

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA  
CENTRO DE PESQUISA E DESENVOLVIMENTO EM ENGENHARIA ELÉTRICA

## **Operadores para Algoritmos Genéticos baseados em aproximações quadráticas de funções de variáveis contínuas**

**Elizabeth Fialho Wanner**

Texto de doutorado submetido à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, como requisito final para obtenção do título de Doutor em Engenharia Elétrica.

Orientador: Prof. Ricardo Hiroshi Caldeira Takahashi

Co-orientador: Prof. Rodney Resende Saldanha

Belo Horizonte, 01 de Setembro de 2006



# Agradecimentos

Normalmente, eu não consigo encontrar uma boa maneira de dizer obrigado: costumo dizer rápido e em voz baixa. Entretanto, como essa tese tem mais de cem páginas, não pretendo economizar espaço nesta seção. Assim, a todos aqueles que me ajudaram nesses últimos anos, aqui vai o meu “muito obrigada”.

Em primeiro lugar, eu gostaria de agradecer ao meu orientador, Ricardo Takahashi. Ele tem sempre me proporcionado críticas construtivas, conselhos e palavras de encorajamento, assim como também muitas oportunidades de desenvolvimento profissional e acadêmico e ajuda financeira. E, acima de tudo, não podia deixar de agradecer a paciência que ele tem demonstrado ao longo desses anos. Há alguns anos atrás, eu era uma matemática teórica dando os primeiros passos no mundo da engenharia. E ninguém pode imaginar quão trôpegos eram esses passos! Mas, pouco a pouco, com muita paciência e gentileza, Ricardo foi me guiando nesse novo caminho que escolhi. Sem ele, eu não teria chegado aqui.

Também gostaria de agradecer aos meus co-orientadores: Rodney Saldanha e Peter Fleming. O Rodney me proporcionou momentos descontraídos e me mostrou o lado fantástico das “aplicações à engenharia”. Peter Fleming, ao me aceitar no seu grupo em Sheffield, me propiciou um dos melhores anos da minha vida. Nesse tempo, pude me dedicar inteiramente ao meu trabalho e me apaixonar pelas técnicas evolucionárias.

Não posso deixar de agradecer ao Frederico Gadelha Guimarães. Sempre presente, tivemos ótimas discussões, algumas estando a milhares de quilômetros de distância um do outro (ah, Yahoo Messenger, muito obrigado!). Como aprendi com esse garoto! Posso dizer que formamos um bela dupla de trabalho!

A todos os meus colegas, da UFMG e de Sheffield, agradeço pelo incentivo e companheirismo. Vocês tornaram minha vida universitária mais agradável. Aos amigos, fora do meio universitário, pela compreensão por estar sempre ausente.

Às secretárias do CPDEE e de Sheffield, em especial à Anete, aqui vai meu muito obrigada. Vocês facilitaram, e muito, a minha vida!

À Capes, pelo apoio financeiro nos últimos dois anos desse trabalho.

Agradeço também aos meus pais, irmão e demais familiares pelo apoio e amor incondicional.

E finalmente, agradeço às duas pessoas mais importantes na minha vida: minha filha Bárbara e meu marido Paulo Sérgio. Por diversas vezes, deixei de dar atenção a vocês, não estive presente em momentos especiais na vida da minha filha. O Paulo Sérgio sempre foi meu espelho e sempre me encorajou a seguir em frente. Divido com vocês dois essa grande conquista. Sem o amor de vocês, nada teria sido possível.



# Resumo

Esta tese investiga a possibilidade do uso de aproximações quadráticas de funções para a construção de novos operadores para uso em algoritmos genéticos, aplicados à otimização de funções de variáveis contínuas. A fórmula básica empregada em todos os casos é a do aproveitamento do conjunto de amostras das funções-objetivo e das funções de restrição do problema que já é normalmente obtido por meio da execução das operações típicas dos algoritmos genéticos. Com esse conjunto de amostras, as aproximações quadráticas das diversas funções são obtidas e, à medida em que o algoritmo genético prossegue obtendo novas amostras, tais aproximações são atualizadas. São aqui propostos, com fundamento em tais aproximações: (i) um operador de correção das coordenadas do espaço de variáveis; (ii) um operador especializado em obter estimativas do ótimo de problemas mono-objetivo com uma única restrição de igualdade; (iii) um operador especializado na obtenção de estimativas do ótimo de problemas mono-objetivo com múltiplas restrições de desigualdade; (iv) um operador especializado na obtenção de estimativas localmente refinadas de pontos pertencentes ao conjunto Pareto-ótimo de problemas multiobjetivo irrestritos; (v) um operador especializado na obtenção de estimativas localmente refinadas de pontos pertencentes ao conjunto Pareto-ótimo de problemas multiobjetivo com restrições de desigualdade. Os três últimos operadores são construídos com base em uma formulação de Desigualdades Matriciais Lineares (LMI's). Como sub-produto desta tese, é proposta ainda uma nova métrica para comparar os desempenhos de algoritmos de otimização multiobjetivo na tarefa de obtenção de amostragens representativas dos conjuntos Pareto-ótimos de problemas, a métrica da *contagem de esferas*. Os resultados obtidos indicam que todos os operadores propostos são capazes de conduzir a melhorias significativas, tanto na velocidade de convergência quanto na precisão das soluções obtidas. Estudos adicionais se fazem necessários, no caso dos operadores multiobjetivo, para aumentar a extensão dos conjuntos de estimativas obtidas do conjunto Pareto-ótimo.



# Abstract

This thesis investigates the possibility of using quadratic approximations of functions with the purpose of building new operators for genetic algorithms, applied to the optimization of continuous variable functions. The basic formulation used in all cases is the employment of the set of samples of objective functions and constraint functions that is already obtained through the execution of the typical operations of genetic algorithms. With this set of samples, the quadratic approximations of the several functions are calculated and, as the genetic algorithm goes getting new samples, such approximations are updated. This thesis proposes, upon such approximations: (i) an operator that performs the coordinate correction of the variable space; (ii) an operator that generates estimates of the optimum for mono-objective problems with a single constraint; (iii) an operator that generates estimates of the optimum for mono-objective problems with several inequality constraints; (iv) an operator that generates locally refined estimates of the Pareto-set points of unconstrained multiobjective problems; and (v) an operator that generates locally refined estimates of the Pareto-set points of multiobjective problems with multiple inequality constraints. The three last operators are built on the basis of Linear Matrix Inequality (LMI) formulations. As a by-product of this thesis, a new metric is proposed here, for the purpose of comparing the performances of multiobjective optimization algorithms in the task of generating representative sample sets for the Pareto-optimal sets of multiobjective problems: the *sphere-counting* metric. The results obtained suggest that all proposed operators are capable of leading to significant enhancements in the convergence rate, in the proportion of convergence and in the solution precision. Additional studies are necessary, in the case of the multiobjective operators, for enhancing the extension of the Pareto-estimate sets that are obtained.





# Lista de Acrônimos

AE – Algoritmo Evolucionário

AEAQ – Algoritmo Elipsoidal com Aproximação Quadrática

AG – Algoritmo Genético

AGRP – Algoritmo Genético Real Polarizado

AGRP-AQR – Algoritmo Genético Real Polarizado com o operador de Aproximação Quadrática para Ótimo Restrito

AQR – Aproximação Quadrática para Ótimo Restrito

EE – Estratégia Evolutiva

FA – Função de Aptidão

LMI – Linear Matrix Inequality

mcAG – Algoritmo Genético com Mudança de Coordenada

MOGA – Multiobjective Genetic Algorithm

MOGA-H – Multiobjective Genetic Algorithm Híbrido

NDCSR – Non-Dominated Combined Set Ratio

$p_c$  – Probabilidade de Cruzamento

$p_m$  – Probabilidade de Mutação

$p_p$  – Probabilidade de Polarização

PG – Programação Genética

sAG – Algoritmo Genético Simples

SPEA – Strenght Pareto Evolutionary Algorithm

SPEA 2 – Strenght Pareto Evolutionary Algorithm 2

VEGA – Vector Evaluated Genetic Algorithm

$\xi$  – Fator de Extrapolação no Cruzamento Polarizado

$\gamma$  – Fator de Dispersão da Função de Aptidão



# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Aproximações Quadráticas</b>	<b>7</b>
2.1	Métodos que Utilizam Aproximações Quadráticas . . . . .	9
2.1.1	Método de Newton . . . . .	10
2.1.2	Métodos quasi-Newton . . . . .	10
2.1.3	Programação Quadrática Sequencial . . . . .	12
2.1.4	Métodos de Região de Confiança . . . . .	13
2.1.5	Método Elipsoidal . . . . .	14
2.2	Metodologias Baseadas em Gradiente . . . . .	15
2.2.1	Método Direto . . . . .	15
2.2.2	Métodos Recursivos . . . . .	17
2.3	Método Livre de Derivadas . . . . .	17
2.3.1	Metodologia . . . . .	18
2.3.2	Custo Computacional . . . . .	20
<b>3</b>	<b>Algoritmo Genético</b>	<b>23</b>
3.1	Operadores . . . . .	25
3.1.1	Função de Aptidão . . . . .	25
3.1.2	Seleção . . . . .	26
3.1.3	Cruzamento Real Polarizado . . . . .	27
3.1.4	Reflexão . . . . .	28
3.1.5	Mutação . . . . .	28
3.1.6	Elitismo . . . . .	29
3.2	Parâmetros . . . . .	29
<b>4</b>	<b>Aproximações Quadráticas em AG Mono-Objetivo</b>	<b>31</b>
4.1	Definições Básicas . . . . .	32

4.1.1	Condições de Otimalidade . . . . .	32
4.1.2	Tratamento de Restrições nos AGs . . . . .	33
4.2	Restrições de Igualdade . . . . .	35
4.2.1	O operador de Busca Local . . . . .	36
4.2.2	AGRP-AQR Híbrido . . . . .	38
4.2.3	Testes e Resultados . . . . .	42
4.2.4	Conclusões . . . . .	48
4.3	Restrições de Desigualdade . . . . .	48
4.3.1	Obtendo as Aproximações . . . . .	49
4.3.2	O operador de Busca Local . . . . .	51
4.3.3	Hibridizando o AG . . . . .	53
4.3.4	Testes e Resultados . . . . .	54
4.3.5	Conclusões . . . . .	62
4.4	Mudança de Coordenada no AG . . . . .	68
4.4.1	Metodologia . . . . .	69
4.4.2	Testes e Resultados . . . . .	72
4.4.3	Conclusões . . . . .	80
<b>5</b>	<b>AG Multiobjetivo</b>	<b>83</b>
5.1	Definições Básicas . . . . .	84
5.2	AG Multiobjetivo . . . . .	86
5.2.1	MOGA: Multi-Objective Genetic Algorithm . . . . .	88
5.2.2	SPEA 2: Strenght Pareto Evolutionary Algorithm 2 . . . . .	89
5.3	Medidas de Desempenho . . . . .	91
5.3.1	Testes Estatísticos . . . . .	94
5.4	Problemas Irrestritos . . . . .	95
5.4.1	Metodologia . . . . .	96
5.4.2	Acoplando a Busca Local . . . . .	102
5.4.3	Testes e Resultados . . . . .	104
5.4.4	Conclusões . . . . .	107
5.5	Problemas Restritos . . . . .	109
5.5.1	Metodologia . . . . .	109
5.5.2	Acoplando a Busca Local . . . . .	113
5.5.3	Testes e Resultados . . . . .	113
5.5.4	Conclusões . . . . .	119
<b>6</b>	<b>Conclusões</b>	<b>127</b>

<b>A</b>	<b>Modificações no Algoritmo Elipsoidal</b>	<b>131</b>
A.1	Formulação do Problema . . . . .	132
A.2	O Algoritmo Elipsoidal Clássico . . . . .	132
A.3	Tratamento de Restrições de Igualdade . . . . .	134
A.4	Algoritmo Elipsoidal com Aproximações Quadráticas . . . . .	135
A.4.1	Problemas Irrestritos . . . . .	136
A.4.2	Problemas com Restrições de Igualdade . . . . .	137
A.5	Testes Numéricos . . . . .	138
A.6	Conclusão . . . . .	141
<b>B</b>	<b>Metodologia LMI</b>	<b>143</b>



# Lista de Figuras

3.1	Esquema básico dos algoritmos da Computação Evolutiva. . . . .	24
3.2	Estrutura básica de um AG tradicional. . . . .	26
4.1	No ponto $x^*$ os gradientes de duas restrições ativas, $\nabla g_1$ e $\nabla g_2$ , e o gradiente da função-objetivo $\nabla f$ estão representados. Neste ponto as condições de Karush-Kuhn-Tucker são satisfeitas, e qualquer perturbação em $x^*$ irá causar um aumento do valor da função-objetivo ou a infactibilidade. . . . .	33
4.2	Curvas de nível de $f$ com a restrição não-linear de igualdade e a solução $x^*$ em um problema do tipo (4.13). . . . .	39
4.3	Caminho, unindo $x_f$ e $x_g$ , que contém $x^*$ , representado na figura pelo círculo preto. . . . .	39
4.4	Esquema básico do método de bisseção utilizado no operador de busca local para problemas com restrição de igualdade. . . . .	40
4.5	Esquema básico do operador de busca local especializado em problemas com restrição não-linear de igualdade. . . . .	41
4.6	Esquema básico de um AG híbrido para tratamento de restrição não-linear de igualdade. . . . .	42
4.7	Curva de convergência para o Problema Quadrático (P1) obtida na resolução de um problema mono-objetivo com uma restrição de igualdade. . . . .	44
4.8	Curva de convergência para o Problema Não-Quadrático (P2) obtida na resolução de um problema mono-objetivo com uma restrição de igualdade. . . . .	45
4.9	Curva de convergência para o Problema Multimodal (P3) obtida na resolução de um problema mono-objetivo com uma restrição de igualdade. . . . .	45
4.10	Curva de convergência para o Problema com Restrição Não-Quadrática (P4) obtida na resolução de um problema mono-objetivo com uma restrição de igualdade. . . . .	46
4.11	Análise da diversidade ao utilizar o operador AQR no problema Rastrigin bidimensional com uma restrição de igualdade. . . . .	47

4.12	Análise da diversidade ao utilizar o operador AQR no problema Rastrigin tridimensional com uma restrição de igualdade. . . . .	48
4.13	Esquema básico de um operador de busca local para problemas com múltiplas restrições de desigualdades. . . . .	53
4.14	Esquema básico de um algoritmo genético híbrido para tratamento de restrições de desigualdade. . . . .	54
4.15	Curva de convergência para o problema (T1). . . . .	63
4.16	Curva de convergência para o problema (T2). . . . .	63
4.17	Curva de convergência para o problema (T3). . . . .	64
4.18	Curva de convergência para o problema (T4). . . . .	64
4.19	Curva de convergência para o problema (T5). . . . .	65
4.20	Curva de convergência para o problema (T6). . . . .	65
4.21	Curva de convergência para o problema (T7). . . . .	66
4.22	Curva de convergência para o problema (T8). . . . .	66
4.23	Curva de convergência para o problema (T9). . . . .	67
4.24	Esta figura mostra um ponto $x_k$ no espaço de coordenadas original. O ponto $x_k$ está situado na curva de nível (linha sólida) de uma função-objetivo quadrática e convexa, que possui a forma de uma elipse alongada $\Phi$ . $x_k$ é também o centro de um círculo $\Omega_1$ de raio $\rho$ . Se $x_k$ está sujeito a uma perturbação de comprimento menor ou igual a $\rho$ , este ponto pode cair dentro de $\Omega_1$ . Se o ponto cai dentro de $\Omega_1 \cap \Phi$ , o ponto perturbado terá um valor melhor para a função-objetivo. . . . .	69
4.25	Esta figura mostra um ponto $\bar{x}_k$ , que corresponde a $x_k$ , nas coordenadas após a mudança. O ponto $\bar{x}_k$ pertence à curva de nível (linha sólida) da função-objetivo, que é agora um círculo $\bar{\Phi}$ . $\bar{x}_k$ está agora no centro de um círculo $\Omega_2$ de raio $\rho$ . Se $\bar{x}_k$ está sujeito a uma perturbação menor ou igual $\rho$ , este ponto pode cair sobre algum ponto dentro de $\Omega_2$ . Se o ponto cai dentro de $\Omega_2 \cap \bar{\Phi}$ , o ponto perturbado terá um melhor valor de função-objetivo. Observe que o volume do conjunto $\Omega_2 \cap \bar{\Phi}$ é maior que o volume do conjunto $\Omega_1 \cap \Phi$ , o que significa que a probabilidade de melhora do valor da função-objetivo após uma perturbação em $\bar{x}_k$ é maior que a probabilidade de melhora para $x_k$ com uma perturbação similar. . . . .	69
4.26	Esquema básico do procedimento de mudança de coordenadas. . . . .	71
4.27	Porcentagem de vezes que cada algoritmo alcançou o mínimo em cada problema. A primeira barra representa o <i>sAG</i> e a segunda barra representa o <i>mcGA</i> . . . . .	75



4.28	Porcentagem de vezes que cada algoritmo alcançou o mínimo na função de Rosenbrock no experimento no qual a população inicial não continha o ponto de mínimo. A primeira barra representa o <i>sAG</i> e a segunda barra representa o <i>mcAG</i> . Neste caso, cada algoritmo foi executado 200 vezes. Para o primeiro intervalo, a população foi de 20 indivíduos e no segundo intervalo, a população foi de 50 indivíduos. . . . .	76
4.29	Curva de convergência para a função de Rastrigin. Em 30 testes, o <i>sAG</i> encontrou o ponto de mínimo 4 vezes e o <i>mcAG</i> 14 vezes. . . . .	78
4.30	Curva de convergência para a função de Schwefel. Em 30 testes, o <i>sAG</i> encontrou o ponto de mínimo 9 vezes e o <i>mcAG</i> 19 vezes. . . . .	79
4.31	Curva de convergência para a função de Rosenbrock. Em 30 testes, o <i>sAG</i> encontrou o ponto de mínimo 6 vezes e o <i>mcAG</i> 18 vezes. . . . .	79
4.32	Curva de convergência para o problema de dimensão 30 (4.60). . . . .	80
5.1	Esquema básico de um algoritmo genético multiobjetivo. . . . .	87
5.2	Contagem de esferas em um problema bi-objetivo. Observe que o conjunto de Pareto possui 20 pontos, mas a contagem de esferas resulta em apenas 9 esferas. . . . .	93
5.3	Em um problema bi-objetivo, a S metric calcula a área da região poligonal coberta pelos pontos $z_1, z_2, z_3$ e um ponto de referência $z_{ref}$ . . . . .	93
5.4	Histograma de frequência obtido através do teste de aleatoriedade. O valor observado está indicado pelo círculo preto. Os resultados são favoráveis para o algoritmo proposto se o resultado observado se encontrar no lado positivo do histograma. . . . .	95
5.5	Exemplo de uma sequência de quatro passos para se obter pontos bem distribuídos sobre o conjunto de Pareto, sendo que cada figura representa um determinado passo. Os pontos $z^*_1$ e $z^*_2$ correspondem aos pontos de mínimo de duas funções-objetivo. O novo ponto calculado em cada passo está representado com um círculo preto. No primeiro passo (primeira figura), o ponto $z_1$ foi obtido escolhendo-se $\gamma$ no intervalo $[0, 1]$ . No segundo passo (segunda figura), como a distância entre $z_1$ e $z^*_2$ é maior do que a distância entre $z_1$ e $z^*_1$ , o novo ponto foi calculado fazendo $\gamma_2 = \frac{\gamma_1+1}{2}$ . No terceiro passo (terceira figura), como a distância entre $z_2$ e $z^*_2$ é maior do que a distância entre $z_2$ e $z^*_1$ , o novo ponto foi calculado fazendo $\gamma_3 = \frac{\gamma_2+1}{2}$ . No quarto passo (quarta figura), como a distância entre $z_3$ e $z_2$ é maior do que a distância entre $z_3$ e $z^*_2$ , o novo ponto foi calculado fazendo $\gamma_4 = \frac{\gamma_2+\gamma_3}{2}$ . . . . .	99

5.6	Caminho paramétrico, dado pela equação (5.22), unindo os pontos de mínimos de cada função-objetivo do problema bi-objetivo (5.21) . . . . .	101
5.7	Região paramétrica, dada pela equação (5.24), unindo os pontos de mínimos das funções-objetivo do problema tri-objetivo (5.23). Os pontos em cada borda do triângulo são obtidos variando-se $\gamma_i$ individualmente no intervalo $[0, 1]$ . . . . .	101
5.8	Etapas necessárias para acoplar a busca local ao SPEA 2. . . . .	103
5.9	Conjunto Pareto, no espaço de objetivos, de uma execução aleatoriamente escolhida dos SPEA 2 e do SPEA 2 híbrido (SPEA H) no problema KUR com 3 variáveis. Apenas um ponto foi selecionado para a busca local no algoritmo híbrido. . . . .	106
5.10	Conjunto Pareto, no espaço de objetivos, de uma execução aleatoriamente escolhida dos SPEA 2 e do SPEA 2 híbrido (SPEA H) no problema KUR com 3 variáveis. Quatro pontos foram selecionados para a busca local no algoritmo híbrido. . . . .	106
5.11	Conjunto Pareto, no espaço de objetivos, de uma execução aleatoriamente escolhida dos SPEA 2 e do SPEA 2 híbrido (SPEA H) no problema KUR com 10 variáveis. Quatro pontos foram selecionados para a busca local no algoritmo híbrido. . . . .	108
5.12	Esquema básico do operador de busca local para problemas multiobjetivo restritos. . . . .	112
5.13	Esta figura ilustra o procedimento de busca local em um problema tri-objetivo quadrático e irrestrito. . . . .	112
5.14	Porcentagem de soluções não-dominadas em cada execução do problema Quadrático (prob. 5.36). A primeira barra corresponde ao MOGA puro e a segunda barra corresponde ao MOGA híbrido. . . . .	116
5.15	Porcentagem de soluções não-dominadas em cada execução do problema Não-Quadrático (prob. 5.37). A primeira barra corresponde ao MOGA puro e a segunda barra corresponde ao MOGA híbrido. . . . .	116
5.16	Porcentagem de soluções não-dominadas em cada execução do problema KUR (prob. 5.38). A primeira barra corresponde ao MOGA puro e a segunda barra corresponde ao MOGA híbrido. . . . .	117
5.17	Porcentagem de soluções não-dominadas em cada execução do problema Multimodal (prob. 5.39). A primeira barra corresponde ao MOGA puro e a segunda barra corresponde ao MOGA híbrido. . . . .	117

5.18	Teste de aleatoriedade, usando a contagem de esferas, para o problema Quadrático (prob. 5.36). O resultado é estatisticamente significativo uma vez que o círculo preto se encontra no lado positivo do histograma. . . . .	119
5.19	Teste de aleatoriedade, usando a contagem de esferas, para o problema Não-Quadrático (prob. 5.37). O resultado é estatisticamente significativo uma vez que o círculo preto se encontra no lado positivo do histograma. . . . .	120
5.20	Teste de aleatoriedade, usando a contagem de esferas, para o problema KUR (prob. 5.38). O resultado é estatisticamente significativo uma vez que o círculo preto se encontra no lado positivo do histograma. . . . .	120
5.21	Teste de aleatoriedade, usando a contagem de esferas, para o problema Multimodal (prob. 5.39). O resultado é estatisticamente significativo uma vez que o círculo preto se encontra no lado positivo do histograma. . . . .	122
5.22	Teste de aleatoriedade, usando a S Metric, para o problema Quadrático (prob. 5.36). O resultado é estatisticamente significativo uma vez que o círculo preto se encontra no lado positivo do histograma. . . . .	122
5.23	Teste de aleatoriedade, usando a S Metric, para o problema KUR (prob. 5.38). O resultado é estatisticamente significativo uma vez que o círculo preto se encontra no lado positivo do histograma. . . . .	123
5.24	Teste de aleatoriedade, usando a S Metric, para o problema Multimodal (prob. 5.39). O resultado não é estatisticamente significativo uma vez que o círculo preto se encontra no meio do histograma. . . . .	123
5.25	Conjunto Pareto-ótimo da execução do MOGA que apresentou melhor valor de S Metric no problema KUR (prob. 5.38). Observe a presença de ponto isolado e regiões desconexas. . . . .	124
5.26	Conjunto Pareto-ótimo da execução do MOGA Híbrido que apresentou melhor valor de S Metric no problema KUR (prob. 5.38). Observe a presença de ponto isolado e regiões desconexas. . . . .	124
5.27	Conjunto Pareto-ótimo da execução do MOGA que apresentou melhor valor da contagem de esferas no problema Multimodal (prob. 5.39). . . . .	125
5.28	Conjunto Pareto-ótimo da execução do MOGA Híbrido que apresentou melhor valor da contagem de esferas no problema Multimodal (prob. 5.39). . . . .	125
5.29	Conjunto Pareto-ótimo da execução do MOGA que apresentou melhor valor da contagem de esferas no problema Multimodal (prob. 5.39) utilizando-se o novo raio para as esferas. . . . .	126

5.30	Conjunto Pareto-ótimo da execução do MOGA Híbrido que apresentou melhor valor da contagem de esferas no problema Multimodal (prob. 5.39) utilizando-se o novo raio para as esferas. . . . .	126
A.1	Corte convencional em um elipsoíde em um problema bi-dimensional. . . . .	133
A.2	Evolução da terceira coordenada de $x_k$ ( $\times$ ) e $x_k^e$ ( $\circ$ ) em um problema quadrático irrestrito de dimensão 10. . . . .	139
A.3	Detalhe ampliado da evolução da terceira coordenada de $x_k$ ( $\times$ ) e $x_k^e$ ( $\circ$ ) em um problema quadrático irrestrito de dimensão 10. . . . .	139
A.4	Evolução da primeira coordenada de $x_k$ ( $\times$ ) e $x_k^e$ ( $\circ$ ) em um problema quadrático restrito de dimensão 10. . . . .	140
A.5	Detalhe ampliado da evolução da primeira coordenada de $x_k$ ( $\times$ ) e $x_k^e$ ( $\circ$ ) em um problema quadrático restrito de dimensão 10. . . . .	140

# Lista de Tabelas

2.1	Tempo computacional médio, em segundos, em cada problema com dimensões variadas. . . . .	20
2.2	Número mínimo de pontos ( $n_m$ ) para a construção da aproximação quadrática em cada problema com dimensões variadas. . . . .	21
2.3	Tempo computacional médio (em segundos) para cada dimensão, variando-se o tamanho da amostra de $n_m$ até $10n_m$ para a construção da aproximação quadrática. . . . .	21
4.1	Valores médios do número de esferas na análise de diversidade em problemas mono-objetivo com restrições de igualdade. . . . .	47
4.2	Características de cada problema teste . . . . .	60
4.3	Resultados estatísticos obtidos pelo AG híbrido em cada problema . . . . .	61
4.4	Resultados estatísticos obtidos pelo AG simples em cada problema . . . . .	61
4.5	Comparação: Resultados obtidos na Literatura. . . . .	61
4.6	Comparação: Resultados obtidos nesse trabalho . . . . .	62
4.7	Número de gerações necessárias para alcançar o ponto de mínimo (NGen) e a porcentagem de vezes que cada algoritmo foi capaz de atingir a solução (P), para problemas quadráticos de dimensão 2, 3 e 4. Algoritmo 1 refere-se ao <i>sAG</i> e algoritmo 2 refere-se ao <i>mcAG</i> . . . . .	73
4.8	Melhor, pior e valor médio obtido pelo <i>sAG</i> e <i>ccGA</i> durante todas as execuções para a função de Rastringin. O símbolo $\sigma$ representa o desvio padrão para esse teste. . . . .	76
4.9	Melhor, pior e valor médio obtido pelo <i>sAG</i> e <i>ccGA</i> durante todas as execuções para a função de Schwefel. O símbolo $\sigma$ representa o desvio padrão para esse teste. . . . .	77
4.10	Melhor, pior e valor médio obtido pelo <i>sAG</i> e <i>ccGA</i> durante todas as execuções para a função de Rosenbrock. O símbolo $\sigma$ representa o desvio padrão para esse teste. . . . .	77

5.1	Parâmetros de cada algoritmo. . . . .	105
5.2	Parâmetros de cada algoritmo. . . . .	105
5.3	Valor médio de cada métrica para o problema Kur com 3 variáveis, selecionando-se apenas um ponto para a busca local. . . . .	105
5.4	Valor médio de cada métrica para o problema Kur com 10 variáveis, selecionando-se apenas um ponto para a busca local. . . . .	107
5.5	Valor médio de cada métrica para o problema Kur com 10 variáveis, selecionando-se quatro pontos para a busca local. . . . .	107
5.6	Valor médio, após 20 testes, do contador de esferas e da S Metric para o problema Quadrático (prob. 5.36). . . . .	115
5.7	Valor médio, após 20 testes, do contador de esferas e da S Metric para o problema Não-Quadrático (prob. 5.37). . . . .	115
5.8	Valor médio, após 20 testes, do contador de esferas e da S Metric para o problema KUR (prob. 5.38). . . . .	118
5.9	Valor médio, após 20 testes, do contador de esferas e da S Metric para o problema Multimodal (prob. 5.39). . . . .	118
A.1	Resultados Numéricos. . . . .	141

# Capítulo 1

## Introdução

Algoritmos evolucionários (AEs) constituem uma classe de métodos de otimização que receberam grande impulso a partir de 1980. São métodos heurísticos de busca, inspirados em analogias com a evolução natural, que utilizam uma *população* de *indivíduos* (soluções-tentativas) que evolui segundo regras de recorrência probabilísticas na direção de regiões melhores do espaço de busca. Os algoritmos evolucionários, via de regra, não são tão eficientes e precisos quanto os algoritmos determinísticos, quando esses estão disponíveis para resolver problemas de determinada classe. Entretanto, há diversas classes de problemas de grande interesse prático para as quais não são conhecidos algoritmos determinísticos capazes de tratá-los em tempo polinomial, ou mesmo de tratá-los, ou ainda que exigiriam grande esforço de adaptação dos algoritmos determinísticos para as características específicas do problema. Em muitos desses casos, os algoritmos evolucionários são capazes de fornecer soluções satisfatórias. Citamos aqui, dentre os problemas cujas variáveis são contínuas (aos quais se restringe o escopo desta tese), os problemas com funções-objetivo ou funções de restrição não-lineares descontínuas, não-diferenciáveis, multi-modais, e de múltiplas escalas: os algoritmos evolucionários são naturalmente adequados para tratar problemas com tais características, que tenderiam a causar dificuldades para a maioria dos métodos determinísticos convencionais. Os AEs são classificados como métodos de otimização global e são métodos robustos e efetivos quando se deseja encontrar um mínimo global aproximado. A natureza exploratória dos AEs permite uma identificação rápida de regiões promissoras no espaço de busca, mas, por outro lado, prejudica a convergência e precisão nos estágios finais do processo de busca. Nesses estágios finais pouca informação nova é obtida através de seus mecanismos de busca local, enquanto seus mecanismos de busca global introduzem perturbações que são muito grandes para permitir uma convergência com alta precisão (veja [1, 2]). Os AEs não utilizam informações locais do espaço de soluções, tais como derivadas, durante o processo de busca, o que faz com que esses algoritmos apresentem uma taxa de

convergência mais lenta se comparados às técnicas clássicas de busca local (veja [3]). Além disso, os AEs, por serem algoritmos irrestritos por natureza, possuem dificuldade de localizar o ponto de ótimo em problemas com restrições, principalmente de igualdade. Esta dificuldade é devida à inabilidade do AE em realizar uma busca em regiões de volume zero, regiões que possuem dimensão menor que o espaço de busca original. Uma alternativa para resolver esses problemas é combinar o AE com uma técnica que possibilite uma busca mais refinada por soluções mais precisas. Resultados promissores com vários tipos de algoritmos híbridos podem ser encontrados em [4, 5].

Os algoritmos evolucionários, no contexto dos problemas de variáveis contínuas, originalmente são capazes de tratar problemas de dimensão relativamente baixa e, em sua formulação original, não previam técnicas específicas para o tratamento de restrições. O desenvolvimento teórico dos algoritmos evolucionários, que vem sendo induzido pela crescente demanda por sua aplicação a problemas cada vez mais complexos, conduz a um programa de pesquisas da comunidade científica internacional, que vem focando entre suas prioridades:

- o desenvolvimento de técnicas para lidar com problemas de dimensões cada vez maiores;
- a criação de mecanismos eficientes para o tratamento de problemas com restrições não-lineares de desigualdade e de igualdade;
- a melhoria da precisão e da taxa de convergência desses algoritmos.

Cabe citar, em particular, a aplicabilidade dos algoritmos evolucionários a problemas de otimização multiobjetivo. Como os algoritmos evolucionários lidam com todo um conjunto de soluções-tentativa que evoluem simultaneamente, eles são capazes de produzir, em uma única execução, todo um conjunto de amostras do *conjunto Pareto-ótimo*, que é o objeto que se procura determinar em tal tipo de otimização. Dessa forma, os algoritmos evolucionários evitam a chamada *escalarização* do problema, que necessariamente ocorre nos algoritmos determinísticos, que são obrigados a produzir uma única estimativa amostral do conjunto Pareto-ótimo a cada execução. Isso significa que, quando comparados com os algoritmos determinísticos, a eficiência relativa dos algoritmos evolucionários fica multiplicada por um fator igual ao número de soluções obtidas ao mesmo tempo. Tal constatação vem motivando um intenso esforço de pesquisa da aplicação de algoritmos evolucionários em problemas multiobjetivo. Dentro do programa de pesquisas da comunidade científica sobre os algoritmos evolucionários encontram-se incluídos com destaque tópicos relacionados com a melhoria da eficiência e da qualidade da descrição do conjunto Pareto-ótimo de problemas de otimização multiobjetivo, dentre os quais citamos:



- a criação de mecanismos para descrever o conjunto Pareto-ótimo de uma maneira mais eficiente;
- a criação de mecanismos para descrever em detalhe regiões já esboçadas do conjunto Pareto-ótimo;
- a criação de critérios para comparação do desempenho de algoritmos.

Esta tese trata precisamente desses seis itens programáticos da pesquisa corrente no campo dos algoritmos evolucionários. As contribuições a seguir detalhadas procuram oferecer respostas às seis questões acima identificadas, a partir de uma única idéia-mestra que articula toda a formulação aqui desenvolvida: este trabalho propõe novos operadores baseados em aproximações quadráticas para as funções do problema, que podem ser utilizados em problemas mono e multiobjetivo, para o tratamento de restrições ou de funções-objetivo. Nos diversos casos, a utilização de aproximações quadráticas se mostra uma ferramenta eficaz para melhorar a convergência dos algoritmos, possibilitando a utilização desses em problemas de dimensão elevada. Além disso, mostrou-se capaz de guiar o processo de busca para regiões de volume zero, permitindo que os AEs sejam utilizados de uma maneira direta e eficiente em problemas com diversos tipos de restrições.

Técnicas de aproximação para funções em algoritmos de otimização têm sido largamente empregadas. A mais popular deles é o método conhecido como superfície de resposta. *response surface, no original em inglês* [6, 7, 8], a qual geralmente envolve regressão polinomial de ordem baixa. Outros métodos são aqueles que se baseiam nos modelos de Kriging [9, 10], em funções de base radial (RBFs) [11, 12] e em redes neurais [13]. Podemos ainda citar os modelos de aproximações quadráticas baseados na representação da função como uma série de Taylor truncada nos termos de ordem dois. Nesses casos a matriz das derivadas de segunda ordem, a matriz *Hessiana*, pode ser calculada através de fórmulas recursivas típicas de métodos quasi-Newton [14]. Um outro método de otimização que utiliza aproximações, porém no contexto de problemas explicitamente formulados no formato restrito, é conhecido como programação quadrática sequencial (SQP) [14] e baseia-se na minimização da função lagrangeana. Todos esses métodos tendem a ser iterativos no sentido de que a aproximação é iterativamente recalculada no decorrer do processo.

Após a construção da aproximação para a função a ser otimizada, essa aproximação pode produzir pontos mais promissores para serem avaliados pela função-objetivo original ou pode substituir a função original no processo de otimização. Essa última abordagem foi empregada em [15], no contexto de algoritmos evolucionários, com o propósito de acelerar o processo de otimização. Outros tipos de aproximações têm sido utilizados em algoritmos evolucionários para aproximar a função-objetivo por uma função analítica mais simples. Por exemplo, em

[16], os pontos aglomerados encontrados durante o processo de otimização foram utilizados para aproximar a função e, periodicamente, uma nova aproximação quadrática usando todos os pontos do espaço de busca era construída. Cada aproximação para a função substituía a função original no cálculo da função de ajuste.

Escolhemos neste trabalho uma aproximação de segunda ordem por ser esse o tipo de aproximação mais simples que permite obter informações sobre a concavidade da função. O processo de construção da aproximação quadrática para as funções apenas utiliza informações que já se tornam disponíveis através do próprio processo de execução usual dos algoritmos evolucionários: as coordenadas dos pontos e suas avaliações nas funções-objetivo e de restrição, respectivamente. Desta forma nenhuma avaliação extra da função se faz necessária para alimentar o processo de aproximação. Uma aproximação quadrática pode ser utilizada de duas formas:

- para corrigir as coordenadas do sistema, obtendo novas coordenadas mais favoráveis ao processo de otimização e
- para estimar as coordenadas de pontos factíveis ou do ponto de ótimo do problema de otimização.

O primeiro caso será aplicado, nesta tese, em problemas mono-objetivo. Será apresentado um operador de correção de coordenadas genérico, baseado em aproximações quadráticas, adequado para problemas mono-objetivo.

O segundo caso será aplicado em problemas mono-objetivo restritos, com restrições de igualdade e de desigualdade, e em problemas multiobjetivo restritos e irrestritos. Serão apresentados neste trabalho quatro operadores de busca local, baseados nas aproximações quadráticas, cada um especializado em problemas do tipo:

- problemas mono-objetivo com uma única restrição não-linear de igualdade;
- problemas mono-objetivo com várias restrições lineares e não-lineares de desigualdade;
- problemas multiobjetivo irrestritos;
- problemas multiobjetivo com várias restrições lineares e não-lineares de desigualdade.

Não há razões, *a priori*, para acreditar que os operadores aqui propostos sejam específicos para algum tipo de algoritmo evolucionário. Em princípio, sob o ponto de vista conceitual, parece razoável acoplar tais operadores a quaisquer algoritmos que se fundamentem na evolução de uma *população* de soluções-tentativa – ou seja, em quaisquer algoritmos evolucionários. No entanto, para o propósito de delimitar o escopo das análises aqui realizadas,

nesta tese foram empregadas apenas algumas poucas versões de Algoritmos Genéticos, que são os algoritmos evolucionários mais conhecidos e melhor compreendidos nos dias de hoje. Todos os operadores propostos foram testados apenas nesses AG. Em todas as situações, a utilização de aproximações quadráticas acoplada ao algoritmo genético se mostrou eficaz, permitindo uma melhora na taxa de convergência do algoritmo e uma maior precisão na solução obtida.

Este trabalho apresenta algoritmos aptos a tratar problemas com as seguintes características:

- variáveis contínuas no espaço  $\mathbb{R}^n$ ;
- função-objetivo não-linear e (possivelmente) multimodal;
- restrições de igualdade e desigualdade, sendo as funções de restrição não-lineares e, possivelmente multimodais.

Esse trabalho está assim dividido:

- o capítulo 2 faz uma revisão dos métodos de otimização clássicos que utilizam aproximações quadráticas e descreve as metodologias empregadas para a construção dessas aproximações;
- o capítulo 3 apresenta o algoritmo-base aqui utilizado— o algoritmo genético— descrevendo brevemente seus operadores;
- o capítulo 4 trata das modificações propostas que serão aplicadas em problemas mono-objetivo. Esse capítulo apresenta a metodologia de mudança de coordenadas em problemas irrestritos e dois operadores de busca local especializados em problemas com restrições de igualdade e de desigualdade.
- as modificações propostas para problemas multiobjetivo são apresentadas no capítulo 5. A utilização das aproximações quadráticas dentro de algoritmos genéticos multi-objetivo em problemas irrestritos e com restrições não-lineares de desigualdade são apresentadas nesse capítulo. Uma nova métrica para avaliar a performance de algoritmos multiobjetivo também é apresentada.
- o capítulo 6 apresenta as conclusões gerais deste trabalho.



# Capítulo 2

## Aproximações Quadráticas

A utilização de técnicas de otimização vem ganhando espaço como ferramenta de projeto nas mais diversas áreas. Entretanto, para uma grande gama de problemas, a função-objetivo só pode ser avaliada de forma numérica, como por exemplo, quando se utiliza o método de elementos finitos, o que usualmente significa um custo computacional elevado. Nesse contexto, a utilização de técnicas de aproximações de funções tem desempenhado um papel relevante pois, essas permitem que o projetista minimize o esforço computacional necessário para encontrar uma solução satisfatória para o problema em questão. Em outras situações, as aproximações não substituem as funções originais, mas apenas são utilizadas para obter boas estimativas para a solução do problema, melhorando dessa forma as propriedades de convergência do algoritmo de otimização escolhido. Em ambos os casos, a seleção da função de aproximação apropriada para modelar a função original desempenha um papel importante no desempenho do algoritmo.

Como um exemplo de técnicas de aproximações, podemos citar o *método de assíntotas móveis* ([17]), uma técnica que utiliza aproximações em série de Taylor, supondo que as variáveis seguem modelos fixos. Esse método considera que a função-objetivo possa ser linearizada em um espaço intermediário conhecido. A técnica de interpolação multiquádrica é um método que pode ser usado para interpolar uma função com várias variáveis, dado um número de amostras da função em questão. Podemos ainda citar os métodos de aproximações baseados na representação da função como uma série de Taylor truncada nos termos de ordem mais baixa, usualmente de ordem um ou dois (aproximações lineares e quadráticas respectivamente). Os métodos de superfície de resposta <sup>1</sup> ([7, 8]), de funções de base radiais <sup>2</sup> ([11, 12]) e as redes neurais ([13]) também constituem exemplos de técnicas de aproximações amplamente empregadas.

---

<sup>1</sup>Tradução para response surface.

<sup>2</sup>Tradução para radial basis function.

As técnicas de aproximações quadráticas constituem o ponto central deste trabalho. Esse tipo de aproximação é a técnica mais simples que permite obter informações a respeito da curvatura da função, uma informação relevante nos algoritmos de otimização. A aproximação quadrática será válida em situações em que as derivadas de ordem superior a dois sejam desprezíveis ou quando a distância entre o ponto conhecido e o ponto aproximado for pequena. Com essas aproximações quadráticas em mãos, é possível propor modificações em alguns algoritmos de otimizações que possibilitem uma melhora na garantia e na velocidade de convergência dos mesmos.

Aproximações quadráticas tem sido amplamente utilizadas em algoritmos de otimização. Basicamente, existem dois princípios que guiam o uso das aproximações quadráticas em cada processo de otimização:

1. as aproximações quadráticas são utilizadas para gerar uma estimativa para o mínimo da função. Essa estimativa do mínimo será, então, a nova estimativa da solução do problema. A estimativa do mínimo é atualizada a cada passo do algoritmo, gerando-se assim uma nova aproximação quadrática ao redor da antiga estimativa disponível.
2. as aproximações quadráticas podem ser utilizadas para efetuar uma mudança de coordenada no espaço das variáveis. Esta mudança de coordenada é empregada como um passo intermediário a cada iteração do algoritmo de otimização. Esse procedimento tem como objetivo definir um novo conjunto de pontos, facilitando assim a busca pelo ponto ótimo do problema.

Neste trabalho, a aproximação quadrática é utilizada tanto para gerar estimativas para o mínimo do problema quanto para definir uma mudança de coordenadas apropriada para agilizar o processo de otimização.

Podemos encontrar várias referências na literatura em que se faz o uso de aproximações quadráticas:

- método de Newton ([14]);
- métodos quasi-Newton ([14]);
- programação quadrática sequencial ([14]) e
- métodos de região de confiança <sup>3</sup> ([18]).

Os métodos de Newton, programação quadrática sequencial e de região de confiança se apoiam na idéia (1): utilizam as aproximações para a obtenção uma nova estimativa para

---

<sup>3</sup>Tradução para Trust Region Methods.

o mínimo da função, enquanto os métodos quasi-Newton têm suas diretrizes na idéia (2). Neste trabalho, apresentaremos novas formas de se utilizar aproximações quadráticas em algoritmos evolucionários de otimização que se baseiam nas idéias (1) e (2).

Neste capítulo vamos descrever os principais métodos conhecidos que utilizam as aproximações quadráticas para tratar problemas de otimização. Inicialmente, suponha que a função  $f(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}$ , com variáveis reais  $x \in \mathbb{R}^n$ , seja de classe  $\mathbb{C}^\infty$ . Toda função de classe  $\mathbb{C}^\infty$  pode ser aproximada por uma função quadrática, dentro de uma região que contenha o ponto de mínimo  $x^*$ . Essa aproximação quadrática, que também é feita ao redor de um ponto  $x_o$  contido nessa região, é obtida truncando-se a série de Taylor associada a essa função e é dada por:

$$\tilde{f}(x) \approx f(x_o) + \nabla f(x_o)(x - x_o) + \frac{1}{2}(x - x_o)^T H(x_o)(x - x_o) \quad (2.1)$$

sendo  $\nabla f(x_o)$  o vetor gradiente da função  $f$  no ponto  $x_o$ , e  $H(x_o)$  a matriz Hessiana da função em  $x_o$ . Para se obter uma aproximação quadrática (2.1) para a função  $f$ , os métodos tradicionais calculam o gradiente e a Hessiana por meio de diversas avaliações da função  $f$  já que, na maioria dos casos, não se conhece a expressão analítica de  $f$ .

Os métodos de Newton e quasi-Newton são representantes dos métodos conhecidos como *Métodos de Direções de Busca*<sup>4</sup> e foram desenvolvidos a partir da idéia básica de fazer o algoritmo evoluir encontrando novos pontos que se situam em direções nas quais a função considerada decresça, se comparado ao ponto atual. A diferença básica entre os representantes dessa classe de métodos consiste na forma como essas direções de movimento são escolhidas. Uma vez feita essa escolha, o algoritmo irá se movimentar para o ponto de mínimo que se encontra sobre a linha definida pela referida direção ou para uma estimativa desse ponto.

O método de programação quadrática sequencial é baseado diretamente na resolução das condições de Lagrange de primeira ordem. Nesse caso, as condições de Lagrange são lineares; então, resolvê-las significa resolver um sistema em um espaço de dimensão  $(n + m)$ .

## 2.1 Métodos que Utilizam Aproximações Quadráticas

Apresentaremos nessa seção alguns métodos conhecidos que fazem uso de aproximações quadráticas.

---

<sup>4</sup>Tradução para Line Search Method.

### 2.1.1 Método de Newton

Seja  $f$  uma função real de uma variável e seja  $\tilde{f}$  uma aproximação quadrática para  $f$  obtida através da equação (2.1):

$$\tilde{f}(x) \approx f(x_o) + \nabla f(x_o)(x - x_o) + \frac{1}{2}(x - x_o)^T H(x_o)(x - x_o). \quad (2.2)$$

Encontrando-se o ponto no qual o gradiente de  $\tilde{f}$  se anula,

$$0 = \nabla \tilde{f}(x) = \nabla f(x_o) + H(x_o)(x - x_o), \quad (2.3)$$

obtemos uma estimativa para o ponto de mínimo da função ([14]):

$$x^* \approx x_o - (H(x_o)^{-1})\nabla f(x_o) \quad (2.4)$$

Se a função  $f$  for quadrática e se conhecermos o gradiente e a Hessiana em qualquer ponto  $x_o$ , através de (2.4) podemos obter o ponto de mínimo  $x^*$  em um único passo. Se a função  $f$  não for quadrática mas aproximável por uma função quadrática numa certa região, através de (2.4) podemos obter estimativas do ponto de mínimo. Se o ponto inicial estiver suficientemente perto da solução original, o método de Newton apresenta convergência garantida. Entretanto para se garantir a convergência nos casos onde o ponto inicial estiver longe da solução, devemos introduzir um parâmetro de busca em (2.4)

$$x_{k+1} = x_k - \alpha_k (H(x_k)^{-1})\nabla f(x_k) \quad (2.5)$$

que será escolhido de modo que a nova estimativa  $x_{k+1}$  garantidamente produza uma diminuição no valor da função-objetivo.

O método de Newton constitui o primeiro e mais direto representante da idéia (1) descrita anteriormente, onde a aproximação quadrática é utilizada para produzir uma estimativa para o mínimo do problema. Para maiores informações sobre provas de convergência e outras características do método de Newton, veja [14].

### 2.1.2 Métodos quasi-Newton

A idéia básica dos métodos quasi-Newton é usar uma estimativa recursiva para a inversa da Hessiana no lugar da inversa utilizada no método de Newton, uma vez que o cálculo da inversa descrito na seção anterior requer muitas avaliações da função. Essa estimativa será obtida usando-se informações disponíveis ao longo do processo de otimização e a estimativa



parcial da inversa da Hessiana será utilizada durante o processo. Isso é particularmente útil na otimização de funções não-quadráticas uma vez que, nesse caso, a Hessiana não é constante.

Os métodos quasi-Newton são os primeiros representantes da idéia (2) descrita no início do capítulo, nos quais uma estimativa da inversa da Hessiana é construída iterativamente, para se definir uma direção de busca  $-H(x_k)^{-1}\nabla f(x_k)$  na qual uma busca unidimensional é realizada, produzindo uma nova estimativa  $x_{k+1}$ . Esse procedimento pode ser interpretado como uma correção do espaço de coordenadas, sendo  $H(x_k)^{-1}$  uma matriz de mudança de coordenadas. Essa mudança de coordenadas faz com que as superfícies de nível da função quadrática aproximada fiquem circulares.

Os métodos de otimização conhecidos como quasi-Newton são desenvolvidos usando essa abordagem recursiva, que permite a construção gradativa de uma matriz que corresponde a uma estimativa para a inversa da Hessiana,  $H \approx H^{-1}$ . Esses métodos garantem que a matriz  $H$  seja simétrica, definida positiva e bem condicionada. Após a obtenção dessa estimativa da inversa da Hessiana, o ponto de mínimo da função pode ser encontrado através de diversos métodos de otimização.

Dois métodos, *DFP* (Davidon-Fletcher-Powell) e *BFGS* (Broyden-Fletcher-Goldfarb-Shanno), foram criados para produzir estimativas recursivas para a matriz  $H$  preservando as propriedades citadas acima. As estimativas da inversa da Hessiana, que serão utilizadas nos cálculos das estimativas para o mínimo no novo método, podem ser então calculadas através das atualizações do método *BFGS* ([14]):

$$H_{k+1} = H_k + c_k \quad (2.6)$$

com

$$c_k = \left(1 + \frac{r_k^T H_k r_k}{r_k^T v_k}\right) \frac{v_k v_k^T}{v_k^T r_k} - \frac{v_k r_k^T H_k + H_k r_k v_k^T}{r_k^T v_k} \quad (2.7)$$

e  $r_k = \nabla f_k - \nabla f_{k+1}$  e  $v_k = x_k - x_{k+1}$ . A estimativa da inversa da Hessiana fornece então uma direção na qual ocorre a busca pelo ponto de mínimo da função.

Para funções não-quadráticas, esses métodos recursivos oferecem uma combinação de vantagens: necessitam apenas das informações dos gradientes e a matriz  $H$  é definida positiva durante todo o processo. Além disso, para problemas quadráticos, a matriz  $H$  converge para a inversa da Hessiana em  $n$  passos, sendo então possível dizer que, nesses casos, a matriz  $H$  converge para a inversa da Hessiana no final do processo de otimização.

### 2.1.3 Programação Quadrática Sequencial

O método de programação quadrática aparece em várias aplicações e forma uma base para os algoritmos de programação não-linear em geral. Assim como o método de Newton, o método de programação quadrática sequencial se baseia na idéia (1) descrita no início deste capítulo. Esse método resolve um problema de otimização não-linear genérico através da resolução de uma sequência de sub-problemas quadráticos. A idéia de resolver um problema não-linear através de uma sequência de sub-problemas não é nova: os métodos de barreira e de penalidade (veja [19]) resolvem uma sequência de sub-problemas irrestritos.

A aproximação quadrática é utilizada para definir cada sub-problema quadrático, expresso como:

$$\begin{aligned} \min \frac{1}{2}x^T Qx + x^T c \\ \text{sujeito a: } \begin{cases} A_i^T x = b_i, & i \in E \\ A_i^T x \leq b_i, & i \in I \end{cases} \end{aligned} \quad (2.8)$$

sendo  $E$  e  $I$  conjunto de índices para as restrições de igualdade e desigualdade respectivamente. A matriz  $Q$  é simétrica e semidefinida positiva.

Um programa quadrático pode ser simplificado e resolvido na forma fechada se possui apenas restrições de igualdade. Considerando o seguinte programa quadrático:

$$\begin{aligned} \min \frac{1}{2}x^T Qx + x^T c \\ \text{sujeito a: } A^T x = b \end{aligned} \quad (2.9)$$

a solução de forma fechada, sob certas condições (matriz  $Q$  sendo definida positiva e matriz  $A$  tendo posto completo), para esse programa é dada por:

$$\begin{aligned} x^* &= Q^{-1}A^T(AQ^{-1}A^T)^{-1}[AQ^{-1}c + b] - Q^{-1}c \\ &= -Q^{-1}[I - A^T(AQ^{-1}A^T)^{-1}AQ^{-1}]c \\ &\quad + Q^{-1}A^T(AQ^{-1}A^T)^{-1}b \end{aligned} \quad (2.10)$$

O programa quadrático generalizado (2.8) com restrições de desigualdade é normalmente resolvido através do método do conjunto ativo ([14]), que divide as restrições em dois grupos: as restrições que serão tratadas como ativas e as que serão tratadas como inativas. As restrições pertencentes ao último grupo são essencialmente ignoradas. A idéia básica dos métodos do conjunto ativo é definir, a cada passo do algoritmo, um conjunto de restrições, chamado de conjunto de trabalho, que será tratado como o conjunto ativo. O conjunto de trabalho é um subconjunto das restrições que estão ativas no ponto atual. Nesse caso, o conjunto de trabalho sempre será composto pelas restrições de igualdade e, eventualmente, por algumas restrições de desigualdade.

### 2.1.4 Métodos de Região de Confiança

Quando se trata de minimização de funções, são chamados de métodos locais aqueles métodos que convergem se a aproximação inicial estiver suficientemente próxima de um ponto estacionário da função  $f$ . O método de Newton é exemplo de um método tipicamente local. Frequentemente os métodos locais são de fato muito eficientes, seja porque bons pontos iniciais são conhecidos, seja porque as regiões de convergência são sempre muito maiores do que previsto pela teoria local. Porém, existem muitos problemas difíceis, em que os métodos puramente locais falham.

Para contornar essa limitação dos métodos locais são utilizadas estratégias para induzir o método a convergir independentemente da aproximação inicial. Os procedimentos mais utilizados para induzir a convergência são as técnicas de busca unidimensional e as estratégias baseadas em regiões de confiança. Nesses dois procedimentos o método é induzido a convergir pela escolha da direção e do tamanho do passo que produzam um efetivo decréscimo da função-objetivo  $f$  em cada iteração.

Quando se utiliza a estratégia de busca unidimensional, determina-se inicialmente uma direção de descida  $d$  a partir de  $x_k$ , ( $d$  tal que  $f(x_k).d < 0$ ) e depois calcula-se, de forma exata ou aproximada, o tamanho do passo  $\lambda_k > 0$  tal que  $f(x_k + \lambda_k.d) < f(x_k)$ .

Se as aproximações das Hessianas usadas não forem necessariamente definidas positivas, a técnica de busca unidimensional não é adequada devido à dificuldade em se determinar direções de descida. No caso em que a matriz Hessiana ou uma aproximação dela  $H_k$  for definida positiva, tem-se  $v^T.H_k.v > 0$ , para todo vetor  $v \in \mathbb{R}^n, v \neq 0$ . Então, uma direção de descida  $d$  é dada pela solução do sistema linear  $H_k.d = -f(x_k)$ .

Os métodos de região de confiança constroem, ao redor da estimativa atual, um modelo para a função-objetivo. Em geral, esse modelo é mais barato de calcular e torna o processo de minimização mais fácil. Modelos quadráticos convexos são usualmente empregados. Supõe-se que esse modelo represente bem a função-objetivo em uma chamada região de confiança. Normalmente, essa região é representada por uma bola centrada ao redor da estimativa atual,  $x_k$ , da seguinte forma:

$$\mathcal{B}_k = \{x \in \mathbb{R}^n \mid \|x - x_k\| \leq \delta_k\} \quad (2.11)$$

sendo  $\delta_k$  o raio da região de confiança. Esse valor indica o quão longe esse modelo está da representação real da função-objetivo.

Um novo ponto, o qual minimiza ou reduz suficientemente o valor do modelo dentro dessa região de confiança estabelecida, é calculado e então o valor de função-objetivo original é calculado para esse ponto. Se a redução no valor da função-objetivo ( $f(x_{k+1}) < f(x_k)$ )

for equivalente à redução observada pela modelagem ( $\tilde{f}(x_{k+1}) < \tilde{f}(x_k)$ ), esse novo ponto é aceito como uma nova estimativa e a região de confiança será centrada ao redor desse ponto (e possivelmente estendida). Por outro lado, se a redução obtida pela função-objetivo for insuficiente se comparada com a redução obtida pelo modelo, a estimativa atual não será alterada e a região de confiança será reduzida.

Temos a seguir a formulação irrestrita do método de região de confiança:

$$\begin{aligned} \min \bar{f}(x) \\ \text{sujeito a : } \|x_0 - x_k\| \leq \epsilon \end{aligned} \quad (2.12)$$

sendo  $\bar{f}(x)$  a aproximação quadrática para a função objetivo original do problema.

O modelo quadrático que aproxima a função-objetivo irá interpolar os valores da funções nos pontos,

$$\bar{f}(y) = f(y), \quad y \in Y. \quad (2.13)$$

Note que a cardinalidade de  $Y$  deve ser igual a

$$\frac{(n+1)(n+2)}{2}$$

sendo  $n$  a dimensão do espaço de variáveis. Porém, essa condição não é suficiente para garantir a existência de um interpolante. Algumas condições geométricas para  $Y$  devem ser acrescentadas para que se possa garantir a existência e a unicidade do interpolante quadrático desejado. Essas condições geométricas se referem à relação existente entre os pontos de interpolação e o espaço de aproximação. Para maiores informações sobre essas condições geométricas, veja [18].

Alguns procedimentos de otimização que utilizam as idéias dos métodos de região de confiança fazem o cálculo da interpolação quadrática sem utilizar nenhuma informação sobre a derivada da função. Esses procedimentos são conhecidos como Região de Confiança Livre de Derivadas <sup>5</sup>.

Assim como o método de Newton e o método de programação quadrática sequencial, os métodos de região de confiança se baseiam na idéia (1) descrita no início deste capítulo.

### 2.1.5 Método Elipsoidal

O método elipsoidal, desenvolvido em 1970 por Shor, se enquadra na categoria de métodos de exclusão de semi-espacos como sendo o primeiro método a partir do qual se demonstrava a convergência em tempo polinomial para problemas lineares ([20, 21]). Comparados com os métodos de planos de corte, o método elipsoidal não cresce em complexidade

---

<sup>5</sup>Tradução para Derivative Free Trust Region.

a cada iteração mas apresenta taxas lentas de convergência em problemas lineares. Devido a este fato, o método simplex para problemas lineares nunca foi substituído pelo algoritmo elipsoidal. Posteriormente, o método elipsoidal foi suplantado, em problemas lineares, pelos métodos dos pontos interiores.

Usualmente, o algoritmo não faz uso de aproximações quadráticas em sua formulação original. No apêndice A apresentamos uma metodologia, baseada em aproximações quadráticas, que pode ser incluída no algoritmo elipsoidal. Essa metodologia leva a uma melhora na taxa de convergência do algoritmo (observada numericamente) quando aplicada a problemas ir-restritos e com restrições lineares de igualdade.

## 2.2 Metodologias Baseadas em Gradiente para Criar as Aproximações Quadráticas

Apresentaremos uma metodologia para construir uma aproximação quadrática para uma função-objetivo genérica. Dois procedimentos serão apresentados a seguir: um utilizando informações do gradiente da função e outro que não utiliza nenhuma informação a respeito dos gradientes.

Dentre os procedimentos que utilizam informações do gradiente, os seguintes métodos serão descritos:

- Método Direto e
- Método Recursivo.

Esses métodos determinam a matriz Hessiana através do cálculo dos gradientes da função. No primeiro caso, calculando-se  $n + 1$  vetores gradiente, sendo  $n$  a dimensão do problema, a Hessiana pode ser diretamente construída. Um problema decorrente dessa metodologia é que não é possível garantir que a Hessiana seja simétrica e positiva definida. O custo computacional deste método é elevado devido ao cálculo dos vetores gradiente.

A escolha da metodologia para a construção da aproximação quadrática deve levar em consideração o algoritmo de otimização que servirá de base ao processo e o custo computacional que cada metodologia acarretará ao processo.

### 2.2.1 Método Direto

A seguir, apresentamos um método direto para estimar a Hessiana da própria função, e portanto sua inversa, utilizando a informação dos gradientes obtidos em vários pontos.

Considere a seguinte equação

$$\nabla f(x) = \nabla f(x_0) + F(x_0).(x - x_0) \quad (2.14)$$

que representa uma aproximação de primeira ordem para o gradiente.

Reescrevendo essa equação para dois pontos distintos  $x_1$  e  $x_2$  e supondo que a Hessiana seja constante em todo o espaço, temos:

$$F(x_1 - x_2) = \nabla f(x_1) - \nabla f(x_2) \quad (2.15)$$

Esse processo pode ser repetido:

$$\begin{aligned} F(x_1 - x_2) &= \nabla f(x_1) - \nabla f(x_2) \\ F(x_2 - x_3) &= \nabla f(x_2) - \nabla f(x_3) \\ &\vdots \\ F(x_{i-1} - x_i) &= \nabla f(x_{i-1}) - \nabla f(x_i) \\ &\vdots \\ F(x_n - x_{n+1}) &= \nabla f(x_n) - \nabla f(x_{n+1}) \end{aligned} \quad (2.16)$$

Definindo os vetores  $v_i$  e  $r_i$  como

$$\begin{aligned} v_i &= x_i - x_{i+1} \\ r_i &= \nabla f(x_i) - \nabla f(x_{i+1}) \end{aligned} \quad (2.17)$$

tem-se que:

$$F[v_1 v_2 \dots v_n] = [r_1 r_2 \dots r_n]. \quad (2.18)$$

Definindo-se  $V = [v_1 \ v_2 \ \dots \ v_n]$  e  $R = [r_1 \ r_2 \ \dots \ r_n]$  obtemos

$$FV = R. \quad (2.19)$$

É possível escolher os vetores  $v_i$  de modo que eles sejam linearmente independentes, o que implica que  $V$  será invertível. Assim

$$F = RV^{-1}. \quad (2.20)$$

Acrescentando-se a hipótese de que  $H$  seja não-singular teremos

$$F^{-1} = VR^{-1}. \quad (2.21)$$

Isso significa que, avaliando-se o gradiente da função  $f$  em apenas  $n + 1$  pontos adequadamente escolhidos, teremos  $H = F^{-1}$ , determinando dessa forma a inversa da Hessiana da função.

Um problema comum desse método de construção da inversa da Hessiana da função-objetivo é a possibilidade da matriz  $H$  construída não ser simétrica ou não ser positiva definida, acarretando dificuldades na convergência do algoritmo.

### 2.2.2 Métodos Recursivos

É possível obter uma aproximação recursiva da estimativa da Hessiana (ou da sua inversa) durante o decorrer do processo de otimização. Essa estimativa parcial é obtida através de informações coletadas durante o processo de otimização. A estimativa atual será então utilizada no próprio processo de otimização.

Essa metodologia é particularmente útil na otimização de funções não-quadráticas, em que a Hessiana não é constante. Dessa forma, esse procedimento permite uma adaptação contínua da estimativa da Hessiana ao seu valor localmente válido. Os métodos:

- correção de Posto 1;
- BFGS;
- DFP

são exemplos desses procedimentos recursivos. O procedimento BFGS foi descrito na seção 2.1.2, por ser um representante da família de métodos conhecidos como quasi-Newton. Detalhes sobre os outros procedimentos podem ser encontrados em [14].

## 2.3 Método Livre de Derivada para Criar as Aproximações Quadráticas

Nos dois métodos de obtenção da Hessiana descritos anteriormente, é necessário o cálculo das derivadas. Entretanto, em alguns casos o cálculo das derivadas é inconveniente, difícil ou até mesmo impraticável. Além disso, quando possível, o cálculo do gradiente de uma função acarreta um alto custo computacional.

Tendo em vista essa dificuldade, apresentamos um outro método, o Método Livre de Derivadas, para obter as aproximações quadráticas. Esse Método Livre de Derivadas, assim como sugere o nome, não faz uso de nenhum cálculo de derivada de função para obter uma aproximação para a Hessiana. A Hessiana, nesse caso, é construída levando-se em conta

as amostras dos pontos e suas respectivas avaliações na função-objetivo. Essas informações estão disponíveis durante o processo de otimização, não sendo portanto necessária nenhuma avaliação adicional da função-objetivo. Além disso, como o método utiliza uma amostra da população e suas respectivas avaliações de função, essa metodologia torna-se aplicável aos algoritmos evolucionários (que também trabalha com uma amostra de pontos) quando queremos obter uma aproximação quadrática semi-local para a função original.

Nessa metodologia poderíamos usar o método de mínimos quadrados para construir a aproximação quadrática porém não teríamos a garantia que a Hessiana será positiva definida. Para forçar que a matriz seja positiva definida utilizamos, por exemplo, um método de programação semi-definida. A metodologia proposta será detalhada a seguir.

### 2.3.1 Metodologia

Seja  $f$  uma função real. Dados o seguinte conjunto de pontos distintos,  $z_1, z_2, \dots, z_n$ , podemos considerar o problema de encontrar a função quadrática

$$h(z) = z^T H z + r^T z + \gamma \quad (2.22)$$

para alguma matriz simétrica  $H$ , um vetor  $r$  e um escalar  $\gamma$ , todos com dimensões compatíveis, tais que

$$f(z_i) \approx h(z_i) \quad (2.23)$$

para  $i = 1, 2, \dots, N$ , sendo  $N$  o número de pontos disponíveis.

Portanto o problema de encontrar  $f$  que satisfaça (2.23) se resume a encontrar  $H$ ,  $r$  e  $\gamma$  tais que o erro de aproximação seja dado por

$$E_i = z_i^T H z_i + r^T z_i + \gamma - h(z_i) \quad (2.24)$$

para  $i = 1, 2, \dots, N$ .

Este é um sistema linear com  $N$  equações nas variáveis de  $H$ ,  $r$  e  $\gamma$ . O número de variáveis em  $H$  é igual a  $n + \frac{n^2-n}{2}$ , e portanto o número total de variáveis é dado por

$$n + \frac{n^2 - n}{2} + n + 1 = \frac{(n+1)(n+2)}{2} \quad (2.25)$$

Se

$$z_i^T H z_i + r^T z_i + \gamma = 0 \Rightarrow H = 0, r = 0, \gamma = 0 \quad (2.26)$$

para  $i = 1, 2, \dots, N$ , com  $N = \frac{(n+1)(n+2)}{2}$ , então existe um única função quadrática  $h$  que



verifica (2.23). Neste caso temos uma *interpolação*. Quando  $N > \frac{(n+1)(n+2)}{2}$ , o sistema linear (2.24) é sobre-determinado e podemos encontrar uma solução de norma mínima:

$$\min_{H,r,\gamma} \|E\|. \quad (2.27)$$

Se a norma corresponder à norma euclidiana, então a função  $h$  é a aproximação quadrática obtida através dos quadrados mínimos. A função  $h$  poderia ser determinada através do método de quadrados mínimos, porém não é possível garantir a convexidade da função.

É importante salientar que essa metodologia só poderá ser aplicada se o número de pontos  $N$  for maior ou igual a

$$\frac{(n+1)(n+2)}{2}$$

sendo  $n$  a dimensão do problema.

Para garantir que a matriz  $H$  seja positiva definida, obtendo assim a convexidade da função, a função  $h$  pode ser obtida resolvendo-se

$$\min \left( t : \sqrt{\sum_i E_i^2} \leq t, H > 0 \right) \quad (2.28)$$

Uma vez que a restrição em (2.28) é uma restrição do tipo cone de Lorentz, o problema semi-definido (2.28) pode ser eficientemente resolvido usando, por exemplo, o programa citado no apêndice B, o SeDuMi [22].

Com a aproximação quadrática obtida

$$h(z) = z^T H z + r^T z + \gamma \quad (2.29)$$

o ponto de mínimo de  $h$  pode ser calculado por

$$z_h = -\frac{1}{2} H^{-1} r \quad (2.30)$$

e a expressão analítica de  $h$  pode ser re-escrita

$$h(z) = (z - z_h)^T H (z - z_h) - c \quad (2.31)$$

sendo

$$c = 0.25 r^T H^{-1} r + \gamma.$$

A metodologia livre de derivadas será utilizada neste trabalho para criar operadores

locais especializados que serão utilizados em algoritmos evolucionários de otimização. Esses operadores irão melhorar a convergência e o desempenho dos algoritmos evolucionários mono e multiobjetivos.

### 2.3.2 Custo Computacional

Uma vez que o método descrito na seção anterior não faz uso de nenhum cálculo de derivada de função e nem necessita de nenhuma avaliação extra de função-objetivo (além daquelas já calculadas), essa metodologia é particularmente indicada em problemas cuja avaliação de função é cara do ponto de vista computacional. Nesses casos, o custo computacional decorrente da construção da aproximação quadrática pode ser justificável.

Com o objetivo de determinar o custo computacional envolvido na construção da aproximação quadrática, realizamos dois experimentos distintos. Nos dois experimentos, utilizamos um computador portátil Toshiba Satellite com processador Pentium 4 com 2.66 GHz e com 240 MB de memória RAM.

No primeiro experimento, utilizando funções com dimensões variando de 2 a 10, e usando uma amostra de 100 pontos e suas respectivas avaliações na função original, medimos o tempo computacional gasto no processo de construção da aproximação quadrática em cada caso. Trinta testes foram realizados com cada caso e a média do tempo computacional foi então calculada. A tabela 2.1 mostra o tempo computacional médio, em segundos, para cada dimensão.

Tabela 2.1: Tempo computacional médio, em segundos, em cada problema com dimensões variadas.

Dimensão	Tempo (em segundos)
2	0.5037
3	0.5224
4	0.5474
5	0.5589
6	0.6052
7	0.6552
8	0.6995
9	0.7703
10	0.8297

Em um segundo experimento, com o propósito de verificar a dependência do custo computacional com o tamanho da amostra, realizamos 30 testes em cada dimensão variando-se o tamanho da amostra e depois calculamos a média. Executamos 10 experimentos com o tamanho da amostra variando desde o número mínimo de pontos exigido até dez vezes esse

valor. A tabela 2.2 mostra o número mínimo de pontos ( $n_m$ ) necessário para a construção da aproximação quadrática em cada dimensão. A tabela 2.3 mostra o tempo computacional médio (em segundos), para cada dimensão, variando-se o tamanho da amostra de  $n_m$  até  $10n_m$ .

Tabela 2.2: Número mínimo de pontos ( $n_m$ ) para a construção da aproximação quadrática em cada problema com dimensões variadas.

Dimensão	$n_m$
2	6
3	10
4	15
5	21
6	28
7	36
8	45
9	55
10	66

Tabela 2.3: Tempo computacional médio (em segundos) para cada dimensão, variando-se o tamanho da amostra de  $n_m$  até  $10n_m$  para a construção da aproximação quadrática.

Dim.	$n_m$	$2n_m$	$3n_m$	$4n_m$	$5n_m$	$6n_m$	$7n_m$	$8n_m$	$9n_m$	$10n_m$
2	0.313	0.343	0.344	0.328	0.406	0.360	0.375	0.437	0.391	0.453
3	0.328	0.312	0.375	0.375	0.407	0.406	0.484	0.485	0.593	0.547
4	0.313	0.343	0.375	0.454	0.484	0.531	0.563	0.672	0.671	0.750
5	0.406	0.407	0.453	0.547	0.656	0.703	0.828	0.922	1.016	1.140
6	0.375	0.469	0.578	0.688	0.859	1.031	1.188	1.359	1.594	1.797
7	0.406	0.578	0.750	0.938	1.156	1.484	1.813	2.172	2.546	3.031
8	0.469	0.688	0.984	1.322	1.719	2.312	2.891	3.516	4.234	5.032
9	0.531	0.891	1.328	1.937	2.688	3.610	4.640	5.735	7.978	8.359
10	0.609	1.172	2.015	2.906	4.078	5.563	7.171	9.913	11.640	14.109

Podemos observar que, em dimensões menores, o aumento do tamanho da amostra não acarreta em um aumento expressivo do tempo computacional. Em dimensões mais elevadas, o aumento do tamanho da amostra leva a um aumento mais significativo no tempo computacional. Entretanto, esse custo computacional torna-se justificável em problemas cuja avaliação da função objetivo, através de métodos numéricos, chega a margem dos minutos.



# Capítulo 3

## Algoritmo Genético

A partir de 1970, a computação evolucionária sofreu grande avanço devido aos trabalhos de Holland, De Jong, Schwefel e Fogel [23]. Logo após, vários trabalhos, como o de Goldberg [24], consolidaram essa nova metodologia que se encontra, nos dias atuais, bastante desenvolvida.

Os algoritmos evolucionários são métodos flexíveis, robustos e capazes de realizarem uma busca global. Os algoritmos evolucionários são métodos classificados como estocásticos, que possuem como característica principal a busca pelo ótimo usando regras de transição probabilísticas. Tais métodos normalmente utilizam apenas as informações contidas nas funções envolvidas, não necessitando de derivadas ou hipóteses sobre a continuidade das funções. Além disso, são técnicas que imitam, de forma simplificada, os fenômenos ou processos de evolução baseados na teoria de Darwin. Os algoritmos evolucionários trabalham com uma população de soluções, executando sobre essa população uma série de operações heurísticas, de forma a fazê-la convergir para a solução do problema.

Os principais representantes dos algoritmos evolucionários são:

- Estratégias Evolutivas (EEs),
- Algoritmos Genéticos (AGs),
- Programação Evolucionária (PE) e
- Programação Genética (PG)

os quais foram desenvolvidos independentemente.

Esses algoritmos modelam o processo de evolução coletiva em uma população de indivíduos, em que cada um desses indivíduos representa não apenas um ponto no espaço de busca mas também uma fonte de informações sobre o ambiente em questão [23]. A população inicial evolui na direção de melhores regiões por meios de processos de *recombinação*,

*mutação e seleção.* O ambiente define uma informação qualitativa para os novos indivíduos, a *função de aptidão*, e o processo de seleção favorece a reprodução daqueles indivíduos mais aptos. A recombinação permite uma troca de informações entre pais e a mutação introduz inovações na população.

A figura 3.1 mostra um esquema básico dos representantes da classe dos algoritmos de Computação Evolutiva.

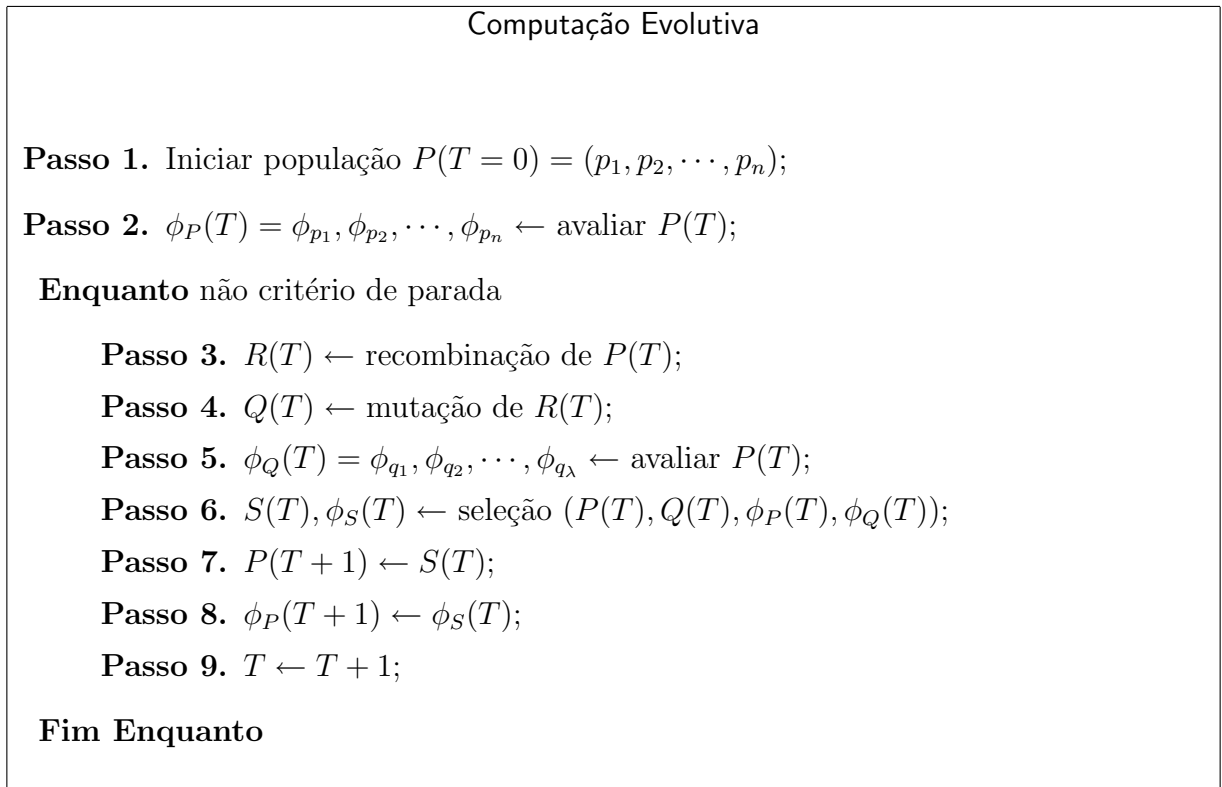


Figura 3.1: Esquema básico dos algoritmos da Computação Evolutiva.

Apesar de todos os representantes dessa classe compartilharem uma mesma estrutura básica, cada um enfatiza diferentes operadores. A diferença básica entre os representantes se deve ao papel que cada operador genético representa. No caso das EEs, a mutação é o operador mais importante, a seleção é determinística (os melhores indivíduos serão deterministicamente selecionados) e a recombinação é assexuada. Os AGs usam métodos probabilísticos de seleção, reprodução sexuada e mutação. Na PE, apenas a mutação é utilizada. Os métodos de PG representam algoritmos genéticos adaptados para resolver problemas de estrutura. Apesar de utilizar os operadores dos AGs, os indivíduos não são representados por variáveis e sim por árvores que codificam estruturas previamente modeladas. Para maiores informações sobre os operadores descritos acima, ver [23].

Aos Algoritmos Genéticos será dada uma ênfase especial, já que esses métodos foram

escolhidos para a utilização das aproximações quadráticas neste trabalho. O AG é caracterizado pela existência de três operadores genéticos:

*cruzamento*: este operador combina a informação contida em dois ou mais indivíduos para gerar outros indivíduos;

*mutação*: este operador utiliza a informação contida em um indivíduo para, estocasticamente, gerar outro indivíduo. Essa operação aumenta a capacidade exploratória dos AGs;

*seleção*: este operador é utilizado para gerar a nova população através de réplicas de alguns indivíduos e eliminação de outros.

Um AG pode ser construído a partir desses três operadores ou pode conter operadores adicionais: elitismo, nicho, busca local, etc.

Os AGs podem utilizar codificação binária ou real para a representação das variáveis. Apesar de tradicionalmente o AG ser utilizado com codificação binária, no caso de problemas com variáveis reais, a codificação real se faz mais simplificada e imediata. Neste trabalho, apenas o AG com codificação real foi utilizado.

A figura 3.2 mostra a estrutura fundamental de um AG com codificação real incluindo-se um operador de elitismo. O operador de elitismo seleciona deterministicamente o melhor indivíduo. Convém salientar que, tradicionalmente, o tamanho das populações se mantém o mesmo ao longo das gerações. Observe que esse algoritmo pode ser modificado alterando-se o tipo de cada operador ou acrescentando outros operadores mais sofisticados.

## 3.1 Operadores

No AG utilizado nesse trabalho, para a minimização da função-objetivo, foram definidas as seguintes operações:

### 3.1.1 Função de Aptidão

Considere o problema de minimização. A função-objetivo é injetada na função de aptidão da seguinte maneira: Seja  $J$  o vetor das avaliações da função-objetivo para os  $N$  indivíduos da população. A equação da função de aptidão (FA) é dada por:

$$\bar{J} = \text{média}(J)$$

$$J_M = \max(J)$$

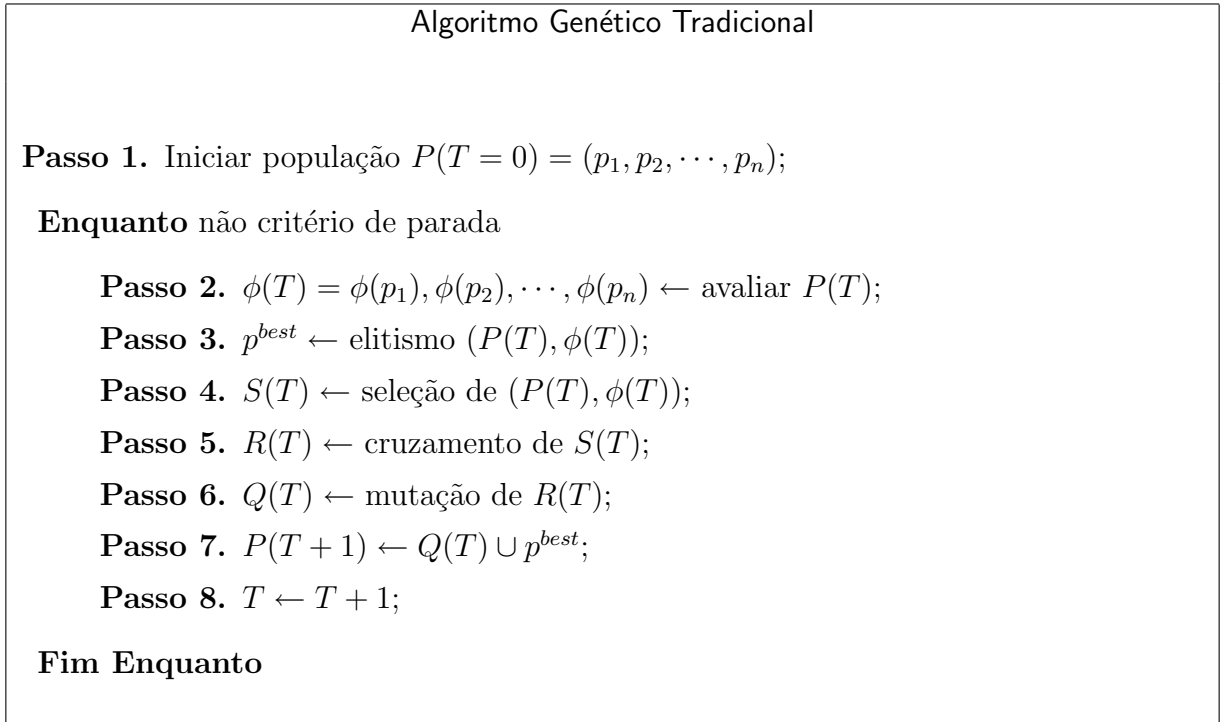


Figura 3.2: Estrutura básica de um AG tradicional.

$$J_m = \min(J)$$

$$v = \frac{\gamma J_m - J_M}{\gamma - 1}$$

$$J_m \geq v \Rightarrow \begin{cases} \alpha = \bar{J} \frac{\gamma - 1}{(J_M - \bar{J})} \\ \beta = \bar{J} \frac{J_M - \gamma \bar{J}}{J_M - \bar{J}} \end{cases}$$

$$J_m < v \Rightarrow \begin{cases} \alpha = \frac{\bar{J}}{(J_M - J_m)} \\ \beta = -\frac{\bar{J} J_m}{J_M - J_m} \end{cases}$$

$$FA = \alpha J + \beta$$

O parâmetro  $\gamma$  é chamado de fator de dispersão na função de aptidão.

### 3.1.2 Seleção

É possível escolher entre seleção por roleta ou por torneio. O método que foi utilizado nesse trabalho, é a seleção por roleta. Nesse caso, é realizada uma seleção de  $N$  indivíduos dentre os  $N$  indivíduos existentes, sendo que cada indivíduo pode ser selecionado mais de



uma vez. A probabilidade de um indivíduo ser selecionado a cada vez é igual ao valor da fração da função de aptidão em relação à soma das funções de aptidão de todos os indivíduos, ou seja,

$$p_i = \frac{FA(i)}{\sum_{i=1}^N FA(i)}$$

Desta maneira, o indivíduo com a maior fração tem maior probabilidade de ser selecionado.

### 3.1.3 Cruzamento Real Polarizado

O cruzamento real polarizado é definido da seguinte forma:

- uma população com  $N$  indivíduos aleatoriamente divididos em  $\frac{N}{2}$  pares. Para cada par, o cruzamento ocorrerá com probabilidade  $p_c$ .
- para cada par sujeito ao cruzamento, a função de aptidão  $FA(x)$  de cada indivíduo é considerada. Os indivíduos, vetores  $n$ -dimensionais das variáveis, são denominados  $x_1$  e  $x_2$ , tais que  $FA(x_2) < FA(x_1)$ .
- o cruzamento real polarizado gera um indivíduo  $x_g$  da seguinte forma:

$$x_g = \alpha x_1 + (1 - \alpha)x_2 \quad (3.1)$$

sendo  $\alpha$  pertencente ao intervalo  $[-\xi, 1 + \xi]$ , com  $\xi$  sendo um fator de extrapolação selecionado no intervalo  $[0, 1]$ . O parâmetro  $\alpha$  é selecionado de acordo com uma distribuição de probabilidade definida por:

$$\alpha = (1 + 2.\xi)\beta_1\beta_2 - \xi \quad (3.2)$$

sendo  $\beta_1$  e  $\beta_2$  variáveis aleatórias com distribuição de probabilidade uniforme no domínio  $[0, 1]$ . Isto acarreta em uma distribuição quadrática de probabilidade para  $\alpha$  garantindo que o novo indivíduo  $x_g$  tenha uma maior probabilidade de estar perto de  $x_1$  (melhor indivíduo) do que de  $x_2$  (o pior indivíduo).

- o outro indivíduo gerado pelo cruzamento é gerado sem polarização, isto é,  $\alpha$  é escolhido no intervalo  $[-\xi; 1 + \xi]$  com probabilidade uniforme.

Convém salientar que, para utilizar o cruzamento real sem polarização, basta que os dois novos indivíduos sejam da forma:

$$x_g = \alpha.x_1 + (1 - \alpha).x_2 \quad (3.3)$$

sendo  $\alpha$  escolhido no intervalo  $[-\xi, 1 + \xi]$  com probabilidade uniforme. A probabilidade do cruzamento ser utilizado com polarização é indicada por  $p_p$ . Desta forma, um indivíduo pode ou não ser gerado por polarização.

Uma avaliação dos efeitos do operador de cruzamento real polarizado pode ser encontrada em [25].

O cruzamento real polarizado imita uma busca seguindo a direção de uma tendência (com uma informação semelhante à fornecida pelo gradiente), o que não é feito por nenhum operador genético convencional, ao mesmo tempo que mantém a vantagem do AG de avaliar apenas a função-objetivo (sem nenhum cálculo de derivadas). No caso de ancestrais localizados próximos um do outro, é realizado um passo semelhante ao do algoritmo do gradiente. Isso acelera a convergência local para o ótimo. No caso de ancestrais que se encontram distantes (possivelmente em bacias de atração distintas), a operação pode ser interpretada como o seguimento de uma tendência de longa distância.

### 3.1.4 Reflexão

No caso de um indivíduo estar fora da região de interesse, o método de reflexão é aplicado para forçar o indivíduo de volta para dentro dessa região. A reflexão para o limite inferior ( $x_L$ ) é definida por

$$x_r = x_L + |x - x_L| \quad (3.4)$$

sendo  $x$  o indivíduo que está no exterior da região de interesse e  $x_r$  o indivíduo após a reflexão e  $x_L$  o vetor de limites inferiores. A reflexão para o limite superior ( $x_U$ ) é analogamente definida por

$$x_r = x_U - |x_U - x|, \quad (3.5)$$

sendo  $x_U$  o vetor de limites superiores e as demais variáveis com o mesmo significado de antes.

### 3.1.5 Mutação

O operador de mutação é definido do seguinte modo: cada indivíduo na população pode estar sujeito à mutação, com probabilidade de  $p_m$ ; se um indivíduo  $x$  sofre mutação, o

indivíduo resultante  $x_m$  é definido por

$$x_m = x + \delta, \quad (3.6)$$

sendo

$$\delta_i = 0.05\beta_i(x_r)_i \quad (3.7)$$

e  $\beta_i$  um número aleatório com distribuição Gaussiana, média zero e variância igual a 1, e  $x_r$  o vetor de diferença entre  $x_L$  e  $x_U$ , os máximos e mínimos dos parâmetros.

### 3.1.6 Elitismo

Caso o melhor indivíduo não tenha sido selecionado para a nova população, ele é nela introduzido, com a exclusão de um indivíduo qualquer, escolhido aleatoriamente.

## 3.2 Parâmetros

O algoritmo genético, conforme descrito nesse capítulo, foi utilizado em todos os testes com os seguintes parâmetros:

- Probabilidade de recombinação ( $p_c$ ): 0.6
- Probabilidade de Polarização ( $p_p$ ): 0.3
- Probabilidade de Mutação ( $p_m$ ): 0.02
- Tamanho de Mutação: 0.05
- Fator de dispersão na função de aptidão ( $\gamma$ ): 1.8
- Fator de extrapolação na recombinação ( $\xi$ ): 0.2



## Capítulo 4

# Aproximações Quadráticas em Algoritmos Genéticos Mono-Objetivo

Neste capítulo apresentaremos duas utilizações distintas das aproximações quadráticas em algoritmos genéticos mono-objetivo que se baseiam nas duas idéias descritas no capítulo 2: o uso das aproximações para corrigir as coordenadas do sistema e para obter estimativas para o mínimo do problema. Essas aproximações quadráticas devem ser construídas utilizando-se a metodologia livre de derivada descrita na seção 2.3, uma vez que o cálculo de derivadas no algoritmo genético representaria um número elevado de avaliações extras das funções.

Em primeiro lugar, aproximações quadráticas para a função-objetivo e para as restrições são construídas com o objetivo de gerar uma estimativa para o ótimo do problema restrito em questão. Nesse caso, apresentaremos uma metodologia para problemas com restrição de igualdade e outra metodologia para problemas com restrições de desigualdade. Em segundo lugar, a aproximação quadrática para a função-objetivo em um problema irrestrito é construída com o propósito de efetuar uma mudança de coordenada, criando uma configuração mais favorável ao processo de otimização.

Nos dois casos, como a aproximação quadrática só é válida localmente, apenas os indivíduos pertencentes ao interior de uma vizinhança de um determinado ponto serão utilizados na construção da aproximação através do método apresentado na seção 2.3.

Inicialmente apresentaremos uma revisão dos conceitos de otimização não-linear mono-objetivo com variáveis contínuas. Os conceitos de solução ótima, otimalidade, multimodalidade serão apresentados de forma sucinta.

## 4.1 Definições Básicas

O problema de otimização mono-objetivo pode ser definido como:

$$\begin{aligned} x^* &= \min_x f(x) \\ \text{sujeito a: } &\begin{cases} g_i(x) \leq 0; & i = 1, 2, \dots, r \\ h_j(x) = 0; & j = 1, 2, \dots, p \end{cases} \end{aligned} \quad (4.1)$$

sendo que  $x \in \mathbb{R}^n$ ,  $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $g(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^r$ , e  $h(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^p$ . As funções coordenada  $g_i$  e  $h_j$  são, respectivamente, funções de restrição de desigualdade e de igualdade. A solução para o problema (4.1) é o ponto  $x^*$ , que atende simultaneamente às restrições e que possui o menor valor para a função-objetivo  $f$ .

### 4.1.1 Condições de Otimalidade

A seguinte condição de otimalidade, conhecida como *Condição de Karush-Kuhn-Tucker*, fornece condições necessárias para que um ponto  $x^*$  seja solução do problema (4.1). Essa condições foram desenvolvidas por Kuhn e Tucker em 1951 e servem de base para diversos algoritmos de otimização.

**Proposição 1** (*Condições de Karush-Kuhn-Tucker para Otimalidade*) *Seja  $x^*$  um ponto factível para o problema de otimização. Para que  $x^*$  seja um ótimo local do problema, deve existir um conjunto de multiplicadores  $\lambda^* \in \mathbb{R}^r$  e  $\mu^* \in \mathbb{R}^p$  com  $\mu^* \geq 0$  tal que:*

$$\nabla f(x^*) + \sum_{i=1}^r \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^p \mu_j^* \nabla h_j(x^*) = 0 \quad (4.2)$$

$$\lambda_i^* g_i(x^*) = 0, \quad \forall i = 1, 2, \dots, r$$

Uma interpretação geométrica para a condição de Karush-Kuhn-Tucker pode ser vista na figura 4.1. No ponto de solução temos duas restrições ativas, portanto é possível encontrar  $\mu_1$  e  $\mu_2$ , ambos positivos, tal que a soma dos vetores gradiente na solução se anule. Ou seja,  $\nabla f(x^*) + \mu_1 \nabla g_1(x^*) + \mu_2 \nabla g_2(x^*) = 0$ .

As condições de otimalidade de Karush-Kuhn-Tucker tornam-se também condições suficientes em problemas convexos, isto é, problemas nos quais a função-objetivo e as funções de restrições são convexas.

Neste trabalho, as condições de Karush-Kuhn-Tucker serão utilizadas para desenvolver um operador especializado no tratamento de restrição não-linear de igualdade em problemas mono-objetivo.

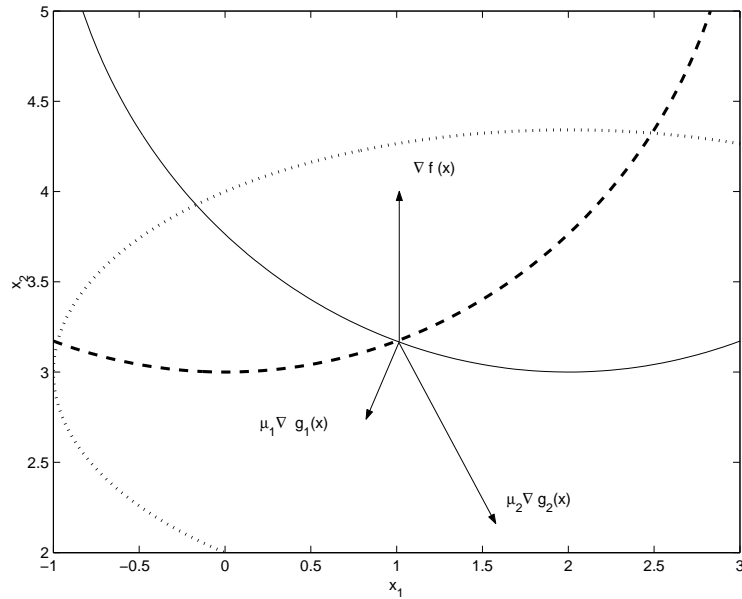


Figura 4.1: No ponto  $x^*$  os gradientes de duas restrições ativas,  $\nabla g_1$  e  $\nabla g_2$ , e o gradiente da função-objetivo  $\nabla f$  estão representados. Neste ponto as condições de Karush-Kuhn-Tucker são satisfeitas, e qualquer perturbação em  $x^*$  irá causar um aumento do valor da função-objetivo ou a infactibilidade.

#### 4.1.2 Tratamento de Restrições nos AGs

Os Algoritmos Genéticos são métodos criados para lidar originalmente com problemas irrestritos. Desta forma, os AGs necessitam de mecanismos específicos para incorporar as restrições à função-objetivo. As técnicas de tratamento de restrições em AGs, que foram desenvolvidas nos últimos anos, podem ser divididas em quatro categorias: uso de funções de penalidades; manter uma população formada apenas por indivíduos factíveis; separação de objetivos e restrições e métodos híbridos.

A maneira tradicional de incorporar restrições nos algoritmos genéticos é pelo método de penalidades, que transforma o problema restrito em problemas irrestritos aproximadamente equivalentes. No método de penalidade para incorporar restrições de desigualdade, a função de aptidão  $F(x)$  é definida como a soma da função-objetivo  $f(x)$  e o termo de penalidade, o qual depende da violação da restrição de desigualdade  $g_i(x)$ :

$$F(x) = f(x) + \sum_{j=1}^r \alpha_j g_j^+ \quad (4.3)$$

sendo que  $g_j^+$  denota o valor de  $g_j$  se este for positivo, ou zero nos outros casos. O parâmetro  $\alpha_j$  corresponde ao parâmetro de penalidade para a  $j$ -ésima restrição de desigualdade.

Usualmente cada restrição de igualdade existente no problema (4.1),  $h_j(x) = 0$ , é transformada em duas restrições de desigualdade da forma

$$\begin{aligned} h_i(x) - \epsilon &\leq 0 \\ -h_i(x) - \epsilon &\leq 0 \end{aligned} \tag{4.4}$$

e assim, o novo problema de otimização associado ao problema (4.1) terá apenas  $r + 2p$  restrições de desigualdade. O termo  $r$  na equação (4.3) pode ser trocado por  $r + 2p$  para incluir todas as restrições de igualdade e de desigualdade. Desta forma, a função de aptidão para o problema restrito (4.1) será dada por

$$F(x) = f(x) + \sum_{j=1}^{r+2p} \alpha_j g_j^+(x) \tag{4.5}$$

Com o propósito de reduzir o número de parâmetros de penalidade, as restrições são normalizadas e apenas um parâmetro de penalidade,  $\alpha$ , é utilizado. Neste trabalho, a função de penalidade utilizada nos casos em que se fez necessário é dada por:

$$F(x) = f(x) + \alpha \sum_{j=1}^{r+2p} g_j^+(x) \tag{4.6}$$

A solução para o método de penalidades se aproxima da solução exata do problema se o parâmetro  $\alpha$  for escolhido arbitrariamente grande. Do ponto de vista prático, o método de penalidades apresenta uma maneira simples e direta de tratar as restrições. Porém, existem dificuldades associadas a esse tipo de método. Se um valor muito grande de  $\alpha$  for escolhido, o problema pode se tornar mal-condicionado. Com  $\alpha$  grande, uma maior ênfase é colocada na factibilidade e a maioria dos processos de otimização irrestrita existentes tende a mover rapidamente em direção a um ponto factível. Apesar da possibilidade deste ponto se situar longe da solução ótima do problema, pode ocorrer uma parada prematura do método, especialmente se houver restrições não-lineares de igualdade (veja [19]). Por outro lado, se  $\alpha$  for muito pequeno, uma grande parte da busca será efetuada em regiões infactíveis e a solução será um ponto com grande violação das restrições. Idealmente, a penalidade deve ser mantida a mais baixa possível, logo acima do limite no qual as soluções infactíveis são ótimas (isto é chamado de *regra da penalidade mínima* (veja [26])). Esse fato, apesar de conceitualmente simples, na prática torna-se difícil, uma vez que a fronteira entre a região factível e infactível não é conhecida na maioria dos casos.

Os AGs podem ainda lidar com as restrições impedindo a incorporação de indivíduos que violem as mesmas. Nesse caso parte-se da premissa que a factibilidade possa ser resolvida



*a priori*, seja analiticamente, seja por meio de uma busca aleatória que conduza com alta probabilidade a pontos factíveis. Existem ainda outras técnicas novas que não se enquadram em nenhuma das 4 categorias descritas acima. Um tratamento amplo que trata desse assunto pode ser encontrado em [27].

Em todos os testes deste capítulo, o algoritmo genético, conforme descrito no capítulo 3, foi inicializado com os mesmos parâmetros listados a seguir:

- Probabilidade de recombinação ( $p_c$ ): 0.6
- Probabilidade de Polarização ( $p_p$ ): 0.3
- Probabilidade de Mutação ( $p_m$ ): 0.02
- Tamanho de Mutação: 0.05
- Fator de dispersão na função de aptidão ( $\gamma$ ): 1.8
- Fator de extrapolação na recombinação ( $\xi$ ): 0.2

## 4.2 Tratamento de Restrições de Igualdade no AG

Restrições de igualdade constituem um problema para os AGs, uma vez que esse tipo de restrição define um conjunto factível de dimensão menor que a região de busca do problema. Os AGs, heurísticas cuja busca é essencialmente baseada na amostragem de todo o espaço, têm baixa probabilidade de encontrar soluções factíveis. Sabemos que a natureza aleatória dos operadores dos AGs permite a busca por regiões ótimas (bacias de atração) no espaço global. Esses operadores fazem com que a busca torne-se *espalhada* pelo espaço, maximizando a chance de encontrar outras bacias de atração. Podemos chamar essa propriedade de *busca por volume*. Entretanto, a aleatoriedade de tais operadores é conflitante com a necessidade de uma busca em objetos bem definidos de *volume zero*, tais como os conjuntos factíveis definidos por restrições de igualdade. A propriedade de *busca por volume* irá produzir um tipo de *movimento de afastamento aleatório* em relação ao objeto factível, o que acarretará na convergência a taxas mais lentas.

Tendo em mente essas dificuldades, desenvolvemos uma metodologia especializada no tratamento de restrição de igualdade em algoritmos genéticos.<sup>1</sup>

Considere um problema da forma (4.1) com apenas uma restrição de igualdade. Podemos construir uma aproximação quadrática para a função-objetivo e para a restrição, obtendo desta forma um problema quadrático associado. É possível encontrar a solução analítica para

---

<sup>1</sup>Os resultados que serão apresentados nessa seção encontram-se publicados em [28].

esse problema associado, porém, de uma maneira geral, essa solução obtida não corresponde à solução do problema original. Entretanto, se as aproximações quadráticas forem precisas, a solução do problema quadrático será uma boa aproximação para a solução do problema original.

Motivado por essas considerações, um novo operador de busca local, especializado no tratamento de restrições de igualdade em problemas de otimização, está sendo proposto. A idéia principal é restringir o AG a encontrar soluções que estão no interior de um objeto que possui a dimensão igual à do conjunto factível. Este objeto é uma aproximação de segunda ordem da superfície factível. Essa solução obtida deve então ser introduzida na população do AG. Desta forma, o operador de busca local pode ser dividido em duas etapas:

- a aproximação de segunda ordem deve ser iterativamente atualizada, e
- o processo evolucionário de busca deve ser feito de forma que as soluções estejam na fronteira da superfície aproximada.

Um operador que segue essas etapas é, então, incluído no AG e desta maneira, vai se opor ao *movimento de afastamento aleatório* descrito acima.

Esse operador, aqui denominado *Aproximação Quadrática para o Ótimo Restrito* (AQR), pode ser interpretado também como um operador de *elitismo* especializado em melhorar, iteração após iteração, a estimativa do conjunto factível. O operador de elitismo convencional, que mantém um único ponto (ou um conjunto de pontos) em cada geração, não promove uma busca pelo conjunto factível, uma vez que esse conjunto é uma superfície  $m$ -dimensional e não um conjunto discreto de pontos. O operador aqui proposto garante uma aproximação para esse objeto no seguinte sentido: a melhor superfície de segunda ordem que aproxima o conjunto factível é mantida e melhorada à medida que o algoritmo prossegue.

### 4.2.1 O operador de Busca Local

Considere um problema não-linear com uma restrição não-linear de igualdade:

$$\begin{aligned} x^* &= \min_x f(x) \\ \text{sujeito a: } &g(x) = 0 \end{aligned} \tag{4.7}$$

sendo  $f$  e  $g$  funções reais, não-lineares e suaves.

Usando as avaliações dos indivíduos disponíveis pelo AG e através da metodologia des-

crita na seção 2.3, obtemos aproximações quadráticas para as funções  $f$  e  $g$  da forma:

$$\tilde{f}(x) = (x - x_f)^T H(x - x_f) - c_1 \quad (4.8)$$

e

$$\tilde{g}(x) = (x - x_g)^T G(x - x_g) - c_2 \quad (4.9)$$

sendo  $x_f$  o ponto de mínimo irrestrito da função-objetivo aproximada e  $x_g$  o ponto de mínimo da restrição aproximada. De posse dessas aproximações, temos as informações necessárias que serão usadas nessa metodologia: a matriz Hessiana e os pontos de mínimo de  $f$  e  $g$ .

Com as aproximações quadráticas obtidas e usando uma mudança de coordenadas apropriada, podemos escrever o problema quadrático associado:

$$x^* = \arg \min_x \hat{f}(z) = (z - z_f)^T (z - z_f) - c_1 \quad (4.10)$$

$$\text{sujeito a: } \hat{g}(z) = (z - z_g)^T Q(z - z_g) - c_2 = 0$$

sendo  $z_f$  o mínimo irrestrito da nova função-objetivo,  $z_g$  o ponto central da restrição quadrática e  $Q$  a matriz Hessiana associada à restrição quadrática.

Considerando os vetores gradientes de  $\hat{f}$  e  $\hat{g}$ ,

$$\begin{aligned} \nabla \hat{f} &= 2(z - z_f) \\ \nabla \hat{g} &= 2Q(z - z_g) \end{aligned}$$

e substituindo essas expressões na condição de Karush-Kuhn-Tucker para esse problema,

$$\nabla \hat{f} + \lambda \nabla \hat{g} = 0,$$

obtemos, após um pouco de álgebra, a equação

$$(I + \lambda Q)z = z_f + \lambda Q z_g \quad (4.11)$$

que descreve um caminho parametrizado por  $\lambda$ , no qual os gradientes de  $\hat{f}$  e  $\hat{g}$  são linearmente dependentes. Este caminho pode ser re-parametrizado por

$$((1 - \gamma)I + \gamma P)z = (1 - \gamma)z_f + \gamma.Q.z_g \quad (4.12)$$

em que  $0 < \gamma < 1$ . Observe que, fazendo  $\gamma = 0$  obtemos  $z = z_f$  e fazendo  $\gamma = 1$  obtemos  $z = z_g$ .

Vamos ilustrar a metodologia em um caso de dimensão dois. Suponha um problema

bidimensional da forma

$$x^* = \min_x f(x) \tag{4.13}$$

$$\text{sujeito: } g(x) = 0$$

onde  $f(\cdot) : \mathbb{R}^2 \mapsto \mathbb{R}$  é uma função real convexa e  $g$  é uma função quadrática. A figura 4.2 mostra, para este exemplo, as curvas de nível da função  $f$  juntamente com a curva que representa a restrição. Na figura estão assinalados os mínimos das funções  $f$  e  $g$ , respectivamente  $x_f$  e  $x_g$  e a solução do problema representado pelo ponto  $x^*$ . Observe que esse ponto está sobre a restrição e além disso satisfaz as condições de Karush-Kuhn-Tucker (vetores gradientes linearmente dependentes). A figura 4.3 ilustra o caminho unindo  $x_f$  e  $x_g$  e que contém o ponto  $x^*$  e é parametrizado pela equação

$$(I + \lambda Q)x = x_f + \lambda Qx_g. \tag{4.14}$$

Observando o caminho que une  $x_f$  e  $x_g$ , vemos que todos os valores de  $x$  compreendidos no intervalo  $[x_f, x^*)$  não satisfazem a restrição e, portanto  $g(x) > 0$ . Da mesma forma, os valores de  $x$  no intervalo  $(x^*, x_g]$  estão do outro lado e portanto  $g(x) < 0$ . Assim, dado algum intervalo deste caminho, é possível garantir a existência de um ponto  $x^*$  tal que  $g(x^*) = 0$ . É possível encontrar esse ponto  $x^*$  através de um método de bisseção. Um algoritmo baseado no método de bisseção, capaz de encontrar o ponto  $x^*$ , está apresentado na figura 4.4.

Após o método de bisseção, devemos retornar às coordenadas do problema original (4.7), nas quais a função-objetivo está sendo calculada. Desta forma, o ponto obtido representa a solução para o problema quadrático (4.10).

A figura 4.5 mostra um esquema do operador de busca local, AQR, apresentado nessa sub-seção.

## 4.2.2 AGRP-AQR Híbrido

A nova metodologia, AGRP-AQR Híbrido (Algoritmo Genético Real Polarizado com o operador AQR), descrito nesse trabalho, lida com restrições não-lineares de igualdade e utiliza a técnica de aproximação quadrática e o método de bisseção como um operador de busca local no AGRP. Este novo operador de busca local melhora a solução uma vez que permite que a restrição de igualdade seja atingida com maior precisão. Vale ressaltar que esse operador não acrescenta nenhum custo computacional adicional para o AGRP, se o custo computacional for medido pelo número de avaliações da função-objetivo.

Uma implementação dessas idéias pode ser feita de várias maneiras. Com o objetivo principal de mostrar a relativa melhora da solução que o operador proporciona, usamos uma

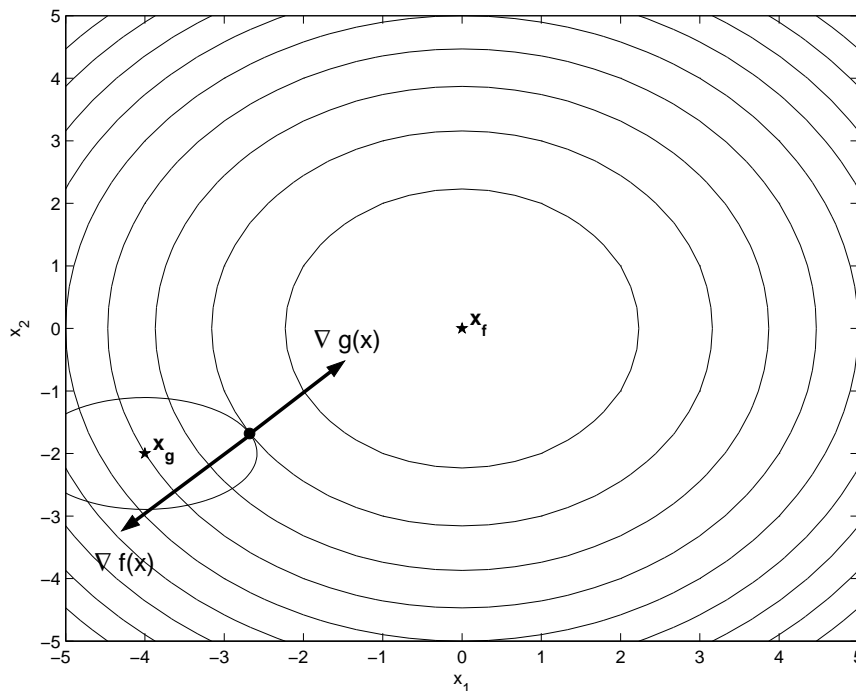


Figura 4.2: Curvas de nível de  $f$  com a restrição não-linear de igualdade e a solução  $x^*$  em um problema do tipo (4.13).

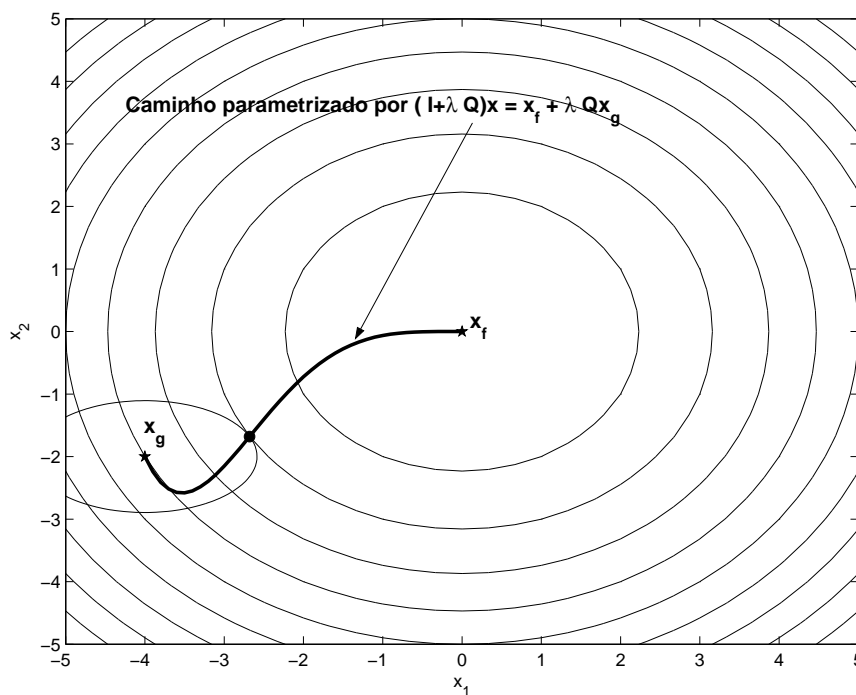


Figura 4.3: Caminho, unindo  $x_f$  e  $x_g$ , que contém  $x^*$ , representado na figura pelo círculo preto.

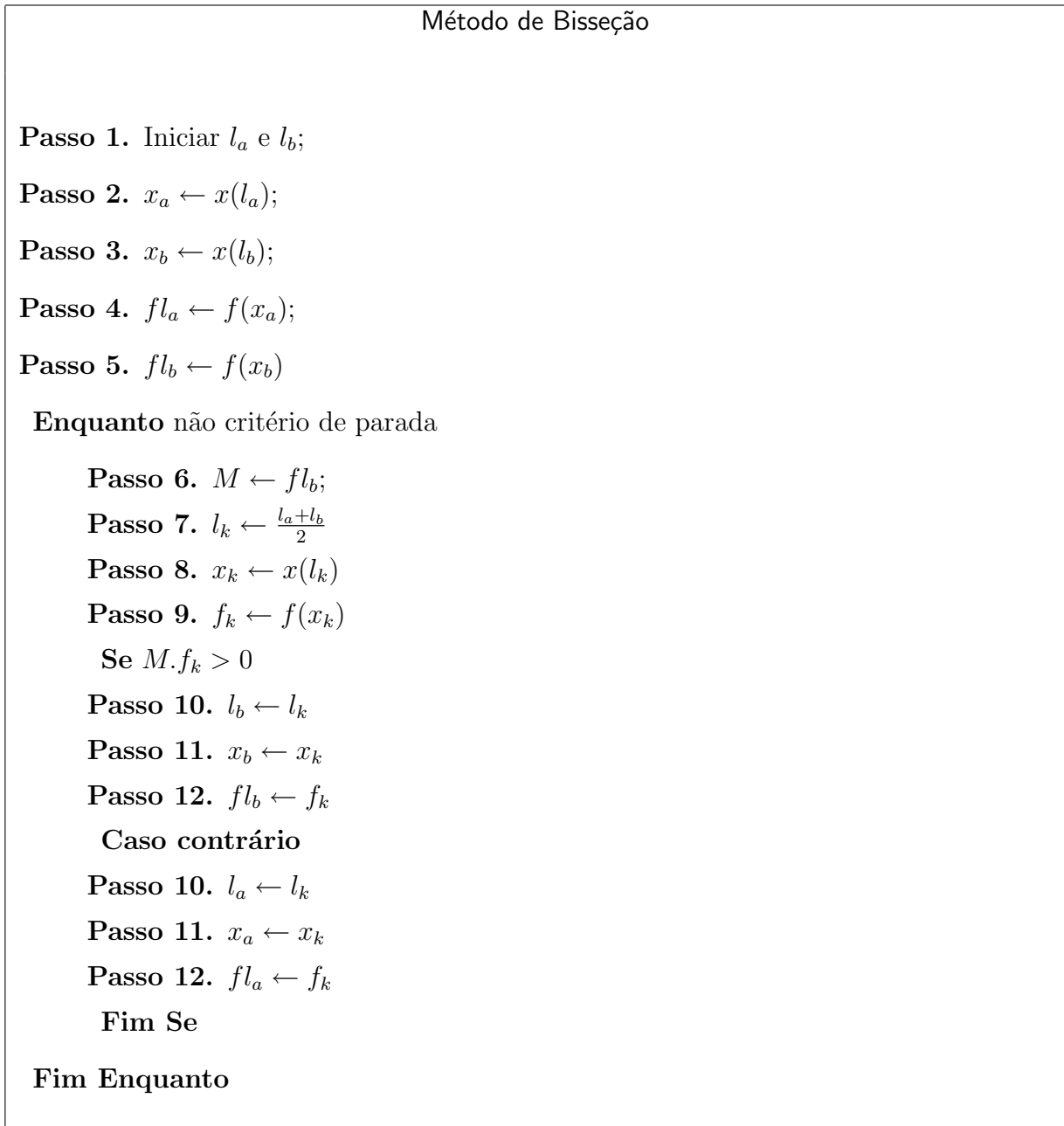


Figura 4.4: Esquema básico do método de bisseção utilizado no operador de busca local para problemas com restrição de igualdade.

série de definições arbitrárias que definem como o operador será acoplado ao AGRP. Para hibridizar o operador com o algoritmo base, estabelecemos as seguintes definições:

- o operador de busca local, AQR, será executado a cada cinco gerações. <sup>2</sup>
- apenas os pontos no interior de uma vizinhança do melhor ponto serão utilizados na

<sup>2</sup>Experimentamos uma variação desse número, mas nenhuma mudança nos resultados foi verificada.

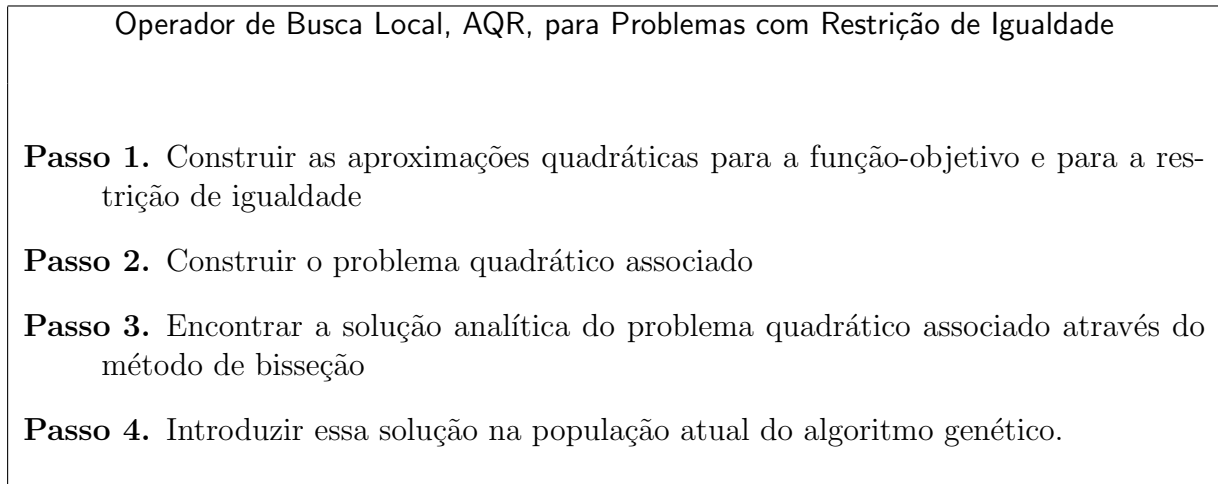


Figura 4.5: Esquema básico do operador de busca local especializado em problemas com restrição não-linear de igualdade.

construção das aproximações quadráticas. Esta vizinhança é uma elipse cujos eixos correspondem a 10% da variação de cada parâmetro. Se o número de pontos nessa vizinhança for menor que o número de pontos necessários para se construir a aproximação, o operador de busca local não será executado.

- finalmente, o ponto de retorno do operador AQR irá deterministicamente substituir o pior ponto da população atual.

Observe que o ponto de retorno do operador ainda poderia ser usado como um critério de parada adicional do algoritmo: a estabilização de tal ponto pode ser interpretada como se o algoritmo tivesse encontrado a solução.

Por se tratar de um problema restrito, a restrição será incorporada à função de aptidão através de uma função de penalidade conforme descrito no início desse capítulo. A restrição de igualdade  $h(x)$  deve ser transformada em duas restrições de desigualdade  $\tilde{h}_1$  e  $\tilde{h}_2$  e a função de aptidão será dada por

$$F(x) = f(x) + 10 \sum_{j=1}^2 \tilde{h}_j^+(x).$$

Observe que a função de aptidão utilizada durante a execução do AG leva em consideração as funções originais do problema de otimização. As aproximações quadráticas para essas funções só serão usadas no operador de busca local.

Apresentamos na figura 4.6, um esquema de um algoritmo genético híbrido para tratamento de restrição não-linear de igualdade.

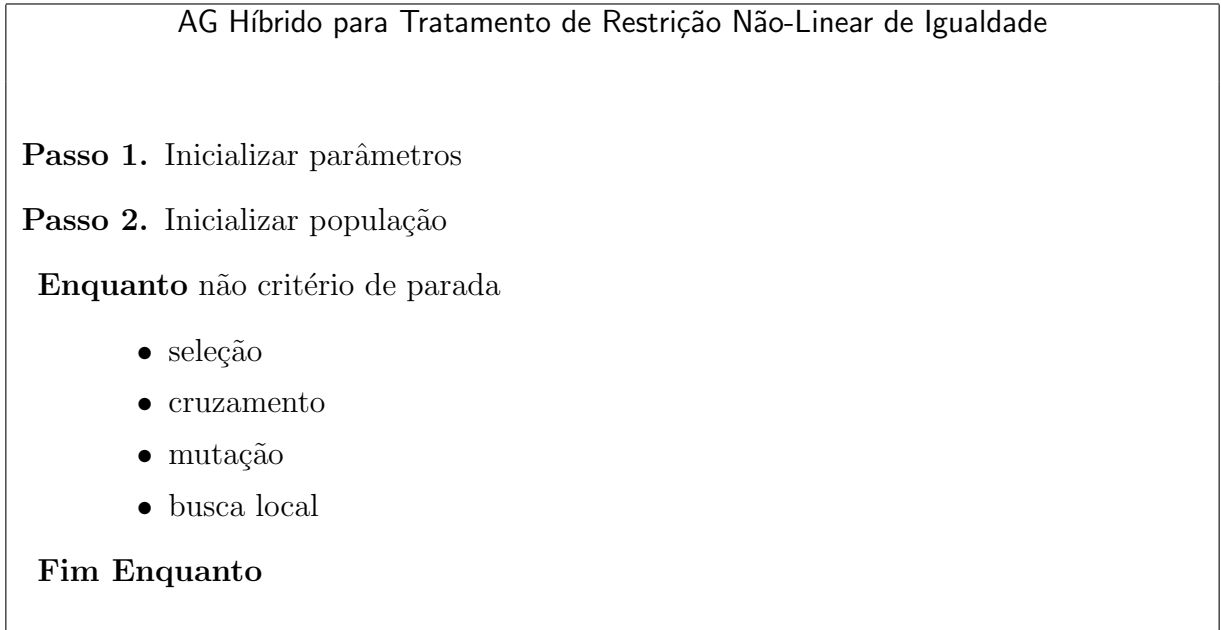


Figura 4.6: Esquema básico de um AG híbrido para tratamento de restrição não-linear de igualdade.

### 4.2.3 Testes e Resultados

Os algoritmos AGRP e o AGRP-AQR Híbrido foram testados em um conjunto de problemas analíticos. Esses problemas foram escolhidos com diferentes graus de dificuldade e características diferentes. Os problemas estão listados a seguir:

(P1) Problema Quadrático:

$$\begin{aligned}
 x^* &= \arg \min_x x_1^2 + x_2^2 + x_3^2 \\
 \text{sujeito a: } &\begin{cases} (x_1 - 2)^2 + (x_2 - 1)^2 + 4(x_3 - 1)^2 = 1 \\ -4 < x_i < 4, \quad i = 1, 2, 3 \end{cases} \quad (4.15)
 \end{aligned}$$

(P2) Problema Não-Quadrático:

$$\begin{aligned}
 x^* &= \arg \min_x (x_1 - 2)^4 + (x_1 - 2x_2)^2 \\
 \text{sujeito a: } &\begin{cases} x_1^2 - x_2 = 0 \\ -4 < x_i < 4, \quad i = 1, 2 \end{cases} \quad (4.16)
 \end{aligned}$$



(P3) Problema Multimodal

$$x^* = \arg \min_x x^T . A^T . A . x - 10 . [1 \quad 1] . \cos(2\pi Ax)$$

$$\text{sujeito a: } \begin{cases} (x_1 - 2)^2 + (x_2 - 2)^2 = 1 \\ -4 < x_i < 4, \quad i = 1, 2 \end{cases} \quad (4.17)$$

sendo

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$$

É importante esclarecer que uma função escalar aplicada sobre um argumento vetorial resulta em um vetor onde em cada coordenada é aplicada na função escalar.

(P4) Problema com Restrição Não-Quadrática:

$$x^* = \arg \min_x 100(x_1^2 - x_2)^2 + (1 - x_1)^2$$

$$\text{sujeito a: } \begin{cases} 2V + \sum_{i=1}^{10} (-x_i . \text{sen}(\sqrt{|x_i|})) \\ -4 < x_i < 4, \quad i = 1, 2 \end{cases} \quad (4.18)$$

sendo  $V = 418.9829101$ . Esse valor de  $V$  é utilizado apenas para deslocar o mínimo irrestrito da função de restrição para o zero.

Cada algoritmo foi executado 30 vezes para cada problema e foi iniciado com uma população de 20 indivíduos e com um número máximo de gerações igual a 100. O número máximo de gerações foi o único critério de parada utilizado para os dois algoritmos. Ao final das execuções do AGRP e do AGRP-AQR Híbrido, a curva média de convergência foi obtida. Esta curva corresponde ao valor médio do melhor indivíduo ao longo das 100 gerações.

As figuras 4.7, 4.8, 4.9 e 4.10 mostram a curva de convergência para cada problema (P1), (P2), (P3) e (P4) respectivamente. Nos gráficos, o eixo das abscissas representa a geração e o eixo das ordenadas representa o valor do logaritmo na base 10 da função-objetivo do melhor indivíduo da correspondente geração. O logaritmo foi usado, exceto no problema não-quadrático (P2) e no problema multimodal (P4), apenas para reforçar a diferença entre as curvas.

Na figura 4.7, podemos ver que o AGRP-AQR Híbrido apresenta uma velocidade de convergência maior se comparada à velocidade de convergência do AGRP. Em torno da 10ª geração, o AGRP-AQR Híbrido encontrou uma solução melhor que a solução final obtida pelo AGRP. Um resultado semelhante pode ser visto se observarmos a curva de convergência das

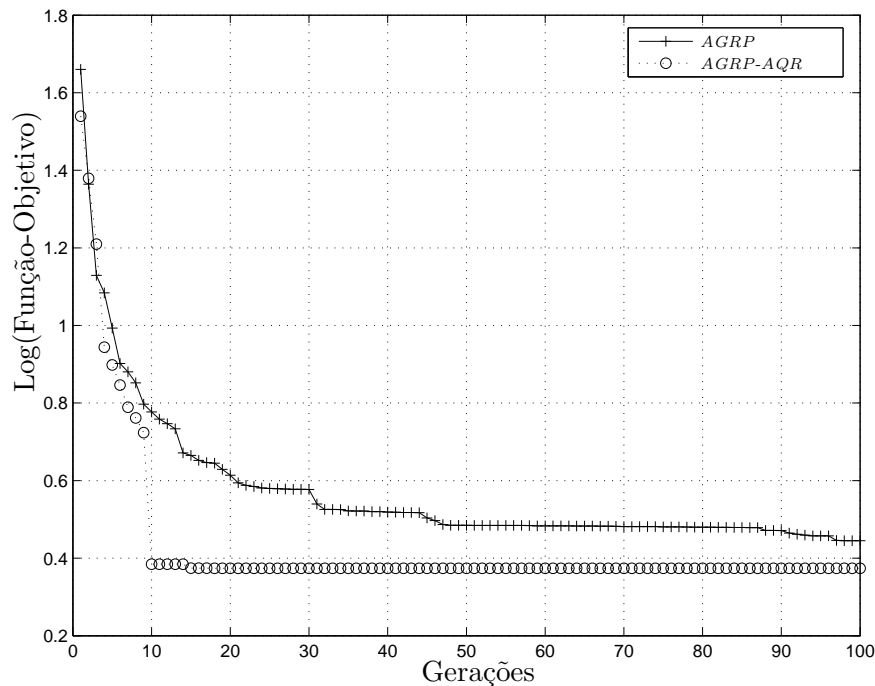


Figura 4.7: Curva de convergência para o Problema Quadrático (P1) obtida na resolução de um problema mono-objetivo com uma restrição de igualdade.

figuras 4.9 e 4.10. Para o problema não-quadrático (fig. 4.8), a velocidade de convergência do AGRP-AQR Híbrido é também maior que a velocidade de convergência do AGRP.

Uma vez que o operador de busca local aqui proposto, o operador AQR, substitui o pior indivíduo pelo indivíduo resultante da busca local, uma importante questão que surge é sobre o efeito deste operador na diversidade da população.

O seguinte experimento foi executado com o objetivo de analisar a diversidade da população ao longo do algoritmo. As etapas desse experimento estão descritas abaixo:

1. Defina um raio  $r$  no espaço de busca;
2. Execute o AG, retornando a população final;
3. Coloque uma esfera de raio  $r$  centrado no ponto ótimo;
4. Inicialize o contador de esfera como 1;
5. Exclua todos os indivíduos que estão localizados no interior dessa esfera;
6. Dentre os indivíduos restantes, centre uma nova esfera de mesmo raio no indivíduo que está mais próximo do centro anterior e incremente o contador de esferas;

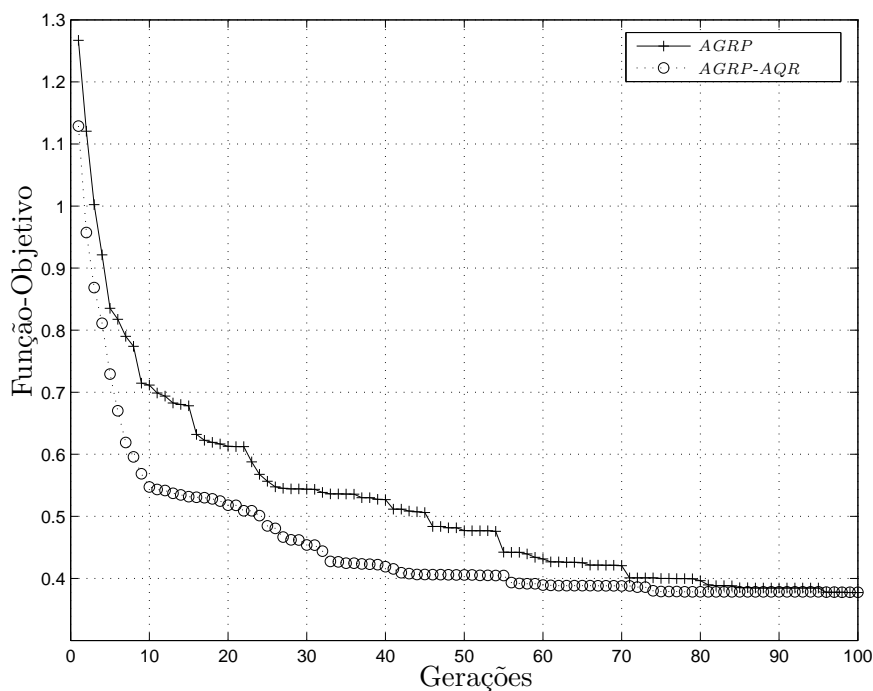


Figura 4.8: Curva de convergência para o Problema Não-Quadrático (P2) obtida na resolução de um problema mono-objetivo com uma restrição de igualdade.

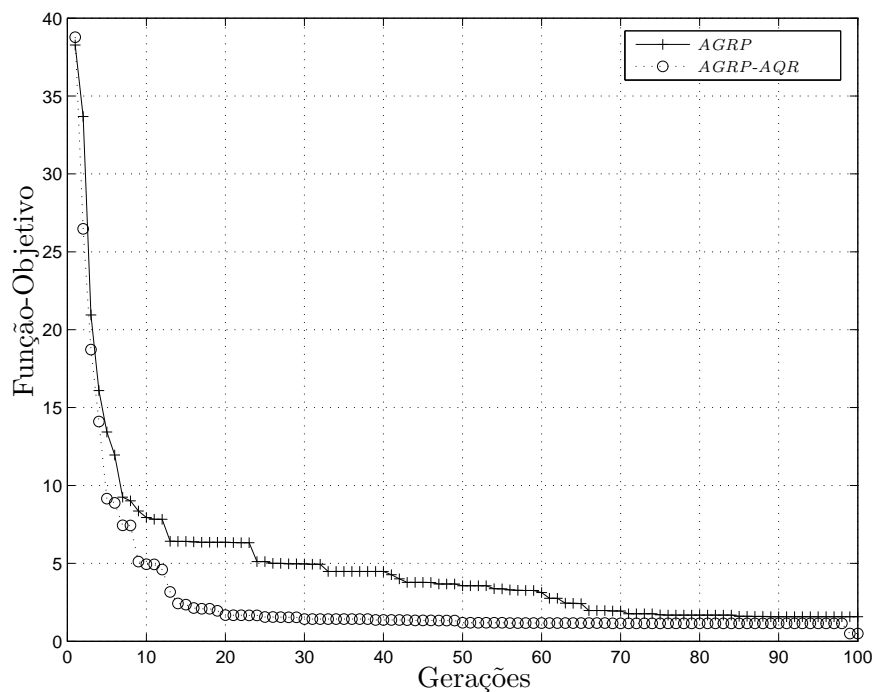


Figura 4.9: Curva de convergência para o Problema Multimodal (P3) obtida na resolução de um problema mono-objetivo com uma restrição de igualdade.

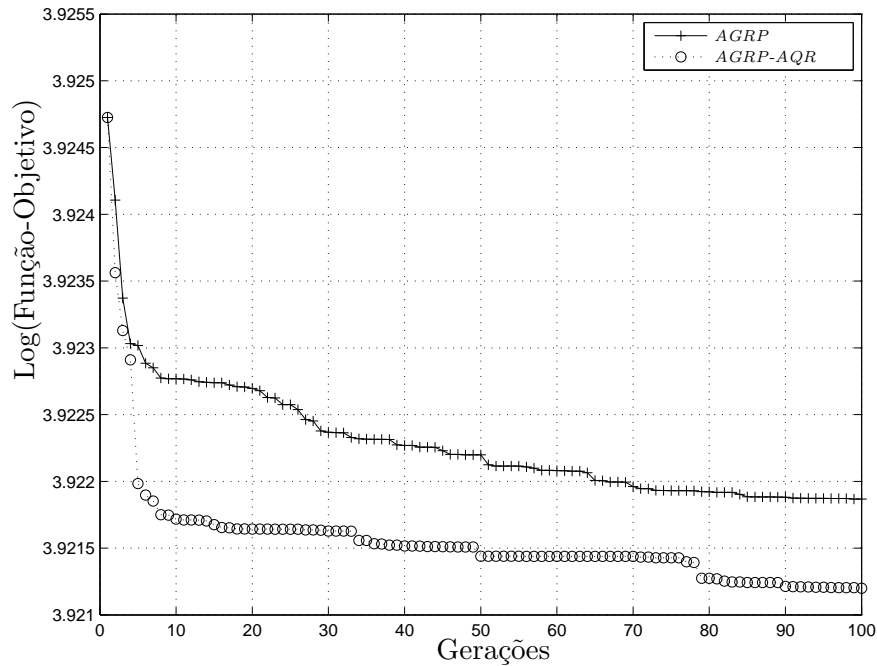


Figura 4.10: Curva de convergência para o Problema com Restrição Não-Quadrática (P4) obtida na resolução de um problema mono-objetivo com uma restrição de igualdade.

7. Vá para o passo 5 até que não reste mais nenhum ponto na população.

Através desse experimento, é possível identificar a população com maior diversidade: simplesmente é a população que obteve um maior número de esferas. Entretanto, com o objetivo de se obter um resultado relevante, devemos rodar esse experimento em várias execuções de cada algoritmo e calcular o valor médio. Rodamos o AGRP e o AGRP-AQR Híbrido, com os mesmos parâmetros listados anteriormente, 30 vezes nos seguintes problemas:

- Rastrigin Rotacionada Bidimensional

$$\begin{aligned}
 x^* &= \arg \min_x x^T \cdot A^T \cdot A \cdot x - 10 \cdot [1 \quad 1] \cdot \cos(2\pi Ax) \\
 \text{sujeito a: } &\begin{cases} (x_1 - 0.9)^2 + (x_2 - 0.9)^2 = 1 \\ -5 < x_i < 5, \quad i = 1, 2 \end{cases} \quad (4.19)
 \end{aligned}$$

sendo

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$$

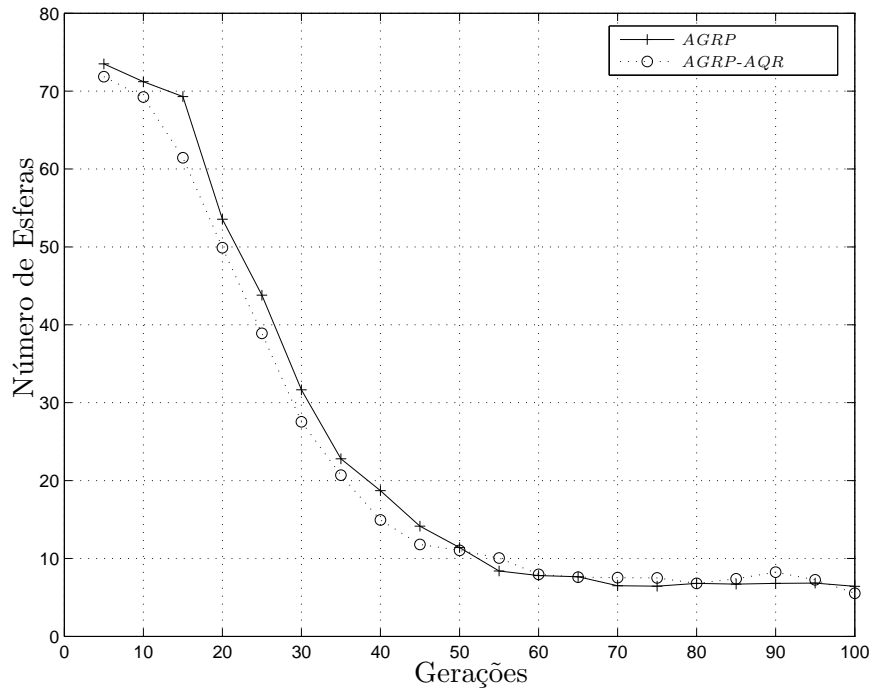


Figura 4.11: Análise da diversidade ao utilizar o operador AQR no problema Rastrigin bidimensional com uma restrição de igualdade.

- Rastrigin tridimensional

$$x^* = \arg \min_x x^T \cdot x - 10 \cdot [1 \quad 1 \quad 1] \cdot \cos(2\pi x)$$

$$\text{sujeito a: } \begin{cases} (x_1 - 0.65)^2 + (x_2 - 0.65)^2 + (x_3 - 0.65)^2 = 1 \\ -5 < x_i < 5, \quad i = 1, 2, 3 \end{cases} \quad (4.20)$$

Para o raio da esfera, o valor  $r = 0.0001$  foi escolhido. Os resultados finais estão mostrados na tabela 4.1. As figuras 4.11 e 4.12 mostram a análise de diversidade, como descrito acima, a cada 5 gerações, para os problemas 4.19 e 4.20.

Problema	AGRP	AGRP-AQR Híbrido
Rastrigin 2D	7.4333	6.4667
Rastrigin 3D	8.3333	7.6000

Tabela 4.1: Valores médios do número de esferas na análise de diversidade em problemas mono-objetivo com restrições de igualdade.

De fato, como podemos observar, o operador AQR produz uma pequena redução na diversidade da população. Este fato é esperado, uma vez que uma melhora do melhor indivíduo implica no possível aumento da possibilidade de extinção dos piores indivíduos.

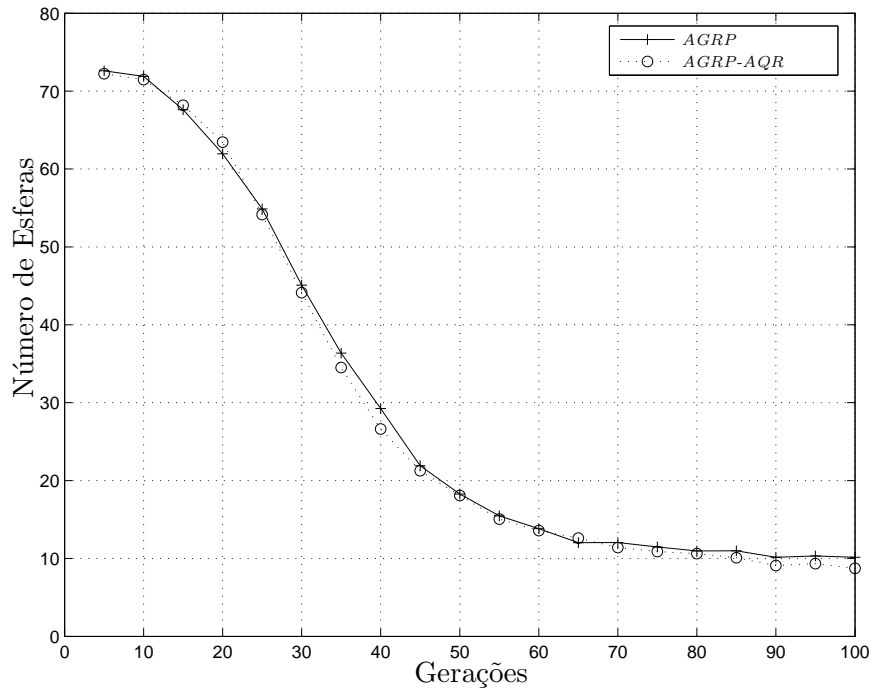


Figura 4.12: Análise da diversidade ao utilizar o operador AQR no problema Rastrigin tridimensional com uma restrição de igualdade.

#### 4.2.4 Conclusões

O algoritmo AGRP-AQR Híbrido apresentou um bom desempenho em problemas com restrições de igualdade. O uso das aproximações quadráticas para ambas as funções, objetivo e de restrição, e um tipo de elitismo presente no operador melhora as propriedades de convergência do AGRP. Os resultados confirmam que o AGRP-AQR Híbrido converge para uma solução melhor e ainda, diminui o número de gerações necessário para alcançar essa solução.

Esse operador AQR não requer nenhum cálculo adicional de avaliação de função por iteração do algoritmo, o que permite que a inclusão desse operador seja recomendada em problemas com restrições de igualdade, principalmente no caso de problemas cujo cálculo da função-objetivo seja computacionalmente caro.

### 4.3 Tratamento de Restrições de Desigualdade no AG

Apresentamos na seção anterior uma metodologia para o tratamento de problemas mono-objetivo com uma única restrição de igualdade que pode ser acoplado a um algoritmo genético. Essa metodologia se baseava na construção de aproximações quadráticas para as funções objetivo e de restrição. Seguindo essa mesma linha de utilização de aproximações

quadráticas para as funções, apresentaremos neste capítulo uma nova metodologia para auxiliar na resolução de problemas mono-objetivos com múltiplas restrições de desigualdade<sup>3</sup>.

Essa metodologia pode ser vista como um operador de busca local e utiliza aproximações (quadráticas ou lineares) para todas as funções envolvidas no problema. Durante a etapa de busca local, essas aproximações fornecem um problema associado cuja função-objetivo é quadrática e cujas restrições de desigualdade são quadráticas ou lineares. Este problema associado é resolvido usando uma formulação baseada na técnica de LMI (*Desigualdade Matricial Linear*) [29]<sup>4</sup>. A solução deste problema associado deve ser incluída na população corrente do AG. À medida que as aproximações tornam-se mais precisas, a solução do problema associado se aproxima da solução do problema original.

Essa metodologia aqui proposta guia o AG na direção da região factível do espaço de busca, e desta forma ajuda o AG a encontrar a solução com maior precisão e com maior rapidez.

### 4.3.1 Obtendo as Aproximações

Considere o seguinte problema de otimização:

$$\begin{aligned} & \text{Minimize } f(x) \\ & \text{sujeito a: } g_i(x) \leq 0, \quad i = 1, \dots, k \end{aligned} \tag{4.21}$$

sendo a função-objetivo real e não-linear e as restrições de desigualdade podendo ser lineares ou não-lineares. O espaço de busca  $\mathcal{S}$  é um retângulo  $n$ -dimensional formado pelos limites inferiores e superiores das variáveis do problema  $x_j^l \leq x_j \leq x_j^u$ , com  $j = 1, 2, \dots, n$ .

Motivado pelo fato de que os AGs trabalham com uma população de possíveis soluções e geram, para esses pontos, os valores da função-objetivo e das restrições, usaremos essas informações disponíveis para gerar as aproximações, sejam elas quadráticas ou lineares. As aproximações quadráticas devem ser geradas utilizando a metodologia livre de derivada descrita na seção 2.3. Nos casos em que as funções originais não admitam uma aproximação localmente convexa, podemos então usar uma aproximação linear para essas funções.

Apenas por conveniência, a metodologia para construção de uma aproximação linear será descrita a seguir.

O método de quadrados mínimos é um procedimento simples e eficiente para gerar uma

---

<sup>3</sup>Os resultados que serão apresentados nessa seção encontram-se em preparação para publicação.

<sup>4</sup>Tradução para Linear Matrix Inequality.

aproximação linear dado um conjunto de pontos. Esse método supõe que a melhor aproximação linear é a curva que garante que a soma dos quadrados dos erros tenha valor mínimo para um dado conjunto de pontos. A metodologia desse método será brevemente descrita.

Seja  $g$  uma função. Dados pontos distintos  $x_1, x_2, \dots, x_m$  precisamos encontrar uma função real e linear  $f_l$  que representa a melhor aproximação linear para  $g$  no sentido de quadrados mínimos, isto é,

$$g(x_i) \approx f_l(x_i) \quad (4.22)$$

para  $i = 1, \dots, m$ .

Como  $f_l$  é uma função afim, podemos escrevê-la como

$$f_l(x) = a_0 + a_1x_1 + a_2x_2 + \dots + a_nx_n \quad (4.23)$$

sendo  $n$  o número de variáveis da função. Sendo  $g_i = g(x_i)$ , se  $m \geq n$ , podemos encontrar a solução de quadrados mínimos

$$\min S = \sum_{i=1}^m (g_i - f_l(x_i))^2 \quad (4.24)$$

ou ainda

$$\min S = \sum_{i=1}^m (g_i - a_0 + a_1x_1 + a_2x_2 + \dots + a_ix_i)^2 \quad (4.25)$$

Calculando todas as derivadas de  $S$  em relação a  $(a_j)_{j=0}^n$  e igualando cada uma a zero,

$$\frac{\partial S}{\partial a_j} = 0 \quad (4.26)$$

$$j = 0, 1, \dots, n$$

obtemos um sistema linear com  $n$  equações e  $n$  variáveis,

$$AX = b \quad (4.27)$$



com

$$A = \begin{pmatrix} m & \sum_{i=1}^m (x_1)_i & \cdots & \sum_{i=1}^m (x_m)_i \\ \sum_{i=1}^m (x_1)_i & \sum_{i=1}^m ((x_1)_i)^2 & \cdots & \sum_{i=1}^m (x_1)_i \cdot (x_m)_i \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^m (x_m)_i & \sum_{i=1}^m (x_1)_i \cdot (x_2)_i & \cdots & \sum_{i=1}^m ((x_m)_i)^2 \end{pmatrix}, \quad (4.28)$$

$$X = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} \quad (4.29)$$

e

$$b = \begin{pmatrix} \sum_{i=1}^m g_i \\ \sum_{i=1}^m g_i \cdot (x_1)_i \\ \vdots \\ \sum_{i=1}^m g_i \cdot (x_m)_i \end{pmatrix}. \quad (4.30)$$

Uma vez que a matriz  $A$  é simétrica e não-singular, podemos resolver o sistema linear  $AX = b$  e obter a seguinte solução:

$$X = A^{-1} \cdot b \quad (4.31)$$

Os valores de  $X$  são os parâmetros da aproximação linear para a função  $g$ .

### 4.3.2 O operador de Busca Local

De posse das aproximações quadráticas e lineares, podemos escrever o problema quadrático associado ao problema (4.21):

$$\begin{aligned} & \min (x - x_f)^T \cdot H \cdot (x - x_f) \\ \text{s.a.: } & \begin{cases} g_i(x) & = (x - x_{g_i})^T \cdot Q_i \cdot (x - x_{g_i}) - C_i \leq 0 \\ g_j(x) & = A_j \cdot x + c_j \leq 0 \\ i & = 1, \dots, p \\ j & = 1, \dots, q \end{cases} \end{aligned} \quad (4.32)$$

sendo  $x_f$  e  $H$ , respectivamente, o mínimo irrestrito e a matriz Hessiana da função-objetivo quadrática. Os símbolos  $x_{g_i}$  e  $Q_i$ , para  $i = 1, \dots, m$  correspondem, respectivamente, ao ponto central e a matriz Hessiana de cada restrição quadrática, e  $g_{l_j}$  é a aproximação linear para alguma restrição.

Usando o problema (4.32) podemos estimar uma solução para o problema (4.21) da seguinte maneira. Usando os resultados apresentados no apêndice B, podemos re-escrever o problema (4.32) por meio do lema de Schur:

$$\begin{aligned}
 x^* = \arg_x \min_{x, \epsilon} \epsilon \\
 \text{sujeito a: } \left\{ \begin{array}{l}
 \left[ \begin{array}{cc} \epsilon & (x - x_f)' \\ x - x_f & H^{-1} \end{array} \right] > 0 \\
 \left[ \begin{array}{cc} C_1 & (x - x_{g_1})' \\ x - x_{g_1} & Q_1^{-1} \end{array} \right] > 0 \\
 \left[ \begin{array}{cc} C_2 & (x - x_{g_2})' \\ x - x_{g_2} & Q_2^{-1} \end{array} \right] > 0 \\
 \vdots \\
 \left[ \begin{array}{cc} C_p & (x - x_{g_p})' \\ x - x_{g_p} & Q_p^{-1} \end{array} \right] > 0 \\
 2a_1 - c_1x - x'c'_1 > 0 \\
 2a_2 - c_2x - x'c'_2 > 0 \\
 \vdots \\
 2a_q - c_qx - x'c'_q > 0
 \end{array} \right. \tag{4.33}
 \end{aligned}$$

Como o complemento de Schur transforma termos quadráticos em termos lineares aumentados, temos as seguintes associações: a primeira restrição de (4.33) diz respeito à transformação da função-objetivo quadrática, as  $p$  seguintes restrições dizem respeito às transformações das restrições não-lineares quadráticas de (4.32) e as  $q$  últimas restrições dizem respeito às restrições lineares de (4.32). Este problema, (4.33), pode ser eficientemente resolvido com qualquer programa para gerar soluções de LMIs baseado em métodos de pontos interiores. Usaremos, assim como no caso da construção da aproximação quadrática, o

SeDuMi para resolver este problema.

Observe que a solução do problema (4.33) também é solução para o problema quadrático associado (4.32) e está portanto, relacionada à solução para o problema original (4.21). Esta solução produz um indivíduo localmente melhor que deverá ser introduzido na população do AG.

A figura 4.13 mostra um esquema do operador de busca local proposto nessa seção.

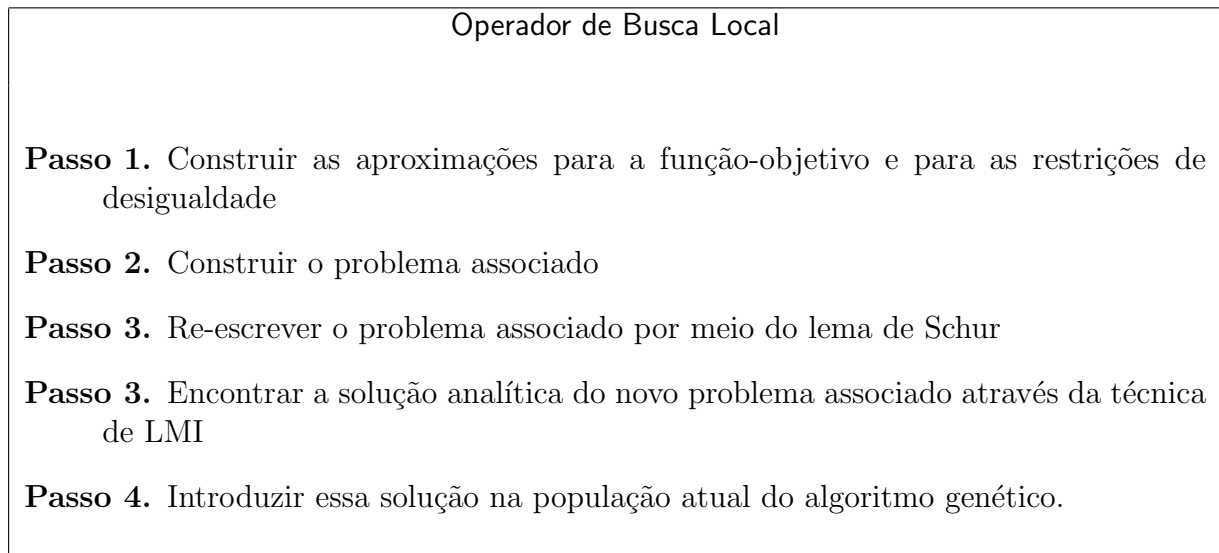


Figura 4.13: Esquema básico de um operador de busca local para problemas com múltiplas restrições de desigualdades.

### 4.3.3 Hibridizando o AG com o Operador de Busca Local

Sabe-se que o sucesso de um algoritmo híbrido é devido à interação entre as habilidades de exploração global inerentes ao AG e as habilidades de exploração local proporcionadas pela busca local. O preço a ser pago é um elevado número extra de avaliações de função e frequentemente uma diminuição da diversidade da população. O custo requerido pela busca local é um tópico importante nos algoritmos híbridos. Este tópico torna-se ainda mais importante em problemas de otimização real com funções do tipo *black-box*.

Uma implementação dessas idéias pode ser feita de várias maneiras. Com o objetivo principal de mostrar a relativa melhora da solução que o operador proporciona, usamos uma série de definições arbitrárias que determinam como o operador será acoplado ao AGRP. Para hibridizar o operador com o algoritmo base, estabelecemos as seguintes definições:

- o operador de busca local será executado a cada cinco gerações. <sup>5</sup>

<sup>5</sup>Experimentamos uma variação desse número, mas nenhuma variação considerável nos resultados foi verificada.

- apenas os pontos no interior de uma vizinhança do melhor ponto serão utilizados na construção das aproximações quadráticas. Esta vizinhança é uma elipse cujos eixos correspondem a 10% da variação de cada parâmetro. Se o número de pontos nessa vizinhança for menor o número de pontos necessários para se construir a aproximação, o operador de busca local não será executado.
- finalmente, o ponto de retorno do operador irá deterministicamente substituir o pior ponto da população atual.

A figura 4.14 apresenta um esquema de um algoritmo genético híbrido especializado no tratamento de restrições de desigualdade.

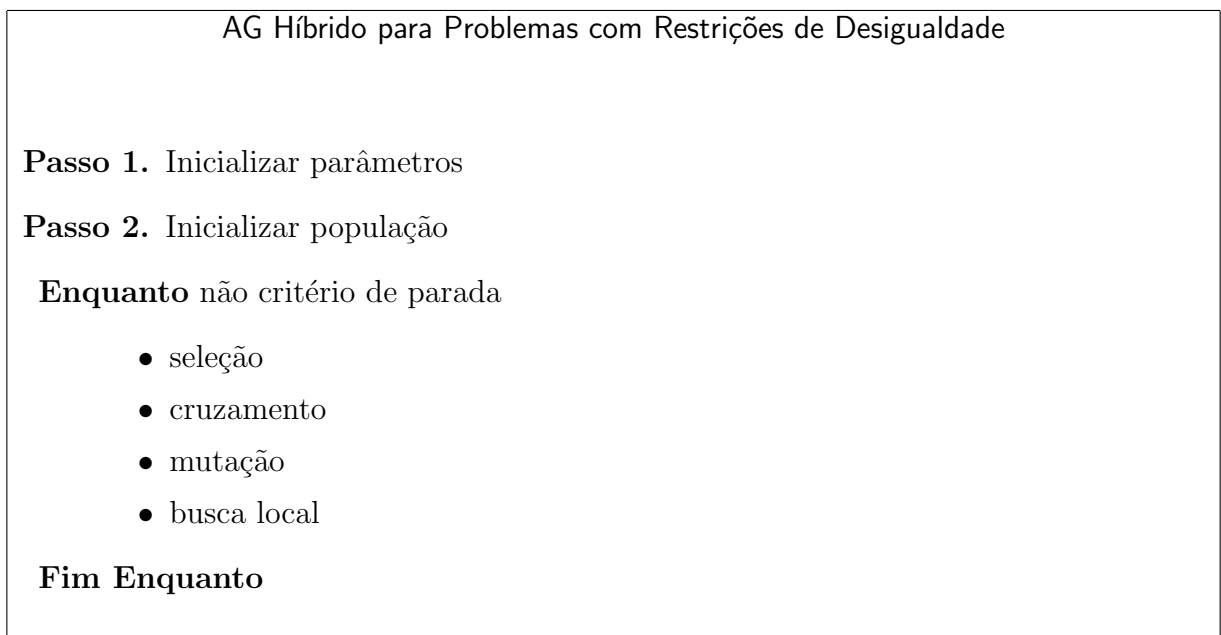


Figura 4.14: Esquema básico de um algoritmo genético híbrido para tratamento de restrições de desigualdade.

#### 4.3.4 Testes e Resultados

Para avaliar o desempenho do algoritmo híbrido proposto, um conjunto de funções de teste foi utilizado. As funções estão descritas a seguir:

**T1** Consideramos o problema bi-dimensional e quadrático definido por:

$$\begin{aligned} \min f(x) &= (x_1 - 4)^2 + (x_2 - 6)^2 \\ \text{sujeito a: } &\begin{cases} g_1(x) = 2(x_1 - 2)^2 + (x_2 - 2)^2 - 1 \leq 0 \\ g_2(x) = 2(x_1 - 1)^2 + 4(x_2 - 2)^2 - 1 \leq 0 \\ -4 \leq x_i \leq 4, \quad i = 1, 2 \end{cases} \end{aligned} \quad (4.34)$$

A solução ótima para esse problema é:

$$x^* = [ 1.4932 \quad 2.3583 ]^T,$$

forneendo  $f^* = 19.5440$ . Na solução, a restrição  $g_2$  está ativa.

**T2** Consideramos o problema bi-dimensional e não-quadrático definido por:

$$\begin{aligned} \min f(x) &= (x_1 - 2)^4 + (x_1 - 2x_2)^2 \\ \text{sujeito a: } &\begin{cases} g_1(x) = (x_1)^2 + x_2 \leq 0 \\ g_2(x) = (x_1 - 2)^2 + (x_2 - 1)^2 - 1 \leq 0 \\ -4 \leq x_i \leq 4 \quad i = 1, 2 \end{cases} \end{aligned} \quad (4.35)$$

A solução ótima para esse problema é:

$$x^* = [ 1 \quad 1 ]^T,$$

a qual fornece  $f^* = 2$ . Na solução, a restrição  $g_2$  está ativa.

**T3** Consideramos o problema quadrático, com 15 variáveis e 12 restrições. A dimensão elevada do espaço de variáveis constitui um problema difícil para qualquer algoritmo de otimização. O problema é definido por:

$$\begin{aligned} \min f(x) &= (x - v)^T(x - v) \\ \text{sujeito a: } &\begin{cases} g_i(x) = (x - c_i)^T \cdot Q_i \cdot (x - c_i) - 1 \leq 0 \\ -4 \leq x_i \leq 4 \quad i = 1, \dots, 12 \end{cases} \end{aligned} \quad (4.36)$$

sendo

$$\begin{aligned} a &= [a_j], \quad a_j = 1 \quad j = 1, 2, \dots, 15 \\ v &= 5 \cdot \vec{a}, \quad c_1 = \left(1 + \frac{\sqrt{15}}{15}\right) \cdot \vec{a}, \\ c_2 &= \left(1 + \frac{\sqrt{30}}{30}\right) \cdot \vec{a}, \quad c_3 = \left(1 + \frac{\sqrt{15}}{30}\right) \cdot \vec{a}, \\ c_4 &= \left(1 + \frac{\sqrt{15}}{15}\right) \cdot \vec{a}, \quad c_5 = \left(1 + \frac{\sqrt{29}}{29}\right) \cdot \vec{a}, \\ c_6 &= \frac{6}{5} \cdot \vec{a}, \quad c_7 = \left(1 + \frac{\sqrt{6}}{12}\right) \cdot \vec{a}, \\ c_8 &= \left(1 + \frac{\sqrt{23}}{23}\right) \cdot \vec{a}, \quad c_9 = \left(1 + \frac{\sqrt{10}}{20}\right) \cdot \vec{a}, \\ c_{10} &= \left(1 + \frac{\sqrt{35}}{35}\right) \cdot \vec{a}, \quad c_{11} = \left(1 + \frac{\sqrt{10}}{20}\right) \cdot \vec{a}, \\ c_{12} &= \left(1 + \frac{\sqrt{3}}{15}\right) \cdot \vec{a} \end{aligned} \quad (4.37)$$

e considerando  $i = j = 15$ ,

$$\begin{aligned}
 Q_1 = (a_{ij}) &= \begin{cases} a_{ij} = 0, & \text{se } i \neq j \\ a_{ij} = 1, & \text{nos outros casos} \end{cases} \\
 Q_2 = 2.Q_1, \quad Q_3 = 4.Q_1, \quad Q_4 = 3.Q_1 & \\
 Q_5 = (a_{ij}) &= \begin{cases} a_{ij} = 0, & \text{se } i \neq j \\ a_{ij} = 1, & \text{se } i = j \text{ e } i = 1 \\ a_{ij} = 2, & \text{nos outros casos} \end{cases}
 \end{aligned} \tag{4.38}$$

$$\begin{aligned}
 Q_6 = (a_{ij}) &= \begin{cases} a_{ij} = 0, & \text{se } i \neq j \\ a_{ij} = 1, & \text{se } i = j \text{ e } i = 1, \dots, 5 \\ a_{ij} = 2, & \text{nos outros casos} \end{cases} \\
 Q_7 = (a_{ij}) &= \begin{cases} a_{ij} = 0, & \text{se } i \neq j \\ a_{ij} = 1, & \text{se } i = j \text{ e } i = 1, \dots, 6 \\ a_{ij} = 2, & \text{nos outros casos} \end{cases}
 \end{aligned} \tag{4.39}$$

$$\begin{aligned}
 Q_8 = (a_{ij}) &= \begin{cases} a_{ij} = 0, & \text{se } i \neq j \\ a_{ij} = 1, & \text{se } i = j \text{ e } i = 1, \dots, 8 \\ a_{ij} = 2, & \text{nos outros casos} \end{cases} \\
 Q_9 = (a_{ij}) &= \begin{cases} a_{ij} = 0, & \text{se } i \neq j \\ a_{ij} = 4, & \text{se } i = j \text{ e } i = 1, \dots, 5 \\ a_{ij} = 2, & \text{nos outros casos} \end{cases}
 \end{aligned} \tag{4.40}$$

$$\begin{aligned}
 Q_{10} = (a_{ij}) &= \begin{cases} a_{ij} = 0, & \text{se } i \neq j \\ a_{ij} = 3, & \text{se } i = j \text{ e } i = 1, \dots, 5 \\ a_{ij} = 2, & \text{nos outros casos} \end{cases} \\
 Q_{11} = (a_{ij}) &= \begin{cases} a_{ij} = 0, & \text{se } i \neq j \\ a_{ij} = 3, & \text{se } i = j \text{ e } i = 1, \dots, 10 \\ a_{ij} = 2, & \text{nos outros casos} \end{cases} \\
 Q_{12} = 5.Q_1 &
 \end{aligned} \tag{4.41}$$

A solução ótima,  $x^*$ , para esse problema é

$$[1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]^T,$$

a qual fornece  $f^* = 213.0872$ . Na solução, a restrição  $g_{12}$  está ativa.

**T4** Consideramos o problema dado por:

$$\begin{aligned} \min f(x) &= (x_1 - 10)^3 + (x_2 - 20)^3 \\ \text{sujeito a:} & \\ \left\{ \begin{array}{l} g_1(x) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0 \\ g_2(x) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0 \\ 13 \leq x_1 \leq 100, \\ 0 \leq x_2 \leq 100 \end{array} \right. & \quad (4.42) \end{aligned}$$

A solução ótima é:

$$x^* = [ 14.095 \quad 0.84296 ]^T,$$

a qual fornece  $f^* = -6961.81388$ . Na solução todas as restrições estão ativas.

**T5** Consideremos o problema dado por:

$$\begin{aligned} \min f(x) &= (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + \\ &\quad + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - \\ &\quad - 4x_6x_7 - 10x_6 - 8x_7 \\ \text{sujeito a:} & \\ \left\{ \begin{array}{l} g_1(x) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0 \\ g_2(x) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0 \\ g_3(x) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0 \\ g_4(x) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0 \\ -10 \leq x_i \leq 10, \quad i = 1, 2, \dots, 7 \end{array} \right. & \quad (4.43) \end{aligned}$$

Este problema pode ser encontrado em [30, 31]. O melhor valor encontrado até agora (veja [31]) é

$$x^* = \begin{bmatrix} 2.330499 & 1.951372 & -0.4775414 & 4.365723 \\ -0.6244870 & 1.038131 & 1.594227 & \end{bmatrix}^T,$$

a qual fornece  $f^* = 680.6300$ . Na solução, as restrições  $g_1$  e  $g_4$  estão ativas.

**T6** Consideramos o problema apresentado pela primeira vez em [32] e usado em [27] e [33]

para analisar o desempenho de vários AGs em problemas restritos.

$$\min f(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + \\ + 37.293239x_1 - 40792.141$$

sujeito a:

$$\left\{ \begin{array}{l} g_1(x) = -85.334407 - 0.0056858x_2x_5 - \\ \quad - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0 \\ g_2(x) = 85.334407 + 0.0056858x_2x_5 + \\ \quad + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 91 \leq 0 \\ g_3(x) = -80.51249 - 0.0071317x_2x_5 - \\ \quad - 0.0029955x_1x_2 - 0.0021813x_3^2 \leq 0 \\ g_4(x) = 80.51249 + 0.0071317x_2x_5 + \\ \quad + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0 \\ g_5(x) = -9.300961 - 0.0047026x_3x_5 - \\ \quad - 0.0012547x_1x_3 - 0.0019085x_3x_4 \leq 0 \\ g_6(x) = 9.300961 + 0.0047026x_3x_5 + \\ \quad + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0 \\ 78 \leq x_1 \leq 102, \\ 33 \leq x_2 \leq 45, \\ 27 \leq x_i \leq 45, \quad i = 3, 4, 5 \end{array} \right. \quad (4.44)$$

A melhor solução conhecida para esse problema é

$$x^* = [ 78 \quad 33 \quad 29.995 \quad 45 \quad 36.776 ]^T,$$

a qual fornece  $f^* = -30665.5$ . Na solução, as restrições  $g_2$  e  $g_5$  estão ativas.

**T7** Consideramos o problema é definido por:

$$\min f(x) = -\frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1+x_2)} \text{sujeito:} \\ \left\{ \begin{array}{l} g_1(x) = x_1^2 - x_2 + 1 \leq 0 \\ g_2(x) = 1 - x_1 + (x_2 - 4)^2 \leq 0 \\ 0 \leq x_i \leq 10 \quad i = 1, 2 \end{array} \right. \quad (4.45)$$

A melhor solução para esse problema é

$$x^* = [ 1.2279713 \quad 4.2453733 ]^T \quad (4.46)$$

a qual fornece  $f^* = -0.095821$ .



**T8** Consideramos o problema:

$$\begin{aligned} \min f(x) &= \frac{\sum_{i=1}^5 0.01((x_i+0.5)^4 - 30x_i^2 - 20x_i)}{5} \\ \text{sujeito a:} & \\ &\begin{cases} g_1(x) = (x - v)^T(x - v) \leq 0 \\ g_2(x) = (x - c)^T H(x - c) \leq 0 \\ -6 \leq x_i \leq 6 \\ i = 1, \dots, 5 \end{cases} \end{aligned} \quad (4.47)$$

sendo

$$\begin{aligned} v &= [ 6 \ 6 \ 6 \ 6 \ 6 ]^T \\ c &= [ 5 \ 6 \ 6 \ 6 \ 6 ]^T \\ H &= \begin{pmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

A função-objetivo é uma função multimodal com mínimo global em

$$x^* = [ -4.4538 \ \dots \ -4.4538 ]$$

e outros mínimos locais situados nos vértices do hiper-quadrado  $X = [ \pm 4.4538 \ \dots \ \pm 4.4538 ]$ . A região factível está localizada na região de um dos mínimos locais da função-objetivo.

**T9** Consideramos o problema

$$\begin{aligned} \min f(x) &= x^T \cdot A^T \cdot A \cdot x - 10 \cdot [1 \ 1] \cdot \cos(2\pi Ax) \\ \text{sujeito a:} & \\ &\begin{cases} g_1(x) = (x_1 - 2)^2 + (x_2 - 2)^2 - 1 \leq 0 \\ -4 \leq x_i \leq 4 \\ i = 1, 2 \end{cases} \end{aligned} \quad (4.48)$$

$$\text{sendo } A = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$$

Sabe-se que o ponto de ótimo está situado na fronteira da região factível, entretanto, não encontramos nenhum valor ótimo para esse problema na literatura.

É possível estimar o tamanho da região factível de um problema qualquer através da métrica  $\rho$ , sugerida em [34]. A métrica  $\rho$  dada por

$$\rho = \frac{|F|}{|T|}, \quad (4.49)$$

sendo  $|T|$  o número de soluções aleatoriamente geradas e  $|F|$  o número de soluções factíveis nesse total de soluções geradas. Esta métrica nos fornece uma estimativa do tamanho da região factível, e conseqüentemente, nos indica o quão difícil é gerar uma solução factível através de um processo aleatório.

A tabela 4.2 mostra um resumo das funções de testes escolhidas, sendo  $n$  o número de variáveis, LI o número de restrições de desigualdade linear, NLI o número de restrições não-lineares de desigualdades e  $\rho$  o tamanho da região factível. O número máximo de gerações que será usado em cada problema, bem como o tamanho da população também estão listados nessa tabela.

Tabela 4.2: Características de cada problema teste

Problema	Tipo de função-objetivo	$n$	LI	NLI	$\rho$	MaxGen	População
T1	quadrática	2	0	2	0.00045	50	50
T2	não-linear	2	0	2	0.00010	50	50
T3	quadrática	15	0	12	0	100	160
T4	não-linear	2	0	2	0.00001	50	50
T5	não-linear	7	0	4	0.00058	100	100
T6	quadrática	5	0	6	0.02765	100	80
T7	não-linear	10	3	5	0.00086	50	50
T8	não-linear	5	0	2	0	100	100
T9	não-linear	2	0	1	0.00510	100	100

Com objetivo de comparação, os dois algoritmos AGRP e o AGP híbrido foram testados 30 vezes com cada problema. As tabelas 4.3 e 4.4 mostram o desempenho do AG simples e híbrido em cada problema. O símbolo (–) significa que o algoritmo não foi capaz de encontrar nenhuma solução factível. Todas as soluções usadas nesse teste estatístico eram soluções factíveis. Podemos ver que o AG híbrido foi capaz de encontrar o ótimo exato em cinco problemas (T1), (T2), (T3), (T7) e (T8), e, nos outros problemas, obteve solução melhores se comparadas com as soluções obtidas pelo AG simples.

Baseado nos resultados obtidos, decidimos compará-los com os resultados apresentados em [35]. Neste trabalho, os autores utilizam um conjunto de funções de teste, entretanto só utilizamos aquelas funções cuja função-objetivo fosse não linear e que apresentassem restrições não-lineares de desigualdade. As tabelas 4.5 e 4.6 mostram um sumário dos melhores resultados obtidos nesse trabalho e compara-os com os resultados encontrados em

Tabela 4.3: Resultados estatísticos obtidos pelo AG híbrido em cada problema

Problema	Valor ótimo	Melhor valor	Valor médio	Pior valor
T1	19.5440	19.5440	19.5440	19.5440
T2	2	2	2.0073	2.0313
T3	213.0872	213.0872	213.0872	213.0872
T4	-6961.8139	-6811.8	-65339.9	-59876,4
T5	680.6300	681.4534	681.6135	682.0012
T6	-30665.5	-	-	-
T7	-0.095821	-0.0958	-0.0951	-0.0899
T8	3.0615	3.0615	3.0615	3.0615
T9	-0.0187	-0.0187	-0.0147	0.0223

Tabela 4.4: Resultados estatísticos obtidos pelo AG simples em cada problema

Problema	Valor ótimo	Melhor valor	Valor médio	Pior valor
T1	19.5440	19.6010	19.7101	20.2102
T2	2	2.0053	2.2025	2.7126
T3	213.0872	243.9816	245.1588	251.6680
T4	-6961.8139	-6298.6	-6.398.6	-5503.6
T5	680.632	691.4714	687.8759	699.2311
T6	-30665.5	-	-	-
T7	-0.095821	-0.09057	-0.0872	-0.0469
T8	3.0615	3.0862	4.0590	9.7844
T9	-0.0187	-0.0186	0.0088	0.1324

Tabela 4.5: Comparação: Resultados obtidos na Literatura.

Prob.	$x^*$	Literatura [35]		
		Melhor	Média	Pior
T1	19.5440	#	#	#
T2	2	#	#	#
T3	213.0872	#	#	#
T4	-6961.8139	-6961.814	-6961.284	-6952.482
T5	680.632	680.632	680.634	680.719
T6	-30665.5	-30655.539	-30655.539	-30655.539
T7	-0.095821	-0.095825	-0.095825	-0.095825
T8	3.0615	#	#	#
T9	-0.0187	#	#	#

[35]. Convém ressaltar que, em [35], os autores usaram 800 como o número máximo de gerações e neste trabalho, este número foi limitado a 100. O símbolo (#) indica que não foi encontrado nenhum valor reportado na literatura para o respectivo problema. O símbolo (-) indica que o algoritmo não foi capaz de encontrar nenhuma solução factível. Como podemos observar, a metodologia descrita nesse capítulo mostra uma performance competitiva em

Tabela 4.6: Comparação: Resultados obtidos nesse trabalho

Prob.	$x^*$	Neste trabalho		
		Melhor	Média	Pior
T1	19.5440	19.5440	19.5440	19.5440
T2	2	2	2	2
T3	213.0872	213.0872	213.0872	213.0872
T4	-6961.8139	-6811.8	-65339.9	-59876,4
T5	680.632	681.4534	681.6135	682.0012
T6	-30665.5	-	-	-
T7	-0.095821	-0.0958	-0.0951	-0.0899
T8	3.0615	3.0615	3.0615	3.0615
T9	-0.0187	-0.0187	-0.0147	0.0223

relação ao AG simples e em relação ao trabalho apresentado em [35].

O máximo número de gerações foi o único critério de parada em todos os algoritmos. Ao final das execuções do AG simples e do AG híbrido, a curva média de convergência foi obtida. As figuras 4.15, 4.16, 4.17, 4.18, 4.19, 4.20, 4.21 e 4.22 mostram a curva de convergência para cada problema. O eixo das abscissas representa a geração e o eixo das ordenadas representa o logaritmo na base 10 do valor da função-objetivo do melhor indivíduo ao longo das gerações. O logaritmo na base 10 foi usado apenas para realçar a diferença entre as curvas. Observe que em todos os casos o algoritmo híbrido obteve um ponto melhor ou igual ao ponto obtido pelo AG simples. Nem sempre a velocidade de convergência do AG híbrido é mais rápida do que a do AG simples, porém na maioria dos casos esse fato pode ser observado. Podemos observar nas figuras 4.16, 4.17e 4.18 um comportamento não-monotônico das curvas de convergência. A explicação para esse comportamento se deve ao fato de que a curva de convergência representa o valor da função-objetivo do melhor indivíduo e não o valor da função-objetivo penalizada. Desta forma, o melhor indivíduo de uma geração que apresenta um valor baixo para função-objetivo pode, no entanto, ser superado na próxima geração por um indivíduo com um valor mais alto para a função-objetivo porém que satisfaz mais adequadamente às restrições.

### 4.3.5 Conclusões

O algoritmo AG Híbrido apresentou uma boa performance em problemas com múltiplas restrições de desigualdade. O uso das aproximações quadráticas para ambas as funções, objetivo e de restrição, melhora as propriedades de convergência do AG. Os resultados confirmam que o AG Híbrido converge para uma solução melhor e ainda, diminui o número de gerações necessário para alcançar essa solução.

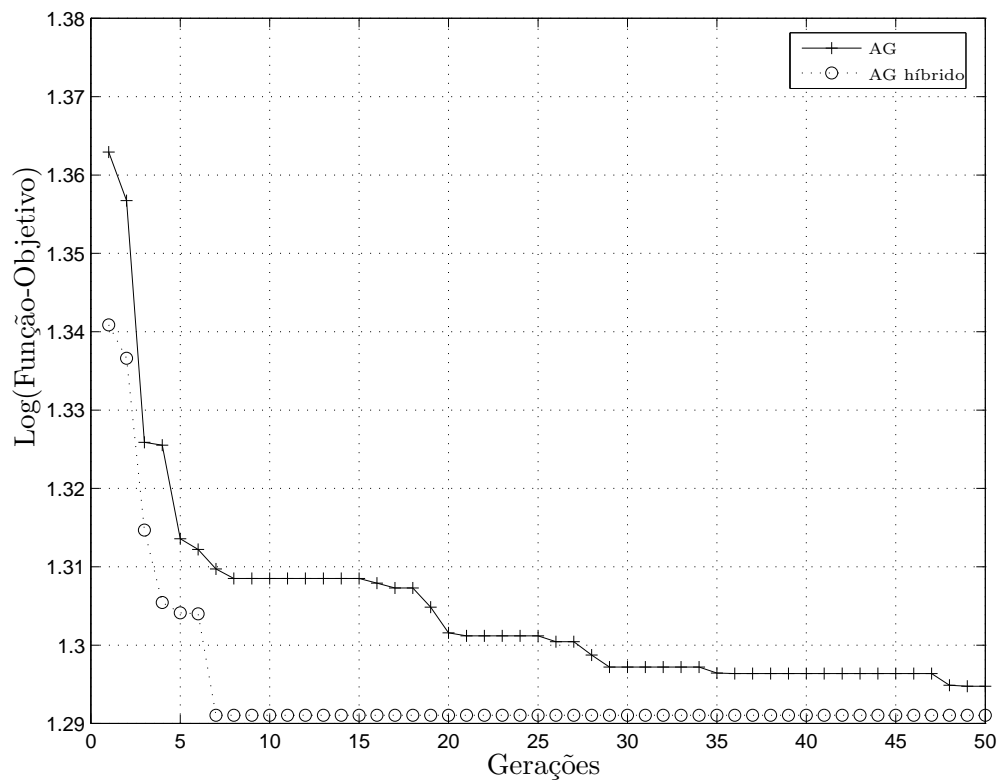


Figura 4.15: Curva de convergência para o problema (T1).

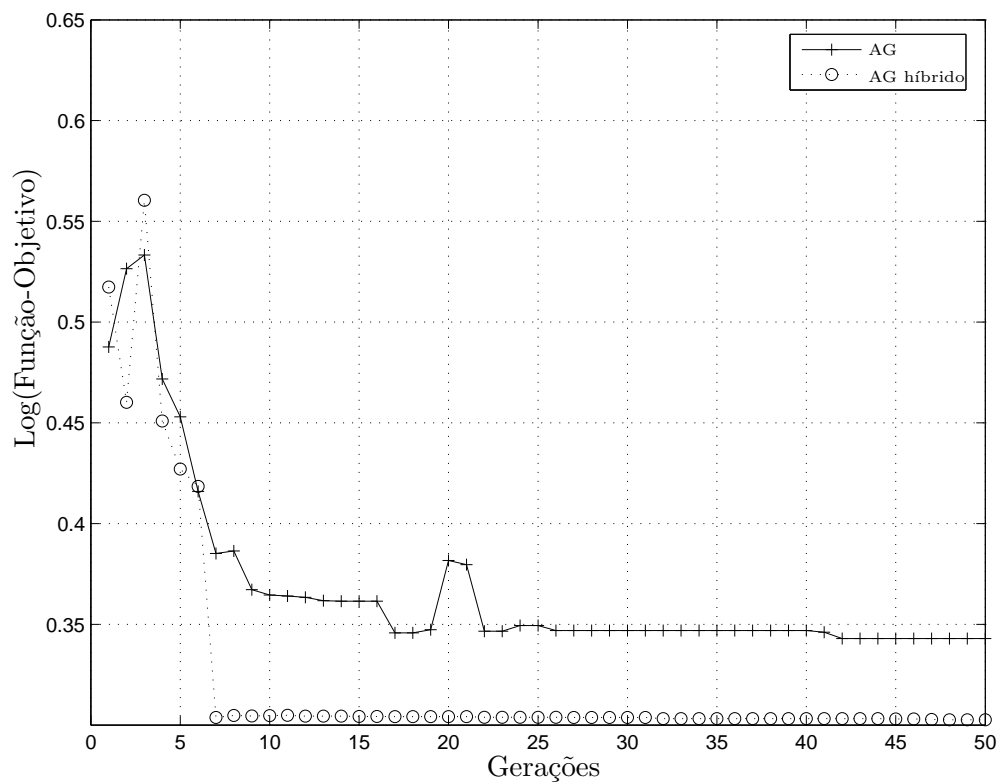


Figura 4.16: Curva de convergência para o problema (T2).

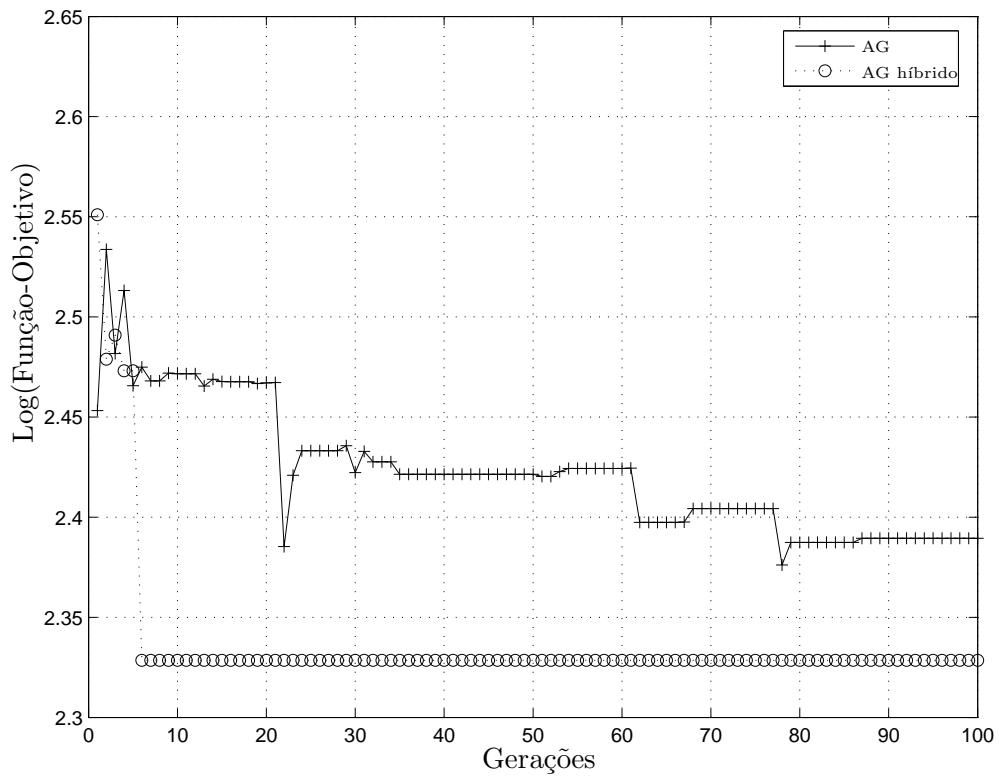


Figura 4.17: Curva de convergência para o problema (T3).

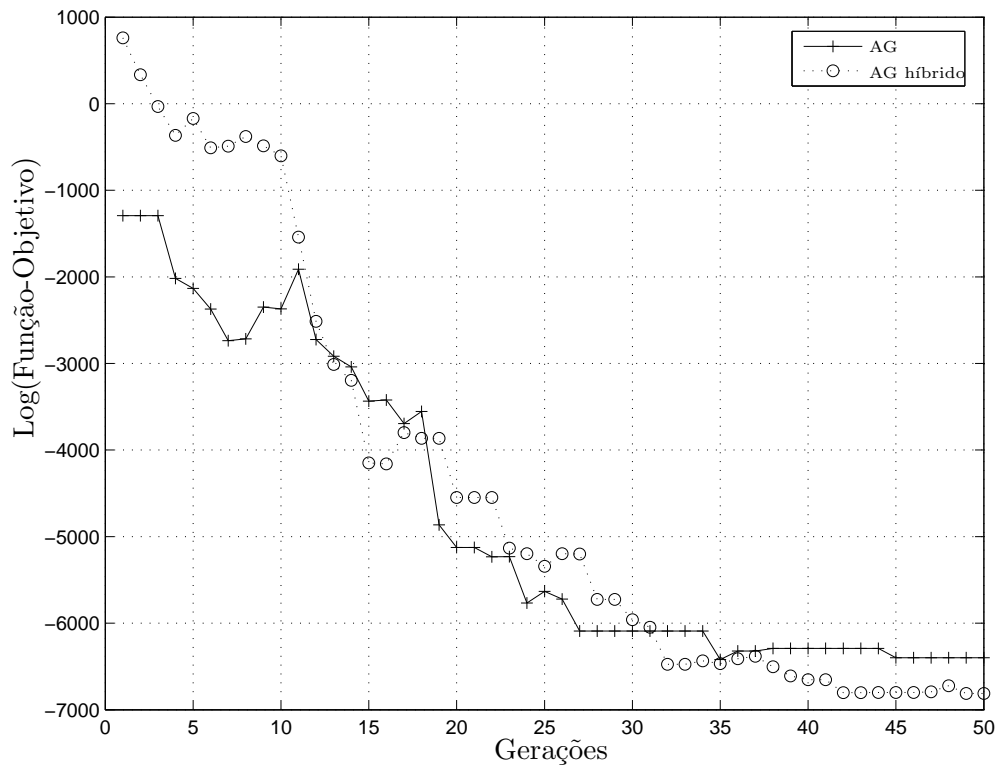


Figura 4.18: Curva de convergência para o problema (T4).

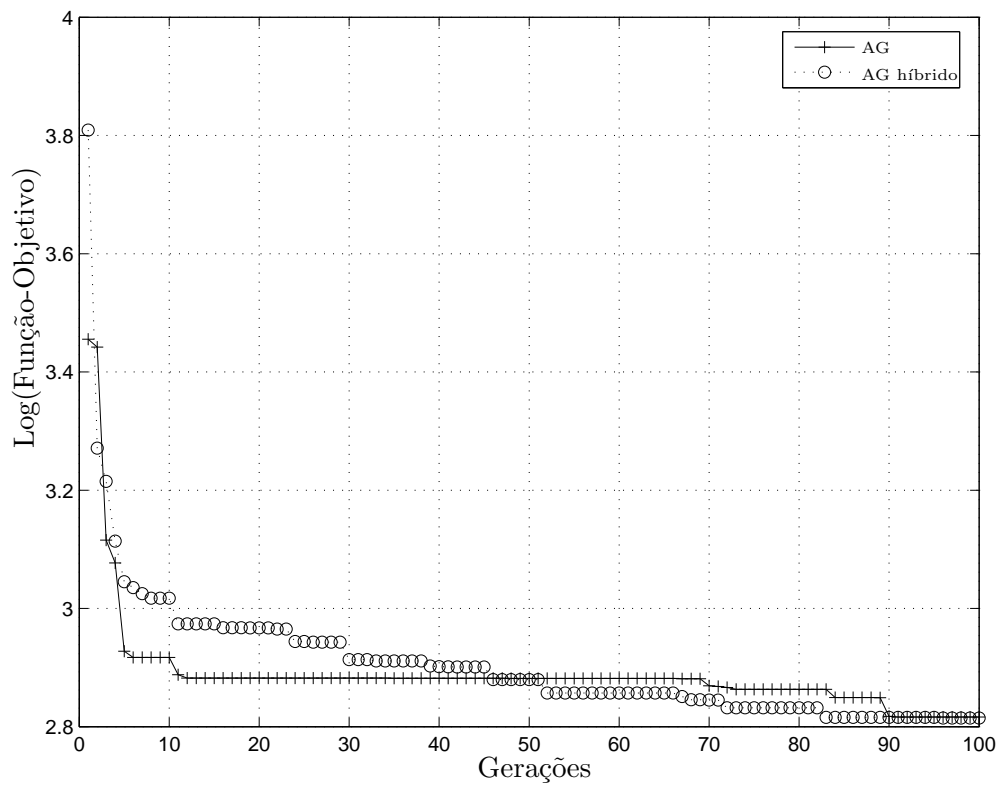


Figura 4.19: Curva de convergência para o problema (T5).

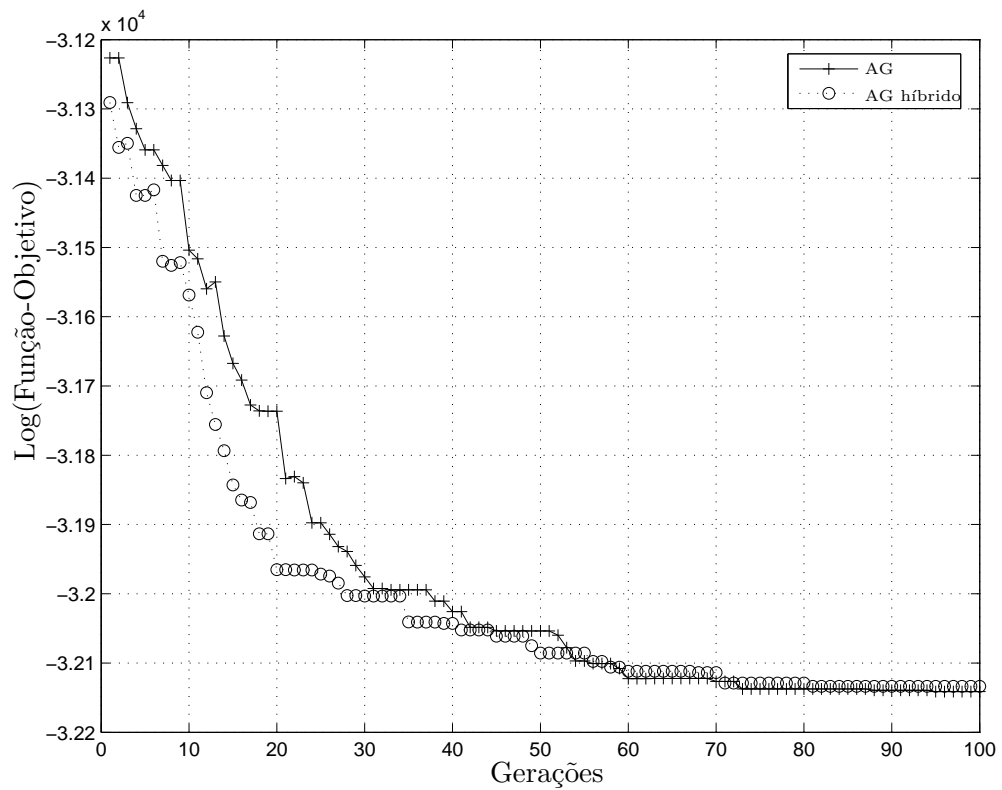


Figura 4.20: Curva de convergência para o problema (T6).

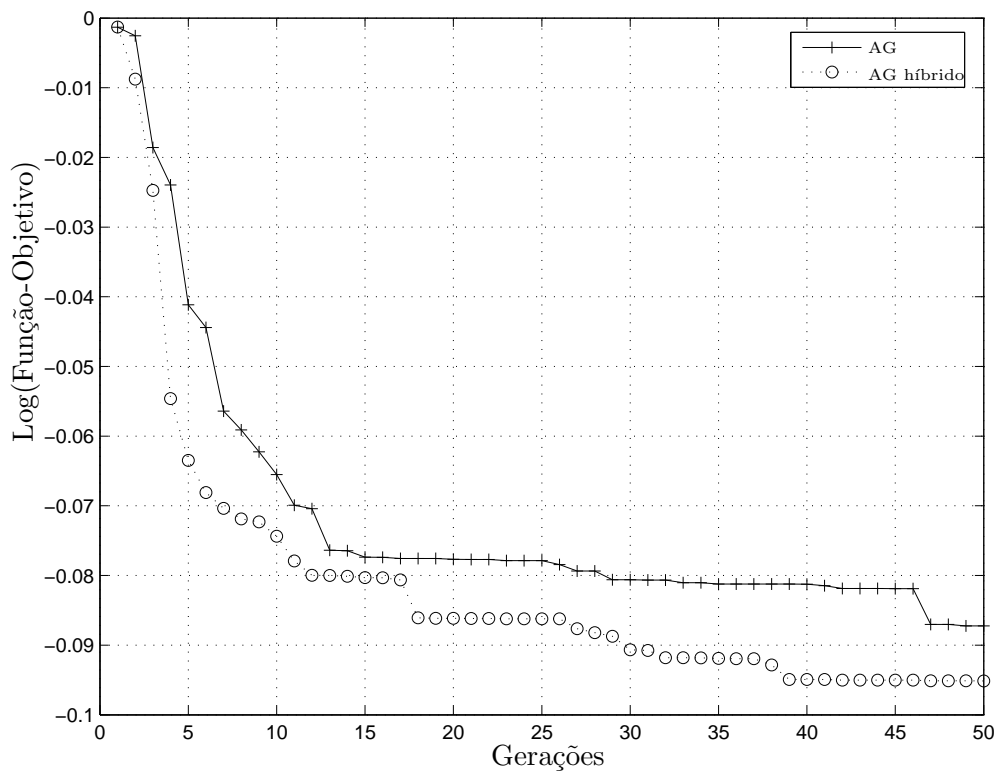


Figura 4.21: Curva de convergência para o problema (T7).

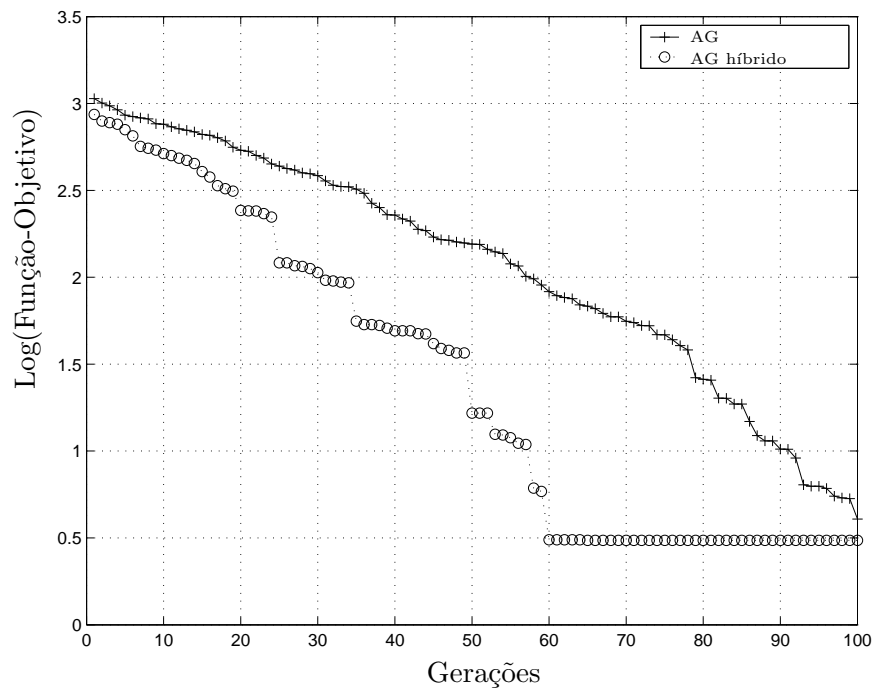


Figura 4.22: Curva de convergência para o problema (T8).



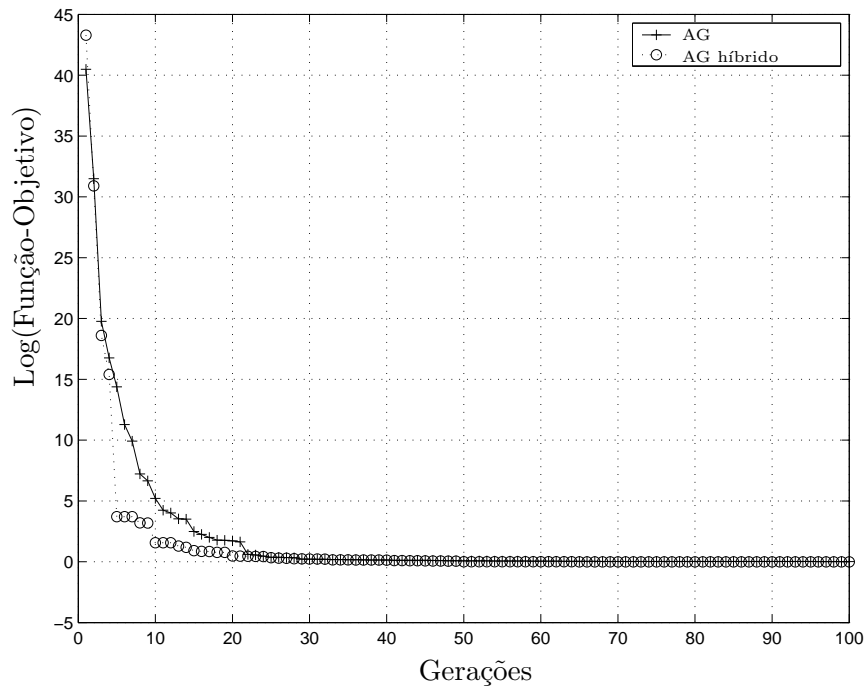


Figura 4.23: Curva de convergência para o problema (T9).

Em alguns problemas o algoritmo híbrido não foi capaz de obter uma solução factível. Novos testes, incluindo uma modificação no algoritmo, estão sendo feitos com o objetivo de melhorar o desempenho do algoritmo. Essa modificação irá construir uma aproximação linear, no lugar de aproximações quadráticas, para algumas restrições. Se observarmos as duas primeiras restrições do problema (T6), é possível supor que uma restrição seja bem aproximadamente convexa, admitindo portanto uma aproximação quadrática convexa. Entretanto, a outra restrição aparenta ser côncava, não sendo portanto bem aproximada por uma função quadrática convexa. Nesse caso, é possível conjecturar que uma aproximação linear seja mais razoável. Esperamos que essa modificação permita que o algoritmo híbrido seja capaz de encontrar soluções factíveis.

Esse operador de busca local proposto nessa seção não requer nenhum cálculo adicional de avaliação de função por iteração do algoritmo, o que permite que a inclusão desse operador seja recomendada em problemas com múltiplas restrições de desigualdade, principalmente no caso de problemas cujo cálculo da função-objetivo seja computacionalmente caro.

## 4.4 Mudança de Coordenada dentro do Algoritmo Genético

Nas duas seções anteriores, vimos exemplos de utilização das aproximações quadráticas em problemas mono-objetivo com restrições. Em ambos os casos, a aproximação quadrática foi empregada usando-se a idéia (1) descrita no capítulo 2: as aproximações quadráticas foram utilizadas para gerar uma estimativa para o mínimo da função.

Nesta seção apresentaremos uma forma de utilização de aproximações quadráticas nos AGs que segue a idéia (2), também descrita no capítulo 2: as aproximações quadráticas podem ser utilizadas para efetuar uma mudança de coordenadas no espaço das variáveis. As aproximações quadráticas serão utilizadas para efetuar uma correção no sistema de coordenadas dentro das iterações dos algoritmos genéticos <sup>6</sup>.

Essa metodologia será empregada na resolução de problemas mono-objetivos irrestritos. Da mesma maneira que nos dois capítulos anteriores, uma aproximação quadrática e convexa é construída, usando as avaliações da função-objetivo disponíveis pelo algoritmo e a matriz Hessiana desta aproximação será utilizada para mudar as coordenadas do espaço de variáveis do problema. Essa mudança de coordenadas levará a novas coordenadas nas quais as novas superfícies de nível sejam circulares. Após a mudança de coordenadas, o algoritmo continua nas novas coordenadas e após algumas iterações, uma nova aproximação quadrática é construída e uma nova mudança de coordenadas é efetuada.

A motivação para essa metodologia reside na observação de que, em presença das novas coordenadas, as operações do GA tornam-se mais eficientes na obtenção da solução do que na presença das coordenadas originais. A razão desta observação pode ser entendida com o auxílio das figuras 4.24 e 4.25.

As figuras 4.24 e 4.25 representam um ponto no espaço original e no transformado, e mostram que uma perturbação de tamanho similar no referido ponto em ambos os espaços apresenta uma maior probabilidade de melhoria no valor da função-objetivo no caso do espaço transformado. Além disso, é possível ver que, no caso das coordenadas originais, é esperado que o ponto se mova (após algumas perturbações e seleções estocásticas) na direção do eixo maior da elipse. Já no espaço transformado, o movimento esperado é na direção do centro da elipse, uma analogia estocástica do “Efeito de Marato”. Isto explica porque espera-se dos operadores genéticos de recombinação e de mutação um comportamento melhor no espaço de coordenadas transformadas.

---

<sup>6</sup>Os resultados que serão apresentados nessa seção encontram-se publicadas em [36].

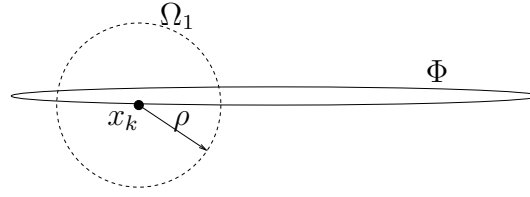


Figura 4.24: Esta figura mostra um ponto  $x_k$  no espaço de coordenadas original. O ponto  $x_k$  está situado na curva de nível (linha sólida) de uma função-objetivo quadrática e convexa, que possui a forma de uma elipse alongada  $\Phi$ .  $x_k$  é também o centro de um círculo  $\Omega_1$  de raio  $\rho$ . Se  $x_k$  está sujeito a uma perturbação de comprimento menor ou igual a  $\rho$ , este ponto pode cair dentro de  $\Omega_1$ . Se o ponto cai dentro de  $\Omega_1 \cap \Phi$ , o ponto perturbado terá um valor melhor para a função-objetivo.

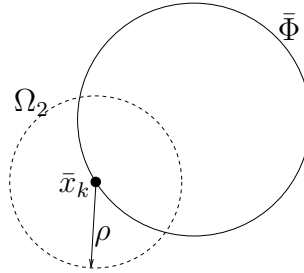


Figura 4.25: Esta figura mostra um ponto  $\bar{x}_k$ , que corresponde a  $x_k$ , nas coordenadas após a mudança. O ponto  $\bar{x}_k$  pertence à curva de nível (linha sólida) da função-objetivo, que é agora um círculo  $\bar{\Phi}$ .  $\bar{x}_k$  está agora no centro de um círculo  $\Omega_2$  de raio  $\rho$ . Se  $\bar{x}_k$  está sujeito a uma perturbação menor ou igual a  $\rho$ , este ponto pode cair sobre algum ponto dentro de  $\Omega_2$ . Se o ponto cai dentro de  $\Omega_2 \cap \bar{\Phi}$ , o ponto perturbado terá um melhor valor de função-objetivo. Observe que o volume do conjunto  $\Omega_2 \cap \bar{\Phi}$  é maior que o volume do conjunto  $\Omega_1 \cap \Phi$ , o que significa que a probabilidade de melhora do valor da função-objetivo após uma perturbação em  $\bar{x}_k$  é maior que a probabilidade de melhora para  $x_k$  com uma perturbação similar.

#### 4.4.1 Metodologia

Considere o problema não-linear irrestrito da forma:

$$x^* = \min_x f(x) \quad (4.50)$$

sendo  $f(\cdot)$  uma função real e não-linear.

Usando a metodologia descrita na seção 2.3 a função  $f(\cdot)$  pode ser aproximada por uma função da forma

$$h(z) = (z - z_f)^T H (z - z_f) - c \quad (4.51)$$

com

$$c = 0.25r^T H^{-1}r + \gamma.$$

( $r$  e  $\gamma$  definidos na seção 2.3 ) sendo  $H$ , por construção, uma matriz positiva definida.

Observe que esta aproximação é *semi-global*, uma vez que esta aproximação utiliza pontos na região total que está sendo coberta pela população do AG.

A metodologia de mudança de coordenadas pretende definir novas coordenadas para o problema de forma que as superfícies de nível da aproximação quadrática se tornem esféricas. Considere a fatorização de Cholesky da matriz  $H$

$$H = L^T L$$

na qual a matriz  $L$  é não-singular. Então, usando a mudança de coordenadas

$$\tilde{z} = Lz, \tag{4.52}$$

a função-objetivo associada pode ser escrita da seguinte forma:

$$\tilde{h}(\tilde{z}) = (\tilde{z} - \tilde{z}_f)^T (\tilde{z} - \tilde{z}_f) + \gamma, \tag{4.53}$$

sendo  $\tilde{z}_f$  o ponto de mínimo irrestrito da função-objetivo aproximada:

$$\tilde{z}_f = -L \left( \frac{1}{2} H^{-1} r \right) = -\frac{1}{2} (L^T)^{-1} r. \tag{4.54}$$

O procedimento de aproximação por uma função quadrática, nesse caso, tem um significado semi-global, uma vez que não se espera que a transformação de coordenada tenha um significado para a função-objetivo, dentro de todo espaço de variável, e nem que tenha significado apenas em uma pequena vizinhança arbitrária do melhor ponto da geração. De fato, como espera-se que a mudança de coordenada melhore o movimento da população em direção ao mínimo da função, parece razoável construir a aproximação utilizando pontos em uma região nem muito grande, nem muito pequena. Essa região deve ser construída ao redor do melhor ponto da geração atual.

Seja  $x_0$  o melhor indivíduo da geração atual. O conjunto de todos os pontos do processo evolucionário é guardado. A aproximação quadrática para a função-objetivo é gerada usando apenas os pontos no interior da vizinhança de  $x_0$ , definida aqui como:

$$\mathcal{N}(x_0) = \{x : (x - x_0)^T R(x - x_0) \leq 1\} \tag{4.55}$$

e

$$R_{ij} = \begin{cases} [0.1(u_i - l_i)]^{-1}, & i = j \\ 0, & i \neq j \end{cases} \tag{4.56}$$

sendo  $l_i$  e  $u_i$ , respectivamente, os valores mínimos e máximos para a  $i$ -ésima variável. Dessa forma, a vizinhança  $\mathcal{N}(x_0)$  é uma região elipsoidal cuja dimensão de cada eixo é função da

variação do parâmetro. É importante salientar que, apesar do tamanho da região da qual as amostras estão sendo retiradas ser mantido constante, a aproximação quadrática tornar-se-á mais representativa na região ao redor do ótimo atual, à medida que o algoritmo evolui. Isto ocorre porque o número de pontos, ao redor do ótimo, aumenta (devido à evolução do AG), o que leva a um aumento da importância dessa região quando a aproximação for construída.

Um procedimento de mudança de coordenadas que pode ser usado durante o ciclo básico do AG está apresentado na figura 4.26.

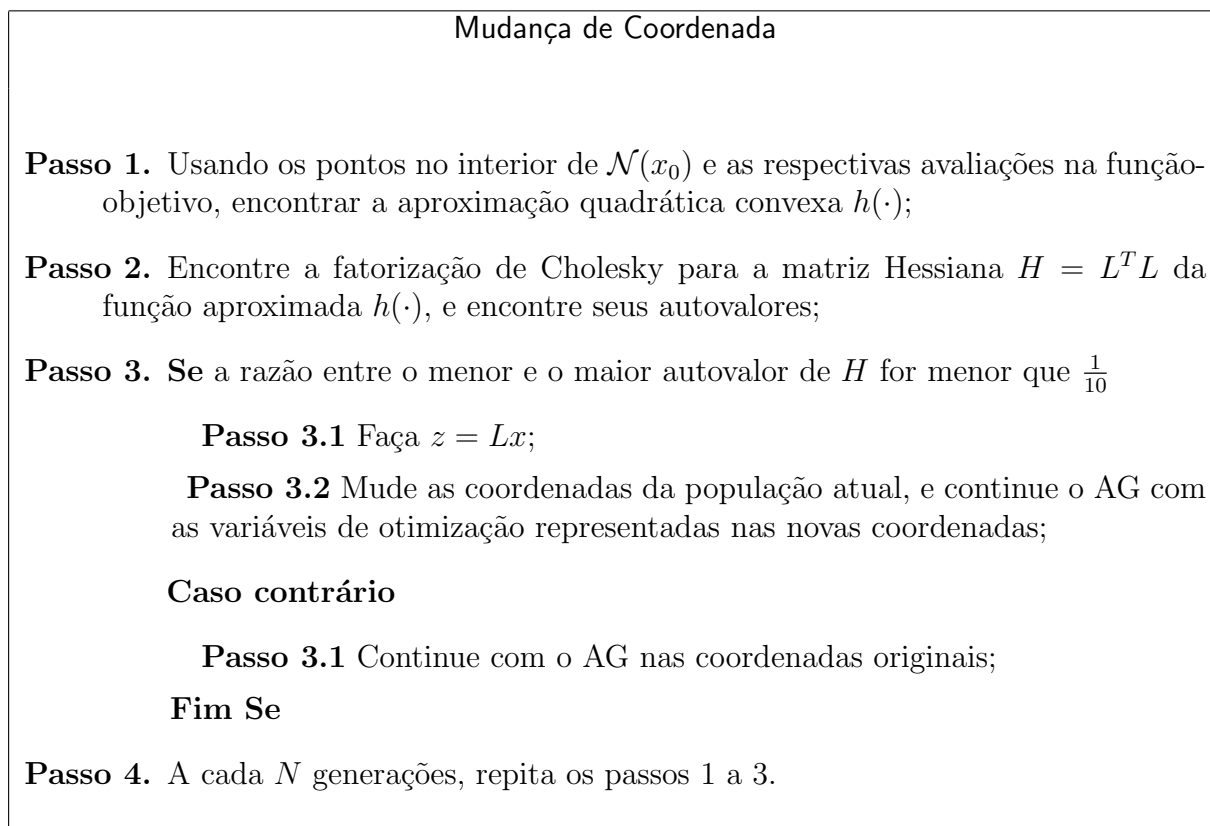


Figura 4.26: Esquema básico do procedimento de mudança de coordenadas.

Seja  $C_N$  o número de condição de uma matriz

$$C_N = \frac{\lambda_s}{\lambda_h}$$

sendo  $\lambda_s$  o menor autovalor e  $\lambda_h$  o maior autovalor dessa matriz. O desempenho de qualquer algoritmo de otimização, incluindo aí os AGs, em um problema quadrático, depende do número de condição da matriz Hessiana da função quadrática: quanto menor o número de condição maior é o número de gerações necessárias para se alcançar o mínimo do problema. Essa situação é ainda pior quando a dimensão do problema aumenta. Esta análise também é válida para problemas não-lineares em geral, uma vez que esses problemas podem ser aproximados por funções quadráticas.

O propósito dos experimentos realizados é determinar o efeito da mudança de coordenada no desempenho do AG simples. A metodologia de mudança de coordenadas foi acoplada ao AG imediatamente após a avaliação da função-objetivo e antes da recombinação. Por simplicidade, no restante desta seção o AG padrão será denotado por *sAG*, e o AG com mudança de coordenadas será denotado por *mcAG*. A construção da aproximação quadrática e a correspondente mudança de coordenada irá ocorrer a cada 10 gerações do *mcAG*. Novamente, como condição matemática, o número de pontos no interior de  $\mathcal{N}(x_0)$  deve ser igual ou maior que

$$\frac{(n+1)(n+2)}{2}$$

sendo  $n$  a dimensão do problema. Se o número de pontos for inferior ao necessário, o procedimento de construção da aproximação não é chamado, e o *mcAG* simplesmente segue sem nenhuma modificação.

#### 4.4.2 Testes e Resultados

O algoritmo proposto *mcAG* e o original *sAG* foram testados 30 vezes em alguns problemas selecionados. Em todos os casos, os dois algoritmos iniciaram com os mesmos parâmetros e mesma população inicial.

Em primeiro lugar, o *mcAG* foi testado com funções quadráticas da forma,

$$\begin{aligned} f(x) &= x^T A x \\ A &= \text{diag} \{ \lambda_1, \lambda_2, \dots, \lambda_n \} \end{aligned}$$

com mínimo na origem, conjunto de autovalores  $\{ \lambda_1, \lambda_2, \dots, \lambda_n \}$  diferentes e dimensão igual a 2, 3 e 4.

Para o problema com dimensão 2, a população do AG foi de 20 indivíduos, para dimensão 3, 30 indivíduos, e para dimensão 4, uma população de 40 indivíduos. Em todos os casos, o espaço de busca foi definido no intervalo  $[-10, 10]$  para todas as variáveis. O critério de parada adotado foi a proximidade com o ótimo analítico menor que  $10^{-3}$ , ou o número máximo de gerações (no caso 100).

A tabela 4.7 mostra, para cada dimensão, o conjunto de autovalores da matriz Hessiana e mostra o número de gerações necessárias para a obtenção do mínimo usando o *mcAG* (NGen2) e mostra também a porcentagem de vezes que o algoritmo foi capaz de obter o ponto ótimo (P2). Para efeito de comparação, o número de gerações necessárias no *sAG* (NGen1) e a porcentagem de vezes que *sAG* foi capaz de encontrar o ótimo (P1) também estão incluídos na mesma tabela. Pode ser observado que, com o uso da mudança de coordenada, o número

Tabela 4.7: Número de gerações necessárias para alcançar o ponto de mínimo (NGen) e a porcentagem de vezes que cada algoritmo foi capaz de atingir a solução (P), para problemas quadráticos de dimensão 2, 3 e 4. Algoritmo 1 refere-se ao *sAG* e algoritmo 2 refere-se ao *mcAG*.

Dimensão	Autovalores	NGen(1)	P(1)	NGen(2)	P(2)
2	1,2	22.5	100%	19.1	100%
2	1,10	24.1	80%	19.6	100%
2	1,100	33.75	56%	19.85	98%
2	1,1000	38.3	32%	19.98	92%
3	1,2,3	33.35	100%	21.2	100%
3	1,10,100	38.45	86%	21.2	97%
3	1,100,1000	45.8	55%	20.9	93%
4	1,2,4,1	41.01	100%	20.45	100%
4	1,10,100,1000	46.6	78%	20.85	94%

de avaliações de função-objetivo é menos sensível ao número de condição da matriz Hessiana do problema.

Os algoritmos *sAG* e *mcAG* foram testados com um conjunto de problemas de *benchmark* encontrados na literatura. Os problemas são:

#### 1. Função de Rastrigin

$$x^* = \arg \min_x x^T A^T A x - 10. [ 1 \ 1 ] \cos(2\pi A x)$$

$$-4 \leq x_i \leq 4$$

$$i = 1, 2$$
(4.57)

sendo

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$$

## 2. Função de Schwefel

$$\begin{aligned}
 x^* &= \arg \min_x 2V + \sum_{i=1}^{10} (-x_i \text{sen}(\sqrt{|x_i|})) \\
 -500 &\leq x_i \leq 500 \\
 i &= 1, \dots, 10
 \end{aligned}
 \tag{4.58}$$

sendo  $V = 418.9829101$

## 3. Função de Rosenbrock

$$\begin{aligned}
 x^* &= \arg \min_x 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \\
 -2.48 &\leq x_i \leq 2.48 \\
 i &= 1, 2
 \end{aligned}
 \tag{4.59}$$

A função de Rastrigin é uma função de estudo bem popular para os AGs devido ao grande espaço de busca e ao grande número de mínimos locais. Ela possui complexidade de  $\mathcal{O}(n \ln(n))$ , sendo  $n$  o número de parâmetros da função. Neste trabalho, usamos apenas  $n = 2$ . A tendência global da função é determinada pelo valor da variável externa  $A$ . O mínimo global está localizado em  $x_i = 0$  para  $i = 1, 2, \dots, n$ , fornecendo 20 para a função-objetivo. A malha de pontos  $x_i = 0$ , exceto em uma coordenada, onde  $x_i = 1$ , fornece 1 para a função-objetivo, o segundo melhor mínimo. Com o aumento da distância do mínimo global, os valores do aptidão dos mínimo locais tornam-se maiores.

A função de Schwefel é relativamente mais simples que a função de Rastrigin, e é caracterizada pelo segundo melhor mínimo encontrar-se distante do ótimo global. Na função de Schwefel,  $V$  é o oposto do mínimo global, valor que é adicionado para que o mínimo se desloque para zero.

A função de Rosenbrock é considerada um caso difícil pois apresenta um vale muito estreito. O topo deste vale é muito afilado, e corre ao redor de uma parábola. Algoritmos que não são capazes de descobrir boas direções não apresentam um bom desempenho nesse problema. O ponto de mínimo está localizado em  $(1, 1)$  e fornece  $f(x) = 0$ .

Vários algoritmos possuem dificuldade de convergência nas proximidades do mínimo dessas funções porque a probabilidade de obter progresso decresce rapidamente à medida que se aproxima do ponto ótimo.

Na primeira parte dos testes com essas funções, dois critérios de parada foram utilizados:



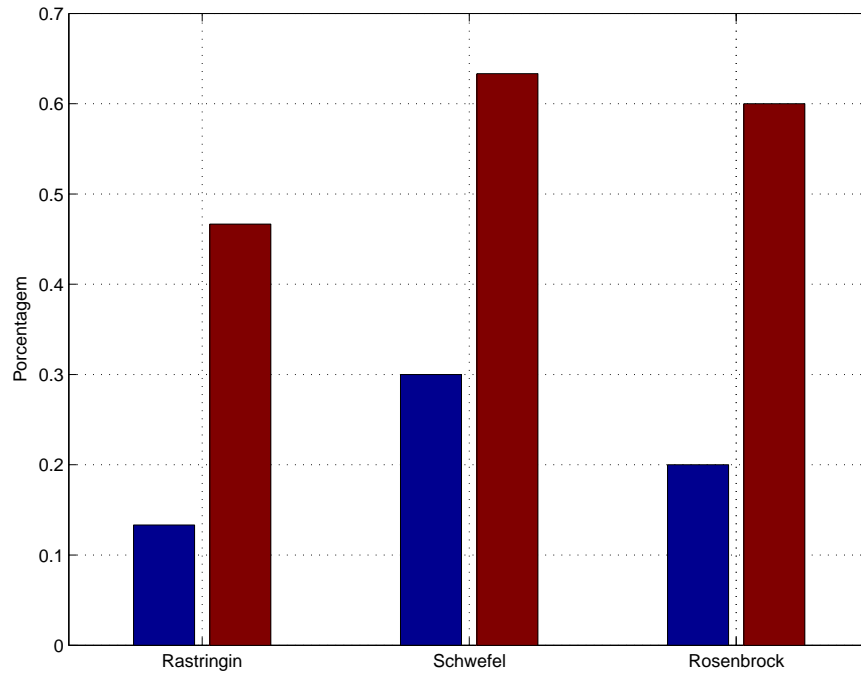


Figura 4.27: Porcentagem de vezes que cada algoritmo alcançou o mínimo em cada problema. A primeira barra representa o *sAG* e a segunda barra representa o *mcGA*.

- máximo número de gerações (100 gerações); ou
- proximidade de 0.001 com o ótimo analítico.

O tamanho da população foi de 20 indivíduos para as funções de Rastrigin e Rosenbrock e de 100 indivíduos para a função de Schwefel.

A figura 4.27 mostra a porcentagem de vezes que cada algoritmo alcançou o ótimo através do critério de proximidade. Podemos observar que o uso da mudança de coordenada melhora a capacidade de convergência quando comparado ao AG padrão.

Usando a função de Rosenbrock, um outro experimento similar foi executado: uma população inicial foi gerada aleatoriamente *em uma região do espaço de busca que não continha o ótimo*  $x^* = [1 \ 1]$ . Todos os outros parâmetros continuaram os mesmos. Dois diferentes intervalos foram escolhidos para se gerar a população inicial:

$$[1] \ 0 \leq x_i \leq -1$$

$$[2] \ -1.8 \leq x_i \leq -2.04$$

A figura 4.28 mostra a porcentagem de vezes que cada algoritmo foi capaz de encontrar o ótimo através do critério de proximidade. Podemos ver que a introdução da mudança de coordenada permite que o algoritmo modificado (*mcAG*) encontre uma boa direção no espaço de busca, diferentemente do AG padrão (*sAG*).

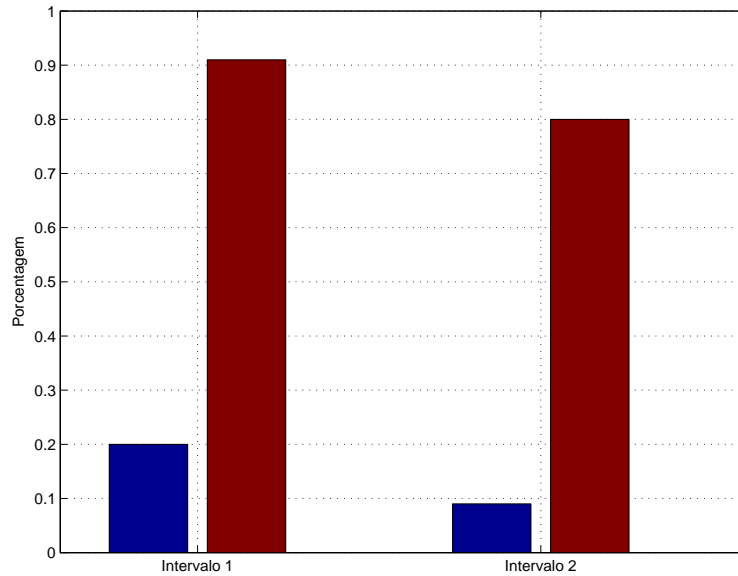


Figura 4.28: Porcentagem de vezes que cada algoritmo alcançou o mínimo na função de Rosenbrock no experimento no qual a população inicial não continha o ponto de mínimo. A primeira barra representa o *sAG* e a segunda barra representa o *mcAG*. Neste caso, cada algoritmo foi executado 200 vezes. Para o primeiro intervalo, a população foi de 20 indivíduos e no segundo intervalo, a população foi de 50 indivíduos.

Na segunda parte dos testes, o número máximo de gerações foi o único critério de parada em ambos os algoritmos. Ao final das 30 execuções do *sAG* e do *mcAG*, a curva média de convergência foi obtida. Esta curva mostra o valor médio do melhor indivíduo ao longo das gerações. Nesses testes, só os experimentos nos quais o algoritmo foi capaz de encontrar o mínimo foram utilizados para gerar a curva média de convergência.

As tabelas 4.8, 4.9 e 4.10 mostram o melhor, o pior e o valor médio obtido pelo *sAG* e pelo *mcAG*, ao final das 30 execuções. O desvio padrão, em cada caso, também está indicado nas tabelas.

Tabela 4.8: Melhor, pior e valor médio obtido pelo *sAG* e *ccGA* durante todas as execuções para a função de Rastrigin. O símbolo  $\sigma$  representa o desvio padrão para esse teste.

Algoritmo	Melhor	Média	Pior	$\sigma$
<i>sAG</i>	-19.9958	-19.0920	-17.6946	0.7357
<i>mcAG</i>	-19.9999	-19.8721	-18.9975	0.4973

As figuras 4.29, 4.30 e 4.31 mostram a curva de convergência para cada problema teste. Apenas os testes nos quais os algoritmos foram capazes de encontrar o ponto de ótimo foram utilizados na construção da curva de convergência. Nos gráficos para as funções de Schwefel

Tabela 4.9: Melhor, pior e valor médio obtido pelo *sAG* e *ccGA* durante todas as execuções para a função de Schwefel. O símbolo  $\sigma$  representa o desvio padrão para esse teste.

Algoritmo	Melhor	Média	Pior	$\sigma$
<i>sAG</i>	0.0123	0.0351	0.3456	0.1042
<i>mcAG</i>	0.0013	0.0114	0.2131	0.0431

Tabela 4.10: Melhor, pior e valor médio obtido pelo *sAG* e *ccGA* durante todas as execuções para a função de Rosenbrock. O símbolo  $\sigma$  representa o desvio padrão para esse teste.

Algoritmo	Melhor	Média	Pior	$\sigma$
<i>sAG</i>	0.0041	0.0655	0.3186	0.1412
<i>ccAG</i>	0.0011	0.0281	0.1356	0.0218

e Rosenbrock, o eixo das abscissas representa as gerações e o eixo das ordenada representa o logaritmo na base 10 do valor da função-objetivo para o melhor indivíduo. No gráfico para função a Rastrigin, o eixo  $y$  representa o próprio valor da função-objetivo do melhor indivíduo. O objetivo do uso do logaritmo na base 10 para o eixo  $y$  foi apenas reforçar a diferença entre as curvas.

Na figura 4.29, observamos que a solução encontrada pelo *mcAG* é melhor que a solução obtida pelo *sAG*. É importante salientar que, neste problema, o *mcAG* encontra primeiramente um mínimo local para o problema em questão, e só depois se desloca para o mínimo global. Observando as figuras 4.30 e 4.31, podemos ver que, em ambos os casos, a velocidade de convergência do *mcAG* é maior do que a velocidade de convergência do *sAG*. Baseado na observação feita para a figura 4.29, podemos conjecturar que o algoritmo *mcAG* também convirja inicialmente para um mínimo local nas funções de Schwefel e de Rosenbrock. Caso estejamos certos, o novo algoritmo torna-se útil no reconhecimento de pontos de mínimos locais.

Finalmente, para examinar a influência do número de variáveis na metodologia aqui proposta, testamos o *sAG* e o *mcAG* com um problema simples de dimensão elevada:

$$x^* = \arg \min_x x^T A x$$

$$-5.12 \leq x_i \leq 5.12 \quad , \quad i = 1, \dots, 30 \quad (4.60)$$

$$A = \text{diag} \{1, 10, 100, 1, 10, 100, \dots, 1, 10, 100\}$$

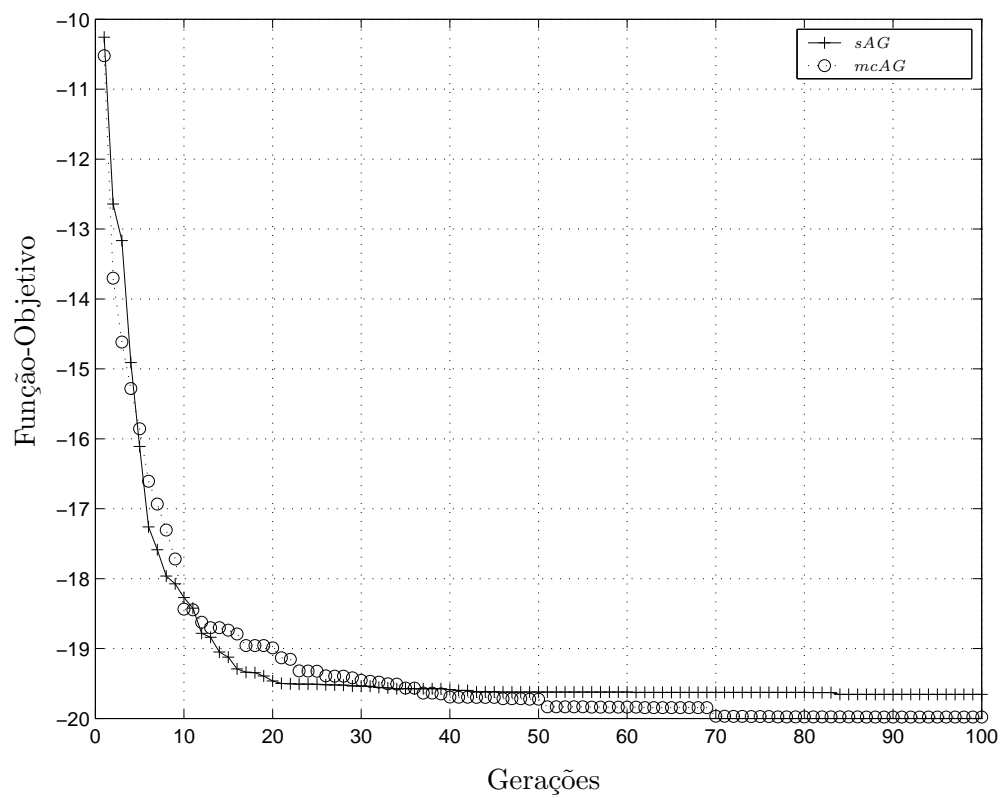


Figura 4.29: Curva de convergência para a função de Rastrigin. Em 30 testes, o *sAG* encontrou o ponto de mínimo 4 vezes e o *mcAG* 14 vezes.

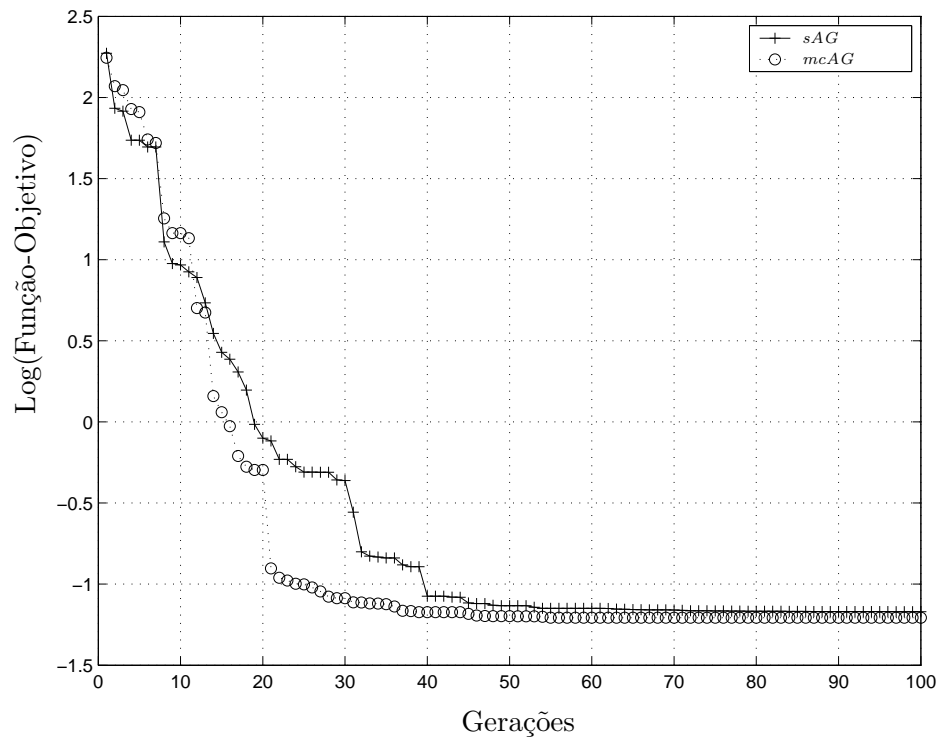


Figura 4.30: Curva de convergência para a função de Schwefel. Em 30 testes, o *sAG* encontrou o ponto de mínimo 9 vezes e o *mcAG* 19 vezes.

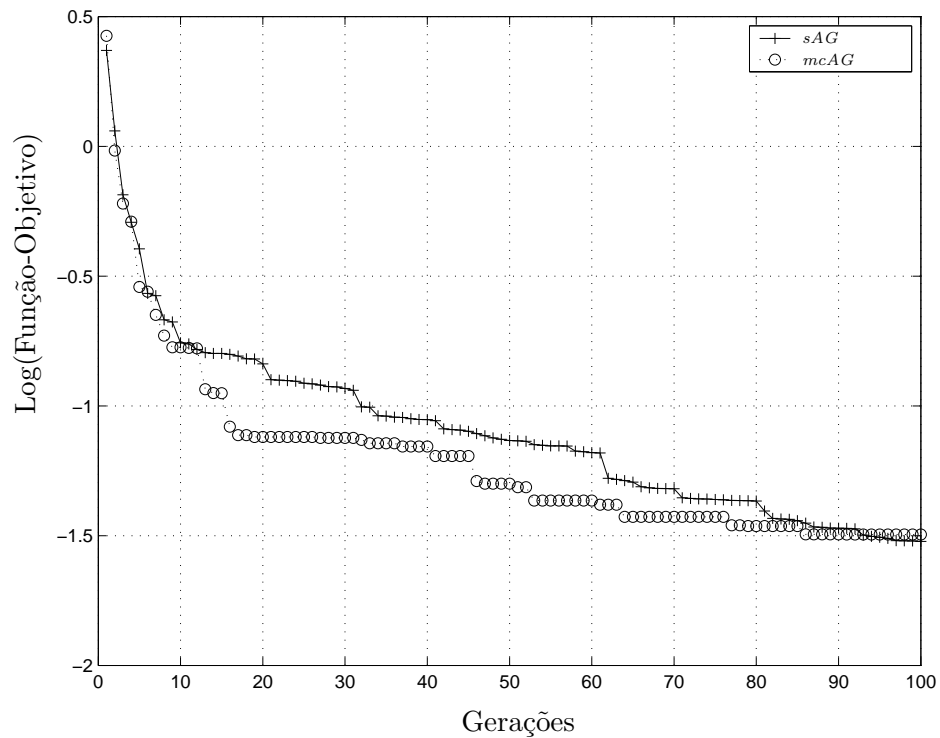


Figura 4.31: Curva de convergência para a função de Rosenbrock. Em 30 testes, o *sAG* encontrou o ponto de mínimo 6 vezes e o *mcAG* 18 vezes.

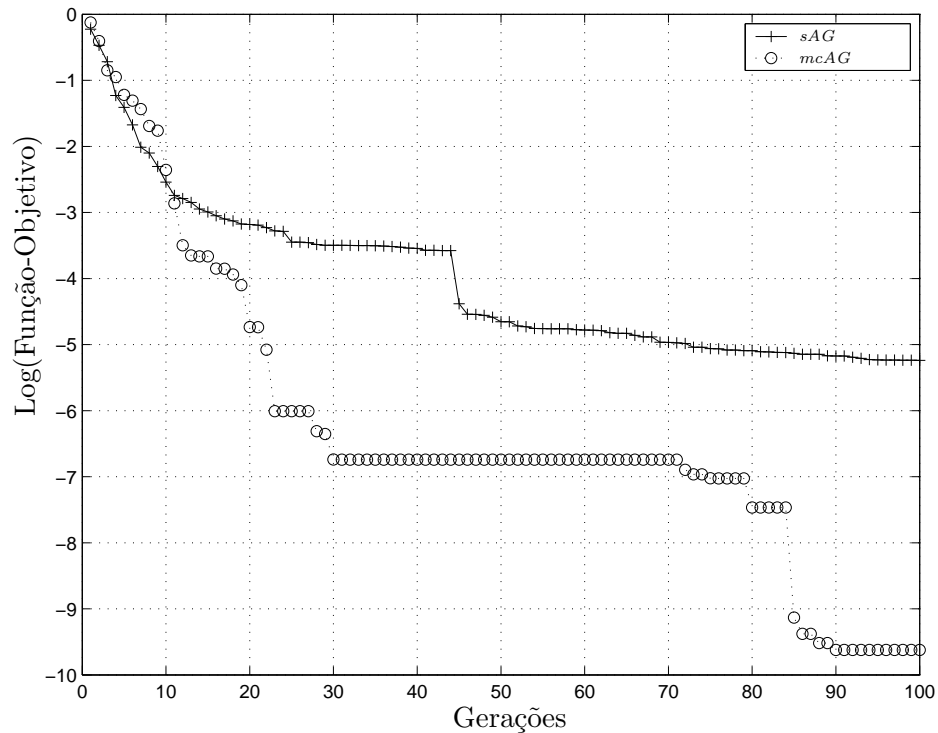


Figura 4.32: Curva de convergência para o problema de dimensão 30 (4.60).

Esta função é suave, unimodal, convexa, simétrica e não possui nenhuma das características discutidas nos testes anteriores. O número elevado de parâmetros resulta em um espaço de busca grande. Nesse caso, uma análise baseada no teorema do *schemata* [37] indicaria que os AGs precisam de uma população muito grande, e portanto, de um elevado número de avaliações de função-objetivo para efetivamente encontrar o mínimo em um problema de dimensão elevada. Tirando o fato da dimensão elevada, este é um problema muito simples.

O *sAG* e o *mcAG* foram testados 30 vezes neste problema. O número máximo de gerações (100) foi o único critério de parada e a população era composta de 120 indivíduos. O *mcAG* foi capaz de encontrar a solução em 90% dos testes enquanto o *sAG* encontrou a solução em apenas 20% dos testes. A figura 4.32 mostra a curva de convergência média para os testes onde ambos os algoritmos foram capazes de achar a solução. Novamente, o logaritmo na base 10 foi utilizado no eixo das ordenadas para reforçar a diferença entre as curvas. Estes resultados sugerem que a metodologia proposta pode melhorar a performance do AG mesmo em problemas de dimensão elevada.

### 4.4.3 Conclusões

A construção de uma aproximação quadrática semi-global para a função-objetivo e a utilização dessa aproximação para definir uma operação de mudança de coordenadas são

ações que podem ser utilizadas na população dos AGs e que podem tornar-se uma importante ferramenta no processo de otimização. Os resultados apresentados neste capítulo sugerem que essa mudança de coordenadas é capaz de melhorar tanto a proporção de vezes nas quais o AG encontra o mínimo quanto o número de avaliações da função-objetivo necessárias para achar tal ponto. Esse comportamento foi verificado estando o ponto de mínimo incluído na região coberta pela população ou estando o ponto no exterior dessa região.

Os resultados sugerem que o procedimento de mudança de coordenadas pode ser incluído em qualquer AG quando aplicado a problemas genéricos irrestritos em que a função-objetivo seja não linear e com variáveis contínuas. Em particular, a otimização de funções do tipo *black-box* pode ser beneficiada com a utilização dessa metodologia.





## Capítulo 5

# Algoritmo Genético Multiobjetivo

Ao desenvolver modelos adequados para otimização de problemas reais, é frequente encontrarmos situações em que precisamos otimizar vários objetivos, e não apenas um. E na grande maioria das vezes, esses objetivos são conflitantes. Nesse caso, não existe uma solução única, mas sim um conjunto de soluções. Essas soluções procuradas são aquelas nas quais a melhora em um objetivo só pode ser atingida com a degradação de outros objetivos. Problemas com dois ou mais objetivos são denominados *multiobjetivo* e requerem algoritmos e ferramentas matemáticas diferentes daquelas utilizadas na resolução de problemas mono-objetivo. Até mesmo o conceito de otimalidade é modificado quando lidamos com problemas multiobjetivo.

Ao longo dos anos, várias metodologias determinísticas foram desenvolvidas para resolver problemas multiobjetivo. Uma grande variedade de técnicas de programação matemática está disponíveis (veja [38]), entretanto, muitas delas apresentam certas limitações ao lidar com esses problemas multiobjetivo. Por exemplo, várias técnicas são suscetíveis à forma do conjunto de solução, outras exigem a hipótese de diferenciabilidade das funções-objetivo e das restrições. A maioria dessas técnicas produz apenas uma solução a cada execução. Dessa forma, várias execuções diferentes do método, cada uma começando de um ponto inicial diferente, devem ser feitas para obtermos uma amostra desse conjunto de soluções do problema.

A principal vantagem do uso de técnicas de otimização evolucionárias em problemas multiobjetivo se deve ao fato de que essas técnicas permitem encontrar, de uma só vez, um conjunto de soluções para o problema multiobjetivo. Além disso, esse tipo de algoritmo é relativamente imune à variação na estrutura matemática do problema.

Neste capítulo apresentaremos duas utilizações distintas das aproximações quadráticas em algoritmos genéticos multiobjetivo que se baseiam na idéia I descrita no capítulo 2: utilizar as aproximações quadráticas para gerar uma estimativa para o ótimo do problema

multiobjetivo. As aproximações quadráticas podem gerar estimativas do ótimo tanto em problemas multiobjetivos irrestritos quanto em problemas com restrições de desigualdade e devem ser construídas utilizando-se a metodologia livre de derivada apresentada na seção 2.3, uma vez que o cálculo de derivadas no algoritmo genético representaria um número elevado de avaliações extras das funções. Nos dois casos, como a aproximação quadrática só é válida localmente, apenas os indivíduos pertencentes ao interior de uma vizinhança de um determinado ponto serão utilizados na construção da aproximação quadrática.

Na próxima seção, vamos fornecer as principais definições relativas a um problema multiobjetivo. Primeiramente, precisamos definir um problema multiobjetivo.

## 5.1 Definições Básicas

O problema de otimização multiobjetivo pode ser formulado como:

$$\begin{aligned} x^* = \min_x f(x) \\ \text{sujeito a: } \begin{cases} g_i(x) \leq 0; & i = 1, 2, \dots, r \\ h_j(x) = 0; & j = 1, 2, \dots, p \end{cases} \end{aligned} \quad (5.1)$$

sendo que  $x \in \mathbb{R}^n$ ,  $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $g(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^r$ , e  $h(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^p$ . As funções  $g_i$  e  $h_j$  são, respectivamente, funções de restrição de desigualdade e de igualdade. Os vetores  $x \in \mathbb{R}^n$  são chamados *vetores de parâmetros* do problema multiobjetivo e formam o *espaço de parâmetros*. Os vetores  $f(x) \in \mathbb{R}^m$  encontram-se num espaço vetorial denominado *espaço de objetivos*.

O propósito de um problema multiobjetivo é determinar um conjunto de indivíduos, denominado *conjunto Pareto-ótimo*, que contêm as soluções de (5.1). Tal conceito se deve ao trabalho de Vilfredo Pareto, que em 1896, definiu sua noção de situação Pareto-ótima, aquela na qual não é possível melhorar algum critério sem tornar outro pior. Essa noção caracteriza a relação conflitante que ocorre em problemas multiobjetivo. Os elementos desse conjunto serão definidos a seguir. Primeiramente, vamos definir o conceito de dominância.

A seguinte notação é empregada para vetores do  $\mathbb{R}^m$ :

$$\begin{aligned} x \leq y &\Rightarrow \{x_i \leq y_i \quad , \quad i = 1, \dots, m\} \\ x \neq y &\Rightarrow \{\exists i | x_i \neq y_i\} \end{aligned}$$

### Definição 1 Dominância:

Sejam  $x_1$  e  $x_2$  dois pontos do espaço de parâmetros. Diz-se que  $x_1$  domina  $x_2$  se  $f(x_1) \leq f(x_2)$  e  $f(x_1) \neq f(x_2)$ . Equivalentemente, diz-se que  $f(x_1)$  domina  $f(x_2)$ , no espaço de objetivos.

**Definição 2 Solução Pareto-ótima:**

Seja  $x^*$  um ponto do espaço de parâmetros. Diz-se que um ponto factível  $x$  é uma **solução Pareto-ótima** do problema multiobjetivo se não existe qualquer outra solução factível  $x$  no espaço de parâmetros tal que  $f(x) \leq f(x^*)$  e  $f(x) \neq f(x^*)$ , ou seja,  $x^*$  não é dominado por nenhum outro ponto factível.

Da mesma forma que em otimização com um único objetivo podemos falar em solução local e global, em otimização multiobjetivo temos solução Pareto-ótima local e global. Tais conceitos são definidos matematicamente a seguir.

**Definição 3 Solução Pareto-ótima local:**

Seja  $x^*$  um ponto factível no espaço de parâmetros. Diz-se que  $x^*$  é uma **solução localmente Pareto-ótima** do problema multiobjetivo numa dada vizinhança se existe  $\epsilon > 0$  tal que não exista qualquer outro  $x$  nessa vizinhança tal que  $f(x) \leq f(x^*)$  e  $f(x) \neq f(x^*)$ , ou seja,  $x^*$  não é dominado por nenhum outro ponto naquela vizinhança.

**Definição 4 Solução Pareto-ótima global:**

Seja  $x^*$  um ponto factível no espaço de parâmetros. Diz-se que  $x^*$  é uma **solução Pareto-ótima global** do problema multiobjetivo se não existe outro  $x$  factível tal que  $f(x) \leq f(x^*)$ .

Um outro conceito útil, relacionado ao conjunto de Pareto-ótimo, é o de solução utópica.

**Definição 5 Solução Utópica:**

A solução utópica  $y_*$  do problema multiobjetivo é definida como:

$$y_{i*} = f_i(x^i), \quad i = 1, 2, \dots, m$$

$$x^i = \min_x f_i(x)$$

A solução utópica  $y_*$  é uma escolha bastante conveniente para ser tomada como origem do espaço de objetivos, uma vez que tal escolha deixa todo o conjunto de Pareto incluído no primeiro quadrante, com interseções com os eixos coordenados correspondentes aos mínimos individuais de cada uma das funções-objetivo do problema.

Também é possível estender as condições de otimalidade de Kuhn-Tucker para o caso multiobjetivo, fornecendo as condições de Karush-Kuhn-Tucker para eficiência. As condições de Karush-Kuhn-Tucker para o problema multiobjetivo apenas com restrições de desigualdade, da forma,

$$\begin{aligned} x^* = \min_x f(x) \\ \text{sujeito a: } \left\{ \begin{array}{l} g_i(x) \leq 0; \quad i = 1, 2, \dots, r \end{array} \right. \end{aligned} \quad (5.2)$$

são dadas por:

**Teorema 1 Condições de Karush-Kuhn-Tucker para Eficiência:**

Uma solução factível  $x^*$  satisfaz as condições de Karush-Kuhn-Tucker para eficiência se:

- i. as funções coordenadas  $f_i$  e  $g_i$  são diferenciáveis e
- ii. existem vetores multiplicadores  $\mu^* \geq 0$ ,  $\lambda^* \geq 0$ , com pelo menos uma desigualdade estrita  $\lambda_i^* > 0$ , tais que

$$g_k(x^*) \leq 0 ; \quad \mu_k^* g_k(x^*) = 0 ; \quad k = 1, 2, \dots, r$$

$$\sum_{j=1}^m \lambda_j^* \nabla f_j(x^*) + \sum_{k=1}^r \mu_k^* \nabla g_k(x^*) = 0$$

## 5.2 Algoritmos Genéticos Multiobjetivo

A primeira idéia da possibilidade de uso de algoritmos evolucionários em problemas multiobjetivo apareceu na tese de doutorado de R.S. Rosenberg, em 1967. Entretanto nenhum algoritmo evolucionário multiobjetivo foi proposto, sendo o problema multiobjetivo re-escrito como um problema mono-objetivo e então resolvido utilizando-se o algoritmo genético.

David Schaffer é normalmente considerado com o primeiro autor a desenvolver um algoritmo evolucionário multiobjetivo durante os anos 80. Esse algoritmo, chamado *Vector Evaluated Genetic Algorithm* (VEGA), consiste em um algoritmo genético simples com um mecanismo de seleção modificado. A cada geração, um número de sub-populações é gerado através de uma seleção proporcional para cada função-objetivo. Essas sub-populações são então misturadas para obter uma nova população, na qual o AG irá aplicar os operadores de cruzamento e mutação da maneira usual. O VEGA apresentou vários problemas, sendo o principal devido à sua inabilidade de reter soluções com performance às vezes acima da média mas não para todos os objetivos. Soluções desse tipo, que seriam boas candidatas a soluções não-dominadas, não sobreviveriam ao esquema de seleção proposto.

A incorporação direta do conceito de soluções Pareto-ótimas em algoritmos evolucionários foi primeiramente proposta por David E. Goldberg em [24]. Enquanto criticava o algoritmo proposto por Schaffer, Goldberg sugeriu o uso de um ranking de soluções não-dominadas e de um mecanismo de seleção para mover a população na direção do conjunto de Pareto. A idéia básica é encontrar um conjunto de soluções na população que são não-dominadas pelo resto da população. Essas soluções, durante a atribuição de aptidão, recebem a maior classificação e são retiradas do conjunto sob análise. Com o restante da população, um outro conjunto de soluções não-dominadas é obtido e os representantes desse conjunto recebem a segunda maior classificação. Esse processo continua até que toda a população tenha recebido

uma classificação. Goldberg ainda sugeriu o uso de alguma técnica de nicho para impedir que o AG convergisse para um único ponto do conjunto de Pareto. Mecanismos de nicho permitiriam que os AGs mantivessem indivíduos ao longo de toda fronteira do conjunto de Pareto. Goldberg não propôs nenhum algoritmo em seu trabalho porém todos os algoritmos propostos posteriormente foram influenciados pelas suas idéias.

O esboço genérico de um algoritmo genético multiobjetivo é apresentado na figura 5.1. Os diversos algoritmos na literatura podem ser obtidos através de variações desse algoritmo geral com suas configurações particulares.

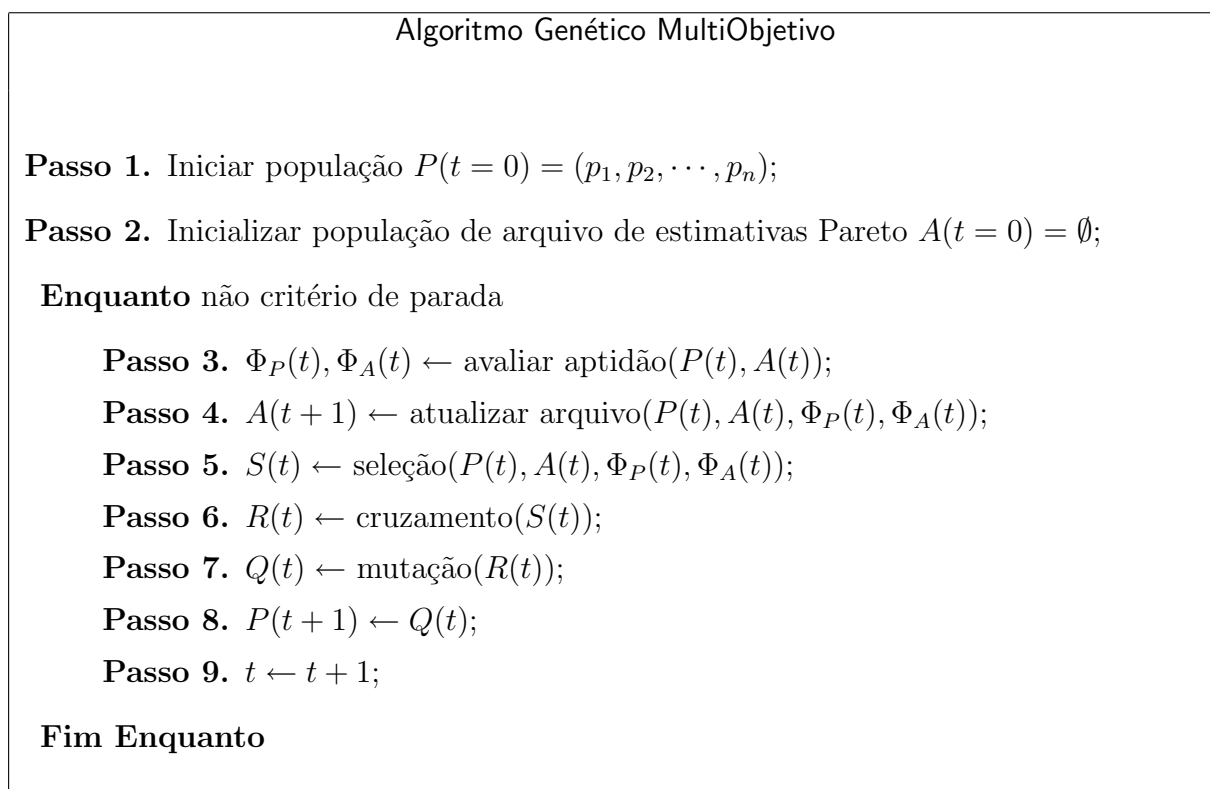


Figura 5.1: Esquema básico de um algoritmo genético multiobjetivo.

No contexto da otimização multiobjetivo, a diferença básica entre um algoritmo genético multiobjetivo e sua versão para otimização escalar reside na forma de atribuir a função de aptidão aos indivíduos. Essa é a principal dificuldade, uma vez que agora temos um vetor de valores de funções-objetivo e devemos atribuir um valor escalar de aptidão a cada indivíduo da população. Com relação aos operadores de cruzamento, mutação e seleção, não há diferenças.

Podemos ainda observar que existe uma população de arquivo que armazena as estimativas do conjunto Pareto encontradas até então. A inclusão de uma população de arquivo representa o elitismo em algoritmos evolucionários multiobjetivo, garantindo que o algoritmo

mantenha as melhores alternativas encontradas em todas as gerações e as forneça ao usuário ao final da execução. A principal motivação para o uso do elitismo nessa classe de algoritmos se deve ao fato que uma solução que é não-dominada em relação à população atual, pode não ser não-dominada em relação a todas populações produzidas até o momento. Então, é preciso que exista uma forma de garantir que as soluções apresentadas sejam não-dominadas com relação a todas populações geradas até o presente momento. Portanto, a maneira mais intuitiva de se fazer isso é através do armazenamento de um arquivo de memória externo de todas as soluções não-dominadas já encontradas. Convém salientar que só soluções não-dominadas por todas as soluções desse arquivo podem ser incorporadas ao mesmo.

Existe uma ligeira diferença entre a seleção nos algoritmos genéticos multiobjetivo e a seleção nos algoritmos genéticos mono-objetivo. Pelo fato de existir agora essa população de arquivo, a seleção pode ser feita usando os indivíduos que integram a população corrente ( $P(t)$ ), ou a população de arquivo ( $A(t)$ ), ou ambas as populações. Nos dois últimos casos, trata-se de uma seleção elitista (elitismo implícito), usada em alguns algoritmos. Uma vez atribuído um valor de aptidão a cada indivíduo em  $P(t)$  e  $A(t)$ , a operação de seleção por roleta ou torneio é executada da maneira tradicional.

Dois algoritmos genéticos multiobjetivo utilizados neste trabalho são descritos a seguir.

### 5.2.1 MOGA: Multi-Objective Genetic Algorithm

Proposto em 1993 por Fonseca e Fleming em [39], é considerado um dos representantes dos algoritmos evolucionários multiobjetivo conhecidos como algoritmos de *primeira geração*. Esse algoritmo, baseado no critério de dominância proposto por Goldberg [24], classifica cada indivíduo de acordo com o número de indivíduos que o dominam. Assim, a classificação recebida por um indivíduo  $p^{(i)}$  é dada por:

$$r(p^{(i)}) = 1 + d_i \quad (5.3)$$

em que  $d_i$  é o número de indivíduos da população que dominam  $p^{(i)}$ . Observe que os indivíduos não dominados recebem classificação  $r = 1$ , o melhor valor possível. Os indivíduos dominados são penalizados de acordo com a densidade da população na região em que se encontram. Quanto mais dominado por outras soluções da população um indivíduo for, pior será sua classificação.

Uma vez classificada a população, a próxima etapa é atribuir um valor de aptidão para cada indivíduo de acordo com sua classificação. Uma função não-linear para escalonar o valor de aptidão dos melhores indivíduos para os piores é utilizada, mas antes o vetor com os valores  $r(p^{(i)})$  é normalizado para o intervalo unitário e transformado para maximização.

O escalonamento é então dado por:

$$\bar{\delta}(p) = [\bar{r}(p)]^\alpha \quad (5.4)$$

em que  $\bar{r}$  corresponde ao valor normalizado para o intervalo  $[0, 1]$ , sendo  $\alpha \neq 1$ . Após o escalonamento não linear, um novo escalonamento linear é utilizado, obtendo-se assim o vetor de valores de aptidão  $\Phi_P(t)$ . O cálculo da função de aptidão pode envolver o uso de técnicas de nicho, como uma forma de priorizar uma melhor distribuição da população sobre a fronteira Pareto-ótima.

A função de aptidão de cada indivíduo é degradada com base na função de compartilhamento de aptidão <sup>1</sup>:

$$\phi(p^{(i)}) \leftarrow \frac{\phi(p^{(i)})}{s_i} \quad (5.5)$$

em que  $\phi$  é o valor de aptidão e  $s_i$  fornece uma estimativa da densidade de indivíduos na vizinhança de  $p^{(i)}$ . Matematicamente,  $s_i$  é dado por:

$$s_i = \sum_{j=1}^N s(d_{ij}) \quad (5.6)$$

com:

$$s(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma}\right)^\alpha, & d_{ij} \leq \sigma_s \\ 0, & d_{ij} > \sigma_{sh} \end{cases} \quad (5.7)$$

O parâmetro  $\sigma_s$  é denominado raio do nicho e controla a dispersão entre os indivíduos da população. Em problemas multiobjetivo, em geral é utilizado nicho no espaço de objetivos, com  $d_{ij}$ . Desta forma, o parâmetro de raio do nicho é um parâmetro a mais no algoritmo e que pode ser dependente do problema.

Uma vez calculado  $\Phi_P(t)$ , todos os indivíduos recebem um valor de aptidão igual ao valor máximo em  $\Phi_P(t)$ . Convém salientar que a versão original do MOGA não emprega uma população de arquivo. Entretanto, a versão do MOGA utilizada nesse trabalho faz uso do arquivo  $A(t)$ .

### 5.2.2 SPEA 2: Strength Pareto Evolutionary Algorithm 2

A segunda geração dos algoritmos evolucionários multiobjetivo começou quando o elitismo tornou-se um mecanismo padrão dos algoritmos. Apesar de alguns estudos preliminares relacionados com a noção de elitismo nos algoritmos evolucionários multiobjetivo, a

---

<sup>1</sup>Tradução para fitness sharing.

introdução formal desse conceito se deve à Eckart Zitzler através da publicação do algoritmo *Strength Pareto Evolutionary Algorithm* (SPEA) em [40]. Após a publicação desse artigo, novos algoritmos incorporaram o mecanismo de elitismo. Na prática, é através do uso do elitismo que torna-se possível garantir a convergência dos algoritmos evolucionários multiobjetivo.

Posteriormente, foram identificadas algumas limitações importantes referentes ao SPEA e uma versão mais sofisticada foi proposta em seguida, o SPEA 2 (veja [41]).

SPEA 2 tem uma estrutura idêntica ao algoritmo multiobjetivo descrito na figura 5.1. Esse algoritmo utiliza um arquivo contendo soluções não-dominadas previamente encontradas. Esse arquivo é denominado *external non-dominated set* e denotado aqui por  $\mathcal{S}$ . A cada geração, soluções não-dominadas são armazenadas nesse conjunto. Para cada indivíduo em  $\mathcal{S}$ , um valor de *força* (strength) é calculado. Esse valor de *força* é similar à classificação dos indivíduos adotada pelo MOGA, uma vez que ele é proporcional ao número de soluções que um determinado indivíduo domina. A aptidão de cada indivíduo da população atual é calculada de acordo com as *forças* de todas as soluções não-dominadas de  $\mathcal{S}$  que dominam esse determinado indivíduo e por todas as soluções que o indivíduo domina. O processo do cálculo da aptidão do SPEA 2 considera, ao mesmo tempo, proximidade do conjunto Pareto real e distribuição das soluções. Dessa forma, no lugar de técnicas de nicho baseadas em distância, o critério de dominância é usado para garantir que as soluções estejam distribuídas ao longo do conjunto de Pareto real. Apesar dessa metodologia não exigir um raio do nicho, a eficiência da mesma depende do tamanho de  $\mathcal{S}$ . De fato, uma vez que o conjunto  $\mathcal{S}$  participa do processo de seleção, se seu tamanho for muito grande, a pressão de seleção poderá ser reduzida, diminuindo assim a busca. Desta forma, os autores sugerem que o tamanho do arquivo  $\mathcal{S}$  fique abaixo de um valor  $\chi$  fixado de maneira heurística.

No SPEA 2, a atualização do arquivo consiste em colocar as soluções não-dominadas de  $A(t) \cup P(t)$  em  $A(t+1)$ . Se o tamanho de  $A(t+1)$  for menor do que  $\chi$ , as melhores soluções em  $A(t) \cup P(t)$ , de acordo com os valores de aptidão, são colocadas em  $A(t+1)$  até que se complete o arquivo. Se o tamanho de  $A(t+1)$  for maior do que  $\chi$ , emprega-se uma estratégia especial de redução do arquivo que garante a preservação das soluções que se encontram na fronteira do conjunto de Pareto. Além desses mecanismos, o SPEA 2 usa também uma técnica de estimar a densidade da região mais próxima que guia a busca com mais eficiência.



## 5.3 Medidas de Desempenho para AG Multiobjetivo

Usualmente a qualidade do conjunto de soluções não-dominadas encontrada por um algoritmo pode ser estimada levando em consideração três aspectos:

- (a) o número de soluções Pareto-ótimas presentes no conjunto;
- (b) a proximidade das soluções obtidas em relação ao conjunto Pareto-ótimo verdadeiro (supondo conhecer sua localização) e,
- (c) a distribuição dessas soluções obtidas no espaço.

O problema de comparar o desempenho de algoritmos evolucionários multiobjetivo tem sido o palco de várias discussões, veja por exemplo [42, 43, 44, 45]. Algumas medidas de performance exigem o conhecimento prévio do conjunto Pareto-ótimo, o que não é possível em alguns problemas reais. Além disso, uma única medida de desempenho não é suficiente para se quantificar os três aspectos citados no parágrafo anterior [42].

Neste trabalho, usaremos algumas medidas de comparação de algoritmos que não necessitam do conhecimento prévio do conjunto Pareto-ótimo real. Porém, devemos identificar aspectos desejáveis no conjunto não-dominado produzido pelos algoritmos.

Com o objetivo de comparar a qualidade de dois conjuntos de soluções, medidas quantitativas que serão utilizadas estão brevemente descritas a seguir. Sejam  $\mathcal{A}$  e  $\mathcal{B}$  o conjunto de soluções não-dominadas produzidas por dois algoritmos distintos.

### (1) Non-Dominated Combined Set Ratio (NDCSR)

Esta é uma medida simples que não necessita do conjunto Pareto-ótimo real. A medida NDCSR é definida a seguir:

Seja  $\mathcal{C}$  = pontos não-dominados de  $(\mathcal{A} \cup \mathcal{B})$ . A medida NDCRS para o conjunto  $\mathcal{A}$  representa a proporção dos elementos em  $\mathcal{C}$  que estão presentes em  $\mathcal{A}$ .

Apesar de ser uma métrica simples, existem problemas associados à essa métrica: um conjunto com um elevado valor da métrica NDCRS pode, geometricamente, ser concentrado em apenas uma região, enquanto outro conjunto com valor menor para a métrica NDCRS pode cobrir uma região maior do conjunto de Pareto. Maiores detalhes podem ser encontrados em [43, 44, 45]

### (2) Contagem de Esferas

Com o objetivo de medir a distribuição de cada conjunto não-dominado obtido, o seguinte experimento será efetuado:

1. defina um raio  $r$  no espaço de objetivos;
2. execute cada algoritmo, obtendo uma estimativa do conjunto de Pareto;
3. coloque uma esfera de raio  $r$  centrada em algum ponto do conjunto de estimativas;
4. inicialize um contador de esferas igual a um;
5. exclua todos os pontos que estão localizados no interior dessa esfera;
6. dentre os pontos restantes, centre uma nova esfera no ponto que está mais próximo ao centro da esfera anterior e incremente o contador de esferas;
7. vá para o passo 5 até que não haja mais nenhum ponto no conjunto de estimativa.

Através deste experimento, é possível identificar o conjunto de estimativas que tem a maior distribuição pelo espaço: é simplesmente o conjunto que obteve o maior número de esferas. A contagem de esferas, proposta nesse trabalho, é uma métrica fácil de se calcular e que pode ser aplicada a problemas com qualquer número de funções-objetivo. Depende apenas do raio escolhido para as esferas. Novos experimentos estão sendo feitos para se avaliar a dependência dessa métrica em relação à esse parâmetro— o raio da esfera. A figura 5.2 ilustra essa métrica em um problema bi-objetivo. Essa métrica foi utilizada em [28, 46].

### (3) S Metric

A definição da S metric foi apresentada em [47]. Essa métrica calcula o hipervolume de uma região multi-dimensional englobada por  $\mathcal{A}$  e por um ponto de referência, calculando portanto o tamanho da região que  $\mathcal{A}$  domina. A idéia básica é que quanto maior a hiper-área que as soluções dominam, melhor é o conjunto.

Essa métrica apenas exige o conhecimento de algum ponto da fronteira superior da região na qual todos os pontos factíveis se encontram. A escolha desse ponto de referência, apesar de arbitrária, afeta o valor da métrica. Além disso, é uma métrica sensível à forma do conjunto de Pareto (presença de pontos isolados, conexidade). Possui um grande custo computacional no caso de problemas com mais de duas funções-objetivo ou em conjuntos não-dominados com grande cardinalidade. Nesse trabalho, usaremos essa métrica apenas em problemas bi-objetivo.

A figura 5.3 ilustra essa métrica em um problema bi-objetivo.

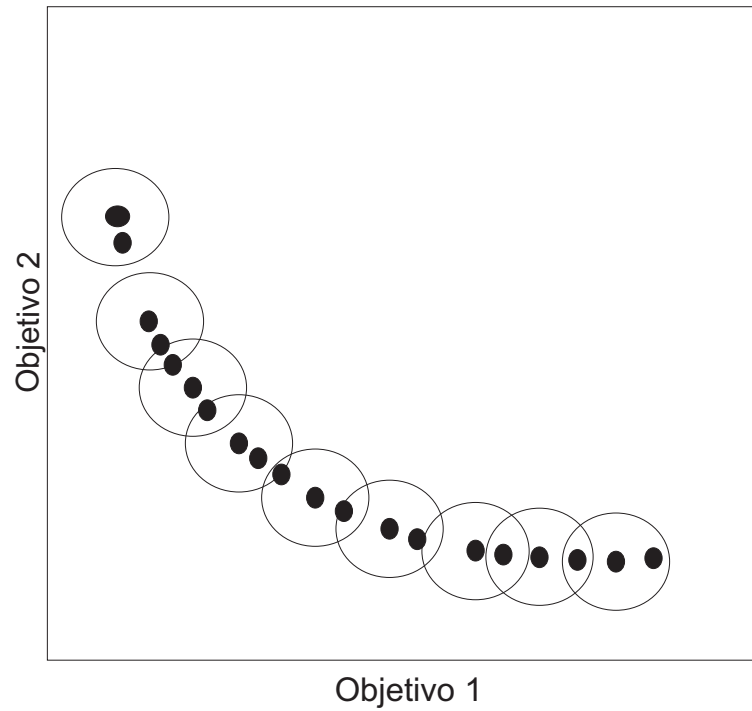


Figura 5.2: Contagem de esferas em um problema bi-objetivo. Observe que o conjunto de Pareto possui 20 pontos, mas a contagem de esferas resulta em apenas 9 esferas.

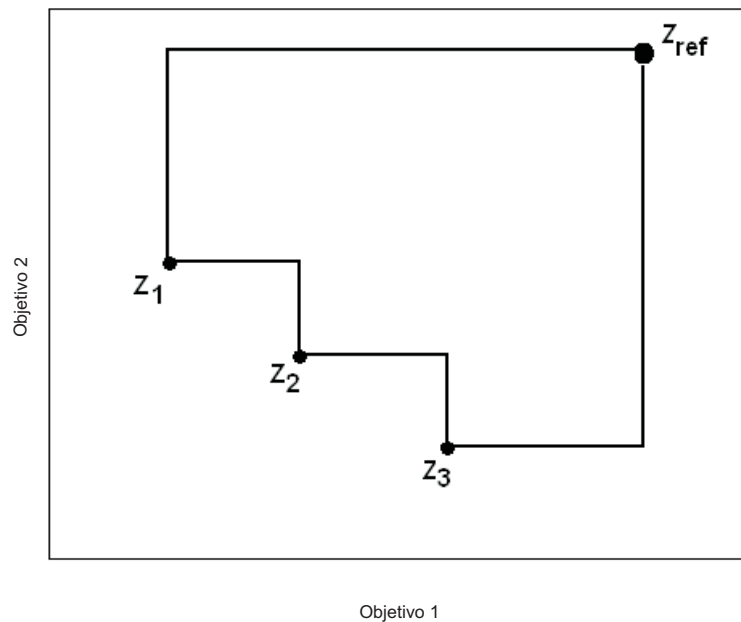


Figura 5.3: Em um problema bi-objetivo, a S metric calcula a área da região poligonal coberta pelos pontos  $z_1$ ,  $z_2$ ,  $z_3$  e um ponto de referência  $z_{ref}$ .

### 5.3.1 Testes Estatísticos

Uma comparação estatística de dois resultados é possível através do uso de um teste estatístico adequado. Neste trabalho, a diferença média entre dois resultados de contadores de esferas (ou alternativamente, dois valores para a *S* metric) é utilizada como ‘objeto’ no teste estatístico descrito nessa seção (veja [48]). A significância desses resultados observados é calculada usando uma técnica aqui denominada *teste de aleatoriedade*. Essa é uma técnica simples cuja idéia básica é que, se o resultado observado ocorreu por acaso, então esse valor não será notado em uma distribuição de frequência dos resultados obtidos ao efetuar uma reordenação das amostras. O *teste de aleatoriedade* segue as seguintes diretrizes:

- calcule o ‘objeto’ que se deseja validar para as amostras de cada algoritmo; esse é o valor observado;
- aleatoriamente reordene metade das amostras de um algoritmo com a metade das amostras do outro. Calcule novamente o objeto desejado como anteriormente;
- repita o passo anterior até que 5000 valores aleatórios tenham sido gerados e construa uma distribuição de frequência desses valores;
- se o valor observado estiver dentro da faixa central do histograma que corresponde a 99% da distribuição, então aceite a hipótese nula. Caso contrário, aceite a hipótese alternativa.

Observe que o valor observado é incluído como um dos valores da reordenação uma vez que, se a hipótese nula for verdadeira então esse valor é um dos possíveis resultados aleatórios. A hipótese nula indica que o valor observado ocorreu por acaso e, portanto, não existe diferença entre os dois algoritmos. A hipótese alternativa indica que o resultado observado não ocorreu por acaso e que um algoritmo é superior ao outro.

Os resultados do *teste de aleatoriedade* são muito simples de se visualizar. Os resultados aleatórios são apresentados no histograma em cinza, enquanto o resultado observado é destacado como um círculo preto. Os testes estatísticos, realizados nesse trabalho, foram feitos usando

**média(Algoritmo Proposto)-média(Algoritmo Clássico)**

e, portanto diferenças situadas no lado positivo do histograma favorecem o algoritmo proposto em relação ao algoritmo clássico. Normalmente é visível se o resultado é estatisticamente significativo ou não, porém algumas vezes uma análise mais rigorosa se faz necessária. A figura 5.4 mostra um histograma típico desse teste estatístico.

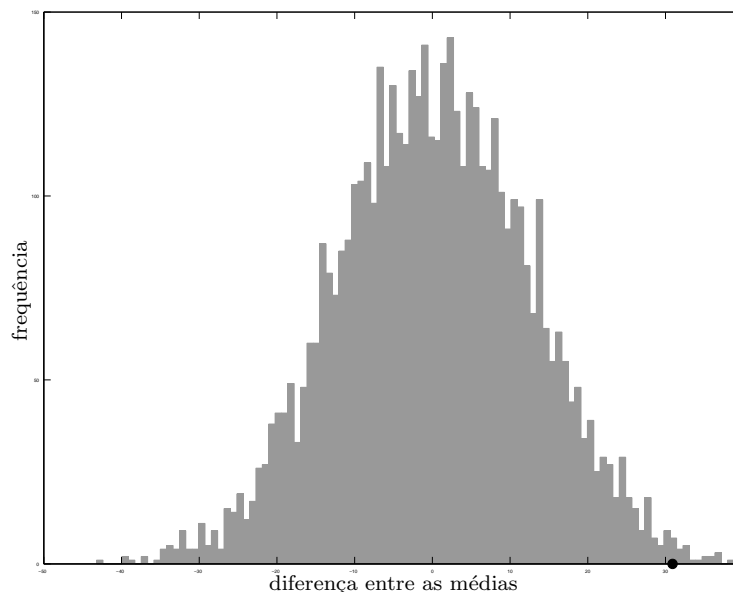


Figura 5.4: Histograma de frequência obtido através do teste de aleatoriedade. O valor observado está indicado pelo círculo preto. Os resultados são favoráveis para o algoritmo proposto se o resultado observado se encontrar no lado positivo do histograma.

## 5.4 Problemas Multiobjetivo Irrestritos

As técnicas multiobjetivo evolucionárias têm obtido sucesso devido a características tais como flexibilidade, capacidade de busca global e a possibilidade de busca e de manter um conjunto de estimativas das soluções ótimas. Entretanto, está agora bem estabelecido que uma tarefa árdua para um AG (tanto em problemas mono quanto em multiobjetivo) é uma busca “mais fina e especializada” em espaços complexos.

Apresentaremos nessa seção um novo procedimento de busca local para problemas multiobjetivo não-lineares irrestritos<sup>2</sup>. Esse procedimento é baseado na construção de aproximações quadráticas locais para as funções-objetivo. Utilizando essas aproximações quadráticas, um problema auxiliar quadrático pode ser construído e as soluções ótimas podem ser obtidas através das condições de Kuhn-Tucker para problemas multiobjetivo. Neste sentido, os modelos quadráticos são usados para produzir indivíduos localmente melhores, evitando o número excessivo de avaliações de funções que ocorre quando a busca local é diretamente aplicada sobre as funções originais [49].

Nesse capítulo, a metodologia de busca local será acoplada ao SPEA 2, com o objetivo de analisar o benefício da hibridização com um algoritmo que representa o atual estado da arte em métodos evolucionários multiobjetivo. Entretanto, esse operador de busca local pode, em princípio, ser usado com qualquer algoritmo evolucionário multiobjetivo.

<sup>2</sup>Os resultados que serão apresentados nessa seção encontram-se publicados em [46].

### 5.4.1 Metodologia

Considere o seguinte problema de otimização não-linear multiobjetivo com restrições não lineares de desigualdade:

$$\begin{aligned} x^* &= \min_x f_i(x) \\ i &= 1, \dots, j \end{aligned} \quad (5.8)$$

$$\text{com } L_i^- \leq x_i \leq L_i^+$$

sendo  $L_i^-$  e  $L_i^+$ , respectivamente, o limite inferior e superior da variável corespondente  $x_i$ .

Seja  $x_0$  o ponto selecionado para a busca local. As aproximações quadráticas para as funções-objetivo do problema são geradas usando a metodologia descrita na seção 2.3. Neste caso, usaremos na construção dessas aproximações apenas os pontos (e as respectivas avaliações de funções) que pertencem a uma vizinhança de  $x_0$ . Essa vizinhança é novamente definida como

$$\mathcal{N}(x_0) = \{x : (x - x_0)^T R(x - x_0) \leq 1\} \quad (5.9)$$

e

$$R_{ij} = \begin{cases} [0.1(u_i - l_i)]^{-1}, & i = j \\ 0, & i \neq j \end{cases} \quad (5.10)$$

sendo  $l_i$  e  $u_i$  respectivamente o valor mínimo e máximo para a  $i$ -ésima variável, sendo  $H$ , por construção, uma matriz semi-definida positiva.

Uma vez obtida uma aproximação quadrática para cada função-objetivo  $\hat{f}_i(z) = z^T H_i z + r_i^T z + \gamma_i$ , podemos obter o ponto de mínimo da função  $\hat{f}_i$ ,

$$z_{f_i} = -\frac{H_i^{-1} r_i}{2},$$

e então, re-escrever a expressão analítica para a função

$$\hat{f}_i(z) = (z - z_{f_i})^T H_i (z - z_{f_i}) - c_i$$

sendo

$$c_i = 0.25 r_i^T H_i^{-1} r_i + \gamma_i.$$

O vetor gradiente pode ser calculado como

$$\nabla(\hat{f}_i) = 2H_i(z - z_{f_i}).$$

Com as aproximações quadráticas calculadas, podemos escrever um problema multiob-

jetivo quadrático associado:

$$\begin{aligned} x^* &= \min_x \hat{f}_i(x) \\ i &= 1, \dots, n \end{aligned} \tag{5.11}$$

$$\text{com } L_i^- \leq x_i \leq L_i^+$$

Escolhendo-se uma das funções-objeto quadráticas, por exemplo  $\hat{f}_1$ , e considerando a fatorização de Cholesky da matriz  $H_1$ ,

$$H = L^T L$$

efetuamos a seguinte mudança de coordenada

$$\tilde{z} = Lz. \tag{5.12}$$

Usando essa mudança de coordenada, podemos re-escrever as expressões para todas as funções-objeto:

$$\begin{aligned} \tilde{f}_1(z) &= (z - z_{f_1})^T (z - z_{f_1}) - c_1 \\ \tilde{f}_i(z) &= (z - z_{f_i})^T Q_i (z - z_{f_i}) - c_i \end{aligned} \tag{5.13}$$

$$i = 2, \dots, n$$

sendo  $z_{f_i}$  os novos pontos de mínimo de  $\tilde{f}_i$  e  $Q_i$  representam as novas matrizes Hessianas de cada função após a mudança de coordenada.

Desta forma, com as expressões dadas em (5.13), podemos re-escrever o problema quadrático associado (5.11)

$$\begin{aligned} x^* &= \min_x \tilde{f}_i(x) \\ i &= 1, \dots, n \end{aligned} \tag{5.14}$$

$$\text{com } L_i^- \leq x_i \leq L_i^+$$

Sabemos que, para o problema quadrático (5.14),

$$\nabla \tilde{f}_1(z) = 2(z - z_{f_1}) \tag{5.15}$$

e

$$\nabla \tilde{f}_i(z) = 2Q_i(z - z_{f_i}) \tag{5.16}$$

representam as expressões analíticas dos vetores gradientes de  $\tilde{f}_1$  e de  $\tilde{f}_i$  para todo  $i = 2, \dots, n$ . Além disso, através das condições de Karush-Kuhn-Tucker, temos

$$\lambda_1 \nabla \tilde{f}_1 + \sum_{i=2}^n \lambda_i \nabla \tilde{f}_i = 0 \quad (5.17)$$

com  $\lambda_i \leq 0$ , para todo  $i$ .

Desta forma, utilizando (5.15), (5.16) e (5.17), obtemos

$$\lambda_1 \cdot (z - z_{f_1}) + \sum_{i=2}^n \lambda_i \cdot Q_i \cdot (z - z_{f_i}) = 0 \quad (5.18)$$

e finalmente

$$\left( I + \sum_{i=2}^j \frac{\lambda_i}{\lambda_1} \cdot Q_i \right) z = z_{f_1} + \sum_{i=2}^j \frac{\lambda_i}{\lambda_1} \cdot Q_i \cdot z_{f_i} \quad (5.19)$$

Os parâmetros,  $\lambda_i$ , da equação (5.19), variam dentro de todo eixo positivo real:

$$0 \leq \lambda_i < \infty,$$

o que não é conveniente. Entretanto, a mesma equação pode ser reparametrizada por

$$\left[ \left( 1 - \sum_i \gamma_i \right) I + \sum_i \gamma_i \cdot Q_i \right] z = \left( 1 - \sum_i \gamma_i \right) z_{f_1} + \sum_i \gamma_i \cdot Q_i \cdot z_{f_i} \quad (5.20)$$

para todo  $i = 1, \dots, n - 1$ , sendo  $\gamma_i \in [0 \ 1]$  para todo  $i$ .

Observando a equação (5.20), podemos ver que essa equação fornece uma região paramétrica unindo os pontos de mínimo  $z_{f_i}$ , dois a dois. Para cada conjunto de parâmetros  $\gamma$ , obtemos um ponto pertencente ao conjunto Pareto-ótimo do problema (5.14).

No final desse procedimento, devemos retornar às coordenadas originais do problema (5.11), nas quais as funções-objetivo são avaliadas. Os pontos que pertencem a essa região parametrizada por (5.20) representam o conjunto Pareto-ótimo para o problema quadrático associado (5.11). À medida que o algoritmo evolui, a melhor função quadrática que aproxima cada função objetivo é modificada, obtendo-se assim, progressivamente, melhores estimativas para o conjunto Pareto-ótimo. Ao final do algoritmo, as funções quadráticas tornam-se mais precisas e o conjunto de estimativas produzido pela busca local representará localmente o conjunto Pareto-ótimo do problema original (5.8).



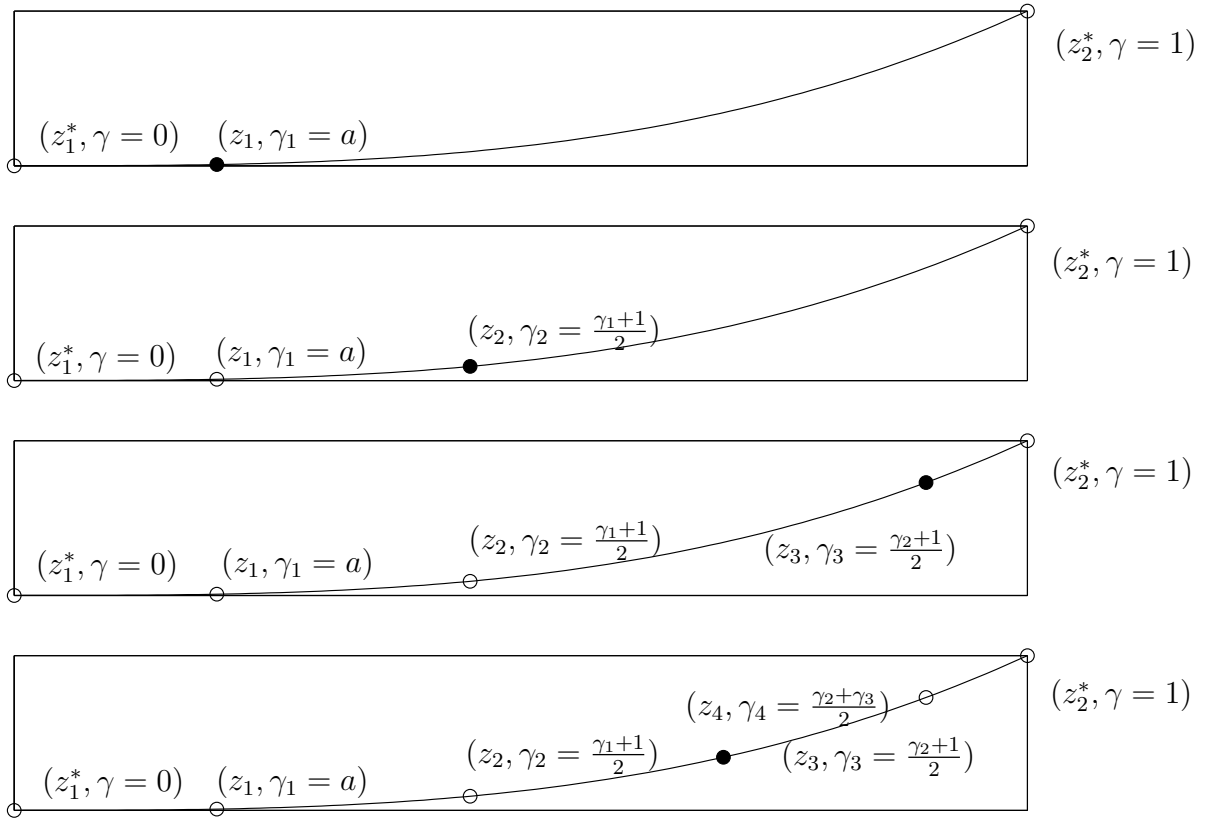


Figura 5.5: Exemplo de uma seqüência de quatro passos para se obter pontos bem distribuídos sobre o conjunto de Pareto, sendo que cada figura representa um determinado passo. Os pontos  $z_1^*$  e  $z_2^*$  correspondem aos pontos de mínimo de duas funções-objeto. O novo ponto calculado em cada passo está representado com um círculo preto. No primeiro passo (primeira figura), o ponto  $z_1$  foi obtido escolhendo-se  $\gamma$  no intervalo  $[0, 1]$ . No segundo passo (segunda figura), como a distância entre  $z_1$  e  $z_2^*$  é maior do que a distância entre  $z_1$  e  $z_1^*$ , o novo ponto foi calculado fazendo  $\gamma_2 = \frac{\gamma_1 + 1}{2}$ . No terceiro passo (terceira figura), como a distância entre  $z_2$  e  $z_2^*$  é maior do que a distância entre  $z_2$  e  $z_1^*$ , o novo ponto foi calculado fazendo  $\gamma_3 = \frac{\gamma_2 + 1}{2}$ . No quarto passo (quarta figura), como a distância entre  $z_3$  e  $z_2$  é maior do que a distância entre  $z_3$  e  $z_2^*$ , o novo ponto foi calculado fazendo  $\gamma_4 = \frac{\gamma_2 + \gamma_3}{2}$ .

Devemos observar que, com o propósito de cobrir o conjunto de Pareto com uma amostra representativa de pontos, os parâmetros  $\gamma$  devem ser escolhidos adequadamente. Neste trabalho, o primeiro parâmetro  $\gamma_i = a$  é escolhido aleatoriamente no intervalo  $[0, 1]$ . Para este valor, calculamos o ponto correspondente  $(z, \gamma = a)$ , e em seguida, calculamos a distância entre  $(z, \gamma = a)$  e o ponto correspondente de  $(z, \gamma_i = 0)$  e a distância entre  $(z, \gamma = a)$  e o ponto correspondente de  $(z, \gamma_i = 1)$ . O próximo  $\gamma_i$  deverá ser associado a um ponto dentro do maior intervalo, em outras palavras, o próximo  $\gamma_i$  é escolhido como o ponto central do maior intervalo,  $[(z, \gamma_i = 0), (z, \gamma_i)]$  ou  $[(z, \gamma_i), (z, \gamma_i = 1)]$ . A figura 5.5 mostra alguns passos para se escolher os pontos usando a metodologia descrita acima.

Desta maneira, as soluções do problema associado (5.11) fornecem um conjunto de indivíduos localmente melhorados. Como introduzir esse conjunto de soluções na população será definido posteriormente.

Como um exemplo, considere o problema bi-objetivo quadrático a seguir:

$$\begin{aligned} x^* &= \min_x f_i(x) \\ i &= 1, 2 \end{aligned} \quad (5.21)$$

sendo

$$f_i(x) = (x - x_{f_i})^T H_i (x - x_{f_i}) \quad , \quad i = 1, 2$$

$$H_i = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$$

$$x_{f_1} = [1; 1] \quad , \quad x_{f_2} = [-5; -5]$$

Usando esse problema bi-objetivo e aplicando o procedimento anteriormente descrito, após a mudança de coordenada, o caminho paramétrico unindo  $x_{f_1}$  e  $x_{f_2}$  é dado por

$$[(1 - \gamma)I - \gamma.G]z = (1 - \gamma)z_{f_1} + \gamma.G.z_{f_2} \quad (5.22)$$

sendo  $z_{f_1}$  e  $z_{f_2}$  os pontos de mínimos das funções e  $G$  é a matriz Hessiana de  $f_2$  após a mudança de coordenada. A figura 5.6 mostra as curvas de nível e o ponto de mínimo de cada função objetivo. Os pontos no caminho unindo os dois mínimos são obtidos variando-se  $\gamma$  no intervalo  $[0, 1]$ .

Considere agora o problema tri-objetivo:

$$\begin{aligned} x^* &= \min_x f_i(x) \\ i &= 1, 2, 3 \end{aligned} \quad (5.23)$$

sendo

$$f_i(x) = (x - x_{f_i})^T \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} (x - x_{f_i}) \quad , \quad i = 1, 2, 3$$

$$x_{f_1} = [1; 1] \quad , \quad x_{f_2} = [-5; -5] \quad , \quad x_{f_3} = [5; -5]$$

Similarmente ao problema bi-objetivo, aplicamos a este problema a metodologia de busca local, e como todas as matrizes Hessianas são iguais à matriz identidade, a região paramétrica

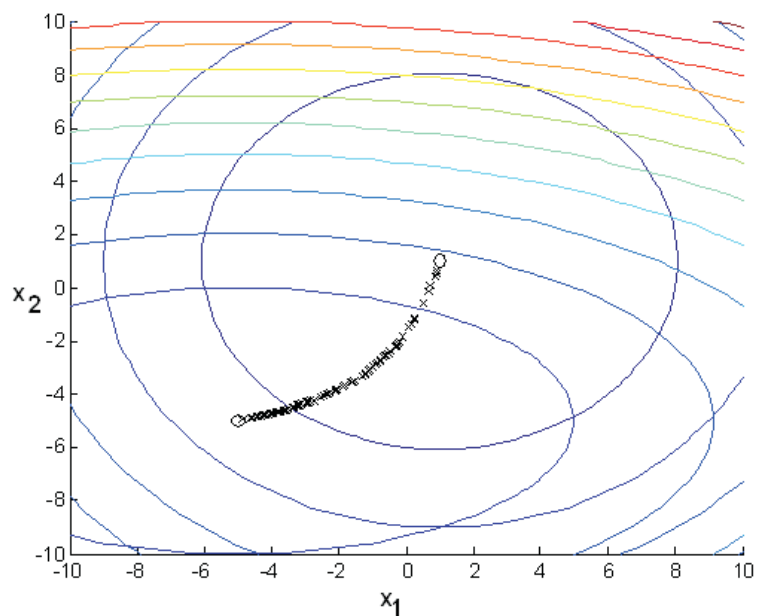


Figura 5.6: Caminho paramétrico, dado pela equação (5.22), unindo os pontos de mínimos de cada função-objetivo do problema bi-objetivo (5.21)

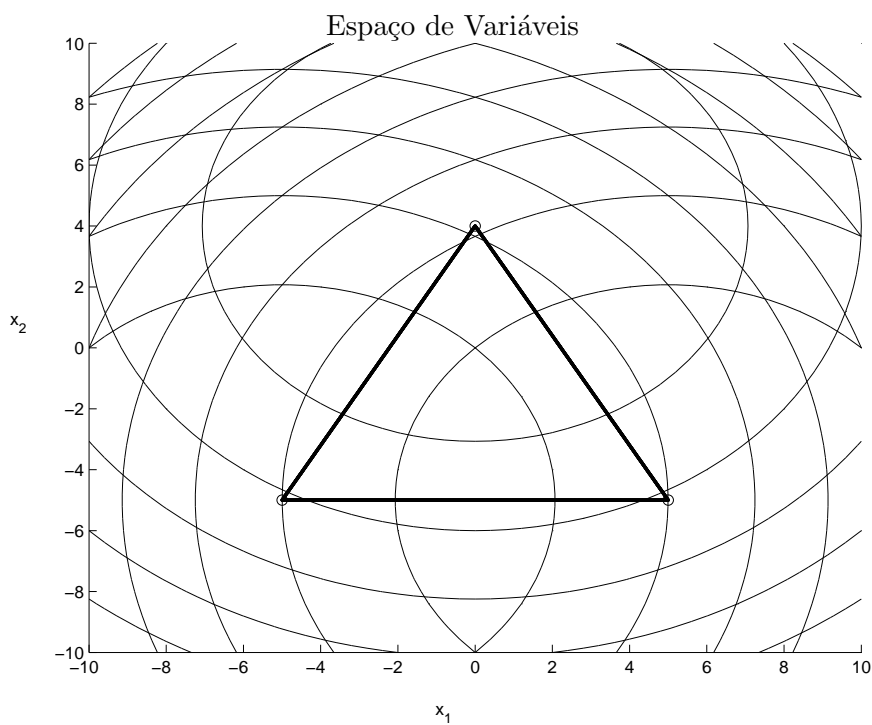


Figura 5.7: Região paramétrica, dada pela equação (5.24), unindo os pontos de mínimos das funções-objetivo do problema tri-objetivo (5.23). Os pontos em cada borda do triângulo são obtidos variando-se  $\gamma_i$  individualmente no intervalo  $[0, 1]$ .

unindo os pontos de mínimo de cada função-objetivo é dada por

$$\begin{aligned} [(1 - \gamma_1\gamma_2)I - \gamma_1 + \gamma_2]x &= (1 - \gamma_1\gamma_2)x_{f_1} + \\ &+ \gamma_1x_{f_2} + \\ &+ \gamma_2x_{f_3} \end{aligned} \tag{5.24}$$

sendo  $x_{f_i}$ , para todo  $i = 1, 2, 3$ , os pontos de mínimo das funções. A figura 5.7 mostra as curvas de nível e os pontos de mínimo de cada função-objetivo. Os pontos na fronteira da região são obtidos variando-se cada  $\gamma_i$  individualmente (sendo os outros iguais a zero) no intervalo  $[0, 1]$ . Os pontos no interior da região são obtidos variando-se os dois valores  $\gamma_i$  ao mesmo tempo.

### 5.4.2 Acoplando a Busca Local No SPEA 2

Cada solução do problema quadrático associado fornece um indivíduo localmente melhorado. Precisamos definir quais soluções serão selecionadas para a busca local e quantos indivíduos localmente melhorados serão gerados para cada indivíduo selecionado.

Em um AG multiobjetivo, é usual armazenar um arquivo de soluções não-dominadas encontradas durante o processo evolucionário. O ponto que será selecionado para a busca local deverá ser escolhido neste arquivo de memória. Com o objetivo de se evitar a seleção de pontos em regiões concentradas, este ponto será selecionado utilizando alguma técnica de nicho, fazendo com que um ponto isolado tenha maior probabilidade de ser escolhido do que um ponto pertencente a uma região densa.

Assim,  $k$  indivíduos serão selecionados desse arquivo de indivíduos não-dominados levando-se em consideração alguma medida de densidade. Em seguida, as aproximações quadráticas serão geradas utilizando-se apenas pontos no interior da vizinhança do ponto selecionado. Finalmente, para cada indivíduo selecionado, aplicamos o operador de busca local para gerar  $r$  novos indivíduos melhorados. Cada novo indivíduo produzido deve, obrigatoriamente, estar contido no interior da vizinhança pré-estabelecida do ponto inicial. Caso essa condição não seja satisfeita, calculamos o ponto na fronteira dessa vizinhança que está mais próximo do novo indivíduo. Após esses passos, devemos avaliar os novos  $k \times r$  indivíduos nas funções-objetivo originais e, então decidir se esses novos indivíduos serão, ou não, incluídos no arquivo de soluções não-dominadas. A figura 5.8 ilustra esse procedimento descrito acima.

O custo dessa metodologia a cada geração, em termos de avaliações de funções-objetivo, é dado pelo número  $m \times k \times r$  sendo  $m$  o número de funções-objetivo do problema. O operador

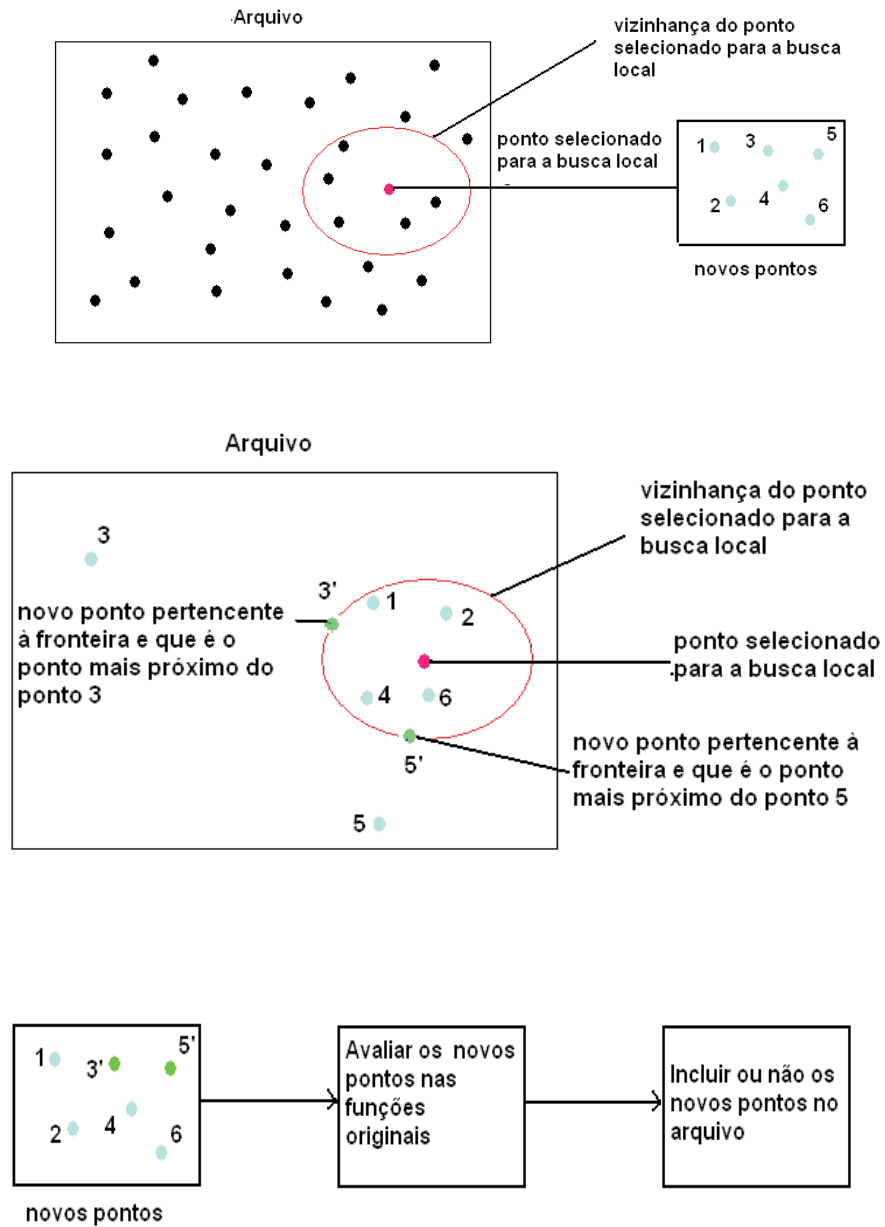


Figura 5.8: Etapas necessárias para acoplar a busca local ao SPEA 2.

de busca local será executado a cada  $t$  gerações. Então, o custo total desta metodologia, em termos de avaliação de função, é dado por

$$\frac{n_g}{t} \times m \times k \times r,$$

sendo  $n_g$  o número total de gerações do algoritmo. A escolha dos parâmetros  $t$ ,  $k$  e  $r$  devem levar em consideração o balanço desejado entre a busca local e a busca global. Quando o problema possui funções-objetivo difíceis de se avaliar (cara do ponto de vista computacional), a escolha desses parâmetros torna-se ainda mais importante.

Para hibridizar o operador de busca local com o SPEA 2, introduziremos a busca local logo após a atualização do arquivo. Os  $k \times m$  novos indivíduos serão avaliados nas funções-objetivo originais e, então, misturados aos indivíduos do arquivo. Finalmente, o arquivo deverá ser atualizado novamente, antes de executar o mecanismo de seleção.

### 5.4.3 Testes e Resultados

O problema a seguir, designado aqui como Problema de KUR, foi utilizados nos testes.

$$\min \begin{cases} f_1(x) = \sum_{i=1}^{n-1} (-10 \cdot \exp(0.2 \sqrt{x_i^2 + x_{i+1}^2})) \\ f_2(x) = \sum_{i=1}^n (|x_i|^{0.8} + \sin^3(x_i)) \end{cases} \quad (5.25)$$

$$-5 \leq x_i \leq 5$$

$$i = 1, \dots, n$$

O problema de Kursawe [50] apresenta uma função multi-modal em uma componente e uma interação dois a dois entre as variáveis da outra componente. O conjunto Pareto-ótimo não é conexo, possui um ponto isolado e partes côncavas e convexas.

Este problema será usado com 3 e 10 variáveis. Para cada caso, ambos os algoritmos, o SPEA 2 puro e o SPEA 2 híbrido foram rodados várias vezes. Os parâmetros utilizados em cada teste estão listados na tabela 5.1. O número máximo de gerações (maxgen), o tamanho da população e o tamanho do arquivo variam de acordo com a dimensão do problema. Esses valores estão listados na tabela 5.2. O número máximo de gerações foi o único critério de parada utilizado.

Com o objetivo de comparar os conjuntos Pareto-ótimo produzido por cada um dos algoritmos, vamos utilizar as métricas NDCSR, a contagem de esferas e a S Metric. Todas essas métricas encontram-se descritas na seção 5.3 deste capítulo.

Tabela 5.1: Parâmetros de cada algoritmo.

Probabilidade de mutação ( $p_m$ )	0.05
Probabilidade de cruzamento ( $p_c$ )	0.8

Tabela 5.2: Parâmetros de cada algoritmo.

Parâmetros	Dimensão 3	Dimensão 10
MaxGen	100	200
Tamanho da População	40	100
Tamanho do Arquivo	30	40

Selecionando-se apenas um ponto para a busca local, a tabela 5.3 mostra os valores médios, após todas as execuções, para cada uma das métricas escolhidas. Escolhendo-se, aleatoriamente, uma execução de cada algoritmo, a figura 5.9 mostra o conjunto Pareto obtido, no espaço dos objetivos. Observando-se essa figura, vemos que o conjunto Pareto obtido pelo SPEA 2 híbrido deixa algumas regiões do espaço sem cobertura. Isto pode ser explicado pela seleção para a busca local de apenas um ponto do arquivo de não-dominados. Portanto, se o arquivo possui baixa diversidade, o mesmo ocorrerá na próxima estimativa do conjunto de Pareto.

Tabela 5.3: Valor médio de cada métrica para o problema Kur com 3 variáveis, selecionando-se apenas um ponto para a busca local.

Métrica	SPEA 2 Puro	SPEA 2 Híbrido
NDCSR	0.4896	0.5104
Contagem de Esferas	25.7	30.4
S Metric	159.7094	165.5716

Com o objetivo de melhorar a distribuição (cobertura) do conjunto Pareto, no espaço dos objetivos, produzido pelo SPEA 2 híbrido, mais de um indivíduo foi selecionado para a busca local. Neste caso, quatro indivíduos foram selecionados e cada um produziu quatro novos indivíduos localmente melhorados. Os resultados para cada uma das métricas foram similares. Escolhendo-se, aleatoriamente, uma execução de cada algoritmo, a figura 5.10 mostra o conjunto Pareto, no espaço de objetivos. Podemos verificar uma melhora na distribuição no espaço comparando-se com a figura 5.9.

Finalmente, investigamos o problema Kur com 10 variáveis. Selecionando-se apenas um

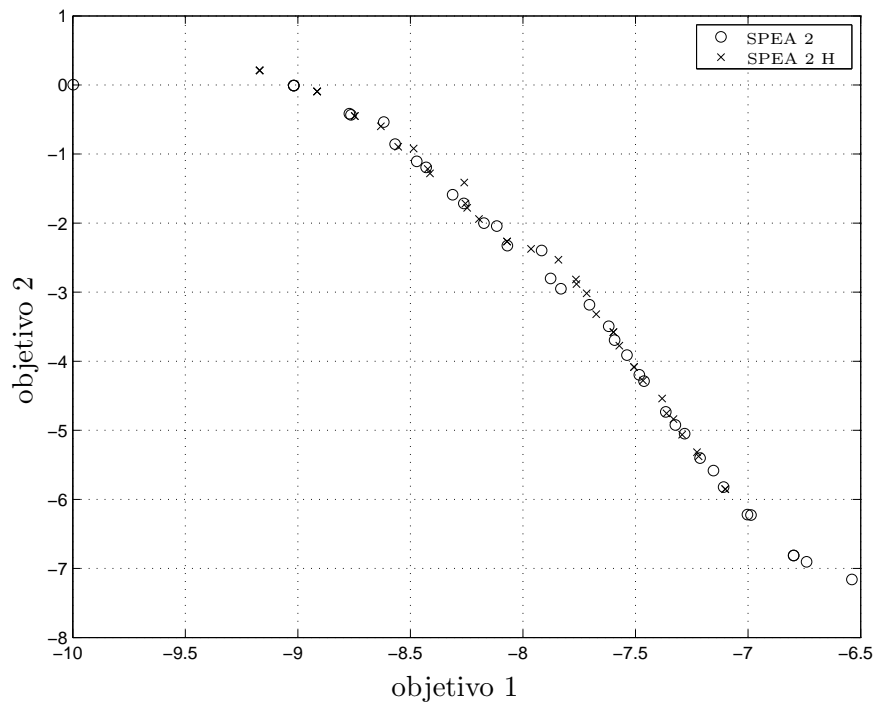


Figura 5.9: Conjunto Pareto, no espaço de objetivos, de uma execução aleatoriamente escolhida dos SPEA 2 e do SPEA 2 híbrido (SPEA H) no problema KUR com 3 variáveis. Apenas um ponto foi selecionado para a busca local no algoritmo híbrido.

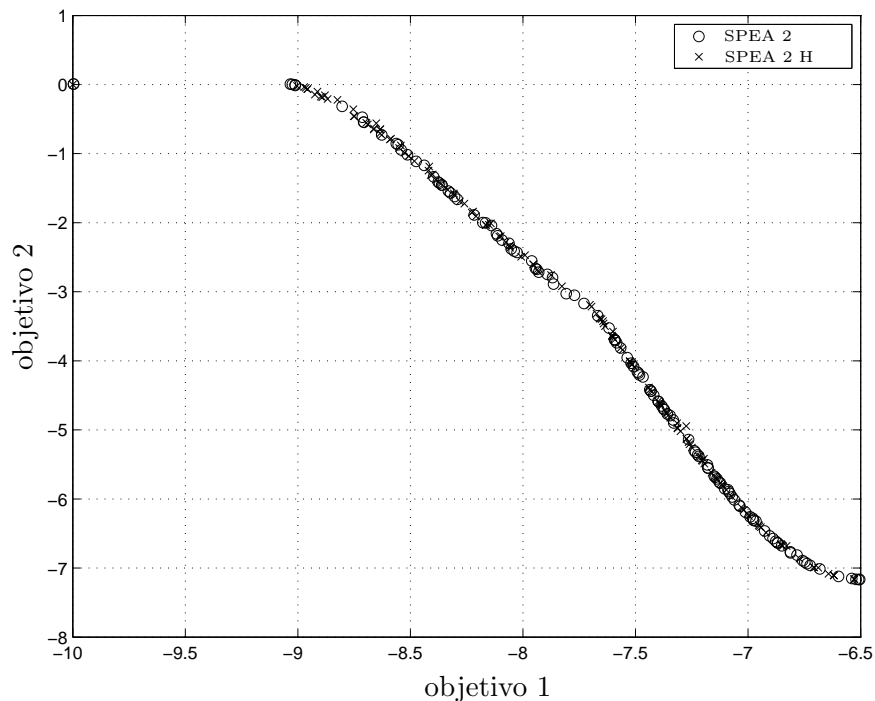


Figura 5.10: Conjunto Pareto, no espaço de objetivos, de uma execução aleatoriamente escolhida dos SPEA 2 e do SPEA 2 híbrido (SPEA H) no problema KUR com 3 variáveis. Quatro pontos foram selecionados para a busca local no algoritmo híbrido.



ponto para a busca local, a tabela 5.4 mostra os valores médios, após todas as execuções, para cada uma das métricas escolhidas. Podemos ver que o algoritmo híbrido apresenta um melhor desempenho em todas as métricas se comparado com o algoritmo puro. Selecionando-se quatro pontos para a busca local, a tabela 5.5 mostra os valores médios, após todas as execuções, para cada uma das métricas escolhidas. Novamente, podemos ver que o algoritmo híbrido apresenta um melhor desempenho em todas as métricas se comparado com o algoritmo puro.

Tabela 5.4: Valor médio de cada métrica para o problema Kur com 10 variáveis, selecionando-se apenas um ponto para a busca local.

Métrica	SPEA 2	SPEA 2 Híbrido
NDCSR	0.3258	0.6742
Contagem de Esferas	19.5	34.2
S Metric	340.9895	452.1711

Tabela 5.5: Valor médio de cada métrica para o problema Kur com 10 variáveis, selecionando-se quatro pontos para a busca local.

Métrica	SPEA 2	SPEA 2 Híbrido
NDCSR	0.3018	0.6982
Contagem de Esferas	25.1	52.7
S Metric	$2.13 \times 10^3$	$2.59 \times 10^3$

A figura 5.11 mostra o conjunto Pareto de uma execução de cada algoritmo escolhida aleatoriamente. Neste caso, quatro pontos foram selecionados para a busca local e cada ponto produziu quatro novos indivíduos melhorados.

#### 5.4.4 Conclusões

Esta seção apresentou um operador de busca local para problemas multiobjetivo irrestritos. Durante a fase de busca local, aproximações quadráticas para todas as funções-objetivo foram utilizadas para gerar um problema quadrático associado. Cada uma das soluções desse problema quadrático associado fornece um novo indivíduo localmente melhorado.

Esse operador de busca local foi acoplado com o SPEA 2 e testado com o problema Kursawe, com 3 e 10 variáveis. O algoritmo híbrido apresentou um desempenho superior à do algoritmo puro, levando-se em conta as métricas escolhidas. Apesar do desempenho

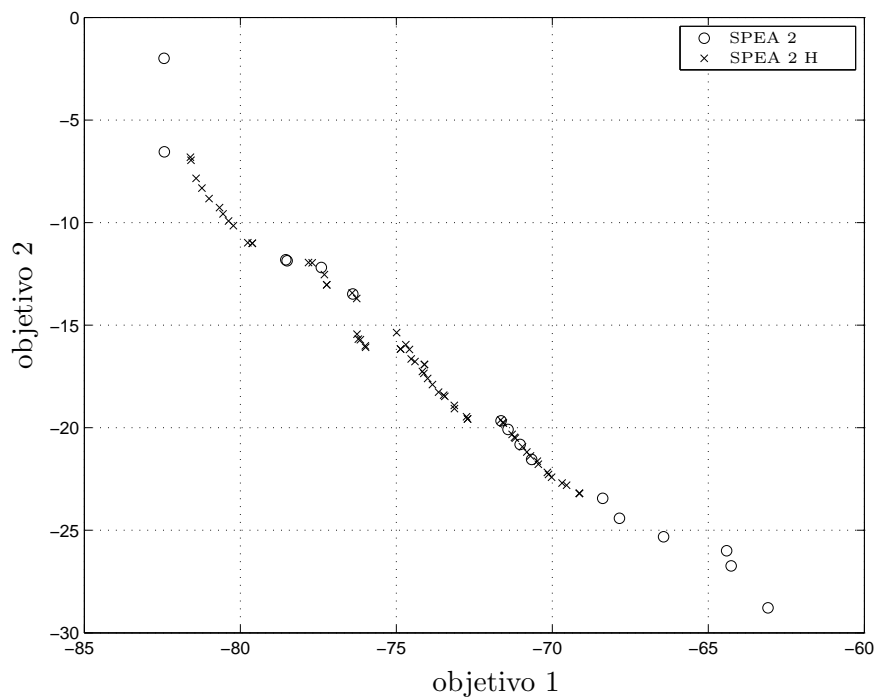


Figura 5.11: Conjunto Pareto, no espaço de objetivos, de uma execução aleatoriamente escolhida dos SPEA 2 e do SPEA 2 híbrido (SPEA H) no problema KUR com 10 variáveis. Quatro pontos foram selecionados para a busca local no algoritmo híbrido.

superior do algoritmo híbrido, ainda é necessário encontrar alguma técnica de nicho que, se utilizada juntamente com a busca local, proporcione uma melhor distribuição do conjunto Pareto-ótimo no espaço de objetivos.

## 5.5 Problemas Multiobjetivo Restritos

Apresentaremos nessa seção uma metodologia para o tratamento de problemas multiobjetivo não-lineares com restrições de desigualdade não-lineares<sup>3</sup>. Usaremos para tanto o princípio da construção de aproximações quadráticas para todas as funções e da construção de um problema mono-objetivo restrito auxiliar com o propósito de criar um operador especializado de busca local para problemas restritos. A formulação utilizada na construção desse problema auxiliar é baseado na metodologia conhecida como *goal attainment* para gerar soluções Pareto-ótimas através de problemas mono-objetivo [52]. Um operador específico para resolver esse problema auxiliar é baseado na técnica de *desigualdade linear matricial* (LMI), descrita no apêndice B, podendo ser resolvido usando o software SeDuMi.

Uma vantagem da metodologia aqui proposta é o tratamento mais preciso e eficiente de restrições em algoritmos evolucionários multiobjetivo. O operador de busca local proposto proporciona uma ferramenta que auxilia no tratamento de restrições e ainda melhora localmente a performance de algoritmos multiobjetivo. Nesse trabalho, a busca local com aproximações quadráticas é utilizada junto ao MOGA.

### 5.5.1 Metodologia

Considere o seguinte problema de otimização não-linear multiobjetivo com restrições não lineares de desigualdade:

$$\begin{aligned} x^* &= \min f(x) \\ \text{sujeito a: } &g(x) \leq 0 \end{aligned} \tag{5.26}$$

sendo que  $x \in \mathbb{R}^n$ . As funções  $f : \mathbb{R}^n \mapsto \mathbb{R}^m$  e  $g : \mathbb{R}^n \mapsto \mathbb{R}^p$  são reais e não-lineares.

Seja  $x_0$  o ponto selecionado para a busca local. As aproximações quadráticas para todas as funções envolvidas no problema são geradas usando a metodologia livre de derivada descrita na seção 2.3. Usaremos na construção dessas aproximações apenas os pontos (e as respectivas avaliações de funções) que pertencem a uma vizinhança de  $x_0$ . Essa vizinhança é novamente definida como

$$\mathcal{N}(x_0) = \{x : (x - x_0)^T R (x - x_0) \leq 1\} \tag{5.27}$$

---

<sup>3</sup>Os resultados que serão apresentados nessa seção encontram-se publicados em [51].

e

$$R_{ij} = \begin{cases} [0.1(u_i - l_i)]^{-1}, & i = j \\ 0, & i \neq j \end{cases} \quad (5.28)$$

sendo  $l_i$  e  $u_i$  respectivamente o valor mínimo e máximo para a  $i$ -ésima variável.

Com as aproximações quadráticas em mãos, um problema multiobjetivo aproximado, no qual todas as funções são quadráticas, pode ser construído.

As aproximações quadráticas para as funções objetivos  $f_i$ ,  $i = 1, \dots, m$ , e para as funções de restrições  $g_j$ ,  $j = 1, \dots, p$ , serão respectivamente denotadas por  $\tilde{f}_i$  e  $\tilde{g}_j$ . A busca local consiste em transformar o problema auxiliar multiobjetivo, definido pelas aproximações quadráticas, em um problema mono-objetivo com restrições adicionais. O seguinte problema restrito pode ser construído a partir do problema multiobjetivo (5.26):

$$(x^*, \gamma^*) = \arg \min_{x, \gamma} \gamma$$

$$\text{sujeito a: } \begin{cases} x \in \tilde{\mathcal{F}} \\ \gamma \in \mathbb{R}_+ \\ \tilde{f}(x) \leq u + \gamma w \end{cases} \quad (5.29)$$

sendo:

$$\tilde{\mathcal{F}} = \{x : \tilde{g}(x) \leq 0\}. \quad (5.30)$$

Esse problema mono-objetivo possui  $m + p$  restrições, sendo  $p$  restrições correspondentes às restrições de desigualdade e as outras  $m$  novas restrições correspondem às funções-objetivo do problema (5.26). O vetor  $u$  pode ser definido como a solução utópica, isto é, o vetor no qual cada coordenada representa o valor mínimo de cada função-objetivo calculado individualmente. Esse vetor pode ser obtido através das aproximações quadráticas. Cada componente da solução utópica é obtida resolvendo-se:

$$z_i^* = \arg \min \tilde{f}_i(x) \in \mathbb{R} \quad (5.31)$$

$$\text{sujeito a: } x \in \tilde{\mathcal{F}}$$

Assim, cada componente do vetor  $u$  é dada por:

$$u_i = \tilde{f}_i(z_i^*) \quad (5.32)$$

Seja  $U_i$  o vetor de objetivos associado com cada  $z_i$ , o qual é obtido com o procedimento acima. Seja  $\mathcal{C}$  o cone gerado pelo vetor  $U_i - u$  sendo  $u$  a origem deste cone. Sabemos que

todas as soluções factíveis do problema auxiliar (5.29) estão situadas no interior desse cone. O vetor  $w$  fornece uma direção no espaço de objetivos e também pertence a esse cone  $\mathcal{C}$ . Desta forma, o vetor  $w$  pode ser encontrado pela fórmula:

$$w = u + \sum_i \alpha_i (U_i - u) \quad (5.33)$$

para  $\alpha_i > 0$

O problema auxiliar (5.29) pode ser resolvido através da metodologia descrita no apêndice B, uma vez que todas as restrições são quadráticas e a função-objetivo é linear. Desta forma, o problema auxiliar (5.29), pode ser re-escrito na forma:

$$x^* = \arg_x \min_{x, \gamma} \gamma$$

$$\text{s.a.} \quad \begin{cases} \begin{bmatrix} 1 & (x - x_i)^T \\ x - x_i & Q_i^{-1} \end{bmatrix} \geq 0, & i = 1, \dots, p \\ \begin{bmatrix} c_i + u_i + \gamma w_i & (x - x_i)^T \\ x - x_i & Q_i^{-1} \end{bmatrix} \geq 0, & i = 1, \dots, m \end{cases} \quad (5.34)$$

As primeiras  $p$  LMIs são referentes às  $p$  restrições de desigualdade não-lineares do problema original. As  $m$  LMIs restantes são referentes às funções-objetivo do problema original.

A figura 5.12 mostra um esquema do operador de busca local proposto nessa seção.

Considere um exemplo simples para ilustrar geometricamente a metodologia. Suponha um problema multiobjetivo irrestrito com três funções-objetivo quadráticas, cujas matrizes Hessianas correspondem à matriz identidade. Assim, todas as soluções Pareto-ótimas se encontram no triângulo mostrado na figura 5.5.1 e a solução utópica é  $u_i = 0$  for  $i = 1, 2, 3$ . Portanto:

$$(x^*, \gamma^*) = \arg \min_{x, \gamma} \gamma$$

$$\text{sujeito a:} \quad \begin{cases} f_1(x) \leq \gamma w_1 \\ f_2(x) \leq \gamma w_2 \\ f_3(x) \leq \gamma w_3 \end{cases} \quad (5.35)$$

Suponha que um valor inicial de  $\gamma$  forneça a situação ilustrada na figura 5.5.1-(A), em cujos pontos no interior do círculo temos  $f_i(x) \leq \gamma w_i$ . Existe uma interseção entre as regiões factíveis para cada restrição, então o valor de  $\gamma$  pode ser diminuído até que uma situação

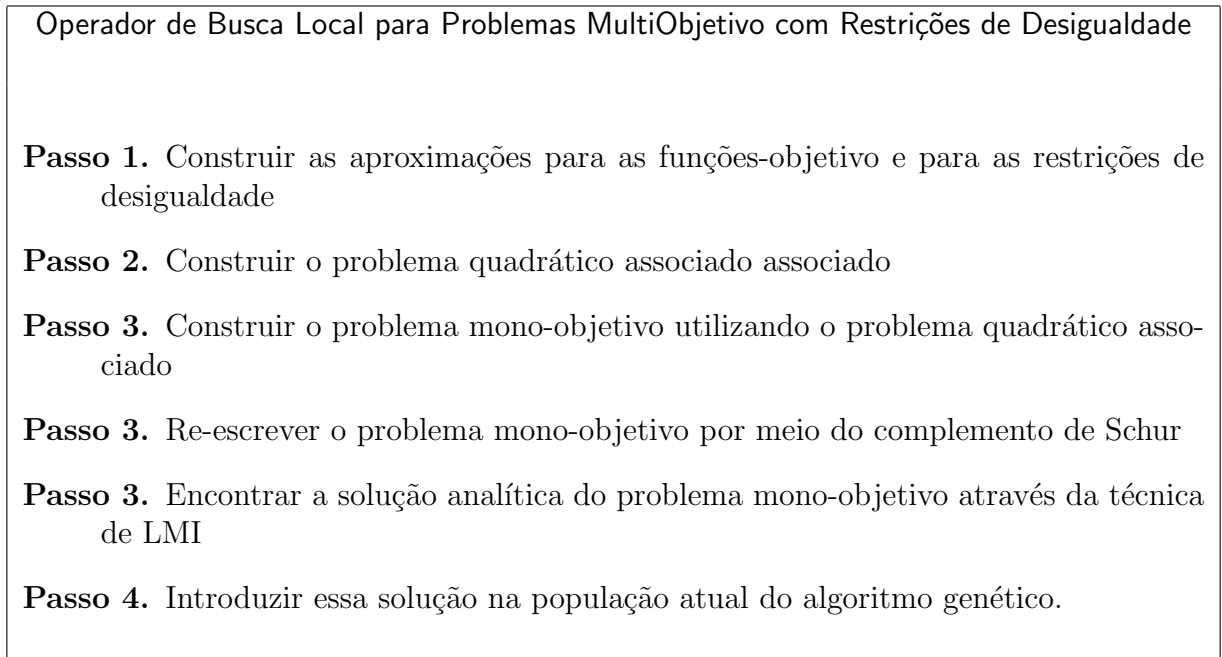


Figura 5.12: Esquema básico do operador de busca local para problemas multiobjetivo restritos.

como a ilustrada em 5.5.1-(B) ocorra. Observe que não é mais possível diminuir o valor de  $\gamma$  sem tornar o problema infactível. Portanto, a formulação em (5.29) minimiza o valor de  $\gamma$  até que um ponto limite seja atingido, o qual corresponde ao valor de  $\gamma$  que faz com que o conjunto factível de todas as restrições tenha medida zero. Variando-se o valor para  $w_i$ , diferentes pontos no interior do triângulo são obtidos.

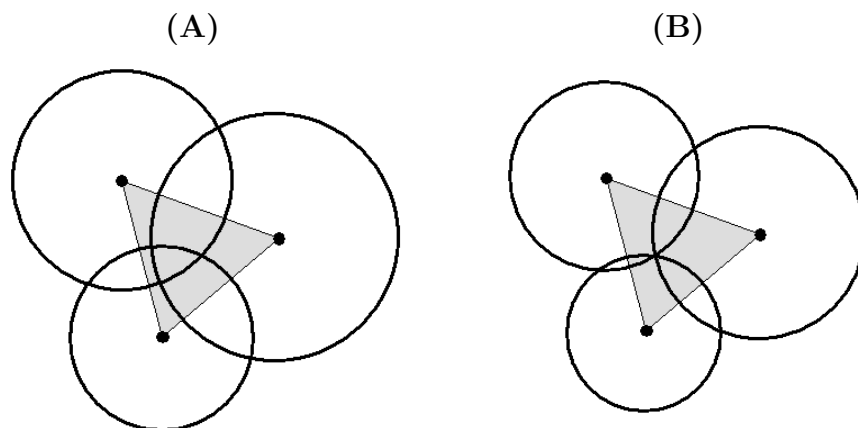


Figura 5.13: Esta figura ilustra o procedimento de busca local em um problema tri-objetivo quadrático e irrestrito.

### 5.5.2 Acoplando a Busca Local no MOGA

Para acoplar a busca local no MOGA é necessário definir quais e quantos indivíduos serão selecionados para a busca local e quantos indivíduos melhorados serão gerados para cada indivíduo selecionado.

Um arquivo com todos as soluções não-dominadas até o momento será criado no MOGA. Esse arquivo representa então a estimativa atual do conjunto Pareto-ótimo. O operador de busca local atuará nesse conjunto.

Vamos selecionar para a busca local  $\mu$  pontos desse arquivo. Após essa seleção, as aproximações quadráticas serão calculadas em torno de cada indivíduo selecionado. Finalmente, o operador de busca local será utilizado para gerar  $\lambda$  novas soluções, através da variação do parâmetro  $w_i$ .

Os  $\mu \times \lambda$  novos indivíduos devem ser avaliados nas funções originais para então introduzi-los na população. Desta forma, o custo total desse procedimento, em termos de avaliações de funções, é dado por  $\mu \times \lambda$ .

A escolha dos parâmetros  $\mu$  e  $\lambda$  deve levar em conta o balanço desejado entre a busca local e a busca global. Se um valor grande de  $\mu$  é escolhido, a busca local será mais intensa e efetuada em diferentes regiões da estimativa do conjunto de Pareto. Entretanto, esse procedimento acarretará em um número maior de avaliações de funções.

### 5.5.3 Testes e Resultados

O MOGA e o MOGA híbrido (com a busca local proposta aqui) foram testados com alguns problemas multiobjetivos analíticos. Os problemas escolhidos foram:

- Problema Quadrático

$$\min \begin{cases} f_1(x) = x_1^2 + x_2^2 \\ f_2(x) = x_1^2 + (x_2 - 4)^2 \end{cases}$$

$$\text{sujeito a: } \begin{cases} g_1(x) = (x_1)^2 + (x_2 - 3)^2 - 1.5^2 \leq 0 \\ g_2(x) = (x_1 - 1)^2 + (x_2 - 1)^2 - 1.5^2 \leq 0 \\ g_3(x) = (x_1 + 1)^2 + (x_2 - 1)^2 - 1.5^2 \leq 0 \\ -5 \leq x_1 \leq 5 \\ -3 \leq x_2 \leq 7 \end{cases} \quad (5.36)$$

- Problema Não-Quadrático

$$\min \begin{cases} f_1(x) = x_1^2 + x_2^2 \\ f_2(x) = x_1^2 + (x_2 - 4)^2 \\ f_3(x) = (x_1 - 2)^4 + (x_1 - 2x_2)^2 \end{cases}$$

$$\text{sujeito a: } \begin{cases} g_1(x) = (x_1)^2 + (x_2 - 3)^2 - 1.5^2 \leq 0 \\ g_2(x) = (x_1 - 1)^2 + (x_2 - 1)^2 - 1.5^2 \leq 0 \\ g_3(x) = (x_1 + 1)^2 + (x_2 - 1)^2 - 1.5^2 \leq 0 \\ -5 \leq x_1 \leq 5 \\ -3 \leq x_2 \leq 7 \end{cases} \quad (5.37)$$

- Problema KUR

$$\min \begin{cases} f_1(x) = \sum_{i=1}^2 (-10 \cdot \exp(0.2 \sqrt{x_i^2 + x_{i+1}^2})) \\ f_2(x) = \sum_{i=1}^3 (\text{abs}(x_i)^{0.8} + 5 \sin(x_i^3)) \end{cases} \quad (5.38)$$

$$-5 \leq x_i \leq 5$$

- Problema Multimodal

$$\min \begin{cases} f_1(x) = x^T \cdot A^T \cdot A \cdot x - 10 \cdot [1 \ 1] \cdot \cos(2\pi Ax) \\ f_2(x) = 0.01 [\sum_{i=1}^2 ((x_i + 0.5)^4 - 30x_i^2 - 20x_i)] \end{cases} \quad (5.39)$$

$$-6 \leq x_i \leq 6$$

sendo

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$$

O objetivo dos testes é avaliar o desempenho do algoritmo híbrido e compará-la ao desempenho do algoritmo puro. Os dois algoritmos, MOGA e MOGA híbrido (MOGA-H), possuíam a mesma configuração: 50 indivíduos para os problemas bi-dimensionais e 60 indivíduos para o problema tri-dimensional. Cada algoritmo foi executado 20 vezes em cada problema, com 30 gerações para os problemas (5.36), (5.38) e (5.39) e com 50 gerações para o problema (5.37). O máximo número de gerações foi o único critério de parada utilizado.



Em cada problema, 5 indivíduos foram selecionados do arquivo de soluções não-dominadas e cada indivíduo produziu 4 novos indivíduos a cada geração.

As soluções não-dominadas foram extraídas do conjunto de soluções não-dominadas produzidos pelo MOGA-H e pelo MOGA em cada uma das 20 execuções. Os resultados, mostrados nas figuras 5.14, 5.15, 5.16 e 5.17, indicam que o número de soluções não-dominadas produzidas pelo MOGA-H é maior se comparado ao MOGA.

As tabelas 5.6, 5.8 e 5.9 mostram o valor médio, após os 20 testes com cada problema, da contagem de esferas e da S Metric para o MOGA e o MOGA Híbrido. Para o problema Não-Quadrático, a tabela 5.7 apresenta apenas o valor médio da contagem de esferas, uma vez que o problema possui três funções-objetivo. Nesses testes, o raio da esfera necessário para se efetuar a contagem de esferas não foi o mesmo para todos os problemas. Para determinar o valor ideal do raio, o seguinte procedimento foi adotado:

- escolher um raio inicial  $r_0$  e efetuar a contagem de esferas para a amostra  $C_0$ ;
- dividir o raio inicial pela metade obtendo um novo raio  $r_1$  e efetuar novamente a contagem de esferas para a amostra  $C_1$ ;
- continuar dividindo o raio pela metade obtendo sempre um novo raio  $r_i$  até que a diferença  $C_i - C_{i-1}$  não seja mais significativa. Esse valor de  $r_i$  será o raio ideal para o determinado problema.

Tabela 5.6: Valor médio, após 20 testes, do contador de esferas e da S Metric para o problema Quadrático (prob. 5.36).

Algoritmo	Métrica	Valor
Moga	Contador de Esferas	455.0334
	S Metric	6.2203
Moga Híbrido	Contador de Esferas	712.8889
	S Metric	9.5567

Tabela 5.7: Valor médio, após 20 testes, do contador de esferas e da S Metric para o problema Não-Quadrático (prob. 5.37).

Algoritmo	Métrica	Valor
Moga	Contador de Esferas	342.0056
Moga Híbrido	Contador de Esferas	678.2311

As figuras 5.18, 5.19, 5.20, e 5.21 mostram os resultados estatísticos usando a contagem de esferas. Uma vez que a contagem de esferas nos fornece uma medida da distribuição do

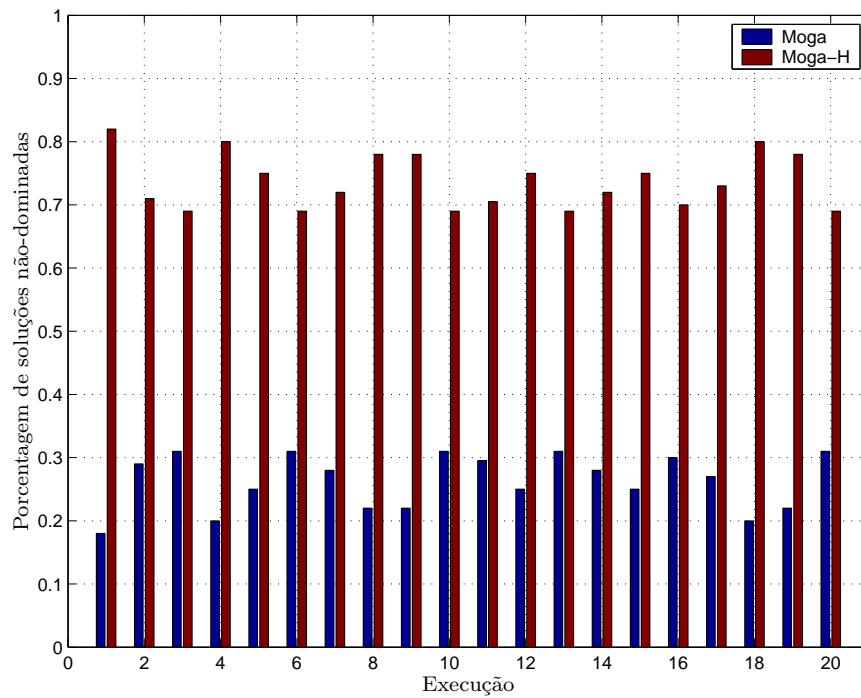


Figura 5.14: Porcentagem de soluções não-dominadas em cada execução do problema Quadrático (prob. 5.36). A primeira barra corresponde ao MOGA puro e a segunda barra corresponde ao MOGA híbrido.

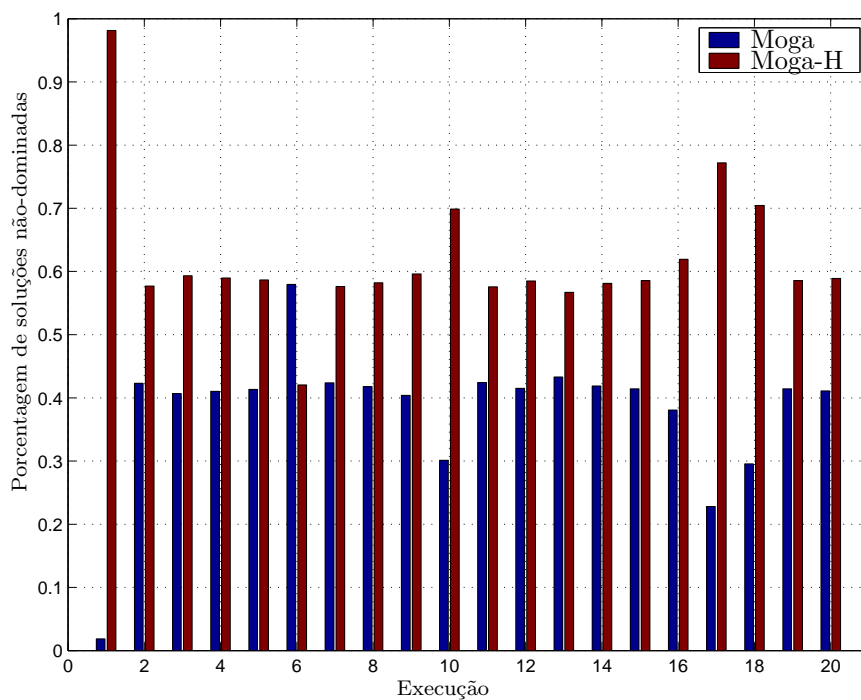


Figura 5.15: Porcentagem de soluções não-dominadas em cada execução do problema Não-Quadrático (prob. 5.37). A primeira barra corresponde ao MOGA puro e a segunda barra corresponde ao MOGA híbrido.

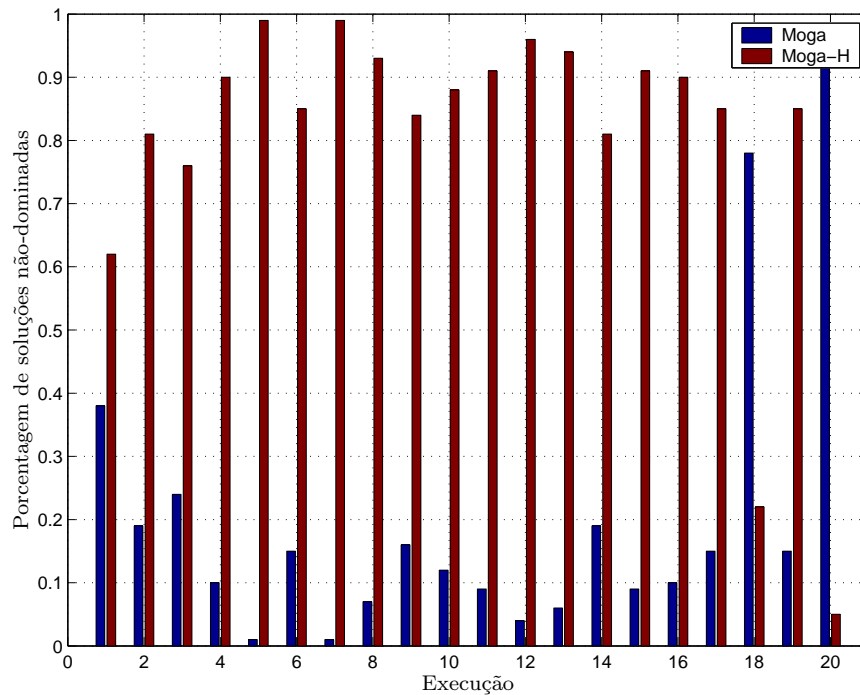


Figura 5.16: Porcentagem de soluções não-dominadas em cada execução do problema KUR (prob. 5.38). A primeira barra corresponde ao MOGA puro e a segunda barra corresponde ao MOGA híbrido.

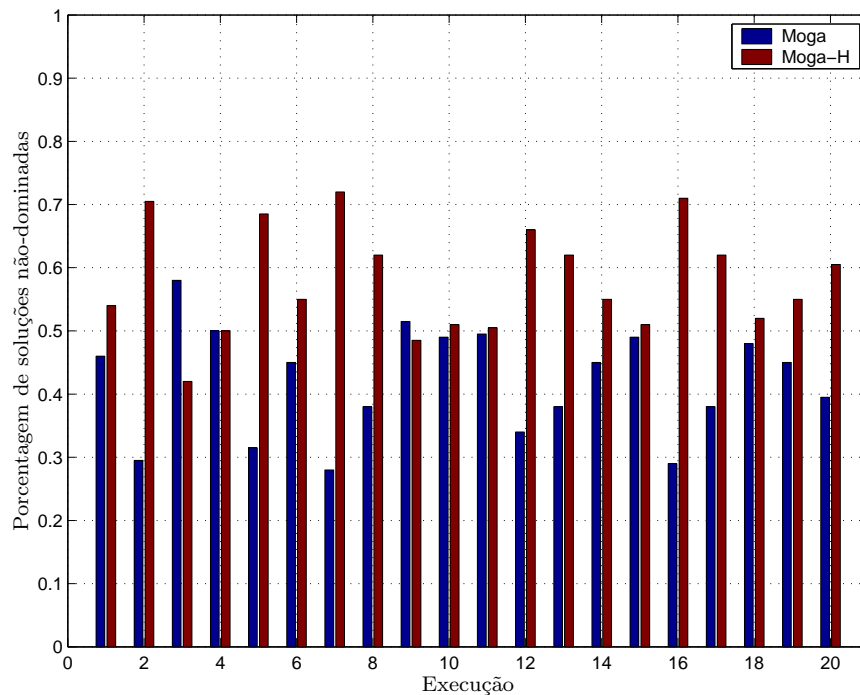


Figura 5.17: Porcentagem de soluções não-dominadas em cada execução do problema Multimodal (prob. 5.39). A primeira barra corresponde ao MOGA puro e a segunda barra corresponde ao MOGA híbrido.

Tabela 5.8: Valor médio, após 20 testes, do contador de esferas e da S Metric para o problema KUR (prob. 5.38).

Algoritmo	Métrica	Valor
Moga	Contador de Esferas	26.6000
	S Metric	$3.4223 \times 10^6$
Moga Híbrido	Contador de Esferas	57.5000
	S Metric	$6.4476 \times 10^7$

Tabela 5.9: Valor médio, após 20 testes, do contador de esferas e da S Metric para o problema Multimodal (prob. 5.39).

Algoritmo	Métrica	Valor
Moga	Contador de Esferas	57.5000
	S Metric	2041.3456
Moga Híbrido	Contador de Esferas	111.2000
	S Metric	2041.4522

conjunto Pareto, podemos ver que o MOGA Híbrido supera o MOGA nesse aspecto nos problemas 5.36, 5.37, 5.38 e 5.39. Os resultados reais observados, destacados com o círculo preto, são estatisticamente significantes.

As figuras 5.22, 5.23, e 5.24 mostram os resultados estatísticos usando a S Metric. Uma vez que o problema (5.37) tem 3 funções-objetivo, a S Metric não foi calculada. Nos problemas (5.36) e (5.38), podemos observar que o MOGA Híbrido supera o MOGA, no que diz respeito ao volume da região dominada pelo conjunto de estimativas obtido pelo algoritmo. Os resultados reais observados, destacados com círculos pretos, são estatisticamente significantes nesses dois problemas. Por outro lado, para o problema (5.39), o resultado estatístico, ilustrado na figura 5.24, mostra que os dois algoritmos apresentam resultados similares para essa métrica.

A S Metric é influenciada pela presença de pontos extremos ou isolados, bem como pela presença de regiões não-conexas no espaço de objetivos. Podemos observar esse fato, ao analisarmos o histograma mostrado na figura 5.23. As figuras 5.25 e 5.26 mostram as soluções ótimas no espaço de objetivos para as execuções do MOGA e MOGA Híbrido que apresentaram o melhor valor da S Metric. É possível ver que o conjunto Pareto-ótimo possui ponto isolado e regiões desconexas, explicando assim a forma não-usual do histograma mostrado na figura 5.23.

Com o objetivo de analisar o resultado estatístico obtido no histograma da figura 5.24, o conjunto Pareto-ótimo para cada algoritmo foi obtido. As figuras 5.27 e 5.28 mostram o conjunto Pareto-ótimo para as execuções de cada algoritmo que obtiveram o melhor valor da contagem de esferas. Ao observar as figuras 5.27 e 5.28, o conjunto Pareto obtido pelo

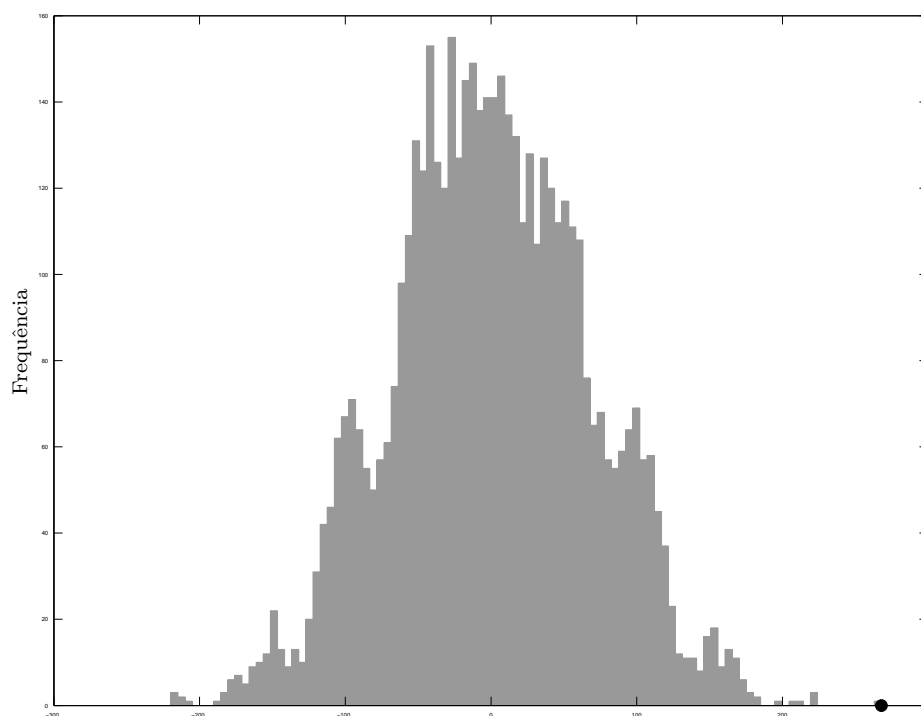


Figura 5.18: Teste de aleatoriedade, usando a contagem de esferas, para o problema Quadrático (prob. 5.36). O resultado é estatisticamente significativo uma vez que o círculo preto se encontra no lado positivo do histograma.

MOGA puro apresenta uma maior distribuição no espaço do que o Pareto obtido pelo MOGA Híbrido. Uma questão que então surge é se o raio escolhido para esse problema não teve uma influência negativa na distribuição do conjunto Pareto-ótimo, ou seja, o raio foi tão pequeno que em uma determinada região (mais densa) a contagem de esferas superou em muito a ausência de pontos em outras regiões do espaço. Com o objetivo de verificar esse fato, um novo raio para a contagem de esferas foi escolhido. Esse raio foi escolhido como sendo  $\frac{1}{50}$  do intervalo total que representa o eixo  $x_1$ . As figuras 5.29 e 5.30 mostram o conjunto Pareto-ótimo para as execuções de cada algoritmo que obtiveram o melhor valor da contagem de esferas para esse novo valor do raio. Podemos observar que o conjunto Pareto-ótimo obtido pelo MOGA Híbrido, utilizando-se o novo raio (figura 5.30), apresentou uma melhor distribuição no espaço se comparado ao conjunto Pareto-ótimo obtido pelo MOGA Híbrido utilizando-se o raio inicialmente escolhido (figura 5.28).

#### 5.5.4 Conclusões

Essa seção apresentou uma metodologia de busca local para problemas multiobjetivo não-lineares com restrições de desigualdades não-lineares. Essa metodologia de busca local

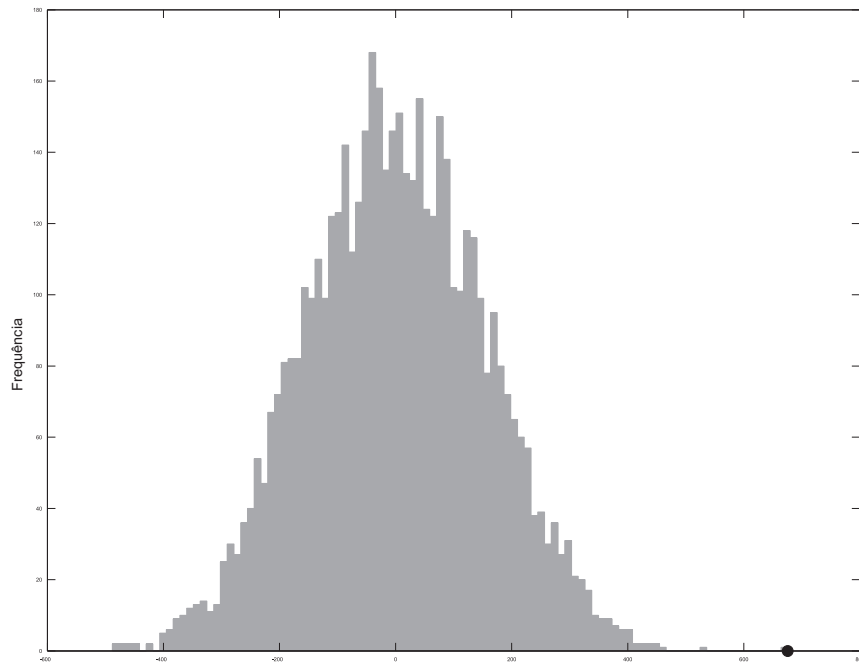


Figura 5.19: Teste de aleatoriedade, usando a contagem de esferas, para o problema Não-Quadrático (prob. 5.37). O resultado é estatisticamente significativo uma vez que o círculo preto se encontra no lado positivo do histograma.

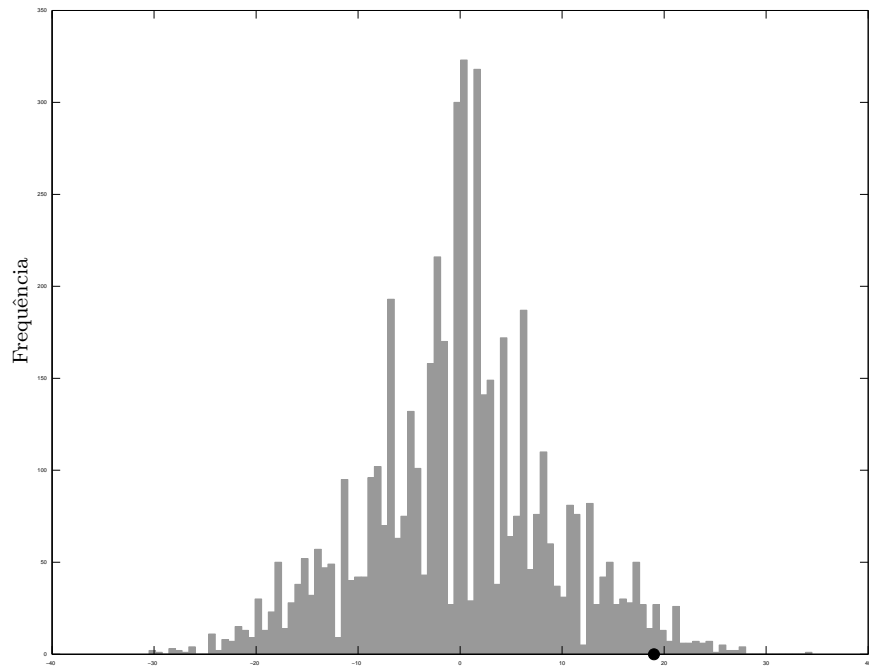


Figura 5.20: Teste de aleatoriedade, usando a contagem de esferas, para o problema KUR (prob. 5.38). O resultado é estatisticamente significativo uma vez que o círculo preto se encontra no lado positivo do histograma.

utiliza aproximações quadráticas para todas as funções envolvidas no problema para contruir um problema quadrático associado. O problema quadrático é então resolvido usando LMI e a solução fornece um novo indivíduo.

A metodologia aqui proposta permite a combinação de operadores de busca local com técnicas evolucionárias multiobjetivo sem nenhum cálculo adicional de função, uma vez que as aproximações quadráticas são construídas utilizando informações disponíveis para o algoritmo. Portanto, essa metodologia é adequada a problemas nos quais a avaliação das funções acarreta um elevado custo computacional (funções *black-box*).

Ainda nessa seção, uma nova métrica para comparação de algoritmos foi apresentada: a contagem de esferas. Essa métrica, que não necessita do conhecimento prévio do conjunto Pareto-ótimo real, é um procedimento simples que nos fornece uma medida de distribuição das soluções no espaço de objetivos. Apesar da simplicidade, esta métrica também é sensível à escolha de um determinado parâmetro, o raio das esferas. Novos experimentos estão sendo feitos com o propósito de verificar a sensibilidade dessa métrica em relação à esse parâmetro.

Os resultados sugerem que o procedimento de busca local aqui proposto pode ser incluído em qualquer AG quando aplicado a problemas multiobjetivo genéricos restritos cujas as funções envolvidas sejam não-lineares. Em particular, a otimização de funções do tipo *black-box* pode ser beneficiada com a utilização dessa metodologia.

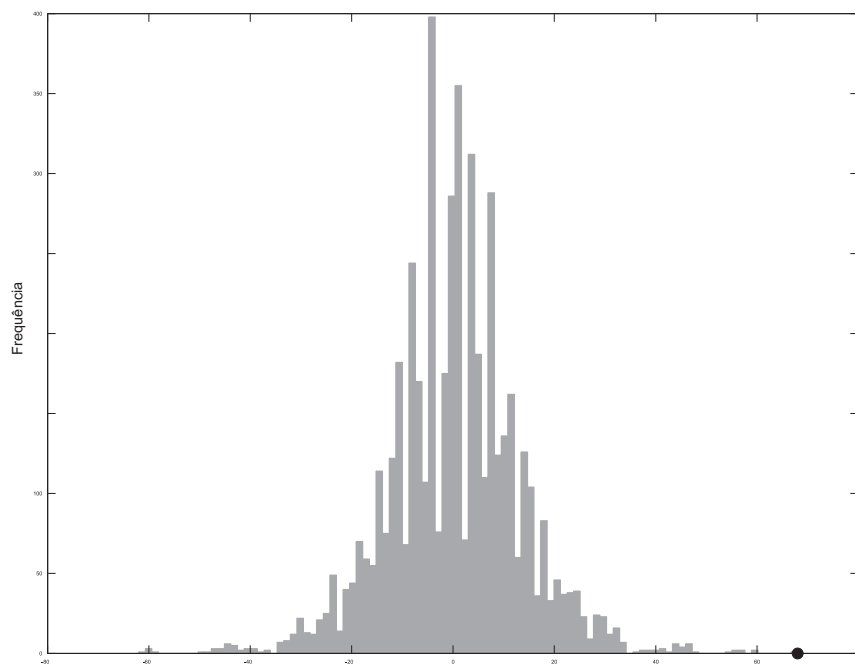


Figura 5.21: Teste de aleatoriedade, usando a contagem de esferas, para o problema Multimodal (prob. 5.39). O resultado é estatisticamente significativo uma vez que o círculo preto se encontra no lado positivo do histograma.

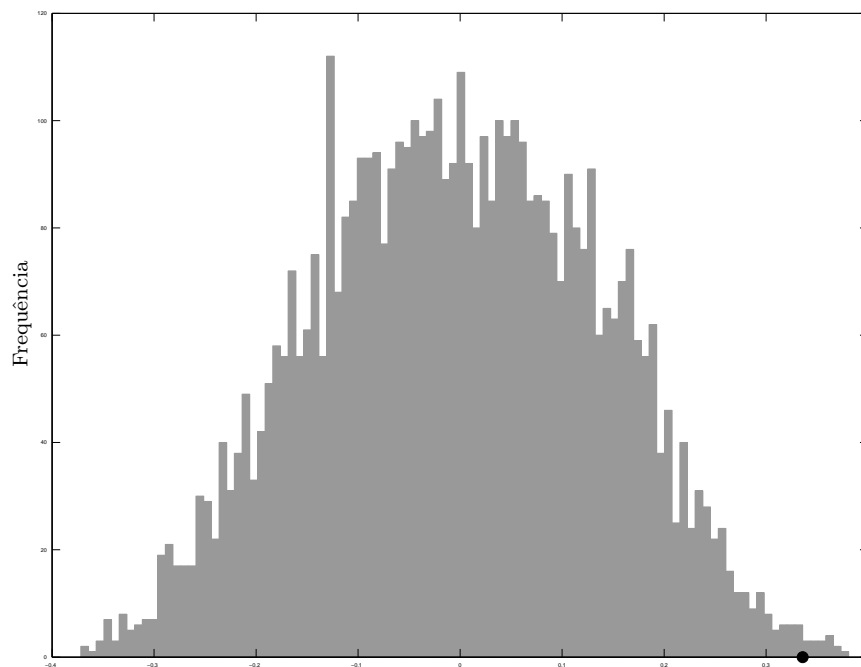


Figura 5.22: Teste de aleatoriedade, usando a S Metric, para o problema Quadrático (prob. 5.36). O resultado é estatisticamente significativo uma vez que o círculo preto se encontra no lado positivo do histograma.



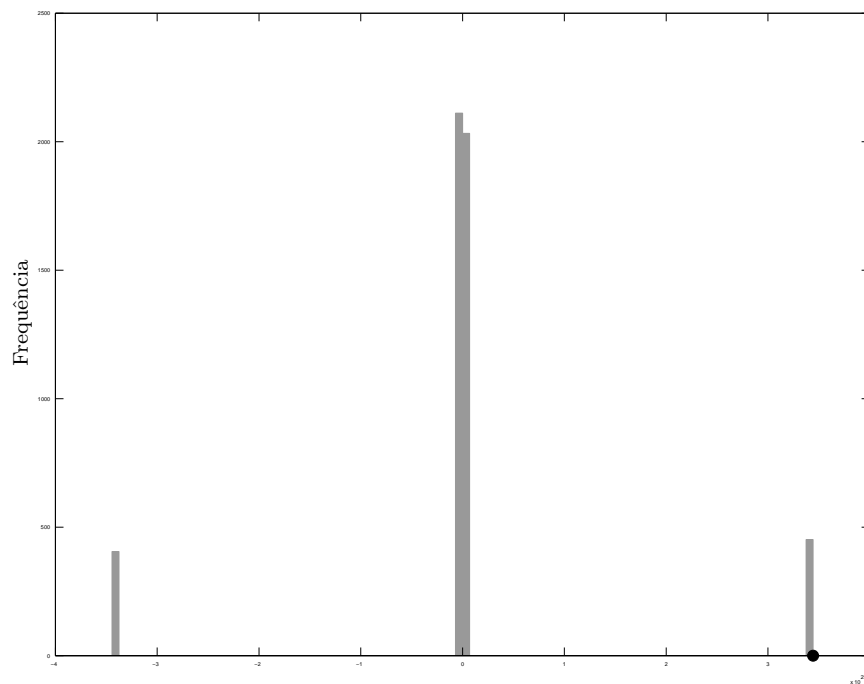


Figura 5.23: Teste de aleatoriedade, usando a S Metric, para o problema KUR (prob. 5.38). O resultado é estatisticamente significativo uma vez que o círculo preto se encontra no lado positivo do histograma.

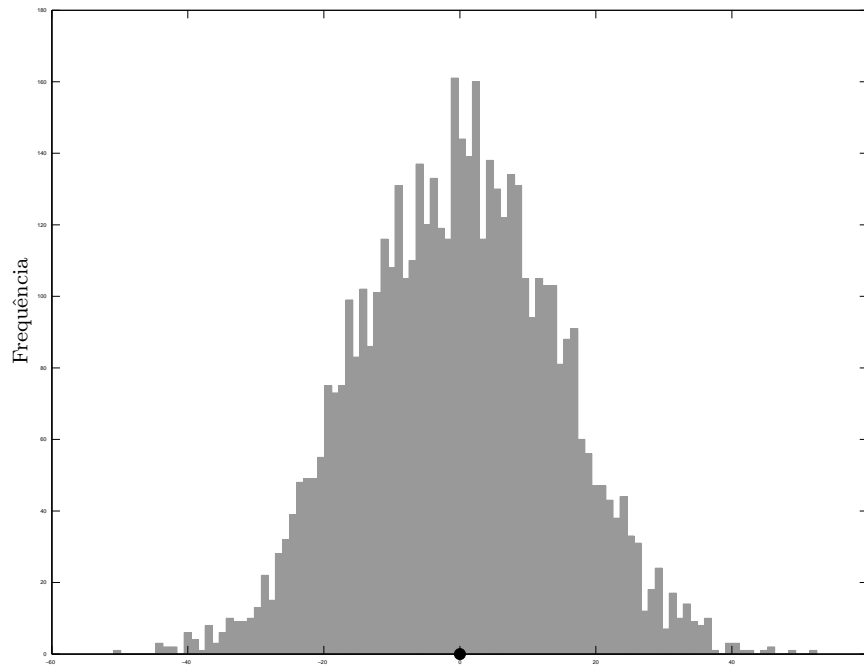


Figura 5.24: Teste de aleatoriedade, usando a S Metric, para o problema Multimodal (prob. 5.39). O resultado não é estatisticamente significativo uma vez que o círculo preto se encontra no meio do histograma.

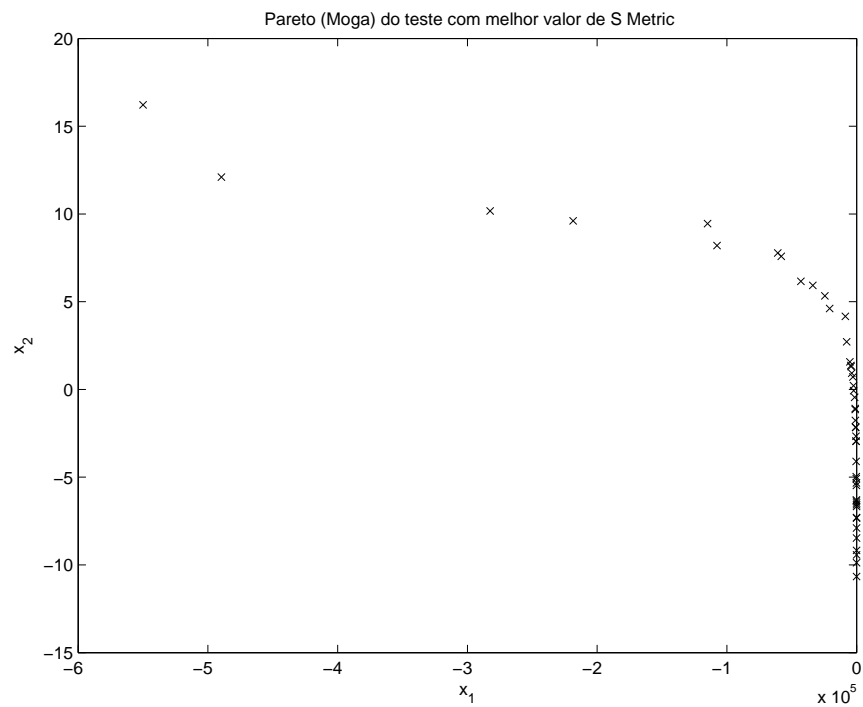


Figura 5.25: Conjunto Pareto-ótimo da execução do MOGA que apresentou melhor valor de S Metric no problema KUR (prob. 5.38). Observe a presença de ponto isolado e regiões desconexas.

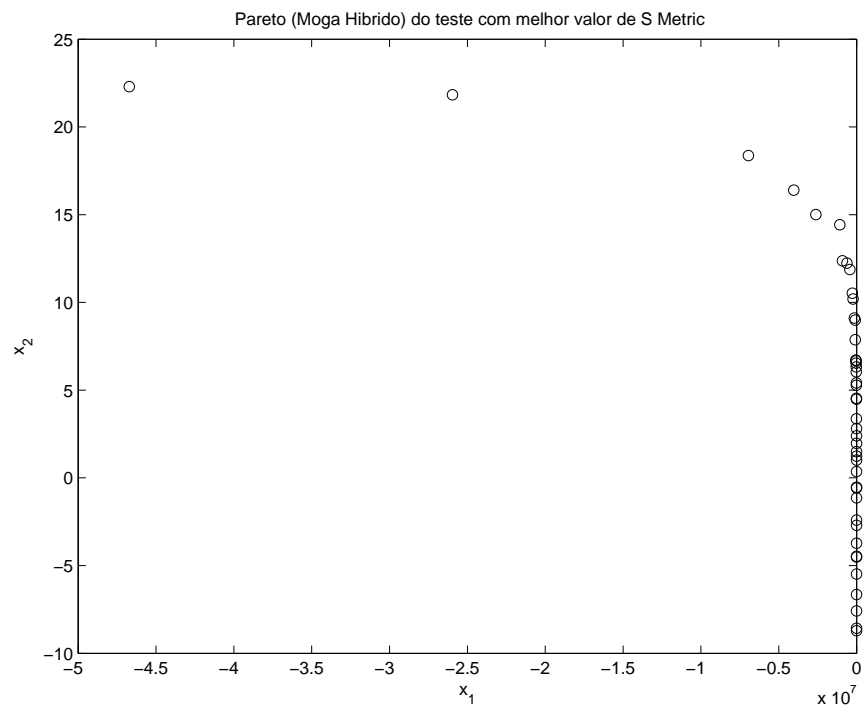


Figura 5.26: Conjunto Pareto-ótimo da execução do MOGA Híbrido que apresentou melhor valor de S Metric no problema KUR (prob. 5.38). Observe a presença de ponto isolado e regiões desconexas.

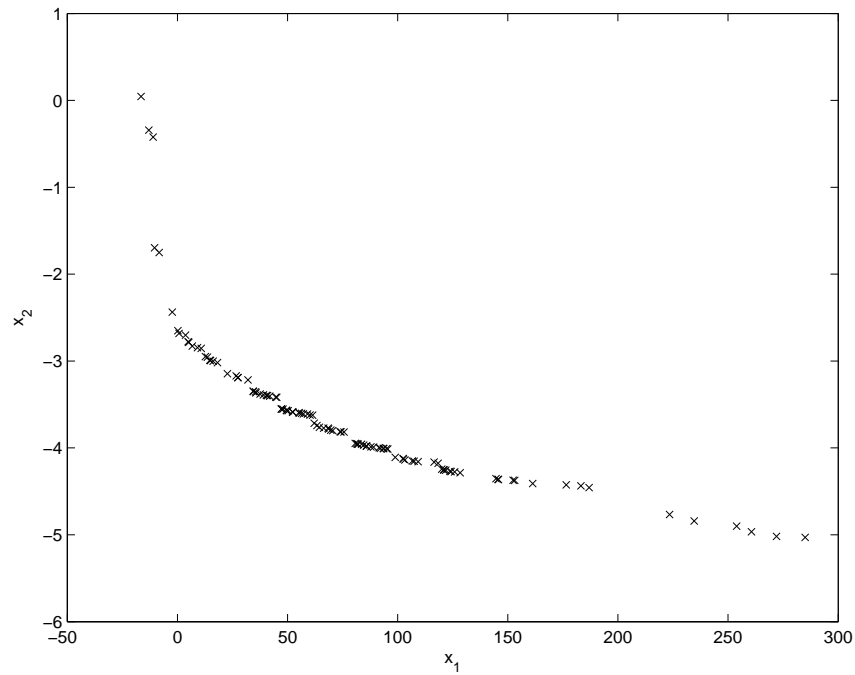


Figura 5.27: Conjunto Pareto-ótimo da execução do MOGA que apresentou melhor valor da contagem de esferas no problema Multimodal (prob. 5.39).

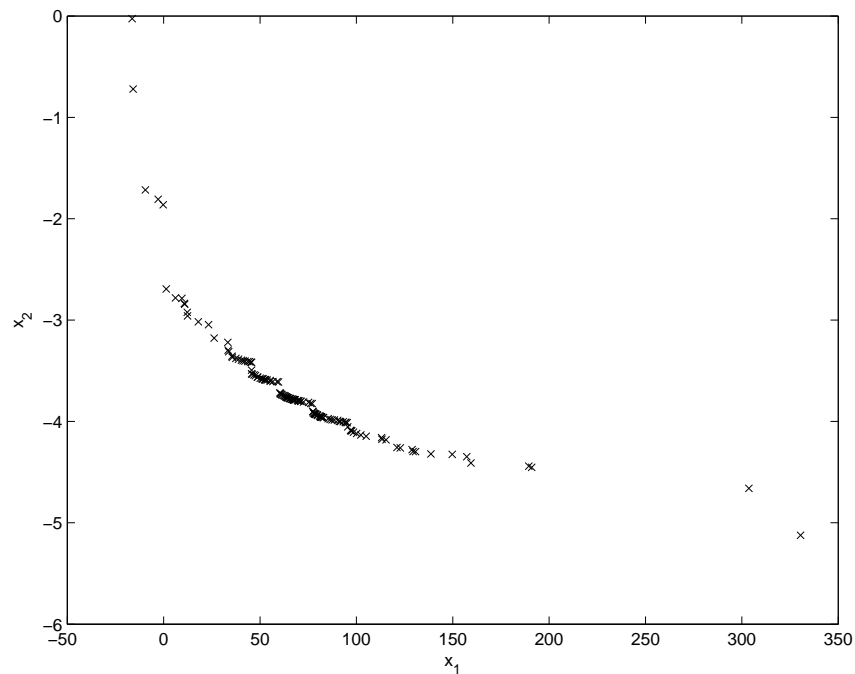


Figura 5.28: Conjunto Pareto-ótimo da execução do MOGA Híbrido que apresentou melhor valor da contagem de esferas no problema Multimodal (prob. 5.39).

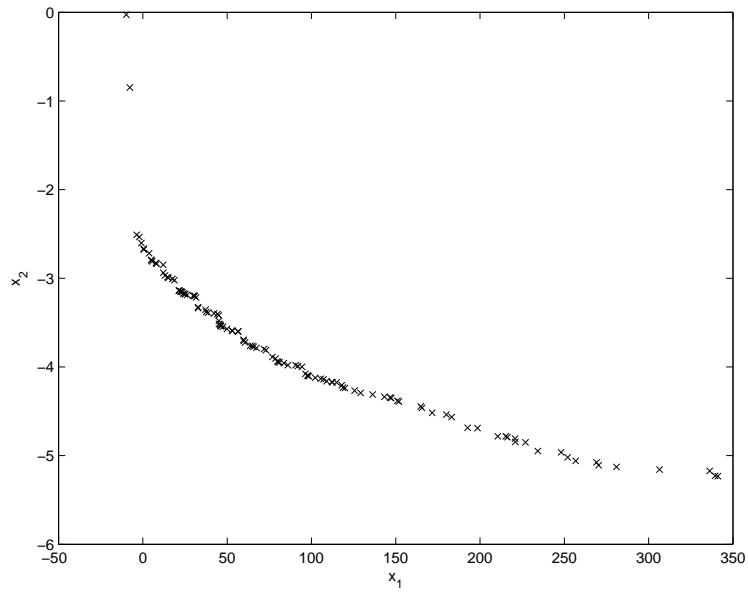


Figura 5.29: Conjunto Pareto-ótimo da execução do MOGA que apresentou melhor valor da contagem de esferas no problema Multimodal (prob. 5.39) utilizando-se o novo raio para as esferas.

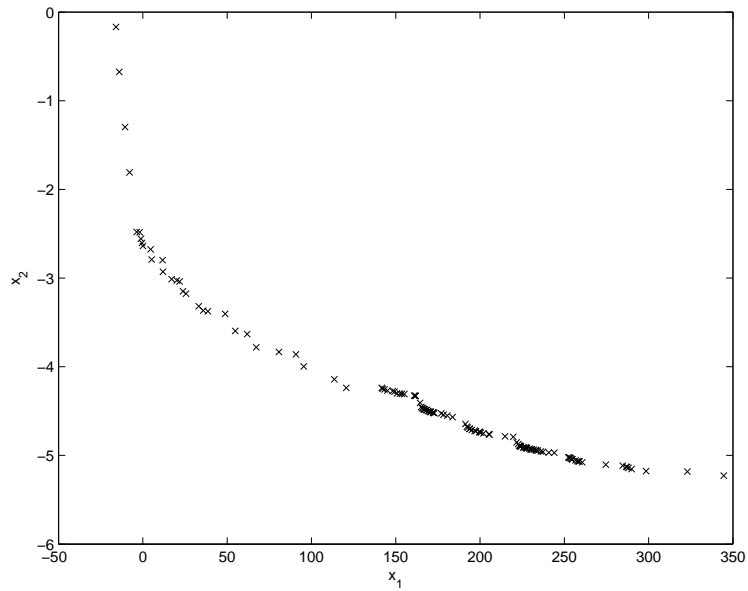


Figura 5.30: Conjunto Pareto-ótimo da execução do MOGA Híbrido que apresentou melhor valor da contagem de esferas no problema Multimodal (prob. 5.39) utilizando-se o novo raio para as esferas.

# Capítulo 6

## Conclusões

Este trabalho apresentou uma metodologia de construção de operadores, baseados em aproximações quadráticas para as funções de um problema de otimização (mono ou multiobjetivo) para serem acoplados a algoritmos genéticos<sup>1</sup> com o objetivo de melhorar a convergência e a precisão do algoritmo. As aproximações quadráticas são construídas utilizando apenas as informações normalmente tornadas disponíveis através da própria execução do AG: as coordenadas dos pontos e suas respectivas avaliações. Não são necessárias, portanto, avaliações extras de função.

As aproximações quadráticas podem ser usadas no AG com duas funções distintas. Primeiro, podemos usá-las em problemas mono-objetivo para corrigir as coordenadas do sistema obtendo novas coordenadas mais favoráveis ao processo de otimização. Os resultados obtidos sugerem que essa mudança de coordenada é capaz de melhorar tanto a proporção de vezes nas quais o AG encontra o mínimo quanto o número de avaliações da função-objetivo necessárias para achar tal ponto.

A segunda função das aproximações quadráticas consiste em obter estimativas, iterativamente, para as coordenadas do ponto factível de mínimo do problema. À medida em que o algoritmo evolui, a estimativa da solução tenderá ao ponto de ótimo (ou ao conjunto Pareto-ótimo) do problema, tanto devido à evolução do próprio algoritmo independente do novo operador quanto devido às estimativas mais refinadas produzidas pela aproximação quadrática, num processo dinâmico produzido pela interação desses dois processos. A partir dessa abordagem, foram apresentadas quatro metodologias distintas para ser aplicadas a problemas do tipo:

---

<sup>1</sup>Tais operadores foram testados apenas com algoritmos genéticos. No entanto, parece razoável que a metodologia aqui desenvolvida possa vir a ser aplicada a outros tipos de algoritmos evolucionários.

- mono-objetivo com restrição de igualdade,
- mono-objetivo com restrições de desigualdade,
- multiobjetivo irrestritos e
- multiobjetivo com restrições de desigualdade.

Em um problema mono-objetivo com apenas uma restrição não-linear de igualdade, é possível construir uma aproximação quadrática para a função-objetivo e para a restrição e obter, com baixo custo computacional, uma solução analítica para o problema quadrático associado a partir da resolução das condições de Karush-Kuhn-Tucker. Assim são construídas aproximações para o ponto de mínimo do problema restrito original. O algoritmo híbrido proposto apresentou um bom desempenho em problemas com restrições de igualdade. O uso das aproximações quadráticas para ambas as funções, objetivo e de restrição, melhora as propriedades de convergência do algoritmo. Os resultados confirmam que o novo algoritmo converge para uma solução melhor e, ainda, diminui o número de gerações necessário para alcançar essa solução. Uma metodologia semelhante, para tratamento de problemas com múltiplas restrições de igualdade, encontra-se em fase de testes.

Em problemas mono-objetivo com restrições de desigualdade, as aproximações quadráticas para a função-objetivo e as funções de restrições são construídas, sendo a estimativa para as coordenadas do ponto de ótimo obtida através da resolução de um problema auxiliar formulado em termos de LMI's. O algoritmo híbrido proposto apresentou um bom desempenho em problemas com múltiplas restrições de desigualdade. O uso das aproximações quadráticas para ambas as funções, objetivo e de restrição, melhora as propriedades de convergência do AG. Os resultados confirmam que o algoritmo híbrido converge para uma solução melhor e ainda, diminui o número de gerações necessário para alcançar essa solução. Entretanto, em alguns problemas o algoritmo híbrido não foi capaz de obter uma solução factível. Novos testes estão sendo feitos com o objetivo de melhorar o desempenho do algoritmo nesta classe de problemas.

Um novo procedimento de busca local para problemas multiobjetivo não-lineares irrestritos foi desenvolvido. Esse procedimento é baseado na construção de aproximações quadráticas locais para as funções-objetivo. Utilizando essas aproximações quadráticas, um problema auxiliar quadrático pode ser construído, cujas soluções Pareto-ótimas podem ser obtidas através das condições de Karush-Kuhn-Tucker para problemas multiobjetivo. Os modelos quadráticos são, assim, usados para produzir indivíduos localmente melhores no problema original, evitando o número excessivo de avaliações de funções que ocorre quando a busca local é diretamente aplicada sobre as funções originais. Esse operador de busca local

foi acoplado com o SPEA-2 e testado em alguns problemas. O algoritmo híbrido apresentou um desempenho superior à do algoritmo puro, levando-se em conta as métricas escolhidas para comparação. Apesar do desempenho superior do algoritmo híbrido, ainda é necessário encontrar alguma técnica de nicho que, se utilizada juntamente com a busca local, proporcione uma melhor distribuição do conjunto Pareto-ótimo no espaço de objetivos.

Apresentamos também uma nova metodologia para o tratamento de problemas multiobjetivo não-lineares com restrições de desigualdade não-lineares. As aproximações quadráticas para as funções foram usadas na construção de um problema mono-objetivo restrito auxiliar, criando um operador especializado de busca local para problemas restritos. Esse problema, aqui resolvido através de uma formulação LMI, fornece um indivíduo localmente melhorado. Esse operador de busca local foi acoplado com o MOGA e testado em alguns problemas. O algoritmo híbrido apresentou um desempenho superior à do algoritmo puro, levando-se em conta as métricas escolhidas para comparação.

Como sub-produto desta tese, uma nova métrica para comparação de algoritmos multiobjetivos foi apresentada: a contagem de esferas. Essa métrica, que não necessita do conhecimento prévio do conjunto Pareto-ótimo real, é um procedimento simples que fornece uma medida de distribuição das soluções no espaço de objetivos. Devido à simplicidade desta técnica proposta, e devido à circunstância atual de carência de boas medidas de comparação entre algoritmos evolucionários de otimização multiobjetivo, é possível que tal métrica venha a se mostrar mais amplamente útil. Como principal aspecto a ser tratado, apontamos a observação de que esta técnica, na sua presente forma, é sensível à escolha de um determinado parâmetro, o raio das esferas.

A metodologia de aproximação quadrática pode, a princípio, ser acoplada a qualquer algoritmo evolucionário em problemas cujas funções sejam não-lineares e com variáveis contínuas. Em particular, a otimização de funções do tipo *black-box* pode ser beneficiada com a utilização dessa metodologia, uma vez que esta faz uso de informação cujo custo de obtenção é elevado, e que de outra forma seria simplesmente descartada. Os resultados obtidos não parecem apontar contra-indicações para tal procedimento. Além disso, mesmo para funções cuja forma analítica esteja disponível, os resultados aqui apresentados parecem indicar que é benéfica a utilização dos operadores propostos.





# Apêndice A

## Modificações no Algoritmo Elipsoidal

O *algoritmo elipsoidal* foi proposto por Shor em 1977, como uma importante contribuição teórica para a teoria de otimização, uma vez que foi o primeiro método a partir do qual se demonstrava a convergência em tempo polinomial para problemas lineares [20, 21]. Entretanto, apesar do método não crescer em complexidade a cada iteração, o algoritmo elipsoidal apresenta taxas lentas de convergência em problemas lineares. Devido a este fato, o método simplex para problemas lineares nunca foi substituído pelo algoritmo elipsoidal.

Entretanto, devido à garantia de convergência em problemas convexos, à habilidade no tratamento de funções não diferenciáveis e à simplicidade, o algoritmo elipsoidal continua sendo um método importante nos processos de otimização de problemas não lineares com várias variáveis e várias restrições [20, 21, 53, 54, 55, 56, 57, 58, 59, 60]. Diversos resultados recentes na otimização de problemas não-lineares considerados difíceis podem ser encontrados, por exemplo, em teoria de controle e automação [61, 62, 63, 64, 65, 66] e em projeto de dispositivos eletromagnéticos [57, 59, 67]. Além disso, o algoritmo elipsoidal continua sendo usado para produzir resultados teóricos em otimização [68, 69].

O algoritmo elipsoidal apresenta algumas dificuldades para resolver problemas com restrições de igualdade. Nesses casos, o algoritmo pode não encontrar o ótimo restrito do problema. Os trabalhos recentes [58, 59, 60] abordam esse aspecto, e propõem algumas modificações no algoritmo com o objetivo de tratar as restrições de igualdade. O algoritmo proposto aqui, o *Algoritmo Elipsoidal com Aproximação Quadrática* (AEAQ), lida com as restrições de igualdade de uma maneira direta e com melhores taxas de convergência.

O algoritmo elipsoidal clássico necessita de um cálculo de gradiente por iteração, o gradiente da função-objetivo, no caso de pontos factíveis, ou o gradiente da restrição mais violada, no caso de pontos infactíveis. Neste método clássico, o gradiente é utilizado apenas para definir o hiperplano que cortará o elipsóide atual e assim gerar o novo elipsóide. A idéia que vamos explorar neste trabalho é muito simples: a informação sobre o gradiente da função

pode ser usada para construir uma estimativa da Hessiana da função-objetivo, utilizando, por exemplo, a estimativa proveniente de métodos quasi-Newton ou métodos diretos descritos na seção 2.2.2 ou na seção 2.2.1. Esta aproximação quadrática da função-objetivo fornece uma estimativa para o mínimo irrestrito, estimativa computada sem nenhuma avaliação adicional da função. Para restrições lineares, a estimativa do mínimo restrito pode ser calculada com baixo custo computacional. Assim como em [58], se as restrições não são lineares, o problema poderá ainda ser resolvido através de uma sequência de aproximações lineares dessas restrições. Claramente, as sequências das estimativas para o mínimo restrito e irrestrito que são geradas desta maneira apresentarão uma convergência mais acelerada para o ótimo da função quando as sequências se encontrarem na vizinhança da solução ótima. Nessa vizinhança, a aproximação quadrática é válida (isso ocorre quando a função é localmente suave).

Entretanto, abordaremos também uma nova metodologia para lidarmos com problemas com restrições de igualdade não lineares, sem a necessidade da linearização das mesmas a cada iteração, baseada na construção de aproximações quadráticas também para a região factível <sup>1</sup>.

## A.1 Formulação do Problema

O problema de otimização convexa que é considerado aqui é definido por:

$$\begin{aligned} x^* = \min_x f(x) \\ \text{subject to: } \begin{cases} g(x) \leq 0 \\ Ax = b \end{cases} \end{aligned} \tag{A.1}$$

sendo  $f(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}$  e cada  $g(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^p$  é uma função real e convexa, e  $A \in \mathbb{R}^{m \times n}$  e  $b \in \mathbb{R}^m$  definem as restrições lineares de igualdade.

## A.2 O Algoritmo Elipsoidal Clássico

Considere o problema não-linear convexo (A.1), possivelmente com restrições de igualdade. O algoritmo elipsoidal começa com um elipsóide  $E_0$ , centrado no ponto inicial  $x_0$ , que contém o mínimo global do problema. A cada iteração, o vetor  $d_k$  corresponde ao gradiente

---

<sup>1</sup>Os resultados desse apêndice foram submetidos à publicação.

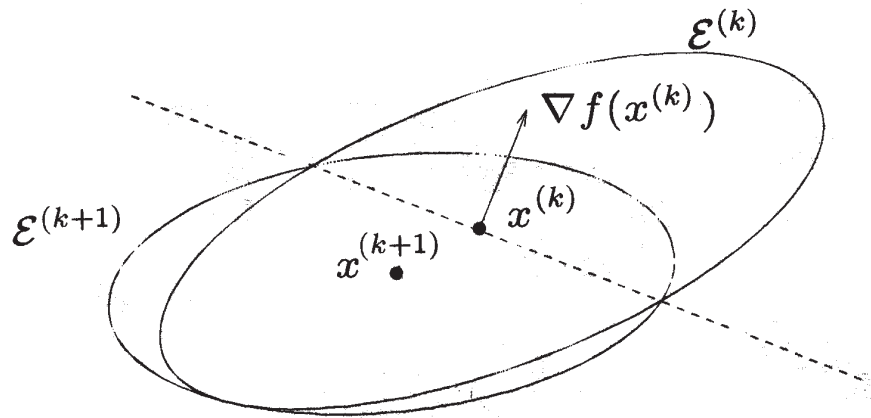


Figura A.1: Corte convencional em um elipsóide em um problema bi-dimensional.

da restrição mais violada se  $x_k$  não é factível ou o gradiente da função-objetivo se  $x_k$  é factível. Este vetor  $d_k$  nos fornece a direção de corte no elipsóide a cada iteração. Usando esta direção

$$d_k = \frac{Q_k \cdot g_k}{\sqrt{g_k^T \cdot Q_k \cdot g_k}} \quad (\text{A.2})$$

as seguintes fórmulas recursivas

$$\begin{aligned} x_{k+1} &= x_k - \frac{1}{n+1} \cdot d_k \\ Q_{k+1} &= \frac{n^2}{n^2-1} \left( Q_k - \frac{2}{n+1} \cdot d_k \cdot d_k^T \right) \end{aligned} \quad (\text{A.3})$$

geram uma sequência de pontos  $x_k$  que correspondem aos centros de cada novo elipsóide. Geometricamente, a cada iteração, um hiperplano  $H_k$  cujo vetor normal é dado por  $d_k$ , é construído de modo que passe por  $x_k$ . Este hiperplano divide o elipsóide  $E_k$  ao meio. O novo elipsóide é o menor elipsóide englobando a metade de  $E_k$  que contém a solução  $x^*$ . A figura A.1 ilustra este fato em um problema bidimensional.

O elipsóide  $E_{k+1}$  pode ser maior que o elipsóide  $E_k$  em diâmetro (máximo comprimento dos semi-eixos) mas é sempre menor em volume. O volume  $V$  do elipsóide  $E_k$  decresce de acordo com

$$V[E_k] = c_n^k \cdot V[E_0] \quad (\text{A.4})$$

sendo

$$c_n = \frac{n}{n+1} \left( \frac{n^2}{n^2-1} \right)^{\frac{n-1}{2}} < 1. \quad (\text{A.5})$$

Observe que o fator de redução do volume dos elipsóides,  $c_n$ , depende apenas da dimensão  $n$  do espaço.

O elipsóide inicial contém, por hipótese, o ponto ótimo e cada novo elipsóide contém este ponto também. Então, a sequência gerada de centros dos elipsóides converge para um elipsóide de volume zero que contém  $x^*$ . Para garantir a convergência do algoritmo, também é necessário que o conjunto factível tenha dimensão  $n$ . Esta particularidade impede que o algoritmo elipsoidal seja diretamente utilizado em problemas com restrições de igualdade.

### A.3 Tratamento de Restrições de Igualdade

Restrições de igualdade constituem uma dificuldade para o método elipsoidal clássico uma vez que essas restrições estarão sempre ativas, o que significa que a função-objetivo nunca será utilizada para definir o corte no elipsóide. Shah et al ([58]) propõe uma modificação no algoritmo elipsoidal clássico que permite resolver problemas não-lineares com restrições de igualdade. O algoritmo de Shah define uma nova fórmula recursiva para o centro do elipsóide, previamente dado por

$$u_k = -\beta_1 \frac{Q_k \cdot d_k}{\sqrt{d_k^T \cdot Q_k \cdot d_k}}, \quad (\text{A.6})$$

que está restrita ao plano

$$\mathcal{F} = \{x \in \mathbb{R}^n : Ax = b\}. \quad (\text{A.7})$$

Definindo  $g_k$  como o vetor gradiente da função-objetivo no ponto  $x_k$ , a direção  $d_k$  que minimiza  $g_k^T d_k$  sobre o elipsóide  $E_k$  e que se encontra sobre o plano  $\mathcal{F}$  é dada por

$$d_k = -\frac{\left( Q_k - Q_k A^T (A Q_k A^T)^{-1} A Q_k \right) g_k}{\sqrt{g_k^T \left( Q_k - Q_k A^T (A Q_k A^T)^{-1} A Q_k \right) g_k}}. \quad (\text{A.8})$$

Se o ponto inicial  $x_0$  se encontra no plano  $\mathcal{F}$ , o algoritmo utiliza as fórmulas recursivas do algoritmo clássico, com  $d_k$  definido como acima, para gerar uma sequência de elipsóides  $E_k$ , contendo  $x^*$ , e cujos centros  $x_k$  pertencendo ao plano  $\mathcal{F}$ .

A restrição de  $d$  ao plano  $F$  pode levar a resultados imprecisos. Para refinar essas estimativas, o algoritmo deve ser reinicializado, sempre com um novo elipsóide menor  $E_0$ , centrado na melhor estimativa  $x_k$  gerada até o momento.

Rugenstein e Kupferrschimid [60] propõem uma modificação no algoritmo proposto por Shah, criando uma duas novas estratégias para o reinício do algoritmo, baseadas nos métodos de conjuntos ativos, objetivando uma melhora na eficiência computacional. Essas novas estratégias são capazes de determinar  $x^*$  no caso de problemas convexos.

Um método alternativo para resolver problemas não-lineares com restrições de igualdade será apresentado. Este método, baseado no algoritmo elipsoidal clássico, calcula estimativas para o ponto de ótimo através do centro  $x_k$  de cada elipsóide. Essa adaptação não interfere nas propriedades de convergência do algoritmo e também não interfere na eficiência do mesmo. Além disso, o método mostra-se mais simples e mais eficiente, quando comparado ao algoritmo de Shah, para uma variedade de problemas convexos com restrições de igualdade.

## A.4 Algoritmo Elipsoidal com Aproximações Quadráticas

Considere primeiramente o problema não-linear com apenas uma restrição de igualdade:

$$\begin{aligned} x^* &= \arg \min_x f(x) \\ \text{subject to: } Ax &= b \end{aligned} \tag{A.9}$$

sendo  $f(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}$  uma função real e convexa.

Uma técnica que funciona em problemas cuja função objetivo é quadrática pode ser estendida a problemas não-quadráticos já que, perto da solução, a aproximação quadrática é válida para funções suaves.

Na metodologia de aproximações quadráticas, as seguintes associações em  $x_k$  podem ser feitas:

$$\begin{aligned} g_k &\leftrightarrow \nabla f(x_k)^T \\ Q &\leftrightarrow F(x_k) \end{aligned} \tag{A.10}$$

nos quais  $\nabla f$  é o gradiente da função-objetivo e  $F$  é a inversa da Hessiana da função. Simi-

larmente ao método de Newton, a cada passo, a solução do problema original é aproximada pela solução de um problema quadrático através dessas associações.

Se o problema for quadrático, a convergência deverá se dar em  $n + 1$  passos, um número de iterações muito menor se comparado com o número de iterações necessário no caso de aplicarmos o algoritmo elipsoidal clássico. Quando aplicamos o novo método a problemas não-quadráticos, não teremos o fim do processo em  $n + 1$  passos, mas é possível continuar o processo e terminá-lo apenas quando algum critério de parada for atingido.

Motivados por essas considerações, apresentamos uma adaptação no algoritmo elipsoidal clássico para ser usado em problemas com restrições de igualdade e de desigualdade.

#### A.4.1 Problemas Irrestritos

O algoritmo elipsoidal gera uma sequência de pontos  $x_k$  que correspondem ao centros dos elipsóides a cada iteração. Para cada  $x_k$  gerado, uma estimativa  $x_{k+1}^e$  para o mínimo do problema  $x^*$  é calculada usando a expressão:

$$x_k^e = x_k - H_k \cdot g_k \quad (\text{A.11})$$

sendo  $H_k$  e  $g_k$ , respectivamente, a Hessiana e o gradiente da função no ponto  $x_k$ . Esta sequência de estimativas converge para o ponto de ótimo, então um novo critério de parada usando essas estimativas é introduzido no algoritmo.

A Hessiana que é utilizada na estimativa acima é construída usando o método *BFGS* [14], com as seguintes fórmulas recursivas:

$$H_{k+1} = H_k + c_k \quad (\text{A.12})$$

sendo

$$c_k = \left( 1 + \frac{r_k^T H_k r_k}{r_k^T v_k} \right) \frac{v_k v_k^T}{v_k^T r_k} - \frac{v_k r_k^T H_k + H_k r_k v_k^T}{r_k^T v_k} \quad (\text{A.13})$$

e  $r_k = g_k - g_{k+1}$  e  $v_k = x_k - x_{k+1}$ . Outra alternativa para a construção da Hessiana é através de uma construção direta usando os vetores  $r_k = g_k - g_{k+1}$  e  $v_k = x_k - x_{k+1}$ . Temos que

$$H[v_1 v_2 \dots v_n] = [r_1 r_2 \dots r_n]. \quad (\text{A.14})$$

Definindo  $V = [v_1 \ v_2 \ \dots \ v_n]$  e  $R = [r_1 \ r_2 \ \dots \ r_n]$ , obtemos:

$$H = R \cdot V^{-1} \quad (\text{A.15})$$

Com  $n + 1$  cálculos da função e seus respectivos gradientes a Hessiana pode ser calculada, no caso de uma função-objetivo quadrática.

### A.4.2 Problemas com Restrições de Igualdade

No caso de problemas restritos, para cada  $x_k$  gerado, duas estimativas para o mínimo serão produzidas. A primeira,  $x_k^e$ , é usada se  $x_k$  for factível e coincide com a estimativa para o caso irrestrito. A outra estimativa,  $x_k^c$ , é usada se  $x_k$  for infactível. A seguir, apresentaremos o procedimento para gerar a estimativa restrita  $x_k^c$ .

Sabemos que  $2H(x_1 - x^*) = g_1$  é a representação analítica do gradiente da função-objetivo no ponto  $x_1$ . Desta maneira temos,

$$2H(x_{k+1} - x_k) = g_{k+1} - g_k.$$

Portanto, dado um ponto  $x_k$ , a estimativa  $x_k^c$  será a solução das duas equações

$$N^T H(x_{k+1} - x_k) = 0 \tag{A.16}$$

$$Ax_{k+1} = b$$

sendo  $N$  uma matriz ortogonal tal que  $AN = 0$ . Note que a estimativa restrita,  $x_k^c$ , produz uma estimativa para o mínimo através da projeção de  $x_k$  sobre o conjunto factível.

O algoritmo elipsoidal verifica a cada iteração se  $x_k$  é factível ou não; porém não sabemos se as estimativas  $x_k^e$  são factíveis ou não. Com o objetivo de evitar essas verificações, o que reduziria a eficiência do algoritmo, devemos calcular ambas as estimativas,  $x_k^e$  and  $x_k^c$ , para cada  $x_k$  e armazená-las em dois arquivos distintos.

Novamente estas estimativas são usadas em um critério de parada. O método termina por esse critério se a convergência for atingida por alguma das estimativas,  $x_k^e$  ou  $x_k^c$ . Apenas quando o critério de parada for atingido, um teste para verificação da factibilidade de  $x_k^e$  deve ser efetuado. Se for factível, este valor será a solução do problema. No outro caso,  $x_k^c$  será a solução.

É importante salientar que essas adaptações no algoritmo elipsoidal clássico não interferem nas propriedades de convergência do algoritmo: a sequência de  $x_k$  é ainda governada pelo algoritmo elipsoidal convencional, e converge sob as mesmas condições. As sequências  $x_k^e$  e  $x_k^c$  são mantidas em paralelo com o algoritmo clássico, e apenas interferem no mesmo se a convergência for detectada, caso em que a nova estimativa é uma solução melhor do que  $x_k$ .

## A.5 Testes Numéricos

O primeiro teste utilizou funções quadráticas de dimensões variadas e mostrou uma efetiva melhora na eficiência do novo algoritmo quando comparado ao algoritmo clássico. Como resultado, o número de gerações foi, em média, reduzido para um terço do número de gerações utilizado pelo algoritmo clássico. As figuras A.2 e A.3 mostram a evolução da terceira coordenada para o algoritmo elipsoidal clássico e para o AEAQ em um problema irrestrito de dimensão 10. As figuras A.4 e A.5 mostram a evolução da primeira coordenada para o algoritmo elipsoidal clássico e para o AEAQ em um problema restrito de dimensão 10. Essas figuras mostram o comportamento típico do vetor  $x_k^e$ , o qual é muito mais estável que  $x_k$ . Após poucas iterações,  $x_k^e$  atinge o critério de parada, e a série termina. A série de vetores  $x_k$  corresponde à solução do algoritmo elipsoidal clássico.

O AEAQ foi executado com uma tolerância de 0.0001 e foi testado com os problemas listados abaixo. Com o propósito de comparação, o elipsoidal clássico e o algoritmo de Shah também foram testados nos mesmos problemas. Os problemas são definidos da seguinte forma:

1.

$$\begin{aligned} x^* &= \arg \min_x 3(x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 2)^2 \\ \text{sujeito a: } & x_1 - 3 = 0 \end{aligned} \quad (\text{A.17})$$

2.

$$\begin{aligned} x^* &= \arg \min_x \sin(x_1 + x_2) + (x_1)^2 + (x_2)^2 + (x_3)^2 \\ \text{sujeito a: } & \begin{cases} x_1 + x_2 = 0 \\ 0.64 \leq x_1 \leq 0.68 \\ -4.35 \leq x_2 \leq -4.1 \end{cases} \end{aligned} \quad (\text{A.18})$$

3.

$$\begin{aligned} x^* &= \arg \min_x \cos(2\pi \cdot x_1 + 2\pi \cdot x_2) \\ \text{sujeito a: } & \begin{cases} x_1 - x_2 - 5 = 0 \\ -1 \leq x_i \leq 1 \\ i = 1, 2 \end{cases} \end{aligned} \quad (\text{A.19})$$

4.

$$\begin{aligned} x^* &= \arg \min_x 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \\ \text{sujeito a: } & \begin{cases} x_1 + x_2 = 0 \\ -2.48 \leq x_i \leq 2.48 \\ i = 1, 2 \end{cases} \end{aligned} \quad (\text{A.20})$$

Os algoritmos foram inicializados com pontos aleatórios (mesmo ponto inicial para os



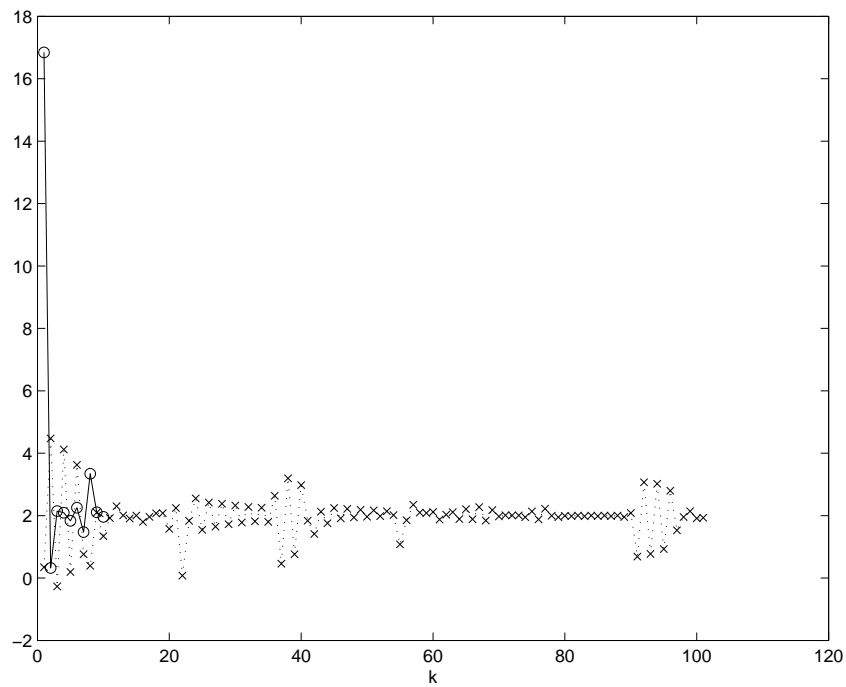


Figura A.2: Evolução da terceira coordenada de  $x_k$  ( $\times$ ) e  $x_k^e$  ( $\circ$ ) em um problema quadrático irrestrito de dimensão 10.

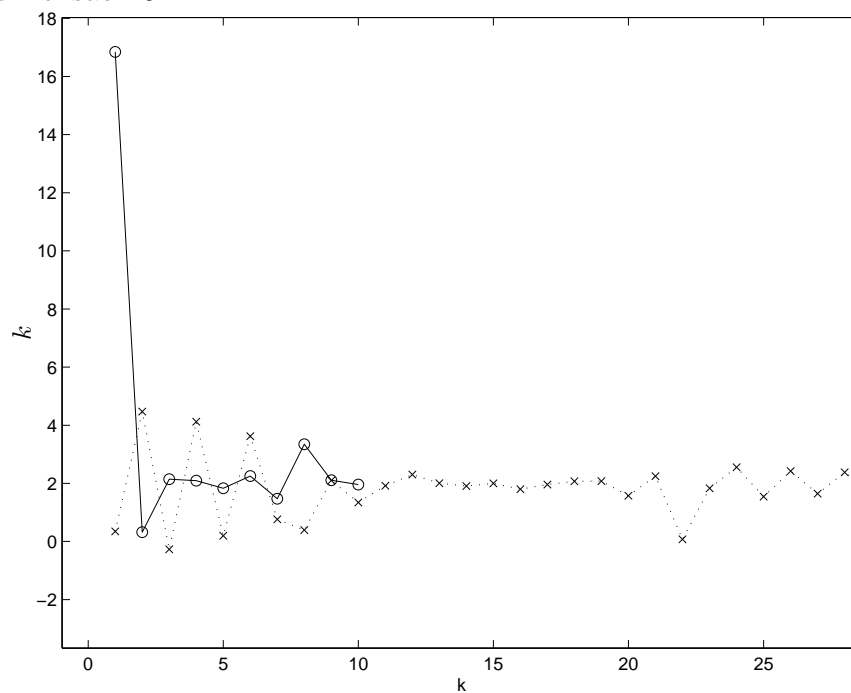


Figura A.3: Detalhe ampliado da evolução da terceira coordenada de  $x_k$  ( $\times$ ) e  $x_k^e$  ( $\circ$ ) em um problema quadrático irrestrito de dimensão 10.

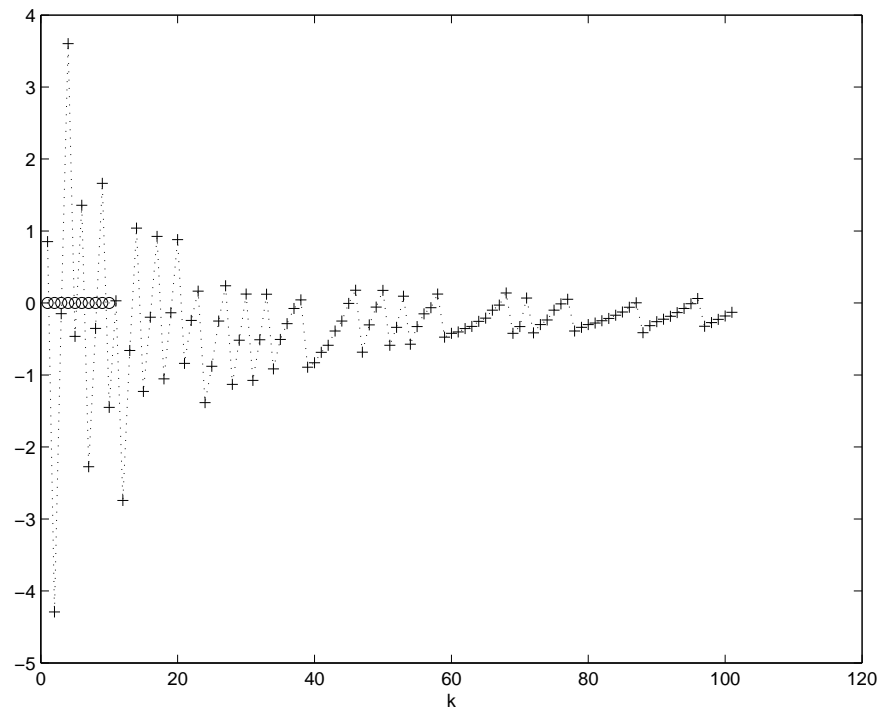


Figura A.4: Evolução da primeira coordenada de  $x_k$  ( $\times$ ) e  $x_k^e$  ( $\circ$ ) em um problema quadrático restrito de dimensão 10.

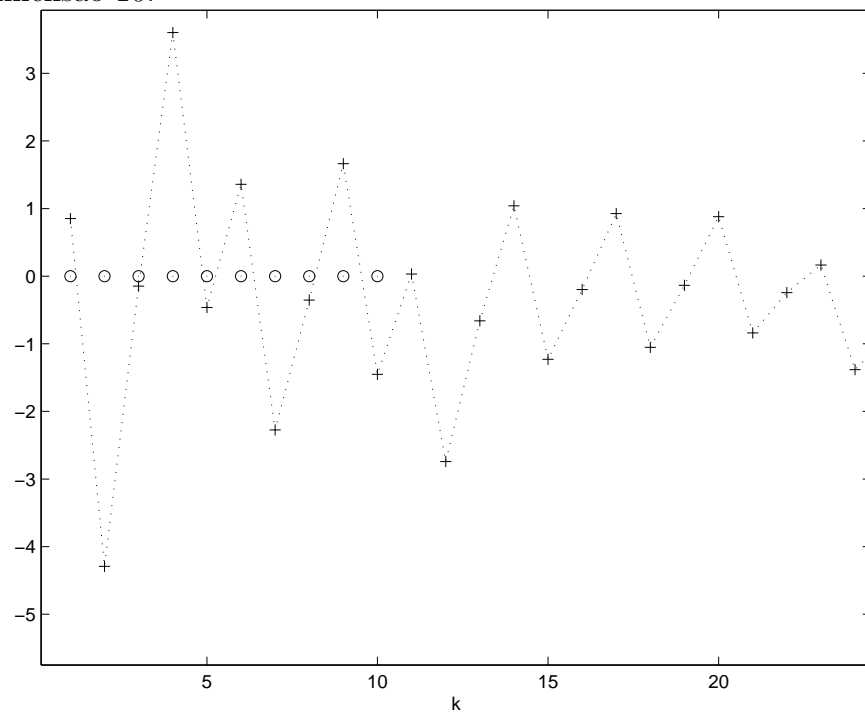


Figura A.5: Detalhe ampliado da evolução da primeira coordenada de  $x_k$  ( $\times$ ) e  $x_k^e$  ( $\circ$ ) em um problema quadrático restrito de dimensão 10.

três algoritmos) em todos os problemas. A tabela A.1 mostra os resultados obtidos pelos três algoritmos. Para cada algoritmo, o número de avaliações de função-objetivo, a norma infinito de  $x^* - x$ , o valor absoluto de  $f^* - f$ , e a porcentagem de soluções corretas estão indicadas.

Tabela A.1: Resultados Numéricos.

Algoritmo		1	2	3	4
AEAQ	N	17	21	15	9
	%acerto	100	92	90	88
	$ x^* - x _\infty$	0.0323	0.0196	0	0.0091
	$ f^* - f _\infty$	0.0073	0.0992	0.0045	0.0050
Shah	N	50	37	14	15
	%acerto	20	89	50	65
	$ x^* - x _\infty$	0.5053	0.0164	0.4223	0.0141
	$ f^* - f _\infty$	0.2588	0.0067	0.0495	0.0072
AE Clássico	N	82	81	16	41
	%acerto	10	0	40	81
	$ x^* - x _\infty$	0.6061	0.3854	0.0432	0.0083
	$ f^* - f _\infty$	1.2000	0.3911	0.1802	0.0061

Em todos os problemas, o AEAQ apresentou um bom desempenho, alcançando a solução com um número inferior de avaliações de função-objetivo se comparado ao algoritmo elipsoidal clássico e ao algoritmo de Shah.

## A.6 Conclusão

Um novo procedimento para ser utilizado dentro do algoritmo elipsoidal foi proposto: uma estimativa para a solução através da aproximação quadrática. Este procedimento tem a vantagem de não interferir nas propriedades de convergência do algoritmo elipsoidal, uma vez que a sequência de soluções do algoritmo clássico permanece inalterada, enquanto a sequências das estimativas são construídas paralelamente. Essa nova sequência de estimativas é gerada utilizando as informações que estão disponíveis a cada iteração do algoritmo elipsoidal (os cálculos dos vetores gradientes das funções), não acrescentando nenhuma avaliação extra de função. O algoritmo elipsoidal com o novo procedimento incluído foi denominado Algoritmo Elipsoidal com Aproximação Quadrática (AEAQ).

Para o caso de problemas com restrições de igualdade, existe um algoritmo recentemente proposto (o algoritmo de Shah) que é uma adaptação do algoritmo elipsoidal e que pode ser comparado ao algoritmo AEAQ proposto. O AEAQ apresentou um desempenho melhor nos testes, mostrando-se mais aplicável uma vez que pode ser usado com problemas irrestritos

e restritos, com as restrições de igualdade como um caso particular.

O custo computacional adicional decorrente da construção da aproximação quadrática é relativamente baixo, já que este custo é proveniente das atualizações do BFGS mais um cálculo de uma projeção no caso da restrição de igualdade. A vantagem de se utilizar o procedimento proposto torna-se mais evidente em problemas nos quais a avaliação da função é cara do ponto de vista computacional.

## Apêndice B

# Metodologia de LMI ( Linear Matrix Inequalities) para Otimização em Problemas Quadráticos

Desigualdades Matriciais Lineares <sup>1</sup> (LMIs) e as técnicas de LMI surgiram como uma poderosa ferramenta em áreas como controle em engenharia, identificações de sistemas e projeto estrutural. Vários de problemas podem ser formulados usando LMI e, uma vez estabelecidos nos termos de LMI, podem ser resolvidos utilizando eficientes algoritmos de otimização convexa, conhecidos como *LMI solvers*. Esses programas são mais rápidos que os métodos clássicos de otimização convexa.

Apresentaremos nesse apêndice uma metodologia para se encontrar o ótimo de um problema quadrático. Essa metodologia é uma particularização das LMIs para funções quadráticas. Iniciaremos com algumas definições e resultados básicos das técnicas de LMI.

Uma LMI é qualquer restrição da forma:

$$A(x) = A_0 + x_1A_1 + \cdots + x_nA_n < 0 \quad (\text{B.1})$$

sendo  $x = (x_1, x_2, \cdots, x_n)$  um vetor de variáveis,  $A_0, A_1, \cdots, A_n$  matrizes simétricas. O símbolo  $A < 0$  significa que a matriz  $A$  é *negativa definida*, isto é, o maior autovalor de  $A(x)$  é negativo. Observe que  $A(x) > 0$  e  $A(x) < B(x)$  são casos especiais de (B.1) uma vez que podem ser re-escritos respectivamente como  $-A(x) < 0$  e  $A(x) - B(x) < 0$ .

A LMI (B.1) representa uma restrição convexa em  $x$  desde que  $A(y) < 0$  e  $A(z) < 0$

---

<sup>1</sup>Tradução para Linear Matrix Inequalities.

impliquem em

$$A \left( \frac{y+z}{2} \right) < 0 \quad (\text{B.2})$$

Desta forma, o conjunto de solução da LMI é um subconjunto convexo de  $\mathbb{R}^n$  e o objetivo de obter uma solução para B.1 pode ser visto como um problema de programação convexa.

O seguinte lema, e alguns resultados adicionais, serão utilizados na construção da nossa metodologia utilizada em problemas restritos mono-objetivo e multiobjetivo:

**Lema 1** (*Lema de Schur*) *As equações (B.3) e (B.4) são equivalentes:*

$$\begin{bmatrix} Q & S \\ S' & R \end{bmatrix} > 0 \quad (\text{B.3})$$

$$\begin{cases} R > 0 \\ Q - SR^{-1}S' > 0 \end{cases} \quad (\text{B.4})$$

sendo  $R$  e  $Q$  matrizes simétricas,  $S$  uma matriz com dimensão compatível, e  $(\cdot) > 0$  denota que o argumento é definido positivo.

O Lema de Schur transforma termos quadráticos em termos lineares aumentados. A prova do Lema 1 pode ser encontrada em [70, 29].

O primeiro resultado, baseado no Lema de Schur, é dado por:

**Lema 2** *Considere o seguinte problema de otimização com função-objetivo quadrática e restrições quadráticas:*

$$x^* = \arg \min (x - x_0)' Q_0 (x - x_0)$$

$$\text{sujeito a: } \begin{cases} (x - x_1)' Q_1 (x - x_1) - 1 < 0 \\ (x - x_2)' Q_2 (x - x_2) - 1 < 0 \\ \vdots \\ (x - x_m)' Q_m (x - x_m) - 1 < 0 \end{cases} \quad (\text{B.5})$$

O problema de otimização (B.5) pode ser re-escrito como:

$$x^* = \arg_x \min_{x, \epsilon} \epsilon$$

$$\text{sujeito a: } \left\{ \begin{array}{l} \left[ \begin{array}{cc} \epsilon & (x - x_0)' \\ x - x_0 & Q_0^{-1} \end{array} \right] > 0 \\ \left[ \begin{array}{cc} C_1 & (x - x_1)' \\ x - x_1 & Q_1^{-1} \end{array} \right] > 0 \\ \left[ \begin{array}{cc} C_2 & (x - x_2)' \\ x - x_2 & Q_2^{-1} \end{array} \right] > 0 \\ \vdots \\ \left[ \begin{array}{cc} C_m & (x - x_m)' \\ x - x_m & Q_m^{-1} \end{array} \right] > 0 \end{array} \right. \quad (\text{B.6})$$

**Prova:**

Troque:

$$\min (x - x_0)' Q_0 (x - x_0)$$

por:

$$\min \epsilon$$

$$\text{sujeito a: } (x - x_0)' Q_0 (x - x_0) < \epsilon$$

As operações restantes são aplicações imediatas do Lema de Schur para as desigualdades quadráticas. ■

Da mesma maneira, outro resultado pode ser estabelecido para problemas com algumas restrições quadráticas e outras lineares:

**Lema 3** Considere o seguinte problema de otimização com função-objetivo quadráticas e restrições quadráticas e lineares:

$$x^* = \arg \min (x - x_0)'Q_0(x - x_0)$$

$$\text{sujeito a: } \left\{ \begin{array}{l} (x - x_1)'Q_1(x - x_1) - 1 < 0 \\ (x - x_2)'Q_2(x - x_2) - 1 < 0 \\ \vdots \\ (x - x_p)'Q_p(x - x_p) - 1 < 0 \\ a_1x - b_1 < 0 \\ a_2x - b_2 < 0 \\ \vdots \\ a_qx - b_q < 0 \end{array} \right. \quad (\text{B.7})$$

O problema (B.7) pode ser re-escrito como:

$$x^* = \arg_x \min_{x,\epsilon} \epsilon$$

$$\text{sujeito a: } \left\{ \begin{array}{l} \left[ \begin{array}{cc} \epsilon & (x - x_0)' \\ x - x_0 & Q_0^{-1} \end{array} \right] > 0 \\ \left[ \begin{array}{cc} 1 & (x - x_1)' \\ x - x_1 & Q_1^{-1} \end{array} \right] > 0 \\ \left[ \begin{array}{cc} 1 & (x - x_2)' \\ x - x_2 & Q_2^{-1} \end{array} \right] > 0 \\ \vdots \\ \left[ \begin{array}{cc} 1 & (x - x_p)' \\ x - x_p & Q_p^{-1} \end{array} \right] > 0 \\ 2a_1 - c_1x - x'c'_1 > 0 \\ 2a_2 - c_2x - x'c'_2 > 0 \\ \vdots \\ 2a_q - c_qx - x'c'_q > 0 \end{array} \right. \quad (\text{B.8})$$



A primeira restrição de (B.8) diz respeito à função-objetivo quadrática, as  $p$  seguintes restrições dizem respeito às restrições não-lineares e as  $q$  últimas restrições dizem respeito às restrições lineares do problema original. Este problema, (B.8), pode ser eficientemente resolvido com qualquer programa para gerar soluções de LMIs baseados nos métodos de pontos interiores. Usaremos, assim como no caso da construção da aproximação quadrática, o SeDuMi para resolver este problema.

SeDuMi é uma toolbox para o MATLAB, que permite a resolução de problemas de otimização com restrições lineares, quadráticas semi-definidas. SeDuMi significa *Self-Dual Minimization* e implementa a técnica de *self-dual imbedding* para a otimização sobre cones homogêneos *self-dual*, ou mais precisamente, otimização sobre cones simétricos. SeDuMi leva vantagem em relação à esparsidade, o que gera benefícios relativos à velocidade. Utilizamos também o *Yalmip* que é uma interface simples do SeDuMi para o MATLAB.



# Referências Bibliográficas

- [1] K. A. De Jong. Genetic algorithms are not function optimizers. *in Foundations of Genetic Algorithms 2*, Darrel L. Whitley Editor(4):5–17, 1993.
- [2] M. Mitchell. *An introduction to genetic algorithms*. MIT Press, 1996.
- [3] J. M. Jonhson and Y. Rahmat-Samii. Genetic algorithms in engineering electromagnetics. *IEEE Antennas and propagation Magazine*, 39(4):7–21, 1997.
- [4] J.M. Renders and H. Bersini. Hybridizing genetic algorithms with hill-climbing methods for global optimization: Two possible ways. In *Proceedings of the first IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, pages 312–317. IEEE Press, 1994.
- [5] J.M Renders and S.P. Flasse. Hybrid methods using genetic algorithms for global optimization. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 26(2):246–258, 1996.
- [6] G. E. P. Box and N. R. Draper. *Empirical Model-Building and Response Surfaces*. Wiley, New York, 1987.
- [7] A. I. Khuri and J. A. Cornell. *Response Surfaces*. Marcel Dekker, New York, 1987.
- [8] R. H. Myers and D.C. Montgomery. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. Wiley, New York, 1995.
- [9] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.
- [10] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Desing and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 1989.
- [11] M. Bjorkman and K. Holmstrom. Global optimization of costly nonconvex function using radial basis functions. *Opt. Eng*, 1(4):373–397, 2000.

- [12] T. Ishikawa and M. Matsunami. An optimization method based on radial basis function. *IEEE Transactions on Magnetism*, 33:1868–1871, 1997.
- [13] W. Chen and S. Varadarajan. Integration of design of experiments and artificial neural networks for achieving affordable concurrent design. *presented at 38th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, Materials Conference*, 1997. Kissimmee, Fl.
- [14] D. G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, 1984.
- [15] R. G. Regis and C. A. Shoemaker. Local function approximation in evolutionary algorithms for the optimization of costly functions. *IEEE Transactions on Evolutionary Computation*, 8(5):490–505, 2004.
- [16] K. Rasheed. An incremental-approximate-clustering approach for developing dynamic reduced models for design optimization. In *Proceedings of the Conference on Evolutionary Computation*, pages 986–993, 2000.
- [17] K. Svanberg. The method of moving asymptotes: a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24:359–373, 1987.
- [18] A. R. Conn, K. Scheinberg, and P. L. Toint. Recent progress in unconstrained nonlinear optimization without derivatives. *Mathematical Programming*, 79:397–414, 1997.
- [19] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley and sons, Inc., 1979.
- [20] R. G. Bland, D. Goldfarb, and M. J. Todd. The ellipsoid method: a survey. *Operations Research*, 29(6):1039–1091, 1981.
- [21] M. Akgül. *Topics in Relaxation and Ellipsoidal Methods*. Number 97 in Research Notes in Mathematics. Pitman Publishing Inc., London, UK, 1984.
- [22] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 99.
- [23] T. Back. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.
- [24] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Learning*. Addison-Wesley, 1989.

- [25] R. H. C. Takahashi, J. A. Vasconcellos, J. A. Ramirez, and L. Krahenbuhl. A multiobjective methodology for evaluation genetic operators. *IEEE Transactions on Magnetics*, 39:1321–1324, 2003.
- [26] L. Davis. *Genetic Algorithm and Simulated Annealing*. Pitman, 1987.
- [27] C. A. Coello Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191:1254–1287, 2002.
- [28] E. F. Wanner, F. G. Guimarães, R. R. Saldanha, R. H. C. Takahashi, and P. F. Fleming. Constraint quadratic approximation operator for treating equality constraints with genetic algorithms. In *Proceedings of the IEEE Congress on Evolutionary Computation*, Edinburgh, UK, 2005. IEEE Press.
- [29] S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. Linear matrix inequalities in system and control theory. *SIAM Studies in Applied Mathematics*, 15, 1997.
- [30] Z. Michalewicz. Genetic algorithms, numerical optimization and constraints. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 151–158, San Francisco, CA, 1995. Morgan Kauffman.
- [31] K. Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186:311–338, 2000.
- [32] D. M. Himmelblau. *Applied Nonlinear Programming*. McGraw–Hill, New York, USA, 1972.
- [33] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. *A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA–II*. KanGAL Report No. 200001, Kanpur Genetic Algorithm Laboratory, Kanpur, 2000.
- [34] Z. Michalewicz and M. Schoenauer. Evolutionary algorithms for constrained parameter optimization problem. *Evolutionary Computation*, 1996.
- [35] E. Mezura-Montes and C. A. Coello Coello. A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Transactions on Evolutionary Computation*, 9(1):1–17, 2005.
- [36] E. F. Wanner, F. G. Guimarães, R. H. C. Takahashi, and P. F. Fleming. Quadratic approximation-based coordinate change in genetic algorithms. In *Proceedings of the*

- IEEE Congress on Evolutionary Computation*, Vancouver, CA, 2006. IEEE Press. (to appear).
- [37] K. Krishna-Kumar, S. Narayanaswamy, and S. Garg. *Solving large parameters optimization problems using a genetic algorithm with stochastic coding*. Genetic Algorithms in Engineering and Computer Science. G. Winter, J. Periaux, M. Galan and P. Cuesta Eds, New York, 1995.
- [38] K. M.iettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, Massachusetts, 1999.
- [39] M. L. Lucian and M. Neamtu. Geometric modeling and computing: Seattle 2003. In *Geometric Modeling and Computing: Seattle 2003*, Modern Methods in Mathematics, Brentwood, TN, 2004. Nashboro Press. ISBN 0-0-9728482-3-1.
- [40] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strenght pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.
- [41] E. Zitzler, M. Laumanns, and L. Thiele. SPEA 2: Improving the Strenght Pareto Evolutionary Algorithms. In K. Giannakoglou et al., editor, *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pages 95–100, 2002.
- [42] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.
- [43] D. A. Van Veldhuizen and G.B. Lamont. On measuring multiobjective evolutionary algorithm performance. In *Proceedings of the Conference on Evolutionary Computation*, pages 204–211. IEEE, 2000.
- [44] J. Knowles and D. Corne. On metrics for comparing nondominated sets. In *Proceedings of the Conference on Evolutionary Computation*, pages 711–216. IEEE, 2002.
- [45] T. Okabe, Y. Jin, and B. Sendhoff. A critical survey of performance indices for multiobjective optimisation. In *Proceedings of the Conference on Evolutionary Computation*, pages 878–885. IEEE, 2003.
- [46] F. G. Guimarães, E. F. Wanner, R. H. C. Takahashi, and P. F. Fleming. Local search with quadratic approximations into memetic algorithms for optimization with multiple criteria. *Evolutionary Computation*, 2006. (submitted).

- [47] E. Zitzler. Evolutionary algorithms for multiobjective optimization: Methods and applications. *Ph.D. Thesis, Swiss Federal Institute of Technology (ETH), Zurich*, 1999.
- [48] R. C. Purshouse. On the evolutionary optimisation of many objectives. *Ph.D. Thesis, Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, UK*, 2003.
- [49] X. Hu, Z. Huang, and Z. Wang. Hybridization of the multi-objective evolutionary algorithms and the gradient-based algorithms. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 870–877. IEEE Press, 2003.
- [50] F. Kursawe. A variant of evolution strategies for vector optimization. *Parallel Problem Solving from Nature*, pages 193–197, 1991.
- [51] E. F. Wanner, F. G. Guimarães, R. H. C. Takahashi, and P. F. Fleming. A quadratic approximation-based local search procedure for multiobjective genetic algorithms. In *Proceedings of the IEEE Congress on Evolutionary Computation*, Vancouver, CA, 2006. IEEE Press. (to appear).
- [52] F. W. Gembicki and Y. Y. Haimes. Approach to performance and sensitivity multiobjective optimization: the goal attainment method. *IEEE Transactions on Automatic Control*, 20(6):769–771, 1975.
- [53] S. T. Dziuban, J. G. Ecker, and M. Kupferschmid. Using deep cuts in an ellipsoidal algorithm for nonlinear programming. *Mathematical Programming Study*, 25:93–107, 1985.
- [54] Yinyu Ye. Karmarkar’s algorithm and the ellipsoid method. *Operations Research Letters*, 6(4):177–182, 1987.
- [55] S. Boyd, L. El-Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM, 1994.
- [56] S. Kim, D. Kim, and K. N. Chang. Using 2 successive subgradients in the ellipsoid method for nonlinear programming. *Journal of Optimization Theory and Applications*, 82(3):543–554, 1994.
- [57] R. R. Saldanha, R. H. C. Takahashi, J. A. Vasconcelos, and J. A. Ramirez. Adaptive deep-cut method in ellipsoidal optimization for electromagnetic design. *IEEE Transactions on Magnetics, Part I*, 35(3):1746–1749, 1999.

- [58] Sharmila Shah, John E. Mitchell, and Michael Kupferschmid. An ellipsoid algorithm for equality-constrained nonlinear programs. *Computers and Operations Research*, 28(1):85–92, 2001.
- [59] R. H. C. Takahashi, R. R. Saldanha, W. Dias-Filho, and J. A. Ramirez. A new constrained ellipsoidal algorithm for nonlinear optimization with equality constraints. *IEEE Transactions on Magnetics*, 39(5), 2003.
- [60] E. K. Rugenstein and M. Kupferschmid. Active set strategies in an ellipsoid algorithm for nonlinear programming. *Computers and Operations Research*, 31(6):941–962, 2004.
- [61] M. Kupferschmid, K. Mohrmann, J. G. Ecker, and H. Kaufman. The ellipsoid algorithm: A new method for feedback gain optimization. *Annual Review in Automatic Programming*, 13(2):85–94, 1985.
- [62] Stephen Boyd and Craig H. Barratt. *Linear Controller Design: Limits of Performance*. Prentice-Hall, 1991.
- [63] M. F. Cheung, S. Yurkovich, and K. M. Passino. An optimal volume ellipsoid algorithm for parameter set estimation. *IEEE Transactions on Automatic Control*, 38(8):1292–1296, 1993.
- [64] Hector Rotstein and Athanasios Sideris.  $\mathcal{H}_\infty$  optimization with time-domain constraints. *IEEE Transactions on Automatic Control*, 39(4):762–779, 1994.
- [65] M. V. Correa, L. A. Aguirre, and R. R. Saldanha. Using steady-state prior knowledge to constrain parameter estimates in nonlinear system identification. *IEEE Transactions on Circuits and Systems I - Fundamental Theory and Applications*, 49(9):1376–1381, 2002.
- [66] S. Kanev, B. de Schutter, and M. Verhaegen. An ellipsoid algorithm for probabilistic robust controller design. *System and Control Letters*, 49:365–375, 2003.
- [67] R. R. Saldanha, J. L. Coulomb, and J. C. Sabonnadiere. An ellipsoid algorithm for the optimum design of magnetostatic problems. *IEEE Transactions on Magnetics*, 28(2):1573–1576, 1992.
- [68] R. M. Freund and J. R. Vera. Condition-based complexity of convex optimization in conic linear form via the ellipsoid algorithm. *SIAM Journal on Optimization*, 10(1):155–176, 1999.



- [69] R. M. Freund and J. R. Vera. On the complexity of computing estimates of condition measures of a conic linear system. *Mathematics of Operations Research*, 28(4):625–648, 2003.
- [70] A. Albert. Conditions for positive and nonnegative definiteness in terms of pseudoinverses. *SIAM J. Appl. Math.*, 17(2):434–440, 1969.