA	48
3.2.2	DEFINIÇÃO DOS PARÂMETROS SENSÍVEIS	48
3.2.3	PROCESSO DE OTIMIZAÇÃO	52
3.3	AVALIAÇÃO DA METODOLOGIA	54
3.3.1	PROBLEMA 1	54
3.3.2	PROBLEMA 2	56
3.3.3	PROBLEMA 3	57
3.3.4	PROBLEMA 4	62
3.3.5	CONSIDERAÇÕES FINAIS	79
4	OTIMIZAÇÃO ROBUSTA DO AMPLIFICADOR CLASSE E	81
4.1	AMPLIFICADORES DE POTÊNCIA	81
4.1.1	AMPLIFICADOR DE POTÊNCIA CLASSE E	83
4.2	OTIMIZAÇÃO ROBUSTA DO AMPLIFICADOR CLASSE E	85
4.2.1	OTIMIZAÇÃO CLÁSSICA COM O NSGA-II	86
4.2.2	COMPARAÇÃO DA UT COM O MÉTODO DE MONTE CARLO	87
4.2.3	PROJETO ROBUSTO DO AMPLIFICADOR CLASSE E	90
4.2.4	CONSIDERAÇÕES FINAIS	96
5	CONCLUSÕES	97
	REFERÊNCIAS BIBLIOGRÁFICAS	99

ANEXOS	103
I ARQUIVOS UTILIZADOS PELO <i>GOLDEN GATE</i>	105

LISTA DE FIGURAS

2.1	Ilustração do conceito de dominância.	8
2.2	O algoritmo evolucionário padrão.	10
2.3	Ilustração de um cruzamento.	12
2.4	Ilustração de uma mutação.	12
2.5	Ilustração de como calcular a distância de aglomeração [1].	14
2.6	Fluxograma do NSGA-II.	17
2.7	Fluxograma do método de Monte Carlo.	20
2.8	Visualização da distribuição das amostras do AE.	22
2.9	Visualização da distribuição das amostras do AE, no domínio das variáveis de entrada.	22
2.10	Comparação entre uma solução ótima robusta e uma solução ótima não-robusta [2].	23
2.11	Comparação entre uma solução ótima robusta e uma solução ótima não-robusta [2].	24
2.12	Diferentes possibilidades para definição de um conjunto de incertezas [2]: (a) retangular e (b) elipsoidal.	25
3.1	Princípio da transformada <i>Unscented</i> [3].	32
3.2	Função densidade de probabilidade contínua e a aproximação discreta.	35
3.3	Função $f(x)$ e seu valor esperado $E[f(x)]$ para os casos analisados.	38
3.4	Erro absoluto do valor esperado.	38
3.5	Comparação do desvio padrão $\sigma_{f(x)}$ analítico com os casos analisados.	39
3.6	Erro absoluto do desvio padrão.	39
3.7	Função $f(x)$ para as duas distribuições: (a) $x \sim N(\frac{2\pi}{3}, \frac{1}{100})$ e (b) $x \sim U(\frac{2\pi}{3} - \frac{\pi}{10}, \frac{2\pi}{3} + \frac{\pi}{10})$	40
3.8	Evolução de $E[f(x)]$ para o método de Monte Carlo para a variável aleatória x de distribuição uniforme (10.000 amostras).	42
3.9	Evolução de $E[f(x)]$ para o método de Monte Carlo para a variável aleatória x de distribuição gaussiana (10.000 amostras).	42
3.10	Evolução de $E[f(x)]$ para o método de Monte Carlo para a variável aleatória x de distribuição uniforme (100.000 amostras).	43
3.11	Evolução de $E[f(x)]$ para o método de Monte Carlo para a variável aleatória x de distribuição gaussiana (100.000 amostras).	43
3.12	Fluxograma da metodologia de projeto robusto.	47
3.13	Fluxograma para a determinação das variáveis sensíveis de projeto.	50
3.14	Visualização da função $f(x_1, x_2, x_3)$ para cada variável.	52
3.15	Opções para otimização robusta.	53
3.16	Comparação do resultado da otimização clássica com a otimização robusta.	55
3.17	Detecção de múltiplas soluções robustas.	55
3.18	Detecção de múltiplas soluções robustas [4].	56
3.19	Comparação do resultado da otimização clássica com a otimização robusta.	57
3.20	Soluções ótimas robustas obtidas pelo algoritmo MOGAPSO [5].	58
3.21	Verificação do valor esperado e do desvio padrão para uma amostragem do espaço de busca.	58
3.22	Frente ótima de Pareto dada obtida a partir do NSGA-II.	59
3.23	Soluções mais robustas entre as soluções ótimas de Pareto para cada função objetivo calculada pela variância e pela métrica B_j	60
3.24	Otimização robusta pelo método 1 para diferentes níveis de σ	61
3.25	Otimização robusta pelo método 2 para diferentes níveis de σ	61
3.26	Avaliação do impacto do parâmetro λ na otimização robusta pelo método 1.	62
3.27	Otimização clássica pelo NSGA-II.	63
3.28	Verificação do comportamento de $f_1(x_1, x_2, x_3)$ para cada variável.	65
3.29	Verificação do comportamento de $f_2(x_1, x_2, x_3)$ para cada variável.	65

3.30	Resultado da otimização para o cenário 1.	69
3.31	Análise de sensibilidade das soluções ótimas não robustas.	71
3.32	Análise de sensibilidade das soluções ótimas robustas dadas pelo método 1.	71
3.33	Análise de sensibilidade das soluções ótimas robustas dadas pelo método 2.	71
3.34	Análise de sensibilidade das soluções ótimas não robustas.	72
3.35	Análise de sensibilidade das soluções ótimas robustas dadas pelo método 1.	72
3.36	Análise de sensibilidade das soluções ótimas robustas dadas pelo método 2.	72
3.37	Avaliação do impacto do parâmetro λ na otimização robusta pelo método 1 para o cenário 1.	74
3.38	Avaliação do impacto do parâmetro λ na otimização robusta pelo método 1 para o cenário 1.	74
3.39	Resultado da otimização para o cenário 2.	75
3.40	Análise de sensibilidade das soluções ótimas não robustas.	77
3.41	Análise de sensibilidade das soluções ótimas robustas dadas pelo método 1.	77
3.42	Análise de sensibilidade das soluções ótimas robustas dadas pelo método 2.	77
3.43	Análise de sensibilidade das soluções ótimas não robustas.	78
3.44	Análise de sensibilidade das soluções ótimas robustas dadas pelo método 1.	78
3.45	Análise de sensibilidade das soluções ótimas robustas dadas pelo método 2.	78
4.1	Circuito classe E apresentado em [6] (adaptado).	83
4.2	Topologia do circuito classe E.	83
4.3	Formas de onda de tensão e corrente no dreno para o amplificador classe E.	84
4.4	Otimização clássica do amplificador classe E utilizando o NSGA-II.	87
4.5	Frente ótima de Pareto para a otimização do amplificador classe E utilizando o NSGA-II. ...	88
4.6	Verificação da eficiência dado uma variação do parâmetro C_o para duas soluções da otimização pelo NSGA-II.	91
4.7	Verificação da potência de saída dado uma variação do parâmetro C_o para duas soluções da otimização pelo NSGA-II.	92
4.8	Comparação dos resultados de otimização robusta para o classe E.	93
4.9	Análise de sensibilidade das soluções em relação a eficiência.	94
4.10	Análise de sensibilidade das soluções em relação a potência de saída.	94
4.11	Verificação da eficiência dado uma variação do parâmetro C_o para duas soluções da otimização robusta pelo método 1.	95
4.12	Verificação da potência de saída dado uma variação do parâmetro C_o para duas soluções da otimização robusta pelo método 1.	96

LISTA DE TABELAS

3.1	Pontos sigma e pesos para distribuição gaussiana $N \sim (0, \sigma^2)$	34
3.2	Pontos sigma e pesos para distribuição uniforme $U \sim (-\sigma, \sigma)$	34
3.3	Arranjo da UT para duas variáveis.....	36
3.4	Comparação do valor esperado e da variância de $f(x)$ para $x \sim U(\frac{2\pi}{3} - \frac{\pi}{10}, \frac{2\pi}{3} + \frac{\pi}{10})$	41
3.5	Comparação do valor esperado e da variância de $f(x)$ para $x \sim N(\frac{2\pi}{3}, \frac{1}{100})$	41
3.6	Comparação do valor esperado dado pelo método de Monte Carlo e pela UT.....	44
3.7	Comparação da variância dado pelo método de Monte Carlo e pela UT.....	44
3.8	Percentual de erro absoluto em relação ao método de Monte Carlo com 100.000 amostras. ..	45
3.9	Comparação do valor esperado e do desvio padrão dado pelo método de Monte Carlo e pela UT.....	46
3.10	Percentual de erro absoluto em relação ao método de Monte Carlo com 1.000.000 amostras.	46
3.11	Análise dos parâmetros sensíveis de projeto.....	51
3.12	Coefficiente de correlação entre as variáveis x_1, x_2 e x_3 e a função $f(x_1, x_2, x_3)$	51
3.13	Parâmetros de simulação do NSGA-II.....	64
3.14	Análise de sensibilidade para a função $f_1(x_1, x_2, x_3)$ de acordo com o cenário 1.....	67
3.15	Análise de sensibilidade para a função $f_2(x_1, x_2, x_3)$ de acordo com o cenário 1.....	67
3.16	Análise de sensibilidade para a função $f_1(x_1, x_2, x_3)$ de acordo com o cenário 2.....	67
3.17	Análise de sensibilidade para a função $f_2(x_1, x_2, x_3)$ de acordo com o cenário 2.....	67
3.18	Análise de sensibilidade da frente ótima de Pareto para a função $f_1(x_1, x_2, x_3)$ de acordo com o cenário 1.....	68
3.19	Análise de sensibilidade da frente ótima de Pareto para a função $f_2(x_1, x_2, x_3)$ de acordo com o cenário 1.....	68
3.20	Análise de sensibilidade da frente ótima de Pareto para a função $f_1(x_1, x_2, x_3)$ de acordo com o cenário 2.....	68
3.21	Análise de sensibilidade da frente ótima de Pareto para a função $f_2(x_1, x_2, x_3)$ de acordo com o cenário 2.....	68
3.22	Análise das soluções ótimas para a função $f_1(x_1, x_2, x_3)$ de acordo com o cenário 1.....	70
3.23	Análise das soluções ótimas para a função $f_2(x_1, x_2, x_3)$ de acordo com o cenário 1.....	70
3.24	Análise das simulações para a função $f_1(x_1, x_2, x_3)$ de acordo com o cenário 1.....	73
3.25	Análise das simulações para a função $f_2(x_1, x_2, x_3)$ de acordo com o cenário 1.....	73
3.26	Análise dos parâmetros sensíveis para a função $f_1(x_1, x_2, x_3)$ de acordo com o cenário 2....	76
3.27	Análise dos parâmetros sensíveis para a função $f_2(x_1, x_2, x_3)$ de acordo com o cenário 2....	76
4.1	Faixa de variação dos elementos constitutivos do classe E.....	86
4.2	Parâmetros de configuração do NSGA-II.....	86
4.3	Valor dos componentes do amplificador classe E (figura 4.2).....	88
4.4	Comparação entre o método de Monte Carlo e a UT para o parâmetro C_o com $\sigma_{C_o} = 0,025$.	89
4.5	Comparação entre o método de Monte Carlo e a UT para o parâmetro C_o com $\sigma_{C_o} = 0,05$..	89
4.6	Comparação entre o método de Monte Carlo e a UT para o parâmetro L_x com $\sigma_{L_x} = 0,125$.	89
4.7	Comparação entre o método de Monte Carlo e a UT para o parâmetro L_x com $\sigma_{L_x} = 0,25$.	89
4.8	Análise de sensibilidade da frente ótima de Pareto.....	90

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Muitos problemas de engenharia podem ser tratados como problemas de busca e otimização de soluções. Otimizar significa buscar a solução ótima de determinado problema. Contudo, para uma classe específica de problemas de engenharia, não basta apenas encontrar as melhores soluções para os problemas, mas encontrar aquelas soluções mais robustas dado um cenário de incertezas. Por soluções robustas, entende-se soluções que mantêm o desempenho especificado mesmo que sujeitas a variação de seus parâmetros constitutivos. Por exemplo, determinado dispositivo eletrônico pode ser fabricado com componentes que possuem certa variação em seus valores nominais, mas ainda assim o dispositivo deve ter seu funcionamento garantido para um pior arranjo de componentes.

A solução de problemas de otimização pode ser obtida por diversos métodos, dependendo das características e particularidades do problema. Um método que merece destaque são os Algoritmos Evolucionários – AE. Estes são tidos como métodos estocásticos e podem ser enquadrados na categoria de programação não-linear. Algoritmos evolucionários têm-se tornado uma boa escolha para solução de problemas de otimização que sejam complexos demais para serem resolvidos pelos métodos tradicionais, que utilizam-se comumente de informações sobre as derivadas para a obtenção de soluções.

Os AEs objetivam a busca de soluções ótimas para os problemas, mas, com a abordagem adequada, podem ser utilizados na busca de soluções robustas. Neste sentido, um método bastante conhecido para definir a robustez de uma solução é a Análise de Sensibilidade – AS [2]. Com a análise de sensibilidade, o desempenho do projeto é verificado em torno da solução nominal obtida. Será mostrado nesta dissertação como a AS é capaz de avaliar o comportamento de determinada solução na presença de incertezas.

A análise de sensibilidade é frequentemente realizada pelo método de Monte Carlo. Contudo, este método requer grande esforço computacional. Uma alternativa para este problema é a Transformada *Unscented* – UT (do inglês *Unscented Transform*) [3, 7]. A UT permite o cálculo das medidas estatísticas de uma variável aleatória que sofre uma transformação não-linear a partir de uma aproximação da distribuição de probabilidade por pontos discretos.

Desta forma, um método que utilize a análise de sensibilidade juntamente com os algoritmos evolucionários permitirá a obtenção de soluções robustas para problemas afetados pelas incertezas de

projeto. Esta dissertação apresenta um algoritmo de otimização robusta utilizando-se da UT e do algoritmo evolucionário denominado NSGA-II [1].

Contudo, o projeto robusto envolve etapas além da implementação do algoritmo de otimização robusta. Neste sentido, esta dissertação apresenta uma metodologia de projeto robusto, com todas as etapas necessárias para a realização eficaz e eficiente de um projeto robusto.

A metodologia de projeto robusto proposta será utilizada para o projeto robusto de um amplificador classe E, que opera na faixa de UHF. O projeto de amplificadores classe E é bastante estudado na literatura [6], por ser um amplificador de alta eficiência e por apresentar um circuito simples com poucos componentes.

1.2 OBJETIVOS DA DISSERTAÇÃO

Pretende-se neste trabalho abordar o tema de otimização robusta e a definição de uma metodologia de projeto robusto para otimização de circuitos de RF/Microondas.

Os objetivos desta dissertação são:

- abordagem do tema de otimização robusta;
- proposta e validação de um método de otimização robusta utilizando a UT e o NSGA-II;
- proposta e validação de uma metodologia de projeto robusto e
- projeto robusto de um amplificador classe E.

1.3 APRESENTAÇÃO DO MANUSCRITO

Esta dissertação é composta de três capítulos mais a presente introdução e a conclusão.

No capítulo 2 é feita uma revisão bibliográfica sobre o tema de estudo. São apresentados os conceitos de otimização multiobjetivo, frente de Pareto e algoritmos evolucionários e é apresentado o algoritmo NSGA-II. Neste capítulo, são apresentados também os conceitos de otimização robusta e as principais abordagens presentes na literatura acerca deste tema.

Em seguida, o capítulo 3 descreve a metodologia empregada para otimização robusta. Os conceitos

da UT são apresentados para a formulação do algoritmo de otimização robusta. A UT é comparada com o método de Monte Carlo. É apresentada a metodologia de projeto robusto, sendo esta avaliada em um problema analítico.

Resultados experimentais são discutidos no capítulo 4, que apresenta uma introdução teórica sobre amplificadores de potência, com ênfase no amplificador classe E. Em seguida a metodologia de projeto robusto é aplicada no projeto robusto deste amplificador.

Por fim, o capítulo 5 apresenta as conclusões desta dissertação assim como as propostas de continuidade de estudo. O anexo contém material complementar relacionado ao *software* utilizado para simulação do amplificador classe E.

2 OTIMIZAÇÃO ROBUSTA

Alguns problemas de otimização têm como requisito que as soluções apresentadas sejam pouco sensíveis a incertezas associadas aos parâmetros constitutivos do projeto. Como exemplo, pode-se citar projetos de dispositivos que possuem incertezas associadas ao processo de fabricação. Nestes, devido às dificuldades e imprecisões na fabricação, o desempenho do dispositivo pode ser degradado ao ser implementado fisicamente. Neste caso, estas incertezas associadas aos parâmetros do projeto devem ser consideradas durante o processo de otimização.

Problemas de otimização podem ser resolvidos utilizando Algoritmos Evolucionários – AE. Algoritmos evolucionários são métodos estocásticos de busca baseados em conceitos da genética natural. Destaca-se, entre os AEs, o algoritmo genético NSGA-II [1], amplamente estudado na literatura.

Contudo, os AEs por si só não são capazes de tratar problemas que envolvem incertezas associadas aos parâmetros de projeto. Para tratar problemas desta categoria, é necessária uma avaliação da robustez das soluções encontradas para o problema. Ou seja, é necessário verificar se determinada solução (ou possível solução) mantém seu desempenho mesmo na presença de incertezas. Esta verificação da robustez das soluções é realizada através de métodos de análise de sensibilidade. Estes métodos consistem na verificação do comportamento de uma solução em sua vizinhança.

Objetivando soluções robustas, é necessário acrescentar conceitos de análise de sensibilidade no AE.

Este capítulo traz uma abordagem evolucionária para o problema da otimização robusta. A busca de soluções robustas é apresentada relacionando o tema de análise de sensibilidade e algoritmos evolucionários.

Este capítulo está organizado da seguinte forma:

- Otimização multiobjetivo: o estudo será iniciado pela formulação clássica do problema de otimização multiobjetivo (seção 2.1). Esta formulação clássica não considera as incertezas associadas aos parâmetros de projeto. Em seguida, serão apresentados os conceitos de algoritmos evolucionários (seção 2.2) e algoritmos genéticos (seção 2.2.1). Será apresentado o NSGA-II, algoritmo genético utilizado neste trabalho (seção 2.2.2).
- Otimização robusta: o estudo a otimização robusta (seção 2.3) iniciar-se-á pelo estudo da análise de sensibilidade (seção 2.3.1), que apresentará o método de Monte Carlo (seção 2.3.1.1) e um

método alternativo baseado nos resultados do AE (seção 2.3.1.2) para avaliação de sensibilidade das soluções. Em seguida será apresentada as justificativas e as vantagens de um projeto robusto (seção 2.3.2) e as suas duas abordagens: análise do pior caso (seção 2.3.3) e análise estatística (seção 2.3.4).

2.1 FORMULAÇÃO CLÁSSICA DO PROBLEMA DE OTIMIZAÇÃO MULTIOBJE-TIVO

Muitos dos problemas presentes na engenharia implicam a otimização de mais de um parâmetro, ou seja, há vários critérios a serem balanceados. Por exemplo, na síntese de um amplificador, deseja-se escolher uma arquitetura que resulte em alto ganho de potência, alta eficiência e baixo consumo. Desta forma, não é incomum encontrar dificuldades quando se tem diversos critérios que devem ser satisfeitos simultaneamente. Quando estes critérios são conflitantes, torna-se necessário encontrar uma solução que satisfaça às condições impostas em diferentes cenários. Problemas desta natureza são conhecidos como Problemas Multiobjetivo – MOP (do inglês *Multi-Objective Problems*), multicritérios ou vetores de otimização [8].

O problema de otimização multiobjetivo pode ser definido como:

$$\begin{aligned} \text{minimizar } & f_j(\mathbf{x}), & j = 1, \dots, m \\ \text{sujeito a } & g_i(\mathbf{x}) \leq 0, & i = 1, \dots, p \end{aligned} \quad (2.1)$$

em que

- $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ é o vetor de n variáveis a serem otimizadas (chamado também de vetor de parâmetros de projeto ou de variáveis de entrada). Estas variáveis x_i possuem um valor mínimo x_i^{min} e um valor máximo x_i^{max} associados, ou seja, $x_i^{min} \leq x_i \leq x_i^{max}$. Estes limites definem o espaço de variáveis de decisão ou o espaço de busca;
- $f_j(\mathbf{x})$ são as m funções a serem otimizadas (neste caso, minimizadas) denominadas funções objetivo ou, no estudo de algoritmos genéticos, funções de aptidão e
- $g_i(\mathbf{x})$ para $i = 1, \dots, p$ são as p restrições à que o problema está sujeito.

Problemas de otimização que envolvem múltiplas funções objetivo são, em alguns casos, resolvidos como problemas mono-objetivo a partir da agregação das diversas funções objetivo em uma só. Neste

caso, o problema resulta em uma solução única. Contudo, em problemas multiobjetivo predomina um conjunto de soluções ótimas, resultado de um compromisso dos requisitos determinados pelas funções objetivo. Assim, a medida em que existem várias funções objetivo a serem otimizadas simultaneamente, não se aplica mais o conceito de apenas uma solução ótima, e sim um conjunto de soluções possíveis com qualidade equivalente. Estas soluções são ótimas em um sentido amplo, onde nenhuma solução é superior em todos os objetivos considerados. Este conjunto de soluções ótimas é conhecido como soluções ótimas de Pareto e constituem a frente ótima de Pareto [9]. Ao se agregar as funções objetivo em uma única função objetivo, são necessárias diferentes ponderações para se obter um conjunto diversificado de soluções [1]. Algoritmos Evolucionários Multiobjetivo – MOEA (do inglês *Multiobjective Evolutionary Algorithms*) são algoritmos que possuem a capacidade de determinar as múltiplas soluções ótimas de Pareto em uma única simulação.

Em problemas com uma única função a otimizar, a escolha da melhor solução é feita com base na escolha do melhor valor escalar da função objetivo, dado que estes valores podem ser ordenados e comparados. Problemas multiobjetivo, nos quais deseja-se determinar a frente ótima de Pareto, requerem uma modificação na forma de ordenamento das soluções. Esta modificação refere-se ao conceito de dominância entre soluções. Em problemas multiobjetivo, duas soluções podem estar relacionadas de dois modos: ou uma solução domina a outra solução, ou nenhuma das duas soluções são dominadas, sendo chamadas de soluções não-dominadas.

O conceito de dominância pode ser enunciado como [10]:

Para problemas com mais de uma função objetivo, por exemplo f_j onde $j = 1, \dots, m$ (para $m > 1$), a solução \mathbf{x}^1 domina a solução \mathbf{x}^2 , se as duas condições são satisfeitas:

1. A solução \mathbf{x}^1 não é pior que \mathbf{x}^2 em todas as funções objetivo, ou $f_j(\mathbf{x}^1) \not\prec f_j(\mathbf{x}^2)$ para todas ($j = 1, \dots, m$) as funções objetivo ¹.
2. A solução \mathbf{x}^1 é estritamente melhor que a solução \mathbf{x}^2 em pelo menos uma função objetivo, ou $f_j(\mathbf{x}^1) \succ f_j(\mathbf{x}^2)$ para pelo menos um $j \in \{1, \dots, m\}$ (pelo menos em uma função objetivo).

Caso uma solução não domine outra solução e vice-versa, elas são ditas não-dominadas. O conjunto formado pelos indivíduos não-dominados constitui a frente de Pareto.

¹em que o operador \succ denota **é melhor** e $\not\prec$ denota **não é pior**.

A figura 2.1 ilustra o conceito apresentado de dominância, para o caso de uma minimização com duas funções objetivo. Em (a), a solução x_1 domina a solução x_2 (e a solução x_2 é dominada pela solução x_1), dado que apresenta valores menores nas duas funções objetivo. Em (b), as soluções x_1 e x_2 são ditas não-dominadas, pois cada uma apresenta um valor melhor em uma das funções objetivo.

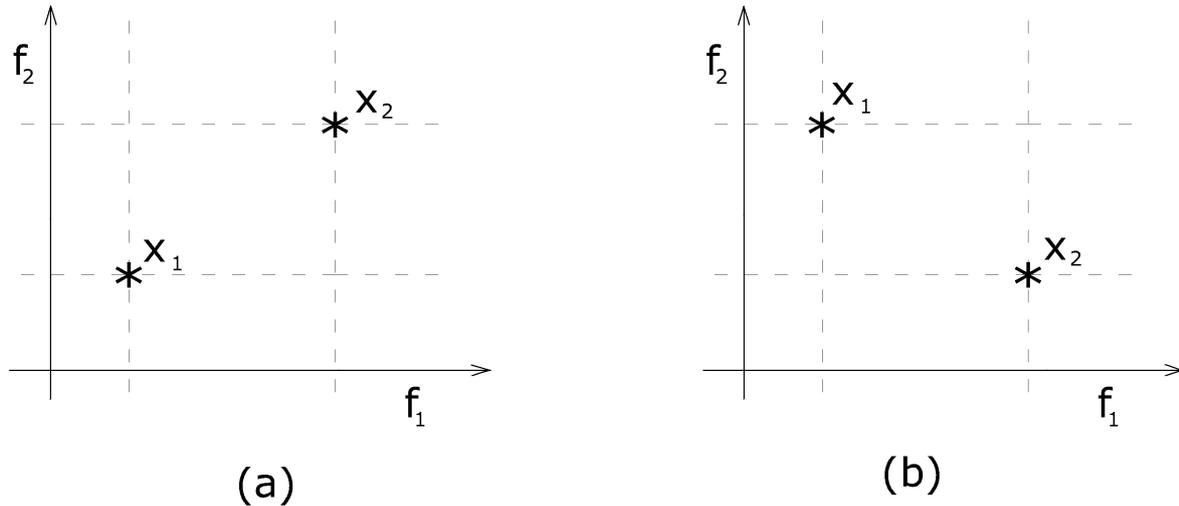


Figura 2.1: Ilustração do conceito de dominância.

Duas metas são importantes para guiar a busca da frente de ótima de Pareto em problemas multiobjetivo:

1. As soluções encontradas pelo algoritmo devem ser o mais próximas possível da frente ótima de Pareto. Esta meta é bastante intuitiva e comum aos diversos algoritmos de otimização, dado que soluções distantes da frente ótima de Pareto não são desejáveis.
2. As soluções encontradas na frente ótima de Pareto devem ter a maior diversidade possível. Esta meta, porém, é específica para problemas de otimização multiobjetivo, pois neste caso é necessário que as soluções estejam bem espalhadas dentro da frente, ou seja, que toda a frente possa ser coberta pelas soluções.

2.2 ALGORITMOS EVOLUCIONÁRIOS

Algoritmos evolucionários – AE – são algoritmos iterativos de busca e otimização baseados no processo biológico da evolução [11]. Seus principais conceitos são baseados na teoria de Darwin sobre a seleção natural. A teoria da seleção natural baseia-se no fato de que as características dos indivíduos que

são mais aptos tendem a permanecer nas gerações futuras, enquanto os indivíduos com características menos favoráveis tendem a não sobreviver, fazendo com que estas características menos favoráveis não permaneçam nas gerações futuras. Em resumo, pode-se dizer que os algoritmos evolucionários são uma abstração do processo de seleção natural.

Computação evolucionária é o ramo de pesquisa que engloba todos os algoritmos evolucionários. As três grandes vertentes estudadas pela computação evolucionária, e que apesar de algumas diferenças assemelham-se por empregar os conceitos da teoria da evolução de Darwin, são os algoritmos genéticos - AG, as estratégias evolucionárias e a programação evolucionária [11].

Os AEs possuem três características fundamentais [11]:

Indivíduos que se reproduzem com herança: os AEs trabalham com uma população de indivíduos a cada geração. Cada indivíduo representa uma possível solução para o problema que se está resolvendo. Estes indivíduos podem se reproduzir (de forma sexuada ou assexuada), gerando filhos que herdam algumas das características de seus pais. Estes traços de herança fazem com que os filhos apresentem certa semelhança com seus pais.

Variação genética: os filhos estão sujeitos à variação genética promovida pela mutação. A mutação permite que os filhos explorem outras características não herdadas pelos pais.

Seleção natural: a avaliação do desempenho do indivíduo no seu ambiente é a medida a partir de um valor de aptidão determinado para aquele indivíduo. A seleção para sobrevivência e reprodução é feita a partir da comparação destes valores de aptidão.

Para entender a ideia trazida pelos algoritmos evolucionários, torna-se interessante apresentar um algoritmo evolucionário padrão onde as características de reprodução com herança, variação genética e seleção natural são mostradas. O fluxograma mostrado na figura 2.2 retrata o algoritmo evolucionário padrão.

O algoritmo evolucionário padrão trabalha, a cada geração t , com uma população P de indivíduos de tamanho N :

$$P_t = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N\} \quad (2.2)$$

O indivíduo \mathbf{x}^l em que $l = 1, \dots, N$ é uma estrutura de dados que contém os valores das variáveis de entrada do problema. Cada indivíduo corresponde a uma amostra no espaço de busca do problema a ser resolvido. O espaço de busca compreende todo o conjunto de soluções possíveis do problema.

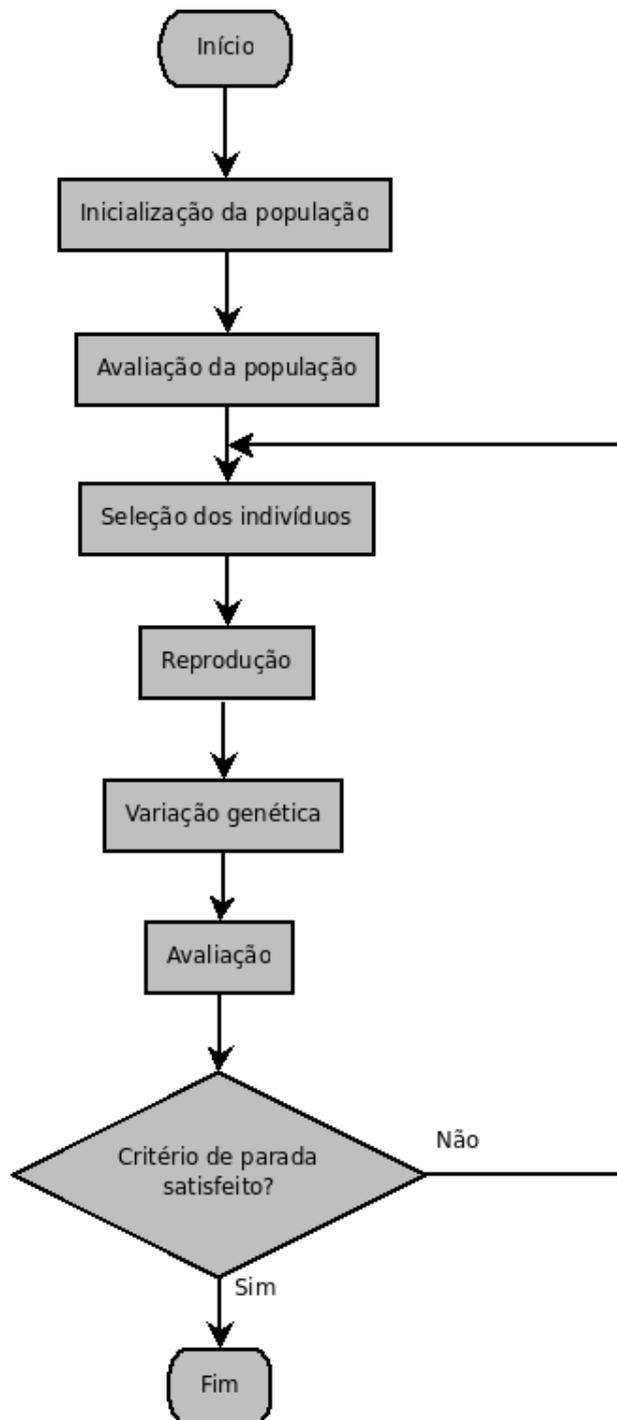


Figura 2.2: O algoritmo evolucionário padrão.

A fim de saber a aptidão de cada indivíduo ou seu nível de adaptação ao ambiente, estes são avaliados de acordo com o problema que se quer otimizar. Esta avaliação é feita através de funções objetivo. O objetivo do AE é a busca de soluções ou a otimização de determinado problema. Para isto, o problema deve ser definido e arranjado em forma de uma ou mais funções objetivo que permitam avaliar quantitativamente a aptidão de qualquer solução.

A população seguinte, a geração $t + 1$, é determinada a partir da seleção e reprodução dos indivíduos. Operações de mutação também podem incidir sobre os indivíduos. A realização destas três operações — seleção, reprodução e mutação — indicam a passagem por uma geração. Este processo se repete até que o critério de parada seja satisfeito. O critério de parada do algoritmo normalmente é identificado como um número máximo de gerações ou o alcance de algum objetivo específico.

A maioria dos algoritmos evolucionários pode ser implementada usando este algoritmo padrão, com algumas diferenças no que diz respeito às formas de representação, seleção, reprodução, entre outros processos que possam ser aplicados. Como citado anteriormente, o algoritmo genético é classificado como algoritmo evolucionário e será apresentado na próxima seção (seção 2.2.1).

2.2.1 Algoritmos Genéticos

Como classe dos algoritmos evolucionários, os algoritmos genéticos também são baseados da teoria de Darwin sobre a evolução. Em adição, os algoritmos genéticos usam vocabulário específico emprestado da genética natural [11]. Em AG, os indivíduos são denominados cromossomos e representam soluções em potencial para o problema. As três características fundamentais dos AEs estão presentes no AG sob o nome de operadores genéticos de seleção, cruzamento e mutação. A seguir cada um deles é detalhado.

Seleção: o objetivo da operação de seleção é direcionar a busca do AG dando preferência na seleção dos indivíduos mais aptos. Assim, os melhores indivíduos são selecionados para cruzamento, permitindo que as melhores características continuem nas gerações futuras. Um dos métodos tradicionais de seleção é o torneio binário. Neste método dois indivíduos são selecionados na população (podendo ser entre todos os indivíduos da população ou somente entre os não selecionados) e o indivíduo mais apto para cruzamento é escolhido. Outro procedimento é a seleção proporcional à aptidão, conhecido em inglês como *Roulette Wheel Selection*, no qual a probabilidade de um indivíduo ser selecionado é diretamente proporcional ao seu valor de aptidão [12].

Cruzamento: o objetivo desta operação de cruzamento (em inglês *crossover*) é a propagação das melhores características para as gerações futuras. Isto é feito a partir da troca de material genético entre os indivíduos, gerando novos indivíduos. Pode ser definida uma probabilidade de cruzamento, que indica a probabilidade de um cruzamento ocorrer. Normalmente esta probabilidade é definida em 100%. Um exemplo de cruzamento é o cruzamento binário com um ponto de troca, executado em cromossomos com codificação binária. Neste tipo de cruzamento, os cromossomos são cortados

em um determinado ponto escolhido aleatoriamente e os segmentos são trocados para se formar os novos cromossomos [12]. A figura 2.3 ilustra este processo.

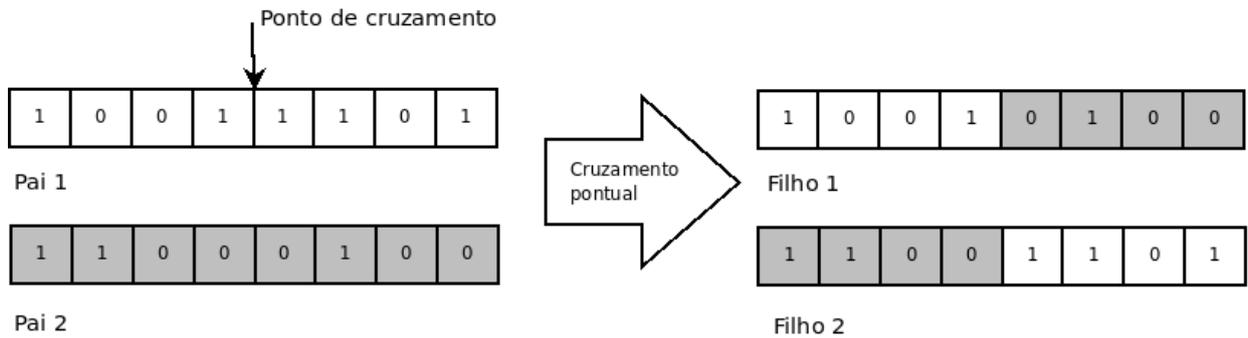


Figura 2.3: Ilustração de um cruzamento.

Mutação: o objetivo desta operação é manter a diversidade genética da população. A mutação é um processo onde um gene é aleatoriamente modificado para a formação de um novo cromossomo. A probabilidade de mutação é normalmente fixada como uma probabilidade baixa, assim como na genética natural, por isso esse operador é tido como secundário [12]. A figura 2.4 ilustra este processo para um cromossomo com codificação binária.

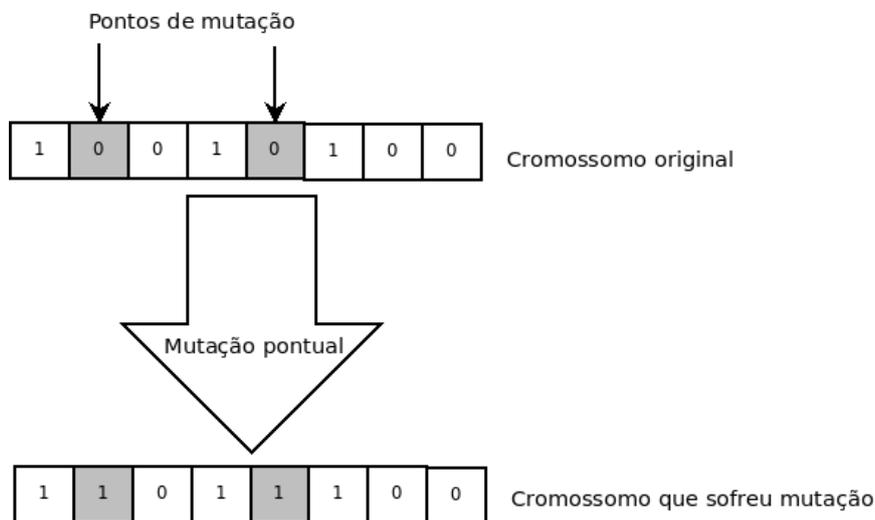


Figura 2.4: Ilustração de uma mutação.

2.2.2 NSGA-II

Diversos algoritmos genéticos têm sido desenvolvidos para solucionar problemas de otimização. Entre eles, pode-se citar o MOGA (do inglês *Multiple Objective Genetic Algorithm*) [13] e o NSGA-II (do

inglês *Nondominated Sorting Genetic Algorithm*) [1]. O NSGA-II tem se mostrado um algoritmo efetivo para busca de soluções em problemas multiobjetivo e por isso tem sido amplamente utilizado em diversas aplicações.

Duas características importantes do NSGA-II:

Elitismo O elitismo consiste na preservação dos melhores indivíduos de cada geração na próxima geração.

O elitismo permite que os melhores indivíduos sejam copiados para a próxima geração, prevenindo a perda de boas soluções uma vez que já foram descobertas. No NSGA-II, a população da geração $t + 1$ é escolhida entre os todos os indivíduos da população pai e todos os indivíduos da população filha, permitindo que caso existam na população pai indivíduos melhores que indivíduos gerados, os pais sejam escolhidos para permanecerem na próxima geração. Desta forma, é assegurado que não haverá decréscimo da melhor solução a cada geração.

Preservação da diversidade de soluções É desejável que o AG seja capaz de manter uma boa diversidade de soluções, como mencionado na seção 2.1. No NSGA-II a diversidade é preservada utilizando um operador denominado comparação de aglomeração (em inglês *crowded-comparison operator*) [1]. Este operador atua na diferenciação e ordenação dos indivíduos dentro de uma mesma frente de Pareto e na seleção de indivíduos para cruzamento.

Para se estimar a densidade de soluções em torno de determinada solução, é necessário calcular a distância de aglomeração (em inglês *crowding distance*). Esta distância é a estimativa do perímetro formado pelo hiper cubo cujos vértices são os vizinhos mais próximos. A figura 2.5 ilustra como calcular a distância de aglomeração para um problema com duas funções objetivo.

Este cálculo requer que a população seja ordenada de acordo com o valor de cada função objetivo em ordem ascendente de magnitude. Em seguida, para cada função objetivo, são designados valores infinitos de distância para as soluções extremas. Para cada uma das soluções intermediárias é designado um valor de distância determinado pelo módulo da diferença dos valores da função objetivo normalizada das duas soluções adjacentes àquela. O cálculo apresentado é então realizado para todas as funções objetivo. O valor final da distância de aglomeração é calculado como a soma dos valores de distância calculados para cada função objetivo. Para estas operações, é necessário que cada função objetivo seja normalizada antes da realização dos cálculos.

O cálculo da distância de aglomeração indica que soluções mais isoladas possuam uma distância de aglomeração maior. Da mesma forma, soluções com baixa distância de aglomeração estão mais rodeadas



Figura 2.5: Ilustração de como calcular a distância de aglomeração [1].

por outras soluções. Assim, na escolha das soluções que estão em uma mesma frente que passarão para a próxima geração são preferíveis as soluções com maior distância de aglomeração a fim de se manter a diversidade.

Definida a fórmula de cálculo da distância de aglomeração, pode-se definir o operador comparação de aglomeração. Este operador é utilizado na comparação de duas soluções, levando em consideração as frentes e as distâncias de aglomeração das soluções. Caso as duas soluções estejam em frentes diferentes, prefere-se a solução que está na melhor frente. Caso estejam na mesma frente, prefere-se aquela solução que possua maior distância de aglomeração, ou seja, a solução que esteja na região menos densa.

Assumindo que cada indivíduo i possui dois atributos:

- Valor de ordenação de acordo com o nível de não-dominância, ou seja, frente de não-dominância do indivíduo: i_{frente}
- Valor local da distância de aglomeração: $i_{\text{distância}}$.

O operador de comparação de aglomeração $i \geq_n j$ é definido como:

$$i \geq_n j \quad \text{se } (i_{\text{frente}} < j_{\text{frente}}) \quad \text{ou} \quad (i_{\text{frente}} = j_{\text{frente}}) \quad \text{e} \quad (i_{\text{distância}} > j_{\text{distância}}) \quad (2.3)$$

2.2.2.1 Operadores genéticos utilizados no NSGA-II

O NSGA-II pode trabalhar com indivíduos com codificação binária ou com codificação real. Nesta dissertação foi utilizada a codificação real para as variáveis do problema. Os operadores de seleção, cruzamento e mutação são implementados da seguinte forma pelo NSGA-II:

Seleção É utilizado o torneio binário com uma modificação a fim de incorporar o operador comparação de aglomeração [1]. Dois indivíduos são selecionados aleatoriamente entre todos os indivíduos da população e em seguida é aplicado o operador comparação de aglomeração (de acordo com a equação 2.3). O indivíduo mais apto é então selecionado.

Cruzamento É utilizado o cruzamento binário simulado – SBX (do inglês *simulated binary crossover*)[1]. O cruzamento SBX é detalhado em [14, 15]. Dado dois pais p^1 e p^2 , o cruzamento gera dois filhos q^1 e q^2 a partir da seguinte fórmula:

$$\begin{aligned} q^1 &= \frac{1}{2} ((1 + \beta)p^1 + (1 - \beta)p^2) \\ q^2 &= \frac{1}{2} ((1 - \beta)p^1 + (1 + \beta)p^2) \end{aligned} \quad (2.4)$$

em que β é um número aleatório dado pela seguinte distribuição:

$$\beta = \begin{cases} \frac{1}{2u\eta_c - 1} : & \text{se } u \leq 0,5 \\ \frac{1}{2(1-u)\eta_c - 1} : & \text{caso contrário} \end{cases} \quad (2.5)$$

em que u é um número aleatório entre 0 e 1 e η_c é um valor positivo que define quão próximo dos pais serão criados os novos indivíduos.

Mutação É utilizada a mutação polinomial [1]. Este tipo de mutação é detalhada em [15]. A mutação é realizada em cada parâmetro do indivíduo com uma probabilidade p_m . Dado uma variável v com valores máximos e mínimos v^{max} e v^{min} , a variável v^* criada a partir da mutação será dada por:

$$v^* = v + (v^{max} - v^{min})\tau \quad (2.6)$$

em que τ tem distribuição de densidade polinomial e é dado por:

$$\tau = \begin{cases} \frac{1}{2r\eta_m + 1} - 1 : & \text{se } r \leq 0,5 \\ 1 - [2(1-r)]\frac{1}{\eta_m + 1} : & \text{se } r \geq 0,5 \end{cases} \quad (2.7)$$

em que r é um número aleatório e η_m é o índice de distribuição de mutação.

2.2.2.2 Fluxograma do NSGA-II

Apresentada estas características do NSGA-II, pode-se apresentar o algoritmo do NSGA-II. A figura 2.6 apresenta o fluxograma do NSGA-II.

Inicialmente é criada uma população aleatória de indivíduos P_0 de tamanho N . A população P_0 é ordenada de acordo com os critérios de não-dominância e o operador de comparação de aglomeração é aplicado na população, definindo assim um valor de aptidão para cada indivíduo.

Em seguida, inicia-se o laço de simulação, que contará as gerações t e que terminará com um critério de parada definido pelo usuário. São realizadas as operações de seleção, cruzamento e mutação para criação de uma população filha Q_0 também de tamanho N . Para a introdução do elitismo, as populações pai P_t e filha Q_t são combinadas, formando uma população R_t de tamanho $2N$:

$$R_t = P_t \cup Q_t. \quad (2.8)$$

Esta população é ordenada de acordo com o critério de não-dominância. Para este procedimento, o NSGA-II utiliza uma função específica denominada *fast-nondominated-sort*, que possibilita uma rápida ordenação envolvendo menos operações [1]. Em seguida é feita a escolha dos N indivíduos que formarão a nova população pai. Esta escolha é feita utilizando o operador comparação de aglomeração. O laço se repete até que o critério de parada seja atendido. Nesta implementação do NSGA-II foi escolhido o número máximo de gerações como critério de parada.

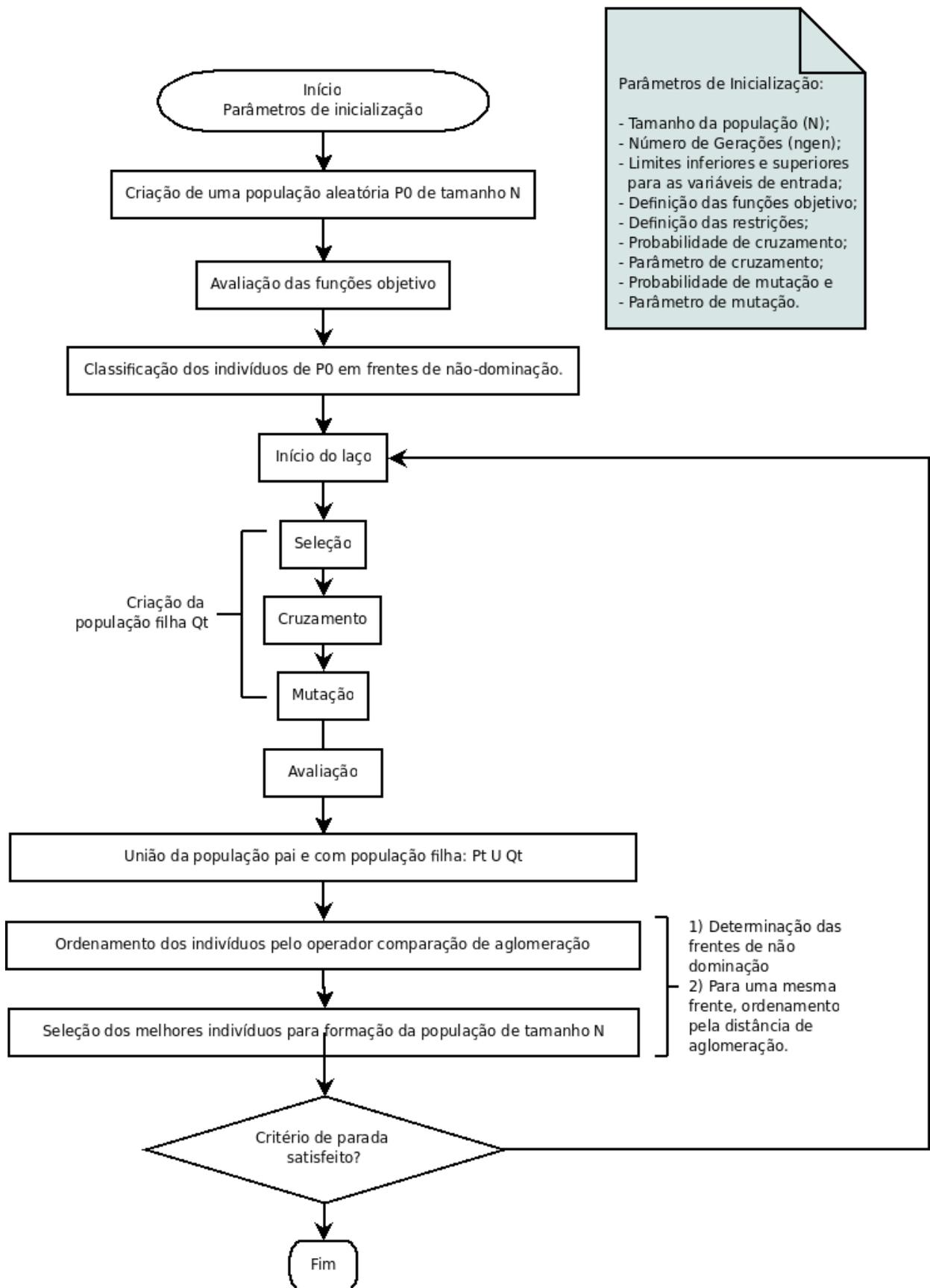


Figura 2.6: Fluxograma do NSGA-II.

2.3 OTIMIZAÇÃO ROBUSTA

No projeto de dispositivos é necessária a permissão de tolerâncias de diversas naturezas. Por exemplo, no processo de fabricação de um dispositivo pode não ser possível encontrar os elementos exatos descritos no projeto ou os elementos não são sempre fabricados com a mesma precisão. Mais ainda, sabe-se que quanto maior for a tolerância no processo de fabricação, mais baratas serão as máquinas e dispositivos necessários para a fabricação [16].

Desta forma, o projetista deve avaliar se no cenário do projeto estão presentes incertezas. As incertezas a que estão sujeitas os dispositivos decorrem das mais diversas origens sendo possível distingui-las em dois grupos [17]:

Incertezas associadas às condições externas. Neste grupo estão as incertezas devido às condições do ambiente de operação, tais como temperatura, umidade, pressão, e também incertezas devido a cargas impostas ao sistema.

Incertezas associadas às variáveis de entrada. Estas incertezas possuem diversas origens, tais como: tolerâncias mecânicas, propriedades do material (condutividade, resistência, por exemplo) e imperfeições no processo de fabricação (os componentes podem não ser fabricados com o valor exato especificado no projeto).

A incerteza associada a um estado estimado pode ser geralmente representada por uma distribuição de probabilidade incorporada à informação sobre seu estado [3]. Comumente, a modelagem da incerteza é realizada pelo uso de variáveis aleatórias, na qual a incerteza pode ser representada por uma ou várias variáveis aleatórias, correlacionadas ou não, podendo mesmo ser de diferentes distribuições de probabilidade.

No exemplo citado acima, para a fabricação de um dispositivo é necessário determinado componente que não é obtido de forma precisa. Analisando as diversas amostras deste componente, pode-se avaliar a aleatoriedade destas amostras, ou sejam como elas se comportam, se é de forma gaussiana ou uniforme, por exemplo. Assim, a incerteza associada àquele componente é determinada.

Devido as incertezas associadas aos componentes, o desempenho do dispositivo pode sofrer sérias degradações quando comparado aos valores especificados em projeto. Neste sentido, é necessário que o projetista esteja preocupado no desempenho do seu produto mesmo em um cenário de incertezas. Quando se trata de problemas de otimização em cenários de incertezas, é necessário a busca de soluções robustas.

Pode-se definir soluções robustas como soluções para as quais os objetivos (funções objetivo que se deseja otimizar) não sofrem variações significativas quando há variação (incerteza) nos parâmetros de projeto [4]. Elas devem ser tão insensíveis quanto for possível a perturbações, incertezas e variações nos parâmetros de projeto.

A principal vantagem do uso de soluções robustas está no fato de que existe uma garantia que o projeto apresentará o desempenho desejado na presença de incertezas (desde que o nível de incertezas seja especificado no projeto). Ainda que o desempenho das soluções robustas possa ser inferior ao desempenho conseguido com as soluções não-robustas, o projetista saberá que o desempenho do seu projeto ficará dentro dos limites aceitáveis, sem surpresas quando da implementação, desde que o modelo seja adequado (represente adequadamente a realizada).

Uma das formas de se verificar a robustez de uma solução é realizando uma análise de sensibilidade. Esta análise consiste na verificação do comportamento desta solução em sua vizinhança. Esta análise pode ser realizada *a posteriori*, ou seja, após a obtenção das soluções é verificada se a solução especificada é robusta na presença de incertezas. A análise de sensibilidade *a posteriori* se utiliza frequentemente do método de Monte Carlo.

Uma outra possibilidade é a realização de um projeto robusto. Neste caso, considera-se já na fase de projeto do dispositivo as incertezas associadas a seus elementos constitutivos, a fim de se fazer uma avaliação do impacto destas variações no desempenho da solução. A inserção direta das medidas de robustez na formulação do problema é essencial pois soluções ótimas robustas podem não coincidir com soluções ótimas não-robustas [18]. Para a realização de um projeto robusto é necessário que o problema de otimização formulado matematicamente pela equação 2.1 seja modificado a fim de considerar a robustez.

A seguir será discutida a análise de sensibilidade (seção 2.3.1). Em seguida será abordado o projeto robusto e suas principais vertentes (seção 2.3.2 à seção 2.3.4).

2.3.1 Análise de Sensibilidade

A análise de sensibilidade permite verificar como determinada função varia em relação a deslocamentos em torno de determinado ponto. É intuitivo pensar que nada melhor que a própria verificação da função para realizar esta análise de sensibilidade. Contudo, a análise da função pode ser muito custosa, pois muitas vezes os resultados destas funções são determinados através de métodos numéricos complexos, ao invés de um simples cálculo analítico. Neste sentido, procuram-se métodos capazes de realizar esta análise com um menor número de avaliações da função.

Uma das aplicações da análise de sensibilidade é seu uso *a posteriori* permitindo a identificação de soluções robustas dentro do conjunto de soluções da frente ótima de Pareto [19]. Assim, dado um nível de tolerância especificado, as soluções obtidas são analisadas para ver se atendem ao limite estabelecido. Neste caso a análise *a posteriori* consiste na realização do método de Monte Carlo em cada uma das soluções da frente ótima de Pareto obtida pelo AE resultando na escolha que apresente menor variação das funções objetivo ou que atenda ao critério de tolerância estabelecido.

2.3.1.1 O método de Monte Carlo

O método de Monte Carlo é um método estatístico que consiste na geração de amostras segundo uma distribuição para servir de parâmetro de entrada na simulação de um processo. Após a realização das diversas simulações, um procedimento de pós-processamento avalia os resultados, gerando as medidas estatísticas desejadas (valor esperado e variância, por exemplo). Este procedimento é mostrado na figura 2.7.

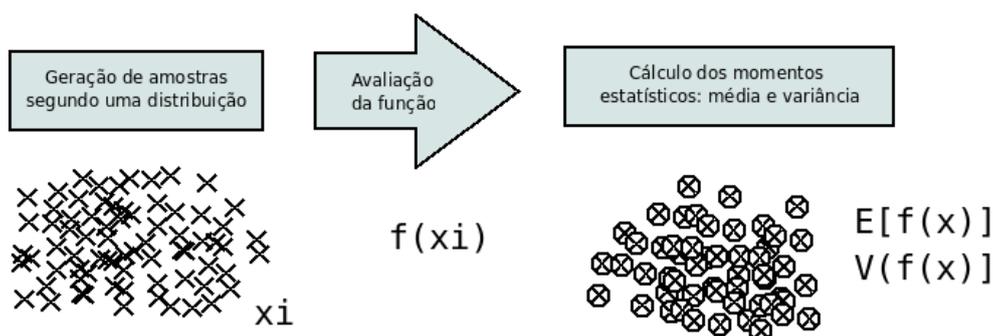


Figura 2.7: Fluxograma do método de Monte Carlo.

Uma das desvantagens do método de Monte Carlo é que este utiliza-se de milhares de simulações, podendo ser bastante oneroso dependendo do tipo do problema em análise. Como os problemas solucionados pelos AEs são complexos e o cálculo das funções objetivo podem ser custosos, esta técnica torna-se impraticável para alguns problemas, sendo necessário outros métodos para tal análise.

Devido a esta desvantagem do método de Monte Carlo, procuram-se outras formas de cálculo das medidas estatísticas de um problema que sejam menos onerosas, a fim de se realizar a análise de sensibilidade.

2.3.1.2 Alternativa ao Método de Monte Carlo

Objetivando a redução do esforço computacional, uma solução alternativa ao método de Monte Carlo advém do uso das amostras do AE. Neste método, a análise de sensibilidade é obtida usando as avaliações das funções objetivo realizadas pelo AE como amostras do comportamento da função nas vizinhanças da solução ótima [20]. A principal ideia por trás deste método está no fato de que métodos de otimização estocásticos já realizam inúmeras avaliações das funções objetivo na busca da solução ótima. Mais especificamente, AEs naturalmente possuem esta capacidade de varrer o espaço das variáveis na busca da solução. As amostras geradas pelo AE são então utilizadas para verificar o comportamento do problema em torno da solução ótima, não onerando o projeto com mais avaliações das funções objetivo.

A figura 2.8 mostra como um AE percorre todo o espaço de busca. Para esta simulação foi utilizado NSGA-II com uma população de 96 indivíduos e 100 gerações. A função utilizada pode ser descrita como [1]:

$$\begin{aligned} f_1(x) &= 1 - \exp\left(-\sum_{i=1}^n (x_i - 1/\sqrt{n})^2\right) \\ f_2(x) &= 1 - \exp\left(-\sum_{i=1}^n (x_i + 1/\sqrt{n})^2\right) \end{aligned} \quad (2.9)$$

em que $x_i \in [-4, 4]$. A frente ótima de Pareto é dada por $x_i \in [-1/\sqrt{3}, 1/\sqrt{3}]$.

A frente ótima obtida com o uso do NSGA-II, para $n = 3$ é mostrada em vermelho nas figuras 2.8 e 2.9.

Analisando as figuras 2.8 e 2.9, em que em azul são mostradas as amostras geradas pelo AE, é possível visualizar que o AE explora, ou seja, gera amostras no espaço de busca, e mais fortemente, as amostras estão mais concentradas em torno da frente ótima de Pareto (região com maior densidade de amostras, mostrada em vermelho). Desta forma, é possível realizar a análise de sensibilidade das soluções que estão na frente de Pareto, pois há grande concentração de amostras do AE nesta região.

Em [21], é proposta uma fórmula para avaliar a sensibilidade das soluções, baseado no uso das amostras já geradas pelo AE. Para comparar quão estável é uma solução de Pareto em relação às outras, é realizada uma associação do valor da função objetivo com a distância no espaço de variáveis. Desta forma, o comportamento de uma solução \mathbf{x}^* para cada função objetivo f_j dentro de um domínio de perturbação (ou domínio de incerteza) D^* é dado pela seguinte equação:

$$B_j(\mathbf{x}^*) = \max_{\mathbf{x}_s, \mathbf{x}_q \in D^*, s \neq q} \frac{|f_j(\mathbf{x}_s) - f_j(\mathbf{x}_q)|}{\|\mathbf{x}_s - \mathbf{x}_q\|} \quad (2.10)$$

em que \mathbf{x}_s e \mathbf{x}_q são amostras dentro de D^* . O domínio D^* é qualquer domínio que contenha \mathbf{x}^* e pode ser definido de diversas formas (o domínio de incerteza será melhor detalhado na seção 2.3.3).

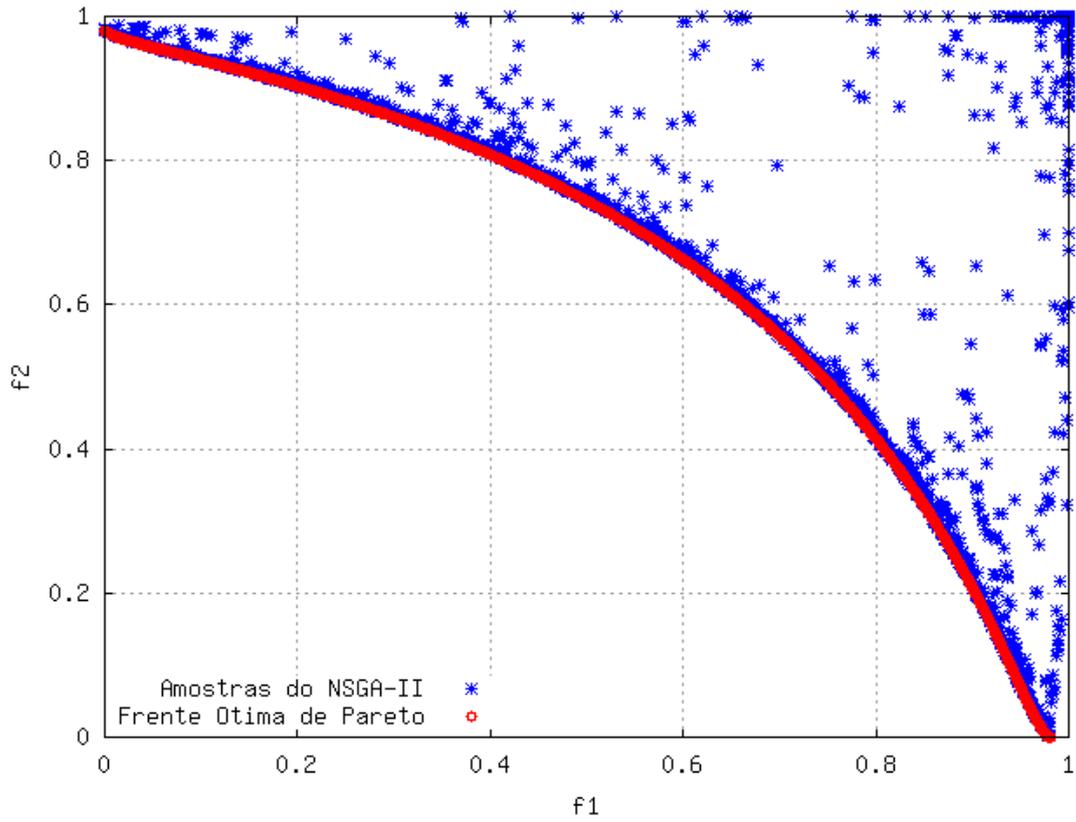


Figura 2.8: Visualização da distribuição das amostras do AE.

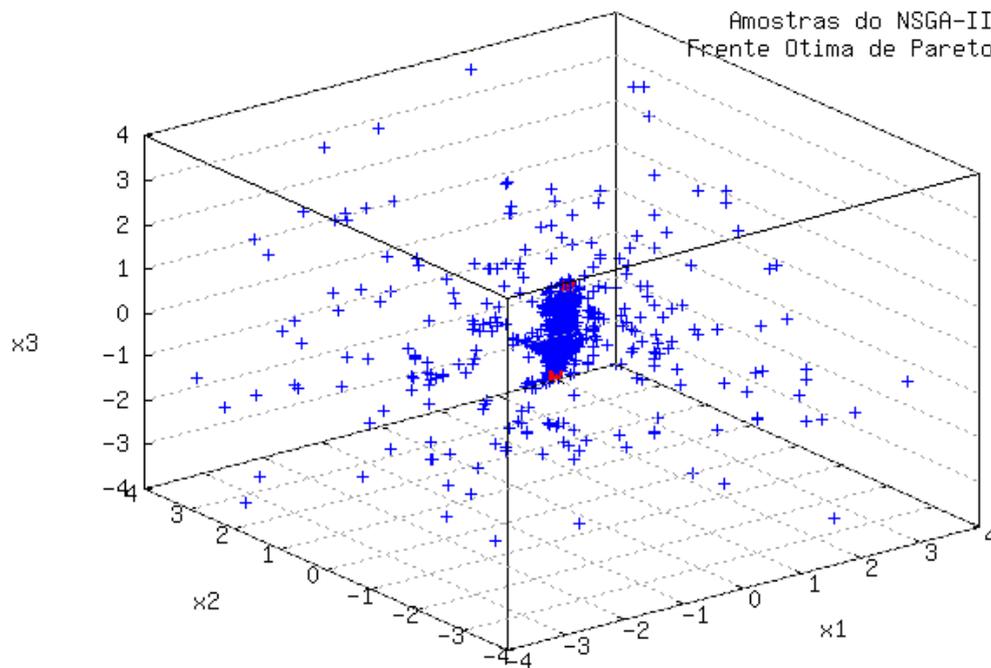


Figura 2.9: Visualização da distribuição das amostras do AE, no domínio das variáveis de entrada.

Em uma análise direta da equação 2.10, percebe-se que quanto maior B_j , mais instável é a função f_j na solução \mathbf{x}^* . Quanto maior o número de amostras dentro do domínio D^* , mais precisa será esta métrica.

2.3.2 Projeto Robusto

Em muitos problemas de otimização, uma análise *a posteriori* pode não ser suficiente para garantir que o projeto possa ser mesmo realizável em um ambiente de incertezas. Dois problemas podem ser levantados na abordagem *a posteriori*:

- as soluções ótimas obtidas podem não coincidir com as soluções ótimas robustas ou
- as soluções ótimas podem ser impraticáveis na presença de incertezas.

Soluções ótimas robustas podem não coincidir com soluções ótimas não-robustas. Soluções ótimas robustas podem ter um desempenho inferior ao de soluções ótimas não-robustas, contudo garantem que os objetivos serão alcançados independentemente das variações nos parâmetros de projeto. A figura 2.10 ilustra esta situação. A função $f(\mathbf{x})$ possui dois mínimos locais. Pode ser observada que a solução ótima, representada por \mathbf{x}^* apresenta o melhor desempenho de $f(\mathbf{x})$ no espaço de busca de soluções. Contudo, ao considerar uma incerteza Δ associada a variável \mathbf{x} verifica-se uma degradação no desempenho da função $f(\mathbf{x})$. Já a solução \mathbf{x}_r^* apresenta um desempenho pior que a solução \mathbf{x}^* , porém é menos afetada a variações. Uma variação Δ na solução \mathbf{x}_r^* causa menos impacto no desempenho $f(\mathbf{x})$.

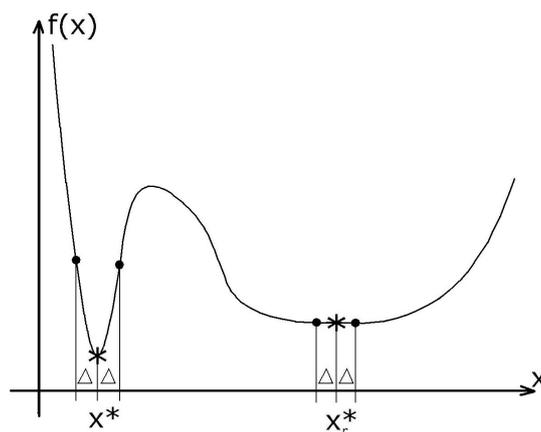


Figura 2.10: Comparação entre uma solução ótima robusta e uma solução ótima não-robusta [2].

Diferenças entre soluções robustas e não-robustas também podem ser verificadas quando se tem problemas com restrições. Em problemas de otimização robusta, as soluções encontradas devem continuar

válidas frente às restrições determinadas para o problema. Esta situação está ilustrada na figura 2.11. Nesta figura, a área hachurada apresenta as soluções inviáveis (ou infactíveis, irrealizáveis) para o problema (soluções onde há restrições de implementação). Suponha que a solução não-robusta encontrada esteja localizada na intersecção das duas restrições (solução x^*). Qualquer variação nesta solução pode tornar esta solução irrealizável. A solução robusta deve então ser tolerante a este tipo de restrição. O processo de otimização deve então considerar a robustez também para as restrições.

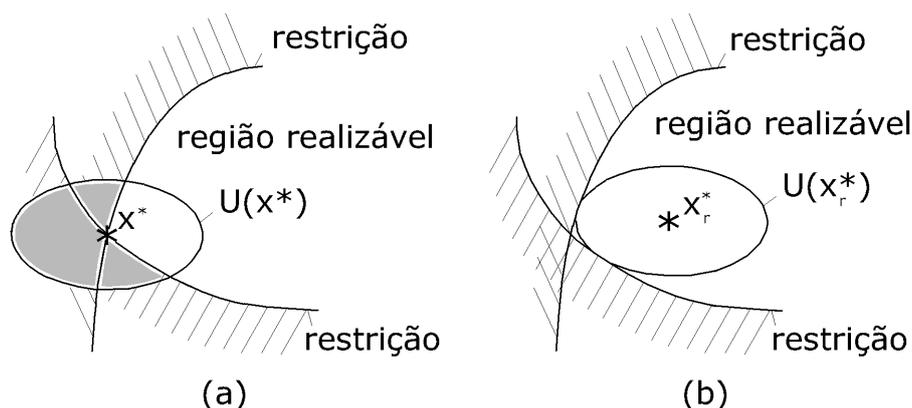


Figura 2.11: Comparação entre uma solução ótima robusta e uma solução ótima não-robusta [2].

Assim, pode-se definir uma solução robusta como aquela que possui duas características [22]:

- é pouco sensível a variações nos parâmetros nominais de projeto e
- permanece realizável para um certo nível de incerteza nos parâmetros nominais de projeto.

Apresentado o conceito de solução robusta e a problemática de apenas uma análise *a posteriori* nas soluções dadas pelo AEs, percebesse a necessidade do algoritmo de otimização ser um algoritmo que trate o problema da robustez. Faz-se necessário guiar a busca de soluções através de soluções robustas.

Diversas são as abordagens encontradas na literatura para tratar o problema da robustez. Pode-se dividir estas abordagens em dois grandes grupos:

Análise do pior caso (em inglês *Worst Case Design*) [2, 23]: consiste em otimizar o problema a partir da pior solução encontrada para uma dada faixa de incerteza.

Análise Estatística [22, 24]: consiste na aplicação de métodos estatísticos para calcular a robustez de determinada solução.

Estas abordagens serão descritas nas seções 2.3.3 e 2.3.4.

2.3.3 Análise do Pior Caso

A análise do pior caso consiste na verificação da pior solução dado um nível de incerteza, a fim de se prosseguir com a otimização. Esta abordagem é tida como conservadora no caso de múltiplas variáveis de entrada, pois é improvável que todas as variáveis do projeto se encontrem em uma situação de pior caso [24, 25].

A análise do pior caso considera um conjunto de incertezas para cada variável de entrada do projeto. Um conjunto de incertezas $U(\mathbf{x})$ é definido como um sub-domínio do espaço de busca que define as possíveis variações do parâmetro \mathbf{x} [2].

Diversas são as maneiras de se definir este conjunto de incertezas. Caso as incertezas associadas às variáveis de projeto sejam independentes e igualmente distribuídas, estas podem ser modeladas na forma de um hiper-retângulo, como na equação 2.11 [2]:

$$U(\mathbf{x}) = \{\zeta \in \mathbb{R}^n : \mathbf{x} - \Delta \leq \zeta \leq \mathbf{x} + \Delta\} \quad (2.11)$$

em que $\Delta = (\Delta_1 \Delta_2 \dots \Delta_n)^T \in \mathbb{R}^n$ é o vetor que representa as maiores variações possíveis dos valores nominais das variáveis do projeto.

Outra maneira é a modelagem da incerteza como um elipsóide, justificável quando se trata de variáveis gaussianas descorrelatadas [2]. A figura 2.12 ilustra, para um espaço de duas dimensões, o conjunto de incertezas para o caso retangular e elipsoidal.

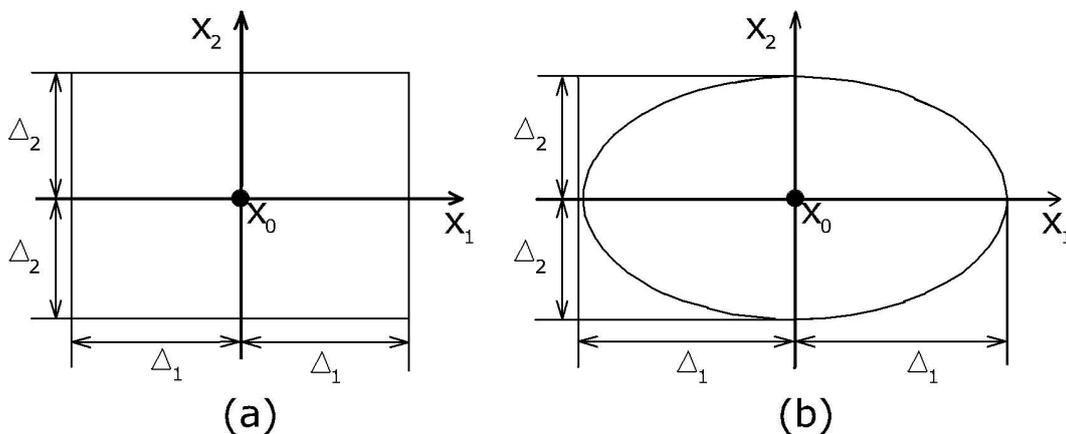


Figura 2.12: Diferentes possibilidades para definição de um conjunto de incertezas [2]: (a) retangular e (b) elipsoidal.

Uma vez modelado o conjunto de incertezas, pode-se definir o problema da robustez como [2]:

$$\begin{aligned} \text{minimizar}_x \quad & \max_{\zeta \in U(\mathbf{x})} \{f_j(\zeta)\}, \quad j = 1, \dots, m \\ \text{sujeito a} \quad & \max_{\zeta \in U(\mathbf{x})} g_k(\zeta) \leq 0, \quad k = 1, \dots, p \end{aligned} \quad (2.12)$$

Para se resolver o problema proposto pela equação 2.12, é necessário avaliar o pior caso dentro do conjunto de incerteza. Contudo, para não acarretar em grande esforço computacional, os parâmetros que levam ao pior caso devem ser identificados rapidamente. Vários métodos foram propostos para identificar este pior caso: o uso das amostras do AG (citado na seção 2.3.1.2) e o método de discretização dos vértices (em inglês *Vertex Discretization*) [2].

O primeiro método citado consiste na verificação da pior amostra já gerada pelo AE dentro do conjunto de incertezas $U(\mathbf{x})$. Caso não tenha nenhuma amostra já computada pelo AE dentro de $U(\mathbf{x})$, é gerada uma nova amostra e esta amostra é reservada para entrar em futuros cálculos. A otimização é feita com base na pior amostra dentro do conjunto de incertezas $U(\mathbf{x})$.

A seguir o método de discretização dos vértices é apresentado.

2.3.3.1 Discretização dos Vértices

Aproveitando a geometria de um conjunto de incertezas retangular, pode-se usar apenas os valores dos vértices para definir o pior caso. Esta abordagem assume que o pior valor encontra-se em um dos vértices do conjunto de incertezas, e se a função é monotônica em U , isto leva a resultados exatos. Para o conjunto de incertezas definido pela equação 2.11, o conjunto U_d de vértices é:

$$U_d(\mathbf{x}) = \{\zeta \in \mathbb{R}^n : \zeta_i = x_i + \Delta_i \vee \zeta_i = x_i - \Delta_i\} \quad (2.13)$$

Neste caso, a função é calculada para todos os pontos em U_d , e o maior valor obtido é definido como o pior caso no conjunto de incertezas U . Contudo, para problemas com muitas variáveis, este torna-se computacionalmente complexo, dado que o número de vértices cresce exponencialmente com o número de variáveis (dimensões) do problema. Em [2] é proposto um método denominado predição do pior vértice (em inglês *Worst Vertex Prediction*) para reduzir este número de avaliações.

Como dito anteriormente, para este método é assumido que a função em análise é monotônica, ou seja, assume uma convexidade local no espaço de busca, o que muitas vezes não é verificado nos problemas reais de otimização [24].

2.3.4 Análise Estatística

Os métodos de otimização robusta desta categoria utilizam medidas estatísticas, como o valor médio $\mathbf{E}[\mathbf{F}(x)]$ e a variância $\sigma_{\mathbf{F}(x)}^2$ (ou desvio padrão $\sigma_{\mathbf{F}(x)}$) na busca de soluções robustas.

A formulação do problema robusto através de uma análise estatística pode ser feita de diversas maneiras, baseadas em um arranjo entre valor médio e variância (ou desvio padrão) [22]:

- Minimização do valor médio das funções objetivo e considerando a variância abaixo de um limiar como uma restrição do problema:

$$\begin{aligned} &\text{minimizar } \mathbf{E}[\mathbf{F}(x)] \\ &\text{sujeito a } \sigma_{\mathbf{F}(x)}^2 \leq \sigma^2 \end{aligned} \quad (2.14)$$

em que σ é o nível de tolerância (desvio padrão aceitável dos resultados) determinado pelo projetista, que representa para cada função objetivo qual a maior variação pode ser admitida.

- Minimização de uma combinação da média e do desvio padrão:

$$\min \mathbf{E}[\mathbf{F}(x)] + (1 - \lambda)\sigma_{\mathbf{F}(x)} \quad (2.15)$$

em que $\lambda \in [0, 1]$ define grau de robustez desejado. Um valor baixo de λ leva a soluções mais robustas. Para $\lambda = 1$ é considerado apenas o valor médio das funções objetivo.

- Minimização multiobjetivo, a partir da otimização do desempenho médio e da minimização da variação de desempenho:

$$\text{minimizar } \begin{bmatrix} \mathbf{E}[\mathbf{F}(x)] \\ \sigma_{\mathbf{F}(x)}^2 \end{bmatrix}. \quad (2.16)$$

Na verdade, pode-se considerar as equações 2.14 e 2.15 como soluções particulares de 2.16.

Outra forma de se abordar o problema de otimização robusta tem sua origem na abordagem de Qualidade de Tagushi [24]. Segundo Tagushi, a qualidade é medida pelo desvio que uma característica apresenta em relação ao valor esperado da mesma. Ainda, vários fatores denominados por Tagushi como fatores de ruído causam prejuízo ao desempenho de determinadas características. Este prejuízo pode então ser avaliado através de uma função de perda de qualidade – QF (do inglês *quality loss function*):

$$\text{QF}(x) = k_1 \langle \text{OF}(\mathbf{x}) \rangle^2 + k_2 \sigma^2 \quad (2.17)$$

em que OF é a função objetivo original, $\langle \rangle$ é o operador média, σ^2 é a variância das funções objetivo e k_1, k_2 são ponderações definidas pelo usuário.

Para todas estas abordagens, são necessários os cálculos das medidas estatísticas de média e variância. Diversas são as formas de se obter estas medidas. A mais comum é o método de Monte Carlo, já citado na seção 2.3.1.1. Outras abordagens levam a cálculos indiretos dessas medidas, como a opção mostrada na seção 2.3.1.2 que consiste no uso das amostras do AE [20].

A seguir serão apresentadas mais duas propostas presentes na literatura: na primeira, é sugerida uma modificação na função objetivo a fim de considerar a incerteza, enquanto que na segunda utiliza-se a abordagem por séries de Taylor.

2.3.4.1 Consideração da incerteza antes da avaliação da função objetivo

Uma amostragem estatística do espaço de busca das variáveis a partir da inserção do ruído antes da avaliação da função objetivo pode ser realizada objetivando a busca de soluções robustas. Neste caso, incertezas são introduzidas nas variáveis de entrada de modo que a avaliação das funções objetivo as considere [4, 26, 27].

Assim, em vez da avaliação de $f(x)$, é realizada a avaliação de $f(x + \delta)$, onde δ representa o valor da incerteza, de distribuição uniforme ou normal por exemplo, gerado antes de cada avaliação da função objetivo.

A seguir, é apresentado o fundamento matemático para esta abordagem.

Seja a média de uma função objetivo:

$$\overline{f(t)} = \sum_{i=1}^N \frac{f(x^i)}{N} = \int f(x)p(x, t)dx \quad (2.18)$$

em que $p(x, t)$ é a distribuição de probabilidade de um indivíduo.

No caso de uma otimização robusta, a média da função objetivo pode ser representada por:

$$\begin{aligned} \overline{f'(t)} &= \sum_{i=1}^N \frac{f(x^i + \delta^i)}{N} \\ &= \int_x dx \int_{\delta} f(x + \delta)p(x, t)q'(\delta)d\delta \\ &= \int_x \left[\int_{\delta} f(x + \delta)q'(\delta)d\delta \right] p(x, t)dx \\ &= \int F(x)p(x, t)dx \end{aligned} \quad (2.19)$$

em que δ é o vetor de ruído e $q'(\delta)$ é a distribuição de δ .

As equações 2.18 e 2.19 mostram que $F(x)$ é equivalente a $f(x)$ na presença de incertezas. De fato,

pode-se dizer que $F(x)$ representa o valor esperado de $f(x + \delta)$. Em um AE, pode-se utilizar $f(x + \delta)$ como função objetivo, visando a obtenção de soluções robustas.

Esta abordagem permite a análise estatística sem o cálculo direto de medidas estatísticas, pois o espaço está sendo amostrado de forma aleatória dentro do domínio de incerteza.

2.3.4.2 Expansão em série de Taylor

Em uma outra abordagem, as medidas estatísticas são determinadas através da expansão da série de Taylor é proposta em [16]. Rejeitando os termos de ordem maior, a média e o desvio padrão podem ser aproximadas usando a expansão de Taylor:

$$E[F(x)] = F(\bar{x})$$

$$\sigma_F = \sqrt{\sum_{i=1}^n \left(\frac{\partial F}{\partial x_i} \right)^2_{\bar{x}} \sigma_{x_i}^2} \quad (2.20)$$

em que n é o número total de variáveis de entrada, \bar{x} é o vetor de média das variáveis de entrada e $\sigma_{x_i}^2$ é a variância a que está sujeita a variável de entrada.

O uso da expansão de Taylor não pode ser aplicado em todos os problemas pois implica no cálculo do gradiente da função. A combinação desta abordagem com AE não é bem vinda justamente pelo fato de que os AEs trabalham normalmente em problemas onde o cálculo do gradiente não é factível.

2.3.5 Considerações finais

Objetivando soluções robustas, é necessário que o processo de otimização seja guiado através destas soluções. Ou seja, é necessária uma modificação na formulação do problema para que seja considerada as medidas estatísticas de média e variância das soluções. A grande dificuldade reside na obtenção destas medidas estatísticas do problema em estudo. O método de Monte Carlo exige um número muito grande de avaliações do problema. Neste capítulo foram apresentados alguns artifícios para contornar este problema. No próximo capítulo será apresentada uma nova abordagem para viabilizar a otimização robusta.

3 PROPOSTA DE UMA METODOLOGIA DE OTIMIZAÇÃO EVOLUCIONÁRIA ROBUSTA

Na busca de soluções robustas para um projeto de otimização, percebe-se a necessidade de avaliação da sensibilidade das soluções. O método de Monte Carlo é frequentemente utilizado para esta avaliação. Contudo é necessário um elevado número de simulações para se alcançar o resultado desejado. Outros métodos apresentados no capítulo 2 propunham alternativas para o projeto robusto, como o uso das amostras geradas pelo AE e a avaliação dos vértices de um domínio de incertezas.

Nesta dissertação, é proposto o uso da transformada *Unscented* – UT (do inglês *Unscented Transform*) como método para cálculo das medidas estatísticas necessárias para a análise de sensibilidade e otimização robusta. A UT é capaz de realizar uma aproximação discreta de uma função densidade de probabilidade contínua, tornando possível a estimativa das medidas estatísticas como a média e a variância.

Com os resultados do valor médio e da variância provenientes do cálculo da UT, pode-se utilizar os AEs na busca de soluções para um projeto robusto, implementando um algoritmo de otimização robusta. Este algoritmo de otimização robusta será feito integrando o NGS-II e a UT. Este assunto será explicado neste capítulo, iniciando pelo embasamento matemático da UT e sua comparação com o método de Monte Carlo. Em seguida, a metodologia para projeto robusto será apresentada (incluindo o algoritmo de otimização robusta) e os resultados da otimização de funções analíticas serão analisados, objetivando validar o algoritmo e a metodologia proposta.

3.1 A TRANSFORMADA UNSCENTED

A transformada *Unscented* é um método para calcular as estatísticas de uma variável aleatória que sofre uma transformação não-linear. Foi desenvolvida por Julier e Uhlman em 1997 [3]. O princípio básico da UT é a aproximação de um mapeamento não-linear por um conjunto de pontos selecionados, chamados de pontos sigma (em inglês *sigma points*). As medidas estatísticas de média e variância do mapeamento estão disponíveis a partir de uma ponderação destes pontos sigma [28].

O problema em questão pode ser colocado da seguinte maneira [3]:

Seja um vetor n -dimensional de variáveis aleatórias \mathbf{x} , com média $\bar{\mathbf{x}}$ e uma co-variância \mathbf{P}_{xx} ,

que sofre uma transformação não-linear que relaciona \mathbf{x} a \mathbf{y} , dada por $\mathbf{y} = g(\mathbf{x})$. Deseja-se então uma aproximação para a média $\bar{\mathbf{y}}$ e a co-variância \mathbf{P}_{yy} para a variável aleatória \mathbf{y} resultante da transformação não-linear.

Este problema pode ser resolvido pelo método de Monte Carlo. A UT, entretanto, apresenta uma nova abordagem de resolução, que requer um menor esforço computacional quando comparado ao método de Monte Carlo.

Considerando que “com um número fixo de parâmetros é mais fácil aproximar uma distribuição gaussiana do que aproximar uma função/distribuição arbitrária não-linear” [3], deseja-se encontrar uma parametrização que capture a informação da média e da co-variância ao passo que permita a propagação direta desta informação através de um conjunto de equações não-lineares. A UT realiza isto a partir da geração de uma distribuição discreta que possua os mesmos momentos de primeira e segunda ordem (ou mais) da função não-linear, onde cada ponto desta aproximação discreta pode ser transformado diretamente. A média e a co-variância do conjunto transformado pode ser tido como a estimativa da transformação não-linear da distribuição original.

A figura 3.1 ilustra este conceito. A função não-linear é aplicada a cada ponto sigma escolhido a fim de se obter os pontos transformados. Uma ponderação destes pontos transformados é calculada para se formar uma estimativa da média e da variância da transformação não-linear.

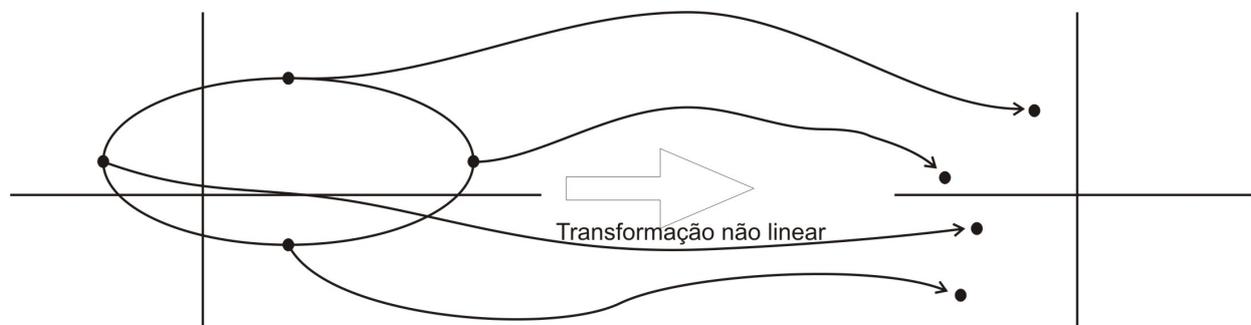


Figura 3.1: Princípio da transformada *Unscented* [3].

A partir desta transformação, é possível obter uma aproximação dos momentos de uma distribuição de probabilidade contínua $w(\hat{u})$ a partir dos momentos de uma distribuição discreta w_i . Desta forma, a UT faz com que a distribuição discreta possua os mesmos momentos da distribuição contínua após a transformação não-linear, de acordo com a seguinte equação [29]:

$$E_d\{\hat{u}^k\} = \int_{-\infty}^{\infty} \hat{u}^k w(\hat{u}) d\hat{u} = \sum_i w_i S_i^k \quad (3.1)$$

em que

- E_d é o valor esperado da distribuição discreta;
- \hat{u} representa o conjunto de variáveis aleatórias, com distribuição de probabilidade conhecida;
- $w(\hat{u})$ é a função densidade de probabilidade contínua;
- $\int_{-\infty}^{\infty} \hat{u}^k w(\hat{u}) d\hat{u}$ é o valor esperado E_c da distribuição contínua;
- w_i é a função densidade de probabilidade discreta;
- S_i são os pontos sigma e
- k é a ordem de aproximação desejada.

Será apresentada a formulação matemática descrita em [29] para a UT. Para verificar se as duas distribuições (contínua e discreta) levam ao mesmo mapeamento não-linear pode-se usar a expansão em séries de Taylor:

$$G(\bar{U} + \hat{u}) = G(\bar{U}) + \frac{dG}{du} \hat{u} + \frac{1}{2!} \frac{d^2G}{du^2} \hat{u}^2 + \frac{1}{3!} \frac{d^3G}{du^3} \hat{u}^3 + \dots \quad (3.2)$$

ou na representação polinomial:

$$G(\bar{U} + \hat{u}) = G(\bar{U}) + p(\hat{u}). \quad (3.3)$$

O valor esperado de $G(\bar{U} + \hat{u})$ é dado por:

$$\bar{G} = E\{G(\bar{U} + \hat{u})\} = E\{G(\bar{U})\} + E\{p(\hat{u})\} = G(\bar{U}) + \bar{P} \quad (3.4)$$

em que \bar{P} é o valor esperado do polinômio de Taylor.

A variância de $G(\bar{U} + \hat{u})$ é dada por:

$$\sigma_G^2 = E\{[G(\bar{U} + \hat{u}) - \bar{G}]^2\} = E\{p(\hat{u})^2\} - \bar{P}^2. \quad (3.5)$$

Dado que a representação compacta para o polinômio de Taylor também é válida para a distribuição discreta 3.1:

$$G(\bar{U} + S_i) = G(\bar{U}) + p(S_i). \quad (3.6)$$

O polinômio é o mesmo pois os pontos sigma S_i pertencem a distribuição de probabilidade de \hat{u} . A comparação das equações 3.1, 3.4, 3.5 e 3.6 resulta em um conjunto de condições para os pontos sigma:

$$\begin{aligned} w_0 &= 1 - \sum_i w_i \\ \sum_i w_i S_i^k &= E\{\hat{u}^k\} \end{aligned} \quad (3.7)$$

A ordem do polinômio de Taylor determina o número e os valores dos pontos sigma S_i e dos pesos w_i necessários para o cálculo da UT. Além disso, a ordem do polinômio de Taylor utilizado para definição dos pontos sigma condiciona a ordem dos momentos que poderão ser preditos pela UT. Por exemplo, para a obtenção de uma estimativa de segunda ordem (variância) para uma variável aleatória é necessário um polinômio de terceira ordem, que levará a três pontos sigma.

3.1.1 Determinação dos pontos sigma

A escolha dos pontos sigma S_i e dos pesos w_i depende da distribuição de probabilidade das variáveis estocásticas envolvidas no problema [28].

Para uma distribuição gaussiana $N \sim (0, \sigma^2)$, os valores dos pesos w_i e dos pontos sigma S_i estão mostrados na tabela 3.1 para dois e três pontos sigma [28]. A figura 3.2 ilustra a distribuição contínua e a distribuição discreta com os pontos sigma (distribuição gaussiana).

N	Pesos	Pontos sigma
2	1/2; 1/2	$-\sigma; \sigma$
3	1/6; 2/3; 1/6	$-\sqrt{3}\sigma; 0; \sqrt{3}\sigma$

Tabela 3.1: Pontos sigma e pesos para distribuição gaussiana $N \sim (0, \sigma^2)$.

Para uma distribuição uniforme $U \sim (-\sigma, \sigma)$, os valores dos pesos w_i e dos pontos sigma S_i estão mostrados na tabela 3.2 para dois e três pontos sigma [7].

N	Pesos	Pontos sigma
2	1/2; 1/2	$-\sqrt{3}/3\sigma; \sqrt{3}/3\sigma$
3	0,278; 0,444; 0,278	$\sqrt{3}/5\sigma; 0; \sqrt{3}/5\sigma$

Tabela 3.2: Pontos sigma e pesos para distribuição uniforme $U \sim (-\sigma, \sigma)$.

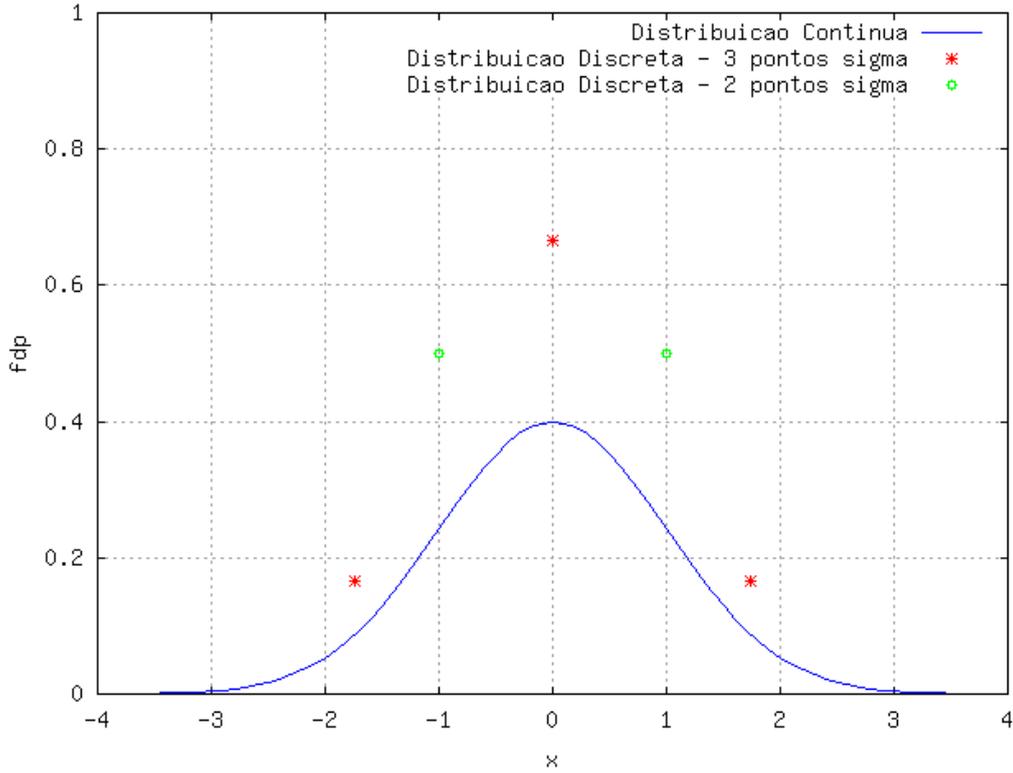


Figura 3.2: Função densidade de probabilidade contínua e a aproximação discreta.

Uma vez escolhidos os pontos sigma, é possível calcular os momentos da transformação não-linear[28]. O valor médio é dado por:

$$\bar{G} = E\{G(\bar{U} + \hat{u})\} = \sum_i w_i G(\bar{U} + S_i). \quad (3.8)$$

Os momentos de ordem superior são dados por:

$$E\{(G(\bar{U} + \hat{u}) - \bar{G})^k\} = \sum_i w_i (G(\bar{U} + S_i) - \bar{G})^k. \quad (3.9)$$

Para o desvio padrão, por exemplo:

$$\sigma_G = \sqrt{\sum_i w_i (G(\bar{U} + S_i) - \bar{G})^2}. \quad (3.10)$$

Para o caso de múltiplas variáveis independentes, a relação entre os pontos sigma e os momentos possuem termos cruzados. Em [28], é mostrada uma forma de cálculo da UT para múltiplas variáveis aleatórias independentes, mostrando que para uma aproximação de segunda ordem são necessários $2^{n_{RV}} + 2n_{RV}$ pontos sigma. Entretanto, será utilizada uma abordagem mais simples, porém que utiliza mais pontos sigma, para o cálculo da UT para múltiplas variáveis [30]. Nesta abordagem, os pontos sigma permanecem os mesmos já calculados e os pesos são multiplicados. Um exemplo está mostrado na tabela 3.3 para o

cálculo dos pesos e dos pontos sigma para duas variáveis independentes de distribuição gaussiana (usando três pontos sigma). Neste caso são necessários nove cálculos para realização da UT.

	Variável 1			
Variável 2	$-\sqrt{3}$	0	$\sqrt{3}$	Pesos
$-\sqrt{3}$	1/36	1/9	1/36	1/6
0	1/9	4/9	1/9	2/3
$\sqrt{3}$	1/36	1/9	1/36	1/6
Pesos	1/6	2/3	1/6	

Tabela 3.3: Arranjo da UT para duas variáveis.

Para o caso de múltiplas variáveis correlacionadas, os pontos sigma são calculados como se as variáveis fossem independentes e em seguida é utilizada uma transformação para obter o novo conjunto de pontos sigma. De fato é realizada uma multiplicação dos pontos sigma por uma matriz de co-variância [28].

3.1.2 Comparação da UT com o método de Monte Carlo

Embora exista grande semelhança entre a UT e o método de Monte Carlo, uma diferença crucial deve ser apontada. Diferentemente do método de Monte Carlo, a UT não seleciona os pontos de forma aleatória, reduzindo significativamente o número de pontos necessários. Com isto, a UT apresenta grande vantagem no quesito esforço computacional requerido para cálculo das medidas estatísticas.

A capacidade de predição do valor esperado e da variância pela UT será analisada nesta seção para uma, duas e três variáveis aleatórias. Além disto, será mostrada uma comparação da UT com o método de Monte Carlo.

3.1.2.1 Caso unidimensional – variável aleatória de distribuição gaussiana

Seja $f(x)$ dada por:

$$f(x) = \cos(x) \tag{3.11}$$

em que x é uma variável aleatória de distribuição gaussiana com média μ_x e variância $\sigma^2 = 0,2$.

A média desta função pode ser analiticamente calculada e é dada por:

$$E[f(x)] = \cos(\mu_x) \exp\left(-\frac{\sigma^2}{2}\right) \tag{3.12}$$

e o desvio padrão é dado por:

$$\sigma_{f(x)} = \sqrt{\left(\frac{1}{2} + \frac{1}{2} \cos(2x) \exp(-2\sigma^2) - \cos(x)^2 \exp(-\sigma^2)\right)}. \quad (3.13)$$

O valor esperado e o desvio padrão foram calculados pelas fórmulas 3.12 e 3.13, pela UT com dois e três pontos sigma e pelo método de Monte Carlo com 500 amostras. Os resultados estão mostrados nas figuras 3.3, 3.4, 3.5 e 3.6.

A função $f(x)$, em azul, e seu valor médio $E[f(x)]$, em vermelho, estão apresentados na figura 3.3 para os casos analítico, UT calculada com três e dois pontos sigma e método de Monte Carlo. Na figura 3.4 é apresentado o erro absoluto dos valores preditos de valor médio pela UT e pelo método de Monte Carlo. A UT calculada com dois e três pontos sigma apresenta resultados satisfatórios, enquanto o método de Monte Carlo apresenta alguns pontos com elevado valor de erro.

A figura 3.5 apresenta, em azul, o desvio padrão $\sigma_{f(x)}$ para os casos analítico, UT calculada com três e dois pontos sigma e método de Monte Carlo. Em vermelho, é apresentado o erro absoluto em relação ao cálculo analítico. Para os cálculos com a UT, este erro apresenta um padrão: para o cálculo da UT com três pontos sigma, este erro é pequeno, e para a UT com dois pontos sigma, quando a função se aproxima dos pontos de mínimo e de máximo (pontos com derivada nula), o resultado é divergente. Para o método de Monte Carlo, o erro não segue um padrão, apresentando valores diversos.

Pode ser observado que para a predição do valor esperado, a UT, seja calculada com dois ou três pontos sigma, apresenta um melhor desempenho que o método de Monte Carlo usando um número significativamente menor de cálculos. Para estes resultados, a UT utilizou dois e três cálculos, enquanto o método de Monte Carlo utilizou 500 cálculos. Para o desvio padrão, somente a UT com três pontos sigma apresenta um resultado satisfatório. Esta constatação já era prevista, pois o número de pontos sigma escolhidos condiciona a precisão dos resultados, sendo que com dois pontos sigma, apenas a estimativa da média apresentará um bom resultado.

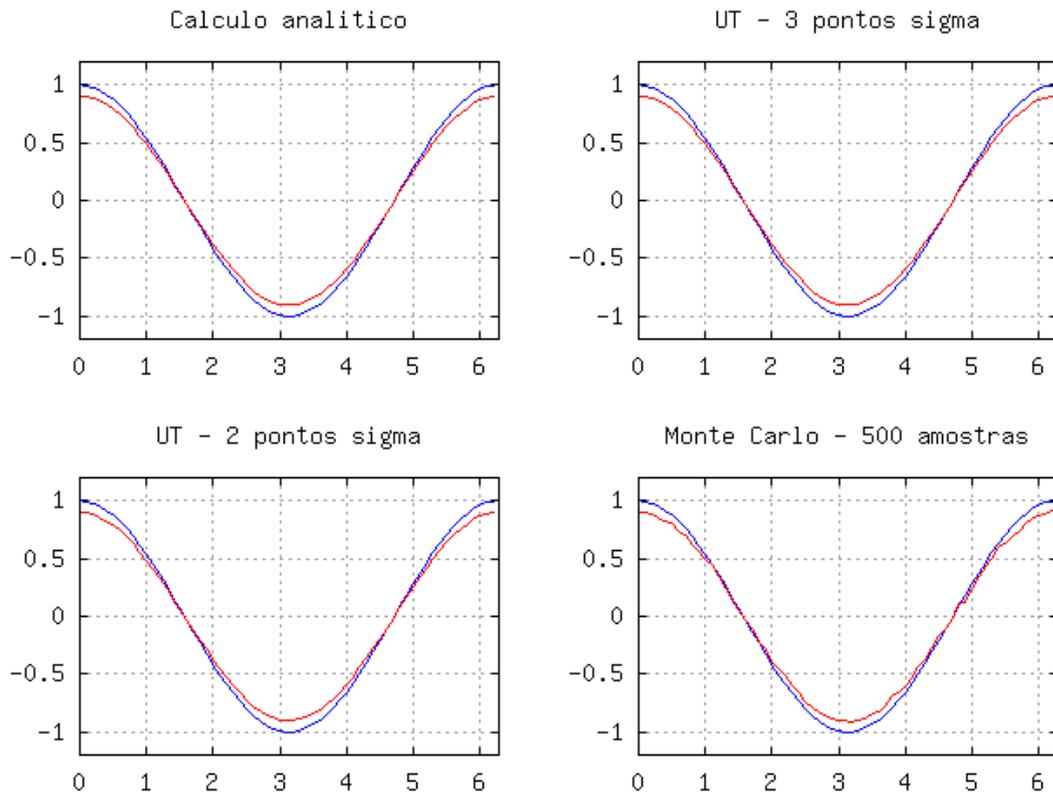


Figura 3.3: Função $f(x)$ e seu valor esperado $E[f(x)]$ para os casos analisados.

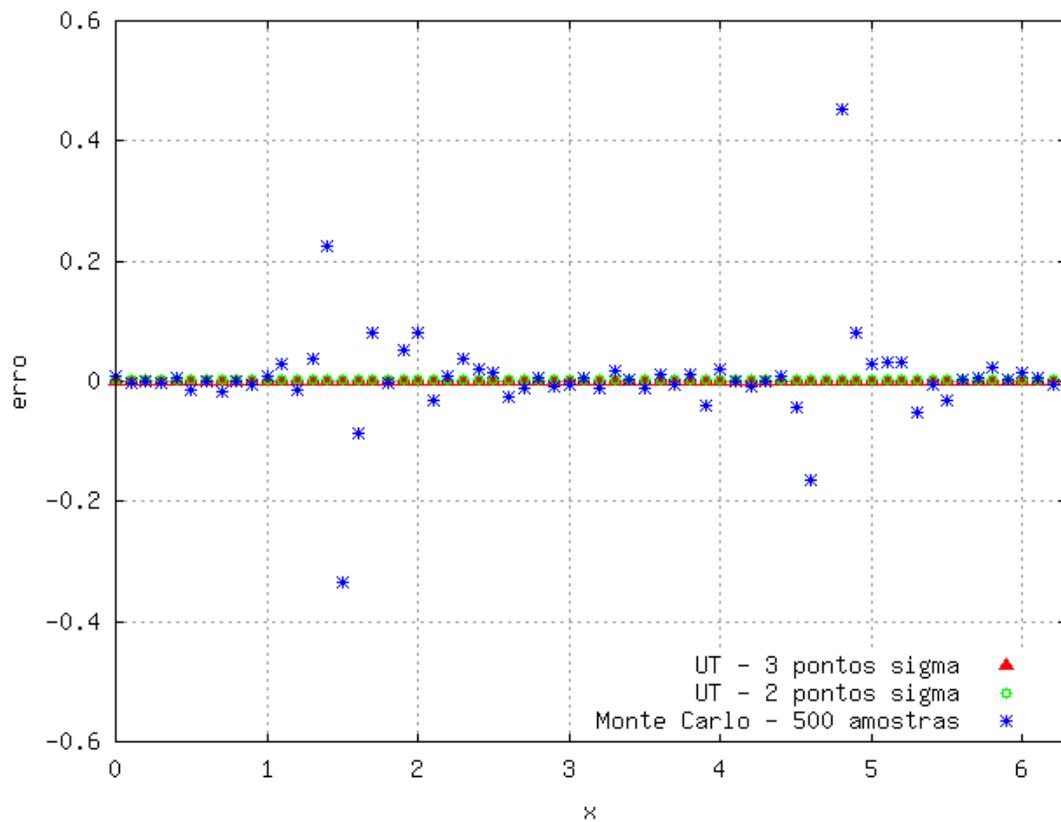


Figura 3.4: Erro absoluto do valor esperado.

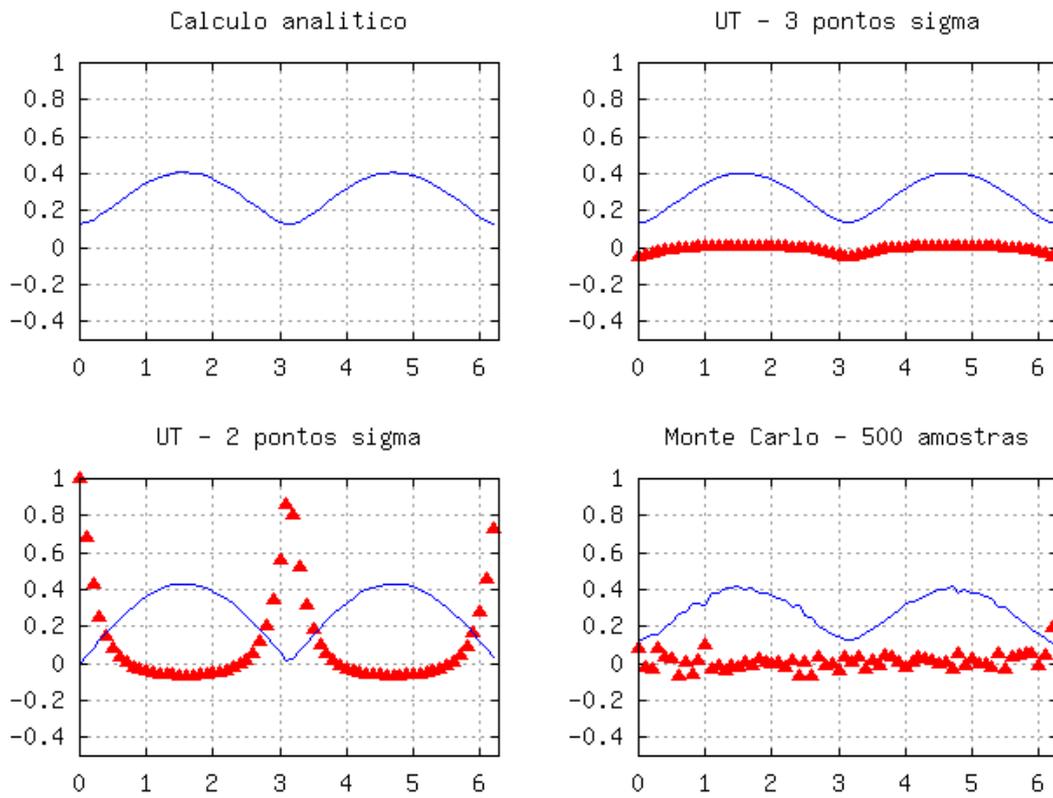


Figura 3.5: Comparação do desvio padrão $\sigma_{f(x)}$ analítico com os casos analisados.

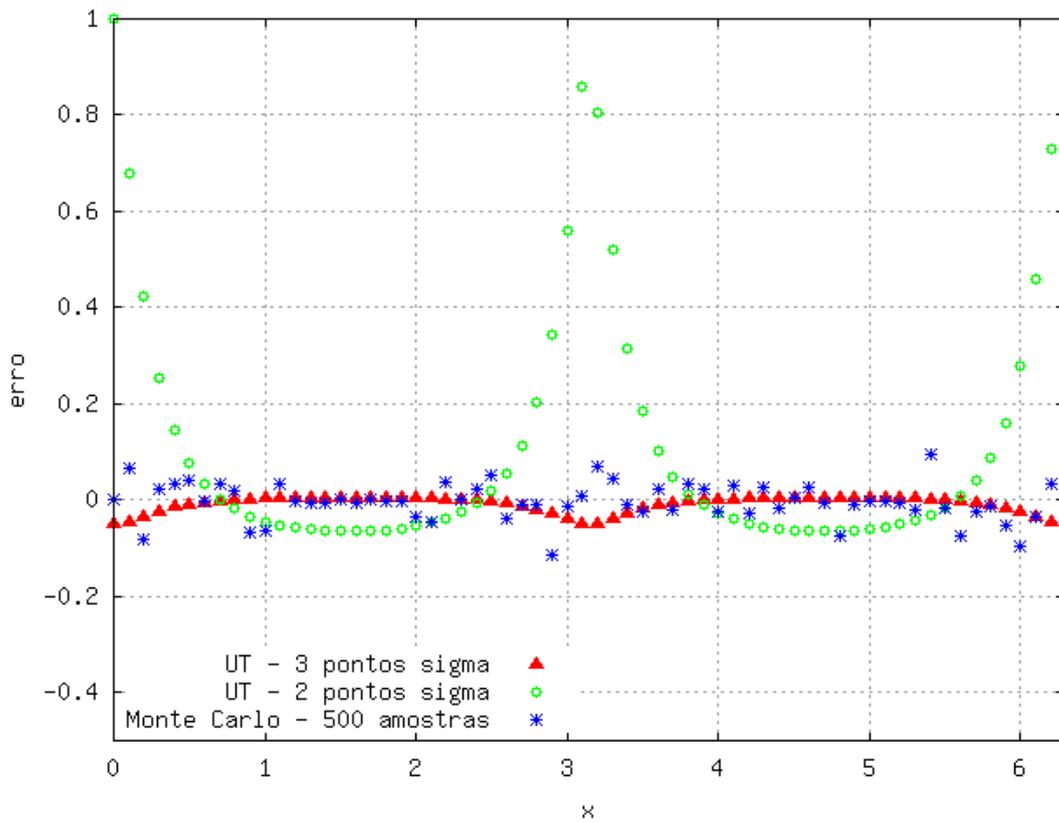


Figura 3.6: Erro absoluto do desvio padrão.

3.1.2.2 Caso unidimensional – distribuição uniforme e distribuição gaussiana

Seja $f(x)$ dada por:

$$f(x) = \cos(2x) \exp(x) \quad (3.14)$$

em que x é uma variável aleatória, podendo ser:

- de distribuição uniforme $U \sim (\frac{2\pi}{3} - \frac{\pi}{10}, \frac{2\pi}{3} + \frac{\pi}{10})$ e
- de distribuição gaussiana (ou normal) $N \sim (\frac{2\pi}{3}, \frac{1}{100})$.

Estas distribuições estão mostradas na figura 3.7.

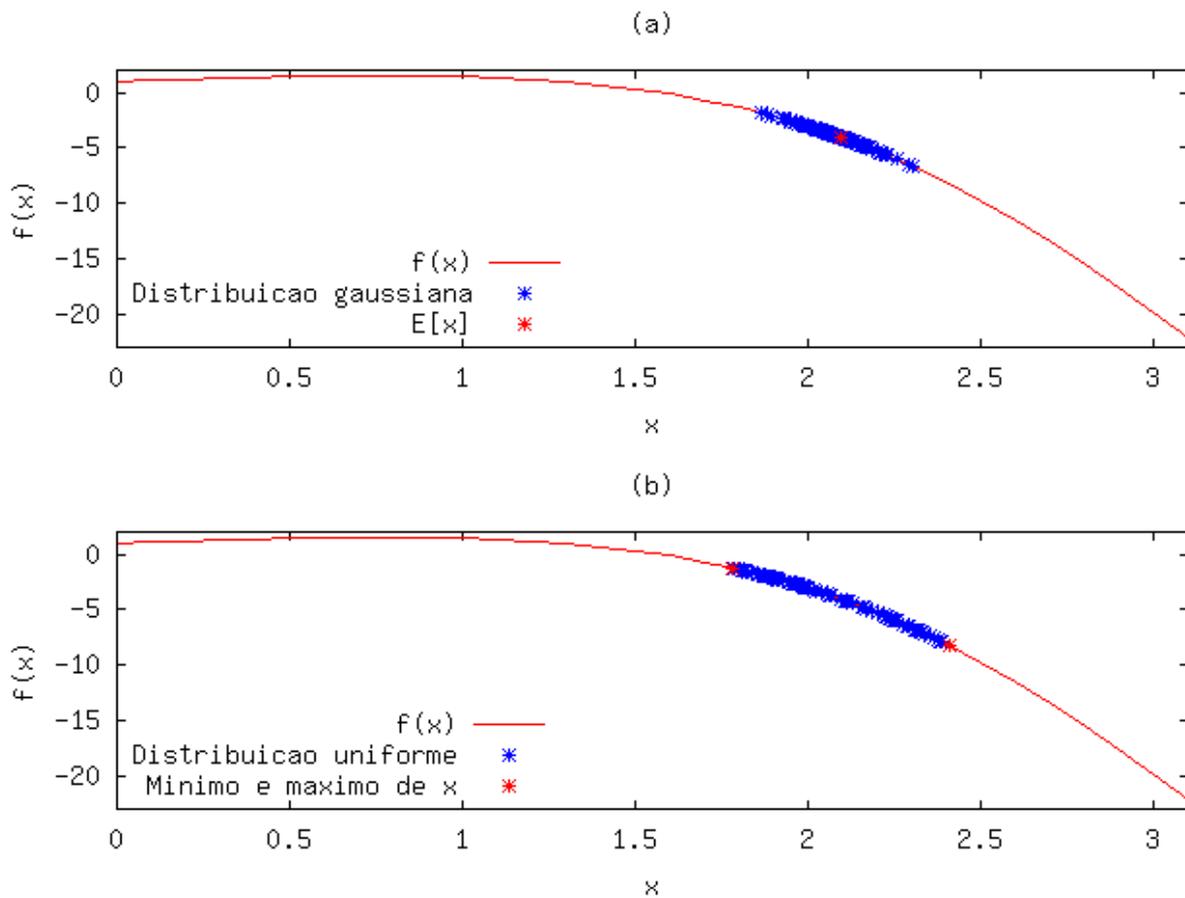


Figura 3.7: Função $f(x)$ para as duas distribuições: (a) $x \sim N(\frac{2\pi}{3}, \frac{1}{100})$ e (b) $x \sim U(\frac{2\pi}{3} - \frac{\pi}{10}, \frac{2\pi}{3} + \frac{\pi}{10})$.

Para calcular o valor médio e a variância de $f(x)$ para as duas distribuições foi utilizado o método de Monte Carlo com 100.000 amostras e a UT. Os resultados estão mostrados na tabela 3.4, para o caso de distribuição uniforme e na tabela 3.5, para o caso de distribuição gaussiana.

	$E[f(x)]$	$\sigma_{f(x)}^2$
Monte Carlo	-4,3015	4,1510
UT-3	-4,2923	4,1358
UT-2	-4,2909	4,0718

Tabela 3.4: Comparação do valor esperado e da variância de $f(x)$ para $x \sim U(\frac{2\pi}{3} - \frac{\pi}{10}, \frac{2\pi}{3} + \frac{\pi}{10})$.

	$E[f(x)]$	$\sigma_{f(x)}^2$
Monte Carlo	-4,1303	1,2370
UT-3	-4,1304	1,2469
UT-2	-4,1305	1,2327

Tabela 3.5: Comparação do valor esperado e da variância de $f(x)$ para $x \sim N(\frac{2\pi}{3}, \frac{1}{100})$.

A relação entre o valor médio dado pelo método de Monte Carlo e o número de amostras necessárias é mostrada nas figuras 3.8, 3.9, 3.10 e 3.11, juntamente com o valor médio calculado pela UT. Para o cálculo da UT foram utilizadas duas (UT-2) e três amostras (UT-3). Como pode ser observado nas figuras, a técnica de Monte Carlo é muito dispendiosa, pois requer um número muito grande de avaliações da função. A UT consegue uma boa aproximação (tabelas 3.5 e 3.4) com um número bem menor de simulações. É possível verificar novamente que para a UT com dois pontos sigma não é possível obter uma boa aproximação da variância.

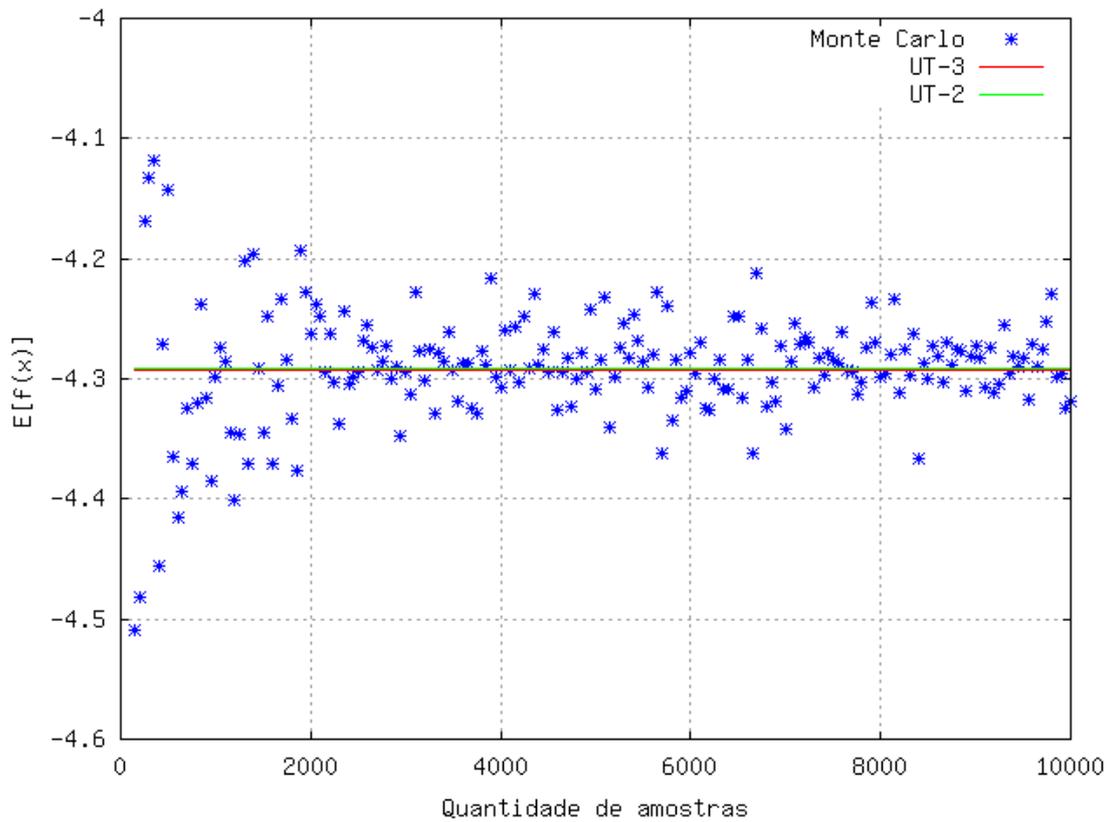


Figura 3.8: Evolução de $E[f(x)]$ para o método de Monte Carlo para a variável aleatória x de distribuição uniforme (10.000 amostras).

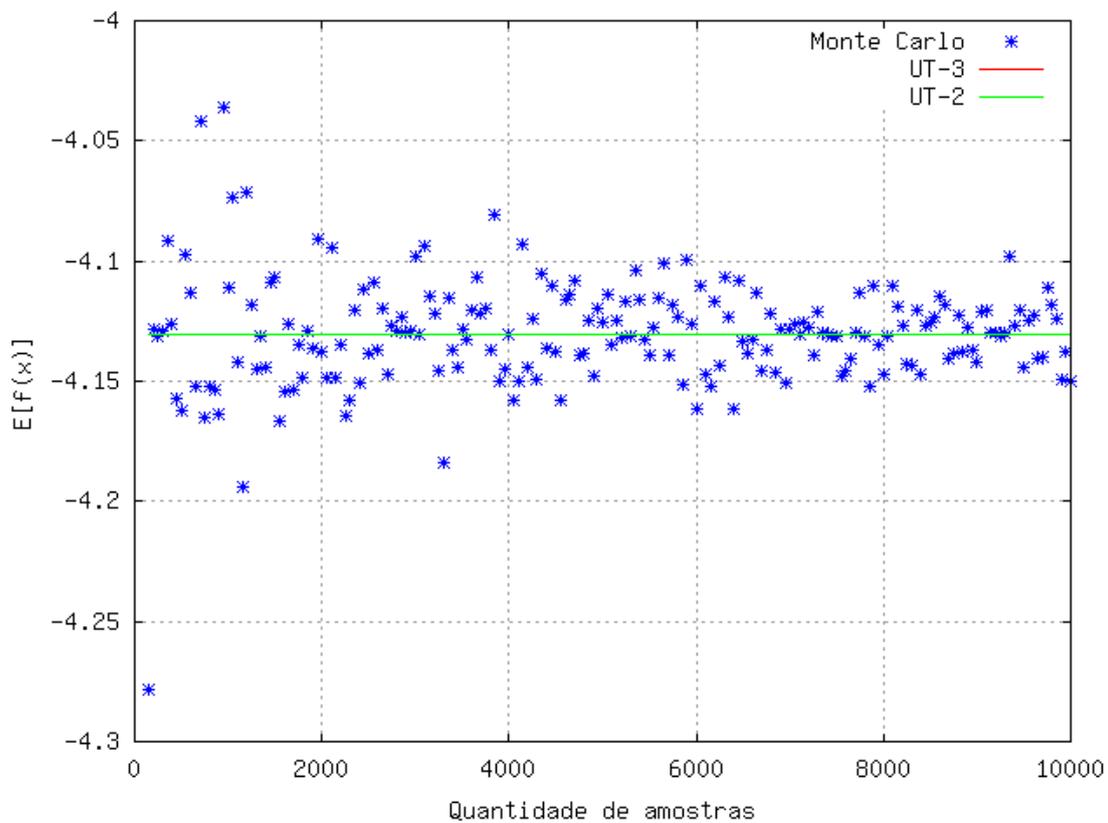


Figura 3.9: Evolução de $E[f(x)]$ para o método de Monte Carlo para a variável aleatória x de distribuição gaussiana (10.000 amostras).

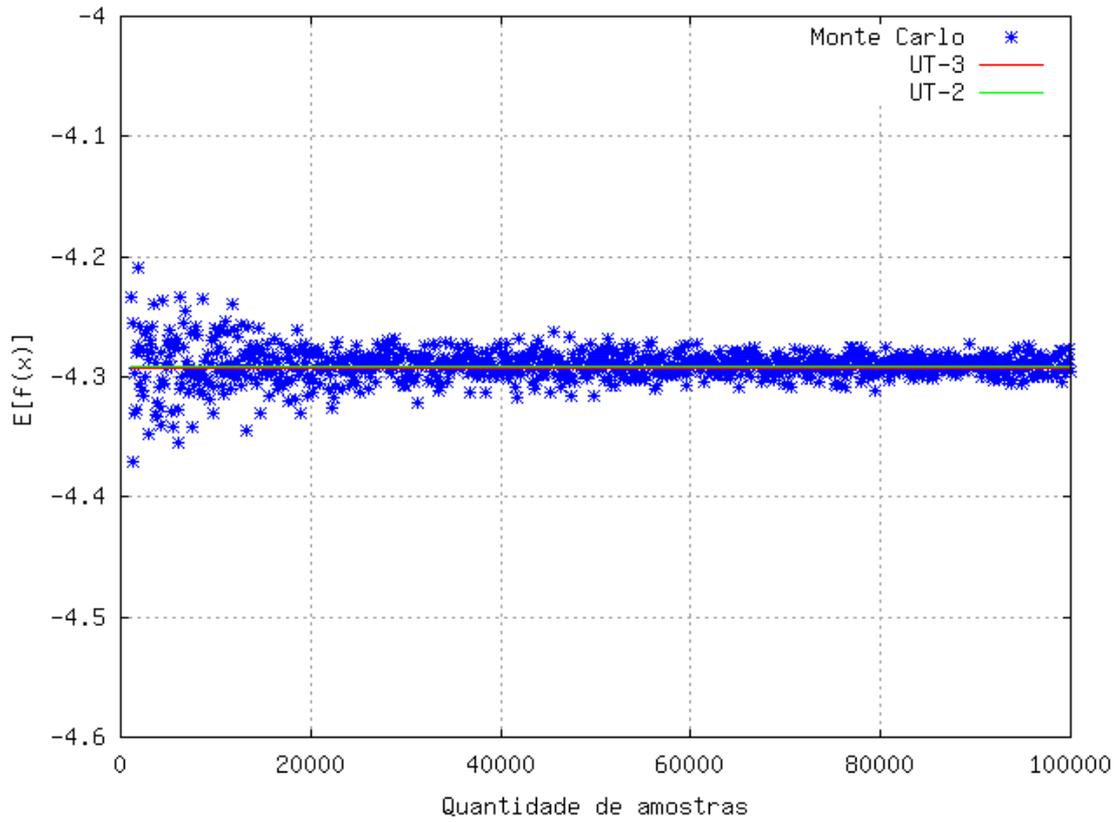


Figura 3.10: Evolução de $E[f(x)]$ para o método de Monte Carlo para a variável aleatória x de distribuição uniforme (100.000 amostras).

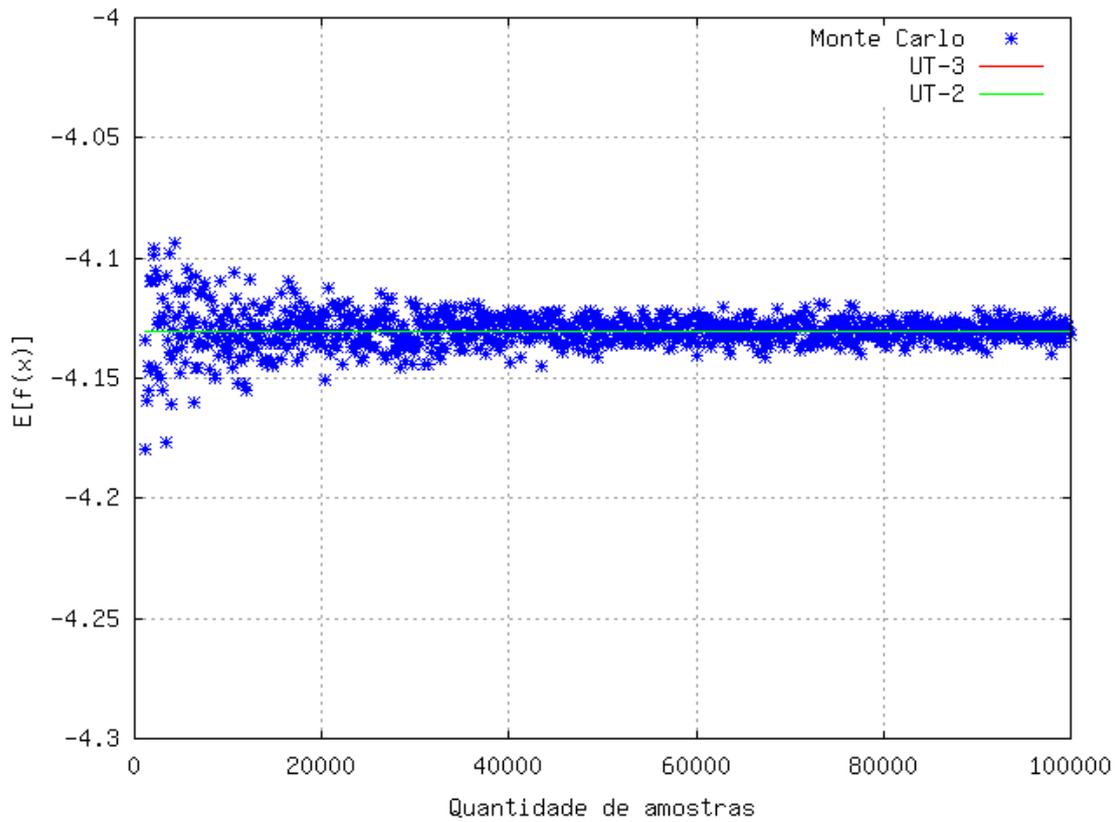


Figura 3.11: Evolução de $E[f(x)]$ para o método de Monte Carlo para a variável aleatória x de distribuição gaussiana (100.000 amostras).

3.1.2.3 Caso bidimensional

Para avaliação do desempenho da UT com duas variáveis será utilizada a mesma equação da seção 2.3.1, equação 2.9, com $n = 2$.

Será considerado x_i como variáveis aleatórias independentes dada pelas seguintes distribuições:

$$\begin{aligned}x_1 &\sim N\left(\frac{1}{10}, \frac{1}{100}\right) \\x_2 &\sim N\left(\frac{3}{10}, \frac{25}{1.000}\right)\end{aligned}\quad (3.15)$$

O método de Monte Carlo foi realizado com 100.000 amostras a fim de obter um resultado mais preciso. Em seguida, foi realizado o mesmo método com 1.000 amostras. Os resultados estão mostrados nas tabelas 3.6 e 3.7.

	$E[f_1]$	$E[f_2]$
MC 100.000 amostras	0,26850	0,70699
MC 1.000 amostras	0,26326	0,71061
UT-3	0,26826	0,70716
UT-2	0,26828	0,70714

Tabela 3.6: Comparação do valor esperado dado pelo método de Monte Carlo e pela UT.

	$\sigma_{f_1}^2$	$\sigma_{f_2}^2$
MC 100.000 amostras	0,0051323	0,0021762
MC 1.000 amostras	0,0049399	0,0020819
UT-3	0,0051368	0,0021779
UT-2	0,0052660	0,0022237

Tabela 3.7: Comparação da variância dado pelo método de Monte Carlo e pela UT.

Tomando os valores obtidos com o método de Monte Carlo com 100.000 amostras como mais precisos, pode-se calcular o erro absoluto das outras estimativas em relação a esta. A tabela 3.8 apresenta estes valores de erro absoluto (percentual).

	Erro $E[f_1]$ %	Erro $E[f_2]$ %	Erro $\sigma_{f_1}^2$ %	Erro $\sigma_{f_2}^2$ %
MC 1.000 amostras	1,95	0,51	3,74	4,33
UT-3	0,08	0,02	0,09	0,07
UT-2	0,08	0,02	2,60	2,18

Tabela 3.8: Percentual de erro absoluto em relação ao método de Monte Carlo com 100.000 amostras.

Estes resultados permitem concluir que a UT é capaz de calcular as medidas estatísticas da transformação não-linear de duas variáveis gaussianas independentes. Ainda, estes resultados permitem concluir que a UT apresenta um ótimo desempenho com um número significativamente menor de cálculos quando comparado ao método de Monte Carlo. Ao verificar os erros absolutos das estimativas, o método de Monte Carlo necessitaria de mais de mil cálculos para obter uma estimativa com baixo erro, enquanto a UT obtém esta estimativa com nove cálculos (UT calculada para três pontos sigma), no caso do valor esperado e do desvio padrão, e com quatro cálculos (UT calculada para dois pontos sigma), no caso do valor esperado apenas. Novamente é possível perceber que no caso da UT calculada com dois pontos sigma, o resultado obtido para o desvio padrão não apresenta um desempenho tão bom quanto o resultado obtido com três pontos sigma.

3.1.2.4 Caso tridimensional

A mesma análise realizada para o caso bidimensional será apresentada para o caso tridimensional. Como função de teste será utilizada novamente a equação da seção 2.3.1, equação 2.9, com $n = 3$. Será considerado x_i como variáveis aleatórias independentes dada pelas seguintes distribuições:

$$\begin{aligned}
 x_1 &\sim N\left(\frac{1}{10}, \frac{25}{1.000}\right) \\
 x_2 &\sim N\left(\frac{1}{10}, \frac{1}{100}\right) \\
 x_3 &\sim N\left(\frac{3}{10}, \frac{25}{1.000}\right)
 \end{aligned} \quad (3.16)$$

As predições foram realizadas com o método de Monte Carlo com 1.000.000 amostras e a UT com dois e três pontos sigma. Os resultados estão mostrados na tabela 3.9. O erro absoluto das estimativas dadas pela UT em relação a estimativa dada pelo método de Monte Carlo estão mostrados na tabela 3.10.

	$E[f_1]$	$E[f_2]$	σ_{f_1}	σ_{f_2}
MC 1.000.000 amostras	0,54193	0,73681	0,13638	0,094696
UT-3	0,54211	0,73695	0,13364	0,093014
UT-2	0,54223	0,73632	0,14919	0,10382

Tabela 3.9: Comparação do valor esperado e do desvio padrão dado pelo método de Monte Carlo e pela UT.

	Erro $E[f_1]$ %	Erro $E[f_2]$ %	Erro σ_{f_1} %	Erro σ_{f_2} %
UT-3	0,03	0,02	5,86	3,39
UT-2	0,05	0,06	5,88	8.36

Tabela 3.10: Percentual de erro absoluto em relação ao método de Monte Carlo com 1.000.000 amostras.

Os resultados do erro percentual mostram que a UT apresenta uma boa aproximação para o valor da média. Já para o desvio padrão não é possível obter uma boa estimativa. Seriam necessários mais pontos sigma para uma boa estimativa do desvio padrão. Com isto verifica-se a questão da dimensionalidade. Para a estimativa da média e desvio padrão relacionada a mais variáveis é necessário um maior número de pontos sigma, ou seja, um maior número de avaliações.

3.1.2.5 Conclusão

Os resultados apresentados mostram que a UT apresenta um bom desempenho no cálculo das estimativas de valor esperado e desvio padrão, com um número significativamente menor de cálculos quando comparada ao método de Monte Carlo. Neste sentido, a UT pode ser utilizada para os cálculos de média e variância necessários para a otimização robusta.

3.2 METODOLOGIA DE PROJETO ROBUSTO

O projeto robusto consiste na modelagem das incertezas associadas ao projeto como variáveis aleatórias e na otimização destas variáveis aleatórias através de um algoritmo de otimização robusta. A otimização robusta requer que as medidas estatísticas — média e variância — das funções a serem otimizadas sejam calculadas. Com o uso da UT é possível realizar este cálculo sem onerar o processo de otimização. Desta

forma, a otimização robusta pode ser realizada utilizando as informações de média e variância calculadas pela UT na formulações da seção 2.3.4 (equações 2.14, 2.15 e 2.16).

Contudo, nem sempre é necessário a otimização robusta de todas as variáveis envolvidas no problema. Uma análise de sensibilidade *a priori* pode ser realizada para eliminar da otimização robusta variáveis que não interferem de maneira significativa na sensibilidade do problema. Mesmo com o uso da UT, uma otimização robusta com muitas variáveis pode ser custosa.

Neste sentido, é proposto o uso da otimização robusta em conjunto com a otimização clássica, em que a robustez das soluções é calculada pela UT. Assim, parâmetros sensíveis serão tratados de acordo com a formulação robusta, utilizando a UT, enquanto outros parâmetros serão otimizados pela formulação clássica.

Será apresentada uma metodologia de projeto robusto em três etapas (figura 3.12):

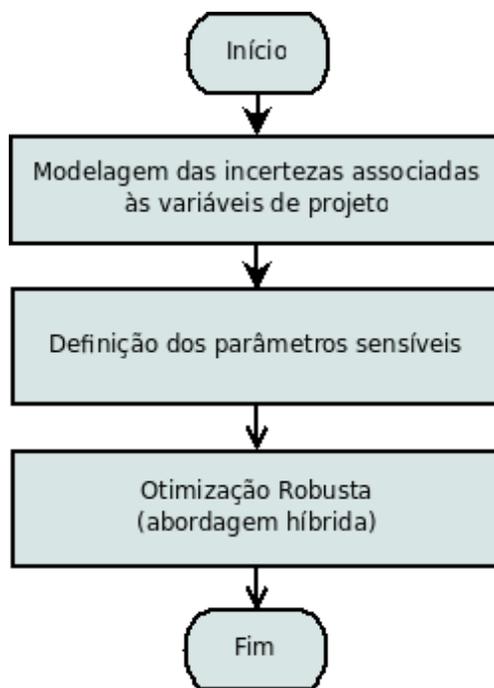


Figura 3.12: Fluxograma da metodologia de projeto robusto.

1. Modelagem das incertezas associadas às variáveis de entrada.
2. Definição dos parâmetros sensíveis de projeto.
3. Otimização conjunta dos parâmetros sensíveis e dos outros parâmetros de projeto, utilizando uma abordagem híbrida.

3.2.1 Modelagem das incertezas associadas às variáveis de entrada

A primeira etapa da otimização robusta é a modelagem das incertezas associadas às variáveis de entrada. Esta caracterização é feita pelo projetista de acordo com as informações que se têm sobre o problema. Por exemplo, no processo de fabricação de um dispositivo, o projetista deve saber caracterizar as incertezas associadas aos componentes de fabricação. Esta caracterização significa escolher uma distribuição de probabilidade que se encaixe para a aleatoriedade da variável em análise. Por exemplo, se a incerteza de uma variável de projeto é definida por uma distribuição gaussiana, é necessário saber qual a variância, e se é definida por uma distribuição uniforme, é necessário definir a faixa de variação.

3.2.2 Definição dos parâmetros sensíveis

A segunda etapa consiste na verificação de quais parâmetros modelados na etapa anterior são parâmetros sensíveis de projeto. Isto porque nem todos os parâmetros são necessariamente impactantes no projeto, acarretando em cálculos desnecessários para a obtenção da média e da variância. O impacto que perturbações nas variáveis de entrada geram nas funções objetivo deve ser avaliado. Esta avaliação acarretará na forma de tratamento destas variáveis no projeto robusto:

- Os parâmetros mais significantes são tratados como variáveis aleatórias e suas medidas estatísticas de média e variância são consideradas.
- A variação dos parâmetros menos significantes é desconsiderada para efeitos de otimização robusta, sendo estes parâmetros considerados apenas a otimização clássica.

Em [24, 31] é mostrada uma abordagem para determinação dos parâmetros sensíveis utilizando o coeficiente de correlação entre as variáveis de entrada e as funções objetivo. O coeficiente de correlação r entre duas variáveis aleatórias x e y é dado por:

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2} \quad (3.17)$$

em que

- $r = 0$ corresponde a não-associação;
- quanto maior o valor de $|r|$, mais forte a associação;
- $r > 0$ corresponde a ambas variáveis crescendo juntas e $r < 0$ corresponde a uma variável ficando menor à medida que a outra fica maior.

Nesta abordagem, para definição dos parâmetros sensíveis, o coeficiente de correlação é calculado entre as variáveis de entrada e as funções objetivo. As variáveis de entrada que apresentam pouca correlação com as funções objetivo (valor da função de correlação próximo a zero) tendem a ter menos impacto nas funções objetivo, podendo ser desconsideradas para a otimização robusta. Contudo, uma crítica deve ser feita ao emprego do coeficiente de correlação para definição dos parâmetros sensíveis. O coeficiente de correlação é utilizado para verificar o quanto uma variável acompanha a função objetivo. Se uma determinada variável não está acompanhando uma função objetivo (por exemplo, tem um comportamento oscilatório), ela pode apresentar um coeficiente de correlação próximo a zero, mas ainda assim ser uma variável sensível de projeto.

Nesta dissertação é apresentada uma nova abordagem para o problema de definição dos parâmetros sensíveis que utiliza a UT.

Para avaliar como uma variável aleatória afeta a função objetivo será realizada uma análise de sensibilidade do problema. Esta análise de sensibilidade será realizada a partir da avaliação da variância da função objetivo dada uma perturbação nas variáveis de entrada. Esta perturbação nas variáveis de entrada é determinada pela modelagem da incerteza realizada na etapa anterior. A variância da transformação será calculada pela UT. Para isso, será calculada a UT para um conjunto de variáveis de entrada dentro do domínio de busca do problema. De posse do valor de variância relacionado a cada variável, é possível verificar quais variáveis são mais sensíveis. As variáveis que resultarem em uma baixa variância podem ser consideradas pouco relevantes para o projeto.

Este procedimento de determinação dos parâmetros sensíveis é mostrado na figura 3.13. Na próxima seção será mostrado um exemplo de determinação dos parâmetros sensíveis.

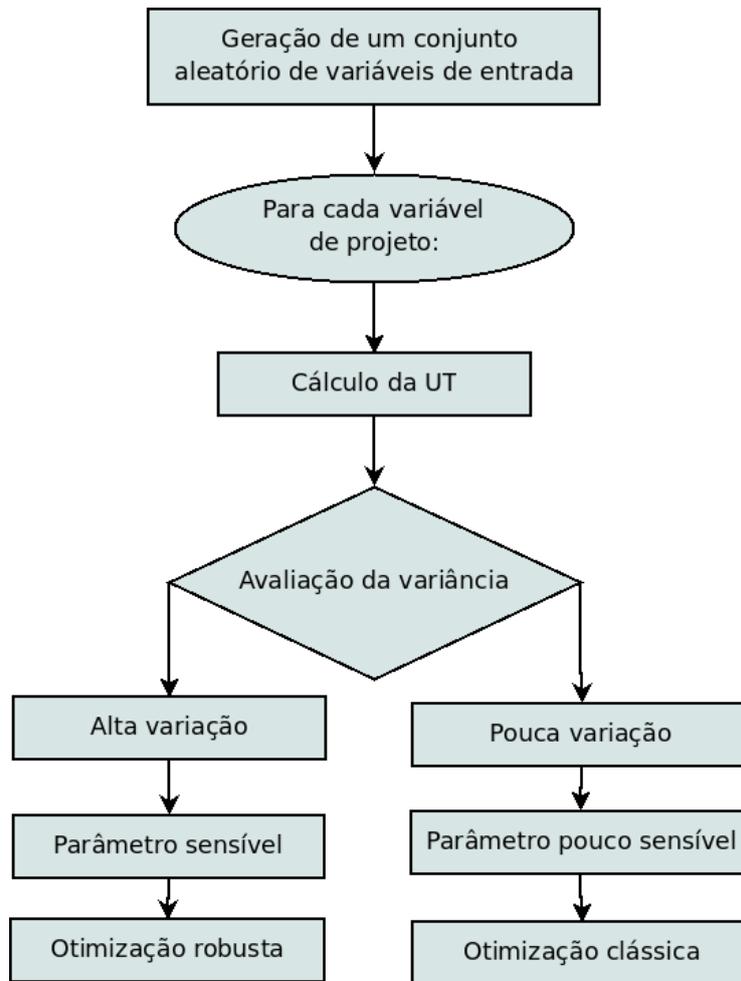


Figura 3.13: Fluxograma para a determinação das variáveis sensíveis de projeto.

3.2.2.1 Exemplo de definição dos parâmetros sensíveis

Para ilustrar esta análise de sensibilidade dos parâmetros de projeto, considere a seguinte função:

$$f(x_1, x_2, x_3) = -x_2 \cos(x_1) - x_2^2 - 2 \cos(x_2^3) + \cos(x_1^2) - 3x_3 \quad (3.18)$$

em que $0, 1 \leq x_1, x_2, x_3 \leq \pi - 0, 1$.

Suponha que a modelagem da incerteza associada às variáveis x_1 , x_2 e x_3 resultou em:

$$\begin{aligned} x_1 &\sim U\left(x_1 - \frac{\pi}{100}, x_1 + \frac{\pi}{100}\right) \\ x_2 &\sim N\left(x_2, \left(\frac{\pi}{50}\right)^2\right) \\ x_3 &\sim N\left(x_3, \left(\frac{\pi}{100}\right)^2\right) \end{aligned} \quad (3.19)$$

Suponha também que é desejável que a variância de $f(x_1, x_2, x_3)$ seja menor que $\frac{\pi}{10}$.

Dado este cenário, deseja-se identificar os parâmetros sensíveis de projeto.

Primeiramente, foi gerado o conjunto aleatório de variáveis de entrada. Para cada variável, foi calculada a variância utilizando a UT de acordo com os parâmetros da equação 3.19. A UT foi calculada com três pontos sigma. A tabela 3.11 apresenta os resultados obtidos.

Variável	$\max \sigma_f^2$	$E[\sigma_f^2]$	$\sigma_f^2 \leq \frac{\pi}{10}$
x_1	0,0063	0,0012	100.000
x_2	3,7091	0,6824	50.856
x_3	0,0088	0,0088	100.000

Tabela 3.11: Análise dos parâmetros sensíveis de projeto.

Ao considerar o valor máximo de variância ($\max \sigma_f^2$ na tabela 3.11), percebe-se que as variáveis x_1 e x_3 apresentam este valor inferior ao limite estabelecido ($\frac{\pi}{10}$). Já a variável x_2 , tem seu valor médio de variância ($E[\sigma_f^2]$) acima do valor permitido. Isto resulta em pouco mais de 50% de amostras abaixo do valor de variância especificado ($\sigma_f^2 \leq \frac{\pi}{10}$). A variável x_2 é portanto a variável sensível de projeto.

Para ilustrar o comportamento de $f(x_1, x_2, x_3)$ para cada variável, é apresentada a figura 3.14. Visualmente, percebe-se que a função $f(x_1, x_2^{cte}, x_3^{cte})$ ¹ apresenta uma suave curva e $f(x_1^{cte}, x_2^{cte}, x_3)$ apresenta um comportamento decrescente linear, enquanto que $f(x_1^{cte}, x_2, x_3^{cte})$ apresenta uma oscilação, retratando grande variação em todo o espaço de busca.

Para esta função também foi calculado o coeficiente de correlação de acordo com a equação 3.17. O resultado está mostrado na tabela 3.12.

Variável	x_1	x_2	x_3
r	0,27042	-0,51248	-0,65397

Tabela 3.12: Coeficiente de correlação entre as variáveis x_1 , x_2 e x_3 e a função $f(x_1, x_2, x_3)$.

Os resultados mostrados na tabela 3.12 indicam que a variável x_1 é a variável menos importante para o projeto e que as variáveis x_2 e x_3 apresentam alta correlação com a função objetivo. Contudo, como pode ser observado, a variável x_3 é apenas uma parcela adicionada a $f(x_1, x_2)$, sendo seu impacto constante em todo o espaço de busca. A variável x_2 é a variável sensível de projeto.

Por fim, um importante ponto deve ser lembrado. A análise dos parâmetros sensíveis depende fortemente de como as variáveis aleatórias são modeladas e isto não é refletido no coeficiente de correlação,

¹Em que *cte* significa que esta variável é mantida constante.

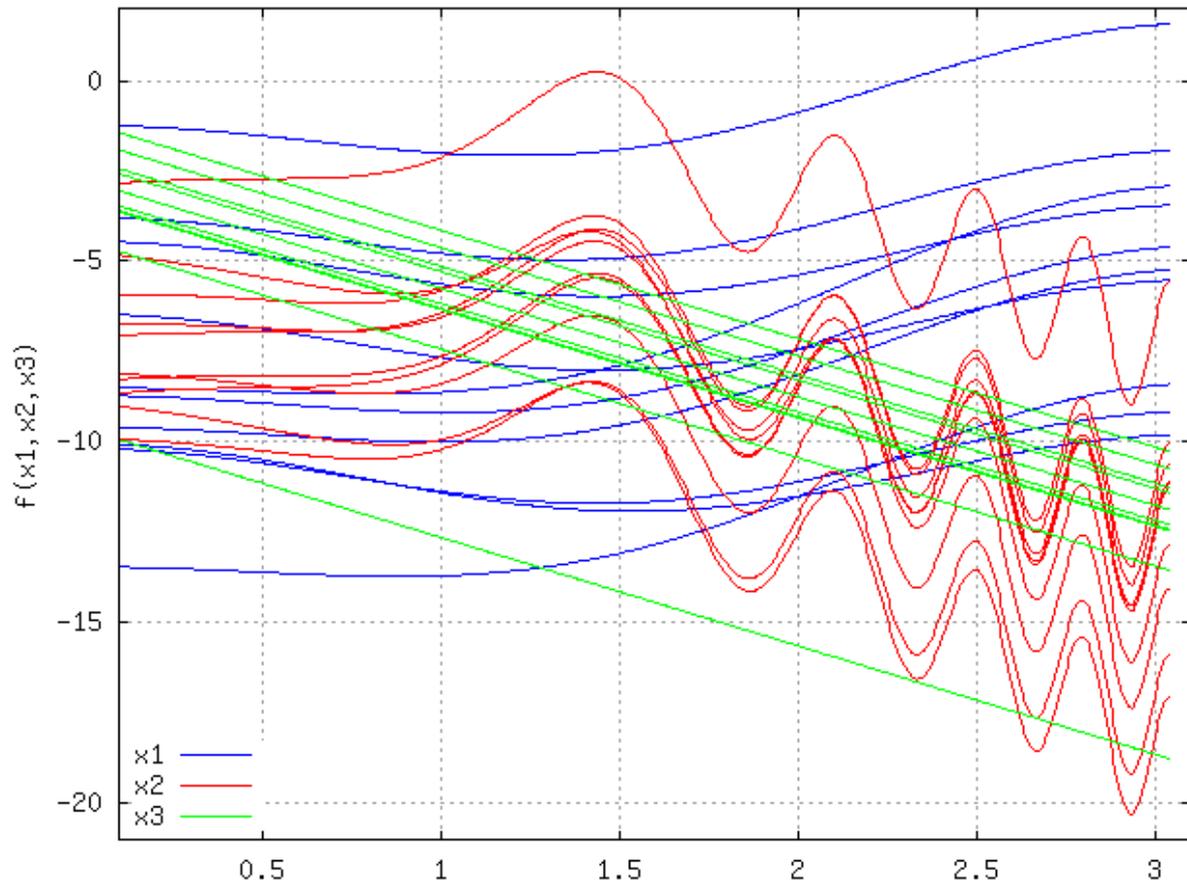


Figura 3.14: Visualização da função $f(x_1, x_2, x_3)$ para cada variável.

somente na análise de sensibilidade. Por exemplo, mesmo com o comportamento verificado na figura 3.14, se a variável x_2 fosse modelada com uma distribuição menos variável, diga-se $U(x_2 - \frac{\pi}{1.000}, x_2 + \frac{\pi}{1.000})$ ou $N(x_2, \frac{\pi}{1.000})$, todos os valores de variância estariam abaixo do limite especificado.

3.2.3 Processo de otimização

Após a definição das variáveis sensíveis de projeto, inicia-se o processo de otimização. Alguns trabalhos propõem o uso da otimização clássica (equação 2.1) para a busca de um conjunto de soluções que servirá de população inicial para a otimização robusta dos parâmetros sensíveis [24, 31]. Assim, o processo de otimização é realizado em duas etapas. Na primeira etapa, as funções objetivo são avaliadas sem considerar as variáveis aleatórias envolvidas no projeto, ou seja, é realizada a otimização clássica. A segunda etapa, chamada de etapa de robustez, envolve a otimização local utilizando somente as variáveis sensíveis e uma das equações 2.14, 2.15, 2.16 ou 2.17.

Nesta dissertação, uma abordagem híbrida é apresentada, ou seja, as variáveis sensíveis e as outras

variáveis são otimizadas em uma única etapa. Assim, o algoritmo é forçado a trabalhar sempre na busca de soluções robustas.

A otimização robusta será realizada escolhendo-se uma das duas equações de minimização apresentadas na seção 2.3.4 (figura 3.15):

Método 1: Minimização de uma combinação da média e do desvio padrão:

$$\min \mathbf{E}[\mathbf{F}(x)] + (1 - \lambda)\sigma_{\mathbf{F}(x)} \quad (3.20)$$

em que $\lambda \in [0, 1]$.

Método 2: Minimização da média e minimização da variância:

$$\text{minimizar} \begin{bmatrix} \mathbf{E}[\mathbf{F}(x)] \\ \sigma_{\mathbf{F}(x)}^2 \end{bmatrix}. \quad (3.21)$$

Estes dois métodos foram escolhidos, entre os apresentados na seção 2.3.4, por serem os métodos mais utilizados na literatura. Será realizada uma comparação destes dois métodos na resolução de problemas analíticos.

A UT é utilizada para cálculo das medidas estatísticas de média $\mathbf{E}[\mathbf{F}(x)]$ e variância $\sigma_{\mathbf{F}(x)}^2$. O algoritmo de otimização robusta foi implementado utilizando NSGA-II com as modificações necessárias para incorporar o cálculo da UT e a avaliação da função objetivo.

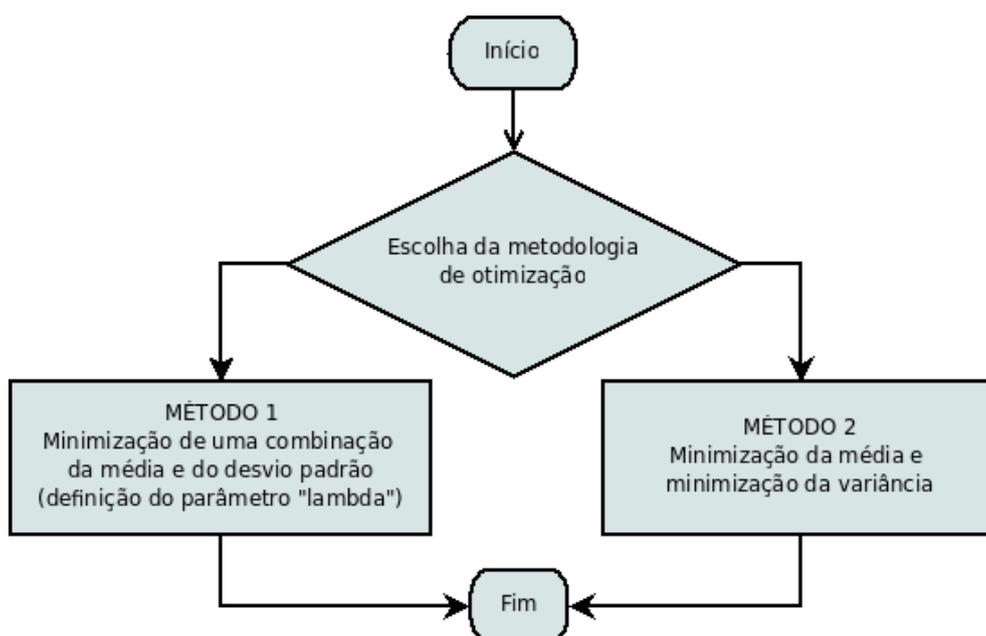


Figura 3.15: Opções para otimização robusta.

3.3 AVALIAÇÃO DA METODOLOGIA

Para introduzir o funcionamento do algoritmo de otimização robusta e da metodologia de projeto robusto apresentados serão estudados alguns problemas de otimização de funções analíticas. Os três primeiros problemas utilizam apenas a otimização robusta através do algoritmo de otimização robusta, por serem problemas simples, enquanto o quarto problema segue a metodologia de projeto robusto apresentada.

3.3.1 Problema 1

Duas funções retangulares simples e mono-objetivo são apresentadas em [4]².

Considere $f_a(x)$ definida como:

$$f_a(x) = \begin{cases} -1 : & -1 \leq x \leq 1 \\ -2 : & -1,5 \leq x \leq 1,7 \\ 0 : & \text{caso contrário} \end{cases} \quad (3.22)$$

em que $x \in [-3, 3]$.

A segunda função é apresentada para verificação da capacidade detecção de múltiplas soluções robustas. Considere $f_{a'}(x)$ definida como:

$$f_{a'}(x) = \begin{cases} -1 : & -2,5 \leq x \leq -1 \quad \text{ou} \quad 0 \leq x \leq 1,5 \\ -2 : & 2 \leq x \leq 2,2 \\ 0 : & \text{caso contrário} \end{cases} \quad (3.23)$$

em que $x \in [-3, 3]$.

Para este problema foi utilizada a otimização robusta do método 1 com $\lambda = 0$, ou seja, a otimização robusta é realizada a partir da minimização da soma do valor médio e do desvio padrão. Para as duas funções foram consideradas variáveis aleatórias gaussianas, com desvio padrão $\sigma = 0,4$ para x em $f_a(x)$ e $\sigma = 0,15$, para x em $f_{a'}(x)$. O cálculo da UT foi feito utilizando três pontos-sigma. Os resultados podem ser visto nas figuras 3.16 e 3.17.

²Em [4], os problemas apresentados são de maximização, mas é fácil perceber que ao multiplicar as funções por -1, obtém-se um problema de minimização.

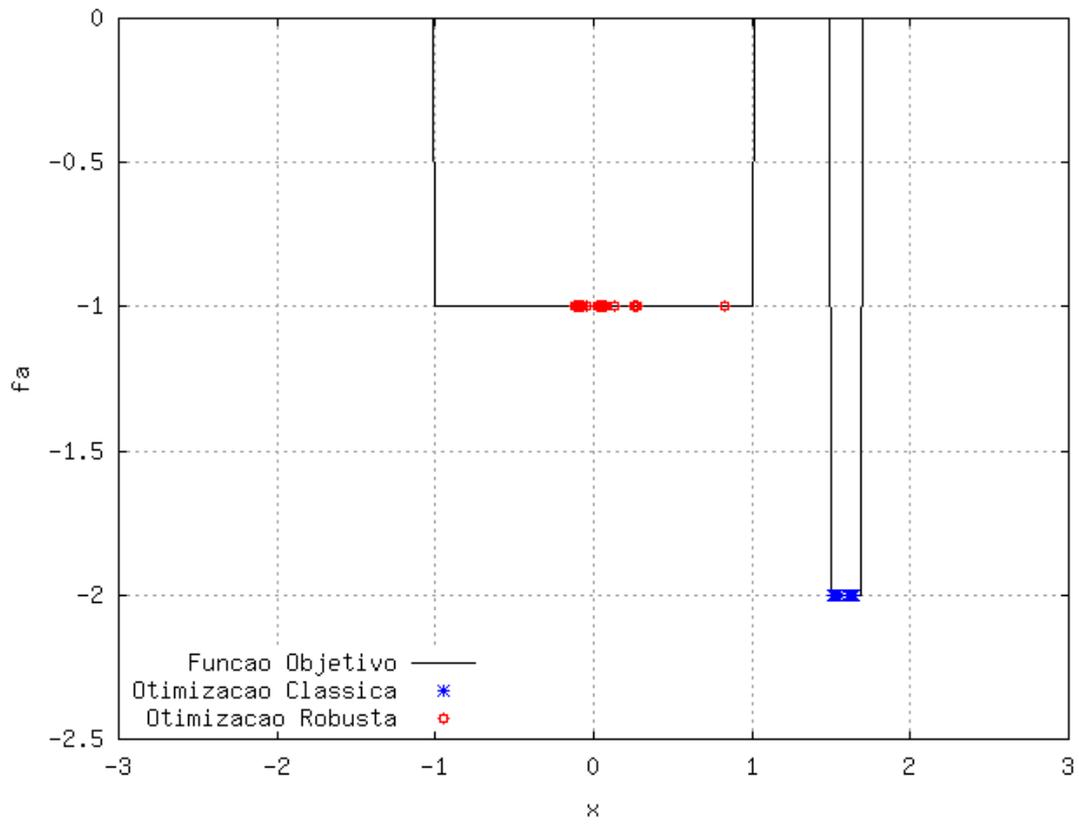


Figura 3.16: Comparação do resultado da otimização clássica com a otimização robusta.

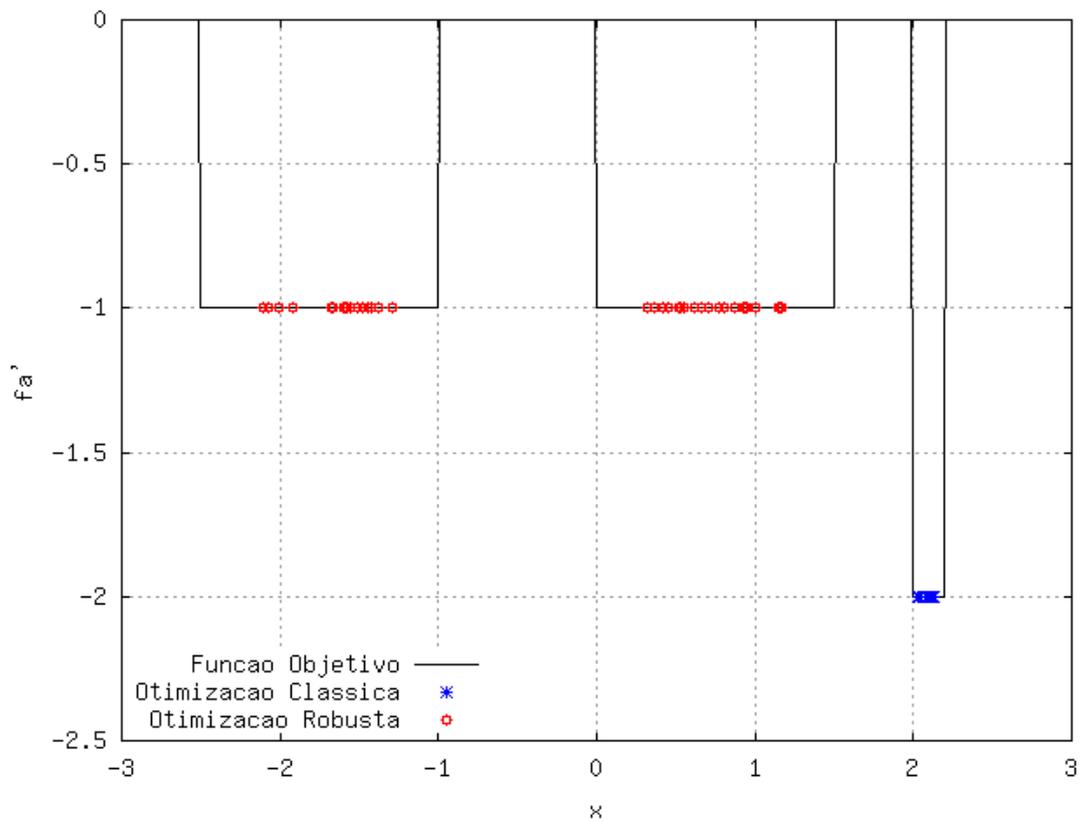


Figura 3.17: Detecção de múltiplas soluções robustas.

A figura 3.18 apresenta o resultado do artigo [4] para a detecção de múltiplas soluções robustas. Para este resultado foi utilizada a abordagem que calcula $f(x + \delta)$, descrita na seção 2.3.4.

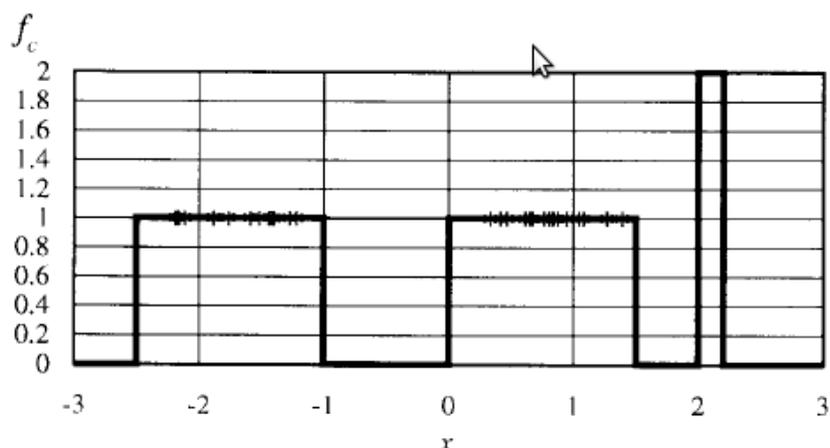


Figura 3.18: Detecção de múltiplas soluções robustas [4].

Os resultados apresentados nas figuras 3.16 e 3.17 mostram que a partir da otimização robusta é possível encontrar soluções mais estáveis. Além disto, estes resultados mostram que é possível encontrar múltiplas soluções robustas.

3.3.2 Problema 2

O segundo problema, também descrito em [4], tem como objeto de estudo a função $f_b(x)$:

$$f_b(x) = \begin{cases} -e^{-2 \ln 2 \left(\frac{x-0,1}{0,8} \right)^2} |\text{sen}(5\pi x)|^{0,5} : & 0,4 \leq x \leq 0,6 \\ -e^{-2 \ln 2 \left(\frac{x-0,1}{0,8} \right)^2} \text{sen}^6(5\pi x) : & \text{caso contrário} \end{cases} \quad (3.24)$$

em que $x \in [0, 1]$.

Esta função apresenta cinco picos, sendo que o terceiro pico apresenta maior suavidade em relação aos outros. Para esta função o valor ótimo não robusto está localizado em $x = 0, 1$, logo no primeiro pico.

Para este teste foi utilizada a otimização robusta do método 1, com $\lambda = 0$. Assim, a otimização robusta é realizada a partir da minimização da soma do valor médio e do desvio padrão. O cálculo da UT foi feito utilizando dois pontos sigma e a variável x foi modelada como uma variável gaussiana de desvio padrão $\sigma_x = 0,03$. O resultado pode ser visto na figura 3.19. Como pode ser visto, o algoritmo apresentado é capaz de encontrar a solução mais robusta para este problema.

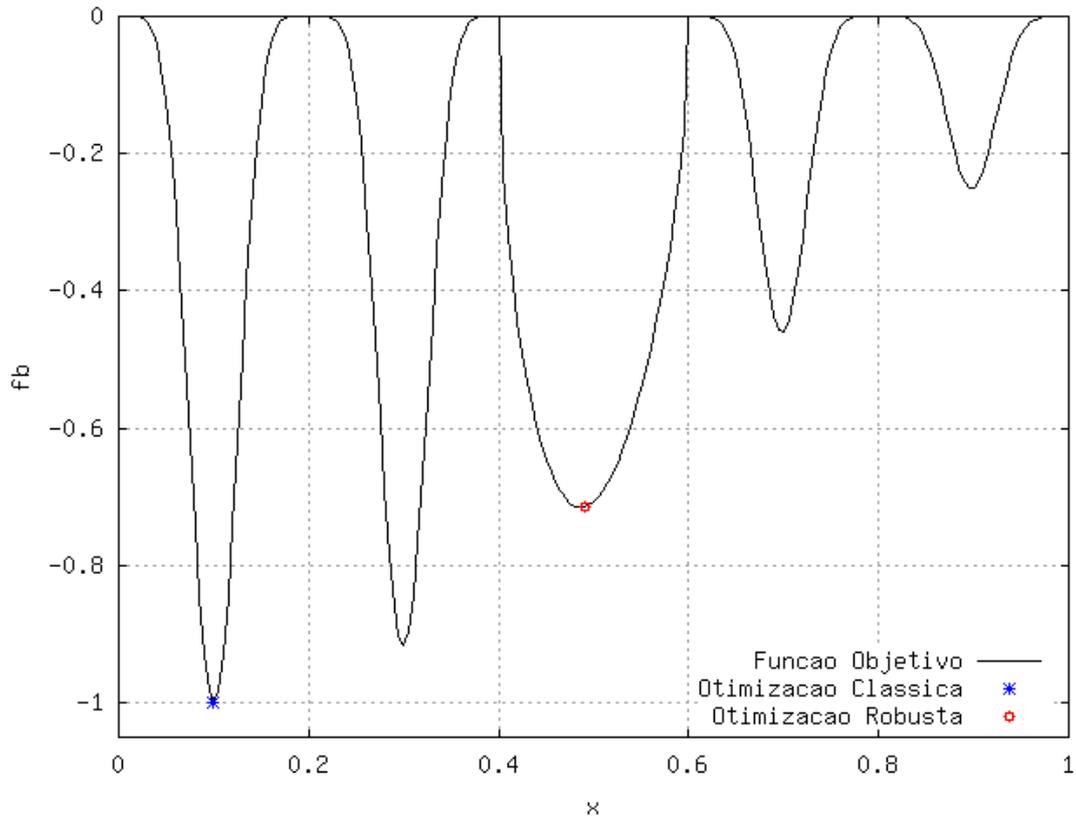


Figura 3.19: Comparação do resultado da otimização clássica com a otimização robusta.

3.3.3 Problema 3

Neste problema será abordada uma minimização multiobjetivo com uma única variável. Seja o problema de minimização de $f_1(x)$ e $f_2(x)$ descrito em [5]:

$$\begin{aligned} f_1(x) &= \exp\left(\frac{|x|}{5}\right) \cos\left(\frac{2\pi x}{|x|^{0,1}}\right) \\ f_2(x) &= 0,1(x - 0,9)(x - 2,0)(x - 2,7)(x - 3,7) \end{aligned} \quad (3.25)$$

no intervalo $-0,5 \leq x \leq 4,5$.

Os resultados encontrados em [5] estão mostrados na figura 3.20. Estes resultados foram obtidos utilizando um algoritmo denominado MOGAPSO (do inglês *Multi-Objective Genetic Algorithm Particle Swarm Optimization*). O processo de otimização foi dividido em duas partes: na primeira parte, é realizada uma busca global por soluções ótimas utilizando o MOGA; na segunda parte, os indivíduos resultantes da primeira parte são divididos em grupos e o algoritmo PSO é aplicado a cada grupo. O algoritmo PSO é bom para busca local, por isto é executado na segunda parte do processo de otimização. A robustez foi tratada pelo algoritmo como função objetivo extra, a partir da métrica apresentada na equação 2.10.

Primeiramente, pode-se analisar a sensibilidade de todo o espaço de soluções. A figura 3.21 mostra

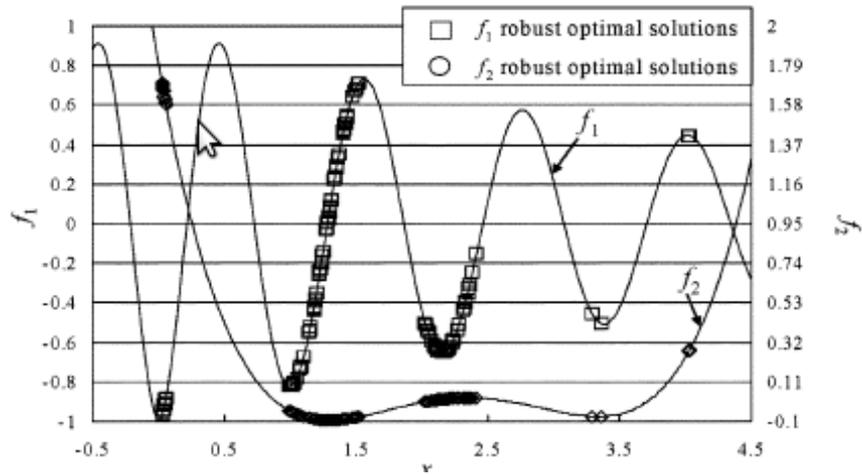


Figura 3.20: Soluções ótimas robustas obtidas pelo algoritmo MOGAPSO [5].

uma varredura aleatória do espaço de busca de soluções, sendo apresentado para cada possível solução o valor esperado e o desvio padrão para as funções objetivo $f_1(x)$ e $f_2(x)$. A incerteza (desvio padrão) associada a variável de entrada x modelada como gaussiana era $\sigma = 0, 1$ e para o cálculo da estimativa do valor esperado e do desvio padrão foi utilizada a UT com três pontos sigma.

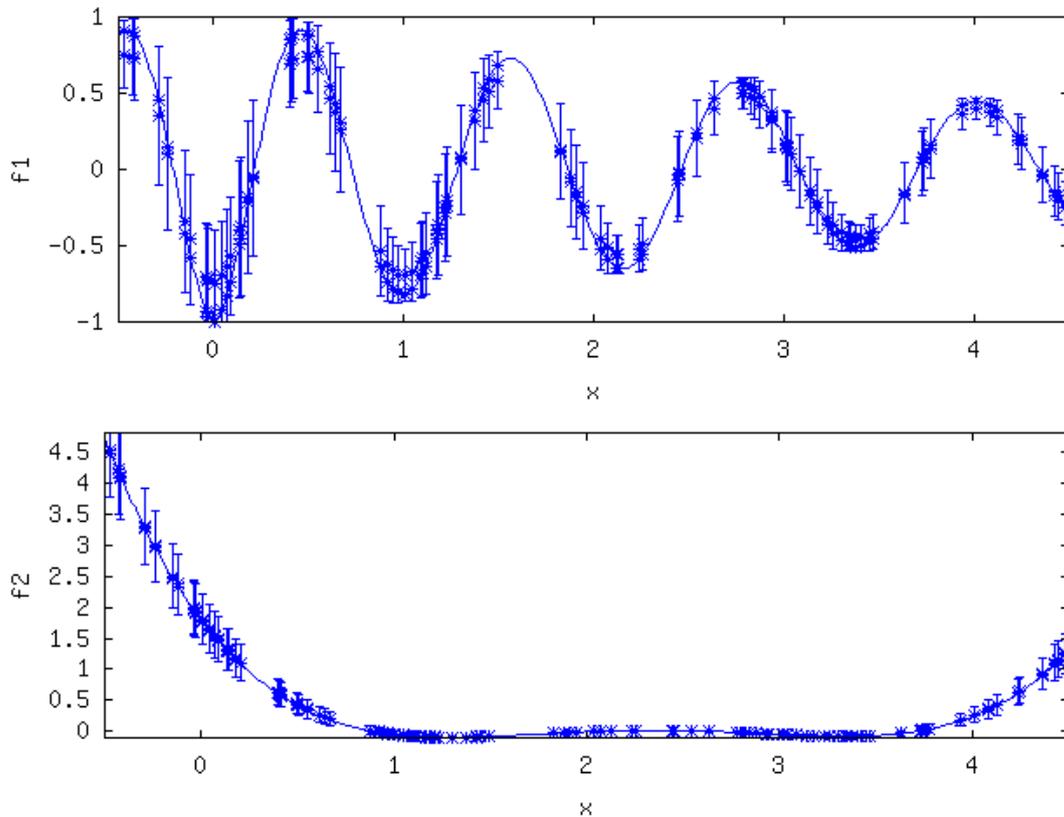


Figura 3.21: Verificação do valor esperado e do desvio padrão para uma amostragem do espaço de busca.

3.3.3.1 Análise de sensibilidade da frente ótima de Pareto

Na figura 3.22, são apresentados os resultados para a otimização clássica, utilizando o NSGA-II. Na figura, visualizam-se dois conjuntos de soluções, formando a frente ótima de Pareto.

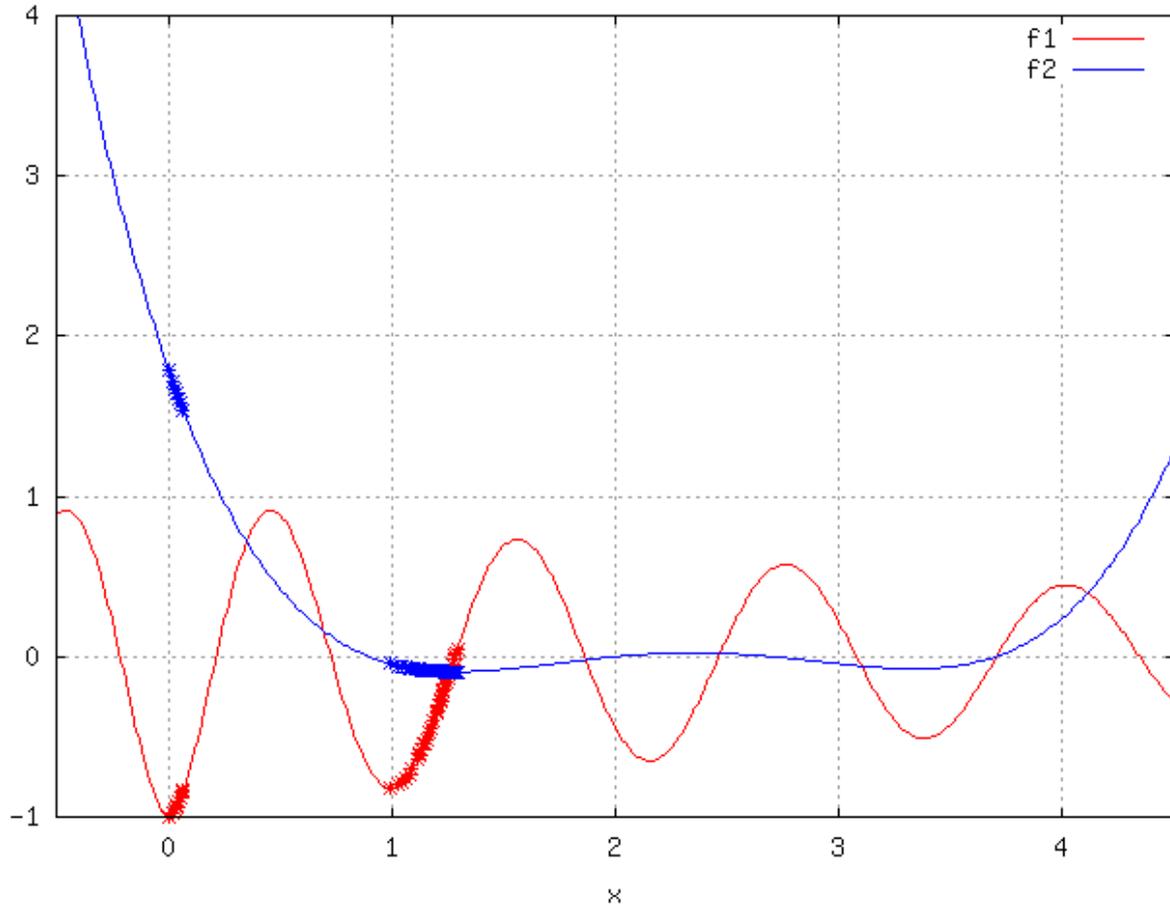


Figura 3.22: Frente ótima de Pareto dada obtida a partir do NSGA-II.

Com o objetivo de comparar a métrica B_j (equação 2.10) com a variância calculada pela UT, foi verificada quais são as soluções mais robustas dada pelos dois métodos dentro da frente ótima de Pareto dada pelo NSGA-II. Os resultados estão mostrados na figura 3.23. O NSGA-II trabalhou com uma população de 48 indivíduos durante 100 gerações, resultando em 4.800 amostras do espaço de busca. Para o cálculo de B_j foi considerada uma tolerância $\Delta = 0,1$ e para o cálculo da variância foi considerada a distribuição $U \sim (x - 0,1, x + 0,1)$. No cálculo da métrica B_j foi observado que o mínimo de amostras utilizadas para uma solução foi de 638 amostras e o máximo foi de 2.865 amostras.

Os resultados da figura 3.23 mostram que a métrica B_j e a variância concordam em quais são as soluções mais robustas para este problema.

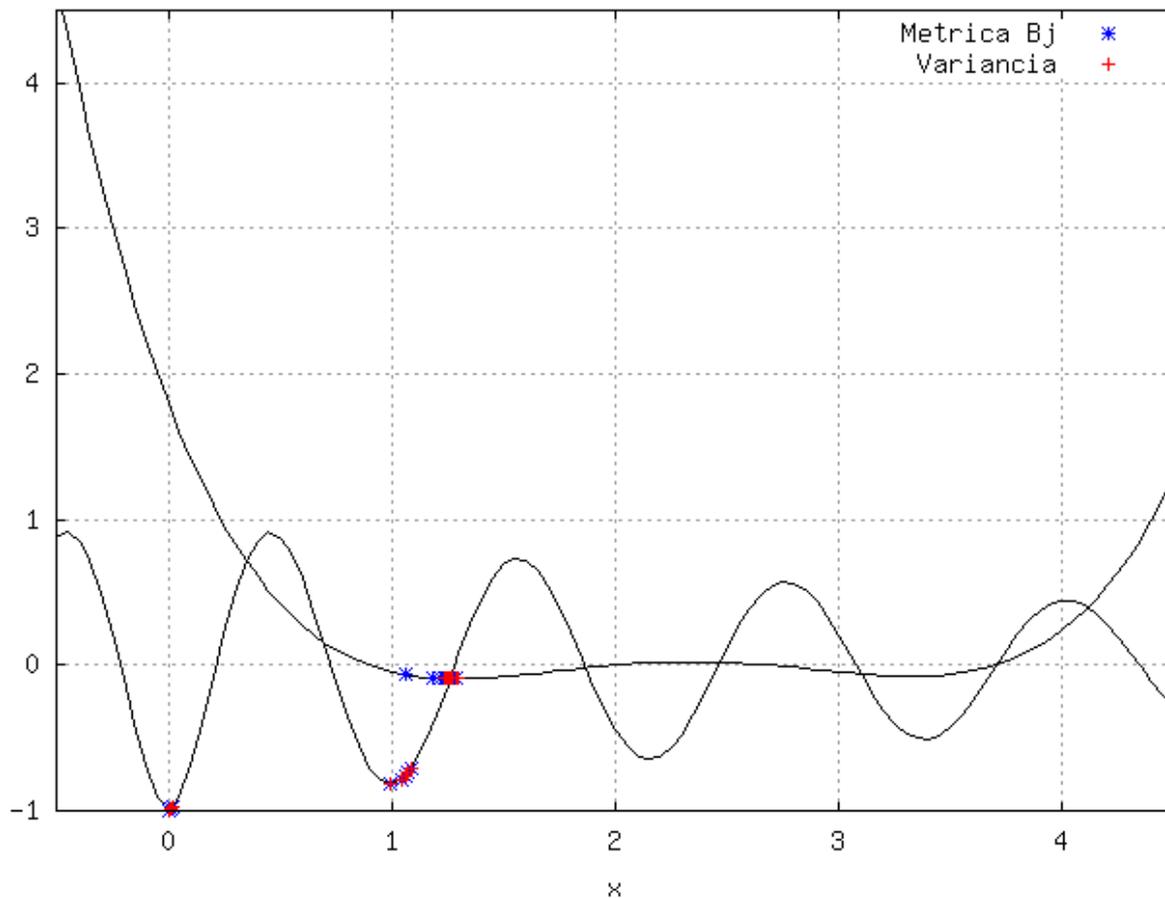


Figura 3.23: Soluções mais robustas entre as soluções ótimas de Pareto para cada função objetivo calculada pela variância e pela métrica B_j .

3.3.3.2 Otimização robusta

A seguir é apresentada a otimização robusta para diferentes níveis de incerteza. A figura 3.24 mostra os resultados da otimização robusta de acordo com o método 1, que considera a minimização de uma combinação do valor esperado e do desvio padrão (neste caso um somatório dos dois valores, dado que $\lambda = 0$). A figura 3.25 mostra os resultados da otimização robusta de acordo com o método 2, que considera a minimização do valor esperado e da variância. Já a figura 3.26 mostra os resultados pra a otimização robusta pelo método 1 para diferentes valores de λ .

Os resultados apresentados a partir da minimização do valor esperado e da variância assemelham-se aos resultados apresentados no artigo [5] (figura 3.20). É fácil perceber o porquê desta semelhança, pois em [5] a minimização robusta é feita também pelo acréscimo de mais uma função objetivo, neste caso dado pela análise das amostras geradas pelo AE seguido do cálculo de B_j (equação 2.10). Assim, ambos os resultados consideram a medida da variância no projeto robusto. O acréscimo de uma função objetivo

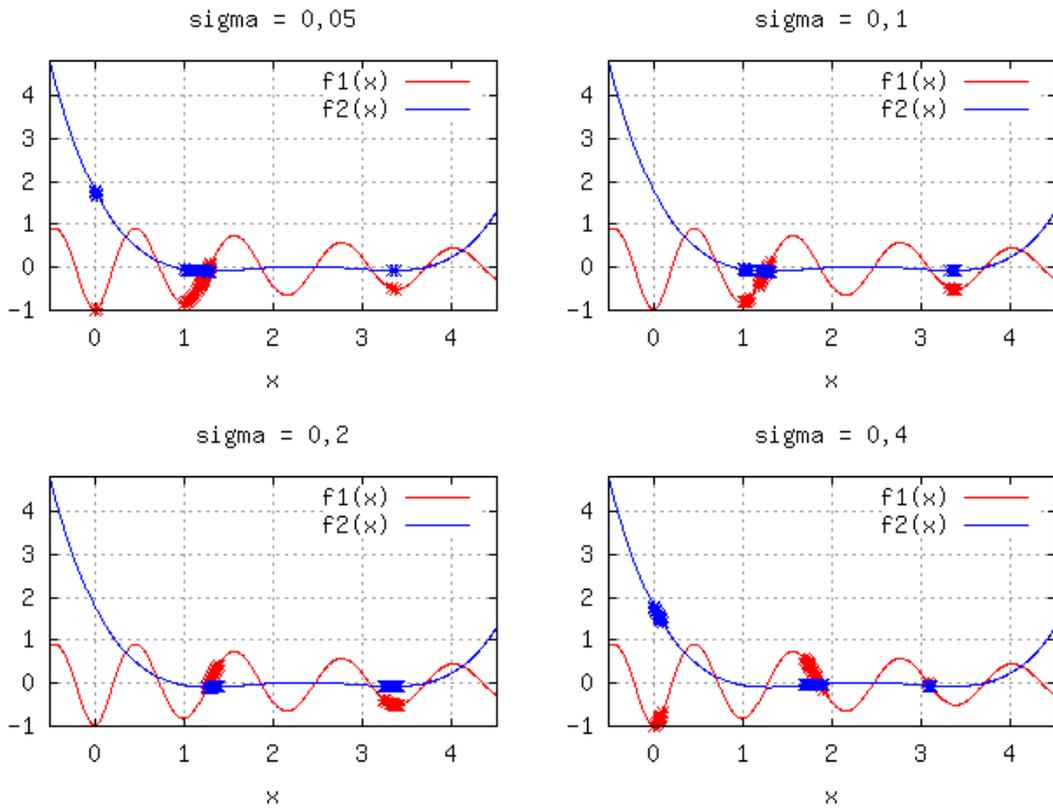


Figura 3.24: Otimização robusta pelo método 1 para diferentes níveis de σ .

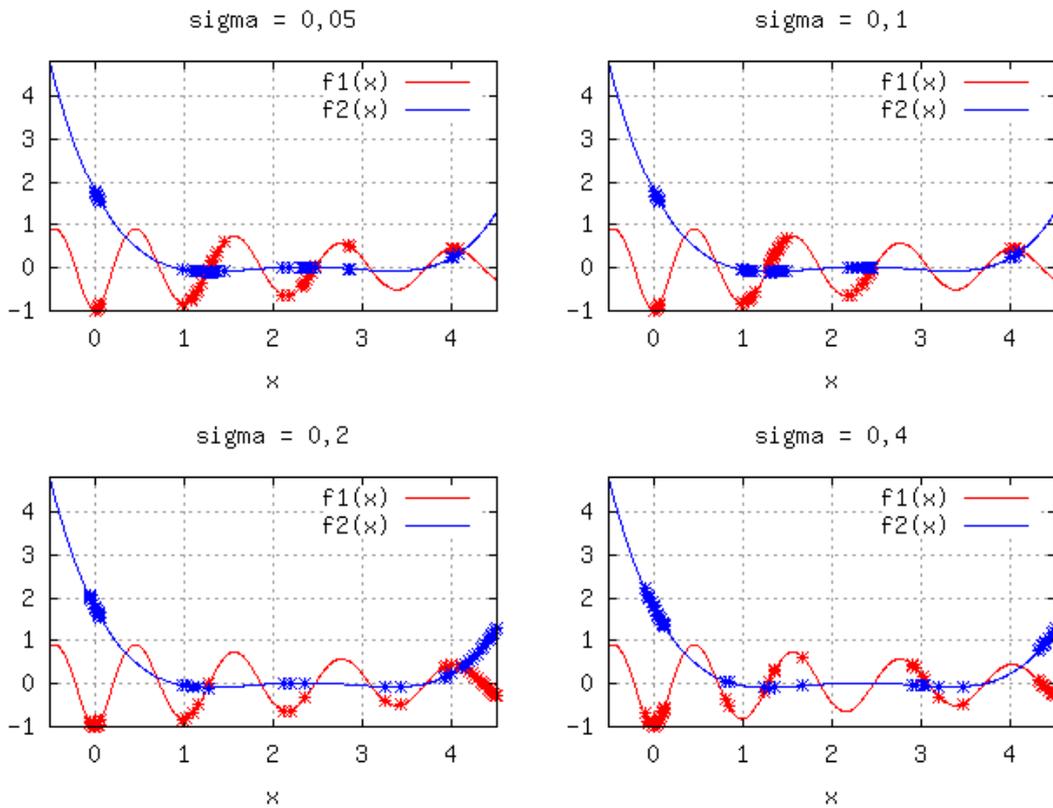


Figura 3.25: Otimização robusta pelo método 2 para diferentes níveis de σ .

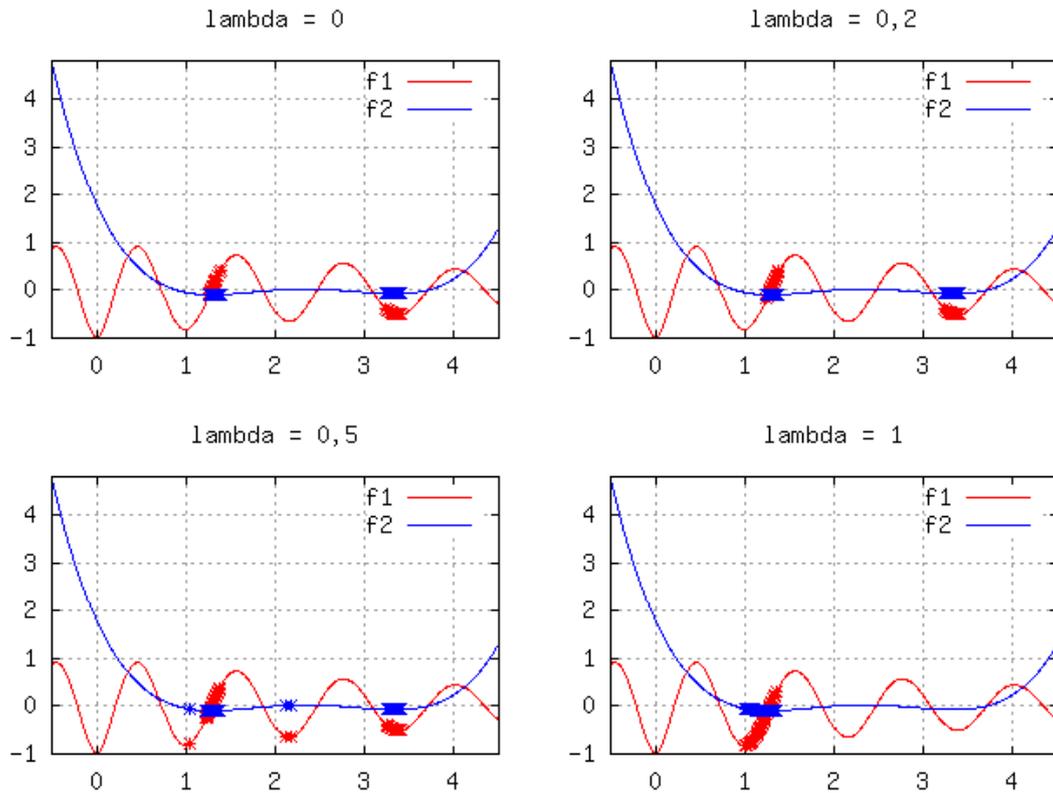


Figura 3.26: Avaliação do impacto do parâmetro λ na otimização robusta pelo método 1.

específica para o cálculo da variância, de acordo com o critério da dominância, pode levar a indivíduos com baixo nível de variância e baixo desempenho, e também a indivíduos com variância alta, com um bom desempenho. O acréscimo da função objetivo faz com que se encontre indivíduos nos extremos das funções objetivo, problema que pode ser contornado realizando uma análise de sensibilidade *a posteriori* dos indivíduos.

3.3.4 Problema 4

Neste problema será abordada uma minimização multiobjetivo e multivariável. Seja o conjunto de funções objetivo dado por:

$$\begin{aligned}
 f_1(x_1, x_2, x_3) &= \frac{1}{10} \left(\frac{1}{4} \cos(x_1^3) - \frac{x_1}{5} \sin(x_2^2) - x_1 x_3 \cos(4x_2) \log \sin(x_1)^2 \right) \\
 f_2(x_1, x_2, x_3) &= x_2 \cos(x_1) + x_1^2 + \cos(x_1^3) + \frac{x_3}{10}
 \end{aligned}
 \tag{3.26}$$

em que $0, 1 \leq x_1, x_2, x_3 \leq \pi - 0, 1$.

Para este problema será adotada a metodologia de projeto descrita na seção 3.2, caracterizando a

incerteza, verificando quais variáveis são mais sensíveis e realizando a otimização robusta. Contudo, será mostrado primeiramente a otimização clássica pelo NSGA-II.

3.3.4.1 Otimização Clássica utilizando o NSGA-II

O resultado da otimização clássica realizada pelo NSGA-II é apresentado na figura 3.27. Em azul está uma amostragem de todo o espaço de busca de soluções, enquanto a frente ótima de Pareto é apresentada em vermelho. Observando a figura, pode-se pensar que as soluções posicionadas no canto superior esquerdo são prováveis soluções não robustas, pois aparentemente estas soluções estão em pontos extremos e isolados.

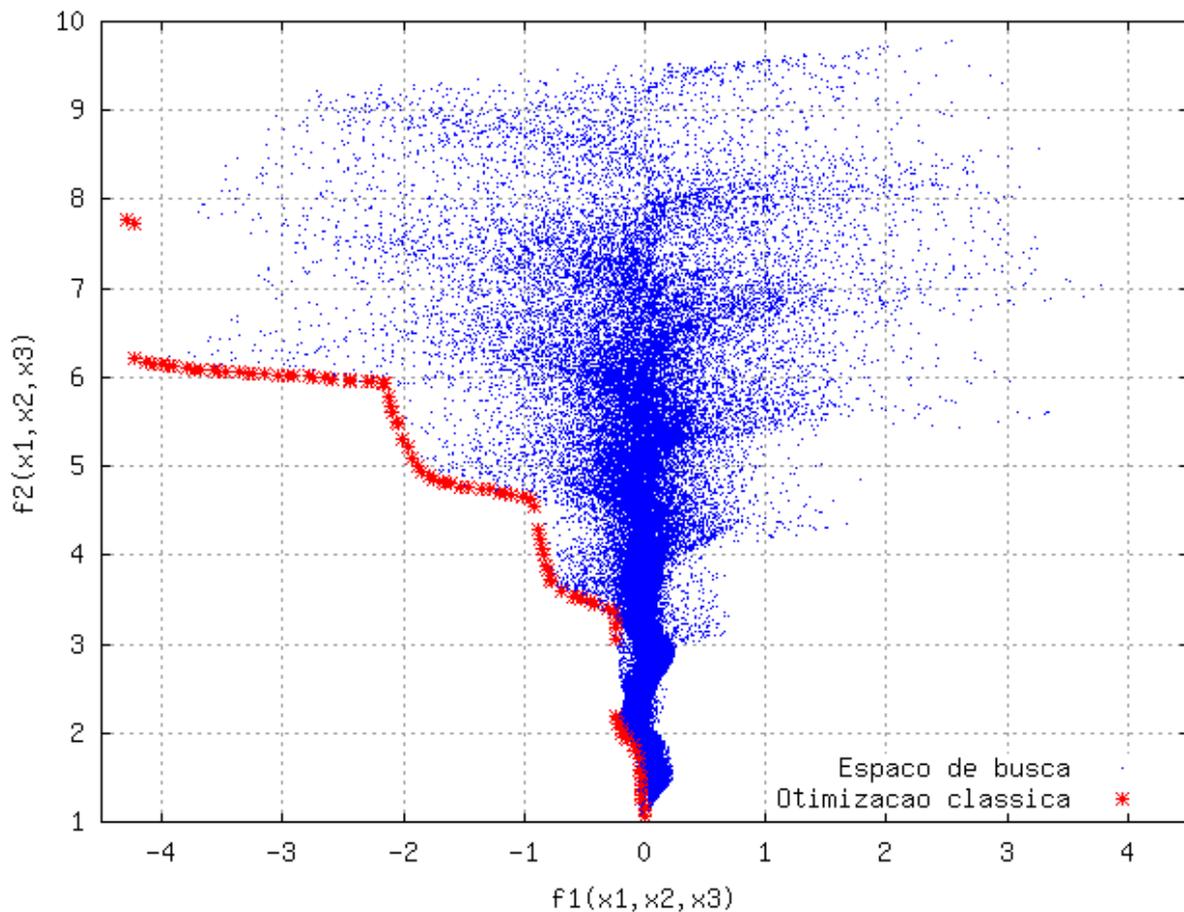


Figura 3.27: Otimização clássica pelo NSGA-II.

Os parâmetros de simulação utilizados estão mostrados na tabela 3.13.

Parâmetro	Valor
Número de Indivíduos	96
Número de Gerações	500
Probabilidade de cruzamento	1
Probabilidade de mutação	0,05
η_c	20
η_m	20

Tabela 3.13: Parâmetros de simulação do NSGA-II.

3.3.4.2 Modelagem das incertezas associadas às variáveis de entrada

Como este problema é um problema analítico, serão definidos dois cenários para estudo:

- Cenário 1:

$$\begin{aligned}
\text{Incerteza } x_1 &\sim N\left(0, \left(x_1 \frac{\pi}{30}\right)^2\right) \\
\text{Incerteza } x_2 &\sim N\left(0, \left(x_2 \frac{\pi}{200}\right)^2\right) \cdot \\
\text{Incerteza } x_3 &\sim N\left(0, \left(x_3 \frac{\pi}{100}\right)^2\right)
\end{aligned} \tag{3.27}$$

- Cenário 2:

$$\begin{aligned}
\text{Incerteza } x_1 &\sim N\left(0, \left(x_1 \frac{\pi}{30}\right)^2\right) \\
\text{Incerteza } x_2 &\sim N\left(0, \left(x_2 \frac{\pi}{30}\right)^2\right) \cdot \\
\text{Incerteza } x_3 &\sim N\left(0, \left(x_3 \frac{\pi}{50}\right)^2\right)
\end{aligned} \tag{3.28}$$

Além disso, será definido que a variância desejada para as funções objetivo é de, no máximo, $\frac{\pi}{100}$.

3.3.4.3 Definição dos parâmetros sensíveis

As figuras 3.28 e 3.29 mostram o comportamento de cada variável para as duas funções objetivo. Visualmente, percebe-se em relação a $f_1(x_1, x_2, x_3)$: a função $f_1(x_1, x_2^{cte}, x_3^{cte})$ apresenta um valor constante até $\frac{2\pi}{3}$ quando inicia um crescimento ou decrescimento quase exponencial, a função $f_1(x_1^{cte}, x_2^{cte}, x_3)$ apresenta um comportamento linear e $f_1(x_1^{cte}, x_2, x_3^{cte})$ apresenta uma oscilação. Em relação a $f_2(x_1, x_2, x_3)$: a função $f_2(x_1, x_2^{cte}, x_3^{cte})$ apresenta um valor bastante oscilatório e crescente, a função $f_2(x_1^{cte}, x_2^{cte}, x_3)$ apresenta um comportamento crescente linear e a função $f_2(x_1^{cte}, x_2, x_3^{cte})$ apresenta comportamento decrescente ou crescente linear.

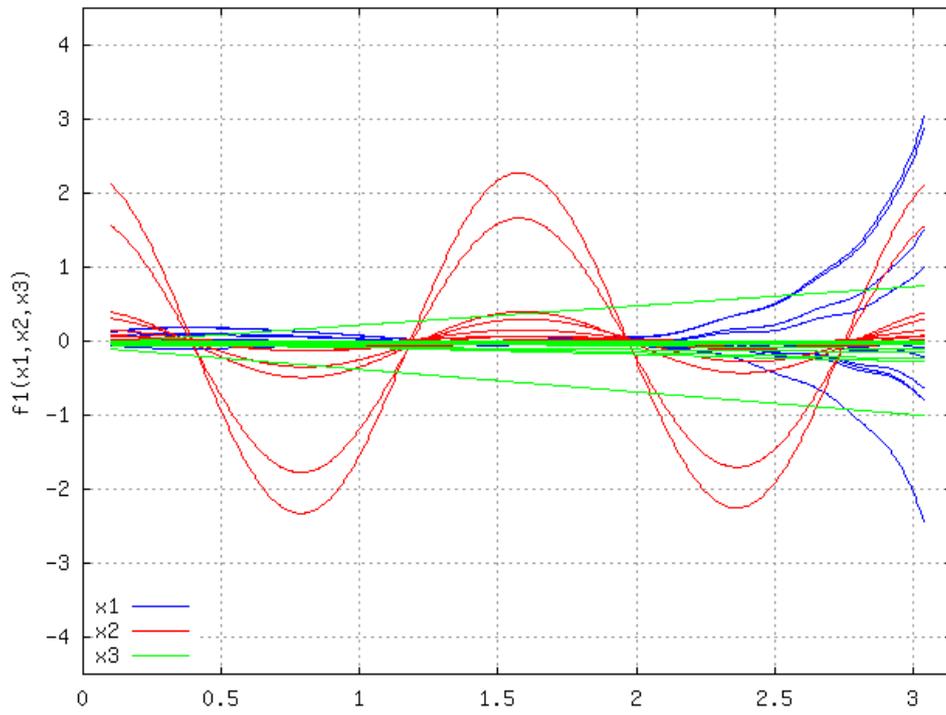


Figura 3.28: Verificação do comportamento de $f_1(x_1, x_2, x_3)$ para cada variável.

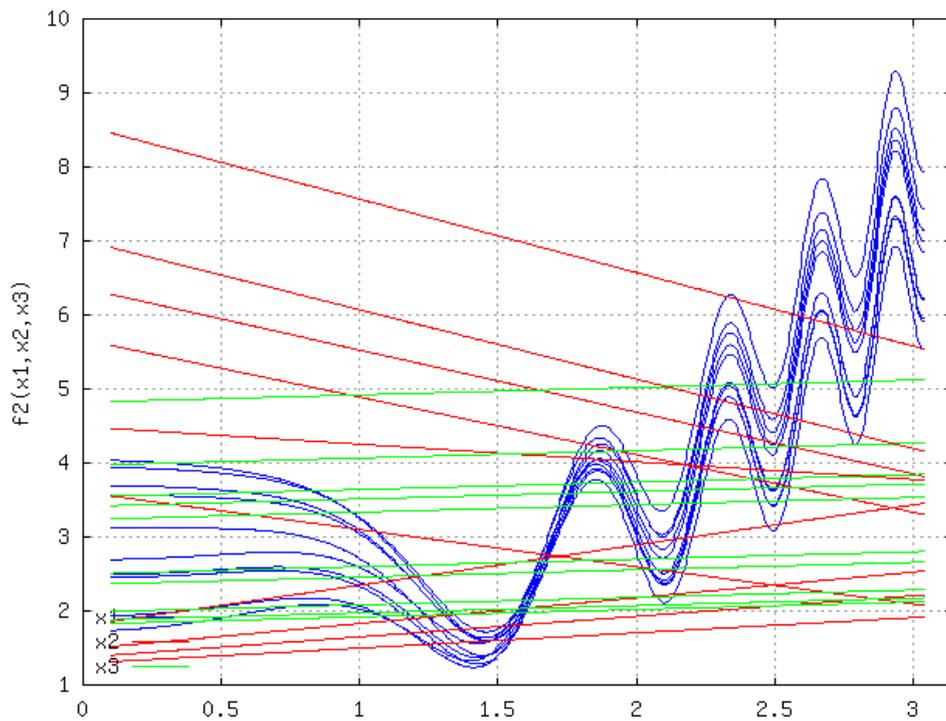


Figura 3.29: Verificação do comportamento de $f_2(x_1, x_2, x_3)$ para cada variável.

Mesmo com esta análise visual, deve-se lembrar que a verificação das variáveis sensíveis de projeto depende fortemente de como estas são modeladas. Desta forma, para os cenários especificados na seção 3.3.4.2 foi realizada a análise de sensibilidade para verificação da variância das funções objetivo. Os resultados estão mostrados nas tabelas 3.14, 3.15, 3.16 e 3.17.

Para o cenário 1, considerando a função $f_1(x_1, x_2, x_3)$, as variáveis aleatórias x_1 e x_2 apresentam valores de variância abaixo do especificado, mas a variável x_1 apresenta um valor médio de variância bem maior que a variável x_2 . Neste mesmo cenário, analisando a função $f_2(x_1, x_2, x_3)$, apenas x_1 apresenta sensibilidade. Para este cenário, pode-se considerar x_1 como a variável sensível de projeto. Para o cenário 2, tanto a variável x_1 quanto a variável x_2 apresenta sensibilidade, com valores médio de variância próximos para a função $f_1(x_1, x_2, x_3)$. Assim, estas duas variáveis serão consideradas como variáveis sensíveis para o projeto robusto.

Outra análise que pode ser realizada é a verificação da sensibilidade na frente ótima de Pareto. Isto pode ser utilizado principalmente em problemas que a avaliação da função objetivo é custosa. A realização da análise de sensibilidade apenas na frente ótima de Pareto (obtida pela otimização clássica) não apresenta grande desvantagem quando comparada a análise de sensibilidade em todo o espaço de busca. Isto porque deseja-se que as soluções robustas estejam próximas a frente ótima de Pareto, sendo necessário avaliar apenas qual impacto das incertezas na frente ótima de Pareto.

Os resultados da AS da frente ótima de Pareto obtida através da otimização clássica para o problema em estudo estão mostrados nas tabelas 3.18, 3.19, 3.20 e 3.20. Os resultados se assemelham aos obtidos anteriormente. No cenário 1, a variável x_1 apresenta um valor significativamente superior de valor médio de variância nas duas funções, confirmando ser esta a variável mais sensível de projeto. No cenário 2, para variações nas variáveis x_1 e x_2 , várias soluções apresentam valor de variância abaixo do especificado em projeto, sendo estas as variáveis mais sensíveis.

Como pode ser visto na apresentação de dois cenários, a sensibilidade de uma variável depende fortemente de como ela é modelada.

Cenário 1			
Variável	$\max \sigma_{f_1}^2$	$E[\sigma_{f_1}^2]$	$\sigma_{f_1}^2 \leq \frac{\pi}{100}$
x_1	8,4828	0,038858	8.985
x_2	0,33946	0,0020616	9.840
x_3	0,013138	$1,469010^{-4}$	10.000

Tabela 3.14: Análise de sensibilidade para a função $f_1(x_1, x_2, x_3)$ de acordo com o cenário 1.

Cenário 1			
Variável	$\max \sigma_{f_2}^2$	$E[\sigma_{f_2}^2]$	$\sigma_{f_2}^2 \leq \frac{\pi}{100}$
x_1	1,5978	0,26007	4.256
x_2	0,0022299	$3,623910^{-4}$	10.000
x_3	$9,122410^{-5}$	$3,172910^{-5}$	10.000

Tabela 3.15: Análise de sensibilidade para a função $f_2(x_1, x_2, x_3)$ de acordo com o cenário 1.

Cenário 2			
Variável	$\max \sigma_{f_1}^2$	$E[\sigma_{f_1}^2]$	$\sigma_{f_1}^2 \leq \frac{\pi}{10}$
x_1	9,5560	0,035540	9.036
x_2	4,6298	0,053847	8.670
x_3	0,063437	$5,383510^{-4}$	9.990

Tabela 3.16: Análise de sensibilidade para a função $f_1(x_1, x_2, x_3)$ de acordo com o cenário 2.

Cenário 2			
Variável	$\max \sigma_{f_2}^2$	$E[\sigma_{f_2}^2]$	$\sigma_{f_2}^2 \leq \frac{\pi}{10}$
x_1	1,5951	0,26364	4.189
x_2	0,099180	0,016742	8.004
x_3	$3,652210^{-4}$	$1,242110^{-4}$	10.000

Tabela 3.17: Análise de sensibilidade para a função $f_2(x_1, x_2, x_3)$ de acordo com o cenário 2.

Cenário 1			
Variável	$\max \sigma_{f_1}^2$	$E[\sigma_{f_1}^2]$	$\sigma_{f_1}^2 \leq \frac{\pi}{100}$
x_1	2,0666	0,78331	12
x_2	0,24133	0,041717	65
x_3	0,017939	0,0045698	96

Tabela 3.18: Análise de sensibilidade da frente ótima de Pareto para a função $f_1(x_1, x_2, x_3)$ de acordo com o cenário 1.

Cenário 1			
Variável	$\max \sigma_{f_2}^2$	$E[\sigma_{f_2}^2]$	$\sigma_{f_2}^2 \leq \frac{\pi}{100}$
x_1	3,9859	2,4372	6
x_2	0,0016138	0,0010925	96
x_3	$9,130710^{-5}$	8,0133e-05	96

Tabela 3.19: Análise de sensibilidade da frente ótima de Pareto para a função $f_2(x_1, x_2, x_3)$ de acordo com o cenário 1.

Cenário 2			
Variável	$\max \sigma_{f_1}^2$	$E[\sigma_{f_1}^2]$	$\sigma_{f_1}^2 \leq \frac{\pi}{100}$
x_1	2,0667	0,78974	14
x_2	6,0778	2,0005	19
x_3	0,071759	0,020125	70

Tabela 3.20: Análise de sensibilidade da frente ótima de Pareto para a função $f_1(x_1, x_2, x_3)$ de acordo com o cenário 2.

Cenário 2			
Variável	$\max \sigma_{f_2}^2$	$E[\sigma_{f_2}^2]$	$\sigma_{f_2}^2 \leq \frac{\pi}{100}$
x_1	4,0095	2,4689	4
x_2	0,072985	0,046691	23
x_3	$3,652310^{-4}$	$3,118010^{-4}$	96

Tabela 3.21: Análise de sensibilidade da frente ótima de Pareto para a função $f_2(x_1, x_2, x_3)$ de acordo com o cenário 2.

3.3.4.4 Cenário 1 - otimização robusta

Neste cenário, será realizado o processo de otimização híbrido, ou seja a variável x_1 será otimizada de forma robusta enquanto as outras variáveis serão otimizadas de forma clássica. Isto é implementado da seguinte forma: a UT é calculada para as funções $f_1(x_1, x_2, x_3)$ e $f_2(x_1, x_2, x_3)$ considerando apenas a variável x_1 , ou seja, as outras variáveis são avaliadas pelo seu valor exato. Os valores de média e variância obtidos são utilizados nos métodos de otimização robusta mostrados na figura 3.15.

A figura 3.30 mostra os resultados para a otimização robusta. É importante notar que nesta figura também está presente a frente ótima de Pareto dada pela otimização clássica, contudo esta frente foi em parte sobreposta pelos resultados da otimização robusta. Assim, percebe-se que a otimização robusta é capaz de selecionar dentro da frente de Pareto dada pelo NSGA-II as soluções mais robustas, e também de procurar outras soluções robustas dentro do espaço de busca das variáveis.

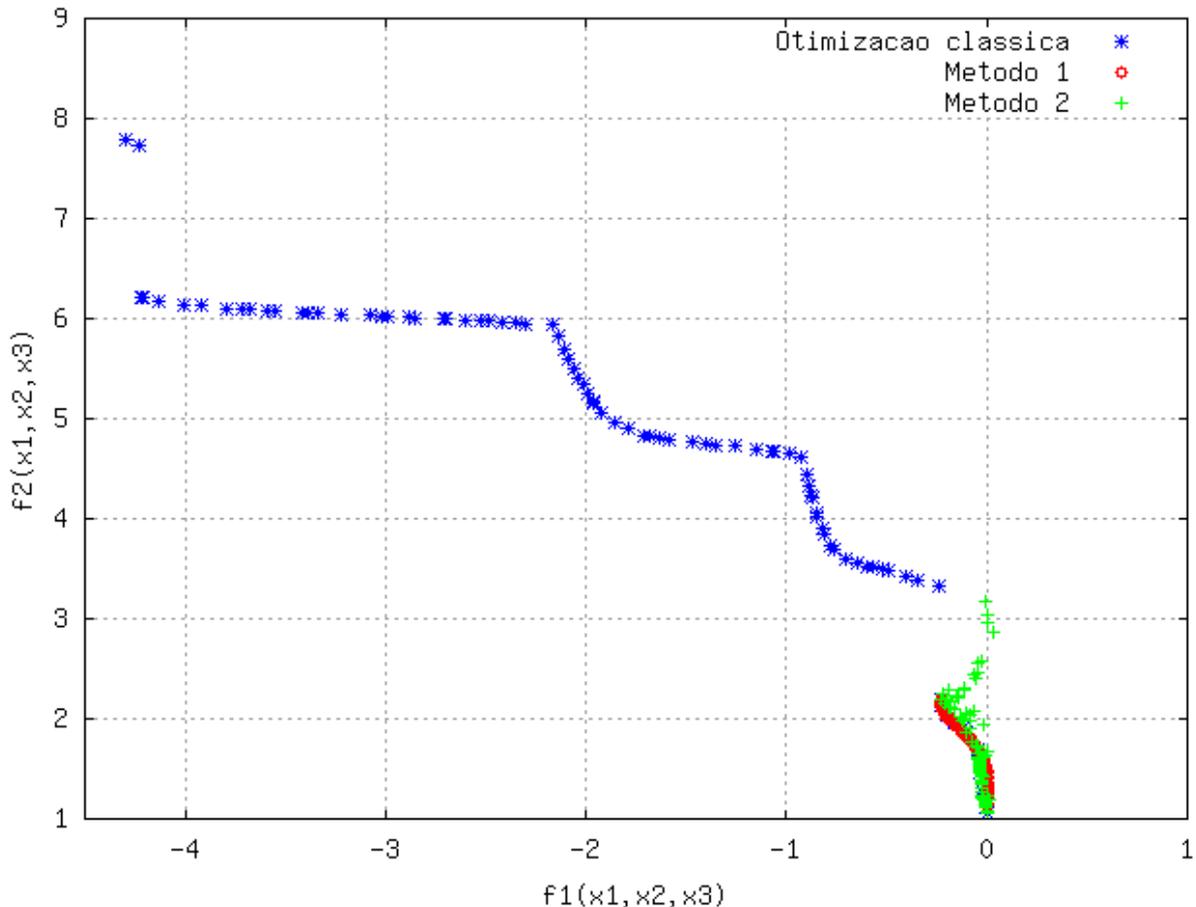


Figura 3.30: Resultado da otimização para o cenário 1.

Também foi realizada uma análise da variância para as otimizações realizadas. O resultado está mostrado nas tabelas 3.22 e 3.23.

Variável	$\max \sigma_{f_1}^2$	$E[\sigma_{f_1}^2]$	$\sigma_{f_1}^2 \leq \frac{\pi}{100}$
Otimização Clássica	2,0667	0,77257	15
Método 1	$4,3525 \cdot 10^{-5}$	$1,1568 \cdot 10^{-5}$	96
Método 2	$4,6963 \cdot 10^{-4}$	$1,1424 \cdot 10^{-4}$	96

Tabela 3.22: Análise das soluções ótimas para a função $f_1(x_1, x_2, x_3)$ de acordo com o cenário 1.

Variável	$\max \sigma_{f_2}^2$	$E[\sigma_{f_2}^2]$	$\sigma_{f_2}^2 \leq \frac{\pi}{100}$
Otimização Clássica	4,0052	2,4561	7
Método 1	$3,8559 \cdot 10^{-4}$	$7,7818 \cdot 10^{-5}$	96
Método 2	0,2303	0,0623	41

Tabela 3.23: Análise das soluções ótimas para a função $f_2(x_1, x_2, x_3)$ de acordo com o cenário 1.

As figuras de 3.31 até 3.36 mostram o comportamento do desvio padrão associado a cada solução para cada método de otimização. Nestas figuras, em azul é mostrado o valor médio e o desvio padrão associados à cada solução. Este cálculo foi realizado pela UT. Em vermelho, é mostrado o desvio padrão aceitável para cada solução, em volta da própria solução (mostrada em * azul).

Como pode ser observado, a otimização com o clássica com o NSGA-II apresenta inúmeras soluções com desvio padrão superior ao limite especificado durante a etapa de modelagem da incerteza. Na escolha do método 1 de otimização robusta, os valores de desvio padrão estão bem inferior aos valores especificados. Para o método 2, algumas soluções estão abaixo apresentam desvio padrão satisfatório enquanto outras extrapolam o desvio padrão aceitável.

Outro ponto que deve ser analisado diz respeito aos limites impostos às variáveis de entrada. Para este problema, as variáveis de entrada estão no intervalo $0, 1 \leq x_1, x_2, x_3 \leq \pi - 0, 1$. Na presença de incertezas, estas variáveis podem ultrapassar este limite. O algoritmo de otimização robusta deve garantir que as variáveis de entrada sejam realizáveis mesmo na presença de incertezas, ou seja, a variação delas deve estar contida no intervalo especificado. Neste sentido, para que uma solução seja considerada realizável, foi considerada a seguinte condição:

$$\begin{aligned}
 x_i - \sigma_{x_i} x_i \sqrt{3} &\geq 0, 1 & \text{e} \\
 x_i + \sigma_{x_i} x_i \sqrt{3} &\leq \pi - 0, 1
 \end{aligned}
 \tag{3.29}$$

Para a otimização clássica com o NSGA-II, 58 soluções são consideradas irrealizáveis de acordo com

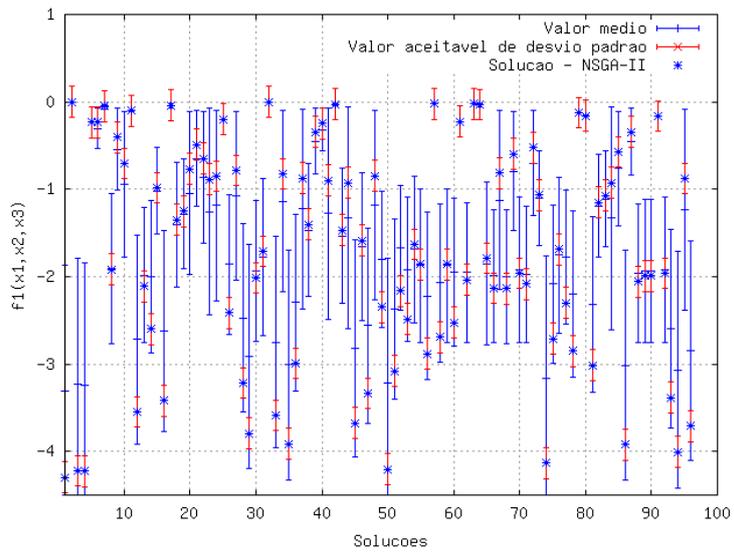


Figura 3.31: Análise de sensibilidade das soluções ótimas não robustas.

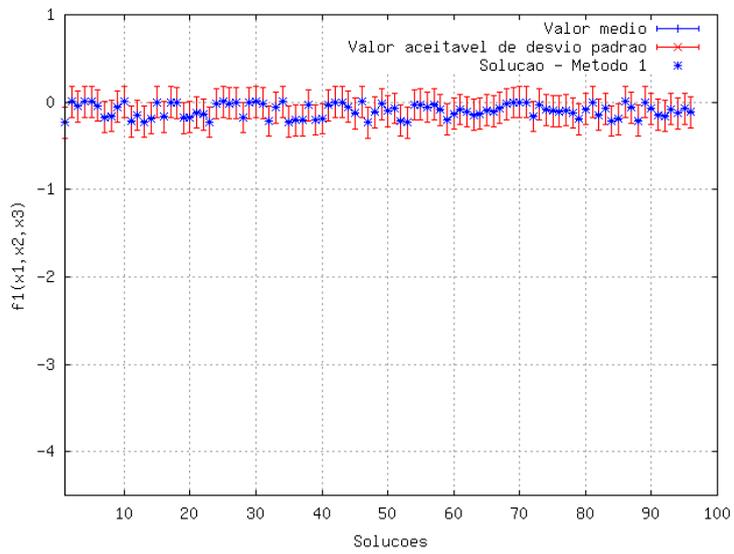


Figura 3.32: Análise de sensibilidade das soluções ótimas robustas dadas pelo método 1.

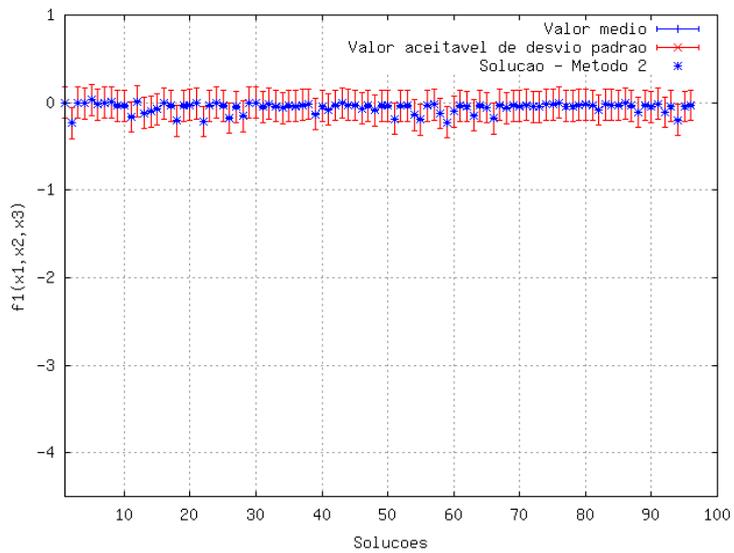


Figura 3.33: Análise de sensibilidade das soluções ótimas robustas dadas pelo método 2.

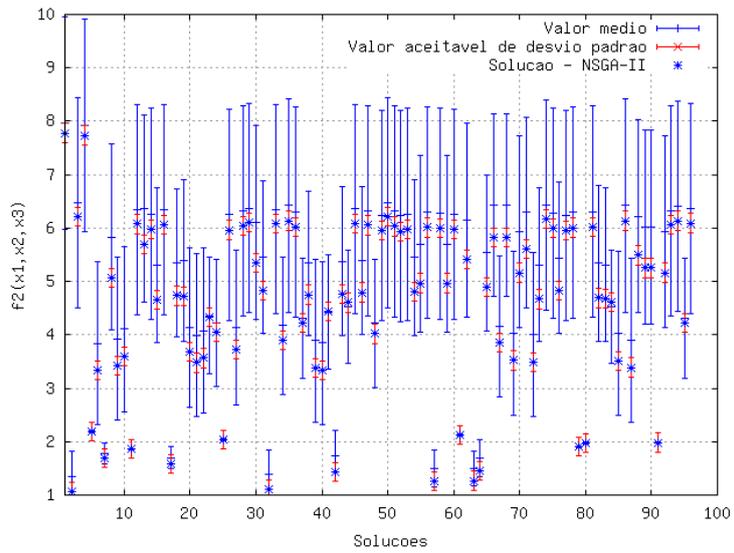


Figura 3.34: Análise de sensibilidade das soluções ótimas não robustas.

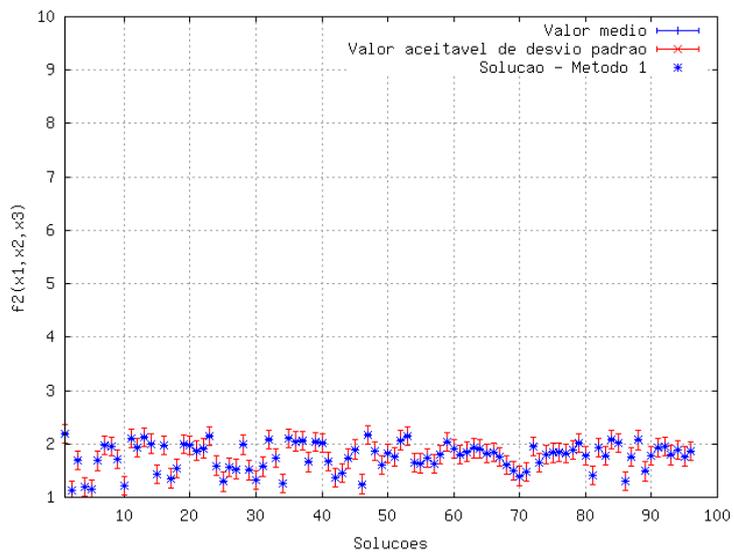


Figura 3.35: Análise de sensibilidade das soluções ótimas robustas dadas pelo método 1.

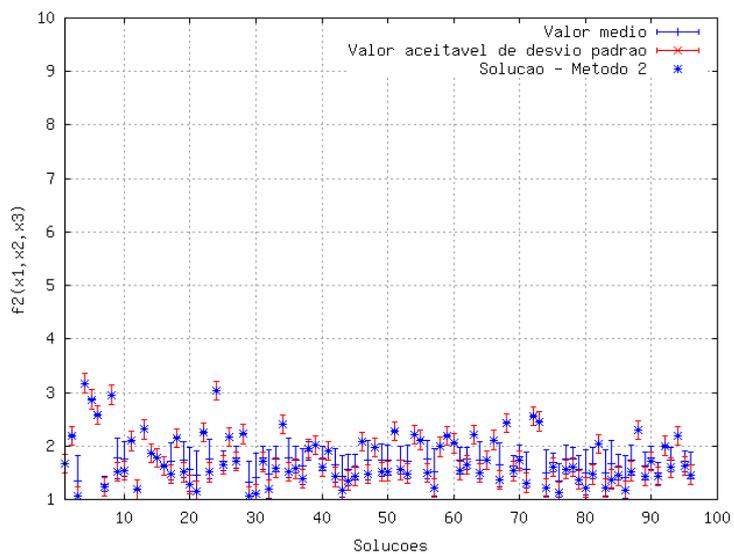


Figura 3.36: Análise de sensibilidade das soluções ótimas robustas dadas pelo método 2.

os limites estabelecidos na equação 3.29. Para a otimização robusta, esta condição já foi incorporada ao algoritmo de otimização. Então, todas as soluções dadas pelo método 1 e pelo método 2 são realizáveis de acordo com a equação 3.29.

Na equação que descreve o método 1 (equação 2.15) há um parâmetro λ que ajusta o grau de robustez desejado para as soluções. Para este problema também foi realizada uma análise sobre o impacto do valor de λ na definição das soluções robustas. O resultado está mostrado nas tabelas 3.24 e 3.25 e nas figuras 3.37 e 3.38.

Variável	$\max \sigma_{f_1}^2$	$E[\sigma_{f_1}^2]$	$\sigma_{f_1}^2 \leq \frac{\pi}{100}$
$\lambda = 0$	$4,777810^{-5}$	$1,174110^{-5}$	96
$\lambda = 0,25$	1,6432	0,40162	58
$\lambda = 0,5$	1,5032	0,46345	49
$\lambda = 0,75$	1,7176	0,53682	50
$\lambda = 1$	1,6908	0,47009	53

Tabela 3.24: Análise das simulações para a função $f_1(x_1, x_2, x_3)$ de acordo com o cenário 1.

Variável	$\max \sigma_{f_2}^2$	$E[\sigma_{f_2}^2]$	$\sigma_{f_2}^2 \leq \frac{\pi}{100}$
$\lambda = 0$	$4,258210^{-4}$	$7,981410^{-5}$	96
$\lambda = 0,25$	2,1456	0,59262	58
$\lambda = 0,5$	1,9365	0,93827	39
$\lambda = 0,75$	1,8736	1,0059	31
$\lambda = 1$	2,1119	1,0851	32

Tabela 3.25: Análise das simulações para a função $f_2(x_1, x_2, x_3)$ de acordo com o cenário 1.

De acordo com a equação 2.15, quanto menor o valor de λ mais robustas serão as soluções. Nas tabelas 3.24 e 3.25, é possível verificar um crescimento do valor médio de variância e uma diminuição no número de soluções com variância abaixo de $\frac{\pi}{100}$ a medida que o valor de λ aumenta. A figura 3.38 apresenta os resultados da otimização clássica e da otimização pelo valor médio ($\lambda = 1$). Como pode ser observado, algumas soluções clássicas compõem a frente de Pareto desta otimização robusta.

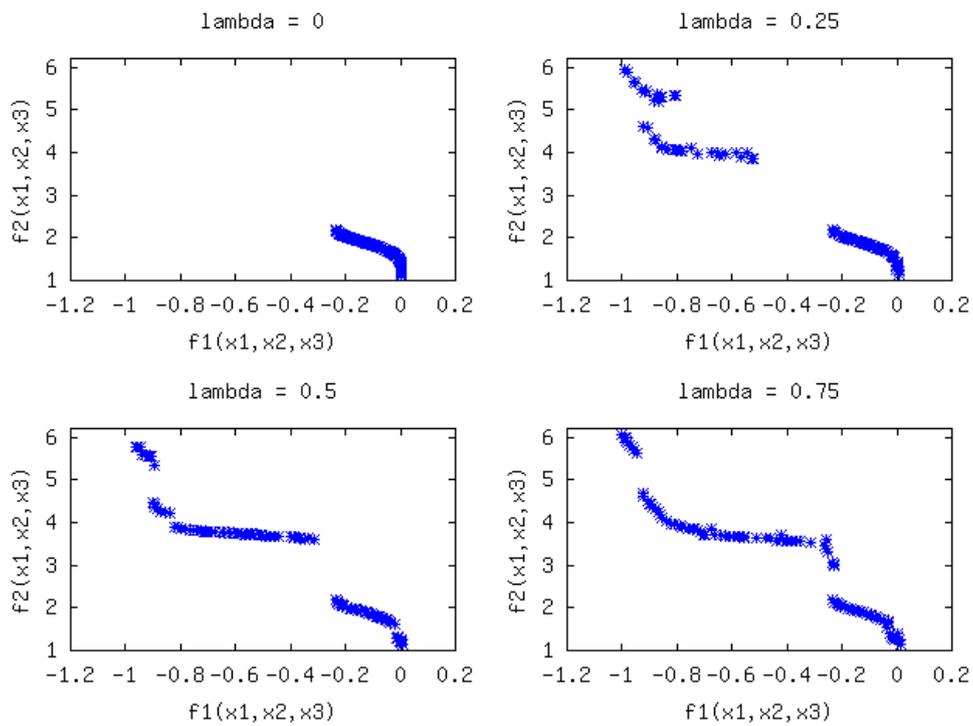


Figura 3.37: Avaliação do impacto do parâmetro λ na otimização robusta pelo método 1 para o cenário 1.

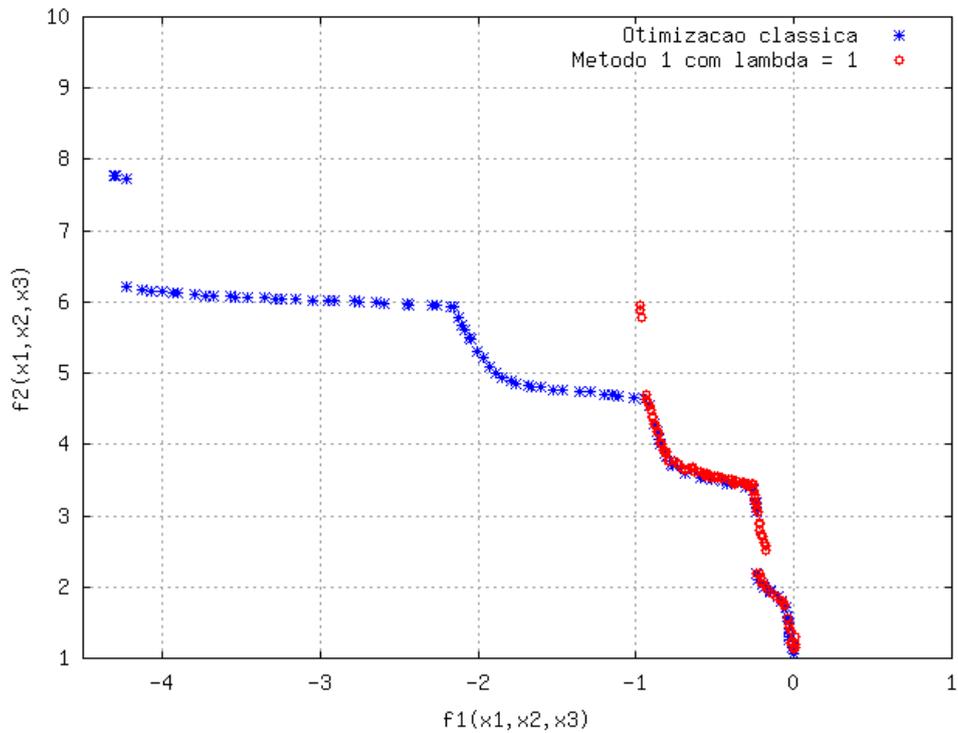


Figura 3.38: Avaliação do impacto do parâmetro λ na otimização robusta pelo método 1 para o cenário 1.

3.3.4.5 Cenário 2 - otimização robusta

Neste cenário, a otimização das variáveis x_1 e x_2 é realizada utilizando a UT para cálculo das medidas estatísticas. Os dois métodos de otimização robusta mostrados na figura 3.15 foram considerados. O resultado está mostrado na figura 3.39.

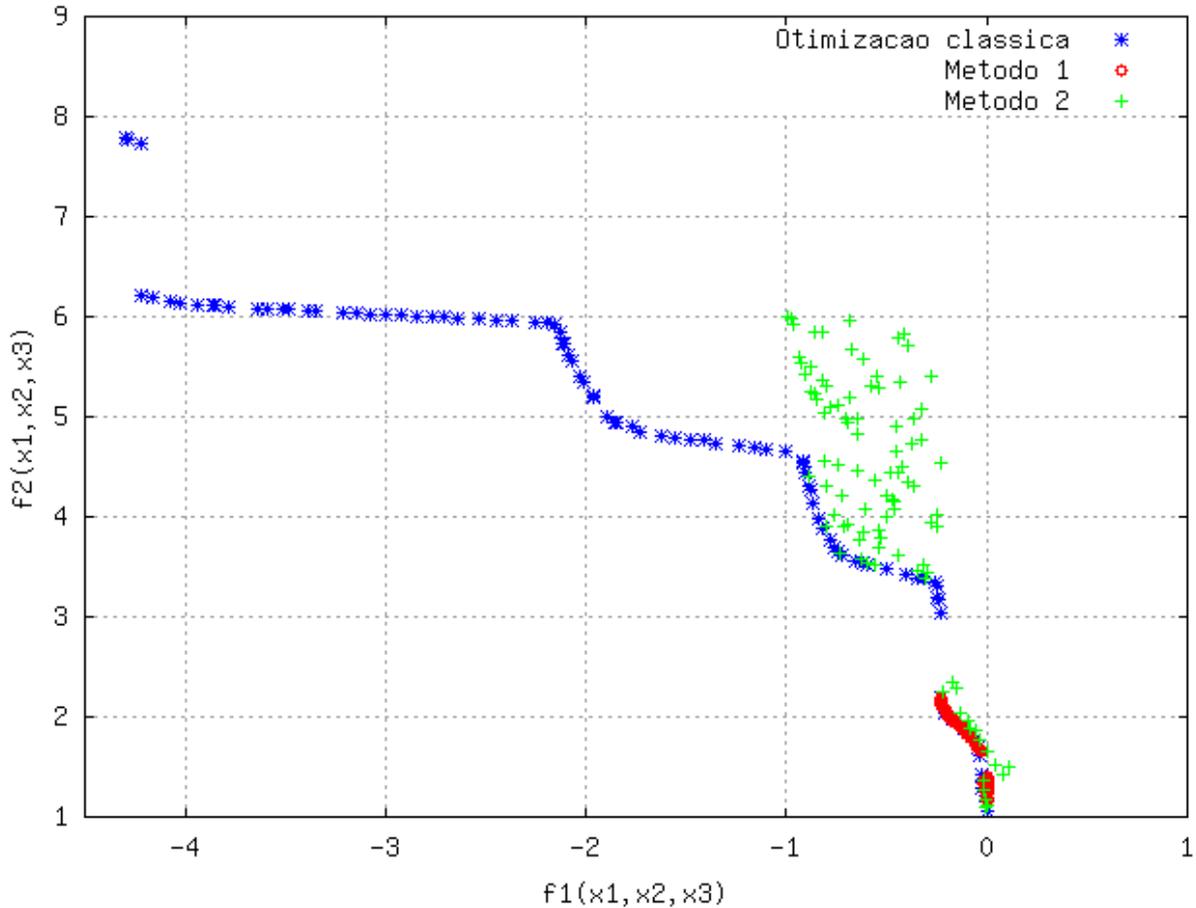


Figura 3.39: Resultado da otimização para o cenário 2.

Também foi realizada uma análise da variância para as otimizações realizadas. O resultado está mostrado nas tabelas 3.26 e 3.27.

Os resultados mostram que é possível a otimização robusta usando a UT para duas variáveis. O método 1 de otimização robusta é capaz de alcançar mais soluções robustas que o método 2, mostrando uma maior eficiência no que diz respeito a variação das soluções.

Para a otimização clássica com o NSGA-II, 61 soluções são consideradas irrealizáveis por violar o valor da variável x_1 ou da variável x_2 de acordo com os limites estabelecidos na equação 3.29. Já para a otimização robusta, todas as soluções são realizáveis de acordo com a equação 3.29.

Cenário 2			
Variável	$\max \sigma_{f_1}^2$	$E[\sigma_{f_1}^2]$	$\sigma_{f_1}^2 \leq \frac{\pi}{100}$
Otimização Clássica	4.7029	2.0074	13
Método 1	0.0011644	3.9188e-04	96
Método 2	1.5171	0.51371	17

Tabela 3.26: Análise dos parâmetros sensíveis para a função $f_1(x_1, x_2, x_3)$ de acordo com o cenário 2.

Cenário 2			
Variável	$\max \sigma_{f_2}^2$	$E[\sigma_{f_2}^2]$	$\sigma_{f_2}^2 \leq \frac{\pi}{100}$
Otimização Clássica	4.0208	2.4939	7
Método 1	0.041320	0.0082347	86
Método 2	2.7923	1.2461	12

Tabela 3.27: Análise dos parâmetros sensíveis para a função $f_2(x_1, x_2, x_3)$ de acordo com o cenário 2.

As figuras de 3.40 até 3.45 mostram o comportamento do desvio padrão associado a cada solução para cada método de otimização. Novamente, nestas figuras, em azul é mostrado o valor médio e o desvio padrão associados a cada solução e em vermelho, é mostrado o desvio padrão aceitável para cada solução, em volta da própria solução (mostrada em * azul).

Como pode ser observado nas figuras 3.40 a 3.45, a otimização clássica com o NSGA-II apresenta inúmeras soluções com desvio padrão superior ao limite especificado durante a etapa de modelagem da incerteza. Para o método 2, isto também é verdadeiro. Apenas o método 1 apresenta um bom desempenho na redução do valor do desvio padrão associado às funções objetivo.

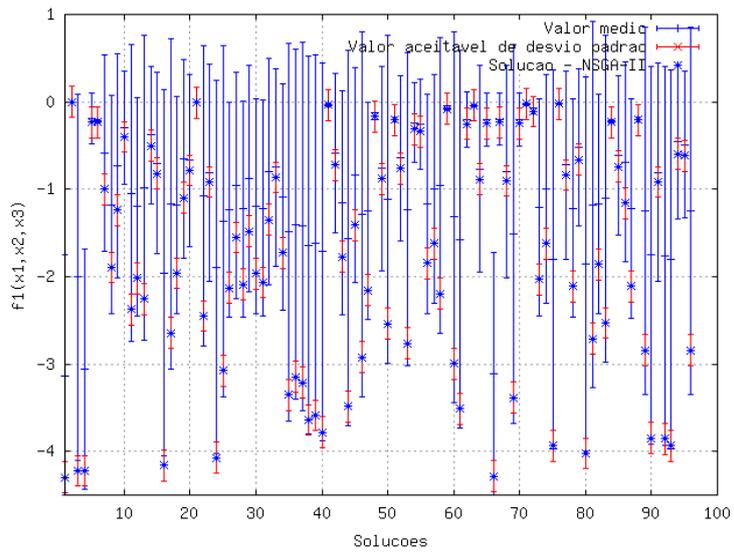


Figura 3.40: Análise de sensibilidade das soluções ótimas não robustas.

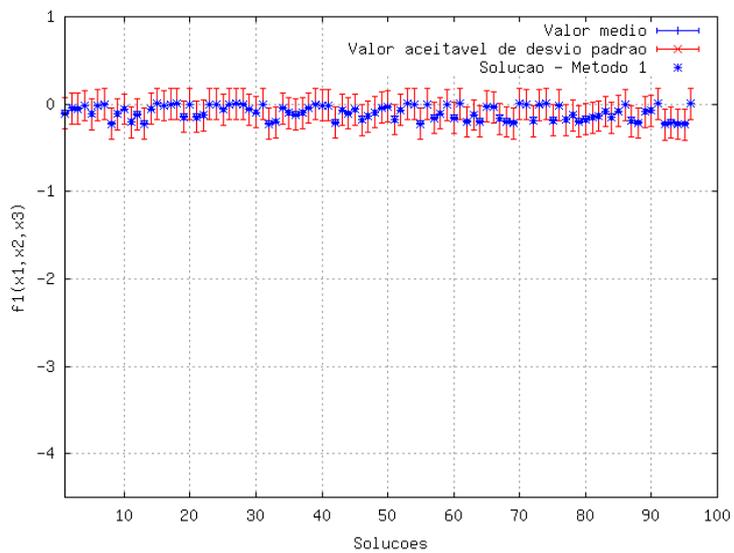


Figura 3.41: Análise de sensibilidade das soluções ótimas robustas dadas pelo método 1.

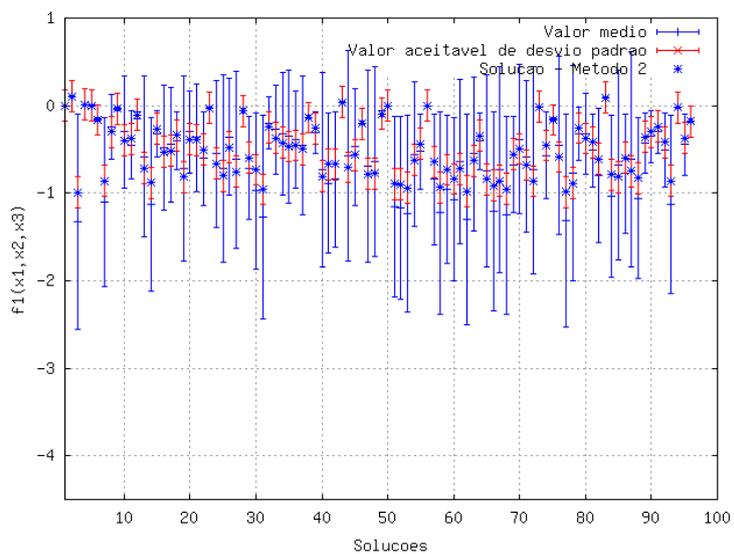


Figura 3.42: Análise de sensibilidade das soluções ótimas robustas dadas pelo método 2.

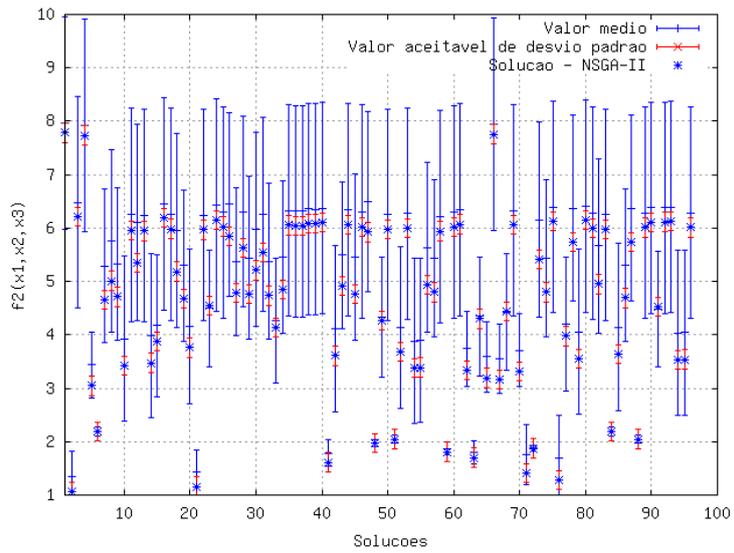


Figura 3.43: Análise de sensibilidade das soluções ótimas não robustas.

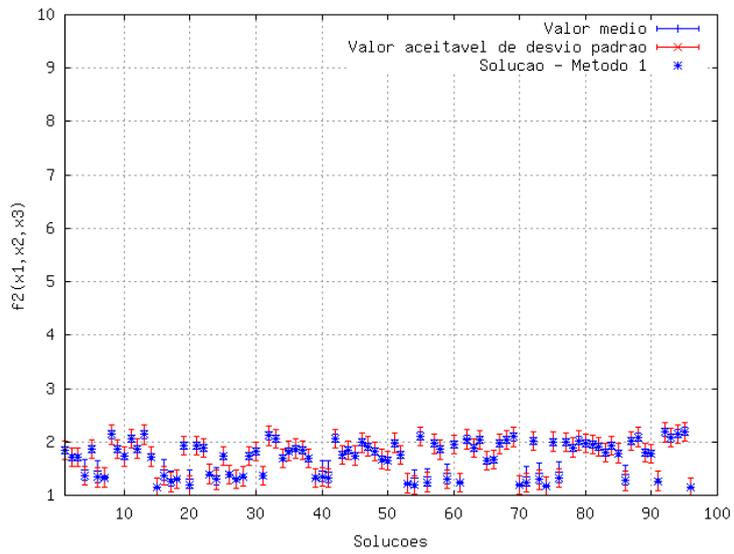


Figura 3.44: Análise de sensibilidade das soluções ótimas robustas dadas pelo método 1.

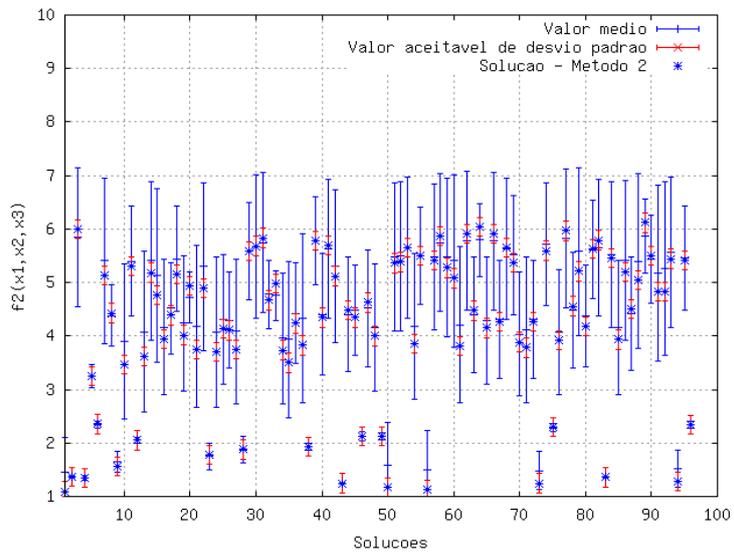


Figura 3.45: Análise de sensibilidade das soluções ótimas robustas dadas pelo método 2.

3.3.5 Considerações finais

Os resultados mostrados nos problemas de 1 a 3 validam o algoritmo de otimização robusta.

O problema mostrado na seção 3.3.4 mostra que a metodologia de projeto robusto especificada é viável para resolução de problemas de projeto robusto. A análise de sensibilidade do problema é essencial para verificação de quais variáveis são impactantes para o problema, necessitando assim de uma otimização robusta.

Entre os dois métodos especificados (3.15), o método 1, que apresenta uma combinação da média e do desvio padrão, apresenta melhores resultados quando se deseja uma minimização da variância das funções objetivo. Este método pode ser ajustado através de um parâmetro λ que define o grau de robustez das soluções, sendo que para soluções mais robustas a soma da média e do desvio padrão apresenta soluções mais robustas. Já com o método 2, que apresenta a minimização da média e a minimização da variância, é possível encontrar soluções com valores mínimos de variância e outras soluções com melhores valores médio das funções objetivo, não sendo um método muito eficaz para minimização do desvio padrão.

Outro aspecto importante é a percepção de que as soluções robustas podem ser tanto soluções dentro da frente ótima de Pareto dada pelo NSGA-II quanto outras soluções do espaço de busca. Por isso, a realização apenas da otimização clássica com a análise de sensibilidade *a posteriori* não é capaz de encontrar todas as soluções robustas do projeto, sendo necessário a utilização de um algoritmo de otimização robusta.

Enfim, os resultados apresentados nos problemas analíticos mostram que o uso da UT combinado com o NSGA-II alcança resultados satisfatórios na busca de soluções robustas e que a metodologia de projeto robusto pode ser aplicada em problemas reais.

4 OTIMIZAÇÃO ROBUSTA DO AMPLIFICADOR

CLASSE E

Apresentada a metodologia de projeto robusto na seção 3.2, esta será aplicada no projeto robusto de um amplificador de potência classe E.

Inicialmente, serão apresentados os principais conceitos de amplificadores de potência com atenção especial ao classe E. Em seguida, será realizado o projeto robusto do classe E de acordo com a metodologia apresentada. Além disto, para este circuito, será realizada uma comparação da UT com o método de Monte Carlo.

4.1 AMPLIFICADORES DE POTÊNCIA

Amplificadores de potência são circuitos que amplificam potência em rádio frequência – RF – de entrada em potência RF de saída mais elevada. São amplamente utilizados em transmissores de sistemas de comunicação em RF. Nesses, o amplificador de potência é o elemento mais crítico em relação ao consumo de potência. Por isso, um importante parâmetro de projeto de amplificadores de potência é a eficiência, que define a relação entre a potência requerida pelo circuito para amplificação e a potência entregue à carga. A eficiência de um amplificador depende fortemente do consumo próprio do transistor e os amplificadores mais eficientes são os não-lineares [6].

A eficiência mede quanto da potência entregue pela fonte de alimentação está sendo convertida em potência de saída. É dada pela razão entre a potência RF de saída entregue à carga na frequência de operação, $P_{RF,saida}$, e a potência DC de alimentação, P_{DC} (onde DC significa corrente contínua, do inglês *direct current*):

$$\eta = \frac{P_{RF,saida}}{P_{DC}}. \quad (4.1)$$

O ganho de potência também é um importante parâmetro de projeto. É dado pela relação entre a potência RF de saída, $P_{RF,saida}$, e a potência RF de entrada, $P_{RF,entrada}$:

$$G = \frac{P_{RF,saida}}{P_{RF,entrada}}. \quad (4.2)$$

Outro conceito importante é o de eficiência de potência adicionada – PAE (do inglês *Power Add*

Efficiency). É dada pela razão entre a diferença de potência do sinal de saída e de entrada (diferença que representa a potência adicionada) pela potência de alimentação:

$$PAE = \frac{P_{RF,saida} - P_{RF,entrada}}{P_{DC}} \quad (4.3)$$

A PAE considera o ganho do dispositivo e pode ser reescrita da seguinte forma:

$$PAE = \eta \left(1 - \frac{1}{G} \right) \quad (4.4)$$

Para um ganho elevado, a PAE apresenta valor próximo ao valor da eficiência η (pode ser observado pela equação 4.4).

Os amplificadores de potência são classificados em classes de acordo com o modo de operação dos dispositivos ativos presentes no circuito, ou seja, de acordo como as formas de onda de corrente e tensão [32]. O dispositivo ativo é o componente que faz a amplificação, neste estudo, um transistor.

As nomenclaturas classe A, B, AB e C referem-se ao ciclo de condução de corrente. Em um amplificador classe A, o dispositivo ativo está conduzindo corrente 100% do tempo (ou seja, durante todo o ciclo); no classe AB entre 50% e 100% do tempo; no classe B, em 50% do tempo e no classe C em menos de 50% do ciclo.

O amplificador classe A opera em modo linear, com o transistor operando na região ativa (região de triodo), o que implica numa dissipação de potência contínua durante toda a operação de amplificação. O amplificador classe A apresenta baixa distorção na corrente sobre a carga e alto ganho, porém um baixo rendimento, pois a perda no transistor ocorre em todo o ciclo. Para o amplificador classe B, a eficiência é maior, contudo a corrente de saída apresenta distorção harmônica sendo necessário o uso de um filtro na saída. O amplificador classe AB apresenta uma distorção menor que um classe B e uma eficiência maior que um classe A. O amplificador classe C apresenta eficiência de até 100%, contudo seu ganho de potência é menor e necessita também de um filtro na saída do sinal.

Os amplificadores classe D, E e F utilizam transistores ativos que operam chaveados, de maneira não-linear. Desta forma, operam em dois estados: fechado, quando operam na região de triodo e aberto, quando operam em região de corte. Considerando os transistores ativos como chaves ideais e desconsiderando os efeitos parasitários do circuito, a eficiência pode chegar a 100%. Ainda em condições reais, a eficiência destes amplificadores é maior que os de classes A, B, AB e C.

Novamente, vê-se o esforço na maximização da eficiência. Amplificadores classe E são tidos como de alta eficiência e além disso apresentam um circuito simples com poucos componentes. Essas razões

determinam o grande interesse dos projetistas pelos amplificadores classe E.

4.1.1 Amplificador de potência classe E

Amplificadores de potência classe E são amplificadores chaveados de alta eficiência empregados em frequências altas, nas quais o tempo de chaveamento é comparável à frequência de operação. O circuito amplificador classe E foi originalmente proposto em [6]. Este circuito é mostrado na figura 4.1. Uma versão para análise deste circuito é mostrada na figura 4.2.

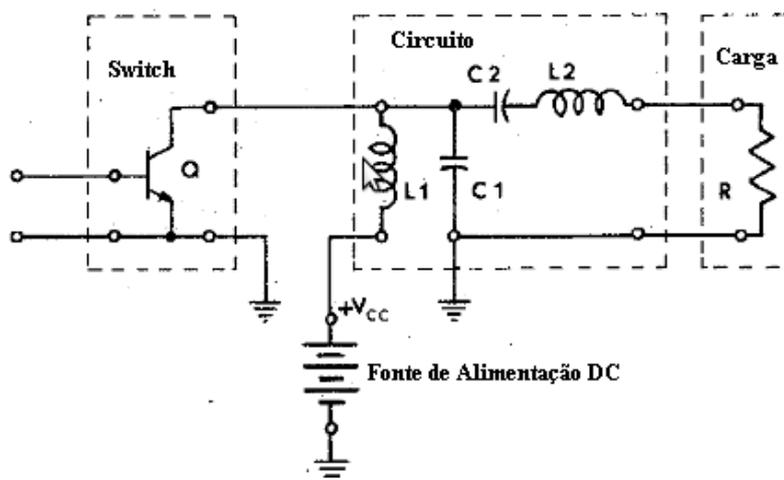


Figura 4.1: Circuito classe E apresentado em [6] (adaptado).

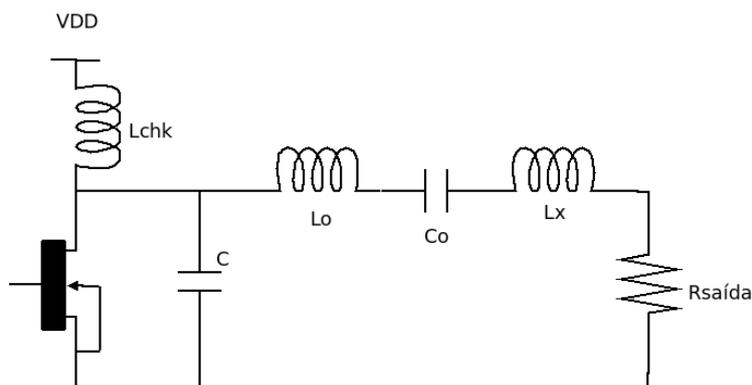


Figura 4.2: Topologia do circuito classe E.

A seguir será descrito cada um dos elementos do circuito apresentado na figura 4.2. Para análise do comportamento ideal, considera-se:

- Transistor: o transistor opera como uma chave ideal, com resistência nula quando fechado e resistência infinita quando aberto. Para garantir o funcionamento do transistor como chave, é

necessário que a tensão de entrada seja uma onda quadrada com ciclo de operação e amplitude desejadas.

- Indutor L_{CHK} : este é o indutor de obstrução, com um valor suficientemente alto para representar uma alta impedância na frequência de operação e permitir a passagem de uma corrente constante da fonte de alimentação V_{DD} para o resto do circuito. Assim, a resistência DC é nula e a reatância é infinita na frequência de operação.
- Capacitor C : representa a capacitância dreno-fonte do transistor e uma capacitância adicional de ajuste. Aqui estão incluídas as capacitâncias parasitárias do transistor e dos componentes conectados ao dreno.
- Capacitor C_o e indutor L_o : representam um circuito ressonante em série, com a finalidade de permitir a passagem da frequência fundamental de operação à carga. O indutor L_x representa uma indutância adicional.

A figura 4.3 mostra as formas de onda de tensão e corrente no dreno para o amplificador classe E. A corrente e a tensão no transistor não se sobrepõem no decorrer do tempo, fazendo com que a dissipação no transistor seja nula.

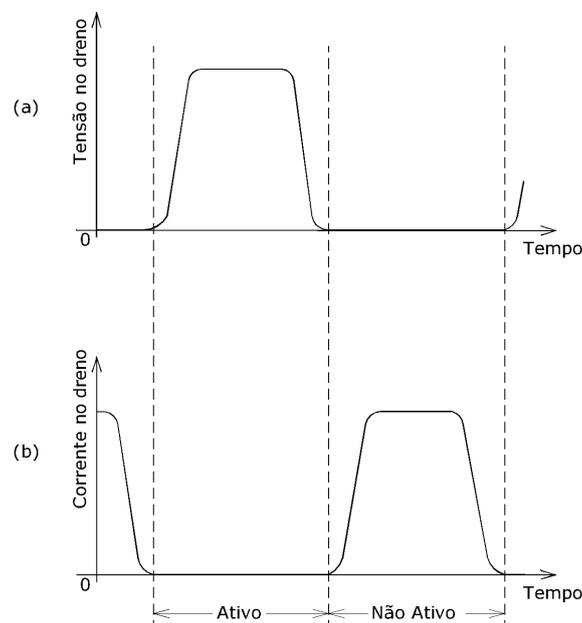


Figura 4.3: Formas de onda de tensão e corrente no dreno para o amplificador classe E.

4.2 OTIMIZAÇÃO ROBUSTA DO AMPLIFICADOR CLASSE E

O circuito analisado é o apresentado na figura 4.2, operando na frequência de 921,25 MHz. Para simulação deste circuito foi utilizado o *software Golden Gate*, que é um *software* comercial para tratamento de circuitos de microondas e RF que utiliza o método do Equilíbrio Harmônico (em inglês *Harmonic Balance*) para a simulação de circuitos. Para maiores informações sobre o *Golden Gate* e seu funcionamento, é recomendada a leitura de [33].

O NSGA-II trabalha de forma integrada com o *Golden Gate* para a otimização dos amplificadores. Dois arquivos de simulação são utilizados pelo *Golden Gate*: `input.scs`, que apresenta a topologia do circuito, e `hb.gg`, que apresenta os parâmetros de simulação e as métricas que serão computadas na otimização. Estes dois arquivos estão no Anexo I.

No arquivo `hb.gg` estão definidos os objetivos (em inglês *goals*) do processo de otimização. Para o classe E, dois parâmetros são os mais importantes: a eficiência e a potência de saída. Assim, os objetivos da otimização do amplificador classe E são:

- Potência de saída de 10 mW: $P_{saída} = 10\text{mW}$ e
- Eficiência de 100%: $\eta = 100\%$.

Durante o processo de otimização, o programa principal realiza chamadas ao *Golden Gate* para a simulação do circuito. Como resultado, o *Golden Gate* retorna os valores das funções objetivo, não os valores reais, mas o erro quadrático do valor obtido da simulação do circuito com o valor especificado. Assim, a minimização realizada pelo NSGA-II é feita a partir deste erro quadrático, ou seja, cada função objetivo é dada por:

$$\text{Função Objetivo} = \sqrt{(f(x)_{\text{calculado}} - f(x)_{\text{especificado}})^2}. \quad (4.5)$$

A faixa de variação dos elementos que serão otimizados está mostrada na tabela 4.1 (referente à figura 4.2). Estas informações estão no arquivo `hb.gg` (Anexo I.2).

Os parâmetros de configuração do NSGA-II utilizados para estas simulações estão mostrados na tabela 4.2.

Elemento	Mínimo	Máximo
V_{DD}	0,5 V	1,5 V
L_{CHK}	10 nH	100 nH
L_o	10 nH	100 nH
C_o	0,039 pF	3,9 pF
L_x	1 nH	100 nH
C	0,063 pF	0,63 pF

Tabela 4.1: Faixa de variação dos elementos constitutivos do classe E.

Parâmetro	Valor
Número de indivíduos	48
Número de gerações	200
Probabilidade de cruzamento	1
Probabilidade de mutação	0,05
Parâmetro de cruzamento η_c	20
Parâmetro de mutação η_m	20

Tabela 4.2: Parâmetros de configuração do NSGA-II.

4.2.1 Otimização Clássica com o NSGA-II

Primeiramente, foi realizada uma otimização do circuito utilizando a formulação clássica de otimização.

A figura 4.4 mostra a evolução dos indivíduos através das gerações. A frente ótima de Pareto, mostrada em azul, varre desde os indivíduos com alta eficiência até os que possuem potência de saída próxima à 10mW. A maior eficiência encontrada foi de 77,3% com potência de saída de 30,68 mW. O indivíduo com potência de saída mais próxima de 10 mW (10,69 mW) possui eficiência de 68,95%. Estes indivíduos estão mostrados na figura 4.5 sob a legenda *Indivíduo B* e *Indivíduo C* respectivamente.

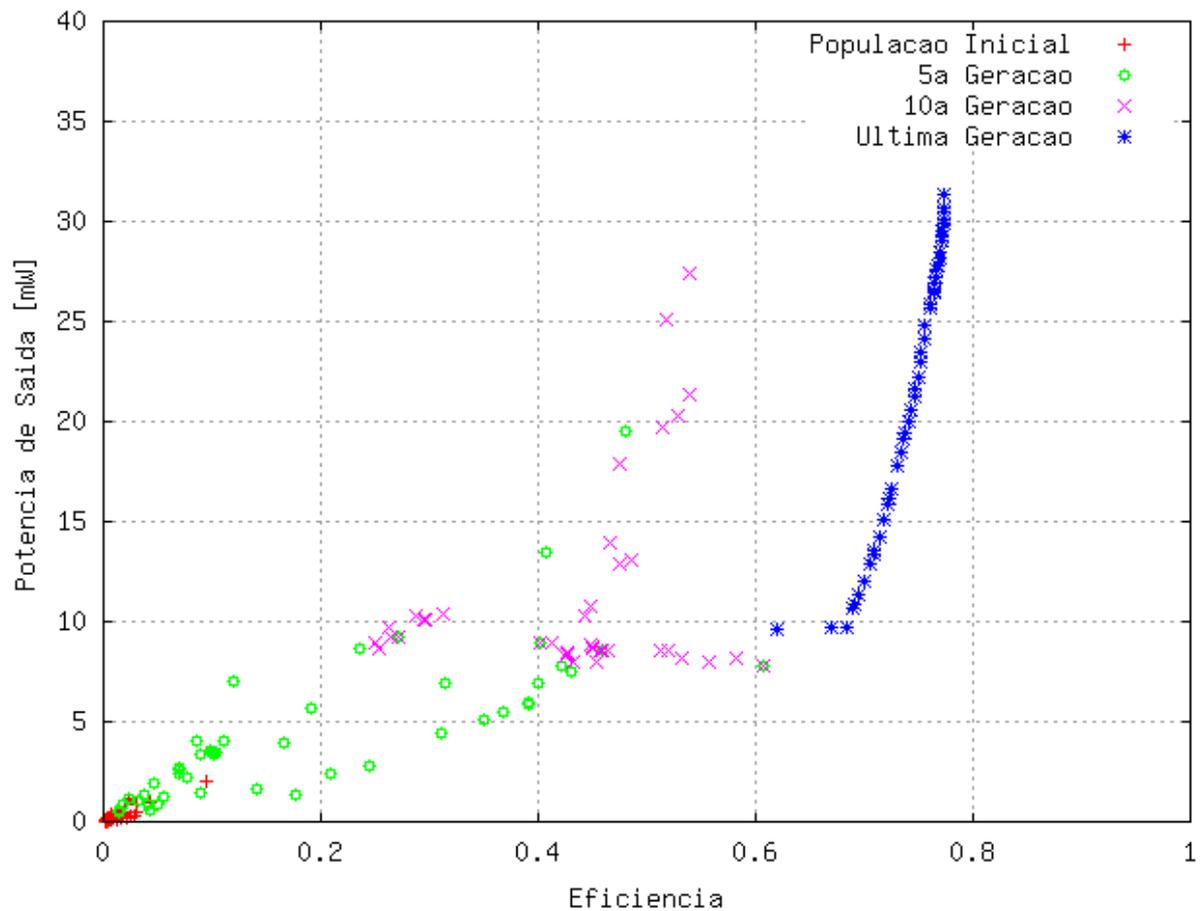


Figura 4.4: Otimização clássica do amplificador classe E utilizando o NSGA-II.

4.2.2 Comparação da UT com o método de Monte Carlo

Para o amplificador classe E também foi realizada a comparação da UT com o método de Monte Carlo. Para isso, foi escolhido um indivíduo da frente ótima de Pareto. Este indivíduo está mostrado na figura 4.5 como *Indivíduo A* e seus parâmetros estão na tabela 4.3. Para este indivíduo a potência de saída é de 16,63 mW e a eficiência é de 72,52%. Para a UT foram utilizados cálculos com 2 e 3 pontos sigma. Para o método de Monte Carlo foram utilizadas 10.000 amostras.

Para comparação da UT com o método de Monte Carlo, foram realizados dois testes:

- Variação do capacitor C_o , de distribuição gaussiana com desvio padrão de 0,025 pF e 0,05 pF (que correspondem aproximadamente à um desvio padrão de 5% e 10% do valor do capacitor). Os resultados estão mostrados nas tabelas 4.4 e 4.5.

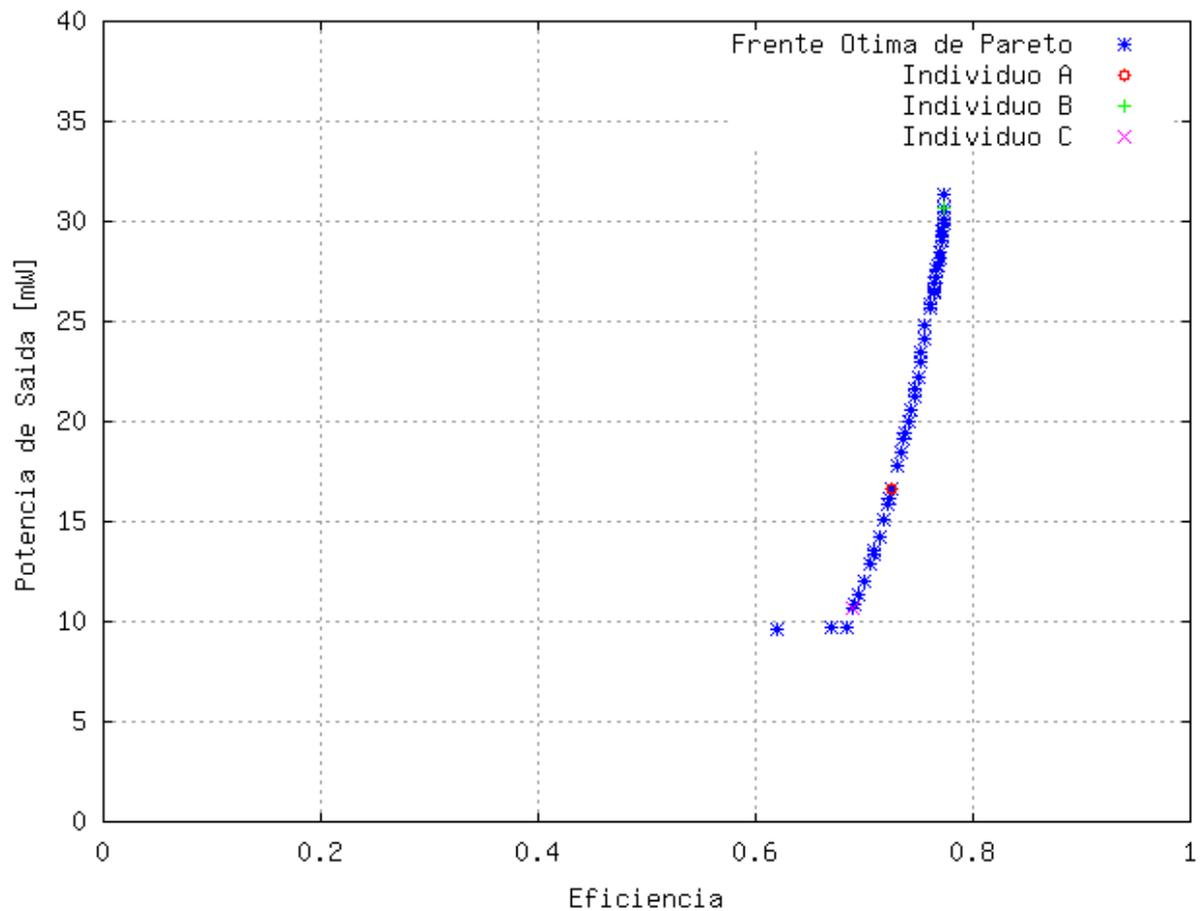


Figura 4.5: Frente ótica de Pareto para a otimização do amplificador classe E utilizando o NSGA-II.

Elemento	Valor
V_{DD}	1,08483 V
L_{CHK}	10,00580 nH
L_o	63,95888 nH
C_o	0,48584 pF
L_x	2,34181 nH
C	0,06378 pF

Tabela 4.3: Valor dos componentes do amplificador classe E (figura 4.2).

- Variação do indutor L_x , com desvio padrão de 0,125 nH e 0,25 nH (que correspondem aproximadamente à um desvio padrão de 5% e 10% do valor do indutor). Os resultados estão mostrados nas tabelas 4.6 e 4.7.

	Eficiência (%)		Potência de Saída (mW)	
	Valor Médio	Desvio Padrão	Valor Médio	Desvio Padrão
Monte Carlo	66,603	0,047448	16,616	0,003335
UT-3	66,618	0,059769	16,496	0,003028
UT-2	67,696	0,025064	14,910	0,002127

Tabela 4.4: Comparação entre o método de Monte Carlo e a UT para o parâmetro C_o com $\sigma_{C_o} = 0,025$.

	Eficiência (%)		Potência de Saída (mW)	
	Valor Médio	Desvio Padrão	Valor Médio	Desvio Padrão
Monte Carlo	58,916	0,12160	14,645	0,00478
UT-3	59,707	0,15101	14,417	0,00397
UT-2	63,332	0,06870	13,319	0,00371

Tabela 4.5: Comparação entre o método de Monte Carlo e a UT para o parâmetro C_o com $\sigma_{C_o} = 0,05$.

	Eficiência (%)		Potência de Saída (mW)	
	Valor Médio	Desvio Padrão	Valor Médio	Desvio Padrão
Monte Carlo	70,196	0,000720	17,035	0,000177
UT-3	70,196	0,000717	17,036	0,000176
UT-2	70,164	0,000387	16,948	0,000088

Tabela 4.6: Comparação entre o método de Monte Carlo e a UT para o parâmetro L_x com $\sigma_{L_x} = 0,125$.

	Eficiência (%)		Potência de Saída (mW)	
	Valor Médio	Desvio Padrão	Valor Médio	Desvio Padrão
Monte Carlo	70,176	0,001474	17,032	0,000355
UT-3	70,177	0,001464	17,034	0,000353
UT-2	70,199	0,000836	16,859	0,000178

Tabela 4.7: Comparação entre o método de Monte Carlo e a UT para o parâmetro L_x com $\sigma_{L_x} = 0,25$.

Os resultados apresentados mostram que a UT é capaz de obter a média e a variância da potência de saída e da eficiência do amplificador classe E dado um nível de incerteza nos parâmetros de entrada

com menos cálculos que o método de Monte Carlo. Pode-se observar pelos resultados que a UT com três pontos sigma provê uma boa aproximação para a média e a variância, enquanto que com dois pontos sigma somente a previsão da média é confiável.

Validado o uso da UT para o amplificador classe E, esta será utilizada, na sua versão com três pontos sigma, no processo de otimização robusta.

4.2.3 Projeto Robusto do Amplificador Classe E

O projeto robusto, seguindo as etapas definidas na seção 3.2, será realizado para o amplificador classe E.

4.2.3.1 Modelagem das incertezas associadas às variáveis de entrada

A caracterização das incertezas associadas aos componentes do dispositivo (variáveis de entrada) é dever do projetista. Para o amplificador classe E, será considerado que estas incertezas têm distribuição de probabilidade gaussiana, com desvio padrão que corresponde a 10% do valor nominal do componente.

4.2.3.2 Definição dos parâmetros sensíveis

O problema de otimização do amplificador classe E envolve a otimização de seis variáveis de entrada. Para a escolha das variáveis que serão otimizadas pelo projeto robusto, foi utilizado o método proposto na seção 3.2.2. A análise de sensibilidade foi realizada apenas na frente ótima de Pareto resultante da otimização clássica, pois o tempo de simulação do circuito iria onerar muito a análise do problema. Os resultados para a análise de sensibilidade da frente ótima de Pareto estão mostrados na tabela 4.8.

Variavel	$E[\sigma_{\eta}^2]$	$E[\sigma_{P_{saída}}^2]$
V_{DD}	$2,3214 \cdot 10^{-4}$	$2,0819 \cdot 10^{-5}$
L_{CHK}	$1,0921 \cdot 10^{-3}$	$1,0882 \cdot 10^{-7}$
L_o	$1,0849 \cdot 10^{-3}$	$1,7324 \cdot 10^{-5}$
C_o	$3,1867 \cdot 10^{-3}$	$1,7368 \cdot 10^{-5}$
L_x	$1,3767 \cdot 10^{-4}$	$6,7327 \cdot 10^{-6}$
C	$1,1939 \cdot 10^{-6}$	$9,0322 \cdot 10^{-11}$

Tabela 4.8: Análise de sensibilidade da frente ótima de Pareto.

Estes resultados apontam 2 variáveis mais sensíveis de projeto: L_o e C_o . Estes parâmetros formam um circuito ressonante em série e definem a frequência de operação do amplificador classe E. Desta forma, estes parâmetros podem ser vistos como um único parâmetro relacionado pela seguinte fórmula:

$$\omega = \frac{1}{\sqrt{L_o C_o}} \quad (4.6)$$

Para simplificar, será considerada apenas a otimização robusta no capacitor C_o , dado que uma variação no parâmetro L_o pode ser diretamente calculada no parâmetro C_o .

Cabe observar que este circuito apresenta pouca variação nos valores de potência de saída e eficiência na presença de incertezas nas variáveis de entrada. Ainda assim, o projeto robusto será realizado para a variável C_o .

As figuras 4.6 e 4.7 mostram a relação da variação do capacitor C_o com a eficiência e a potência de saída para duas soluções da frente ótima de Pareto.

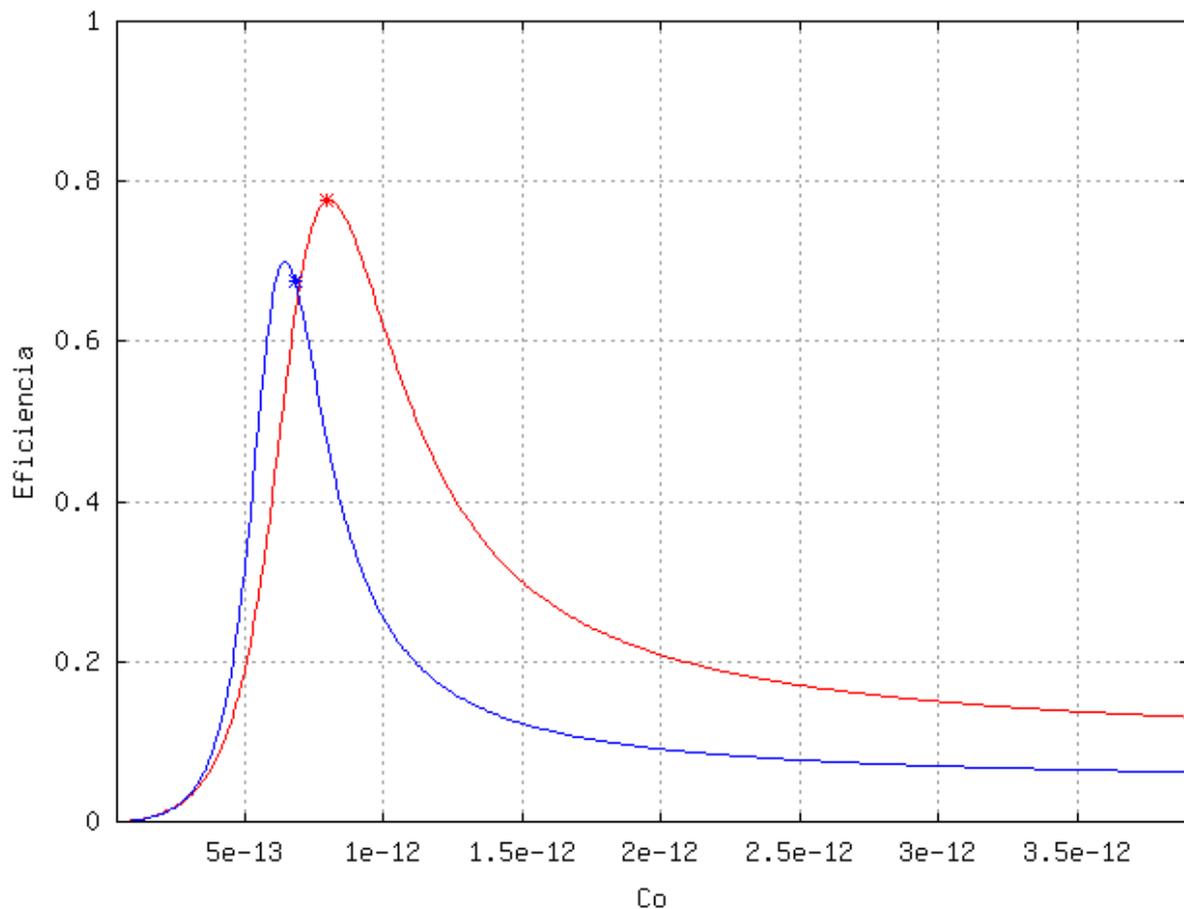


Figura 4.6: Verificação da eficiência dado uma variação do parâmetro C_o para duas soluções da otimização pelo NSGA-II.

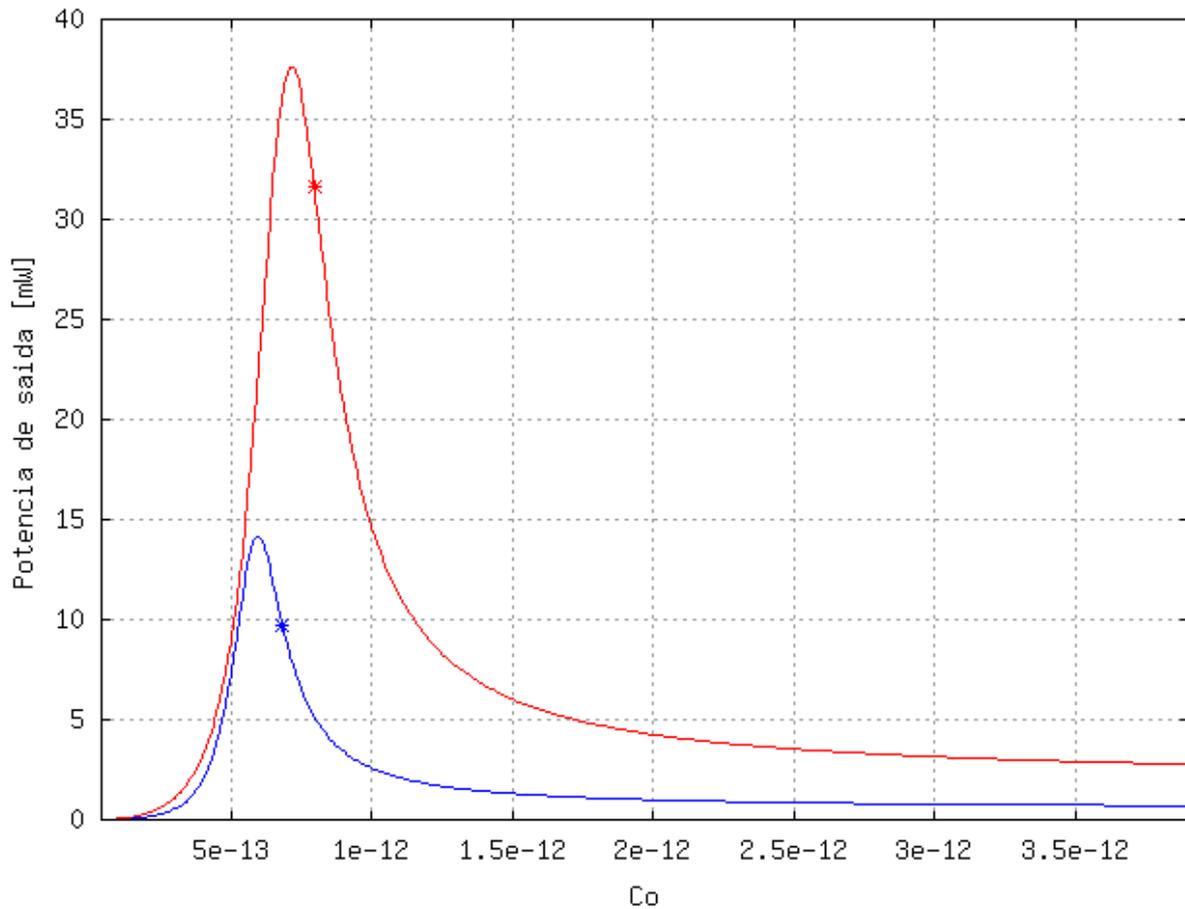


Figura 4.7: Verificação da potência de saída dado uma variação do parâmetro C_o para duas soluções da otimização pelo NSGA-II

Como pode ser observado nas figuras 4.6 e 4.7, as duas soluções apresentadas estão em pontos de alta variação em relação a potência de saída e a eficiência. Uma variação no valor do capacitor pode levar a um desempenho diferente do desempenho nominal resultante do processo de otimização. Percebe-se que na presença de incertezas estas soluções são soluções não robustas. É desejável que com o projeto robusto do amplificador classe E estas soluções estejam em regiões mais planas, ou seja, em regiões em que o desempenho especificado não se altere na presença de perturbações.

4.2.3.3 Processo de otimização

Após a verificação do parâmetro sensível de projeto, pode-se prosseguir no processo de otimização robusta. Os dois métodos mostrados na figura 3.15 foram aplicados na otimização robusta do amplificador classe E. Os resultados estão mostrados na figura 4.8

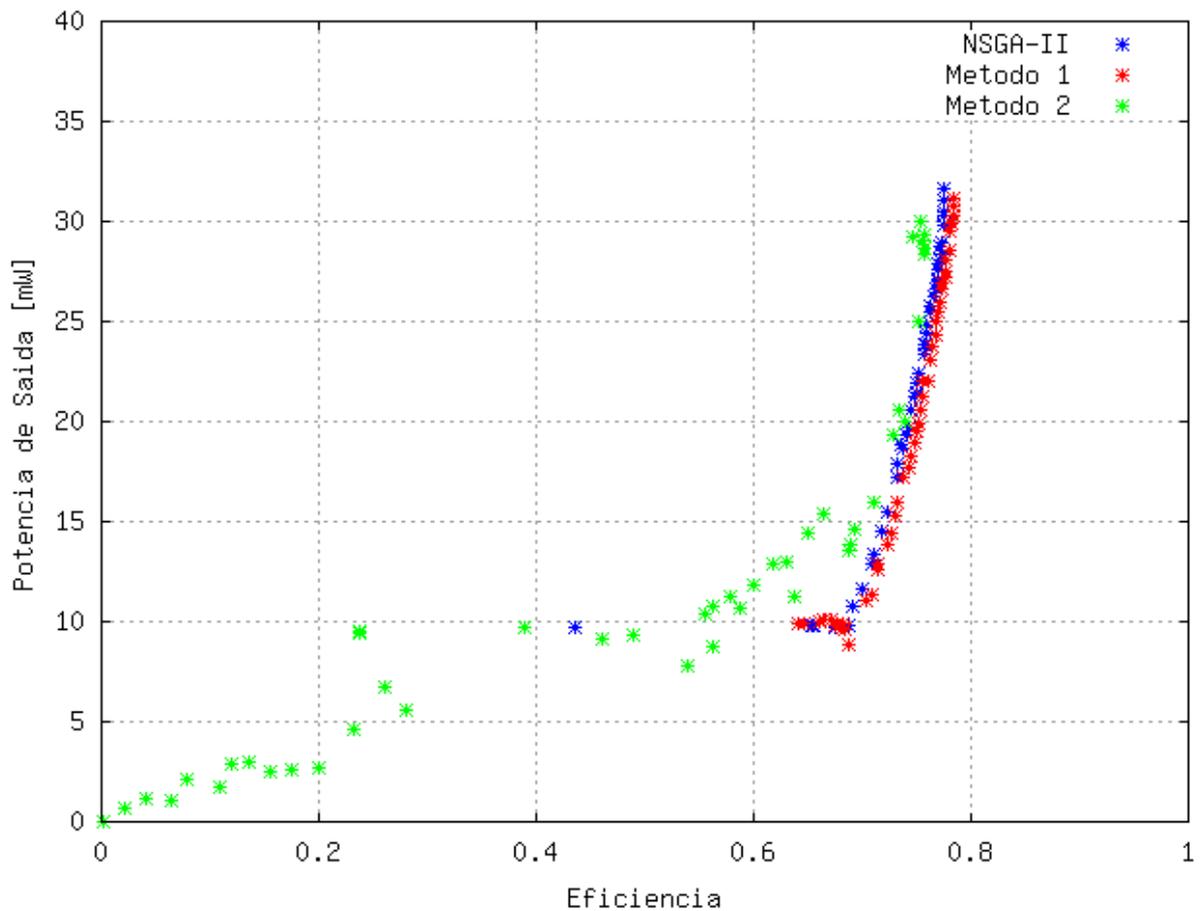


Figura 4.8: Comparação dos resultados de otimização robusta para o classe E.

A frente de Pareto obtida pela otimização robusta utilizando o método 1 tem formato semelhante a frente ótima de Pareto obtida pela otimização clássica. Contudo, comparando as soluções robustas com as soluções não robustas para um mesmo valor de potência de saída, as soluções robustas apresentam uma eficiência maior.

A otimização robusta pelo método 2 não apresentou resultados satisfatórios, como pode ser observado na figura 4.8. Muitas soluções estão extremamente distantes da frente de Pareto, com eficiência muito baixa, o que não é desejado no projeto do amplificador classe E.

As figuras 4.9 e 4.10 mostram os valores de desvio padrão para a otimização clássica (NSGA-II) e para os métodos 1 e 2. Como o desvio padrão associado a estes resultados é muito baixo, para a apresentação destes resultados o desvio padrão da eficiência foi multiplicado por 100 e o desvio padrão da potência de saída foi multiplicado por 10.

Estas figuras mostram que o método 1 é capaz de reduzir drasticamente o desvio padrão associado as soluções, enquanto que o método 2 não tem esta mesma capacidade. Os resultados da otimização robusta

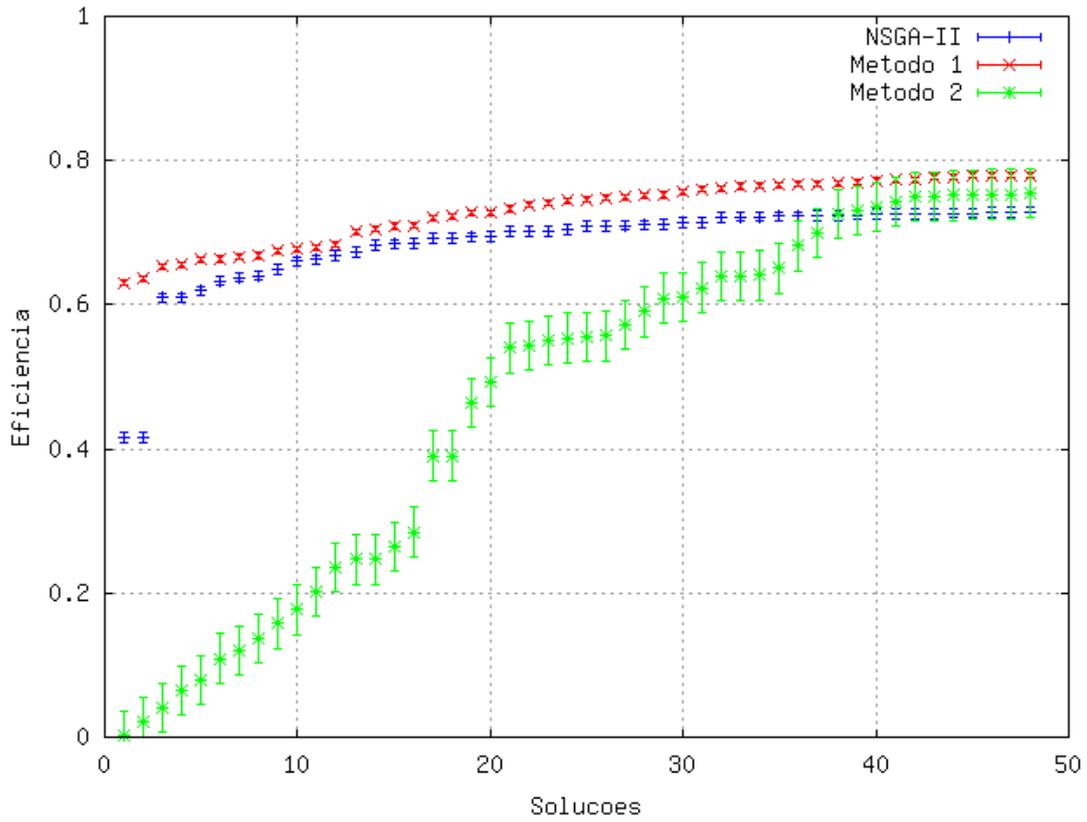


Figura 4.9: Análise de sensibilidade das soluções em relação a eficiência.

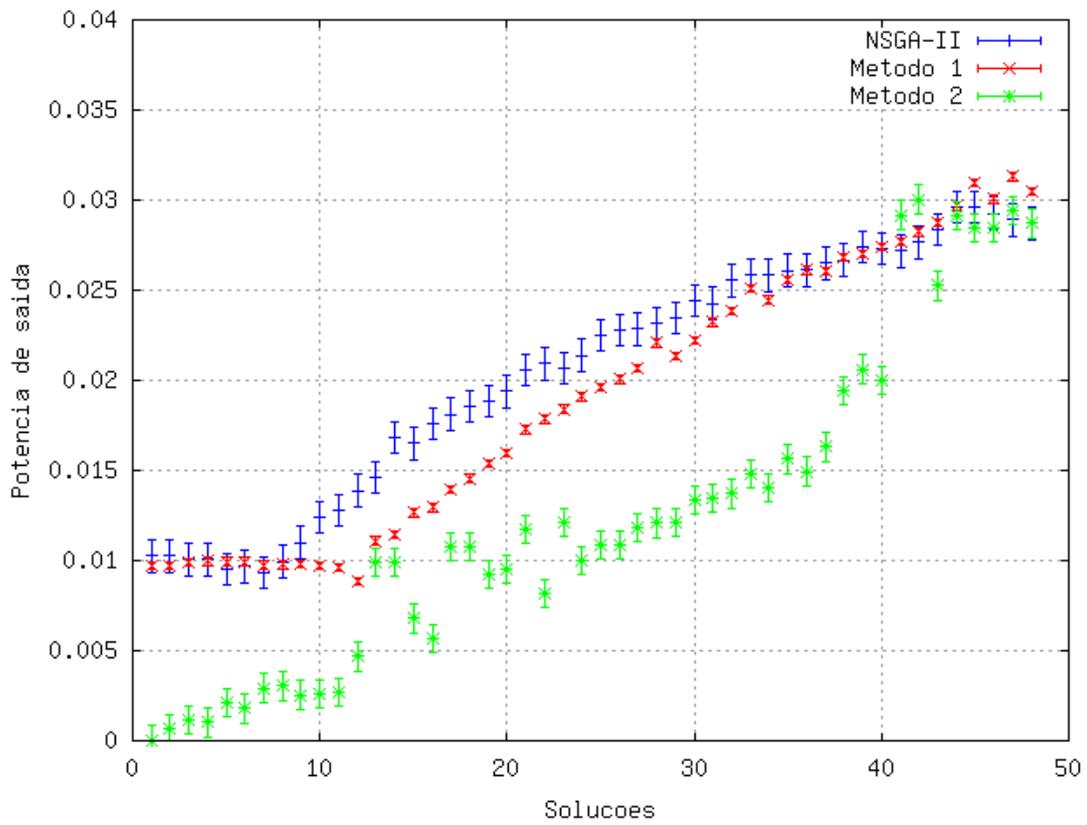


Figura 4.10: Análise de sensibilidade das soluções em relação a potência de saída

com o método 1 apresentam desvio padrão menor que os resultados da otimização clássica (NSGA-II), com uma potência de saída e eficiência em níveis desejados para um amplificador classe E.

As figuras 4.11 e 4.12 mostram a relação da variação do capacitor C_o com a eficiência e a potência de saída para duas otimizadas pelo método 1 de otimização robusta.

Os resultados mostrados nas figuras 4.11 e 4.12 são significativamente diferente dos resultados mostrados nas figuras 4.6 e 4.7 obtidas da otimização clássica. As soluções obtidas pela otimização robusta (método 1) estão posicionadas em regiões planas, enquanto as soluções obtidas pelo NSGA-II estavam em regiões inclinadas. Estes resultados mostram que a otimização robusta é capaz de encontrar soluções mais robustas, mais imunes à variações nos parâmetros de entrada.

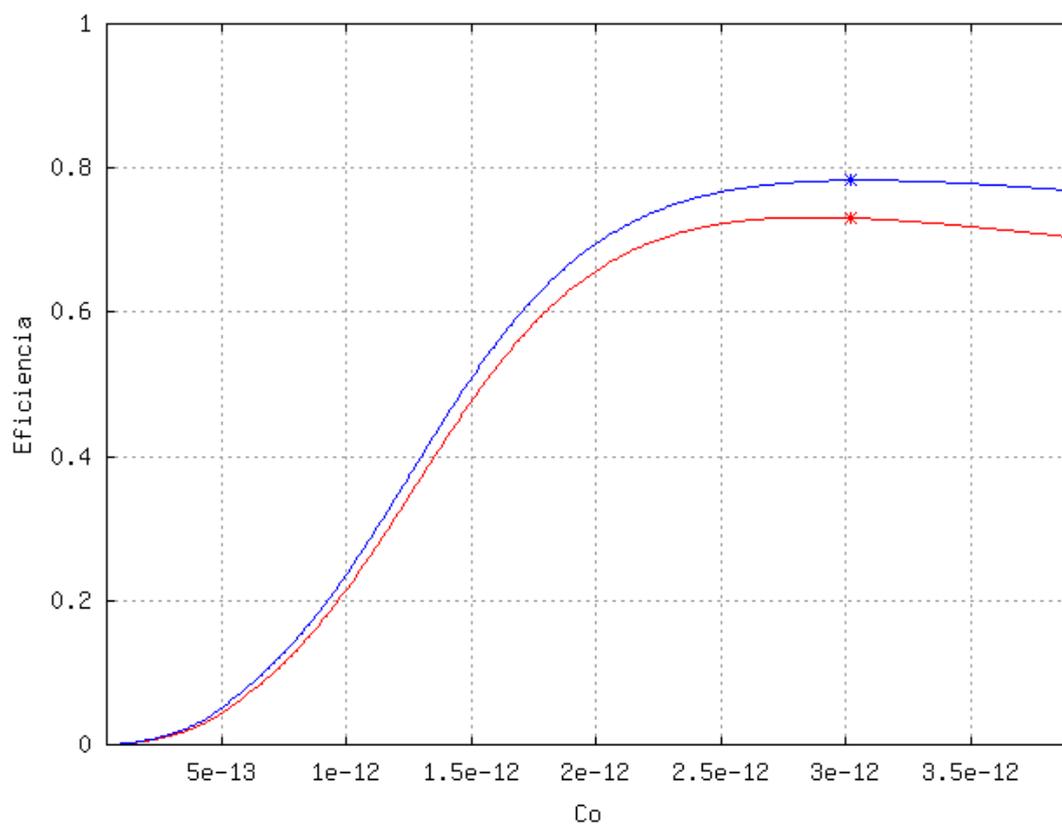


Figura 4.11: Verificação da eficiência dado uma variação do parâmetro C_o para duas soluções da otimização robusta pelo método 1.

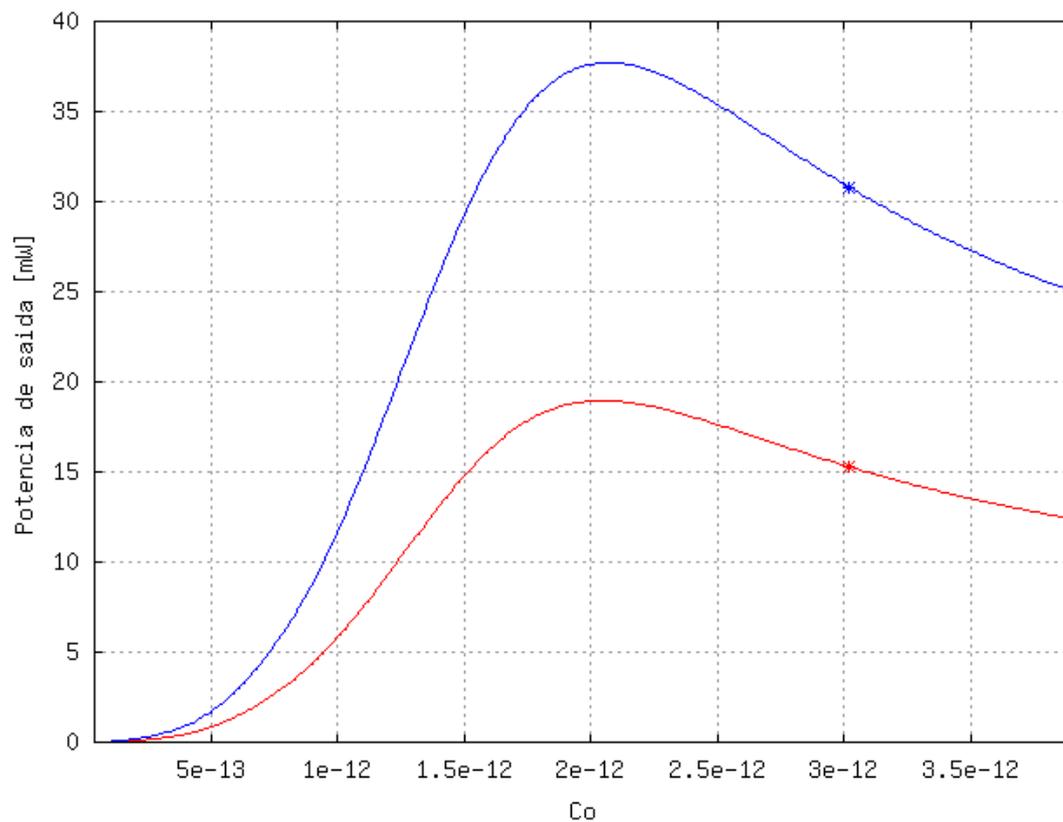


Figura 4.12: Verificação da potência de saída dado uma variação do parâmetro C_o para duas soluções da otimização robusta pelo método 1.

4.2.4 Considerações finais

É possível obter soluções mais robustas para o amplificador classe E utilizando a metodologia de projeto robusto apresentada. Novamente foi verificado que o método 1 de otimização robusta é o método que resulta em soluções com menor desvio padrão.

5 CONCLUSÕES

O projeto robusto visa atender processos de otimização que têm incertezas associadas às variáveis de entrada. Muitos métodos para busca de soluções robustas vêm sendo apresentados na literatura. Estes métodos requerem que seja feita uma análise de sensibilidade das possíveis soluções. Para isto, é necessário o cálculo da média e da variância das funções objetivo quando se têm incertezas associadas às variáveis de entrada. Este cálculo pode ser bastante custoso e muitas vezes é realizado pelo método de Monte Carlo. Soluções alternativas para este problema foram propostas na literatura. Nesta dissertação foi utilizada a transformada Unscented para o cálculo da média e da variância das funções a serem otimizadas. O cálculo da UT não acarreta grande esforço adicional e por isso pode ser usado na otimização.

Nesta dissertação, foi proposta uma metodologia para o projeto robusto, que envolve três etapas: a modelagem das incertezas associadas às variáveis de entrada, a definição dos parâmetros sensíveis de projeto e a otimização conjunta dos parâmetros sensíveis e dos outros parâmetros de projeto.

A modelagem das incertezas associadas às variáveis de entrada teve como objetivo a caracterização das incertezas envolvidas no processo. Em seguida, foi necessário definir como estas incertezas associadas às variáveis de entrada afetavam o problema. Com a UT foi possível determinar como a variação de um parâmetro afeta o desempenho do projeto. Assim foi possível definir quais variáveis são as mais sensíveis no projeto. Por fim, foi realizado o processo de otimização, de forma híbrida, ou seja, variáveis sensíveis foram otimizadas pela formulação robusta enquanto as variáveis pouco sensíveis foram otimizadas pela formulação clássica. Esta otimização robusta também utilizou a UT em sua formulação.

Para a otimização robusta foi implementado um algoritmo que associava a UT ao algoritmo NSGA-II. O algoritmo resultante foi testado em funções analíticas para verificar a capacidade de busca de soluções robustas. Os resultados mostraram que é possível encontrar soluções robustas utilizando o NSGA-II com as alterações demandadas pela UT.

Além disso, a metodologia apresentada de projeto robusto foi testada com uma função analítica. A metodologia apresentada pôde ser aplicada e mostrou-se eficaz na caracterização do projeto robusto. A metodologia de projeto robusto propunha duas alternativas para otimização, chamadas nesta dissertação de método 1 e método 2. Os resultados mostraram que a otimização robusta pelo método 1, ou seja, pela minimização de uma combinação da média e do desvio padrão, apresenta os melhores resultados. Neste método, o desvio padrão associado às funções objetivo é reduzido de forma significativa quando comparado

ao desvio padrão das soluções resultantes da otimização clássica.

Os resultados mostram que a otimização robusta é capaz de selecionar dentro da frente ótima de Pareto dada pelo NSGA-II as soluções mais robustas, o que também seria possível pela análise de sensibilidade a posteriori. Contudo, na otimização robusta, têm-se a certeza que as soluções encontradas são as mais robustas de projeto. Além disso, pela otimização robusta, outras soluções fora da frente dada pelo NSGA-II também podem ser encontradas.

Por fim, a metodologia proposta foi aplicada no projeto robusto de um amplificador classe E. Apesar de este amplificador não apresentar grande sensibilidade em relação aos parâmetros de entrada, a otimização robusta a partir da minimização da soma da média com o desvio padrão apresentou resultados ainda menores de desvio padrão.

Conclui-se que a metodologia apresentada para projeto robusto está validada para ser utilizada em problemas de otimização que tenham incertezas associadas as variáveis de entrada. Diversos são os ramos de aplicação da metodologia apresentada: síntese de antenas, processo de fabricação de dispositivos microeletrônicos, processo de fabricação de dispositivos de controle mecânico, entre outros.

Como continuidade deste trabalho, recomenda-se a avaliação do uso da variância como restrição, uso de outro algoritmo para otimização clássica, que seja capaz de obter resultados em diversos *clusters* para a etapa de análise de sensibilidade e definição dos parâmetros sensíveis de projeto, aplicação da UT com outras distribuições de probabilidade. Ainda, no estudo dos amplificadores de potência, a otimização robusta do classe E com seus efeitos parasitários.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] DEB, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, v. 6, n. 2, p. 182–197, Apr 2002.
- [2] STEINER, G.; WEBER, A.; MAGELE, C. Managing uncertainties in electromagnetic design problems with robust optimization. *Magnetics, IEEE Transactions on*, v. 40, n. 2, p. 1094–1099, March 2004.
- [3] JULIER, S.; UHLMANN, J. K. *A general method for approximating nonlinear transformations of probability distributions*. 1996. Eng. Dept. Univ. Oxford, Tech. Report., Oxford.
- [4] TSUTSUI, S.; GHOSH, A. Genetic algorithms with a robust solution searching scheme. *Evolutionary Computation, IEEE Transactions on*, v. 1, n. 3, p. 201–208, Sep 1997.
- [5] NISHIDA, N.; TAKAHASHI, Y.; WAKAO, S. Robust design optimization approach by combination of sensitivity analysis and sigma level estimation. *Magnetics, IEEE Transactions on*, v. 44, n. 6, p. 998–1001, June 2008.
- [6] SOKAL, N.; SOKAL, A. Class e-a new class of high-efficiency tuned single-ended switching power amplifiers. *Solid-State Circuits, IEEE Journal of*, v. 10, n. 3, p. 168–176, Jun 1975.
- [7] MENEZES, L. de et al. Modeling device manufacturing uncertainty in electromagnetic simulations. *Microwave Symposium Digest, 2008 IEEE MTT-S International*, p. 1385–1388, June 2008.
- [8] LU, H.; YEN, G. Multiobjective optimization design via genetic algorithm. *Control Applications, 2001. (CCA '01). Proceedings of the 2001 IEEE International Conference on*, p. 1190–1195, 2001.
- [9] ZITZLER, E.; THIELE, L. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *Evolutionary Computation, IEEE Transactions on*, v. 3, n. 4, p. 257–271, Nov 1999.
- [10] DEB, K. Solving goal programming problems using multi-objective genetic algorithms. *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, v. 1, p. –84 Vol. 1, 1999.
- [11] CASTRO, L. N. de. *Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications*. 2006. Chapman Hall; 1 edition.

- [12] GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. 1989. Addison-Wesley Publishing Company, Inc.
- [13] FONSECA, C. M.; FLEMING, P. J. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In: *Genetic Algorithms: Proceedings of the Fifth International Conference*. [S.l.]: Morgan Kaufmann, 1993. p. 416–423.
- [14] DEB, K.; AGRAWAL, R. B. Simulated binary crossover for continuous search space. *Complex Syst.*, v. 9, p. 115–148, April 1995.
- [15] RAGHUWANSHI, M.; KAKDE, O. Survey on multiobjective evolutionary and real coded genetic algorithms.
- [16] YOON, S.-B. et al. Robust shape optimization of electromechanical devices. *Magnetics, IEEE Transactions on*, v. 35, n. 3, p. 1710–1713, May 1999.
- [17] HACKER, K.; LEWIS, K. Robust design through the use of a hybrid genetic algorithm. *Proceedings of the ASME Design Engineering Technical Conference*, v. 2, p. 702–12, 2002.
- [18] PIERGIORGIO, A. et al. Robust target functions in electromagnetic design. *COMPEL: Int J for Computation and Maths. in Electrical and Electronic Eng.*, v. 22, p. 549–560(12), 18 June 2003.
- [19] CAVALIERE, V. et al. Robust design of high field magnets through monte carlo analysis. *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, v. 22, n. 3, p. 589–602, 2003.
- [20] TAKAHASHI, R. et al. Sensitivity analysis for optimization problems solved by stochastic methods. *Magnetics, IEEE Transactions on*, v. 37, n. 5, p. 3566–3569, Sep 2001.
- [21] AVILA, S. et al. Sensitivity analysis applied to decision making in multiobjective evolutionary optimization. *Magnetics, IEEE Transactions on*, v. 42, n. 4, p. 1103–1106, April 2006.
- [22] GUIMARAES, F.; LOWTHER, D.; RAMIREZ, J. Multiobjective approaches for robust electromagnetic design. *Magnetics, IEEE Transactions on*, v. 42, n. 4, p. 1207–1210, April 2006.
- [23] SPAGNUOLO, G. Worst case tolerance design of magnetic devices by evolutionary algorithms. *Magnetics, IEEE Transactions on*, v. 39, n. 5, p. 2170–2178, Sept. 2003.
- [24] CIOFFI, M.; FORMISANO, A.; MARTONE, R. Stochastic handling of tolerances in robust magnets design. *Magnetics, IEEE Transactions on*, v. 40, n. 2, p. 1252–1255, March 2004.

- [25] STEINER, G. et al. Statistical robust design using the unscented transformation. *COMPEL: Int J for Computation and Maths. in Electrical and Electronic Eng.*, v. 24, p. 606–619(14), 1 February 2005.
- [26] TSUTSUI, S. A comparative study on the effects of adding perturbations to phenotypic parameters in genetic algorithms with a robust solution searching scheme. *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, v. 3, p. 585–591 vol.3, 1999.
- [27] MARUYAMA, T.; IGARASHI, H. An effective robust optimization based on genetic algorithm. *Magnetics, IEEE Transactions on*, v. 44, n. 3, p. 990–993, June 2008.
- [28] MENEZES, L. de et al. Efficient computation of stochastic electromagnetic problems using unscented transforms. *Science, Measurement and Technology, IET*, v. 2, n. 2, p. 88–95, March 2008.
- [29] MENEZES, L. de et al. Efficient extraction of statistical moments in electromagnetic problems solved with the method of moments. *Microwave and Optoelectronics Conference, 2007. IMOC 2007. SBMO/IEEE MTT-S International*, p. 757–760, 29 2007–Nov. 1 2007.
- [30] PEREIRA, J.; MENEZES, L. de; BORGES, G. Statistical analysis of induced ground voltage using the tlm+ut method. *Electromagnetic Compatibility, 2008. EMC 2008. IEEE International Symposium on*, p. 1–4, Aug. 2008.
- [31] CIOFFI, M. et al. A fast method for statistical robust optimization. *Magnetics, IEEE Transactions on*, v. 42, n. 4, p. 1099–1102, April 2006.
- [32] WEBER, R. J. *Introduction to Microwave Circuits: Radio Frequency and Design Applications*. 2001. IEEE Press Series on RF and Microwave Technology Roger, D. Pollard and Richard Booton, Series Editors.
- [33] CASTRO, L. F. *Otimização de Amplificadores Classes E/F em Tecnologia CMOS utilizando-se Algoritmo Genético e Técnica de Equilíbrio Harmônico*. 2007. Dissertação de Mestrado em Engenharia Elétrica. Universidade de Brasília.

I. ARQUIVOS UTILIZADOS PELO GOLDEN GATE

Listagem I.1: input.scs

```
// Generated for: spectre
// Generated on: Aug 19 11:07:21 2006
// Design library name: amp_pot
// Design cell name: amp_pot
5 // Design view name: schematic
simulator lang=spectre
global 0
parameters entrada=1.2 alimentacao=2.4//1.012 //BIAS //alimentacao=0.78
include "/tools/cadence/AMS_3.60_CDS/spectre/c35/mcparams.scs"
10 include "/tools/cadence/AMS_3.60_CDS/spectre/c35/cmos53.scs" section=cmostm
include "/tools/cadence/AMS_3.60_CDS/spectre/c35/res.scs" section=restm
include "/tools/cadence/AMS_3.60_CDS/spectre/c35/cap.scs" section=captm
include "/tools/cadence/AMS_3.60_CDS/spectre/c35/bip.scs" section=biptm
include "/tools/cadence/AMS_3.60_CDS/spectre/c35/ind.scs" section=indtm
15
// Library name: amp_pot
// Cell name: amp_pot
// View name: schematic

20 Vin (netG 0) vsource type=pulse val0=0.0 val1=entrada period= 1/freq \
    delay=0 rise=2p fall=2p width=(1/(freq*2))-4p
MNO (netD netG 0 0) modn w=5000u l=0.4u as=4.25e-09 ad=4.25e-09 \
    ps=5001.7000u pd=5001.7000u nrd=0.0001 nrs=0.0001 ng=1

25 Val (netVAL 0) vsource dc=BIAS mag=0 type=dc

Lrfc (netVAL netD) inductor l =Lobst

Lo (netD net02) inductor l =Lsaida
30 Co (net02 net03) capacitor c =Csaida

LX (net03 netOUT) inductor l =Lxis

C (netD 0) capacitor c =Cshunt
```

35

```
Rout (netOUT 0) resistor r = 50
```

```
simulatorOptions options reltol=100e-6 vabstol=1e-6 iabstol=1e-12 temp=27 \  
    tnom=27 homotopy=all limit=delta scalem=1.0 scale=1.0 \  
40 compatible=spice2 gmin=1e-12 rforce=1 maxnotes=5 maxwarns=5 digits=5 \  
    cols=80 pivrel=1e-3 ckptclock=1800 sensfile="../psf/sens.output"  
tran tran stop=30n errpreset=conservative write="spectre.ic" \  
    writefinal="spectre.fc" annotate=status maxiters=5  
finalTimeOP info what=oppoint where=rawfile  
45 designParamVals info what=parameters where=rawfile  
saveOptions options save=allpub
```

Listagem I.2: hb.gg

```
#include "input.scs"
```

```
simulation hb
```

```
{
```

```
5 variable:
```

```
{
```

```
main sweep freq {nom=0.92125G, min=0.92125G, max=0.92125G, step=0.025G};
```

```
main optimization BIAS {nom = 1.0, min = 0.5, max = 1.5};
```

```
10 main optimization Lobst {nom = 86.0n, min = 10.0n, max = 100.0n};
```

```
main optimization Lsaida {nom = 76.0n, min = 10.0n, max = 100.0n};
```

```
main optimization Csaida {nom = 0.39p, min = 0.039p, max = 3.9p};
```

```
main optimization Lxis {nom = 10n, min = 1.0n, max = 100.0n};
```

```
main optimization Cshunt {nom = 0.63p, min = 0.063p, max = 0.63p};
```

```
15
```

```
//main numeric BIAS {nom = 8.713756e-01};
```

```
//main numeric Lobst {nom = 1.000578e-08};
```

```
//main numeric Lsaida {nom = 1.693895e-08};
```

```
20 //main numeric Csaida {nom = 1.953401e-12};
```

```
//main numeric Lxis {nom = 3.824844e-09};
```

```
//main numeric Cshunt {nom = 6.412948e-14};
```

```
};
```

```
25
```

```
probe:
```

```
{
```

```

#instance pmeter output (netOUT, 0)    { elmt = "Rout" };
#instance pmeter input  (netG, 0)      { elmt = "Vin" };
30 #instance pmeter trans (netD, 0)     { elmt = "MN0" };
#instance pmeter alimen (0, netVAL)    { elmt = "Val" };
};

analysis:
35 {
    harmonic_balance {
        tnom = 27.000000, temp = 27.000000, print_dc_state = 2,
        gmin_stepping = 1, dc_max_iter = 500, iabstol = 1.000000e-12,
        optimization_cont = 1, gmin = 1.000000e-12, conv_err = 0.100000,
40 gcomp = 1.000000e-12, stepping = 1, scale = 1.000000e+00,
        print_inventory = 1, print_parameter = 0, print_variable = 0, print_perf =
            1,
        process_spare = 0, log_newton_step = 0,
        deprecated_par_behavior = 1,
        err_print = 2, disable_probes = 0, hb_estim = 2, initial_transient = 0,
45 force_hb_converge = 0, osc_start_noise = 1.000000e-04, calc_dc = 1, algo =
            3,
        hb_krylov_residu = 1.0e-04, preserve_hb_memory = 0,
            krylov_precond_diag_size = 0,
        krylov_max_iter = 100, over_samp = 2, intermod_order_trunc = 0,
        multitone_fft = 2, osc_anal = 1, undef_par_behavior=1
    };
50 };

signal:
{
    rf_carrier { nb_tone = 1, frq1 = freq, nh1 = 10 };
55 };

specification:
{
    specification_variable { var = [ freq ] };
60

    perf Efficiency { expr = abs(output.p(1,0,0)) / abs(alimen.p(0,0,0)) };
    perf Power_OUT_Watt { expr = abs(output.p(1,0,0)) };

    goal Power_OUT { perf="Power_OUT_Watt", type="equal", value=10e-3};

```

```
65     goal Efficiency_Max {perf="Efficiency", type="equal", value=1.0};
    };

    task:
    {
70     optimization
        {
            var = [
                BIAS , Lobst , Lsaida , Csaida , Lxis , Cshunt
            ]
75     , max_trial = 200
        , max_iter = 200
        , cp=1
        , NumInd=20
        };
80     };
};
```
