O PROBLEMA DE MIN-MAX-MIN COM RESTRIÇÕES PELO MÉTODO DE NELDER-MEAD

Angela Maria Silva Gonçalves

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientador: Paulo Roberto Oliveira

Rio de Janeiro
Agosto de 2013

O PROBLEMA DE MIN-MAX-MIN COM RESTRIÇÕES PELO MÉTODO DE NELDER-MEAD

Angela Maria Silva Gonçalves

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Paulo Roberto Oliveira, Dr.Ing.

Prof.ª Lilian Markenzon, D.Sc.

Prof. Abilio Pereira de Lucena Filho, D.Sc.

Prof. Luiz Satoru Ochi, D.Sc.

Prof. Luidi Gelabert Simonetti, D.Sc.

Prof.ª Luziane Ferreira de Mendonça, D.Sc.

Prof. Raymundo de Oliveira, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
AGOSTO DE 2013

Gonçalves, Angela Maria Silva

O Problema de Min-Max-Min com restrições pelo Método de Nelder-Mead/Angela Maria Silva Gonçalves. – Rio de Janeiro: UFRJ/COPPE, 2013.

XIII, 125 p.: il.; 29, 7cm.

Orientador: Paulo Roberto Oliveira

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2013.

Referências Bibliográficas: p. 74 – 77.

1. Nelder-Mead. 2. localização. 3. min-max-min. I. Oliveira, Paulo Roberto. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Ao meu amigo
Raymundo de Oliveira

Agradecimentos

Ao amigo e orientador Prof. Dr. Paulo Roberto Oliveira, devo gratidão pela amizade, carinho, paciência e orientação.

À Profª. Drª. Luziane Ferreira de Mendonça um profundo agradecimento, pela solidariedade em um momento difícil da minha vida e que generosamente me acolheu, propondo o tema e acompanhando o desenvolvimento deste trabalho com grande dedicação, competência e sugestões valiosas.

Agradeço a Profª. Drª. Lilian Markenzon e ao Prof. Dr. Abílio Pereira de Lucena Filho pela contribuição como membros da banca no Exame de Qualificação e pela participação dos membros da banca examinadora da defesa.

Agradeço aos professores e funcionários do Programa de Pós-graduação em Engenharia de Sistemas e Computação da COPPE/UFRJ.

Em particular a minha filha Julia agradeço a compreensão pela falta de atenção em função das horas e horas passadas em frente a um computador.

Por fim agradeço especial àqueles que sempre me apoiaram e que são os que mais compartilham da minha alegria: minha família e meus amigos, em especial Virgínia Tapajós e Milton Flores.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

O PROBLEMA DE MIN-MAX-MIN COM RESTRIÇÕES PELO MÉTODO DE NELDER-MEAD

Angela Maria Silva Gonçalves

Agosto/2013

Orientador: Paulo Roberto Oliveira

Programa: Engenharia de Sistemas e Computação

O objetivo desta tese é a resolução de um problema de *min-max-min* com restrições utilizando o Método de Nelder-Mead.

Como uma das características do problema é a multiplicidade de mínimos locais o método é reiniciado com novos valores iniciais na tentativa de busca de um mínimo global. Para a construção dos novos pontos de partida do método, a proposta aqui apresentada são os pontos de uma curva de Lissajous.

Esta modelagem pode ser vista como a localização em uma região S de n centros de serviços que podem ser antenas, armazéns, hospitais, estações de rádio base etc. com o objetivo de atender da melhor maneira possível os usuários do serviço a ser prestado.

Pode ocorrer que um ou mais destes centros fiquem localizados fora da região S. Para resolver este problema, é proposto o acréscimo de uma função de penalização à função objetivo, com a intenção de colocar os centros de serviços dentro da região S.

Resultados de exemplos computacionais para a modelagem proposta são apresentadas.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

A CONSTRAINED MIN-MAX-MIN PROBLEM USING NELDER-MEAD METHOD

Angela Maria Silva Gonçalves

August/2013

Advisor: Paulo Roberto Oliveira

Department: Systems Engineering and Computer Science

The aim of this thesis is to solve a min-max-min problem with constrainsts, using the Nelder-Mead method.

Since one of the characteristics of this problem is the existence of multiple local minima, the method is restarted with using new initial values attempting at finding a global minimum. For the construction of the new starting points of the method, the approach presented here make use of points of a Lissajous curve.

This model considered here can be applied to study of location in a region S of n service centers that can be antennas, warehouses, hospitals, radio basis stations etc in order to find the best possible position of service centers.

It may occur that one or more of these service centers are located outside the region S. To solve this problem, we propose the addition of a penalty function to the objective function, to force the installations within the area S.

Results of computational examples for the proposed model are presented.

Sumário

Lista de Figuras **x**

Lista de Tabelas **xii**

1 Introdução **1**

2 O Problema de Min-max-min **4**

2.1 Introdução 4

2.2 Localização de Antenas 7

2.3 Modelagem do Problema 9

2.4 O Problema de min-max-min e o Método de Nelder-Mead 10

3 Método de Nelder-Mead **13**

3.1 Métodos Simplex 13

3.2 Método de Nelder-Mead 14

3.3 Algoritmo de Nelder-Mead 15

3.4 Características do Simplex de Nelder-Mead 19

3.5 Propriedades do Método 21

3.6 Convergência do Método 26

4 Implementação Computacional **29**

4.1 Introdução 29

4.2 Implementação do Método para o Problema 30

4.2.1 Implementação da Função Objetivo 30

4.2.2 Retângulo Mínimo de Pontos na Região S 31

4.2.3 Violação da Fronteira da Região S 31

4.2.4 Construção do Simplex Inicial 34

4.3 A Curva de Lissajous 37

4.3.1 Paralelismo 38

4.4 Aplicação do Método de Nelder-Mead 40

4.5 Refinamento da Solução 41

5 Experimentos Numéricos **43**
5.1 Introdução . . . 43
5.2 Exemplos de Validação . . . 44
5.2.1 2 Centros e 08 Pontos . . . 44
5.2.2 3 Centros e 12 Pontos . . . 45
5.2.3 3 Centros e 20 Pontos . . . 46
5.3 Exemplos Comparativos . . . 47
5.3.1 Recobrimento do Estado do Rio de Janeiro . . . 47
5.3.2 Recobrimento do Estado de Nova York . . . 54
5.4 Exemplos Produzidos . . . 57
5.4.1 Geração de Pontos . . . 59
5.4.2 Recobrimento do Estado do Pará . . . 60
5.4.3 Recobrimento do Brasil . . . 63

6 Conclusão **70**

7 Trabalhos Futuros **72**

Referências Bibliográficas **74**

A Programa de Otimização sem Derivadas **78**

B Programas Auxiliares **97**
B.1 **Programa Gerar Dados** . . . 97
B.2 **Programa Contar Números** . . . 101

C Recobrimento do Estado do Rio de Janeiro **102**
C.1 **Experimentos** . . . 102
C.2 **Raios de Cobertura** . . . 107

D Recobrimento do Estado de NY **108**
D.1 **Experimentos** . . . 108
D.2 **Raios de Cobertura** . . . 113

E Recobrimento do Estado do Pará **114**
E.1 **Experimentos** . . . 114
E.2 **Raios de Cobertura** . . . 119

F Recobrimento do Brasil **120**
F.1 **Experimentos** . . . 120
F.2 **Raios de Cobertura** . . . 125

Lista de Figuras

2.1 $f_j(x), \quad j = 1, 2, 3.$ 6
2.2 $\min_j f_j(x), \quad j = 1, 2, 3.$ 6
2.3 $\max_x \min_j f_j(x), \quad j = 1, 2, 3.$ 6
2.4 Exemplo de pontos e centros em uma Região S. 9
2.5 Localização dos centros e dos pontos a serem atendidos. 11

3.1 Iteração do Método de Nelder-Mead em $\Re^2$; 16
3.2 Função de Himmelblau. 17
3.3 Iterações na Função de Himmelblau. 18
3.4 Exemplos de Simplexes não degenerados. 20
3.5 Sequência dos Simplexes do exemplo de McKinnon. 27

4.1 Retangulo Mínimo de Pontos. 31
4.2 Violação de Fronteira $\Re^2$. 32
4.3 Elipse com 6 centros. 35
4.4 Exemplo de um Simplex Inicial em $\Re^2$. 36
4.5 Curva de Lissajous com $\omega_1 = 5.0$ e $\omega_2 = 6.0$. 38
4.6 Retângulo com C_1(elipse) e a Curva de Lissajous. 39
4.7 Diferentes pontos de partida do algorítmo. 39
4.8 Refinamento da melhor solução. 41
4.9 Tela do Programa. 42

5.1 Iteração Inicial e Final com 8 pontos e 2 centros. 44
5.2 Iteração Inicial e Final com 12 pontos e 3 centros. 45
5.3 Iteração Inicial e Final com 20 pontos e 3 centros. 46
5.4 Recobrimento do Estado do Rio de Janeiro obtido na literatura. . . . 47
5.5 Rio de Janeiro: C_1, Retângulo Mínimo e Curva de Lissajous. 49
5.6 Primeira iteração do método. 51
5.7 Última iteração do método. 51
5.8 Recobrimento Final Refinado do Estado do Rio de Janeiro. 52
5.9 Soluções: Final - Final Refinada - Fonte. 52
5.10 NY: C_1, Retângulo Mínimo e Curva de Lissajous. 54

5.11 NY: Primeira iteração do método. 55
5.12 NY: Solução por Nelder-Mead e solução obtida na literatura. 56
5.13 NY: Solução refinada por Nelder-Mead e solução obtida na literatura. 56
5.14 Retângulos, Triângulos e Pontos da Fronteira no Estado do Pará. . . 57
5.15 Exemplo de Retângulos e Triângulos no mapa do Brasil. 58
5.16 Geração de Pontos. 59
5.17 Pará: C_1, Retângulo Mínimo e Curva de Lissajous. 60
5.18 Primeira iteração do método. 61
5.19 Última iteração do método. 62
5.20 Brasil: C_1, Retângulo Mínimo e Curva de Lissajous. 64
5.21 Primeira iteração. 65
5.22 Última Iteração. 66
5.23 Refinamento com parâmetro de penalização $M = 0$. 67
5.24 Refinamento com parâmetro de penalização $M = 1, 2, 3$. 67

C.1 Raios de Cobertura e Coordenadas das Antenas do Rio de Janeiro. . 106
C.2 Raios de Recobrimento dos experimentos do Rio de Janeiro. 107

D.1 Raios de Cobertura e Coordenadas das Antenas de NY. 112
D.2 Raios de Recobrimento dos experimentos de NY. 113

E.1 Raios de Cobertura e Coordenadas das Antenas do Pará. 118
E.2 Raios de Recobrimento dos experimentos do Pará. 119

F.1 Raios de Cobertura e Coordenadas das Antenas do Brasil. 124
F.2 Raios de Recobrimento dos experimentos do Brasil. 125

Lista de Tabelas

4.1 Representação do Simplex Inicial. 35
4.2 Coeficientes do Método de Nelder-Mead. 40

5.1 Tabela com 8 pontos a serem recobertos. 44
5.2 Tabela com 12 pontos a serem recobertos. 45
5.3 Tabela com 20 pontos a serem recobertos. 46
5.4 Tamanhos de eixos das elipses e retângulos de Lissajous 48
5.5 Comparação das soluções. 52
5.6 Valores de f e dos maiores raios para $M = 0$, $M = 1$, $M = 2$ e $M = 3$ 68
5.7 Parâmetros das soluções do Rio de Janeiro, Nova York, Pará e Brasil. 69

C.1 Experimento a). 103
C.2 Experimento b). 103
C.3 Experimento c). 103
C.4 Experimento d). 104
C.5 Experimento e). 104
C.6 Experimento f). 104
C.7 Experimento g). 105
C.8 Experimento h). 105
C.9 Experimento i). 105
C.10 Experimento j). 106

D.1 Experimento a). 108
D.2 Experimento b). 109
D.3 Experimento c). 109
D.4 Experimento d). 109
D.5 Experimento e). 110
D.6 Experimento f). 110
D.7 Experimento g). 110
D.8 Experimento h). 111
D.9 Experimento i). 111
D.10 Experimento j). 111

E.1 Experimento a). 114
E.2 Experimento b). 115
E.3 Experimento c). 115
E.4 Experimento d). 115
E.5 Experimento e). 116
E.6 Experimento f). 116
E.7 Experimento g). 116
E.8 Experimento h). 117
E.9 Experimento i). 117
E.10 Experimento j). 117

F.1 Experimento a). 120
F.2 Experimento b). 121
F.3 Experimento c). 121
F.4 Experimento d). 121
F.5 Experimento e). 122
F.6 Experimento f). 122
F.7 Experimento g). 122
F.8 Experimento h). 123
F.9 Experimento i). 123
F.10 Experimento j). 123

Capítulo 1

Introdução

Considere o problema de minimização irrestrita:

$$\min_{x \in \Re^n} f(x) \tag{1.1}$$

onde $f : \Re^n \to \Re$.

Para o cálculo de um mínimo local do problema (1.1) onde f é continuamente diferenciável, a condição necessária de otimalidade [5] de primeira ordem exige que o gradiente da função seja o vetor nulo. Problemas deste tipo podem ser resolvidos muitas vezes com métodos clássicos como o Método do Gradiente, Newton e Quase-Newton. Entretanto, em muitas aplicações práticas, o uso de derivadas pode ser impossível, inviável, ou ainda, se f depende de uma simulação, haverá erros na avaliação da função e portanto a aproximação das derivadas por diferenças finitas, por exemplo, pode tornar o resultado pouco preciso, além de tornar o programa lento devido ao cálculo excessivo de avaliações da função a ser minimizada.[10]

Para resolver o problema de minimização (1.1) sem o cálculo de quaisquer derivadas, têm-se os métodos *derivative-free* [10]. Estes métodos diferem na maneira de usar os valores de f para calcular uma nova iteração. Dentre os métodos *derivative free* estão aqueles que constróem um modelo linear ou quadrático de f onde cada iteração é definida a partir da minimização em uma região de confiança, além dos métodos de busca padrão, de direção conjugada, *simulated annealing* e métodos *simplex* que são métodos de busca direta[1].

O mais popular dos métodos *simplex* é o Método de Nelder-Mead [26], proposto por J. A. Nelder e R. Mead em 1965. Apesar de existirem variantes deste método, foi escolhida a sua versão original pois embora estas variantes tenham teoremas de convergência, o problema (1.1) não satisfaz as hipóteses destes teoremas. Partindo de um simplex (politopo) inicial não degenerado com $n+1$ vértices em $\Re^n$, o Método

[1] E não tem relação alguma com o método de George Dantzig para problemas de Programação Linear[27].

de Nelder-Mead a cada iteração, atualiza o "pior" vértice do simplex através de reflexões, expansões, contrações ou reduções, de tal maneira que o novo simplex satisfaça uma determinada condição de descida.

A escolha do Método de Nelder-Mead para a resolução de um problema de *min-max-min* é uma nova forma de resolver este problema, visto que formas de resolução já usadas não fazem uso de métodos numéricos para a formulação exata, nem o uso de métodos derivative-free e nem mesmo de eurísticas.

O objetivo deste trabalho é uma nova forma de resolução de um problema de *min-max-min* utilizando um método de minimização sem derivadas, exemplificando com a localização de n centros de serviço dentro de uma determinada região plana S.

Uma das características do problema é a multiplicidade de mínimos locais, o método é reiniciado várias vezes em busca de um mínimo local. Para isto, é apresentada uma proposta usando os pontos de uma curva de Lissajous [16] para a construção dos pontos de partida do método.

A aplicação do Método de Nelder-Mead exige a construção de um simplex inicial com $n+1$ vértices $C_i, i = \ldots n+1$. A construção do primeiro vértice deste simplex (C_1), também é uma nova contribuição, como será mostrado no **Capítulo 4**. Os outros vértices são gerados a partir de C1.

As outras propostas novas aqui apresentadas são: a penalização da função objetivo com a intenção de colocar os centros de serviços dentro da região S, como também o de usar o Método de Nelder-Mead para resolução de um problema com restrição visto que este método é usado para problemas irrestritos.

Este trabalho está estruturado da seguinte maneira:

O **Capítulo 2** define formalmente um problema de *min-max-min*, exemplificando com o Problema de Localização de Antenas em uma região plana S, sua formulação matemática, sua aplicação na atualidade e os métodos clássicos para a resolução deste problema.

O **Capítulo 3** descreve os Métodos Simplex, onde o Método de Nelder-Mead se inclui, bem como o algorítmo, as características do simplex, as propriedades e a convergência do Método de Nelder-Mead.

O **Capítulo 4** detalha a implementação computacional do algorítmo de Nelder-Mead adaptado ao problema de *min-max-min*. Neste capítulo são construídos a função objetivo, o retângulo mínimo de pontos a serem recobertos na região S, uma penalização da função objetivo com o intuito de evitar a violação da fronteira de S, o simplex inicial, o procedimento para reiniciar o método com outros valores iniciais na tentativa de busca de um mínimo global, bem como o refinamento da solução.

O **Capítulo 5** mostra o resultado dos experimentos numéricos com três Exemplos de Validação, que demonstram a eficácia do método para problemas de pequeno

porte, onde a solução é conhecida "a priori"; dois exemplos comparativos que constam na literatura; e dois exemplos construídos com os dados do problemas gerados através de uma rotina construída para a resolução dos mesmos.

No **Capítulo 6** estão as conclusões do desempenho do método de Nelder-Mead para o problema de recobrimento obtidas através da análise dos resultados feitos com os testes numéricos mostrados no capítulo anterior.

O **Capítulo 7** apresenta propostas para trabalhos futuros, acrescentando mais restrições ao Problema de Localização de Antenas.

No **Apêndice A** está o programa usado para a resolução do problema em questão elaborado em linguagem de programação Delphi 7.0; no **Apêndice B** estão os programas auxiliares *Gerar Dados* e *Contar Números* também implementados em linguagem de programação Delphi 7.0 usados nos Exemplos Construídos; nos **Apêndices C** e **D**, encontram-se os resultados dos experimentos numéricos para os **Exemplos Comparativos** (Recobrimento dos Estados de Nova York e do Rio de Janeiro) e nos **Apêndices E** e **F** os resultados para os **Exemplos Construídos** (Recobrimento do Estado do Pará e do Brasil) respectivamente.

Capítulo 2

O Problema de Min-max-min

2.1 Introdução

Os problemas *min-max-min*

$$\min_{x \in S \subset \Re^n} f(x) = \min_{x \in S \subset \Re^n} \max_{y \in Y \subset \Re^n} \min_{z \in Z \subset \Re^n} \phi(x, y, z) \tag{2.1}$$

surgem com grande frequência na modelagem de inúmeros problemas nas mais diversas áreas, tais como: planejamento da dimensão de plantas baixas de circuitos eletrônicos, com restrições para evitar sobreposição ([9], [14], [25], [8]); determinação da movimentação de robôs evitando obstáculos no caminho ([17], [18], [33], [22]); problemas de determinação de centros, tolerâncias e ajustes ([4], [12], [40], [31], [36]), planejamento de estruturas de construção civíl sob incerteza (incertezas inerentes às resistências das estruturas dos materiais, modelos físicos, etc.) [28].

Ainda podemos citar sua vasta aplicação na localização de centros de serviço, onde o objetivo é escolher e alocar os centros dentro de uma determinada região com custo mínimo e atender a demanda dos usuários como por exemplo [21]:

- Corpo de bombeiros;
- Delegacias de Polícias;
- Hospitais;
- Seções de votação;
- Armazéns;
- Antenas, etc.

Embora não haja questionamento sobre a aplicabilidade desse tipo de formulação, a quantidade de algoritmos para sua resolução é bastante restrita; alguns dos

métodos existentes fazem uso de estratégias de buscas exaustivas ou se destinam à resolução de casos muito específicos. Como exemplos podemos citar algoritmos do tipo *cut-map* para busca de pontos viáveis em um problema com restrições max-min [40]; problemas em que a região viável pode ser representada por vértices [4]; substituição das funções a serem minimizadas por novas variáveis [18]. Alguns métodos buscam por soluções baseando-se na minimização de aproximações sucessivas suaves de $f(x)$ (sequência minimização de funções suaves que se aproximam gradualmente da função objetivo original), como por exemplo [30] e [43].

O problema (2.1) pode ser considerado um caso particular de problemas de otimização de valor ordenado (*OVO Problem* [3], [2], [1]). Existem alguns algoritmos propostos na literatura para a resolução de problemas OVO, entre os quais podemos citar: um método do tipo Cauchy, onde a cada iteração as funções que são maximizadas na composição da função objetivo são aproximadas por funções lineares [1]; em [3], foi realizada uma aproximação por funções quadráticas ao invés de lineares, gerando um método quase-Newton. Em ambos os casos é possível obter resultados teóricos de convergência global [1], e sob hipóteses ainda mais restritivas, resultados de convergência superlinear e quadrática [2]. Entretanto, essas hipóteses não são verificadas para o caso específico dos problemas *min-max-min*.

Para o problema *min-max-min* finito e semi-infinito[1] existem condições de otimalidade fortes de primeira ordem (assim como um algoritmo baseado em região de confiança que converge para pontos estacionários [32], com alto custo computacional) e um conjunto de condições de otimalidade fracas de primeira ordem (que permitem a construção de algoritmos mais eficientes, mas que convergem para pontos estacionários fracos [29]).

Uma razão para a existência de poucos algoritmos é a estrutura desse tipo de problema. Funções *max-min* são contínuas mas não-diferenciáveis, de natureza não convexa, e possuem vários minimizadores e maximizadores locais [2], [15]; logo, (2.1) é considerado um problema de difícil resolução.

Como exemplo gráfico de um problema de *max-min*, sejam

$$f_j : \Re \to \Re \qquad j = 1, 2, 3$$

e seja o problema:

$$\max_x \min_j f_j(x). \tag{2.2}$$

Gráficamente o problema pode ser visualizado nas figuras 2.1, 2.2 e 2.3 a seguir, onde estão desenhados 3 exemplos de funções com o objetivo de ver a não diferenciabilidade da função objetivo do problema.

[1]Finito quando são consideradas um número finito de funções e semi-infinito um número infinito de funções [44].

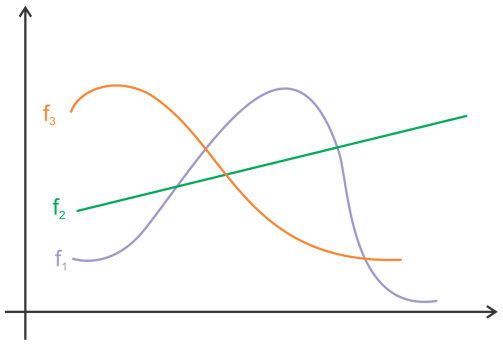

Figura 2.1: $f_j(x), \quad j = 1, 2, 3.$

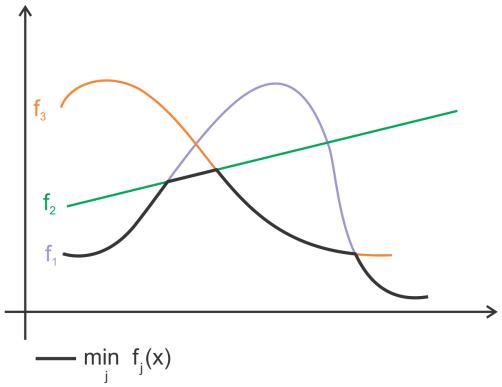

Figura 2.2: $\min\limits_{j} f_j(x), \quad j = 1, 2, 3.$

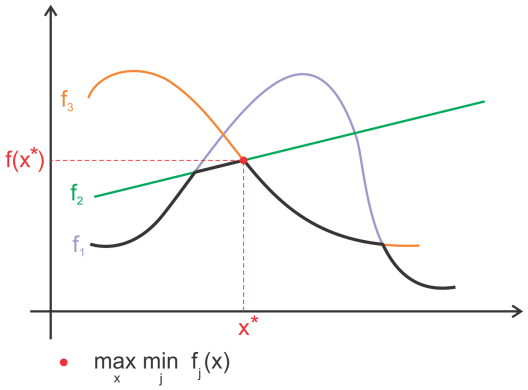

Figura 2.3: $\max\limits_{x} \min\limits_{j} f_j(x), \quad j = 1, 2, 3.$

A figura 2.3 mostra que a função $\min\limits_{j} f_j(x), \quad j = 1, 2, 3$ é definida por partes, que nos pontos extremos dos intervalos de definição da função ela é não diferenciável e com ponto de máximo em $(x^* \ , \ f(x^*))$.

No problema a ser resolvido neste trabalho, um determinado número de *max-min's* será calculado:

$$(x_1^* f(x_1^*)),\ (x_2^* f(x_2^*)),\ (x_3^* f(x_3^*)),\ \ldots$$

e o Método de Nelder-Mead irá buscar o de menor imagem.

Devido ao fato do problema (2.1) ser não-diferenciável, neste trabalho propomos um método de otimização *derivative-free* [10], o método Nelder–Mead [26] implementado com estratégias de reinício e refinamento: o método é reiniciado várias vezes em busca de um mínimo local, tendo como proposta os pontos de uma curva de Lissajous [16] para a construção dos novos pontos de partida do método. Tal método é aplicado na resolução do problema da determinação do posicionamento de um número pré-fixado de centros de serviço de tal forma que a qualidade do atendimento aos clientes (inversamente proporcional à distância entre tais centros e os clientes) seja a melhor possível.

2.2 Localização de Antenas

Há inúmeros exemplos onde o sistema de comunicação é sem fio, e há a necessidade de alocar uma ou mais estações base para a cobertura da área em questão, como em lojas de departamentos [11], onde os vendedores são munidos com equipamento portátil para atender o consumidor em qualquer lugar do prédio; ou grandes armazéns, onde os trabalhadores atualizam estoque enviando os dados para uma central através de um *palmtop* sem fio ou através de códigos de barra; ou ainda o planejamento de uma rede de telefonia celular.

O problema de localização de estações de rádio base consiste em escolher, em uma região definida, os locais onde um número determinado de estações devem ser instaladas, de forma a atender a demanda do número de pontos (usuários) da melhor maneira possível.

Um exemplo deste problema é o planejamento de uma rede de telefonia celular, que a cada ano vem apresentando um grande crescimento tanto no que diz respeito ao número de clientes como também na prestação de serviços e facilidades oferecidas aos usuários.

Para que seja permitida a comunicação entre dois telefones móveis ou entre móveis e fixos [23] são necessários:

1) uma Rede de Telefonia Pública Comutada que atende aos telefones fixos;

2) uma Rede de Telefonia Móvel, associada à Central de Controle e Comutação;

3) algumas Estações de Rádio Base (ERB's).

Vários fatores são considerados para o problema da determinação da localização das ERB's [23], dentre eles:

- Dimensão da região escolhida;
- Relevo da região;
- Interferências;
- Demanda de usuários;
- O número de ERB's;
- Potência de transmissão;
- Custo.

O objetivo do problema geral da localização e do número de ERB's é escolher e alocar dentro de uma determinada região, as antenas com custo mínimo e atender a demanda dos usuários. Estas escolhas são conflitantes, visto que, quanto menor o número de antenas instaladas, menor será o custo. Por outro lado, o atendimento da demanda e a qualidade de serviços podem ser prejudicados.
Devido às características deste problema de minimização, a determinação de uma solução exata torna-se extremamente difícil, e a solução encontrada pode ser apenas uma solução viável [42].

Obs: 2.1. *O exemplo de localização das estações base, foi apenas uma maneira de ilustrar o objetivo deste trabalho que é usar o Método de Nelder-Mead para resolver um problema de min-max-min com restrições, cuja finalidade é atender da melhor maneira possível os usuários do serviço proposto.*

Neste trabalho o objetivo é portanto determinar dentro de uma região plana as coordenadas de um número fixo pré-determinado de centros de serviços que serão chamadas de centros ou antenas, de tal maneira que a maior distância de um ponto (usuário) ao centro mais próximo seja mínima e que todos os pontos desta área sejam beneficiados (Figura 2.4), cuja modelagem inicial é proposta por exemplo em [43], como será visto na seção 2.3.

O problema da localização de estações base, por ter uma função objetivo de natureza não convexa, não diferenciável, com múltiplos mínimos locais e dependente do número de usuários a serem atendidos [15], é considerado um problema de difícil implementação.

A otimização da localização de estações base pode ser vista em duas etapas [15]. A primeira etapa consiste em construir a função objetivo de acordo com as

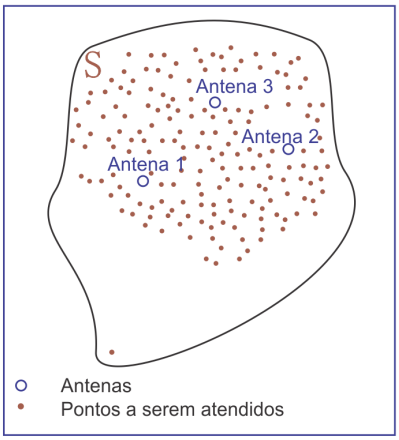

Figura 2.4: Exemplo de pontos e centros em uma Região S.

características do problema e a segunda etapa consiste em determinar uma estratégia para otimizar a localização das estações base.

Na primeira etapa, dependendo do critério a ser priorizado para a escolha da localização das estações base, ha inúmeras variantes para a abordagem do problema, com diversas formas de penalizações na função objetivo.

Na segunda etapa, várias técnicas são utilizadas para a resolução do problema, nas áreas de programação inteira, programação não linear, metaheurísticas, heurísticas e teoria dos grafos.

Neste trabalho a resolução deste problema foi feita com uma modelagem cujo objetivo é a solução de um problema de *min-max-min*, que por ser não linear, não diferenciável e de grande porte, é considerado extremamente difícil [43].

2.3 Modelagem do Problema

Seja S $\subset \Re^2$ a região onde as antenas devem ser alocadas.
Sejam:

- $s_i \in S, \;\; i = 1, ..., k$ os pontos a serem cobertos pelas centros (antenas) e
- c_i as coordenadas dos centros que beneficiarão os ponto a serem atendidos: $c_i = (x_i, y_i) \in \Re^2 \;\; i = 1, ..., n$.

O conjunto de antenas deve ser instalado de maneira a prover cada ponto desta região um bom nível de cobertura de serviços, que é inversamente proporcional à distância entre o próprio ponto e o centro mais próximo.

A construção da função objetivo do problema de *min-max-min* é feita conforme descrito a seguir.

Para cada ponto genérico s a ser recoberto, calcula-se a distância deste ponto a cada uma das antenas; dessa forma, pode-se identificar a distância $d(s, c_i)$ de cada ponto s à antena mais próxima de cada ponto.

$$d(s, c) = \min_{i=1,\ldots,n}(||s - c_i||). \tag{2.3}$$

Portanto tem-se um conjunto das n menores distâncias ligando cada ponto (usuário) à antena mais próxima.

Como a distância de cada ponto a ser recoberto pela antena é inversamente proporcional à qualidade do sinal, e como devem ser levados em consideração todos os pontos que recebem o serviço, uma medida importante é o cálculo de $D(c)$ que é a maior das menores n distâncias entre cada ponto a ser recoberto e a antena mais próxima.

$$D(c) = \max_{s \in S} d(s, c) \tag{2.4}$$

Para que a qualidade de cobertura seja a melhor, deve-se minimizar a cobertura mais crítica, levando em conta que todos os pontos devem ser cobertos, isto é, determinar as posições das antenas (n antenas) para que a maior distância $D(c)$ seja mínima, ou seja:

$$\min_{c} D(c). \tag{2.5}$$

Portanto de (2.3) e (2.4), o problema a ser resolvido é:

$$\min_{c} \max_{s \in S} \min_{i=1,\ldots,n}(||s - c_i||). \tag{2.6}$$

A figura 2.5 mostra um exemplo de uma aproximação ainda não otimizada, para a busca da melhor localização de 3 antenas em uma região S.

Neste exemplo, o ponto s é o ponto mais distante de qualquer uma das antenas, sendo este ponto mais próximo da antena 1. O objetivo é alocar as antenas de tal forma a minimizar a distância entre o ponto s e a antena 1, isto é, minimizar a maior das menores distâncias.

2.4 O Problema de min-max-min e o Método de Nelder-Mead

Várias formulações da função objetivo, com abordagens distintas e técnicas diferentes, são encontradas na literatura para a resolução do problema de alocação de antenas, dentre elas: para o problema de planejamento de uma rede de comunicação móvel usando Relaxação Lagrangeana [23]; para o problema de otimizar a eficiên-

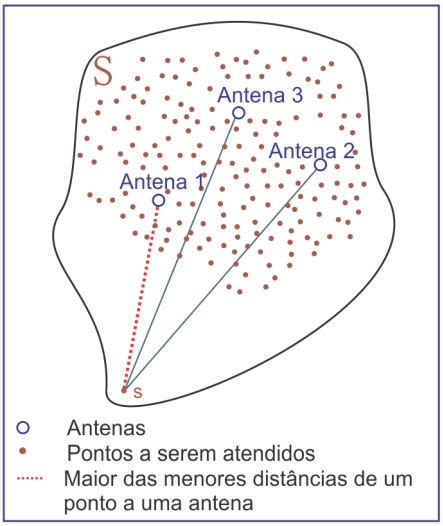

Figura 2.5: Localização dos centros e dos pontos a serem atendidos.

cia espectral de celulares "indoor" [35]; para problemas de telefonia celular usando *Simulated Annealing*, Busca Tabu e Algoritmos Genéticos [6]; para localização de estações de rádio base usando a técnica de Suavização Hiperbólica [43] e usando os algoritmos de Hooke and Jeeves, quase-Newton e Gradientes conjugados [13].

O método escolhido para resolver este problema de *min-max-min* foi o Método de Nelder-Mead, porque além de ser um dos métodos de busca direta mais utilizados em problemas de minimização sem restrições, é um método que não utiliza quaisquer derivadas ou aproximações das derivadas da função objetivo, visto que o problema é não diferenciável.

Como será visto nos capítulos a seguir, o método de Nelder-Mead aceita adaptações, que permitirão a incorporação de restrições quanto à localização das antenas.

Tem-se então o problema 2.6 sujeito $c_i \in S, \quad \forall i$. Ou seja:

$$\min_{c} \max_{s \in S} \min_{i=1,\ldots,n} (||s - c_i||)$$
$$s.a\ c_i \in S \quad \forall i$$

.

Devido à característica de múltiplos mínimos locais que o problema apresenta, o método foi reiniciado várias vezes.

O uso das Curvas de Lissajous em conjunto com o algorítmo de minimização foi realizado em [16] como uma estratégia de perturbação próxima da solução encontrada com o intuito de escapar de um mínimo local. Para este trabalho as Curvas de Lissajous serão utilizadas como forma de reinicializar o método várias vezes.

Com a melhor solução obtida foi feito o refinamento desta solução, com o intuito de obter a melhor solução possível.

A forma de tratamento das restrições, a construção dos pontos de partida do método com os pontos da Curvas de Lissajous e o refinamento da melhor solução obtida serão propostas e estudadas no **Capítulo 4**.

Capítulo 3

Método de Nelder-Mead

3.1 Métodos Simplex

Um *simplex* em $\Re^n$ é um politopo formado pelo conjunto de *n+1* pontos em $\Re^n$

$$\{x_1, x_2, \ldots, x_n, x_{n+1}\}$$

chamados vértices do *simplex*.

Os métodos *simplex* são métodos de busca direta. Estes métodos partem de um simplex inicial, e a cada iteração um novo simplex é construído depois que alguns testes são realizados com os valores da função nos vértices. Os vértices atualizados, satisfazem de alguma maneira uma condição de descida.

O primeiro método *simplex*, de acordo com Tseng [39], foi proposto por Spendley, Hext e Himsworth [34] e parte de um simplex inicial em $\Re^n$ com *n+1* vértices e a cada iteração tenta excluir o *pior* deles (onde a função tem seu maior valor), através da reflexão isométrica deste vértice em relação ao centróide dos n *melhores* vértices, gerando um novo vértice com o valor da função menor que o do vértice excluído, ou repetir a mesma operação para o segundo *pior* vértice, refletindo-o isometricamente em relação ao centróide de outros n vértices.

Se o valor da função não puder ser melhorado no *simplex* atualizado, ele então é reduzido por uma fração em torno de seu melhor vértice, e em seguida uma nova iteração é feita. Neste método, o conjunto dos ângulos internos do simplex permanece constante.

O segundo Método *Simplex* foi publicado por John Nelder e Roger Mead em 1965 [26] com o objetivo de melhorar o método anterior. Diferentemente deste, no novo método os ângulos do *simplex* podem se tornar arbitrariamente pequenos.

Mudanças neste método vêm sendo propostas desde então, assim como novos métodos baseados em *simplex*. O Método de Nelder-Mead é descrito com detalhes na próxima seção.

3.2 Método de Nelder-Mead

O método de Nelder-Mead tem sido um dos métodos de busca direta mais utilizados em problemas de minimização sem restrições de uma função de n variáveis, especialmente nas áreas de química, engenharia química e medicina [20].

A popularidade do método de Nelder-Mead deve-se a diversos fatores, dentre eles:

- a simplicidade de sua implementação computacional;
- a não necessidade do cálculo explícito ou implícito de quaisquer derivadas da função objetivo, usando apenas valores da função;
- em cada iteração, poucas avaliações da função objetivo são feitas, sendo necessário no máximo $n + 2$ avaliações da função como é descrito na seção 2.3.
- o acentuado decréscimo do valor da função objetivo nas primeiras iterações.

Obs: 3.1. *Na prática raramente ocorrem iterações com $n+2$ avaliações da função, sendo o mais comum o cálculo de uma ou duas avaliações da função por iteração.[42]*

O Método de Nelder-Mead resolve o problema (1.1) da maneira descrita a seguir.

Sejam f: $\Re^n \to \Re$ e um simplex com *n+1* vértices.

Em uma iteração do método, os *n+1* vértices do simplex, x_1, x_2, ... , x_{n+1} pertencentes a $\Re^n$, são ordenados de acordo com o crescimento dos valores de f, isto é:

$$f(x_1) \leq f(x_2) \leq \cdots \leq f(x_{n+1})$$

onde x_1 é definido como o *melhor* vértice e x_{n+1} o *pior* vértice.

O reposicionamento desses vértices leva em consideração 4 coeficientes:

- ρ - coeficiente de reflexão,
- χ - coeficiente de expansão,
- γ - coeficiente de contração,
- σ - coeficiente de redução,

os quais, de acordo com o artigo original de Nelder-Mead [26], devem satisfazer

$$\rho > 0, \quad \chi > 1, \quad 0 < \gamma < 1 \quad \text{e} \quad 0 < \sigma < 1.$$

A escolha padrão dos coeficientes é dada por

$$\rho = 1, \quad \chi = 2 \quad \gamma = \tfrac{1}{2} \quad \text{e} \quad \sigma = \tfrac{1}{2}.$$

O método tenta substituir o pior vértice do *simplex* por outro com um melhor valor. O novo vértice é obtido através da reflexão, expansão ou contração do pior vértice ao longo da reta que passa por este vértice e o centróide dos n *melhores* vértices.

A cada iteração o *pior* vértice é substituído por um novo vértice ou o simplex é reduzido em torno do *melhor* vértice.

3.3 Algoritmo de Nelder-Mead

Algoritmo 3.1. *Uma iteração do método de Nelder-Mead é feita da seguinte maneira:*

1. ***Ordenação:*** *Ordenar os n+1 vértices:* $f(x_1) \leq f(x_2) \leq \cdots \leq f(x_{n+1})$.

2. ***Centróide:*** *Calcular o centróide dos n melhores vértices;*

$$\bar{x} = \sum_{i=1}^{n} \frac{x_i}{n}.$$

3. ***Reflexão:*** *Calcular* x_r*(vértice refletido):*
   $x_r = \bar{x} + \rho(\bar{x} - x_{n+1}) = (1 + \rho)\bar{x} - \rho x_{n+1}$
   Se $f(x_1) \leq f(x_r) < f(x_n)$ *faça* $x_{n+1} = x_r$ *e encerre a iteração.*
   Ver figura 3.1 (a).

4. ***Expansão:*** *Se* $f(x_r) < f(x_1)$*, calcular* x_e*(vértice expandido):*
   $x_e = \bar{x} + \chi(x_r - \bar{x}) = \bar{x} + \rho\chi(\bar{x} - x_{n+1}) = (1 + \rho\chi)\bar{x} - \rho\chi x_{n+1}$.
   Se $f(x_e) < f(x_r)$ *faça* $x_{n+1} = x_e$ *; e encerre a iteração senão faça* $x_{n+1} = x_r$*, e encerre a iteração.*
   Ver figura 3.1 (b).

5. ***Contração:*** *Se* $f(x_r) \geq f(x_n)$

   5.1 ***Externa:*** *Se* $f(x_n) \leq f(x_r) < f(x_{n+1})$
   Calcular x_{ce} *(vértice de contração externa):*
   $x_{ce} = \bar{x} + \gamma(x_r - \bar{x}) = \bar{x} + \rho\gamma(\bar{x} - x_{n+1}) = (1 + \rho\gamma)\bar{x} - \rho\gamma x_{n+1}$
   Se $f(x_{ce}) \leq f(x_r)$ *faça* $x_{n+1} = x_{ce}$ *e encerre a iteração; senão, ir para o passo 6. Ver figura 3.1 (c).*

   5.2 ***Interna:*** *Se* $f(x_r) \geq f(x_{n+1})$
   Calcular x_{ci} *(vértice de contração interna):*
   $x_{ci} = \bar{x} - \gamma(\bar{x} - x_{n+1}) = (1 - \gamma)\bar{x} + \gamma x_{n+1}$
   Se $f(x_{ci}) < f(x_{n+1})$ *faça* $x_{n+1} = x_{ci}$ *e encerre a iteração; senão, ir para o passo 6. Ver figura 3.1 (d).*

6. ***Redução:*** *Calcular os vetores* $v_i = x_1 + \sigma(x_i - x_1),\ i = 2, \cdots, n+1$. *Os vértices (ainda não ordenados) para a próxima iteracão são:* $x_1, v_2, \cdots, v_{n+1}$. *Ver figura 3.1 (e).*

Critério de Parada

Dada uma tolerância Δ_{tol}, o Critério de Parada no artigo de Nelder-Mead [26] leva em conta o valor da função nos vértices do simplex

$$\sqrt{\sum_{i=1}^{n+1} \frac{(f(x_i) - f(\bar{x}))^2}{n}} < \Delta_{tol}.$$

A figura 3.1 mostra as possibilidades de atualização do simplex no método de Nelder-Mead em duas dimensões com a escolha padrão dos coeficientes.

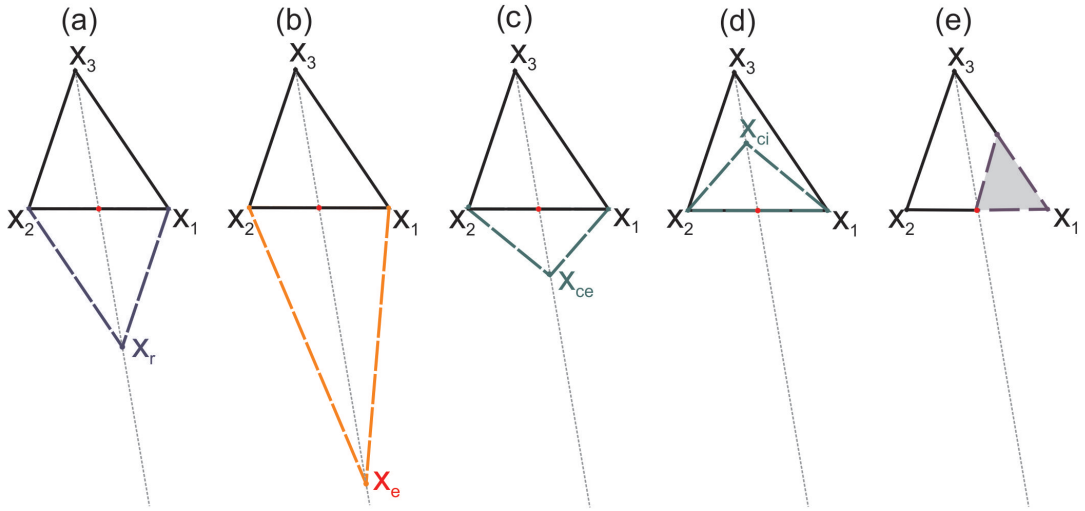

Figura 3.1: Iteração do Método de Nelder-Mead em $\Re^2$;
(a) Reflexão (b) Expansão (c) Contração Externa (d) Contração Interna (e) Redução.

No exemplo a seguir o método de Nelder-Mead é aplicado a um problema-teste ilustrando seu comportamanto.

Exemplo 3.1. *Função de Himmelblau: (ver Figura 3.2)*

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2.$$

possui quatro minimizadores globais distintos:

$f(3.0\,,\ 2.0) = 0.0$
$f(-2.805118\,,\ 3.131312) = 0.0$
$f(-3.779310\,,\ -3.283186) = 0.0$
$f(3.584428\,,\ -1.848126) = 0.0$

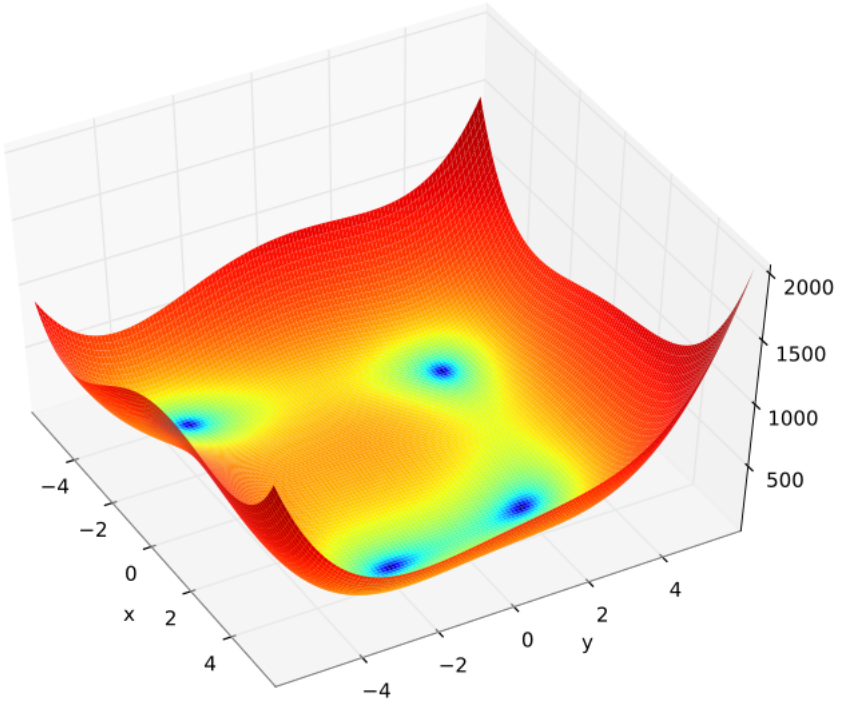

Figura 3.2: Função de Himmelblau.

A figura 3.3 mostra algumas iterações do método de Nelder-Mead para a Função de Himmelblau[1].

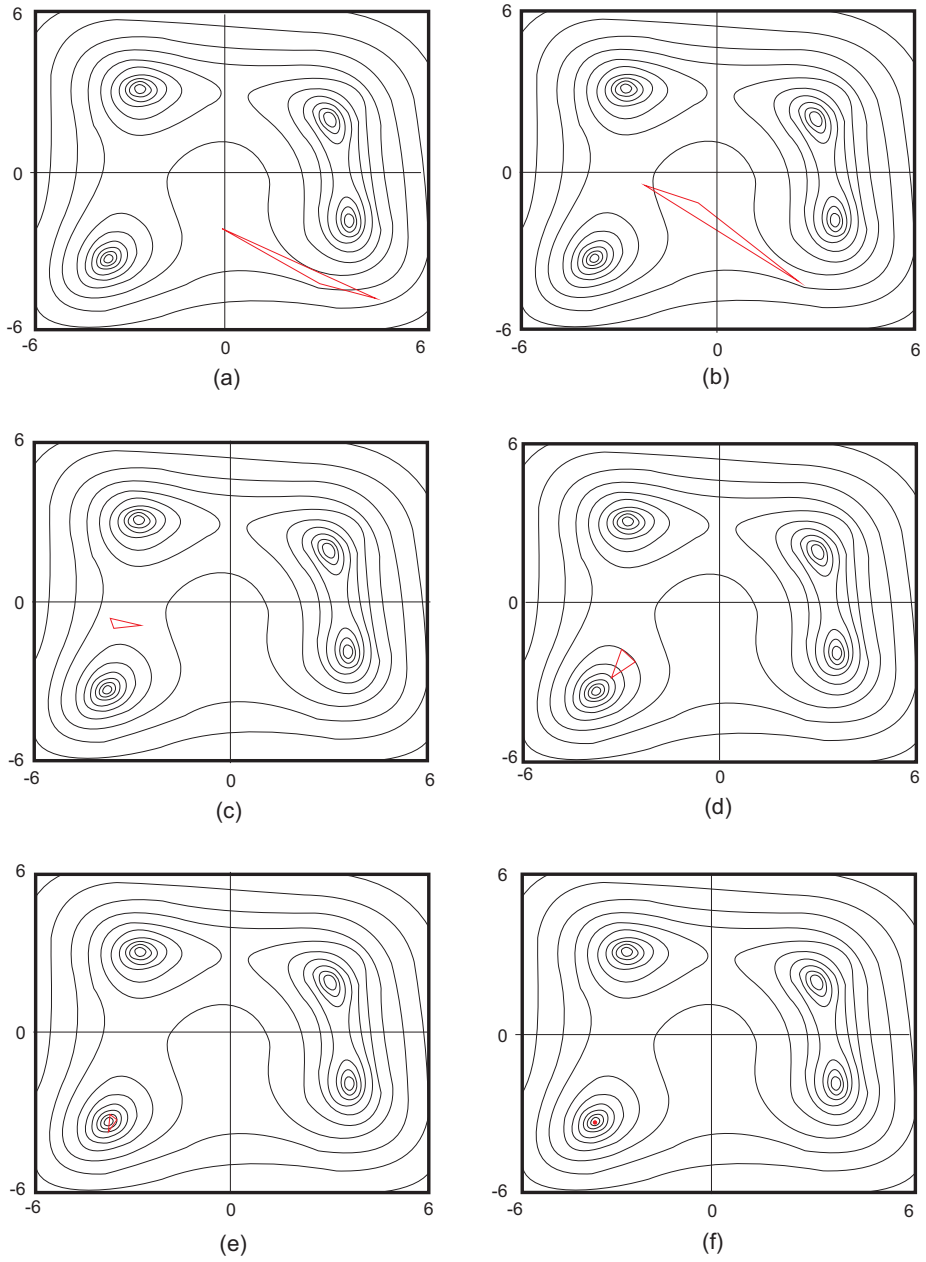

Figura 3.3: Iterações na Função de Himmelblau.

Obs: 3.2. *A Figura 3.3(a) mostra o simplex inicial mais próximo do mínimo* $(3.584428\,, -1.848126)$*, e algumas iterações até convergir para o mínimo* $(-3.779310\,, -3.283186)$ *Figura 3.3(f), o que mostra que o método não converge converge para o minimizador mais próximo do simplex inicial.*

[1]A visualização completa de como o Método de Nelder-Mead realiza suas iterações para a Função de Himmelblau pode ser encontrada em:
`http://upload.wikimedia.org/wikipedia/commons/9/96/Nelder_Mead2.gif`

3.4 Características do Simplex de Nelder-Mead

Pelo algoritmo de Nelder-Mead, o vértice alterado do simplex onde não há Redução terá uma das seguintes formas:
$x_r = \bar{x} + \rho(\bar{x} - x_{n+1})$
$x_e = \bar{x} + \rho\chi(\bar{x} - x_{n+1})$
$x_{ce} = \bar{x} + \rho\gamma(\bar{x} - x_{n+1})$
$x_{ci} = \bar{x} - \gamma(\bar{x} - x_{n+1})$.

Logo, se na iteração $k-1$ não há Redução, então o novo vértice da iteração k pode ser escrito como:

$$v^{(k)} = \bar{x}^{(k)} + \tau(\bar{x}^{(k)} - x_{n+1}^{(k)}) = (1+\tau)\bar{x}^{(k)} - \tau x_{n+1}^{(k)}, \tag{3.1}$$

onde:

- $\tau = \rho$ caso a Reflexão seja aceita;
- $\tau = \rho\chi$ caso a Expansão seja aceita;
- $\tau = \rho\gamma$ caso a Contração Externa seja aceita;
- $\tau = -\gamma$ caso a Contração Interna seja aceita.

Além da estrutura geral do vértice alterado, o método de Nelder-Mead possui outras propriedades listadas no Teorema 3.1, mostradas em [20].
Para enunciar o Teorema 3.1, considere as definições abaixo:

- Δ_k a matriz $n \times (n+1)$ é a representação matricial do simplex Δ_k cujas colunas são os vértices deste simplex:

$$\Delta_k = (x_1^{(k)} \quad x_2^{(k)} \ \ldots \ x_{n+1}^{(k)})$$

- M_k é a matriz $n \times n$ cuja j-ésima coluna representa a *face* de Δ_k entre x_j^k e x_{n+1}^k.

$$M_k \equiv (x_1^{(k)} - x_{n+1}^{(k)} \quad x_2^{(k)} - x_{n+1}^{(k)} \ \ldots \ x_n^{(k)} - x_{n+1}^{(k)}).$$

- O volume de Δ_k é dado por:

$$vol(\Delta_k) = \frac{|det(M_k)|}{n!}.$$

Por (3.1), se não há Redução na iteração k, cada vértice testado do simplex da iteração k pode ser escrito como:

$$z_k(\tau) = \Delta_k t(\tau), \quad onde \quad t(\tau) = \left(\frac{1+\tau}{n}, \ldots, \frac{1+\tau}{n}, -\tau\right)^T.$$

Definição 3.1. *Um simplex Δ_k é não degenerado se M_k é não singular, ou seja, se $vol(\Delta_k) > 0$.*

Geometricamente um simplex não degenerado é um segmento de reta sobre uma reta, um triângulo sobre um plano, um tetraedro em um espaço de três dimensões etc.
A figura 3.4 mostra exemplos de simplexes não degenerados.

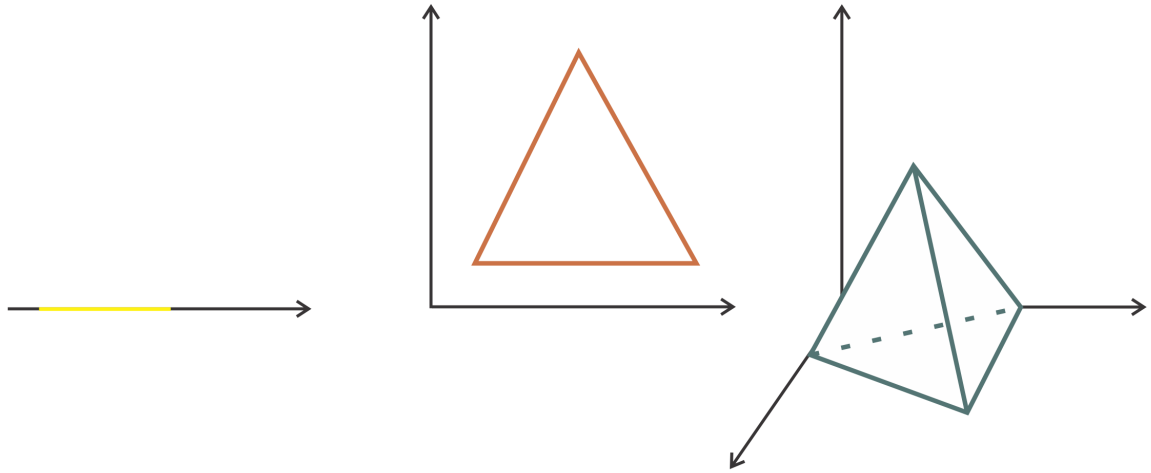

Figura 3.4: Exemplos de Simplexes não degenerados.

Teorema 3.1. *Volume e não degeneração do simplex de Nelder Mead:*

(1) Se o simplex inicial é não degenerado, então todos os outros simplex gerados pelas iterações do método de Nelder-Mead também serão não degenerados.

(2) Se não há Redução na iteração k, $vol(\Delta_{k+1}) = |\tau| vol(\Delta_k)$.

(3) Se há Redução na iteração k, $vol(\Delta_{k+1}) = \sigma^n vol(\Delta_k)$.

Demonstração. (2) Sem perda de generalidade, seja $x_{n+1}^k = 0$, isto é, o pior vértice da iteração k está na origem. Se na iteração k não há Redução, por (3.1) o novo vértice pode ser escrito como:

$$v^{(k)} = M_k w, \qquad onde \quad w = \left(\frac{1+\tau}{n}, \ldots, \frac{1+\tau}{n}\right)^T,$$

então os vértices do novo simplex Δ_{k+1} são as colunas de M_k e o vetor $M_k w$.
Como o volume de um simplex independe da ordem dos vértices, seja $v^{(k)}$ o pior

vértice de Δ_{k+1}.
Então:

$$|det(M_{k+1})| = |det(M_k - M_k we^T)| = |det(M_k)|\,|det(I - we^T)|,$$

onde $e = (1, \ldots, 1)^T$.
A matriz $I - we^T$ tem $n-1$ autovalores iguais a 1 e um autovalor igual a

$$1 - w^T e = -\tau.$$

Como o determinante de uma matriz é o produto de seus autovalores,

$$det(I - we^T) = -\tau.$$

Logo,

$$|det(M_{k+1})| = |-\tau|\,|det(M_k)| \quad \Rightarrow \quad vol(\Delta_{k+1}) = |\tau| vol(\Delta_k).$$

(3) Se na iteração k há Redução, então cada vértice do simplex é multiplicado por σ. Assim

$$|det(M_{k+1})| = |det(\sigma M_k)| = \sigma^n |det(M_k)|.$$

(1) Um simplex é não degenerado se $vol(\Delta_k) \neq 0$
Como $\tau \neq 0$ e $\sigma \neq 0$, de (2) e (3) segue que se $vol(\Delta_0) \neq 0$, o volume dos outros simplex gerados pelo método de Nelder-Mead também serão não degenerados.

□

3.5 Propriedades do Método

Nesta seção são apresentadas várias propriedades importantes do método de Nelder-Mead que podem ser vistas por exemplo em [10].

Do algoritmo 3.1, segue trivialmente que :

- A cada iteração do Método de Nelder-Mead, o número de avaliações da função necessárias são:
  a) 1 caso a Reflexão seja aceita;
  b) 2 caso a Expansão ou Contração sejam aceitas;
  c) $n+2$ caso a Redução seja aceita.
- Se a Redução é aceita o valor da função nos vértices x_i, $i = 2 \ldots n+1$, pode aumentar.

Definição 3.2. *Uma função f é estritamente convexa em $\Re^n$ se, para todo par de pontos x, y, com $x \neq y$ e para todo λ, $0 < \lambda < 1$ vale a desigualdade:*

$$f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda) f(y). \tag{3.2}$$

Se f é estritamente convexa em $\Re^n$ e

$$c = \sum_{i=1}^{k} \lambda_i \, x_i, \quad com \quad 0 < \lambda_i < 1 \quad e \quad \sum_{i=1}^{k} \lambda_i = 1,$$

então

$$f(c) < \sum_{i=1}^{k} \lambda_i \, f(x_i), \quad logo \quad f(c) < max\{f(x_i), ..., f(x_k)\}. \tag{3.3}$$

Teorema 3.2. *Se f é estritamente convexa e se o Método de Nelder-Mead parte de um simplex inicial não degenerado, então não há Redução nas iterações do método.*

Demonstração. Em uma iteração do método, a Redução só ocorre quando o ponto de Contração Externa x_{ce} ou ponto de Contração Interna x_{ce} não são aceitos.
O Centróide $\bar{x}$ dos n melhores vértices é definido como:

$$\bar{x} = \sum_{i=1}^{n} \frac{x_i}{n}.$$

Se $n = 1, \quad f(\bar{x}) = f(x_n)$.
Se $n > 1$:

$$\bar{x} = \sum_{i=1}^{n} \lambda_i \, x_i, \quad com \quad \lambda_i = \frac{1}{n} \quad \forall i.$$

Então, por (3.3):

$$f(\bar{x}) < \sum_{i=1}^{n} \frac{1}{n} f(x_i) \quad logo, \quad f(\bar{x}) < max\{f(x_i), ..., f(x_n)\};$$

portanto $f(\bar{x}) < f(x_n)$.
Se x_r é o ponto de reflexão e $f(x_n) \leq f(x_r) < f(x_{n+1})$, é feita a Contração Externa.
O ponto de Contração Externa x_{ce} dado por:

$$x_{ce} = (1 + \rho\gamma)\bar{x} - \rho\gamma x_{n+1} = \bar{x} + \gamma(x_r - \bar{x}), \quad 0 < \gamma < 1$$

é uma combinação linear do Centróide $\bar{x}$ e do ponto refletido x_r, logo, por (3.2):

$$f(x_{ce}) < max\{f(\bar{x}), f(x_r)\}.$$

Como $f(\bar{x}) \leq f(x_n)$ e $f(x_n) \leq f(x_r)$,

$$max\{f(\bar{x}), f(x_r)\} = f(x_r).$$

Logo: $f(x_{ce}) < f(x_r)$.
Portanto o ponto refletido x_{ce} é aceito e não há Redução.
O mesmo argumento vale para a Contração Interna, visto que $f(x_{n+1}) < f(x_r)$ e o ponto de Contração Interna x_{ci} é uma combinação convexa de $\bar{x}$ e x_{n+1}. □

Definição 3.3. *Um índice k^* é chamado de índice de mudança da iteração k se é o menor índice de um vértice que se diferencia entre a iteração k e $k+1$. Isto é:*

$$k^* = min\{i \,|\, x_i^{(k)} \neq x_i^{(k+1)}\}.$$

Por exemplo, se há Expansão na iteração k, $k^* = 1$, porque o vértice substituído da iteração k para a iteração $k+1$ é o vértice de índice 1, x_1.

Se em uma iteração não há Redução:

$$f(x_j)^{(k+1)} = f(x_j)^{(k)} \quad e \quad x_j^{(k+1)} = x_j^{(k)}, \quad j < k^*; \tag{3.4}$$

$$f(x_{k^*})^{(k+1)} < f(x_{k^*})^{(k)} \quad e \quad x_{k^*}^{(k+1)} \neq x_{k^*}^{(k)}; \tag{3.5}$$

$$f(x_j)^{(k+1)} = f(x_{j-1})^{(k)} \quad e \quad x_j^{(k+1)} = x_{j-1}^{(k)}, \quad j > k^*. \tag{3.6}$$

Portanto se não há Redução, o vetor $(f(x_1), \ldots, f(x_{n+1}))$ é lexicograficamente estritamente decrescente, isto é:

$$\sum_{i=1}^{n+1} f(x_i)^{(k+1)} < \sum_{i=1}^{n+1} f(x_i)^{(k)}.$$

Teorema 3.3. *Se f é uma função limitada inferiormente em $\Re^n$ e se o método de Nelder-Mead parte de um simplex inicial não degenerado, então:*

(1) a sequência $\{f(x_1)^{(k)}\}$ é convergente;

(2) em toda iteração k onde não há Redução, $f(x_i)^{(k+1)} \leq f(x_i)^{(k)}$, $1 \leq i \leq n+1$, com desigualdade estrita para pelo menos um valor de i;

(3) se o número de Reduções é finito, então:

(3.1) cada sequência $\{f(x_i)^{(k)}\}$ converge quando $k \to \infty$, $1 \leq i \leq n+1$;

(3.2) $f_i^* \leq f(x_i)^{(k)}$, $\forall k$, $1 \leq i \leq n+1$, onde $f_i^* = \lim_{k\to\infty} f(x_i)^{(k)}$;

(3.3) $f_1^* \leq f_2^* \leq \cdots \leq f_{n+1}^*$.

(4) se há somente um número finito de iterações sem Redução, então todos os vértices do simplex convergem para um único ponto.

Demonstração. (1) Se não há Redução, $f(x_1)^{(k+1)} \leq f(x_1)^{(k)}$.
Se há Redução, o único vértice mantido no novo simplex é x_1, então

$f(x_1)^{(k+1)} = f(x_1)^{(k)}$. Logo a sequência é monótona e decrescente.
Como f é limitada inferiormente, então a sequência $\{f(x_1)^{(k)}\}$ é monótona, decrescente e limitada, logo convergente.

(2) Se não há Redução na iteração k, então $\exists\, x_j$, o novo vértice do simplex para a iteração $k+1$, tal que:

- Se a Expansão é aceita:
$$f(x_j)^{(k)} \;<\; f(x_1)^{(k)}.$$

- Se a Reflexão, Contração Externa, Contração Interna são aceitas:
$$f(x_1)^{(k)} \leq f(x_j)^{(k)} < f(x_n)^{(k)}.$$

  Então:

- Se a Expansão é aceita:

$$\begin{aligned} f(x_i)^{(k+1)} &< f(x_i)^{(k)}, \qquad i = j; \\ f(x_i)^{(k+1)} &\leq f(x_i)^{(k)}, \qquad j < i \leq n+1. \end{aligned}$$

- Se a Reflexão, Contração Externa ou Contração Interna são aceitas:

$$\begin{aligned} f(x_i)^{(k+1)} &= f(x_i)^{(k)}, \qquad 1 \leq i \leq j \\ f(x_i)^{(k+1)} &\leq f(x_i)^{(k)}, \qquad j < i \leq n \\ f(x_i)^{(k+1)} &< f(x_i)^{(k)}, \qquad i = n+1. \end{aligned}$$

(3) A prova deste ítem está no fato de que a cada iteração os vértices são ordenados de forma que $f(x_1) \leq\; f(x_2) \leq \cdots \leq f(x_{n+1})$ e como f é limitada inferiormente, as sequências são monótonas decrescentes e limitadas, logo são convergentes.

(4) Se há somente um número finito de iterações sem Redução, então, pelo Teorema 3.1, quando há Redução na iteração $k-1$, na iteração k:

$$vol(\Delta_k) = \sigma^n vol(\Delta_{k-1}),$$

como $0 < \sigma < 1$,

$$vol(\Delta_k) \to 0 \; quando \; k \to \infty,$$

logo,

$$|det(M_k)| \to 0,$$

ou seja, as *faces* de Δ_k entre x_j^k e x_{n+1}^k tenderão a zero, logo todos os vértices do simplex convergem para um único ponto. □

Obs: 3.3. *A convergência para um único ponto não implica que este ponto seja um ponto estacionário.*

Teorema 3.4. *Seja f é uma função limitada inferiormente em $\Re^n$. Se o método de Nelder-Mead parte de um simplex inicial não degenerado, se não há Reduções, e se existe um inteiro j, $1 \leq j \leq n$, para o qual*

$$f_j^* \leq f_{j+1}^*, \qquad onde \qquad f_j^* = \lim_{k\to\infty} f(x_j)^{(k)}, \tag{3.7}$$

então existe uma iteração com índice K tal que para todo $k \geq K$, o índice de mudança satisfaz:

$$k^* > j, \tag{3.8}$$

isto é, os j primeiros vértices de todos os simplex permanecem fixos após a iteração K.

Demonstração. Provando por contradição.
Pela hipótese (3.7), $f_j^* + \delta = f_{j+1}^*$ para algum $\delta > 0$.
Seja $\epsilon > 0$ tal que $\delta - \epsilon > 0$.
Como $f_j^* = \lim_{k\to\infty} f(x_j)^{(k)}$, existe K tal que para todo $k \geq K$, $f(x_j)^{(k)} - \epsilon \leq f_j^*$.
Então, para todo $k \geq K$,

$$f(x_j)^{(k)} < f(x_j)^{(k)} - \epsilon + \delta \leq f_j^* + \delta = f_{j+1}^*.$$

Mas pelo Teorema 3.3, item 3, para algum índice l , $f_{j+1}^* \leq f_{j+1}^{(l)}$. Então para todo $k \geq K$ e algum índice l ,

$$f(x_j)^{(k)} < f_{j+1}^{(l)}, \tag{3.9}$$

Mas se $k^* \leq j$ para algum $k \geq K$, então (como não há Redução), por (3.6):

$$f(x_{j+1})^{(k+1)} = f(x_j)^{(k)} \quad e \quad x_j^{(k+1)} = x_{j-1}^{(k)}, \quad j > k^*$$

o que contradiz (3.8).
Logo $k^* > j$ para todo $k \geq K$. □

Corolário 3.1. *Seja f uma função limitada inferiormente em $\Re^n$. Supondo que o método de Nelder-Mead parte de um simplex inicial não degenerado e que não há Reduções; se o índice de mudança é 1 infinitamente, então $f_1^* = \ldots = f_{n+1}^*$.*

Demonstração. Se $k^* = 1$ em todas as iterações e não há Redução, então:

$$f(x_1)^{(k+1)} < f(x_1)^{(k)} \quad e \quad x_1^{(k+1)} \neq x_1^{(k)}, \quad \text{e}$$
$$f(x_i)^{(k+1)} = f(x_{i-1})^{(k)} \quad e \quad x_i^{(k+1)} = x_{i-1}^{(k)}, \quad \text{i} > 1$$

Como

$$f_i^* = \lim_{k\to\infty} f(x_i)^{(k)} \quad e \quad f_{i-1}^* = \lim_{k\to\infty} f(x_{i-1})^{(k)},$$

então:

$$f_i^* = f_{i-1}^*, \quad \text{i} > 1$$

Portanto:

$$f_1^* = \ldots = f_{n+1}^*$$

□

As propriedades do método de Nelder-Mead não garantem a convergência do método para problemas com dimensão maiores que $n = 1$, como será visto na próxima seção, com exemplos ilustrativos que constam na literatura.

3.6 Convergência do Método

No artigo de Nelder-Mead [26] publicado em 1965 a convergência do método não é demonstrada.

Para testar a convergência do Método, os autores apresentaram com êxito uma aplicação do método em três conhecidas funções. Estas funções, todas com ponto de mínimo zero são:

(1) Rosembrock's parabolic valley (Rosembrock (1960))
$y = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$
ponto inicial: $(-1.2, 1)$

(2) Powell's quartic function (Powell (1962))
$y = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$
ponto inicial: $(3, -1, 0, 1)$

(3) Fletcher and Powell's helical valley (Fletcher e Powell (1963))
$y = 100[x_3 - 100\theta(x_1, x_2)]^2 + [\sqrt{x_1^2 + x_2^2} - 1]^2 + x_3$

onde: $2\pi\theta(x_1, x_2) = \begin{cases} \arctan(x_2/x_1) & x_1 > 0 \\ \pi + \arctan(x_2/x_1) & x_1 < 0 \end{cases}$
ponto inicial: $(-1, 0, 0)$

Em 1989, Virginia Torczon [37], modifica o método de Nelder-Mead fazendo reflexões e possíveis expansões ou contrações dos n piores vértices em relação ao melhor vértice. Se o melhor vértice do simplex anterior não for melhorado, então é feita a redução em torno do melhor vértice.

Em 1991, a autora demonstra em [38] a convergência global de seu método para funções continuamente diferenciáveis e limitadas inferiormente.

Em 1998, McKinnon em [24] provou teoricamente para uma classe de funções estritamente convexas que o método de Nelder-Mead converge para um ponto não-estacionário.

Um caso particular nesta classe de funções é a função definida em $\Re^2$, dada por:

$$f(x_1, x_2) = \begin{cases} 360x_1^2 + x_2 + x_2^2 & x \leq 0, \\ 60x_1^2 + x_2 + x_2^2 & x_1 > 0. \end{cases}$$

Com valores iniciais:

$$x_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad x_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad x_3 = \begin{bmatrix} (1+\sqrt{33})/8 \\ (1-\sqrt{33})/8 \end{bmatrix}$$

o método de Nelder-Mead converge para o ponto $(0, 0)$ com $f(x) = 0$, sendo minimizador da função $(0, -0.5)$ com $f(x) = -0.25$, como mostra a Figura 3.5.

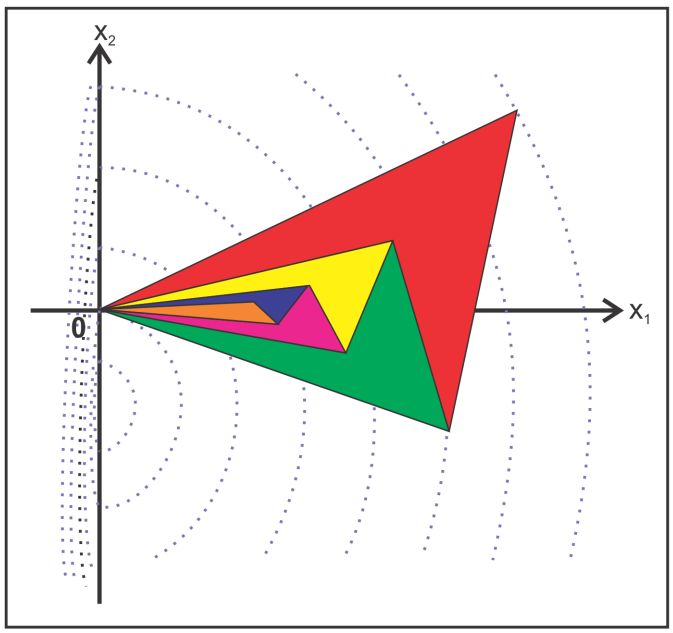

Figura 3.5: Sequência dos Simplexes do exemplo de McKinnon.

Entretanto, é interessante observar que na prática, o método foi programado em Delphi 7.0 e, sem usar qualquer critério de parada, foi capaz de sair da origem e

convergir para o mínimo da função com 1336 iterações e 703 avaliações da função, usando os valores iniciais acima.

A justificativa, em princípio, deve-se ao fato da propagação de erros causados pelo computador, visto que quando o valor de x_3 foi aproximado:

$$x_3 = \begin{bmatrix} (1+\sqrt{33})/8 \\ (1-\sqrt{33})/8 \end{bmatrix} \quad \cong \begin{bmatrix} 0.84 \\ -0.59 \end{bmatrix}$$

o método convergiu com menos iterações.

Ainda em 1998, em [20], Jeffrey C. Lagarias, James A. Reeds, Margaret H. Wright e Paul E. Wright provaram a convergência do método para dimensão 1.

Desde então inúmeros artigos foram publicados, modificando o método original, impondo condições tanto em relação à função, quanto ao simplex (volume e ângulos), para provar a convergência do Método de Nelder-Mead modificado para um ponto estacionário com dimensões maiores que 1.

Recentemente, em [19] é considerada uma variante do algorítmo de Nelder-Mead, que não permite a Expansão. Este artigo mostra que em duas dimensões, partindo de um simplex não degenerado, para qualquer função duas vezes continuamente diferenciável, com Hessiana positiva definida e com conjuntos de nível limitado, o algoritmo sempre converge para o minimizador.

Apesar do método de Nelder-Mead não apresentar garantia de convergência para um mínimo local, ele é amplamente usado pelo fato de atender a uma ampla classe de funções onde o uso de derivadas não é permitido. Além disso, na prática o algoritmo converge para a grande maioria dos problemas testados na literatura.

No próximo capítulo é feita uma descrição do problema de *min-max-min*, para o qual o método Nelder-Mead será implementado.

Capítulo 4

Implementação Computacional

4.1 Introdução

Os problemas de localização de antenas se caracterizam pelo número elevado de mínimos locais. Além disso, é conhecido que o Método de Nelder-Mead não leva necessariamente a um ponto estacionário, como mostra o exemplo de McKinnon em duas variáveis.

Dessa forma, para alocar n antenas, partindo-se de um simplex inicial com $2n+1$ vértices, cada um com as $2n$ coordenadas dos n centros (antenas), nada garante que o método chegará ao mínimo global, e nem mesmo a um mínimo local.

Apesar destas limitações, o Método de Nelder-Mead é muito eficiente no que diz respeito à significante melhora que ele produz nas primeiras iterações [42], além de sua eficiência em se tratando de uma função cuja avaliação é lenta.

Várias questões deverão ser consideradas para a resolução deste problema de *min-max-min* usando o Método de Nelder-Mead tais como:

- a construção da função objetivo para o problema;
- a construção da Região S;
- a garantia de que os centros (antenas) estejam localizados dentro da região S;
- a construção do simplex inicial;
- o tamanho do simplex inicial, que é determinante para a eficiencia do método.
- uma técnica para recomeçar o método, partindo sempre de pontos iniciais distintos.

Seja C o vetor com as coordenadas das n antenas.

$$C = (c_1, \ldots, c_n)^T,$$

onde c_i corresponde às coordenadas (x_i, y_i) da antena i $\quad$ i=1, . . . , n.

A dimensão do vetor C é igual a 2 vezes o número de antenas; portanto a dimensão do problema em questão é $2n$.

4.2 Implementação do Método para o Problema

4.2.1 Implementação da Função Objetivo

São dados k pontos fixos pontos, $s_j \in \Re^2 \;\; j = 1, ..., k$, situados numa região plana, a serem recobertos por centros (antenas) em n pontos dessa região.

Trata-se de determinar as $2n$ coordenadas $c_i = (x_i, y_i) \in \Re^2 \;\; i = 1, ..., n$, de n centros a serem colocados na região S. Pelo Método de Nelder-Mead em um problema de dimensão $2n$ é construído um simplex com $2n + 1$ vértices, e cada vértice terá sua imagem aferida pela função objetivo (2.6)

Durante a implementação, esses $2n+1$ vértices são armazenados em uma matriz

$$Centros = [\, C_1 \quad C_2 \quad \ldots \quad C_{2n+1} \,]^T \tag{4.1}$$

onde cada C_i , $i = 1, ..., 2n + 1$, é um vetor com as coordenadas de n centros, ou seja, a matriz $Centros$ é uma matriz $(2n + 1) \times 2n$.

Os pontos $s_j \in \Re^2 \;\; (j = 1, ..., k)$ a serem recobertos pelas antenas também são armazenados matricialmente na matriz $Dados$

$$Dados = [s_1^T \vdots s_2^T \vdots \cdots \vdots s_k^T]^T$$

portanto a $Dados$ é uma matriz $k \times 2$.

Os dados para a construção da função objetivo foram assim implementados para solucionar o problema

$$\min_{c} \max_{j=1,...,k} \min_{i=1,...,n} d_{ij} \tag{4.2}$$

4.2.2 Retângulo Mínimo de Pontos na Região S

Dados os k pontos pertencentes à região S que contém os pontos a serem recobertos pelos sinais das antenas, pode-se definir:

- $MaxX$ e $MinX$ a maior e menor abcissas pertencentes a esta região;
- $MaxY$ e $MinY$ a maior e menor ordenadas pertencentes a esta região.

A região S fica inserida em um Retângulo Mínimo de Pontos, cujos lados são paralelos aos eixos X e Y, com tamanhos $TamanhoX$ e $TamanhoY$, onde:
$TamanhoX = MaxX - MinX$
$TamanhoY = MaxY - MinY$

No centro do retângulo são traçados dois novos eixos X^* e Y^* paralelos aos eixos cartesianos X e Y, de maneira que a intercecção dos eixos X^* e Y^* fique posicionado no centro do Retângulo Mínimo de Pontos conforme mostra a Fig. 4.1. O objetivo da construção do Retângulo Mínimo de Pontos bem como o deslocamento dos eixos é devido à construção do simplex inicial do Método de Nelder- Mead, como será descrito na seção 4.2.4

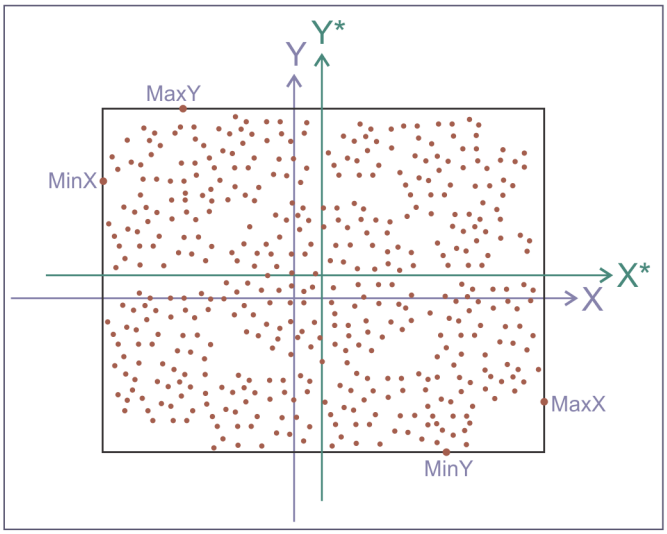

Figura 4.1: Retangulo Mínimo de Pontos.

4.2.3 Violação da Fronteira da Região S

Nada impede que a solução do problema de localização dos centros em uma determinada região S tenha como solução um ou mais centros além da fronteira da região S.
Para resolver este problema é possível:

a) remover, se possível, a restrição de limite;

b) atribuir ao vértice fora dos limites um valor de imagem muito grande, o que automaticamente forçaria uma operação de Contração ou Redução durante a execução de Nelder-Mead.

Entretanto, para resolver a questão de violação de fronteira em b) só é viável se a Região S for um hipercubo [41], ver figura 4.2, o que não é o caso do problema de *min-max-min* aqui proposto visto que a região a ser recoberta é sempre uma região geográfica.

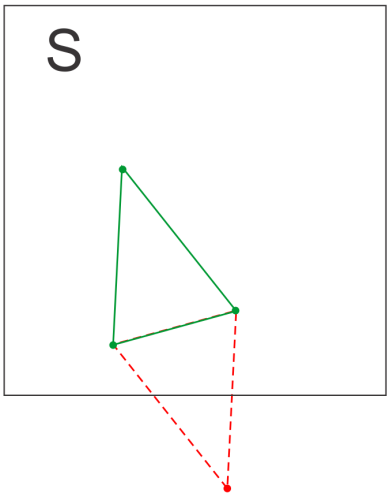

Figura 4.2: Violação de Fronteira $\Re^2$.

Para resolver o problema de uma ou mais antenas fora da região S, foi construída uma função de penalização, com o objetivo de colocar os centros dentro da região S. Esta função é adicionada à função objetivo multiplicada por um parâmetro de penalização M.

Esta função é construída de maneira a mensurar a distância dos centros aos pontos a serem recobertos, e portanto ao minimizá-la os centros fora da fronteira são postos dentro da região S, ou seja:

1. Para cada centro $c_i = (x_i, y_i) \in \Re^2$ $i = 1, ..., n$ calcula-se o quadrado da distância desse centro a cada ponto dado:

$$dc_{ij} = (x_i - sx_j)^2 + (y_i - sy_j)^2 \quad j = 1, ..., k \tag{4.3}$$

2. Para cada centro calcula-se o menor dos quadrados dessas distâncias, locali-

zando assim o ponto a ser recoberto mais próximo do centro c_i.

$$dc_i^* = \min_{j=1,\ldots,k} dc_{ij} \tag{4.4}$$

obtendo-se n dessas menores distâncias ao quadrado;

3. Determina-se o maior valor desses n valores:

$$Dc_{max} = \max_{i=1,\ldots,n} dc_i^*, \tag{4.5}$$

isto é:

$$Dc_{max} = \max_{i=1,\ldots,n} \min_{j=1,\ldots,k} dc_{ij} \tag{4.6}$$

O problema de minimização (2.6) passou a ser:

$$\min_c \left\{ \left[\max_{j=1,\ldots,k} \min_{i=1,\ldots,n} d_{ij} \right] + M \left[\max_{i=1,\ldots,n} \min_{j=1,\ldots,k} dc_{ij} \right] \right\} \tag{4.7}$$

ou:

$$\min_c \left\{ \left[\max_{s\in S} \min_{i=1,\ldots,n} (||s - c_i||) \right] + M \left[\max_{s\in S} \min_{j=1,\ldots,k} (||s_j - c||) \right] \right\} \tag{4.8}$$

onde M é o parâmetro de penalização.

A modelagem aqui proposta corresponde a uma generalização da formulação do problema de recobrimento 2.6, incorporando a restrição de fronteira sem a necessidade de dados adicionais.

Obs: 4.1. *O problema (4.8) é portanto uma função que minimiza:*

- *a maior das menores distâncias de cada ponto a ser recoberto a cada centro:*

$$\left[\max_{s\in S} \min_{i=1,\ldots,n} (||s - c_i||) \right],$$

- *acrescida da função de penalização, que calcula a maior das menores distâncias de cada centro a cada ponto a ser recoberto, multiplicada pelo parâmetro M:*

$$M \left[\max_{s\in S} \min_{j=1,\ldots,k} (||s_j - c||) \right].$$

4.2.4 Construção do Simplex Inicial

O simplex inicial possui seus dados armazenados na matriz *Centros* (4.1) e possui uma construção especialmente desenvolvida para este problema:

$$Centros = [C_1 \quad C_2 \quad \dots \quad C_{2n+1}]^T$$

Construção do Primeiro conjunto de Centros - C_1

Seja

$$C_1 = (C_{11}, C_{12}, C_{13}, C_{14}, \cdots, C_{12n-1}, C_{12n})$$

o primeiro conjunto de centros a ser fornecido como primeiro vértice do simplex inicial para a execução do Nelder-Mead. Como uma proposta deste trabalho, todos os centros de C_1 ficarão sobre uma elipse, cujo centro está sobre a origem dos eixos X^* e Y^*, e os eixos da elipse estão sobre os dois eixos X^* e Y^*, sendo os tamanhos dos semi-eixos da elipse dados por $(TamanhoX)/K_e$ e $(TamanhoY)/K_e$ respectivamente, com K_e um fator escolhido adequadamente.

Na determinação das posições de cada centro i, (C_{12i-1}, C_{12i}) $\quad i = 1 \cdots n$, do primeiro conjunto de centros C_1 é feito conforme o algoritmo a seguir.

Algoritmo 4.1. ***Construção do Primeiro conjunto de Centros - C_1***

1. *Dados iniciais:* $\alpha = 0, \quad i = 1, \quad n =$ *número de centros.*
2. $C_{12i-1} = \cos\alpha$
3. $C_{12i} = \sin\alpha$
4. $\alpha = \alpha + 2\,\pi/n$
5. $i = i + 1$
6. *Se* $i \leq n$ *volte para 2.*
7. *FIM*

O primeiro centro (C_{11}, C_{12}) é posicionado sobre a parte positiva do eixo X^*, distante $(TamanhoX)/K_e$ da origem.

Os demais $n-1$ centros serão determinados girando, sucessivamente, de um ângulo α o raio que liga cada centro ao centro da elipse, com $\alpha = 2\pi/n$, sendo assim gerados os n centros inicialmente propostos, como uma das $2n+1$ propostas iniciais necessárias para aplicação do Método de Nelder-Mead.

A Figura 4.3 mostra um exemplo do primeiro conjunto de centros C_1 com 6 centros sobre a elipse.

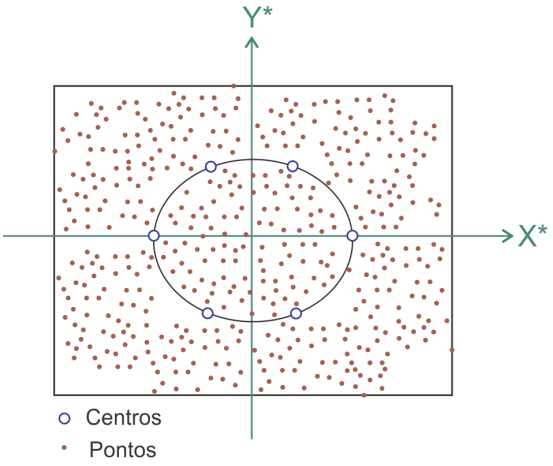

Figura 4.3: Elipse com 6 centros.

Construção dos demais conjunto de centros $\mathbf{C}_i's$, $i = 2 \ldots 2n+1$:

Construído o primeiro conjunto de centros C_1, os outros $2n$ conjuntos de centros para formar o primeiro simplex do Método de Nelder-Mead foram construídos em função de C_1.

A tabela 4.1 mostra as coordenadas C_i's do Simplex Inicial citado em [41], apresentado inicialmente por Spendley, Hext e Himsworthem [34]. Com esta construção, o simplex gerado é regular se o passo s_j for constante, satisfazendo portanto a exigência da não degeneração do simplex inicial do algotitmo de Nelder-Mead, além de gerar automaticamente os outros vértices a partir de C_1, o que facilitará o reinício do método como será visto em 4.3.

Tabela 4.1: Representação do Simplex Inicial.

Vértices	Coordenada 1	Coordenada 2	Coordenada 3	. . .	Coordenada 2n
C_1	C_{11}	C_{12}	C_{13}	. . .	C_{12n}
C_2	$C_{11} + p_1$	$C_{12} + q_2$	$C_{13} + q_3$	. . .	$C_{12n} + q_{2n}$
C_3	$C_{11} + q_1$	$C_{12} + p_2$	$C_{13} + q_3$	. . .	$C_{12n} + q_{2n}$
C_4	$C_{11} + q_1$	$C_{12} + q_2$	$C_{13} + p_3$	. . .	$C_{12n} + q_{2n}$
.	.	.	.	. . .	.
.	.	.	.	. . .	.
.	.	.	.	. . .	.
.	.	.	.	. . .	.
C_{2n+1}	$C_{11} + q_1$	$C_{12} + q_2$	$C_{13} + q_3$	. . .	$C_{12n} + p_{2n}$

Nesta tabela,

- s_i é o tamanho do passo para a coordenada i ;
- $p_i = s_i\ [\sqrt{2n+1} + 2n - 1]\ /\ 2n\sqrt{2}$;

- $q_i = s_i\ [\sqrt{2n+1}$ - 1] / $2n\sqrt{2}$.

Neste trabalho $s_i = s$, ou seja, o tamanho do passo permaneceu fixo para todas as $2n$ variáveis de cada conjunto de centro C_i, $i = 2, \ldots, 2n+1$.
A escolha de s foi obtida testando seu valor em experimentos diferentes, e então foi escolhido o melhor tamanho do simplex para o problema em questão como mostra as tabelas nos Apêndices C, D, E e F.

A figura 4.4 mostra um Simplex Inicial em $\Re^2$ com vértice inicial (20.0, 20.0) e passo 60.0. Logo, $p_1 = 57.96$ e $q_1 = 15.59$, obtendo assim os outros dois vértices (77.96, 35.59) e (35.59, 77.96).

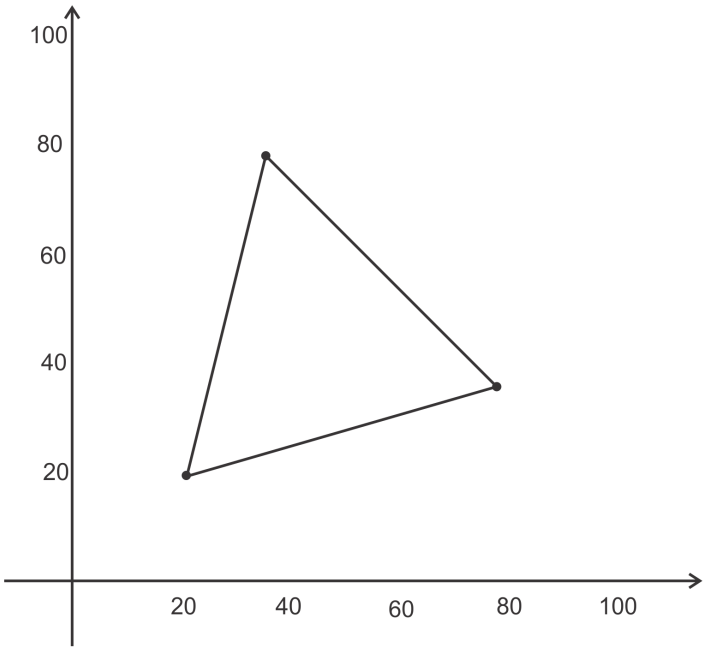

Figura 4.4: Exemplo de um Simplex Inicial em $\Re^2$.

Para a maioria dos problemas práticos, é recomendado iniciar o método com um simplex "grande" de modo a evitar que as iterações acabem prematuramente ou tenham uma melhora pouco expressiva, entretanto, no caso de um problema com múltiplos mínimos, como é o caso deste problema de *min-max-min*, a sugestão é que o Simplex Inicial seja "pequeno", e partindo de diferentes pontos, escolher o melhor resultado obtido com estes experimentos, sem entretando ter a garantia de ter encontrado um mínimo global [41].

4.3 A Curva de Lissajous

Como uma das características de um problema de recobrimento é a multiplicidade de mínimos locais, e como o Método de Nelder-Mead não garante a convergência para um ponto de mínimo, para ampliar as possibilidades de encontrar um valor mínimo, ou um ponto próximo ao mínimo [42], o método é reiniciado várias vezes, partindo portanto de novos Simplexes Iniciais.

Os novos Simplexes iniciais são construídos alterando apenas a posição do centro da elipse construída para formar o primeiro conjunto de centros C_1.
Como é a partir de C_1 que são gerados os outros conjuntos de centros $C_i's$, $i = 2, \ldots, 2n+1$, tem-se portanto novos Simplexes Iniciais.

Para tanto, uma outra proposta desta tese é construção das elipses de tal maneira que seus centros são os pontos da trajetória de uma curva pertencente à família de *Curvas de Lissajous*, estudada por Nathaniel Bowditch em 1815, e mais tarde por Jules Antoine Lissajous, em 1857.

A trajetória de uma *Curva de Lissajous* em $\Re^2$ é produzida pelas equações paramétricas:

$$x_i = a\,\sin(\omega_1\,t) + c_x$$
$$y_i = b\,\sin(\omega_2\,t + \varphi) + c_y$$

onde:

- a e b são as amplitudes do movimento na direção dos eixos,
- ω_1 e ω_2 são as frequências das oscilações, adequadamente escolhidos e
- c_x e c_y são as coordenadas do ponto inicial da tragetória da curva.

As *Curva de Lissajous* além de suaves, são densas no retângulo limitado por x = ± a e y = ± b [16], o que garante que a trajetória formada pelos pontos desta curva recobrirão este retângulo[1], gerando portanto um novo Simplex Inicial e então método é recomeçado.

A elipse C_1 é então construída da seguinte maneira:

$$x = x_i + \bar{a}\,\cos\alpha$$
$$y = y_i + \bar{b}\,\sin\alpha$$

[1]Entretanto pontos da curva de Lissajous gerados com parâmetros diferentes (valores de t) podem ser iguais

onde:
x_i e y_i são os centros da elipse dados pelos pontos da trajetória da *Curva de Lissajous* com:
a = TamanhoX/K_L;
b = TamanhoY/K_L;
$\omega_1 = 5.0$
$\omega_2 = 6.0$
$\varphi = 0.0$
(c_x , c_y) a origem dos eixos X* e Y*;
$\bar{a}$ = TamanhoX/K_e;
$\bar{b}$ = TamanhoY/K_e;
$\alpha = 2\pi/n$
K_L e K_e escolhidos apropriadamente através de experimentos.

A Figura 4.5 mostra a evolução da curva, com o parâmetro t iniciando em 0 com incremento 0.25, i.e., $\delta t = 0, 0.25, 0.5 \ldots$

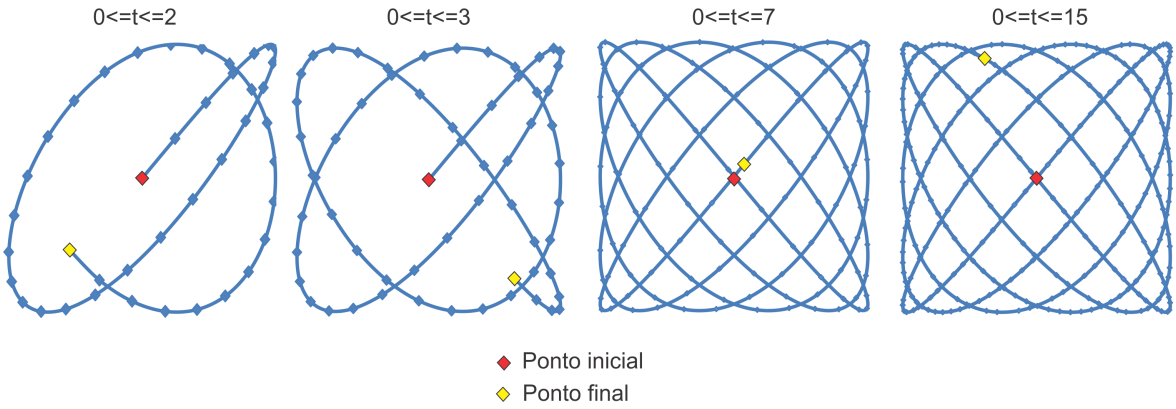

Figura 4.5: Curva de Lissajous com $\omega_1 = 5.0$ e $\omega_2 = 6.0$.

Uma outra vantagem do uso da Curva de Lissajous é a possibilidade de usar o paralelismo que será descrito na seção a seguir.

4.3.1 Paralelismo

Em sua tese de doutorado, Virgia Torczon [37] desenvolveu um algorítmo de busca multidirecional, comparou com os métodos de Nelder-Mead e de Spendley, Hext and Himsworth e explorou o paralelismo em seu método de busca multidirecional, afirmando que os algoritmos de busca multidirecional são inerentemente associados ao paralelismo. O algoritmo faz para quatro *loops* que podem ser executados em paralelo: o loop de inicializar a *procedure* e três *loops* para cada uma das etapas de expansão, reflexão e contração.

No trabalho aqui apresentado, o primeiro conjunto de centros C_1 foi construído formando uma elipse cujo centro é um ponto de uma curva de Lissajous (Figura 4.6).

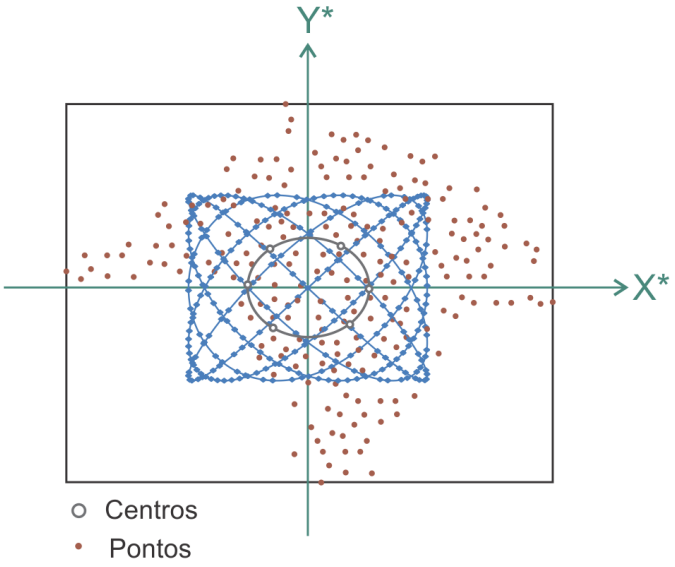

Figura 4.6: Retângulo com C_1(elipse) e a Curva de Lissajous.

Como todos os outros n conjuntos de centros $C_i's, \quad i = 2 \cdots 2n+1$ que formam o Simplex Inicial são construídos a partir de C_1, o paralelismo pode ser usado para inicializar o algorítmo com diferentes e consecutivos intervalos de t (Figura 4.7).

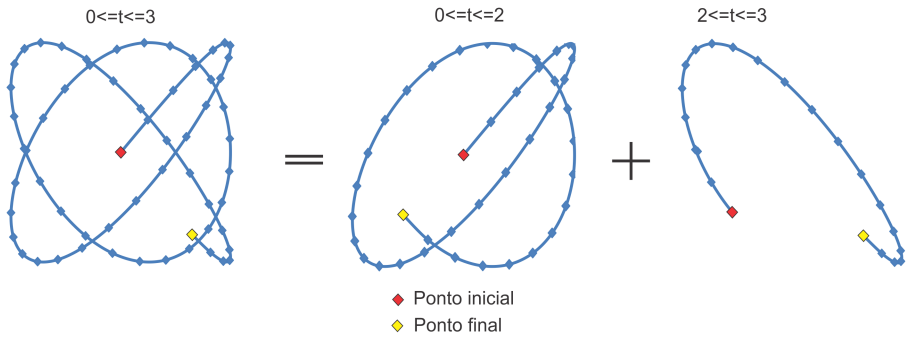

Figura 4.7: Diferentes pontos de partida do algorítmo.

Obs: 4.2. *Com o objetivo de encontrar a melhor solução possível, foram feitos experimentos com várias combinações dos tamanhos dos retângulos onde as Curvas de Lissajous são geradas e os tamanhos dos Simplexes Iniciais, como mostra a Figura 5.5.*

Obs: 4.3. *Os parâmetros da curva de Lissajous foram escolhidos verificando uma distribuição mais homogênea possível dentro do retângulo de pontos para evitar uma concentração de pontos em uma determinada parte deste retângulo no início da trajetória.*

4.4 Aplicação do Método de Nelder-Mead

Com a função objetivo f: $\Re^{2n} \rightarrow \Re$ adaptada para o método e os 2n+1 conjuntos de n antenas $C_i's, \quad i = 1 \cdots 2n+1$ com 2n coordenadas construídos, o Método de Nelder-Mead foi aplicado com os coeficientes usados pelos autores e que são mais usados na literatura:

Tabela 4.2: Coeficientes do Método de Nelder-Mead.

Coeficientes	Valores
ρ	1
χ	2
γ	1/2
σ	1/2

Critério de Parada
Para cada conjunto de centros

$$C_i = (C_{i1}, C_{i2}, C_{i3}, C_{i4}, \cdots C_{i2n-1}, C_{i2n}) \quad i = 1, \ldots, 2n+1$$

calcula-se para cada k = 1, ..., 2n:

$$C_{ik} - C_{jk} \qquad i, j = 1, \ldots, 2n+1$$

Seja m = max $||C_{ik}$ - $C_{jk}||_2$ onde || x $||_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$

Como a distância percorrida pelo simplex está relacionada como tamanho do simplex, o Critério de Parada adotado foi:
Dada uma tolerância $\epsilon > 0$ e um número máximo de iterações N, parar as iterações quando:

$$\text{m} < \epsilon \qquad \text{ou número de iterações} > \text{N}$$

Obs: 4.4. *Com o Critério de Parada descrito acima, o método termina quando o diâmetro do simplex se tornar menor que a tolerância ϵ escolhida.*

4.5 Refinamento da Solução

Obtida a melhor solução do problema de *min-max-min*, foi feito um refinamento desta solução, colocando-a como um dos vértices do Simplex Inicial, mais precisamente C_1, e com um tamanho do passo da base $s_i \quad i = 2 \cdots 2n + 1$ "pequeno", foi gerado um simplex reduzido numa vizinhaça da melhor solução.
Com este novo Simplex Inicial Reduzido foi feita uma nova rodada do Método de Nelder-Mead.
O algoritmo abaixo mostra este procedimento.

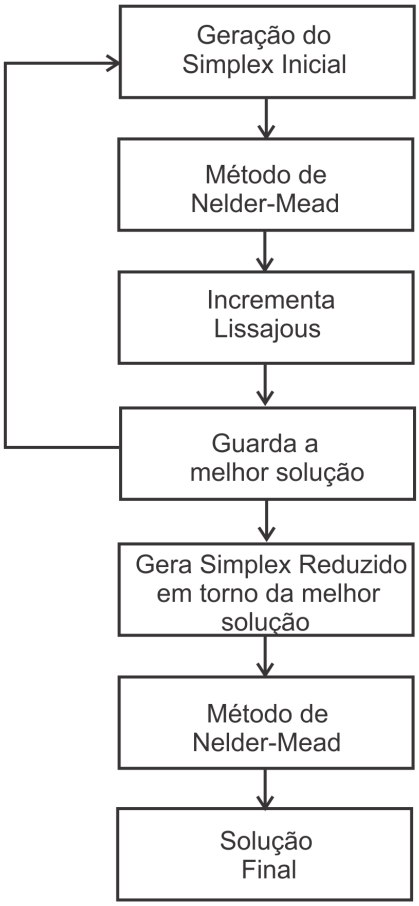

Figura 4.8: Refinamento da melhor solução.

Obs: 4.5. *Para o Refinamento da solução foi utilizado o mesmo programa sem percorrer os pontos da curva de Lissajous e em C_1 foi posta a melhor solução encontrada no Programa Principal.*

Obs: 4.6. *A figura 4.9 mostra um exemplo da tela do Programa Principal.*

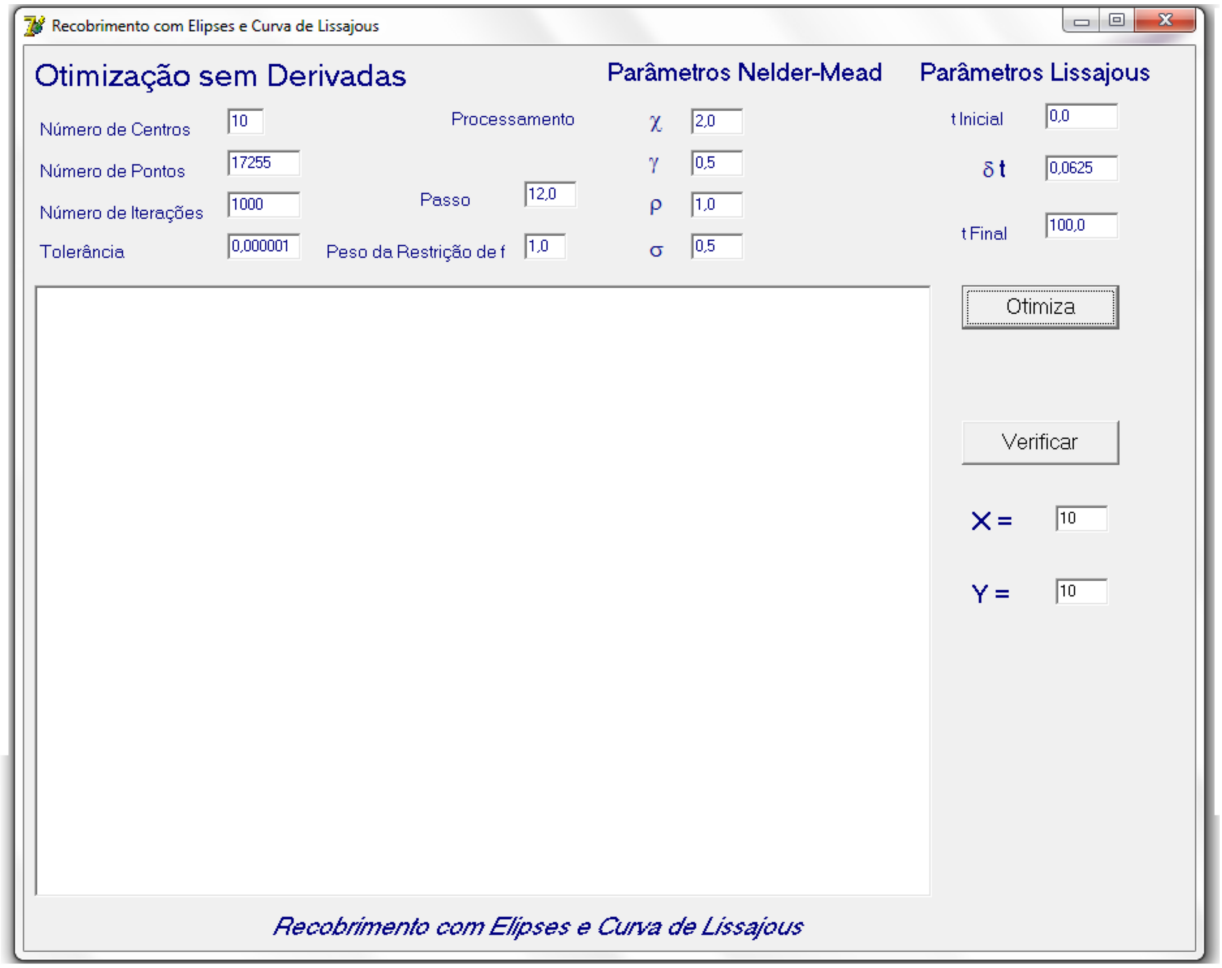

Figura 4.9: Tela do Programa.

Capítulo 5

Experimentos Numéricos

5.1 Introdução

Os experimentos numéricos foram feitos em 3 diferentes categorias: Exemplos de Validação, Exemplo Comparativo, e Exemplos Produzidos.

Estes experimentos foram eleborados em liguagem de programação Delphi 7.0 e executados em um notebook Dell XPS com processador Intel(R)Core(TM)i5 com 4,00 GB de memória.

Foram feitos testes numéricos com a declaração das variáveis em formas diferentes de precisão (*single*, *double* e *extended*), e verificou-se que não há um padrão de precisão a ser seguido.

Os três **Exemplos de Validação**, onde as soluções são conhecidas previamente, foram construídos com poucos pontos com o intuito de testar o Método de Nelder-Mead para o problema de recobrimento.

Como **Exemplos Comparativos** foram feitos os recobrimentos dos Estados do Rio de Janeiro e de Nova York.
Para estes exemplos, existem resultados computacionais disponíveis em [7], onde os autores resolvem o Problema de Recobrimento usando a técnica da Suavização Hiperbólica.

Os **Exemplos Produzidos** foram implementados para o recobrimento do Estado do Pará e do Brasil descritos nas seções 5.4.2 e 5.4.3 respectivamente.

Para os **Exemplos Produzidos** foram implementados 2 rotinas computacionais descritos na seção 5.4:

- o programa ***Gerar Dados*** para gerar os pontos a serem recobertos, e
- o programa ***Contar Números*** que exibe o número de pontos gerados em *Gerar Dados*.

5.2 Exemplos de Validação

5.2.1 2 Centros e 08 Pontos

Como primeiro exemplo de validação foram gerados 8 pontos sobre 2 círculos de raio unitário e centros em C_1 e C_2[1].

Tabela 5.1: Tabela com 8 pontos a serem recobertos.

i	x_i	y_i
1	-1.414214	0.0707107
2	-1.414214	-0.0707107
3	0.0	0.0707107
4	0.0	-0.0707107
5	1.414214	0.0707107
6	1.414214	-0.0707107
7	2.828427	0.0707107
8	2.828427	-0.0707107

A figura abaixo mostra solução na primeira e na última iteração do método respectivamente.

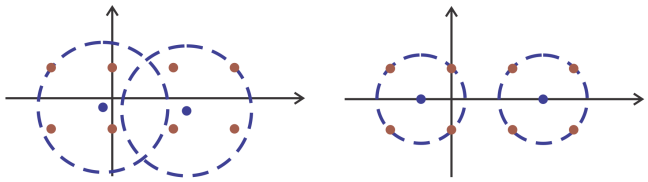

Figura 5.1: Iteração Inicial e Final com 8 pontos e 2 centros.
Raio Inicial: 1.5 Raio Final: 1.0000008345

Localização dos Centros e tamanho do raio previamente conhecidos:

$$C_1 = (-0.707107, 0.0), \quad C_2 = (2.12132, 0.0), \quad r = 1.0$$

Tempo de execução: 0.1 segundo
Número de Iterações: 161

[1]Qualquer outro tipo de exemplo poderia ser testado, desde que a solução seja previamente conhecida. Por exemplo, pontos em uma linha reta

5.2.2 3 Centros e 12 Pontos

Para este exemplo foram gerados 12 pontos sobre 3 círculos de raio unitário e centros em C_1, C_2 e C_3.

Tabela 5.2: Tabela com 12 pontos a serem recobertos.

i	x_i	y_i
1	-1.414214	0.0707107
2	-1.414214	-0.0707107
3	0.0	0.0707107
4	0.0	-0.0707107
5	1.414214	0.0707107
6	1.414214	-0.0707107
7	2.828427	0.0707107
8	2.828427	-0.0707107
9	1.414214	-2,121321
10	1.414214	-3.535535
11	2.828427	-2,121321
12	2.828427	-3.535535

A figura 5.2 mostra a solução na primeira e na última iteração do método respectivamente.

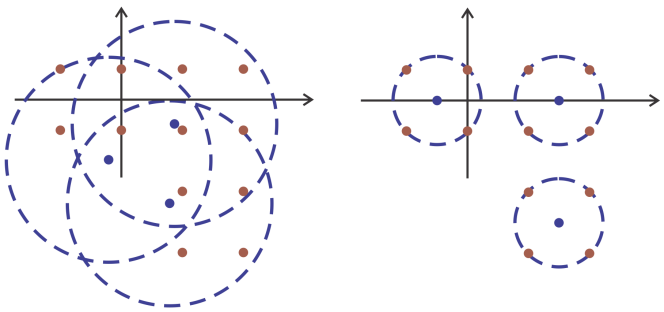

Figura 5.2: Iteração Inicial e Final com 12 pontos e 3 centros.
Raio Inicial: 2.37 Raio Final: 1.002.

Localização dos Centros e tamanho do raio previamente conhecidos:

$C_1 = (-0.707107, 0.0), \quad C_2 = (2.12132, 0.0), \quad C_3 = (2.12132, -2.828428), \quad r = 1.0$

Tempo de execução: 0.1 segundo
Número de Iterações: 249

5.2.3 3 Centros e 20 Pontos

Como terceiro exemplo de validação foram gerados 12 pontos sobre 3 círculos de raio unitário e centros em C_1, C_2 e C_3.

Tabela 5.3: Tabela com 20 pontos a serem recobertos.

i	x_i	y_i	i	x_i	y_i
1	0.366	0.0	11	-0.1589	0.1589
2	1.366	0.0	12	-1.732	1.366
3	0.0	0.1589	13	-1.732	0.366
4	0.0	0.366	14	-0.866	1.866
5	0.0	1.366	15	0.1589	-0.1589
6	1.732	1.366	16	0.0	-0.366
7	1.732	0.366	17	0.0	-1.366
8	0.866	1.866	18	1.732	-1.366
9	-0.366	0.0	19	1.732	-0.366
10	-1.366	0.0	20	0.866	-1.866

A figura 5.3 mostra a solução na primeira e na última iteração do método respectivamente.

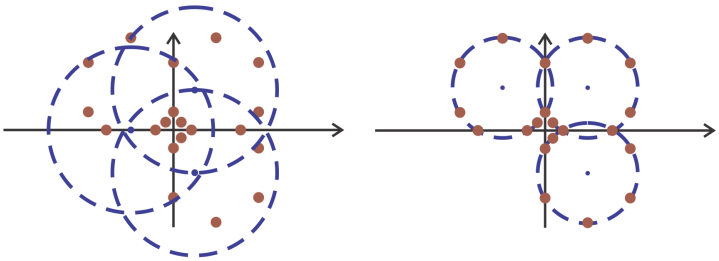

Figura 5.3: Iteração Inicial e Final com 20 pontos e 3 centros.
Raio Inicial: 1.675 Raio Final: 1.017.

Localização dos Centros e tamanho do raio previamente conhecidos:

$$C_1 = (-0.86602, 0.86602), \quad C_2 = (0.86602, 0.86602), \quad C_3 = (0.86602, -0.86602), \quad r = 1.0$$

Tempo de execução: 0.1 segundo
Número de Iterações: 214

Conclui-se então que o algorítmo implementado obteve sucesso nos **Exemplos de Validação**, garantindo a solução correta nos 3 testes.

5.3 Exemplos Comparativos

Os Exemplos Comparativos tiveram como fonte exemplos que constam na literatura [7], para o recobrimento da cidade do Rio de Janeiro com 3539 pontos para o recobrimento com 9 centros e para o estado de Nova York com 7225 pontos e 7 centros.

O que será comparado entre as soluções nestes exemplos é apenas o raio de cobertura; não é possível comparar o tempo de execução, nem o número de avaliações da função ou qualquer outra medida de desempenho dado a discrepância entre as estruturas do algoritmo proposto e os implementados na literatura (complexidade e abordagens diferentes).

5.3.1 Recobrimento do Estado do Rio de Janeiro

A solução do problema obtida em [7] é:
Maior Raio = 8.0036889924429
Valor de f = 64.059036008
Coordenadas dos 9 Centros:

1) (47.497579365155 , -3.4522761367104)
2) (33.836780017461 , 4.0692695860557)
3) (10.684442530386 , -12.295564330915)
4) (10.610498976764 , 0.82315041654431)
5) (-1.6799609733389 , -2.7606177703228)
6) (20.468205274639 , 3.7367964731822)
7) (22.958960768869 , -8.0385074361554)
8) (47.268736899303 , 8.8653395321917)
9) (35.998844458840 , -7.0867096559278)

A Figura 5.4 mostra a localização destes centros.

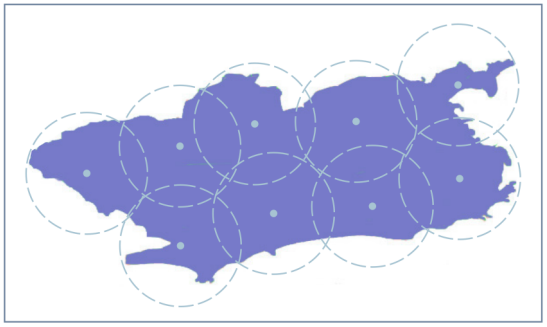

Figura 5.4: Recobrimento do Estado do Rio de Janeiro obtido na literatura.

Com os dados deste problema, foram feitos experimentos com tamanhos de eixos das elipses onde estão os primeiros conjuntos de centros C_1 (RE) e de retângulos

das Curvas de Lissajous (RL) mostrados na tabela s seguir: onde $TamanhoX$ e

Tabela 5.4: Tamanhos de eixos das elipses e retângulos de Lissajous

a)	RE = $TamanhoX$ x $TamanhoY$	e	RL = $TamanhoX$ x $TamanhoY$;
b)	RE = $TamanhoX/2$ x $TamanhoY/2$	e	RL = $TamanhoX$ x $TamanhoY$;
c)	RE = $TamanhoX/4$ x $TamanhoY/4$	e	RL = $TamanhoX$ x $TamanhoY$;
d)	RE = $TamanhoX$ x $TamanhoY$	e	RL = $TamanhoX/2$ x $TamanhoY/2$;
e)	RE = $TamanhoX/2$ x $TamanhoY/2$	e	RL = $TamanhoX/2$ x $TamanhoY/2$;
f)	RE = $TamanhoX/4$ x $TamanhoY/4$	e	RL = $TamanhoX/2$ x $TamanhoY/2$;
g)	RE = $TamanhoX$ x $TamanhoY$	e	RL = $TamanhoX/4$ x $TamanhoY/4$;
h)	RE = $TamanhoX/2$ x $TamanhoY/2$	e	RL = $TamanhoX/4$ x $TamanhoY/4$;
i)	RE = $TamanhoX/4$ x $TamanhoY/4$	e	RL = $TamanhoX/4$ x $TamanhoY/4$;
j)	RE = $TamanhoX/2$ x $TamanhoY/2$	e	RL = $TamanhoX/5$ x $TamanhoY/5$.

$TamanhoY$ são os lados do Retângulo Mínimo de Pontos.

Neste exemplo:
$TamanhoX = 64.4722976685$ e
$TamanhoY = 29.9612007141$.

Para cada uma das 10 combinações de tamanhos descritos de a) a j), foram feitos experimentos com o tamanho do passo s variando entre os inteiros de 1 a 12.

Foram feitas 500 iterações para cada ponto (reinício do método) da Curva de Lissajous com os parâmetos: t = 0, ...,500, incremento $\delta t = 0.0625$, $\omega_1 = 5.7$, $\omega_2 = 5.9$.

Com o melhor resultado obtido em cada ítem de a) a j), foram feitas 5000 iterações, e com o melhor destes resultados foram feitas 10000 iterações, para posterior refinamento, com o objetivo de melhorar a solução.

A tolerância usada foi 0.000001 e o parâmetro de penalização $M = 1.0$ (descrito em 4.2.3).

Obs: 5.1. *No Apêndice C -* ***Recobrimento do Estado do Rio de Janeiro****, em C.1 -* ***Experimentos****, estão todos os resultados de cada experimento descrito nesta seção.*

A Figura 5.5 mostra o Estado do Rio de Janeiro inserido no Retângulo Mínimo e as combinações dos tamanhos dos eixos das elipses do conjunto de centros C_1 e dos retângulos das Curvas de Lissajous descritos nos ítens de a) a j).

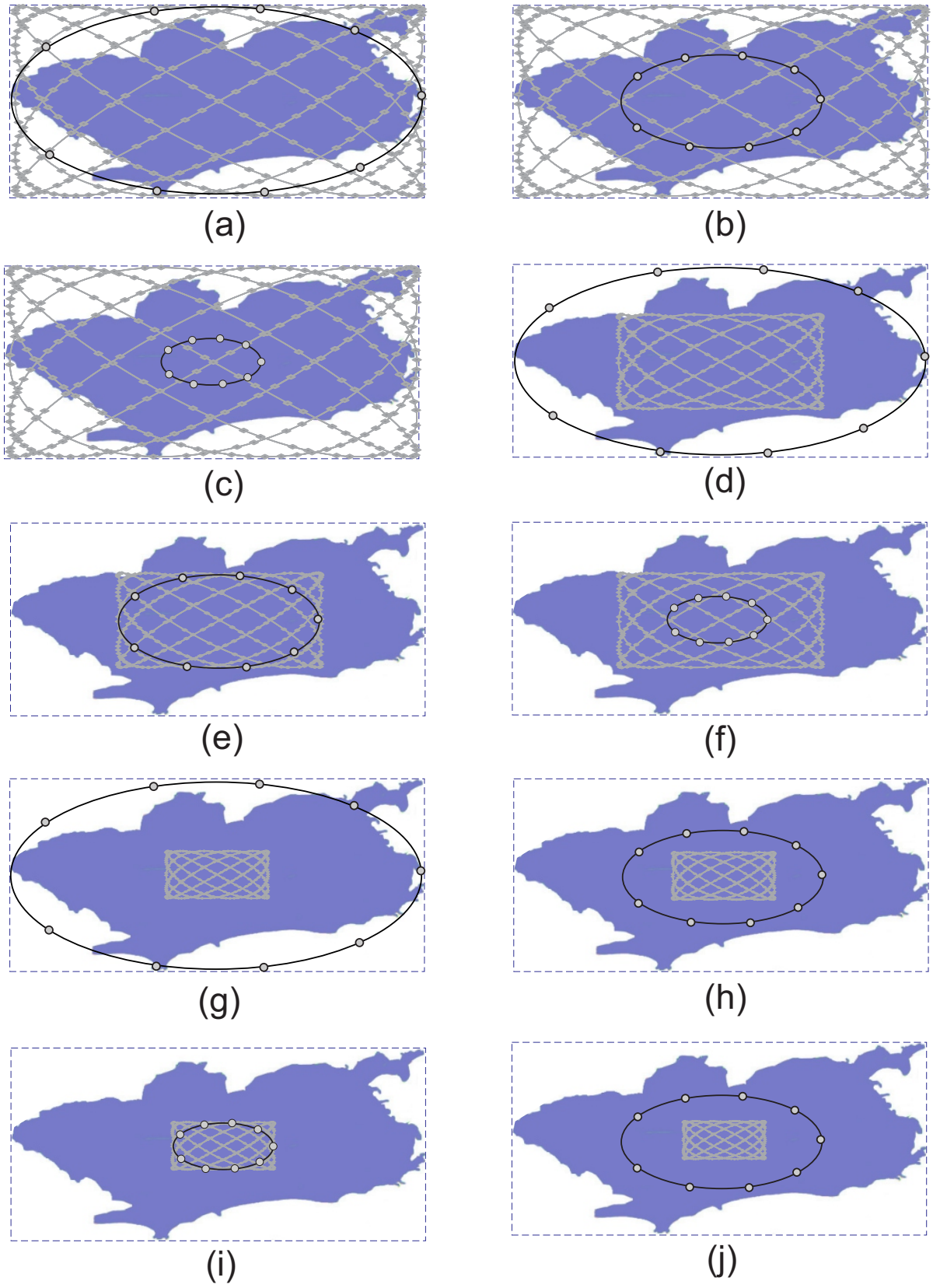

Figura 5.5: Rio de Janeiro: C_1, Retângulo Mínimo e Curva de Lissajous.

O melhor resultado obtido com os experimentos para o recobrimento para o Estado do Rio de Janeiro foi com os seguintes parâmetros:

- Eixos da elipse de C_1 RE = $TamanhoX/2$ x $TamanhoY/2$
- Retângulo de Lissajous RL = $TamanhoX/5$ x $TamanhoY/5$
  (item j da Figura 5.5)

A solução do problema obtida foi:

- Maior Raio = 8.044712066650
- Valor de f = 64.882118225098
- Coordenadas dos 9 Centros:
  1) (47.363449096680 , -2.659626960754) Raio = 8.0304641724
  2) (33.158096313477 , 4.152007579803) Raio = 8.0447120667
  3) (10.514169692993 , -12.136947631836) Raio = 8.0371904373
  4) (11.195415496826 , 0.293844282627) Raio = 7.9883708954
  5) (-1.447932243347 , -2.555695295334) Raio = 8.0447120667
  6) (19.684661865234 , 3.415480136871) Raio = 7.9065179825
  7) (23.791072845459 , -8.958263397217) Raio = 8.0447101593
  8) (46.962646484375 , 9.536380767822) Raio = 8.0224924088
  9) (36.054138183594 , -7.167820453644) Raio = 8.0186901093
- Tolerância Atingida = 0.00000137
- Tamanho do Passo da Base = 12.000
- Parâmetro de Penalização M = 1.0
- t final de Lissajous = 201.12500
- δt de Lissajous = 0.06250
- Tempo de Execução para Passo da Base 12 = 54 min.

As Figuras 5.6 e 5.7 mostram a localização dos melhores centros e seus raios de cobertura na primeira e na última iteração do método.

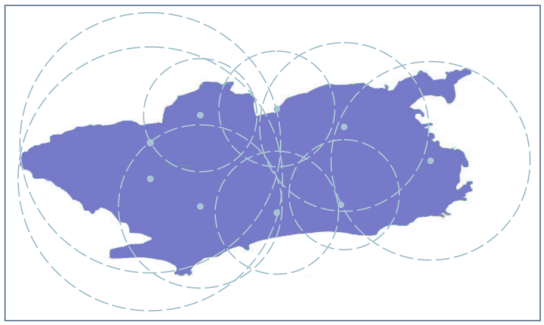

Figura 5.6: Primeira iteração do método.

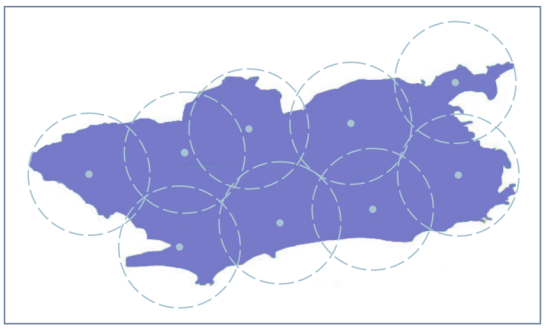

Figura 5.7: Última iteração do método.

Usando a solução obtida como um vértice (C_1) do Simplex Inicial, foi feito o Refinamento da Solução (seção 4.5) com tamanho do passo da base igual a 1.0 e a solução encontrada foi:

- Maior Raio = 8.0178756246
- Valor de f = 64.2863295322
- Coordenadas dos 9 Centros e seus Raios de Cobertura:
  1) (47.3677093826 , -2.6754264855) Raio = 8.0178756246
  2) (33.1799494839 , 4.1140872006) Raio = 8.0169073264
  3) (10.6308535395 , -12.2698326464) Raio = 8.0041859057
  4) (11.2476093821 , 0.2691270374) Raio = 7.9955694452
  5) (-1.4829534890 , -2.4090995876) Raio = 7.9957083999
  6) (19.8333370142 , 3.5708017852) Raio = 7.8700696303
  7) (23.7941659964 , -8.8581812425) Raio = 7.9977894089
  8) (46.9037002290 , 9.4890058624) Raio = 7.9479026376
  9) (36.2745673135 , -7.1415630512) Raio = 7.8224285579
- Tolerância Atingida = 0.00000001

- Tamanho do Passo da Base = 1.0
- Parâmetro de Penalização M = 0.0

A Figura 5.8 mostra a localização dos centros encontrados na solução refinada, e a Figura 5.9 mostra as soluções implementadas neste trabalho antes do Refinamento a), após o Refinamento b) e a solução encontrada em [7] c).

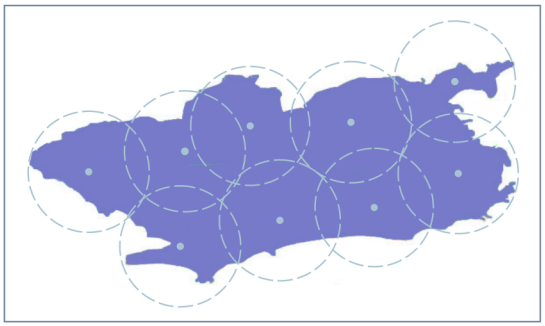

Figura 5.8: Recobrimento Final Refinado do Estado do Rio de Janeiro.

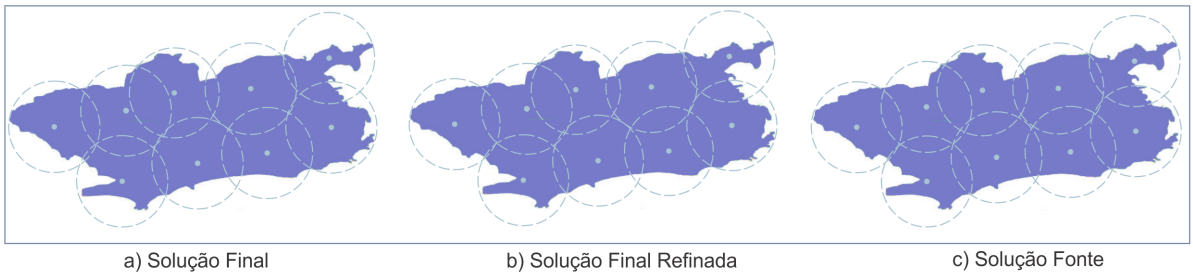

Figura 5.9: Soluções: Final - Final Refinada - Fonte.

A tabela 5.5 mostra os valores dos raios de cobertura da solução obtida, da solução encontrada na literatura e da solução da literatura refinada, isto é, posta como um dos vértices do simplex inicial e feita uma rodada do Método de Nelder-Mead.

Tabela 5.5: Comparação das soluções.

Soluções	Raios
Solução Refinada	8.0178756246402
Solução Fonte	8.0036889924429
Solução Fonte Refinada	8.0035073221432

Obs: 5.2. *No Apêndice C -* ***Recobrimento do Estado do Rio de Janeiro****, em C.2 -* ***Raios de Cobertura****, a figura C.2 mostra os melhores raios encontrados*

nos experimentos de a) a j) associados aos tamanhos dos retângulos de Lissajous e elipses onde estão os conjuntos de centros C_1.

Obs: 5.3. *Os parâmetros usados para todos os experimentos a seguir foram:*

δt de Lissajous	*= 0.0625;*
t de Lissajous t	*= 0, $\cdots$,500;*
Parâmetro de Penalização M	*= 1.0;*
Parâmetro de Penalização no Refinamento M	*= 0.0;*
Tamanho do Passo	*= 1, $\cdots$,12;*
Tamanho do Passo no Refinamento	*= 0;*
Número de iterações para cada t	*= 500;*
Número de iterações no Refinamento	*= 5000;*
Critério de parada ϵ	*= 0.0000001;*
Frequências da Curva de Lissajous ω_1 e ω_2	*= 5.89 e 5.0 respectivamente.*

5.3.2 Recobrimento do Estado de Nova York

Com os 7225 pontos para um recobrimento do Estado de Nova York com 7 centros obtidos na literatura, onde o maior raio encontrado foi 23.0212229401, o Retângulo Mínimo de Pontos mede:
$TamanhoX = 150.0000000000$ e $TamanhoY = 109.0000000000$.

Os experimentos foram feitos com os tamanhos de eixos das elipses onde estão C_1 e retângulos das curvas de Lissajous mostrados na tabela 5.4.

A figura 5.10 mostra os Retângulos Mínimos e as combinações dos tamanhos dos eixos das elipses do conjunto de centros C_1 e dos retângulos das Curvas de Lissajous.

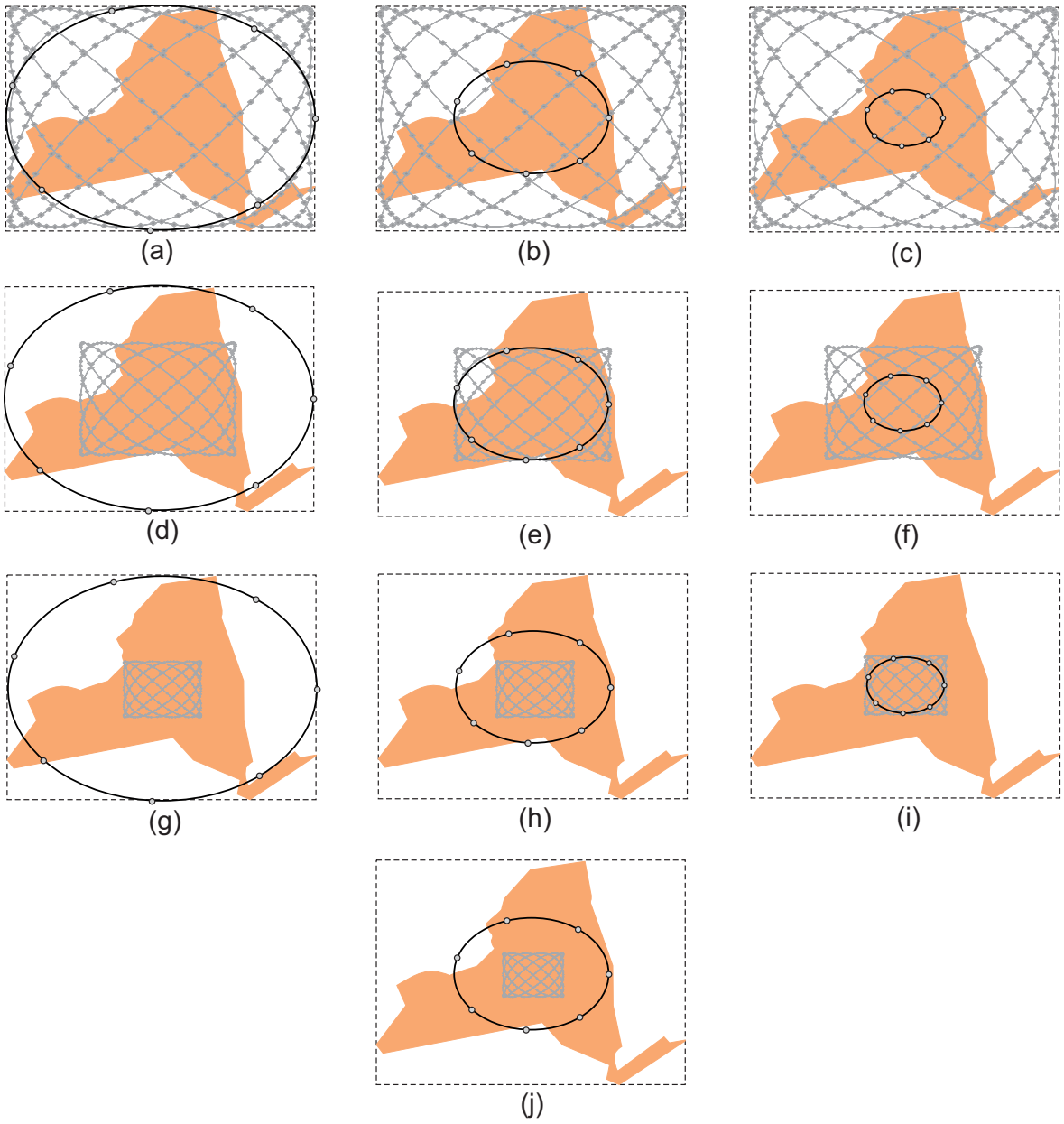

Figura 5.10: NY: C_1, Retângulo Mínimo e Curva de Lissajous.

A solução do recobrimento de NY foi a seguinte:

- Maior Raio = 23.072999954224
- Valor de f = 536.8745117188
- Coordenadas dos 7 Centros:
  1) (618.455078125000 , 221.091857910156) Raio = 23.0729999542
  2) (599.330017089844 , 260.658020019531) Raio = 22.8804340363
  3) (577.603759765625 , 296.360870361328) Raio = 22.9719753265
  4) (558.886718750000 , 272.916961669922) Raio = 23.0727386475
  5) (509.287597656250 , 236.982513427734) Raio = 23.0702533722
  6) (539.456298828125 , 245.876495361328) Raio = 23.0445575714
  7) (580.103515625000 , 234.993377685547) Raio = 23.0729446411
- Tolerância Atingida = 0.00014950
- Tamanho do Passo da Base = 10.00
- t final de Lissajous = 197.87500
- Tempo de Execução para Passo da Base 10 = 80 min.

Com:
Eixos da elipse de C_1 RE = $TamanhoX/4$ x $TamanhoY/4$
Retângulo de Lissajous RL = $TamanhoX/4$ x $TamanhoY/4$
(item i da Figura 5.10)

As figuras 5.11 e 5.12 mostram a solução na primeira iteração do método e a solução final ao lado da solução obtida na literatura reespectivamente.

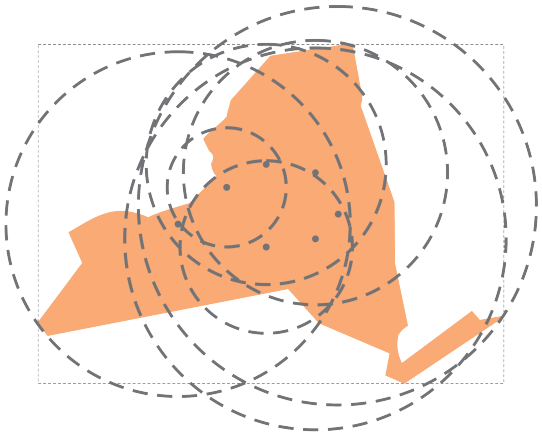

Figura 5.11: NY: Primeira iteração do método.

O Refinamento da solução foi feito e os seguintes resultados foram obtidos:

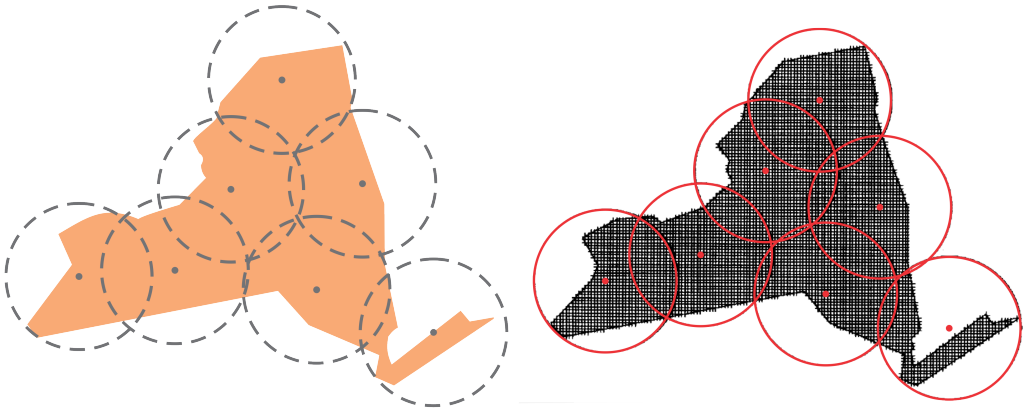

Figura 5.12: NY: Solução por Nelder-Mead e solução obtida na literatura.

- Maior Raio = 23.063687813474
- Valor de f = 531.933695557408
- Coordenadas dos 7 Centros:

1)	(618.3339156404 ,	221.7359273741)	Raio =	23.0636878135
2)	(599.2632718086 ,	260.9395899729)	Raio =	22.5934923994
3)	(577.6140463632 ,	296.4117774546)	Raio =	22.9832973626
4)	(559.2828407376 ,	273.1021668532)	Raio =	22.9736728281
5)	(509.2111913770 ,	237.1294069596)	Raio =	23.0496774901
6)	(539.7722702678 ,	246.1906546255)	Raio =	23.0076385414
7)	(580.2327927393 ,	234.7980617016)	Raio =	22.9683226951

- Tolerância Atingida = 0.00000001

A figura 5.13 mostra a solução refinada e a solução obtida na literatura.

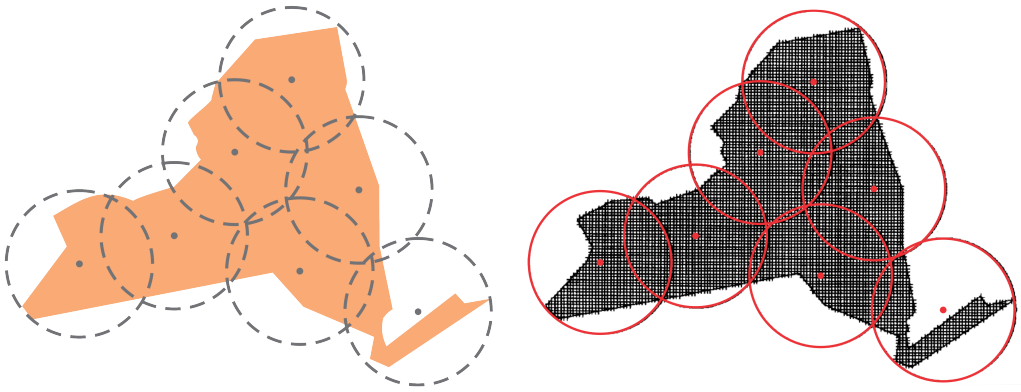

Figura 5.13: NY: Solução refinada por Nelder-Mead e solução obtida na literatura.

Obs: 5.4. *Com o refinamento o maior raio diminuiu, entretanto uma antena ficou fora da região S pelo fato de no refinamento o parâmetro de penalização M ser igual a zero. Ficamos então com a solução não refinada que é uma solução viável.*

Comparando a solução obtida para o recobrimento do estado de Nova York com o resultado da literatura nota-se claramente a importância da penalização da função proposta neste trabalho, que coloca os centros dentro da fronteira da região S.

Obs: 5.5. *No Apêndice D -* ***Recobrimento do Estado de Nova York****, em D.1* ***Experimentos*** *estão todos os resultados de cada experimento feitos nesta seção e em D.2 -* ***Raios de Cobertura****, a figura mostra os melhores raios encontrados nos experimentos de a) a j) associados aos tamanhos dos retângulos de Lissajous e elipses onde estão os conjuntos de centros* C_1.

5.4 Exemplos Produzidos

Para os Exemplos Produzidos foram construídos dois exemplos para o recobrimento do Estado do Pará e do Brasil.

A geração de pontos para o recobrimento de cada um destes exemplos foi feita com o auxílio do programa *Gerar Dados*.

Para usar o *Gerar Dados*, a região a ser recoberta é dividida em retângulos e triângulos, como mostrado nos mapas do Estado do Pará na figura 5.14 e do Brasil na figura 5.15.

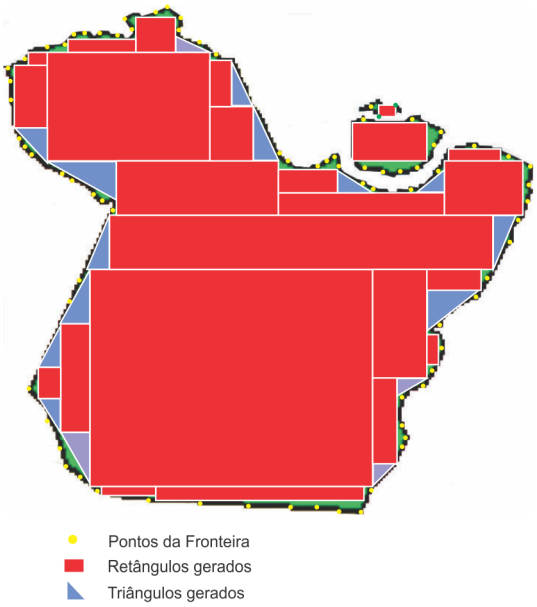

Figura 5.14: Retângulos, Triângulos e Pontos da Fronteira no Estado do Pará.

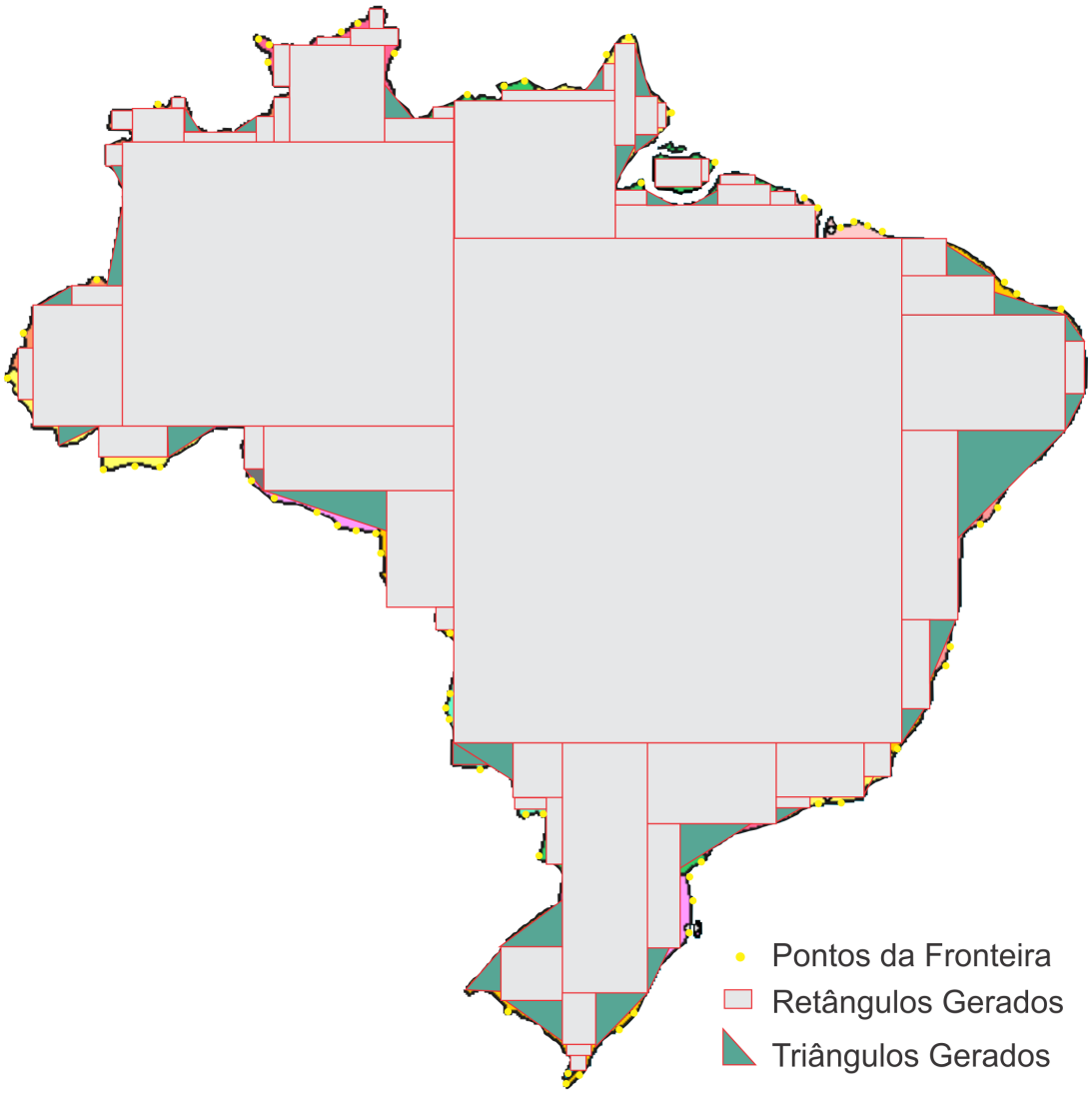

Figura 5.15: Exemplo de Retângulos e Triângulos no mapa do Brasil.

Com coordenadas iniciais e finais apropriadas de cada retângulo ou triângulo o programa preenche com pontos cada uma destas figuras geométricas, como mostra a subseção 5.4.1.

5.4.1 Geração de Pontos

No programa *Gerar Dados* as entradas são:

- as coordenadas iniciais (x_i, y_i);
- as coordenadas finais (x_f, y_f);
- um número inteiro entre 1 e 5 que especifica se o usuário quer preencher um retângulo de pontos, ou um triângulo específico de pontos, conforme mostra a figura 5.16;

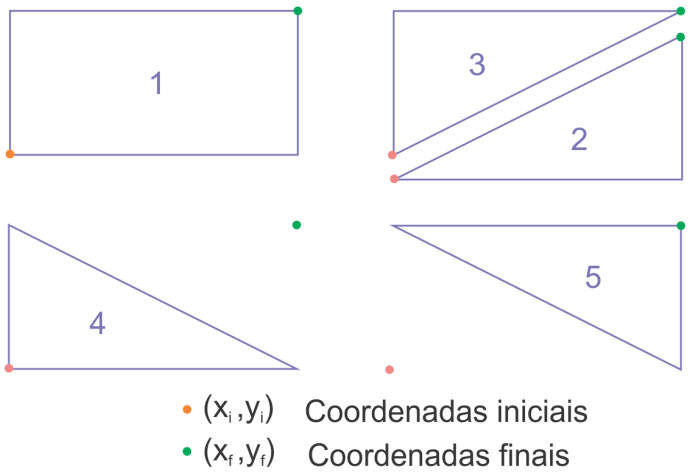

Figura 5.16: Geração de Pontos.

Com estas entradas o programa preenche a figura geométrica escolhida com pontos a serem recobertos com um espaçamento pré-determinadado entre estes pontos.

Com os pontos gerados pelo programa *Gerar Dados*, mais os pontos colocados na fronteira, a região a ser recoberta fica preenchida. Estes pontos serão a entrada dos dados do Programa Principal.

Para computar a totalidade dos pontos a serem recobertos em cada um dos Exemplos Produzidos, os pontos obtidos no programa *Gerar Dados*, juntamente com os pontos colocados na fronteira, foram colocados como entrada do programa *Contar Números* cuja saída é a quantidade desses pontos.

Obs: 5.6. *No Apêndice B -* ***Programas Auxiliares****, na seção B.1* ***Programa Gerar Dados*** *está o programa Gerar Dados e na seção B.2* ***Programa Contar Números*** *está o programa Contar Números.*

5.4.2 Recobrimento do Estado do Pará

O exemplo do Estado do Pará para o recobrimento com 5 centros foi criado com 4381 pontos, sendo 71 pontos postos na fronteira e 3416 pontos gerados no programa *Gerar Dados*.

Com estes dados, foram repetidos os experimentos da tabela 5.4
Neste exemplo, o Retângulo Mínimo de Pontos mede:
$TamanhoX = 84.0999984741$ e;
$TamanhoY = 80.8000030518$.

A figura 5.17 mostra o Estado do Pará inserido no Retângulo Mínimo e as combinações dos tamanhos dos eixos das elipses do conjunto de centros C_1 e dos retângulos das Curvas de Lissajous de a) a j).

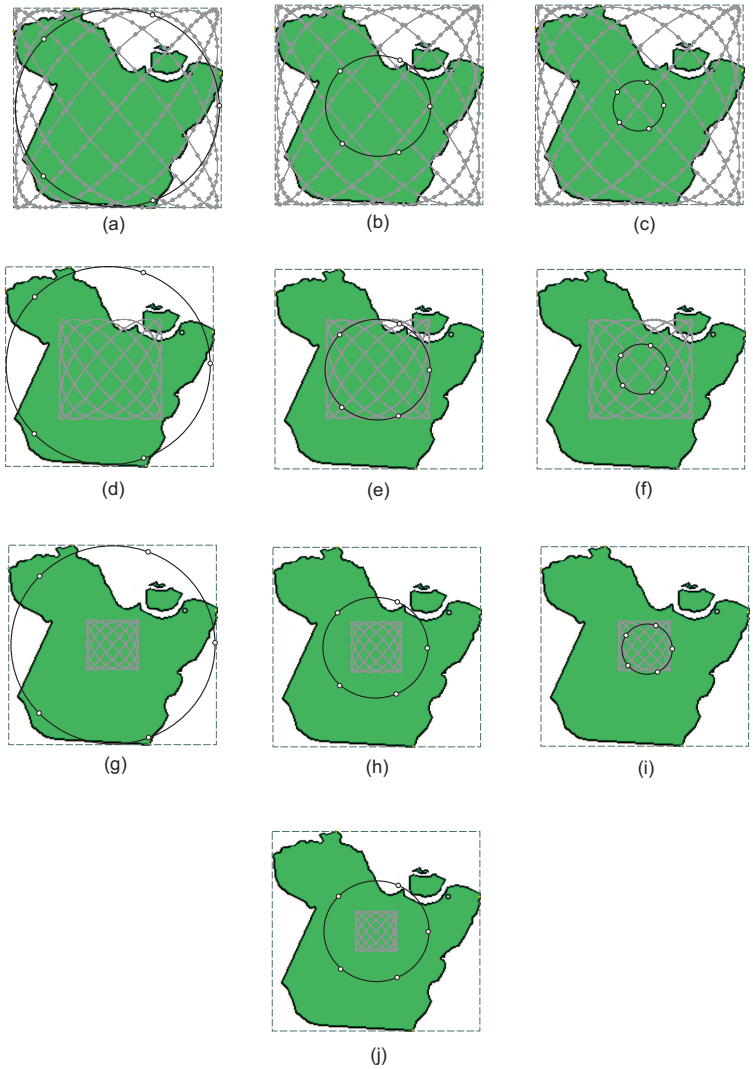

Figura 5.17: Pará: C_1, Retângulo Mínimo e Curva de Lissajous.

Foram obtidos dois melhores resultados com raios de cobertura iguais com os experimentos para o recobrimento do Estado do Pará com os seguintes parâmetros:

a) Eixos da elipse de C_1 RE = $TamanhoX/4$ x $TamanhoY/4$
Retângulo de Lissajous RL = $TamanhoX/4$ x $TamanhoY/4$
ver figura 5.17(i), e

b) Eixos da elipse de C_1 RE = $TamanhoX$ x $TamanhoY$
Retângulo de Lissajous RL = $TamanhoX/2$ x $TamanhoY/2$
ver figura 5.17(d).

Solução do problema a):

- Maior Raio = 20.465103149414
- Valor de f = 419.0966796875
- Coordenadas dos 5 Centros:
  1) (68.066146850586 , 50.857818603516) Raio = 20.4592952728
  2) (31.055591583252 , 51.864654541016) Raio = 20.4578819275
  3) (21.508142471313 , 71.478530883789) Raio = 20.3813629150
  4) (26.240303039551 , 23.999004364014) Raio = 20.4431362152
  5) (52.259914398193 , 20.631563186646) Raio = 20.4651031494
- Tolerância Atingida = 0.00000661
- Tamanho do Passo da Base = 4.0
- t final de Lissajous = 261.93750
- Tempo de Execução para Passo da Base 4.0 = 32 min.

As Figuras 5.18 e 5.19 mostram a localização dos melhores centros e seus raios de cobertura na primeira e na última iteração do método do **problema a)**.

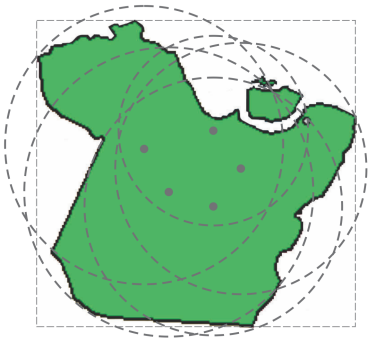

Figura 5.18: Primeira iteração do método.

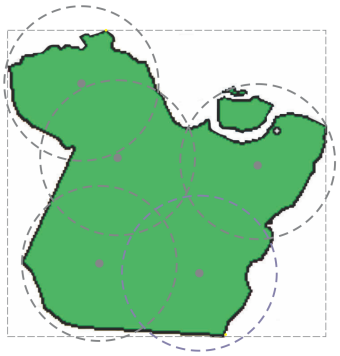

Figura 5.19: Última iteração do método.

O Refinamento da Solução foi feito e a solução encontrada foi:

- Maior Raio = 20.465103149414
- Valor de f = 418.820434570313
- Coordenadas dos 5 Centros e seus Raios de Cobertura:
  1) (68.0661468506 , 50.8578186035) Raio = 20.4592952728
  2) (31.0555915833 , 51.8646545410) Raio = 20.4578819275
  3) (21.5081424713 , 71.4785308838) Raio = 20.3813629150
  4) (26.2403030396 , 23.9990043640) Raio = 20.4431362152
  5) (52.2599143982 , 20.6315631866) Raio = 20.4651031494
- Tolerância Atingida = 0.00000934

Obs: 5.7. *Os raios encontrados na solução refinada foram os mesmos encontrados na solução final.*

Vale também observar que mesmo com pontos distintos iniciais de Lissajous, o algoritmo proposto pode convergir para a mesma solução

Solução do problema b):

- Maior Raio = 20.465103149414
- Valor de f = 419.0967102051
- Coordenadas dos 5 Centros:
  1) (68.460937500000 , 50.881790161133) Raio = 20.4352760315
  2) (31.064207077026 , 51.871883392334) Raio = 20.4538345337
  3) (21.835908889771 , 72.189224243164) Raio = 20.1800193787
  4) (24.936555862427 , 23.192687988281) Raio = 20.4566249847
  5) (52.259918212891 , 20.631557464600) Raio = 20.4651031494
- Tolerância Atingida = 0.00001009

- Tamanho do Passo da Base = 6.0
- t final de Lissajous = 352.75000
- Tempo de Execução para Passo da Base 6.0 = 68 min.

Com o Refinamento da Solução a solução encontrada foi:

- Maior Raio = 20.465103149414
- Valor de f = 418.820434570313
- Coordenadas dos 5 Centros e seus Raios de Cobertura:
  1) (68.4608764648 , 50.8809509277) Raio = 20.4344482422
  2) (31.0658302307 , 51.8696708679) Raio = 20.4513435364
  3) (21.8369789124 , 72.1911010742) Raio = 20.1807250977
  4) (24.9387569427 , 23.1928138733) Raio = 20.4554862976
  5) (52.2599143982 , 20.6315631866) Raio = 20.4651031494
- Tolerância Atingida = 0.00001206

Obs: 5.8. *O maior raio obtido com os parâmetros de b) após o refinamento foi igual ao maior raio obtido após o refinamento com os parâmetros de a). O Refinamento da Solução melhorou apenas o valor de f enquanto o maior raio permaneceu inalterado.*

Obs: 5.9. *No Apêndice E -* ***Recobrimento do Estado do Pará****, em E.1* ***Experimentos*** *estão todos os resultados de cada experimento feitos nesta seção e em E.2 -* ***Raios de Cobertura****, a figura E.2 mostra os melhores raios encontrados nos experimentos de a) a j) associados aos tamanhos dos retângulos de Lissajous e elipses onde estão os conjuntos de centros* C_1.

5.4.3 Recobrimento do Brasil

Para o exemplo do Brasil foram gerados 17255 pontos para um recobrimento com 10 centros.

O Retângulo Mínimo de Pontos mede:
$TamanhoX$ = 18.0000000000 e;
$TamanhoY$ = 18.0000000000.

A figura 5.20 mostra o Brasil inserido no Retângulo Mínimo e os dados mostrados na tabela 5.4.

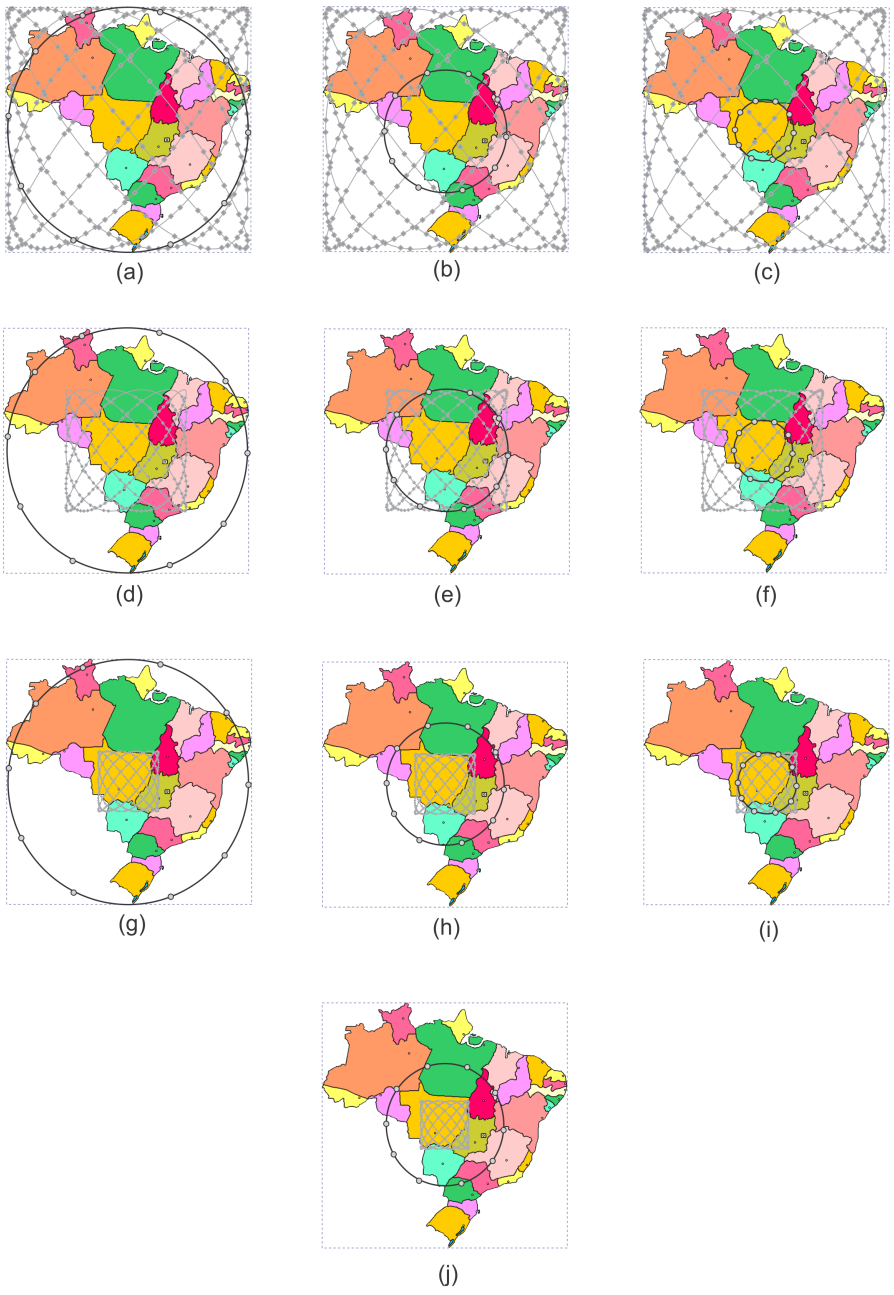

Figura 5.20: Brasil: C_1, Retângulo Mínimo e Curva de Lissajous.

O melhor resultado obtido com os experimentos para o recobrimento do Brasil foi com os seguintes parâmetros:

- Eixos da elipse de C_1 RE = $TamanhoX/2$ x $TamanhoY/2$
- Retângulo de Lissajous RL = $TamanhoX/5$ x $TamanhoY/5$
  (item j da Figura 5.20)

A solução do problema obtida foi:

- Maior Raio = 2.877809047699
- Valor de f = 8.3485116959
- Coordenadas dos 10 Centros:

1)	(16.493835449219 ,	12.586162567139)	Raio =	2.8774635792
2)	(12.453983306885 ,	12.998661041260)	Raio =	2.8734254837
3)	(13.326730728149 ,	16.456928253174)	Raio =	2.8778080940
4)	(8.430986404419 ,	16.682651519775)	Raio =	2.8775827885
5)	(4.677764892578 ,	16.534061431885)	Raio =	2.8721432686
6)	(3.781500816345 ,	12.936258316040)	Raio =	2.8747658730
7)	(7.891020774841 ,	11.780706405640)	Raio =	2.8778083324
8)	(10.568807601929 ,	8.412597656250)	Raio =	2.8778090477
9)	(10.996812820435 ,	3.788817405701)	Raio =	2.8745522499
10)	(14.768033981323 ,	7.883383750916)	Raio =	2.8778083324

- Tolerância Atingida = 0.00000048
- Tamanho do Passo da Base = 8.0
- t final de Lissajous = 344.43750
- Tempo de Execução para Passo da Base 8.0 = 153 min.

As Figuras 5.21 e 5.22 mostram a localização dos melhores centros e seus raios de cobertura na primeira e na última iteração do método para o recobrimento do Brasil.

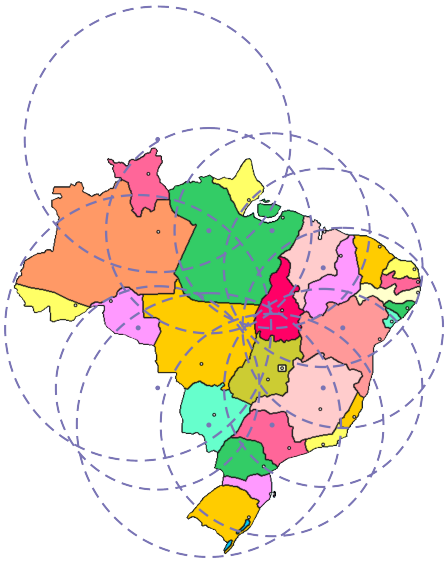

Figura 5.21: Primeira iteração.

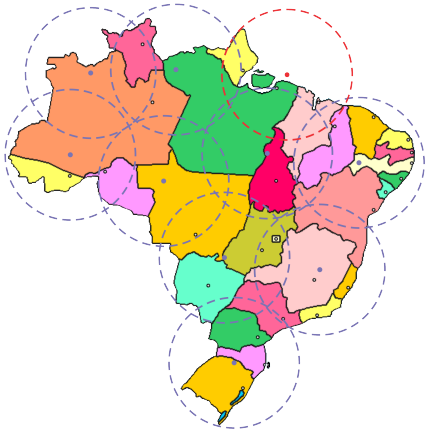

Figura 5.22: Última Iteração.

O Refinamento da Solução foi feito com tamanho do passo da base igual a 1.0 e a solução encontrada foi:

- Maior Raio = 2.876830339432
- Valor de f = 8.276152610779
- Coordenadas dos 10 Centros e seus Raios de Cobertura:

1)	(16.5072402954 ,	12.6152820587)	Raio =	2.8580803871
2)	(12.4530487061 ,	13.0180168152)	Raio =	2.8706054688
3)	(13.3118495941 ,	16.4819812775)	Raio =	2.8495504856
4)	(8.4400539398 ,	16.6758117676)	Raio =	2.8674497604
5)	(4.6944103241 ,	16.5268783569)	Raio =	2.8669042587
6)	(3.7801892757 ,	12.9452934265)	Raio =	2.8768131733
7)	(7.8966026306 ,	11.7860755920)	Raio =	2.8554689884
8)	(10.5727844238 ,	8.4087982178)	Raio =	2.8768203259
9)	(11.0150852203 ,	3.7863903046)	Raio =	2.8766851425
10)	(14.7716178894 ,	7.8805313110)	Raio =	2.8768303394

- Tolerância Atingida = 0.00000384

A figura 5.23 mostra o posicionamento das antenas encontradas na solução refinada.

Os experimentos para o recobrimento dos Estados do Rio de Janeiro do Pará e de Nova York foram feitos com o parâmetro de penalização $M = 1.0$ para a resolução do problema e as antenas ficaram posicionadas dentro destes estados.O refinamento da solução foi feita com o parâmetro $M = 0.0$.

Na solução encontrada para o recobrimento do Brasil uma antena ficou fora do país tanto na solução do problema como no refinamento da solução. Para resolver este problema o refinamento da solução foi feito aumentando o parâmetro de penalização M com o intutito de colocar a antena posicionada fora da região para dentro

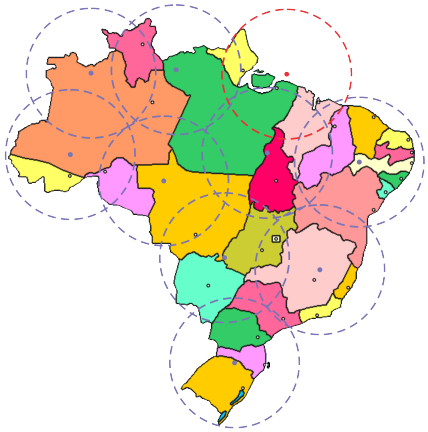

Figura 5.23: Refinamento com parâmetro de penalização $M = 0$.

desta.
A figura 5.24, mostra o movimento da antena para o interior da região conforme o parâmetro M é aumentado.

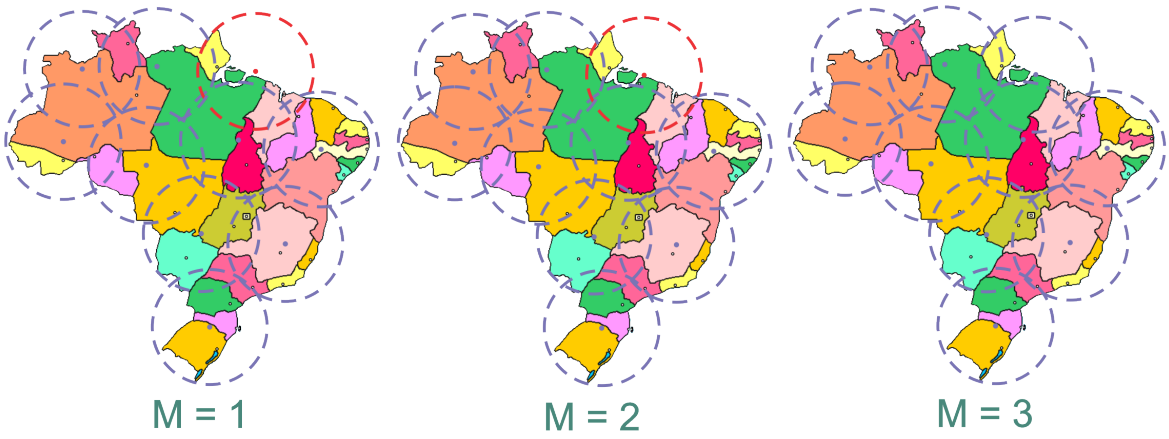

Figura 5.24: Refinamento com parâmetro de penalização $M = 1, 2, 3$.

A tabela a seguir mostra os valores da função objetivo f e dos maiores raios encontrados com os parâmetros de refinamento $M = 0$, $M = 1$, $M = 2$ e $M = 3$.

Tabela 5.6: Valores de f e dos maiores raios para $M = 0$, $M = 1$, $M = 2$ e $M = 3$

	Valor de f	Maior Raio
$M = 0$	8.276152610779	2.876830339432
$M = 1$	8.328153610229	2.876830816269
$M = 2$	8.196063041687	2.859779596329
$M = 3$	8.244339942932	2.869855403900

Obs: 5.10. *É interessante observar que em princípio era esperado que aumentando o parâmetro de penalização o valor do maior raio aumentasse, entretanto para $M = 2$ o maior raio diminuiu o que mostra que foi encontrado um novo mínimo local o que é justificado pela característica do problema de possuir uma* ***enorme*** *quantidade de mínimos locais.*

A melhor solução refinada encontrada com um centro fora da região S foi então com o parâmetro de penalização $M = 2$ com os seguintes resultados:

- Maior Raio = 2.859779596329
- Valor de f = 8.196063041687
- Coordenadas dos 10 Centros e seus Raios de Cobertura:
  1) (16.5612945557 , 12.5202846527) Raio = 2.8572940826
  2) (12.3306980133 , 12.9430875778) Raio = 2.8568992615
  3) (13.1179389954 , 16.2924137115) Raio = 2.8597795963
  4) (8.2782535553 , 16.5813217163) Raio = 2.8597757816
  5) (4.6243877411 , 16.4553852081) Raio = 2.8572025299
  6) (3.7148356438 , 12.9157686234) Raio = 2.8052008152
  7) (7.8407135010 , 11.7512569427) Raio = 2.8551595211
  8) (10.6295776367 , 8.2784852982) Raio = 2.8334875107
  9) (11.0508441925 , 3.7591714859) Raio = 2.8595094681
  10) (14.7129993439 , 7.8897719383) Raio = 2.8597700596
- Tolerância Atingida = 0.00000486
- Parâmetro de Penalização M = 2.0

A melhor solução refinada encontrada com todos os centros dentro da região S foi com o parâmetro de penalização $M = 3$ com os seguintes resultados:

- Maior Raio = 2.869855403900
- Valor de f = 8.244339942932

- Coordenadas dos 10 Centros e seus Raios de Cobertura:
  1) (16.6119232178 , 12.5603504181) Raio = 2.8770389557
  2) (12.4460811615 , 12.9906711578) Raio = 2.8691456318
  3) (13.0740060806 , 16.2456207275) Raio = 2.8764014244
  4) (8.5615301132 , 16.7158164978) Raio = 2.8766908646
  5) (4.8397064209 , 16.5861701965) Raio = 2.8342952728
  6) (3.6172900200 , 13.0235157013) Raio = 2.7610645294
  7) (7.8747539520 , 11.7917022705) Raio = 2.8754529953
  8) (10.6042671204 , 8.3784446716) Raio = 2.8742392063
  9) (11.0788898468 , 3.7591252327) Raio = 2.8770446777
  10) (14.7775230408 , 7.8888549805) Raio = 2.8770461082
- Tolerância Atingida = 0.00000470
- Parâmetro de Penalização M = 3.0

Obs: 5.11. *No Apêndice F -* ***Recobrimento do Brasil****, em F.1* ***Experimentos*** *estão todos os resultados de cada experimento feitos nesta seção e em F.2 -* ***Raios de Cobertura****, a figura F.2 mostra os melhores raios encontrados nos experimentos de a) a j) associados aos tamanhos dos retângulos de Lissajous e elipses onde estão os conjuntos de centros* C_1*.*

Com os experimentos feitos neste trabalho, os parâmetros das melhores soluções dos experimentos feitos para o recobrimento dos Estados do Rio de Janeiro, de Nova York, do Pará, e do Brasil são:

Tabela 5.7: Parâmetros das soluções do Rio de Janeiro, Nova York, Pará e Brasil.

	Eixos da elipse	Lados de Lissajous	Passo da base
Rio de Janeiro	$TamanhoX/2$ e $TamanhoY/2$	$TamanhoX/5$ e $TamanhoY/5$	12.0
Nova York	$TamanhoX/4$ e $TamanhoY/4$	$TamanhoX/4$ e$TamanhoX/4$	10.0
Pará	$TamanhoX/4$ e $TamanhoY/4$ $TamanhoX$ e $TamanhoY$	$TamanhoX/4$ e $TamanhoY/4$ $TamanhoX/2$ e $TamanhoY/2$	4.0 6.0
Brasil	$TamanhoX/2$ e $TamanhoY/2$	$TamanhoX/5$ e $TamanhoY/5$	8.0

Onde se conclui que não há um padrão de tamanhos de eixos de elipses, retângulos das Curvas de Lissajous e passo de base a ser seguido.

Capítulo 6

Conclusão

Neste trabalho, é importante observar que o algoritmo proposto (conforme o exposto no Capítulo 5) obteve sucesso em todas as suas execuções embora tenha uma estrutura bastante simples, no sentido de conseguir obter uma solução viável para o problema abordado o que demonstra a robustez do algoritmo. As soluções são compatíveis com aquelas expostas na literatura, sendo que neste trabalho elas obedecem a restrição de não violação de fronteira, como também não exigem interferência externa ou calibração periódica de quaisquer parâmetros.

Cabe observar que a execução do algoritmo proposto necessita de um tempo maior para a obtenção da solução devido à presença da penalização na função objetivo. Quando não há penalização (nesse caso, a restrição é ignorada), o tempo de execução do algoritmo reduz drasticamente.

O preço pago por todo o procedimento automatizado consiste em um tempo maior de execução do que o exibido nos algoritmos expostos na literatura. Entretanto, em tais algoritmos, embora o tempo de execução seja menor, é necessária a intervenção do usuário para o ajuste de parâmetros ao longo do tempo.

A restrição foi construída de tal maneira que consegue assegurar que os centros fiquem posicionados na mesma região dos pontos a serem recobertos, sem a necessidade de conhecer a fronteira de tal região (não foram necessários dados adicionais sobre a fronteira, o que seria muito dispendioso, e algumas vezes, e tais dados são desconhecidos).

Outro ponto importante é a geração do simplex inicial. O primeiro simplex é construído a partir de um único vértice de forma a assegurar a não degeneração desse simplex. Cada vértice do simplex corresponde a um possível posicionamento dos centros (coordenadas x e y).

A construção do primeiro vértice do simplex inicial foi feita de maneira que suas coodenadas (centros) residem sobre uma elipse, cujo centro é um ponto da curva de Lissajous, portanto pontos diferentes da curva de Lissajous geram centros diferentes das elipses dos primeiros vértices. Dessa forma, o método reinicia automaticamente

seguindo a trajetória dos pontos da curva de Lissajous. Essa característica permite a implementação do algoritmo proposto de forma parelela se desejável.

As propostas aqui apresentadas surgiram conforme o inesperado ou as dificuldades apareceram. A idéia inicial era a resolução de um problema *min-max-min* sem restrições pelo método de Nelder-Mead. Depois o primeiro vértice C_1 com o primeiro conjunto de centros foi posto sobre uma a elipse e o reinício do método era feito através pontos aleatórios. Para que os pontos de partida do método recobrissem o retângulo de pontos, o centro da elipse passou a ser um ponto da trajetória de uma curva de Lissajous. E finalmente, através de experimentos foi observado que algum centro poderia ficar fora da região geográfica a ser recoberta, para solucionar este problema a função objetivo foi penalizada de forma a colocar este centro para dentro da região e o problema passou a ser a resolução de um problema *min-max-min* com restrições pelo método de Nelder-Mead.

O algoritmo mostrou-se robusto e com um bom desempenho para a resolução de um problema de *min-max-min* com restrições, gerando uma solução viável e cuja formulação permite ser lapidada para contemplar outros casos, incorporando novas restrições para situações específicas dependendo do cenário do problema conforme será visto na seção a seguir.

Capítulo 7

Trabalhos Futuros

Com a formulação proposta neste trabalho, podem ser resolvidos problemas de *min-max-min* de forma a incluir restrições com as características próprias de cada problema conforme mencionado no seção anterior. Tais alterações permitem a resolução de problemas específicos de grande aplicação, tais como:

- **Localização de estações base em uma região não plana.**
  Para determinados problemas, a altura da localização de um conjunto ou da totalidade das estações base é variável.
  Exemplos em áreas *indoor* (internas), onde as estações base são alocadas no interior de um estabelecimento, como descrito em [11], passam a ser problemas em $\Re^3$, onde as coordenadas para a localização de cada estação base passam a ser x, y, z, diferentemente do problema tratado neste trabalho onde a região é plana e portanto cada centro está em $\Re^2$ com coordenadas x, y.

- **Localização de antenas com restrições quanto à localização.**
  Em alguns exemplos restrições quanto a localização das antenas devem ser levadas em conta, em locais onde a geografia não comporta antenas (lagos, etc).

- **Localização de antenas com restrições quanto a amplitude do sinal.**
  É o caso onde há a necessidade de alocar antenas em regiões onde existem obstáculos que impedem a propagação do sinal (morros).

- **Localização de antenas em uma região com algumas antenas já existentes.**
  Quando há um aumento de pontos a serem recobertos após algumas antenas já terem sido instaladas.

- **Localização de antenas com algumas em locais prédeterminados.**
  Se em uma determinada região há locais com uma grande concentração de

pontos a serem recobertos há a necessidade de alocar antenas nestes locais para haja um bom nível de cobertura.

Além disso:

- Após a realização dos experimentos, verificou-se que é possível usar a penalização apenas na fase do refinamento o que fará que o tempo de execução do algoritmo seja muito reduzido.

- O algoritmo pode ser feito de forma híbrida, começando com o método de Nelder-Mead visto ele reduz bastante a imagem da função objetivo nas primeiras iterações, e nas iterações posteriores usar outro método.

- O reinício do método com o uso da trajetória de uma curva de Lissajous pode ser usada com figuras geométricas diferentes da elipse proposta neste trabalho, podendo até a escolhas dos centros ser posicionada aleatoriamente numa vizinhança do ponto desta curva.

- Se da fronteira a região for totalmente conhecida (através da computação gráfica por exemplo), é possível realizar adaptações no método proposto para que os vértices gerados fiquem sempre dentro da região, retirando restrição.

- Os pontos da curva de Lissajous onde foram obtidos os melhores resultados foram para o Brasil:344.43750; Pará:352.75000 e 261.93750; Nova York:197.87500 e Rio:201.12500, não havendo, portanto um padrão a ser seguido. Como neste trabalho o parâmetro t variou de 0.0 a 500.0, próximos experimentos poderão ser feitos com um número de pontos maior ($t > 500.0$) para o reinício do método, com o intuito de ampliar a possibilidade de aumentar a qualidade da solução do problema.

Referências Bibliográficas

[1] R. Andreani, C. Dunder, and J. M. Martínez. Order-value optimization: formulation and solution by means of a primal cauchy method. *Mathematical Methods of Operations Research*, 58:387–399, 2003.

[2] R. Andreani, C. Dunder, and J. M. Martínez. Nonlinear-programming reformulation of the order-value optimization problem, mathematical methods of operations research. *Methods of Operations Research*, 61:365–384, 2005.

[3] R. Andreani, J. M. Martínez, M. Salvatierra, and F. Yano. Quasi-newton methods for order-value optimization and value-at-risk calculations. *Pacific Journal of Optimization*, 2:11–33, 2006.

[4] J. W. Bandler, P. C. Liu, and H. Tromp. A nonlinear programming approach to optimal design centering, tolerancing, and tuning. *IEEE Transactions on Circuits and Systems*, 23:155–165, 1976.

[5] M. S. Bazaraa, C. M. Shetty, and H. D. Sherali. *Nonlinear Programing-Theory and algotithms*. John Wiley and Sons, New York, 3th edition, 1979.

[6] K. Bhaskar and S. B. Wicker. Global search techniques for problems in mobile communications. *Telecommunications Optimization: Adaptive and Heuristic Approaches*, 2000.

[7] J. A. M. Brito and A. E. Xavier. Optimal covering of plane domains by circles via hyperbolic smoothing. *Journal of global optimization*, 31(3):493–504, 2005.

[8] T. Chen and M. K. H. Fan. On convex formulation of the floor-plan area minimization problem. *Proceedings of the International Symposium on Physical Design*, 1998.

[9] C. K. Cheng, X. Deng, Y. Z. Liao, and S. Z. Yao. Symbolic layout compaction under conditional design rules. *IEEE Transactions on Computer-Aided Design*, 11:475–486, 1992.

[10] A. R. Conn, K. Scheinberg, and L. N. Vicent. *Introduction to Derivative-Free Optimization*. SIAM, Philadelphia, 1st edition, 2009.

[11] S. J. Fortune, D. M. Gay, B. W. Kernighan, O. Landron, R. A. Valenzuela, and M. H. Wright. Wise design of indoor wireless systems: practical computation and optimization. *IEEE Computational Science and Engineering*, 2:52–68, 1995.

[12] J. Gaal, L. Gefferth, K. Geher, E. Halasz, and T. Tron. New algorithms and computer programs for design centering, tolerancing, and tuning under environmental influence. *Proceedings of the 1981 European Conference on Circuit Theory and Design, Delft University Press*, 1981.

[13] D. S. Hanif, C. M. Pendyala, and T. S. Rappaport. Optimal location of transmitters for micro-cellular radio communications system design. *IEEE Journal on Selected Areas in Communications*, 14:662–673, May 1996.

[14] D. Hichbaum. Complexity and algorithms for logical constraints with applications to vlsi layout, compaction, and clocking, studies in locational analysis. *Proceedings of ISOLDE*, 5:159–164, 1993.

[15] I. Howitt and S. Y. Ham. Base station location optimization. *in Proc. Vehicular Technology Conference, IEEE 50th*, 4:2067–2071, Sept. 1999.

[16] M. S. Júnior. *Otimização Global Usando Trajetórias Densas e Aplicações*. Tese de doutorado, Universidade Federal de Campinas, 2005.

[17] A. A. Kassim and K. V. Kumar. Path planning for autonomous robots using neural networks. *Journal of Intelligent Systems*, 7:33–55, 1997.

[18] C. Kirjner and E. Polak. On the conversion of optimization problems with maxmin constraints to standard optimization problems. *SIAM Journal on Optimization*, 8:887–915, 1998.

[19] J. Lagarias, B. Poonen, and M. H. Wright. Convergence of the restricted nelder-mead algorithm in two dimensions. *SIAM Journal on Optimization*, 22, 2012.

[20] J. Lagarias, J. Reeds, M. H. Wright, and P. E. Wright. Convergence proprieties of the nelder-mead simplex method in low dimentions. *SIAM Journal on Optimization*, 9:112–147, 1998.

[21] A. Lima, B. Rodrigues, F. Wang, and Z. Xu. k-center problems with minimum coverage. *Theoretical Computer Science*, 332:1–17, 2005.

[22] R. W. Longman, V. H. Schulz, and H. G. Bock. Path planning for satellite mounted robots, dynamics and control of structures in space. *Edited by C. L. Kirk and D. J. Inman Computational Mechanics Publications*, 3:17–28, 1996.

[23] G. R. Mateus. *Introdução à Computação Móvel.* 11a Escola de Computação, 1998.

[24] K. I. M. McKinnon. Convergence of the nelder-mead simplex method to a nonstationary point. *SIAM Journal on Optimization*, 9:148–158, 1998.

[25] T. S. Moh, T. S. Chang, and S. L. Hakimi. Globally optimal floorplanning for a layout problem. *IEEE Transactions on Circuits and Systems*, 43:713–720, 1996.

[26] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7:308–313, 1965.

[27] J. Nocedal and S. J. Wrigh. Derivative-free optimization. In P. Glynn and S. M. Robinson, editors, *Numerical Optimization*, pages 220–244. Springer-Verlag, New York, 2006.

[28] E. Polack and J. O. Royset. Algorithms for finite and semi-infinite min-max-min problems using adaptive smoothing techniques. *Journal of Optimization Theory and Applications*, 119:421–457, 2003.

[29] E. Polak. *Smoothing Techniques for the Solution of Finite and Semi-Infinite Min- Max-Min Problems.* High Performance Algorithms and Software for Nonlinear Optimization, Edited by G. Di Pillo and A. Murli, Kluwer Academic Publishers, Dordrecht, Netherlands, 2003.

[30] E. Polak, J. O. Royset, and R. S. Womersley. Algorithms with adaptive smoothing for finite min-max problems. *Journal of Optimization Theory and Application*, 119:457–484, 2003.

[31] E. Polak and A. S. Vingentelli. Theoretical and computational aspects of the optimal design centering, tolerancing, and tuning problem. *IEEE Transactions on Circuits and Systems*, 26:795–813, 1979.

[32] D. Ralph and E. Polak. A first-order algorithm for semi-infinite min-max- min problems. *SIAM Journal on Optimization (submitted)*, 2002.

[33] J. A. Reeds and L. A. Shepp. Optimal paths for a car that goes both forward and backward. *Pacifi Journal of Mathematics*, 145:367–393, 1990.

[34] W. Spendley, G. R. Hext, and F. R. Himsworth. Sequential application of simplex designs in optimization and evolucionary operation. *Technometrics*, 4:441–461, 1962.

[35] D. Stamatelos and A. Ephremides. Spectral efficiency and optimal baseplacement for indoor wireless networks. *IEEE Journal on Selected Areas in Communications*, 14:651–661, 1996.

[36] A. TITS. On the optimal design centering, tolerancing, and tuning problem, journal of optimization theory and applications. *Journal of Optimization Theory and Applications*, 45:487–494, 1985.

[37] V. Torczon. *Multi-directonal Search-A Direct Search Algorithm for Parallel Machines*. Ph.d. thesis, Rice University, Houston,TX, May 1989.

[38] V. Torczon. On convergence of the multidirectional search algorithm. *SIAM Journal on Optimization*, 1:123–145, 1991.

[39] P. Tseng. Fortified-descent simplicial search method. *SIAM Journal on Optimization*, 10:260–288, 1999.

[40] A. Voreadis and D. Q. Mayne. A cut map algorithm for design problems with parameter tolerances and tuning. *IEEE Transactions on Circuits and Systems*, 29:288–298, 1982.

[41] F. H. Walters, L. R. Parker, S. L. Morgan, and S. N. Deming. *Sequential Simplex Optimization*. CRC Press, USA, 1991.

[42] M. H. Wright. Optimization methods for base station placement in wireless applications. *IEEE Vehicular Technology Conference*, 1:387–391, May 1998.

[43] A. E. Xavier and J. A. M. Brito. Modelagens min-max-min para o problema de localização de estações de radio base. *Pesquisa Operacional*, 26(2):295–319, 2006.

[44] P. E. Yau. *On Algorithms for Nonlinear Minimax and Min-Max-Min Problems and theire Efficieny*. Ph.d. thesis, Naval Postgraduate School, 2011.

Apêndice A

Programa de Otimização sem Derivadas

Programa com Elipses e Curva de Lissajous

Obs: A.1. *No Programa, o conjunto de centros sobre a elipse será o vértice C_{n+1}, e não C_1, e os demais n conjuntos de centros C_1, C_2, C_3, . . . C_n serão portanto construídos a partir de C_{n+1}.*

* Preâmbulo *

```
unit untOtimiza;          {Recobrimento com Elipses e Curva de Lissajous}
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls;
type
TForm1 = class(TForm)
Label1: TLabel;
btnOtimiza: TButton;
ListOtimiza: TListBox;
Label5: TLabel;
edtChi: TEdit;
Label6: TLabel;
edtGama: TEdit;
Label7: TLabel;
edtRo: TEdit;
edtNumero_de_Iteracoes: TEdit;
Label4: TLabel;
Label8: TLabel;
lblProcessamento: TLabel;
Label12: TLabel;
```

```
Label13: TLabel;
edtNumPontos: TEdit;
Label14: TLabel;
edtnumcentros: TEdit;
Label15: TLabel;
edtSigma: TEdit;
Label10: TLabel;
EditEpsilon: TEdit;
Label9: TLabel;
edttiMax: TEdit;
Label2: TLabel;
edttiInicial: TEdit;
Label11: TLabel;
edtDeltati: TEdit;
Label16: TLabel;
Label17: TLabel;
edtMult: TEdit;
Label18: TLabel;
EditPasso: TEdit;
Label3: TLabel;
Label19: TLabel;
procedure btnOtimizaClick(Sender: TObject);
procedure Antena vs Usuário(Sender: TObject);
private   {Private declarations}
public    {Public declarations}
end;
var
Form1: TForm1;
implementation
{$R *.dfm}
Const
NumPontosMax = 20000;  {Número máximo de pontos lidos no plano}
numcentrosMax = 30;    {Número máximo de centros dos círculos}
type
DadosNumPontos = array[1..NumPontosMax,1..2] of single;
Linha_de_centros = array[1..2*numcentrosMax] of single;
procedure TForm1.btnOtimizaClick(Sender: TObject);
const Pi = 3.14159265;
var
Chi,FuncaoNovo,Funcaoexpandido,Sigma,
FuncaoContraiNovo,Gama,Ro,
```

```
MinX,MinY,MaxX,MaxY,TamanhoX,TamanhoY,
TamanhoXm,TamanhoYm,Alfa, Mult, Passo,PassoInicial,
ValorAnterior,MaiorDistancia,Envoltoria,Epsilon,
ti,tiMax,RAIO,deltati,tiInicial,Envoltoria_ate_agora,QGRZ,MaxMin,Maxmin_ate_agora,
QGteste,p,q:single;
QG,Tempo_de_Inicio,Tempo_Final,Dia:string;
I,J,K,M,N,O,Troca,Num,Numero_de_Iteracoes,NumReflexao1,
NumReflexao2,NumReflexao3,NumExtensao,NumContracaoExterna,NumContracaoInterna,
NumReducao,Itamanho1,Itamanho2,ItamanhoAnterior1,ItamanhoAnterior2,
iteracaoAnterior,numcentros,numpontos,ITERMAX,NUMLissajous,ilissajous,NumPassos:Integer;
Medio,Novo,Expandido,ContraiNovo,CentrosTeste,QGLINHADECENTROS: Linha_de_centros;
Ordem: array[1..2*numcentrosMax+1] of integer;
Centros: array[1..2*numcentrosMax+1] of Linha_de_centros;
Valor: array[1..2*numcentrosMax+1] of single;
Indice: array[1..2*numcentrosMax+1] of Integer;
Centro_do_retangulo: array[1..2] of single;
CentroAnterior: array[1..numcentrosMax,1..2] of single;
MAXVAR,MINVAR,DISTMAXMIN:array[1..numcentrosMax] of single;
ValorTeste: array[1..2,1..2,1..2,1..2,1..2,1..2] of single;
lissajous: array[1..32001,1..2] of single;
dados:DadosNumPontos;
arquivo2,arquivo3,arquivolissajous:textfile;
Hora_do_Dia,Dia_do_teste:TDateTime;
```

* Fim do Preâmbulo *

* Função a ser minimizada *

```
Function Funcao(L:Linha_de_centros;dados:DadosNumPontos):single;
var
I,K:Integer;
QGR,Min,MaxMin:single;
begin
MaxMin := 0.0;
For I := 1 to NumPontos do Begin
Min := (L[1]-dados[I,1])*(L[1]-dados[I,1]) + (L[2]-dados[I,2])*(L[2]-dados[I,2]);
For K := 2 to numcentros do Begin
QGR := (L[2*K-1]-dados[I,1])*(L[2*K-1]-dados[I,1])+(L[2*K]-dados[I,2])*(L[2*K]-dados[I,2];
IF Min > QGR then Min := QGR; end;
IF MaxMin < Min then MaxMin := Min;
end;
QGRZ := 0.0;
For K := 1 to numcentros do BEGIN
```

```
Min := (L[2*K-1]-dados[1,1])*(L[2*K-1]-dados[1,1])+(L[2*K]-dados[1,2])*(L[2*K]-dados[1,2];
For I := 2 to NumPontos do Begin
QGR := (L[2*K-1]-dados[I,1])*(L[2*K-1]-dados[I,1])+(L[2*K]-dados[I,2])*(L[2*K]-dados[I,2];
IF QGR < Min then Min := QGR;
end;
IF Min > QGRZ then QGRZ := Min;
end;
result := MaxMin + Mult*QGRZ;
end;
```

* Fim da Função a ser minimizada *

C: Vetor Indice indica a ordem dos valores (MaxMin) de cada conjunto de centros, Indice[1] é a linha de centros de menor valor

* Procedure Classifica os valores da função *

```
Procedure Classifica;
var
I,J:Integer;
Begin
For J := 2*numcentros downto 1 do
For I := 1 to J do
If Valor[Indice[I]] > Valor[Indice[I+1]]
then Begin
Troca := Indice[I];
Indice[I] := Indice[I+1];
Indice[I+1] := Troca;
end;  end;
```

C: Classifica coloca em Indice[1] o número da linha de centros de menor valor de f e Indice[2*numcentros +1] o número da linha de centros de maior valor de f

* Fim da Procedure Classifica os valores da função *

* Procedure Preparação dos Arquivos *

```
Procedure PreparacaodeArquivos;
Begin
AssignFile(arquivo2,
'C:\Users\angela\Searches\Minmaxmin elipses-Brasil\Brasil172555.txt');
reset (arquivo2);
AssignFile(arquivolissajous,
'C:\Users\angela\Searches\Minmaxmin elipses-Brasil\dadoslissajous_25.txt');
reset (arquivolissajous);
AssignFile(arquivo3,
```

```
'C:\Users\angela\Searches\Minmaxmin elipses-Brasil\historico.txt');
{$I-}
reset (arquivo3);
{$I+}
I := IOResult;
If I = 2 then Rewrite(arquivo3)
else Append(arquivo3);
end;
```

* Fim da Procedure Preparação dos Arquivos *

* Procedure Entrada dos Parâmetros *

```
Procedure Parametros;
Begin
Chi := StrToFloat(edtChi.text);
Gama := StrToFloat(edtGama.text);
Ro := StrToFloat(edtRo.text);
Sigma := StrToFloat(edtSigma.text);
Numero_de_Iteracoes := StrToInt(edtNumero_de_Iteracoes.text);
NumPontos := StrToInt(edtNumPontos.Text);
numcentros := StrToInt(edtnumcentros.Text);
Epsilon:= StrToFloat(editEpsilon.text);
tiMax := StrToFloat(edttiMax.text);
tiInicial := StrToFloat(edttiInicial.text);
deltati := StrToFloat(edtdeltati.text);
Mult := StrToFloat(edtMult.text);
PassoInicial := StrToFloat(editPasso.text);
```

* Fim da Procedure Entrada dos Parâmetros *

* Impressão dos parâmetros *

```
writeln(arquivo3);
QG := ' ************************************************************************** ';
writeln(arquivo3,QG);
writeln(arquivo1,QG);
QG:=' ************************* Histórico de Resultados ************************ ';
lblProcessamento.Caption := 'Processando';
writeln(arquivo3,QG);
writeln(arquivo3);
write(arquivo3,' Número de Iterações =',Numero_de_Iteracoes:5 );
writeln(arquivo3,' Tolerância = ',Epsilon:8:7);
write(arquivo3,' Número de centros =',numcentros:4);
```

```
write(arquivo3,' Número de pontos =',numpontos:9);
writeln(arquivo3);
writeln(arquivo3,' t inicial de Lissajous =',tiInicial:10:4);
writeln(arquivo3,' t final de Lissajous =',tiMax:10:4);
writeln(arquivo3,' Delta t de Lissajous =',deltati:10:4);
writeln(arquivo3);
end;
```

* Fim da Impressão dos parâmetros *

* Procedure Zerar Contadores *

```
Procedure Zera_Contadores;
Begin
NumReflexao1:=0;
NumReflexao2:=0;
NumExtensao:=0;
NumContracaoExterna:=0;
NumContracaoInterna:=0;
NumReducao:=0;
end;
```

* Fim da Procedure Zerar Contadores *

* Procedure Leitura dos Dados de Entrada *

```
Procedure Ler_Dados_Entrada;
var
I,J:Integer;
Begin
For I := 1 to NumPontos do Begin       {Ler Pontos a serem recobertos}
For J := 1 to 2 do
read(arquivo2,dados[I,J]);
readln(arquivo2); end;
For I := 1 to 8000 do Begin            {Ler Pontos de Lissajous}
For J := 1 to 2 do
read(arquivolissajous,lissajous[I,J]);
readln(arquivolissajous); end;
end;
```

* Fim da Procedure Leitura dos Dados de Entrada *

* Procedure Construção do Retângulo de Pontos *

```
Procedure Retangulo_dos_Pontos;
var
I:Integer;
Begin
MinX:= Dados[1,1];MinY := Dados[1,2];
MaxX := Dados[1,1];MaxY := Dados[1,2];
For I := 2 to Numpontos do Begin
If MinX > Dados[I,1] then MinX := Dados[I,1];
If MinY > Dados[I,2] then MinY := Dados[I,2];
If MaxX < Dados[I,1] then MaxX := Dados[I,1];
If MaxY < Dados[I,2] then MaxY := Dados[I,2];
end;
QG := ' Min de X = ' + FloatToStrF(MinX,ffFixed,15,10);
listOtimiza.Items.Add(QG);
writeln(arquivo3,QG);
QG := ' Max de X = ' + FloatToStrF(MaxX,ffFixed,15,10);
listOtimiza.Items.Add(QG);
writeln(arquivo3,QG);
QG := ' Min de Y = ' + FloatToStrF(MinY,ffFixed,15,10);
listOtimiza.Items.Add(QG);
writeln(arquivo3,QG);
QG := ' Max de Y = ' + FloatToStrF(MaxY,ffFixed,15,10);
listOtimiza.Items.Add(QG);
writeln(arquivo3,QG);
QG := '  ';
listOtimiza.Items.Add(QG);
writeln(arquivo3,QG);
TamanhoX := MaxX - MinX;
TamanhoY := MaxY - MinY;
TamanhoXm := TamanhoX/4.0;
TamanhoYm := TamanhoY/4.0;
writeln(arquivo3,' TamanhoX = ',TamanhoX:15:10);
writeln(arquivo3,' TamanhoY = ',TamanhoY:15:10);
writeln(arquivo3);
Centro_do_retangulo[1] := MinX + TamanhoX/2.0;
Centro_do_retangulo[2] := MinY + TamanhoY/2.0;
end;
```

* Fim da Procedure Construção do Retângulo de Pontos *

* Procedure Geração dos Centros *

```
Procedure Gera_Centros;
```

```
var
J,I:Integer;
```

C: Geração do Primeiro Conjunto de Centros

C: Esses centros gerados ficarão na última linha da matriz Centros

```
Begin
For J := 1 to numcentros do Begin
Centros[2*numcentros+1,2*J-1] := TamanhoXm * lissajous[ilissajous,1] +
Centro_do_retangulo[1]+ (TamanhoXm/2) * cos(Alfa);
Centros[2*numcentros+1,2*J] := TamanhoYm * lissajous[ilissajous,2] +
Centro_do_retangulo[2] + (TamanhoYm/2) * sin(Alfa);
QG := ' Centros -> '+ FloatToStrF(Centros[2*numcentros+1,2*J-1],ffFixed,15,10)+
' '+FloatToStrF(Centros[2*numcentros+1,2*J],ffFixed,15,10);
listOtimiza.Items.Add(QG);
```

C: Gira alfa para gerar os outros centros a partir do Primeiro Conjunto de Centros S.S.O.

```
Alfa := Alfa + 2.0*PI/numcentros;
end;
q := (Passo *(SQRT(2*numcentros+1)-1)/(2*numcentros*SQRT(2.0)));
p := (Passo *(SQRT(2*numcentros+1)+(2*numcentros-1))/(2*numcentros*SQRT(2.0)));
For I := 1 to 2*numcentros do
For J := 1 to 2*numcentros do
Centros[I,J] := Centros[2*numcentros+1,J] + q;
For I := 1 to 2*numcentros do
Centros[I,I] := Centros[2*numcentros+1,I] + p;
end;
```

* Fim da Procedure Geração dos Centros *

* Procedure Centróide e Ponto Refletido *

```
Procedure Centroide;
var
I,J:Integer;
Begin
For I := 1 to 2*Numcentros do Begin
Medio[I] := 0.0;
For J := 1 to 2*numcentros do
Medio[I] := Medio[I] + Centros[Indice[J],I];
Medio[I] := Medio[I]/(2*numcentros);
Novo[I] := (1.0+ro)*Medio[I]-ro*Centros[Indice[2*numcentros+1],I];
end;
FuncaoNovo:= Funcao(Novo,dados);
end;
```

* Fim da Procedure Centróide e Ponto Refletido *

* Procedure Redução *

```
Procedure Reducao;
var
I,K:Integer;
Procedure Classifica;
var
I,J:Integer;
Begin
For J := 2*numcentros downto 1 do
For I := 1 to J do
If Valor[Indice[I]] > Valor[Indice[I+1]]
then Begin
Troca := Indice[I];
Indice[I] := Indice[I+1];
Indice[I+1] := Troca;
end;
end;
Begin
Inc(NumReducao);
For I := 2 to 2*numcentros+1 do begin
For K := 1 to 2*numcentros do
Centros[Indice[I],K] := Centros[Indice[1],K]-
Sigma*(Centros[Indice[1],K]-Centros[Indice[I],K]);
Valor[Indice[I]] := Funcao(Centros[Indice[I]],dados);
end;
For I := 1 to 2*numcentros+1 do
Indice[I] := I;                                          {Classifica Pontos}
Classifica;
end;
```

* Fim da Procedure Redução *

* Procedure Valor Refletido Melhor *

```
Procedure ValorRefletidoMelhor;
var
I,K:Integer;
begin
Inc(NumReflexao1);
For I := 1 to 2*numcentros do          {vai tentar expandir}
```

```
expandido[I] :=(1.0 + Ro*Chi)*Medio[I]-Ro*Chi*Centros[Indice[2*numcentros+1],I];;
Funcaoexpandido := Funcao(expandido,dados);
IF Funcaoexpandido < FuncaoNovo
then Begin                {A Expansão foi aceita}
Inc(NumExtensao);
NumReflexao1 := NumReflexao1 - 1;
For I := 1 to 2*numcentros do
Novo[I] := expandido[I];
FuncaoNovo := Funcaoexpandido;
end;                      {valor refletido ou expandido vai entrar como o melhor}
For K := 2*numcentros+1 downto 2 do begin
For I := 1 to 2*numcentros do
Centros[Indice[K],I] := Centros[Indice[K-1],I];
Valor[Indice[K]] := Valor[Indice[K-1]];
end;                      {entrada do novo melhor valor}
For I := 1 to 2*numcentros do
Centros[Indice[1],I] := Novo[I];
Valor[Indice[1]] := FuncaoNovo;       end;
```

* Fim da Procedure Valor Refletido Melhor *

* Procedure Novo Valor Melhor que Penúltimo *

```
Procedure NovoValorMelhorQuePenultimo;
var
I,J,K : Integer;
begin {B11}
Inc(NumReflexao2);                    {testa se o refletido será aceito}
For I := 2 to 2*numcentros do begin {B12}
IF FuncaoNovo < Valor[Indice[I]] then
begin                                 {determina onde deve entrar o refletido}
For K := 2*numcentros+1 downto I+1 do begin
For J := 1 to 2*numcentros do         {deslocamento dos piores que o refletido}
Centros[Indice[K],J] := Centros[Indice[K-1],J];
Valor[Indice[K]] := Valor[Indice[K-1]];
end;
For J := 1 to 2*numcentros do          {A Reflexão foi aceita}
Centros[Indice[I],J] := Novo[J];
Valor[Indice[I]] := FuncaoNovo;
break;
end;
end;
end;
```

* Fim da Procedure Novo Valor Melhor que Penúltimo *

* Procedure Novo Valor Melhor que Último *

```
Procedure NovoValorMelhorQueUltimo;
var
I,J,K :Integer;
begin {Vai tentar Contracao Externa}
For I := 1 to 2*numcentros do
ContraiNovo[I] := (1.0 + Ro*Gama)*Medio[I]-
Ro*Gama*Centros[Indice[2*numcentros+1],I];
FuncaoContraiNovo := Funcao(ContraiNovo,dados);{Testa se Contraído é melhor que Refletido}
IF FuncaoContraiNovo < FuncaoNovo then         {A Contracao Externa foi aceita}
begin
Inc(NumContracaoExterna);
For I := 1 to 2*numcentros+1 do begin          {Posiciona o novo Centro que entrou}
IF FuncaoContraiNovo < Valor[Indice[I]] then
begin
For K := 2*numcentros+1 downto I+1 do begin
For J := 1 to 2*numcentros do
Centros[Indice[K],J] := Centros[Indice[K-1],J];
Valor[Indice[K]] := Valor[Indice[K-1]];
end;
For J := 1 to 2*numcentros do
Centros[Indice[I],J] := ContraiNovo[J];
Valor[Indice[I]] := FuncaoContraiNovo;
break;
end;
end;
end else REDUCAO;                               {A redução foi aceita}
end;
```

* Fim da Procedure Novo Valor Melhor que Último *

* Procedure Novo Valor Pior ou Igual ao Último *

```
Procedure NovoValorPiorOuIgualaoUltimo;
Var
I,J,K:Integer;
begin                                  {Vai testar a Contracao Interna}
For I := 1 to 2*numcentros do
ContraiNovo[I] := (1.0 - Gama)*Medio[I]+Gama*Centros[Indice[2*numcentros+1],I];
FuncaoContraiNovo := Funcao(ContraiNovo,dados);
```

```
IF FuncaoContraiNovo >= Valor[Indice[2*numcentros+1]] then Reducao
else Begin                                          {A Contração Interna foi aceita}
Inc(NumContracaoInterna);
For I := 1 to 2*numcentros + 1 do begin
IF FuncaoContraiNovo < Valor[Indice[I]] then
begin
For K := 2*numcentros+1 downto I+1 do begin
For J := 1 to 2*numcentros do
Centros[Indice[K],J] := Centros[Indice[K-1],J];
Valor[Indice[K]] := Valor[Indice[K-1]];
end;
For J := 1 to 2*numcentros do
Centros[Indice[I],J] := ContraiNovo[J];
Valor[Indice[I]] := FuncaoContraiNovo;
break;
end;
end;
end;
end;
```

* Fim da Procedure Novo Valor Pior ou Igual ao Último *

* Procedure Teste Final *

```
Procedure TesteFinal(L:Linha_de_centros;dados:DadosNumPontos);
var
I,K,IPT,SomaNumPontos:Integer;
QGR,Min,MaxMin:single;
MenorDistPonto_Centro: array[1..numpontosMax] of single;
MaiorDistCentro_Ponto: array[1..numcentrosMax] of single;
PontoMaisDistCentro,NumeroPontosCentro: array[1..numcentrosMax] of Integer;
CentroMaisPerto: array[1..numpontosMax] of Integer;
begin
MaxMin := 0.0;
For I := 1 to NumCentros do Begin
MaiorDistCentro_Ponto[I] := 0.0;
PontoMaisDistCentro[I] := 0;
NumeroPontosCentro[I] := 0;
end;
For I := 1 to NumPontos do Begin
Min := (L[1]-dados[I,1])*(L[1]-dados[I,1]) + (L[2]-dados[I,2])*(L[2]-dados[I,2]);
Min := 1000000.0;
IPT := 100;
```

```
For K := 1 to numcentros do Begin
QGR := (L[2*K-1]-dados[I,1])*(L[2*K-1]-dados[I,1])+(L[2*K]-dados[I,2])*(L[2*K]-dados[I,2]);
IF Min > QGR then Begin
Min := QGR;
IPT := K; end;
end;
IF MaxMin < Min then MaxMin := Min;
INC(NumeroPontosCentro[IPT]);
CentroMaisPerto[I] := IPT;
MenorDistPonto_Centro[I] := Min;
IF MaiorDistCentro_Ponto[IPT] < Min then Begin
MaiorDistCentro_Ponto[IPT] := Min;
PontoMaisDistCentro[IPT] := I;
end;
end;
SomaNumPontos := 0;
For I := 1 to numcentros do Begin
SomaNumPontos := SomaNumPontos + NumeroPontosCentro[I];
write(arquivo3, L[2*(I-1)+1]:8:4);
write(arquivo3, L[2*(I-1)+2]:8:4);
write(arquivo3,' Número de Pontos = ',NumeroPontosCentro[I]:4);
QGteste := MaiorDistCentro_Ponto[I];
QGteste := sqrt(QGteste);
write(arquivo3,' Ponto Mais Distante = ',
dados[PontoMaisDistCentro[I],1]:6:2, dados[PontoMaisDistCentro[I],2]:6:2,
' Distância = ', QGteste:15:10);
writeln(arquivo3);
end;
writeln(arquivo3,' Total de Pontos = ',SomaNumPontos:5);
RAIO:=sqrt(Maxmin);
writeln(arquivo3);
write(arquivo3,' Melhor Valor da Função = ',ValorAnterior:15:10);
write(arquivo3,' Raio Máximo = ',RAIO:15:10);
writeln(arquivo3);
end;
```

* Fim da Procedure Teste Final *

* Procedure Nelder-Mead *

```
Procedure Nelder_Mead;
var
I,J: Integer;
```

```
Begin
NUM := 0;
repeat                                  {Inicio das Iterações}
Inc(NUM);
If FuncaoNovo < Valor[Indice[1]] then ValorRefletidoMelhor
else If FuncaoNovo < Valor[Indice[2*numcentros]] then
NovoValorMelhorQuePenultimo
else If FuncaoNovo < Valor[Indice[2*numcentros+1]] then
NovoValorMelhorQueUltimo
else NovoValorPiorOuIgualaoUltimo;
```

* Critério de Parada *

```
MaiorDistancia := 0.0;
Envoltoria := 0.0;
FOR I := 1 to 2*numcentros do Begin
MAXVAR[I] := Centros[1,I];
MINVAR[I] := Centros[1,I];
FOR J := 2 to 2*numcentros+1 do
IF Centros[J,I] > MAXVAR[I]
then MAXVAR[I] := Centros[J,I]
else IF Centros[J,I] < MINVAR[I]
then MINVAR[I] := Centros[J,I];
DISTMAXMIN[I] := MAXVAR[I] - MINVAR[I];
Envoltoria := Envoltoria + DISTMAXMIN[I]*DISTMAXMIN[I];
IF DISTMAXMIN[I] > MaiorDistancia then MaiorDistancia := DISTMAXMIN[I];
end;
Envoltoria := Sqrt(Envoltoria);
IF Envoltoria < Epsilon then Begin
ITERMAX := NUM; break; end;            {Fim do Critério de Parada}
until NUM = Numero_de_Iteracoes;
end;
```

* Fim da Procedure Nelder-Mead *

* Procedure Testa Se Melhorou *

```
procedure Testa_se_Melhorou;
Var
J:Integer;
Begin                       {Guarda os melhores valores de f e dos centros até agora}
IF Valor[Indice[1]]<ValorAnterior then Begin
ValorAnterior := Valor[Indice[1]];
For J := 1 to numcentros do Begin
```

```
CentroAnterior[J,1]:=Centros[Indice[1],2*J-1];
CentroAnterior[J,2]:=Centros[Indice[1],2*J];
end;
QG := ' Conjunto de Centros com seus valores classificados após '+
IntToStr(NUM)+' Iterações' + ' t Lissajous = ' + FloatToStrF(ti,ffFixed,19,4) +
' Tolerancia = '+ FloatToStrF(Envoltoria,ffFixed,9,8);
Envoltoria_ate_agora:= Envoltoria;
writeln(arquivo3,QG);
QG:= ' Centros ' + IntToStr(Indice[1])+ ' ';
For J := 1 to 2*numcentros do
QG := QG + FloatToStrF(Centros[Indice[1],J],ffFixed,15,10) + ' ' ;
writeln(arquivo3,QG);
QG := ' Melhor Valor da Função -> ' + ' '+ FloatToStrF(Valor[Indice[1]],ffFixed,15,10);
listOtimiza.Items.Add(QG);
writeln(arquivo3,QG);
writeln(arquivo3);
end;
end;
```

* Fim da Procedure Testa Se Melhorou *

* Procedure Impressões Finais *

```
procedure Impressoes_Finais;
Var
I,J:Integer;
Begin
lblProcessamento.Caption := 'FINAL';
writeln(arquivo3,' Número de Reflexões com Tentativa de Expansão = ',NumReflexao1:5);
writeln(arquivo3,' Número de Reflexões Diretas = ',NumReflexao2:5);
J:= NumReflexao1+NumReflexao2;
writeln(arquivo3,' Número de Reflexões Total = ',J:7);
writeln(arquivo3,' Número de Expansões = ',NumExtensao:5);
writeln(arquivo3,' Número de Contrações Externas = ',NumContracaoExterna:5);
writeln(arquivo3,' Número de Contracões Internas = ',NumContracaoInterna:5);
writeln(arquivo3,' Número de Reduções = ',NumReducao:5);
I := NumReflexao1+NumReflexao2+NumExtensao+NumContracaoExterna+
NumContracaoInterna+NumReducao;
writeln(arquivo3,' Número Total de Iterações = ',I:9);
K:= NumReflexao2 + 2*(NumReflexao1 + NumExtensao + NumContracaoExterna +
NumContracaoInterna) + (2*numcentros+2)*NumReducao;
writeln(arquivo3,' Número Total de Avaliações de f = ',K:9);
writeln(arquivo3,' Peso da Restrição de f = ',Mult:3:2);
```

```
write(arquivo3,' Tamanho do Passo da Base = ',Passo:3:1);
writeln(arquivo3);
writeln(arquivo3,' Tolerância Atingida = ',Envoltoria_ate_agora:9:8);
writeln(arquivo3);
QG := '  ';
Hora_do_Dia := Time;
Tempo_Final := TimeToStr(Hora_do_Dia);
QG := 'Inicio às '+ Tempo_de_Inicio + ' Final às '+ Tempo_Final;
For J := 1 to numcentros do Begin
For I := 1 to 2 do
QGLINHADECENTROS[2*(J-1)+I] := CentroAnterior[J,I];
end;
end;
```

* Fim da Procedure Impressões Finais *

*** INÍCIO DO PROGRAMA DE MINIMIZAÇÃO ***

```
begin
Hora_do_Dia := Time;
Dia_do_teste := Date;
Tempo_de_Inicio := TimeToStr(Hora_do_Dia);
DIA := DateToStr(Dia_do_teste);
PreparacaodeArquivos;
Parametros;                              {Leitura de parametros}
Ler_Dados_Entrada;                       {Leitura dos Dados de Entrada}
Retangulo_dos_Pontos;                    {Gravacao_Dados_Entrada;}
NumPassos := 1;                          {Rotina da mudanças dos passos}
Passo := PassoInicial;
repeat
ValorAnterior := 100000.0; // reinicializa o melhor valor
ti := tiInicial - deltati;
ilissajous :=StrtoInt( FloattoStr(tiInicial/deltati));
Zera_Contadores;                         {Zera os Contadores}
repeat // início das iterações
inc(ilissajous);
ti := ti + deltati;
alfa:=0.0;
Gera_Centros;                            {Valor[I] representa o valor de f para
                                          a linha de centros Centros[I]}
For I := 1 to 2*numcentros+1 do Begin    {Calcula Valores }
Valor[I] := Funcao(Centros[I],dados);
Indice[I] := I;
end;
Classifica;
Nelder_Mead;
Testa_se_Melhorou;
until ti >= tiMax;
```

C: Mais um ponto de partida para aplicar Nelder-Mead

*** FIM DO PROGRAMA DE MINIMIZAÇÃO ***

```
Impressoes_Finais;
TesteFinal(QGLINHADECENTROS,dados);
writeln(arquivo1);
writeln(arquivo3);
Write(arquivo3,' Inicio às ',Tempo_de_Inicio,' Final às ',Tempo_Final );
Write(arquivo3, ' ', Dia);
writeln(arquivo3);
Passo := Passo + 1.0;
Inc(NumPassos);
until NumPassos > 12.2;                {Fim da rotina de variação dos Passos}
QG := ' FIM ';
listOtimiza.Items.Add(QG);
listOtimiza.Items.Add(QG);
CloseFile(arquivo2);
CloseFile(arquivo3);
CloseFile(arquivolissajous);
end;
```

* Procedure Antena vs Usuário *

```
procedure TForm1.Button1Click(Sender: TObject);
var
dados:DadosNumPontos;
arquivo4,arquivo5:textfile;
I,IPT,NumPontos:Integer;
X,Y,MIN,DISTANCIA:single;
MenorDistPonto_Centro: array[1..numpontosMax] of single;
CentroMaisPerto: array[1..numpontosMax] of Integer;
begin
NumPontos := StrToInt(edtNumPontos.Text);
AssignFile(arquivo4,'C:\Users\angela\Searches\Minmaxmin elipses-Brasil\Torres.txt');
reset(arquivo4);
AssignFile(arquivo5,'C:\Users\angela\Searches\Minmaxmin elipses-Brasil\sintese.txt');
{$I-}
reset (arquivo5);
{$I+}
I := IOResult;
If I = 2 then Rewrite(arquivo5)
else Append(arquivo5);
MIN := 10000.0;
X := StrtoFloat(edtX.Text);
Y := StrtoFloat(edtY.Text);
FOR I := 1 to numpontos do    Begin
```

```
read(arquivo4,dados[I,1]);
read(arquivo4,dados[I,2]);
read(arquivo4,CentroMaisPerto[I]);
read(arquivo4,MenorDistPonto_Centro[I]);
readln(arquivo4);
DISTANCIA := (dados[I,1] - X)*(dados[I,1] - X)+(dados[I,2] - Y)*(dados[I,2] - Y);
IF DISTANCIA < MIN then BEGIN
MIN := DISTANCIA;
IPT := I;                                    end;
                              end;
listOtimiza.items.add(' A antena que alimenta esse ponto X = '+
FloatToStrF(X,ffFixed,5,1)+ '   Y = '+FloatToStrF(Y,ffFixed,5,1)+' é a '+
InttoStr(CentroMaisPerto[IPT]));
writeln(arquivo5,' Testando o Obstáculo ');
writeln(arquivo5,' A antena que alimenta o ponto X = ',
X:5:2,'  Y = ',Y:5:2,' é a ',CentroMaisPerto[IPT]:2);
CloseFile(arquivo4);
CloseFile(arquivo5);
end;
```

* Fim da Procedure Antena vs Usuário *

```
end.
```

Apêndice B

Programas Auxiliares

B.1 Programa Gerar Dados

```
unit untOtimiza;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls;
type
TForm1 = class(TForm)
Label1: TLabel;
Button1: TButton;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Label2: TLabel;
edtDelta: TEdit;
procedure Button1Click(Sender: TObject);
private   {Private declarations}
public    {Public declarations}
end;
var
Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var
```

```
X,Y,inclinacao,XFF,delta:single;
I,numpontos: INTEGER;
XI,XF: array[1..1000,1..2] of single;
Modo: array[1..1000] of Integer;
arquivo1,arquivo2:textfile;
begin
delta := StrtoFloat(edtdelta.text);
AssignFile(arquivo1,'C:\Users\angela\Searches\Gerar Dados\Para\Triang_e_Quadr.txt');
reset(arquivo1);
AssignFile(arquivo2,'C:\Users\angela\Searches\Gerar Dados\Para\Cobertura.txt');
Rewrite(arquivo2);
readln(arquivo1,numpontos);
For I := 1 to numpontos do Begin
read(arquivo1,XI[I,1],XF[I,1],XI[I,2],XF[I,2],Modo[I]);
readln(arquivo1);
end;
{X := XI; Y := YI;}
For I := 1 to numpontos do Begin
inclinacao := (XF[I,2] - XF[I,1])/(XI[I,2] - XI[I,1]);
Case Modo[I] of
1: Begin
Y := XF[I,1];
repeat
X := XI[I,1];
repeat
writeln(arquivo2,X:8:3,Y:8:3);
X := X+delta;
until X > XI[I,2]+0.01;
Y := Y + delta;
until Y > XF[I,2]+0.01;
end;
2: Begin
Y := XF[I,1];
XI[I,1] := XI[I,1] - delta / inclinacao;
repeat
XI[I,1] := XI[I,1] + delta / inclinacao;
X := XI[I,1];
repeat
writeln(arquivo2,X:8:3,Y:8:3);
X := X + delta;
until X > XI[I,2]+0.01;
```

```
Y := Y + delta;
until Y > XF[I,2]+0.01;
end;
3: Begin
Y := XF[I,1];
XFF := XI[I,1] - delta / inclinacao;
repeat
X := XI[I,1];
XFF := XFF + delta / inclinacao;
repeat
writeln(arquivo2,X:8:3,Y:8:3);
X := X + delta;
until X > XFF+0.01;
Y := Y + delta;
until Y > XF[I,2]+0.01;
end;
4: Begin
Y := XF[I,2];
XFF := XI[I,1] - delta / inclinacao;
repeat
X := XI[I,1];
XFF := XFF + delta / inclinacao;
repeat
writeln(arquivo2,X:8:3,Y:8:3);
X := X + delta;
until X > XFF+0.01;
Y := Y - delta;
until Y < XF[I,1]-0.01;
end;
5: Begin
Y := XF[I,2];
XI[I,1] := XI[I,1] - delta / inclinacao;
repeat
XI[I,1] := XI[I,1] + delta / inclinacao;
X := XI[I,1];
repeat
writeln(arquivo2,X:8:3,Y:8:3);
X := X + delta;
until X > XI[I,2]+0.01;
Y := Y - delta;
until Y < XF[I,1]-0.01;
```

```
end;
end;
end;
CloseFile(arquivo1);
CloseFile(arquivo2);
end;
end.
```

B.2 Programa Contar Números

```
unit untOtimiza;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls;

type
TForm1 = class(TForm)
Label1: TLabel;
Button1: TButton;
Label6: TLabel;
edtN: TEdit; {Contar Número de Pontos}
procedure Button1Click(Sender: TObject);
private   {Private declarations}
public    {Public declarations}
end;
var
Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var
XI:single;
N: INTEGER;
arquivo1:textfile;
begin
AssignFile(arquivo1,'C:\Users\angela\Searches\Contar Números\entrada.txt');
reset (arquivo1);
N := 0;
repeat
INC(N);
readln(arquivo1,XI);
until XI = 1000.0;
Dec(N);
edtN.Text := IntToStr(N);
CloseFile(arquivo1);
end;
end.
```

Apêndice C

Recobrimento do Estado do Rio de Janeiro

C.1 Experimentos

Os experimentos do Estado do Rio de Janeiro foram feitos com 3539 pontos para o recobrimento com 9 centros e Retângulo Mínimo medindo 64.4722976685 x 29.9612007141.

As amplitudes das Curvas de Lissajous foram $\omega_1 = 5,7$, $\omega_2 = 5,9$.

Para cada tamanho de passo, em cada ponto de Lissajous foram feitas 500 iterações, e para o tamanho de passo com menor raio obtido, foram feitas 5000 iterações com o correspondente valor do ponto da Curva de Lissajous com o objetivo de tentar aprimorar o melhor resultado obtido. (última linha da Tabela). A tolerância usada foi $\epsilon = 0.000001$

Na coluna *Iterações* estão N^o de iterações para cada ponto de Lissajous(500) / N^o de iterações do primeiro ao último ponto de Lissajous.

Nas colunas *t Lissajous* as variações do parâmetro t de Lissajous, e *t final* o melhor parâmetro t de Lissajous para aquele conjunto de experrmentos respectivamente.

Na coluna *Av. f* está o número de avaliações da função f.
Na última coluna estão os Raios de Cobertura, isto é, a maior distância entre um centro e um ponto para aquele experimento.

Tabela C.1: Experimento a).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/4000500	0 - 500	1.0	141.1250	5117686	0.225029	85.449997	9.217155
500/4000500	0 - 500	2.0	485.5625	5265146	0,162060	78.708885	8.861107
500/4000500	0 - 500	3.0	79.8750	5372809	0.257021	74.979965	8.647758
500/4000500	0 - 500	4.0	20.3750	5466511	0.382693	71.024307	8.421507
500/4000500	0 - 500	5.0	60.0625	5537334	0.265496	70.117729	8.367028
500/4000500	0 - 500	6.0	341.6875	5601537	0.182926	71.373260	8.436223
500/4000500	0 - 500	7.0	66.6875	5662796	0.105160	68.679070	8.280188
500/4000500	0 - 500	8.0	28.1250	5708899	0.148085	70.554100	8.392853
500/4000500	0 - 500	9.0	10.5000	5742364	0.278547	67.504227	8.206961
500/4000500	0 - 500	10.0	42.9375	5777991	0.179962	67.656479	8.216632
500/4000500	0 - 500	11.0	413.9375	5804771	0.025005	68.249649	8.250538
500/4000500	0 - 500	12.0	475.6875	5829439	0.151650	69.589302	8.336347
5000		**9.0**	**10.5000**		**0.000010**	**67.126961**	**8.181658**

Tabela C.2: Experimento b).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/4000500	0 - 500	1.0	14.8750	5195747	0.287541	82.877380	9.084674
500/4000500	0 - 500	2.0	456.3750	5334426	0.024992	78.434822	8.838840
500/4000500	0 - 500	3.0	219.8750	5427931	0.070617	76.364075	8.729652
500/4000470	0 - 500	4.0	282.1875	5494343	0.090361	73.344139	8.553867
500/4000500	0 - 500	5.0	93.6875	5555867	0.096040	75.382927	8.674852
500/4000500	0 - 500	6.0	488.3125	5594652	0.256747	70.500282	8.388156
500/4000500	0 - 500	7.0	313.0625	5641016	0.124326	69.836823	8.347503
500/4000500	0 - 500	8.0	421.6250	5675636	0.148515	69.443176	8.325009
500/4000485	0 - 500	9.0	489.4375	5707497	0.095419	70.562202	8.393422
500/4000500	0 - 500	10.0	348.8750	5734163	0.159448	67.467018	8.203359
500/4000500	0 - 500	11.0	267.8750	5758332	0.011176	69.572845	8.336289
500/4000500	0 - 500	12.0	172.5000	5782169	0.290020	69.373543	8.298828
5000		**10.0**	**348.8750**		**0.000001**	**67.202239**	**8.187640**

Tabela C.3: Experimento c).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/4000500	0 - 500	1.0	92.0625	5212439	0.2888222	99.377380	9.912640
500/4000500	0 - 500	2.0	262.375	5361287	1.237994	90.019134	9.481861
500/4000500	0 - 500	3.0	47.3125	5461272	0.759259	93.519760	9.644710
500/4000500	0 - 500	4.0	49.6250	5537938	0.026890	80.44610	8.961175
500/4000500	0 - 500	5.0	183.0000	5604936	0.184312	82.496353	9.078239
500/4000500	0 - 500	6.0	218.8125	5656485	0.296522	72.262702	8.475911
500/4000500	0 - 500	7.0	470.6875	5670764	0.165286	77.718093	8.804356
500/4000500	0 - 500	8.0	49.0625	5719013	0.458476	76.870330	8.758191
500/4000500	0 - 500	9.0	125.1250	5745496	0.101398	76.035835	8.714007
500/4000500	0 - 500	10.0	442.5625	5765164	0.212533	80.344795	8.958222
500/4000500	0 - 500	11.0	327.9375	5797482	0.043255	71.668159	8.458315
500/4000500	0 - 500	12.0	330.6875	5822526	0.105460	70.929115	8.415103
5000		**9.0**	**125.1250**		**0.000004**	**75.996315**	**8.711350**

Tabela C.4: Experimento d).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/4000500	0 - 500	1.0	253.5625	5317124	0.626106	77.619034	8.804522
500/4000500	0 - 500	2.0	287.1250	5500889	0.028949	76.154037	8.720300
500/4000500	0 - 500	3.0	318.0000	5624403	0.046348	71.790115	8.467018
500/4000500	0 - 500	4.0	292.1250	5711965	0.027706	70.117928	8.367192
500/4000500	0 - 500	5.0	22.0625	5781795	0.309723	69.525330	8.328815
500/4000500	0 - 500	6.0	129.6250	5851455	0.063663	66.944748	8.174975
500/4000500	0 - 500	7.0	305.9375	5910541	0.165132	69.861069	8.351217
500/4000500	0 - 500	8.0	216.0625	5959556	0.233902	69.188393	8.312304
500/4000500	0 - 500	9.0	84.8750	6004355	0.971937	69.288971	8.317106
500/4000500	0 - 500	10.0	308.6250	6035336	0.050118	68.619469	8.278049
500/4000500	0 - 500	11.0	177.4375	6056248	0.018728	71.157852	8.427125
500/4000500	0 - 500	12.0	459.1875	6080654	0.176619	68.336479	8.256906
5000		**6.0**	**129.6250**		**0.000021**	**66.814407**	**8.166654**

Tabela C.5: Experimento e).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/4000500	0 - 500	1.0	95.3125	5411949	0.208928	82.466255	9.071528
500/4000500	0 - 500	2.0	269.,0000	5588210	0.331310	77.457443	8.769353
500/4000500	0 - 500	3.0	158.1875	5697598	0.172892	73.000847	8.531579
500/4000500	0 - 500	4.0	346.6875	5775157	0.076313	74.533180	8.627510
500/4000500	0 - 500	5.0	13.7500	5847993	0.068071	68.780090	8.272302
500/4000472	0 - 500	6.0	202.2500	5876566	0.244747	71.233017	8.434127
500/4000500	0 - 500	7.0	310.8750	5922725	0.174471	69.633018	8.334352
500/4000500	0 - 500	8.0	469.0000	5948969	0.164804	67.045937	8.180102
500/4000491	0 - 500	9.0	492.7500	5973864	0,110423	67.929047	8.236211
500/4000500	0 - 500	10.0	22.0625	5998045	0.025819	67.395096	8.201622
500/4000500	0 - 500	11.0	13.8125	6042960	0.027585	68.007225	8.239176
500/4000500	0 - 500	12.0	189.6875	6087946	0.065869	68.334656	8.259109
5000		**8.0**	**469.0000**		**0.000005**	**66.478630**	**8.147676**

Tabela C.6: Experimento f).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/4000500	0 - 500	1.0	206.6875	5402009	0.349436	83.73278	9.142711
500/4000500	0 - 500	2.0	103.0625	5603988	0.468218	87.579910	9.352131
500/4000500	0 - 500	3.0	56.2500	5701196	0.077232	80.656158	8.972989
500/4000486	0 - 500	4.0	351.6250	5750336	0.221243	79.515686	8.911177
500/4000500	0 - 500	5.0	191.8125	5820701	0.283539	75.663375	8.689021
500/4000500	0 - 500	6.0	233.1250	5876696	0.095197	72.408340	8.502264
500/4000489	0 - 500	7.0	32.0000	5905787	0.225141	72.700668	8.520638
500/4000482	0 - 500	8.0	191.8125	5941115	0.080510	69.474174	8.329722
500/4000500	0 - 500	9.0	187.3750	5966736	0.171042	70.461914	8.382308
500/4000500	0 - 500	10.0	422.7500	5987881	0.074405	69.109466	8.306407
500/4000500	0 - 500	11.0	460.7500	6005869	0.202269	72.844620	8.519018
500/4000500	0 - 500	12.0	375.8750	6018333	0.148721	71.811440	8.464600
5000		**10.0**	**422.7500**		**0.002557**	**67.814018**	**8.228839**

Tabela C.7: Experimento g).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/4000500	0 - 500	1.0	149,9375	5419127	0.269211	79.227661	8.892543
500/4000500	0 - 500	2.0	227.6250	5606422	0.278158	73.490791	8.554447
500/4000500	0 - 500	3.0	348.3750	5722696	0.159584	70.380974	8.382442
500/4000475	0 - 500	4.0	29.7500	5821972	0.202321	70.251900	8.370806
500/4000500	0 - 500	5.0	450.3125	5908360	0.011213	67.256248	8.193570
500/4000476	0 - 500	6.0	29.7500	5965162	0.104703	71,098267	8.424300
500/4000492	0 - 500	7.0	403.9375	6022023	0.254119	72.037384	8.457693
500/4000487	0 - 500	8.0	249.6250	6071816	0.018814	69.384865	8.321362
500/4000500	0 - 500	9.0	124.5625	6110144	0.131294	70.989944	8.416546
500/4000454	0 - 500	10.0	14.8750	6115917	0.038487	71.014862	8.418241
500/4000500	0 - 500	11.0	179.0625	6153825	0.400605	68.986542	8.294765
500/4000500	0 - 500	12.0	103.6875	6200922	0.013464	70.430870	8.383689
5000		**5.0**	**450.3125**		**0.000009**	**67.254532**	**8.193468**

Tabela C.8: Experimento h).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/4000500	0 - 500	1.0	456.4375	5529042	0.095323	78.346886	8.845612
500/4000500	0 - 500	2.0	40.8125	5712358	1.000863	73.856155	8.587510
500/4000500	0 - 500	3.0	168.0625	5835787	0.112958	72.994987	8.538142
500/4000500	0 - 500	4.0	375.8750	5903303	0.008421	70.252312	8.373899
500/4000499	0 - 500	5.0	250.1875	5971857	0.041601	66.666298	8.157529
500/4000473	0 - 500	6.0	477.8125	6019528	0.039412	67.448914	8.206515
500/4000463	0 - 500	7.0	1.0000	6069898	0.179741	68.242012	8.244850
500/4000466	0 - 500	8.0	181.9375	6077012	0.030409	68.468323	8.267294
500/4000442	0 - 500	9.0	60.5625	6115577	0.128507	67.846230	8.230320
500/4000415	0 - 500	10.0	312.6875	6132733	0.081287	66.952408	8.176503
500/4000500	0 - 500	11.0	270.0000	6166047	0.002193	67.712639	8.220394
500/4000500	0 - 500	12.0	10.4375	6193481	0.353258	68.129074	8.245124
5000		**10.0**	**312.6875**		**0.000002**	**66.420090**	**8.142869**

Tabela C.9: Experimento i).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/4000500	0 - 500	1.0	470.1250	5399127	0.332251	79.133972	8.883337
500/4000500	0 - 500	2.0	387.2500	5598858	3.785914	80.918907	8.987965
500/4000500	0 - 500	3.0	65.0625	5689506	0.290538	74.200752	8.607752
500/4000500	0 - 500	4.0	276.6875	5744877	0.231280	75.301918	8.668941
500/4000500	0 - 500	5.0	107.5000	5784844	0.095665	72.662254	8.506503
500/4000488	0 - 500	6.0	237.0000	5849489	0.110499	72.281654	8.446320
500/4000500	0 - 500	7.0	136.1250	5913952	0.053500	68.487038	8.267614
500/4000474	0 - 500	8.0	324.6875	5932998	0.068557	69.117393	8.307871
500/4000462	0 - 500	9.0	4.3750	5948966	0.014124	68.481972	8.264567
500/4000498	0 - 500	10.0	255.1875	5974824	0.063438	68.104530	8.246806
500/4000500	0 - 500	11.0	427.6875	5993769	0.206255	68.129303	8.222600
500/4000500	0 - 500	12.0	312.5625	6019871	0.059315	68.739876	8.281199
5000		**10.0**	**255.1875**		**0.000004**	**67.897293**	**8.232223**

Tabela C.10: Experimento j).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/4000470	0 - 500	1.0	395.8125	5529093	0.230858	78.482719	8.852431
500/4000500	0 - 500	2.0	65.5625	5729374	0.512460	73.674744	8.575996
500/4000500	0 - 500	3.0	51.1875	5845349	0.114910	68.953018	8.296005
500/4000491	0 - 500	4.0	409.5625	5921027	0.173282	71.427559	8.436560
500/4000448	0 - 500	5.0	319.1250	5989336	0.105707	68.179024	8.250334
500/4000445	0 - 500	6.0	184.0625	6025298	0.094944	68.084877	8.242244
500/4000370	0 - 500	7.0	483.3125	6074899	0.047868	67.021515	8.175034
500/4000440	0 - 500	8.0	253.5000	6091447	0.048184	68.535538	8.270839
500/4000473	0 - 500	9.0	363.7500	6132177	0.087831	67.500320	8.208481
500/4000289	0 - 500	10.0	461.9375	6161041	0.017139	66.329765	8.134830
500/4000442	0 - 500	11.0	345.6250	6173091	0.030663	65.988113	8.118289
500/4000500	0 - 500	12.0	201.1250	6152850	0.183716	66.578153	8.153041
5000		**12.0**	**201.1250**		**0.000004**	**65.127533**	**8.060891**

Centros e Raios

a)	8.1816	**b)**	8.1876	**c)**	8.4012	**d)**	8.1666	**e)**	8.1476
47.3481	-2.2240	49.2405	-1.6313	46.5791	10.6340	47.9621	-3.4585	48.5034	-2.8995
46.2887	10.9069	46.5880	9.4457	46.6371	-4.5703	46.7495	8.9981	49.2352	9.9248
32.8678	4.5993	33.7434	3.1312	36.5841	-9.3402	33.5437	3.4920	35.1132	3.4538
18.2141	2.5365	18.1452	2.8075	19.9121	2.7280	18.3686	2.4926	21.0503	5.6726
-2.1207	-2.2474	-1.7197	-1.1184	8.8672	0.4558	4.9019	0.9215	9.1556	0.5849
6.2979	0.9687	4.7620	-1.8631	-1.8341	-4.4138	-1.9616	-1.6448	-2.0444	-3.6373
10.7949	-11.9194	10.6172	-11.4707	10.6457	-11.5816	10.2194	-10.8235	11.2083	-13.1044
23.4374	-8.2722	23.6832	-8.6744	21.0819	-7.6200	22.5935	-8.1910	22.8355	-6.7842
36.4151	-7.8051	38.6608	-7.9272	34.0001	2.3425	35.8543	-9.3739	36.3596	-8.7628
f)	8.2288	**g)**	8.1934	**h)**	8.1428	**i)**	8.2322	**j)**	8.0609
46.9961	10.7456	47.8527	-2.7105	47.8740	-2.2956	47.6112	-4.6792	47.3118	-2.8295
47.1622	-2.7255	46.6727	9.7967	46.4740	10.0232	47.0835	8.1617	47.1337	9.3071
36.2302	-6.0531	33.5499	3.4975	32.9638	3.6604	32.4412	4.5653	33.2219	4.0243
20.4440	2.7562	17.8388	2.6052	18.0666	2.3755	18.3779	2.0333	19.7603	3.3392
-1.3095	-1.3445	4.5815	-1.4193	5.1151	-2.3656	5.7602	1.9170	-1.3879	-2.5312
10.4637	-11.1242	-1.7124	-3.0458	-2.1817	-3.0634	5.7602	1.9170	11.1237	0.1810
11.5266	0.9773	10.8967	-12.0540	10.5011	-12.2812	10.6721	-11.6458	10.6791	-11.8131
22.3366	-7.7542	23.1036	-7.6304	22.4740	-8.2389	24.8742	-9.3856	24.0270	-9.1160
33.7990	3.6761	36.8015	-10.6491	37.3080	-7.9497	38.1678	-4.8248	36.0515	-7.4119

Figura C.1: Raios de Cobertura e Coordenadas das Antenas do Rio de Janeiro.

C.2 Raios de Cobertura

A figura C.2 mostra os melhores resultados (Raios de Cobertura) encontrados em cada um dos conjuntos de experimentos das tabelas acima.

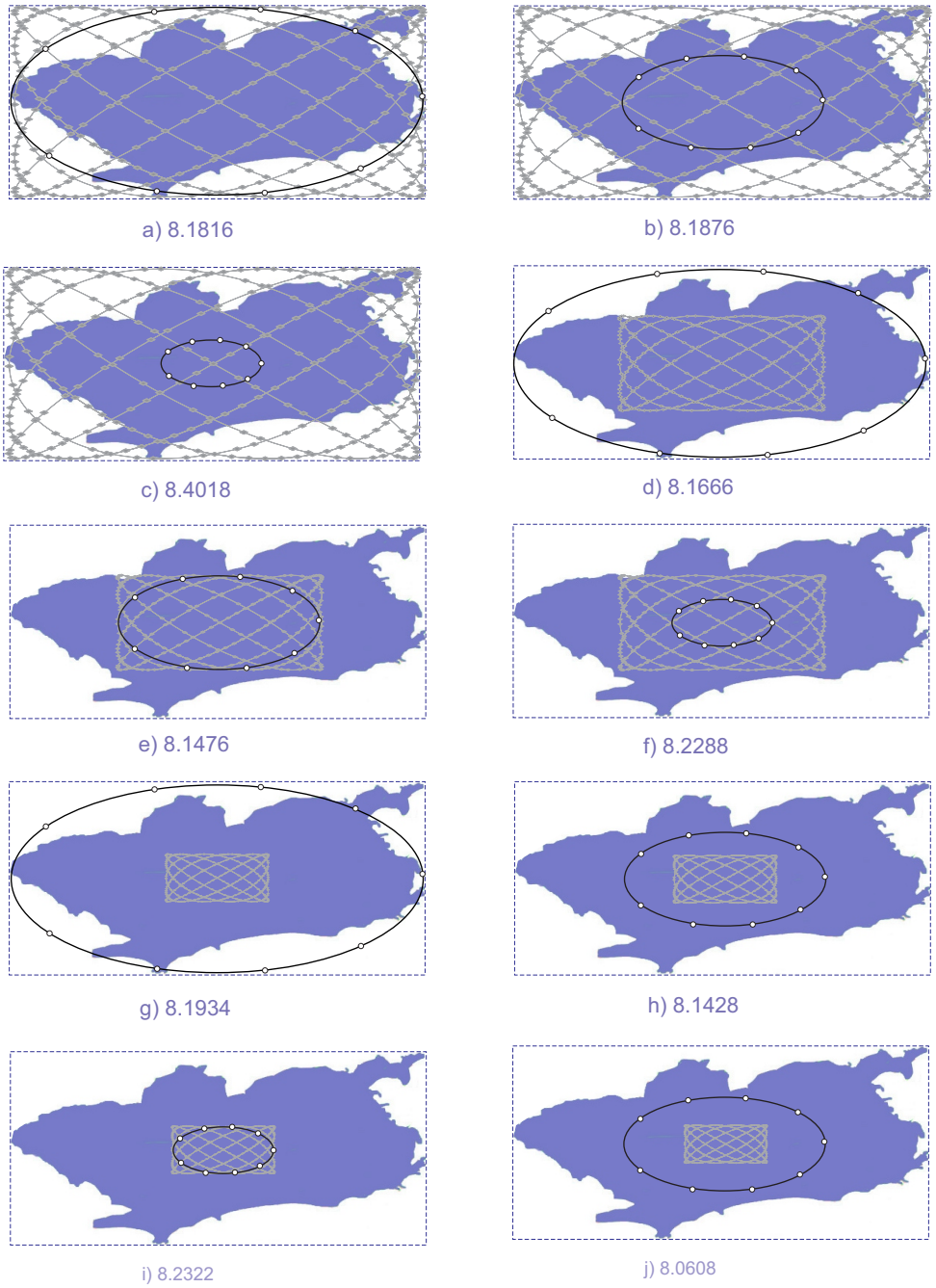

Figura C.2: Raios de Recobrimento dos experimentos do Rio de Janeiro.

Apêndice D

Recobrimento do Estado de NY

D.1 Experimentos

Os experimentos do Estado de Nova York foram feitos com 7225 pontos para o recobrimento com 7 centros e Retângulo Mínimo medindo 150.0 x 109.0.

As amplitudes das Curvas de Lissajous foram $\omega_1 = 5.89$, $\omega_2 = 5.0$.

A tolerância usada foi $\epsilon = 0.000001$

Tabela D.1: Experimento a).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	292.1875	5333885	0.040014	737.04694	27.12084
500/0000000	0-500	2.0	292.1875	5333885	0.040014	737.04694	27.12084
500/0000000	0-500	3.0	154.0625	5448042	0.119573	650.26477	25.49809
500/0000000	0-500	4.0	32.1875	5502968	0.964711	607.64539	24.64120
500/0000000	0-500	5.0	263.3750	5550672	0.035362	591.90985	24.32320
500/0000000	0-500	6.0	54.9375	5603101	2.460850	588.52820	24.16097
500/0000000	0-500	7.0	312.0000	5654540	4.899877	588.12927	24.20963
500/0000000	0-500	8.0	27.7500	5701192	0.184837	595.24835	24.32972
500/0000000	0-500	9.0	409.3750	5751461	0.147698	565.35205	23.52460
500/0000000	0-500	10.0	152.5625	5806459	0.142059	585.83270	24.18722
500/0000000	0-500	11.0	442.6250	5872486	0.082652	554.66400	23.52495
500/0000000	0-500	12.0	284.3750	5899073	0.238045	571.48248	23.89828
5000		**9.0**	**409.3750**		**0.000072**	**563.61542**	**23.48280**

Tabela D.2: Experimento b).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	172.8125	5384100	0.594375	729.89685	27.01016
500/0000000	0-500	2.0	325.3750	5453818	4.045401	672.20959	25.92311
500/0000000	0-500	3.0	349.3750	5511201	0.092544	671.48456	25.70962
500/0000000	0-500	4.0	441.1250	5572799	0.107371	601.81952	24.52361
500/0000000	0-500	5.0	415.8750	5621920	1.080316	622.74353	24.94826
500/0000000	0-500	6.0	0.0000	5703105	0.059922	590.55359	24.29508
500/0000000	0-500	7.0	146.0625	5762317	0.228965	607.39923	24.59933
500/0000000	0-500	8.0	291.9375	5796620	4.650261	606.31256	24.60407
500/0000000	0-500	9.0	38.9375	5846985	0.042065	581.82153	24.11741
500/0000000	0-500	10.0	130.8125	5891650	0.183937	580.58282	23.97034
500/0000000	0-500	11.0	260.8125	6011894	0.000097	577.91766	23.95317
500/0000000	0-500	12.0	462.4375	6055853	0.073937	601.37793	24.51792
5000		**11.0**	**260.8125**		**0.000097**	**577.91766**	**23.95317**

Tabela D.3: Experimento c).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	24.5625	5380455	2.037888	656.62189	25.61635
500/0000000	0-500	2.0	388.8750	5451972	0.437060	620.49268	24.89915
500/0000000	0-500	3.0	409.0625	5516315	0.057528	609.25500	24.67375
500/0000000	0-500	4.0	427.2500	5563839	0.045278	596.06567	24.37262
500/0000000	0-500	5.0	130.6875	5599344	0.073262	591.44952	24.31349
500/0000000	0-500	6.0	481.4375	5660127	0.215641	597.35156	24.43454
500/0000000	0-500	7.0	18.1875	5711248	0.094940	596.57050	24.28451
500/0000000	0-500	8.0	184.7500	5733514	0.436309	580.49811	24.03419
500/0000000	0-500	9.0	497.1250	5782290	0.012980	568.22272	23.83239
500/0000000	0-500	10.0	20.1875	5839210	0.140355	576.59851	23.87490
500/0000000	0-500	11.0	6.9375	5885237	0.000162	567.43695	23.74705
500/0000000	0-500	12.0	304.0000	5921224	0.090340	572.88477	23.92476
5000		**11.0**	**6.9375**		**0.000162**	**567.43695**	**23.74705**

Tabela D.4: Experimento d).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	300.3750	5420783	7.401075	672.21411	25.91537
500/0000000	0-500	2.0	499.2500	5498092	1.000041	620.92841	24.91229
500/0000000	0-500	3.0	93.8750	5576135	0.098441	611.99774	24.73287
500/0000000	0-500	4.0	439.9375	5655807	0.844247	610.59979	24.70457
500/0000000	0-500	5.0	254.3750	5732201	0.274661	596.14777	24.41059
500/0000000	0-500	6.0	423.8750	5808406	0.086323	594.32043	24.37160
500/0000000	0-500	7.0	325.9375	5870394	0.023946	568.96790	23.84544
500/0000000	0-500	8.0	311.5000	5969043	0.160517	578.15759	23.59386
500/0000000	0-500	9.0	51.8750	6029160	0.209821	566.50348	23.72136
500/0000000	0-500	10.0	67.5000	6089229	0.106959	584.38922	24.16891
500/0000000	0-500	11.0	12.8125	6226708	0.008138	568.36456	23.73462
500/0000000	0-500	12.0	365.9375	6297547	0.957540	567.14032	23.81114
5000		**8.0**	**311.500**		**0.000216**	**576.72784**	**23.55889**

Tabela D.5: Experimento e).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	41.0000	5497660	0.204418	694.43469	26.34678
500/0000000	0-500	2.0	227.6250	5630064	0.240979	579.94354	24.07586
500/0000000	0-500	3.0	266.1250	5725130	0.172751	620.66315	24.90090
500/0000000	0-500	4.0	463.1250	5901648	0.107892	629.04791	25.07659
500/0000000	0-500	5.0	458.6875	6060152	0.000061	597.93848	24.44736
500/0000000	0-500	6.0	0.0000	6218965	0.059922	590.55359	24.29508
500/0000000	0-500	7.0	59.3125	6390591	0.004679	585.46948	24.18064
500/0000000	0-500	8.0	463.1250	6547792	0.377114	603.93329	24.56647
500/0000000	0-500	9.0	384.5625	6684917	0.248459	580.41150	24.08462
500/0000000	0-500	10.0	130.1875	6827586	0.055522	558.30237	23.62347
500/0000000	0-500	11.0	467.8125	7028437	0.032835	570.40045	23.87733
500/0000000	0-500	12.0	211.7500	7070393	0.002340	557.56067	23.60406
5000		**12.0**	**211.7500**		**0.000225**	**557.56067**	**23.60406**

Tabela D.6: Experimento f).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	162.1250	5499047	0.286860	647.75677	25.39854
500/0000000	0-500	2.0	34.6875	5619198	0.095893	641.49731	25.13826
500/0000000	0-500	3.0	31,5000	5715535	0.034094	587.27484	24.22846
500/0000000	0-500	4.0	293.3750	5831994	0.222386	586.44623	24.21077
500/0000000	0-500	5.0	128.0625	5974510	0.097795	570.82928	23.60876
500/0000000	0-500	6.0	321.1250	6044227	0.452665	583.89520	24.14613
500/0000000	0-500	7.0	218.6875	6156537	0.075097	579.33215	24.06261
500/0000000	0-500	8.0	409.6250	6203667	0.069715	577.01117	23.97793
500/0000000	0-500	9.0	490.6875	6308508	0.000129	577.66754	24.02956
500/0000000	0-500	10.0	423.5000	6358721	0.090157	546.20825	23.35503
500/0000000	0-500	11.0	460.8125	6465336	0.100047	560.61151	23.66578
500/0000000	0-500	12.0	388.3125	6489492	0.000116	567.84784	23.82479
5000		**10.0**	**423.5000**		**0.000136**	**546.12878**	**23.35439**

Tabela D.7: Experimento g).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	234.8750	5473340	10.47343	650.48706	25.30921
500/0000000	0-500	2.0	313.9375	5560990	0.240222	594.17389	24.29750
500/0000000	0-500	3.0	135.8750	5640673	0.124500	607.83899	24.61731
500/0000000	0-500	4.0	258.3750	5714175	0.186885	601.20544	24.51440
500/0000000	0-500	5.0	218.9375	5765015	0.164605	602.27594	24.53197
500/0000000	0-500	6.0	119.6875	5855353	0.173459	578.39600	24.04475
500/0000000	0-500	7.0	35.5000	5930808	2.538809	575.17340	23.97794
500/0000000	0-500	8.0	38.1250	5990944	0.246591	568.21490	23.79089
500/0000000	0-500	9.0	197.6250	6057207	0.165368	573.69849	23.94874
500/0000000	0-500	10.0	366.1875	6154045	0.010709	579.04309	24.05530
500/0000000	0-500	11.0	403.5625	6207392	0.562646	568.85358	23.84220
500/0000000	0-500	12.0	395.0625	6286716	0.030108	553.30914	23.51714
5000		**12.0**	**395.0625**		**0.000070**	**553.19904**	**23.51465**

Tabela D.8: Experimento h).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	233.0625	5589820	0.293535	651.98676	25.38998
500/0000000	0-500	2.0	104.0000	5823016	0.539539	624.18170	24.97529
500/0000000	0-500	3.0	116.3750	6143502	0.048486	574.79071	23.96884
500/0000000	0-500	4.0	98.6250	6426057	0.120225	591.25494	24.14930
500/0000000	0-500	5.0	406.4375	6725949	0.016269	583.90369	24.15981
500/0000000	0-500	6.0	273.6250	6985495	0.000031	562.04980	23.62345
500/0000000	0-500	7.0	264.1250	7322409	0.014011	578.78265	24.05346
500/0000000	0-500	8.0	483.1250	7567798	0.014548	566.00079	23.76043
500/0000000	0-500	9.0	42.1250	7783915	0.000122	578.38776	24.04470
500/0000000	0-500	10.0	253.8125	7941943	0.024727	583.58844	24.06618
500/0000000	0-500	11.0	251.3125	7945414	0.111264	569.19781	23.83843
500/0000000	0-500	12.0	45.8750	8211233	0.064497	573.92834	23.95024
5000		**6.0**	**272.6250**		**0.000031**	**562.04980**	**23.62345**

Tabela D.9: Experimento i).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	185.5625	5497878	0.143713	583.22736	24.14406
500/0000000	0-500	2.0	288.8750	5646766	0.169521	568.54114	23.80298
500/0000000	0-500	3.0	221.8750	5811125	0.172096	583.24860	24.14611
500/0000000	0-500	4.0	10.6250	6016066	0.074019	581.91486	24.10281
500/0000000	0-500	5.0	479.8750	6146819	0.191291	570.58734	23.87938
500/0000000	0-500	6.0	272.9375	6277669	0.048168	560.90430	23.67636
500/0000000	0-500	7.0	128.0000	6522857	0.038504	561.74939	23.69508
500/0000000	0-500	8.0	25.1875	6648207	0.086990	569.59473	23.85933
500/0000000	0-500	9.0	90.6250	6633859	0.070664	553.70667	23.52352
500/0000000	0-500	10.0	197.8750	6692382	0.071629	541.44751	23.14551
500/0000000	0-500	11.0	327.3750	6735616	0.029499	546.14362	23.33253
500/0000000	0-500	12.0	372.8125	6878697	0.008011	554.03387	23.53343
5000		**10.0**	**197.8750**		**0.000150**	**536.87451**	**23.07300**

Tabela D.10: Experimento j).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	411.9375	5609539	1.731644	673.38232	25.42215
500/0000000	0-500	2.0	410.1875	5869267	1.537712	606.73834	24.60538
500/0000000	0-500	3.0	23.0000	6131941	0.126629	597.01538	24.42682
500/0000000	0-500	4.0	23.9375	6499064	0.026044	591.12964	24.30836
500/0000000	0-500	5.0	273.0000	6861805	0.868623	594.30450	24.32890
500/0000000	0-500	6.0	123.7500	7253811	0.029012	590.54321	24.29379
500/0000000	0-500	7.0	466.1875	7481755	1.786382	564.14148	23.74472
500/0000000	0-500	8.0	80.6875	7831176	6.231113	582.06366	24.10654
500/0000000	0-500	9.0	136.7500	7944763	0.110360	569.13098	23.84879
500/0000000	0-500	10.0	2.8125	7873301	0.052450	580.25421	23.98215
500/0000000	0-500	11.0	324.1875	7796039	0.044215	574.37750	23.90403
500/0000000	0-500	12.0	370.6875	7817328	0.027020	558.93634	23.63228
5000		**12.0**	**370.6875**		**0.000178**	**558.41132**	**23.62119**

Centros e Raios

a)	23.4828	b)	23.9531	c)	23.7470	d)	23.5588	e)	23.6040
618.9882	223.1992	618.1865	220.8006	617.9221	220.4020	617.8131	223.7831	619.8072	216.3469
577.7402	297.1433	584.0294	290.5105	595.6982	260.34063	594.8356	264.1414	600.8355	267.6318
591.8554	261.8966	561.1331	286.3507	576.0814	295.7907	574.2817	298.2635	574.9662	295.6796
539.3942	245.7375	541.6650	249.1181	567.1602	278.1309	556.2434	266.3609	554.9599	266.9629
509.3005	237.4623	511.2610	235.2899	508.2597	238.6724	507.0084	238.9082	508.4486	238.3926
580.3906	233.1003	585.1723	254.5420	540.9469	244.7694	537.2589	244.9263	536.9677	239.6993
563.8886	273.8602	574.4884	229.3208	580.3931	237.7191	580.0821	228.5196	582.6997	236.3507
f)	23.3543	**g)**	23.5146	**h)**	23.6234	**i)**	23.0636	**j)**	23.6211
618.9028	219.8267	619.9624	220.1526	620.1485	216.9275	618.4550	221.0918	621.2639	220.9331
597.8198	259.6506	575.7753	295.7521	574.0584	296.9215	599.3300	260.6580	598.7662	268.8234
577.5686	295.6116	563.9359	271.2005	587.4515	265.9892	577.6037	296.3608	572.8170	302.2392
555.8066	272.4603	508.4113	238.2969	542.8326	266.6548	558.8867	272.9169	556.2858	267.1720
508.4109	238.119	540.8511	243.9135	510.4823	236.3605	509.2875	236.9825	507.6444	238.9516
583.1173	237.4476	590.2533	266.7648	553.8006	235.9093	539.4562	245.8764	538.3823	241.0356
540.4398	244.2557	584.4500	237.2266	590.3731	235.9992	580.1035	234.9933	583.9906	234.7463

Figura D.1: Raios de Cobertura e Coordenadas das Antenas de NY.

D.2 Raios de Cobertura

A figura D.2 mostra os melhores resultados (Raios de Cobertura) encontrados em cada um dos conjuntos de experimentos das tabelas acima.

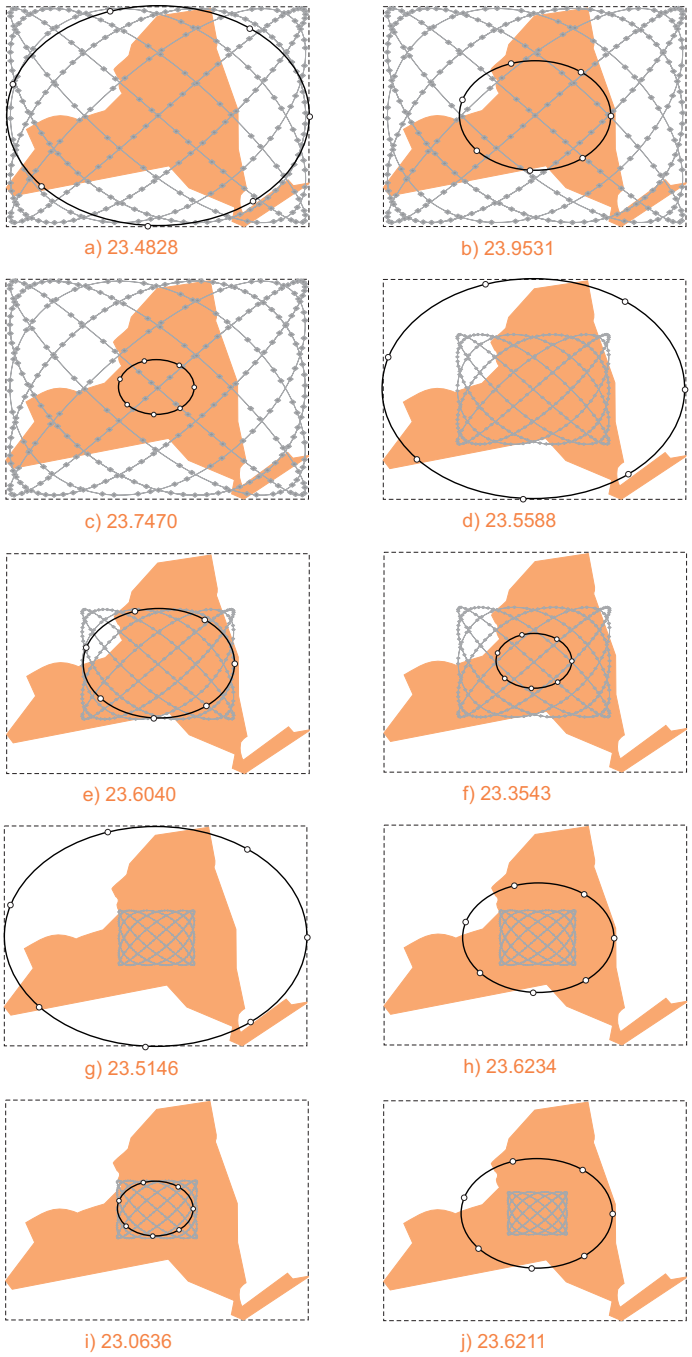

Figura D.2: Raios de Recobrimento dos experimentos de NY.

Apêndice E

Recobrimento do Estado do Pará

E.1 Experimentos

Os experimentos do Estado do Pará foram feitos com 3416 pontos para o recobrimento com 5 centros e Retângulo Mínimo medindo 84.0999984741 x 80.8000030518.

As amplitudes das Curvas de Lissajous foram $\omega_1 = 5., 7$, $\omega_2 = 5.9$.

A tolerância usada foi $\epsilon = 0.000001$

Tabela E.1: Experimento a).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/3999939	0-500	1.0	93.7500	5935500	0.019829	457.47357	21.38151
500/3999028	0-500	2.0	212.9375	6188715	0.024816	450.44479	21.21188
500/3997926	0-500	3.0	362.1250	6473750	0.026037	429.35104	20.70920
500/3997060	0-500	4.0	195.1875	6730596	0.000135	436.08984	20.87410
500/3995927	0-500	5.0	66.3125	7116968	0.005485	429.37128	20.71004
500/3995061	0-500	6.0	269.5625	7274839	0.000017	425.33725	20.61527
500/3992869	0-500	7.0	61.,3125	7498202	0.089062	423.55219	20.56757
500/3992531	0-500	8.0	406.4375	7648420	0.001463	421.55478	20.52126
500/3991543	0-500	9.0	330.1875	7857946	0.009514	422.32928	20.54479
500/3991965	0-500	10.0	311.5000	8084275	0.004301	424.97018	20.60590
500/3992178	0-500	11.0	119.0000	8225045	0.000263	424.60889	20.59517
500/3990745	0-500	12.0	299.5625	8442539	0.000008	422.16824	20.53681
5000		**8.0**	**406.4375**		**0.000006**	**421.48419**	**20.48335**

Tabela E.2: Experimento b).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/3998168	0-500	1.0	107.5000	6237619	0.000008	451.18936	21.23041
500/3996879	0-500	2.0	72.3125	6513471	0.112520	435.05136	20.84832
500/3996678	0-500	3.0	210.0000	6825249	0.017465	436.32843	20.88220
500/3995788	0-500	4.0	62.4375	7071202	0.019963	433.34744	20.80790
500/3994534	0-500	5.0	29.4375	7285655	0.001422	433.11392	20.80365
500/3993859	0-500	6.0	70.1250	7640313	0.004344	427.27698	20.66133
500/3993141	0-500	7.0	138.8750	7777166	0.029242	429.56891	20.71885
500/3991388	0-500	8.0	385.3750	7952725	0.037525	420.43918	20.49562
500/3991360	0-500	9.0	26.2500	8162305	0.280176	421.28757	20.51573
500/3989509	0-500	10.0	251.1875	8299504	0.000641	421.34787	20.51664
500/3989518	0-500	11.0	314.0000	8432433	0.000015	429.95874	20.73073
500/3990321	0-500	12.0	202.8125	8504306	0.001782	422.38184	20.54042
5000		**8.0**	**385.3750**		**0.000009**	**420.43219**	**20.49537**

Tabela E.3: Experimento c).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/3996885	0-500	1.0	472.8750	6530419	0.009082	438.93402	20.94198
500/3994176	0-500	2.0	346.1250	6849254	0.000004	442.05164	21.01840
500/3992898	0-500	3.0	203.9375	7188657	0.000009	441.69937	21.01291
500/3993232	0-500	4.0	377.0000	7481625	0.000008	422.17746	20.53935
500/3991506	0-500	5.0	111.3125	7682589	0.000536	424.31351	20.58846
500/3992459	0-500	6.0	139.5000	7912895	0.012048	429.41754	20.71606
500/3989654	0-500	7.0	218.2500	8020429	0.009816	430.17996	20.73152
500/3992044	0-500	8.0	143.2500	8314298	0.000000	420.01361	20.48594
500/3990627	0-500	9.0	342.7500	8499160	0.000893	429.20053	20.70681
500/3989508	0-500	10.0	385.5000	8607377	0.001214	422.67911	20.54911
500/3989615	0-500	11.0	361.5625	8664789	0.009865	423.41861	20.56605
500/3990857	0-500	12.0	406.6875	8680380	0.000057	428.74548	20.69888
5000		**8.0**	**143.2500**		**0.000000**	**420.01361**	**20.48594**

Tabela E.4: Experimento d).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/3998945	0-500	1.0	24.2500	6175879	0.002137	459.60583	21.38631
500/3997158	0-500	2.0	276.5625	6603030	0.000218	430.18121	20.72993
500/3996367	0-500	3.0	128.1250	7048040	0.002356	431.59155	20.76810
500/3994584	0-500	4.0	432.7500	7312509	0.002571	431.06848	20.75612
500/3992026	0-500	5.0	480.5625	7617193	0.002337	431.77310	20.76954
500/3992925	0-500	6.0	352.7500	7971648	0.000010	419.09671	20.46510
500/3991248	0-500	7.0	386.8750	8213484	0.008468	425.08182	20.60966
500/3991017	0-500	8.0	364.6250	8382620	0.039765	422.24634	20.53680
500/3987851	0-500	9.0	415.1875	8591078	0.002306	429.52023	20.71813
500/3989456	0-500	10.0	391.0625	8773810	0.043546	420.86237	20.50997
500/3989297	0-500	11.0	204.8125	9011251	0.001425	420.13202	20.48871
500/3985963	0-500	12.0	22.8125	9165792	0.003570	421.54752	20.51896
5000		**6.0**	**352.7500**		**0.000010**	**419.09671**	**20.46510**

Tabela E.5: Experimento e).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/3995049	0-500	1.0	343.8750	7024641	0.161263	439.32495	20.95150
500/3992793	0-500	2.0	298.2500	7784500	0.105825	434.96643	20.85217
500/3990026	0-500	3.0	409.5000	8176834	0.003262	434.85287	20.84641
500/3987332	0-500	4.0	291.0000	8588706	0.002245	433.18201	20.80752
500/3988260	0-500	5.0	495.5000	8785334	0.022620	428.60419	20.68863
500/3987616	0-500	6.0	18.2500	8906339	0.000008	425.61963	20.61708
500/3988330	0-500	7.0	479.5000	9249595	0.062063	422.26044	20.53688
500/3988248	0-500	8.0	277.4375	9399513	0.187556	423.86017	20.58009
500/3984974	0-500	9.0	377,8125	9447480	0.001010	421.28726	20.51435
500/3985120	0-500	10.0	130.6875	9682889	0.000004	422,82294	20.55514
500/3985102	0-500	11.0	286.1875	9811795	0.011542	420.84720	20.50566
500/3985002	0-500	12.0	314.7500	9941632	0.004544	421.39651	20.51306
5000		**11.0**	**286.1875**		**0.000003**	**420.8394**	**20.50547**

Tabela E.6: Experimento f).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/3988381	0-500	1.0	441.9375	8223176	0.003091	445.90503	21.10501
500/3984261	0-500	2.0	224.6875	8853622	0.005544	433.19547	20.80441
500/3983473	0-500	3.0	95.8125	9351827	0.007131	430.02405	20.72704
500/3985180	0-500	4.0	283.8750	9671675	0.000019	422.70096	20.55400
500/3982394	0-500	5.0	370.9375	9913885	0.051435	422.71762	20.55007
500/3982982	0-500	6.0	125.5625	10243147	0.002520	421.58551	20.52280
500/3980785	0-500	7.0	273.2500	10280594	0.020924	419.90201	20.48433
500/3981815	0-500	8.0	242.2500	10308085	0.000572	420.78183	20.50317
500/3979206	0-500	9.0	104.3750	10285975	0.013022	425.63892	20.62135
500/3982839	0-500	10.0	56.0000	10327922	0.241734	421.57974	20.52234
500/3980540	0-500	11.0	185.8750	10139011	0.002376	423.57306	20.57126
500/3981331	0-500	12.0	8.5000	10209438	0.002623	419.30997	20.46512
5000		**7.0**	**273.2500**		**0.000009**	**419.89685**	**20.48434**

Tabela E.7: Experimento g).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/3997013	0-500	1.0	308.1250	6426761	0.044714	434.72922	20.84076
500/3994628	0-500	2.0	480.4375	6912242	0.192602	442.82770	21.03854
500/3993984	0-500	3.0	397.6250	7316162	0.000016	429.88995	20.72992
500/3992525	0-500	4.0	235,5625	7685049	0.005867	433.22684	20.80752
500/3989034	0-500	5.0	365.5625	8100160	0.003431	428.83133	20.69889
500/3987501	0-500	6.0	76.7500	8392741	0.000009	430.23154	20.73370
500/3988212	0-500	7.0	300.7500	8599122	0.000442	420.39621	20.49535
500/3987573	0-500	8.0	181.6875	8699208	0.018070	432.40750	20.78389
500/3987789	0-500	9.0	223.4375	8799657	0.001471	426.75070	20.64494
500/3986408	0-500	10.0	10.8125	9024419	0.000015	423.51645	20.57125
500/3987594	0-500	11.0	295.3125	9227511	0.049099	425.62610	20.62356
500/3985582	0-500	12.0	355.9375	9212526	0.040675	429.72339	20.72060
5000		**7.0**	**300.7500**		**0.000017**	**420.39612**	**20.49535**

Tabela E.8: Experimento h).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/3990861	0-500	1.0	458.2500	8011120	0.002251	426.29587	20.63733
500/3983167	0-500	2.0	459.3125	9254821	0.173308	434.09558	20.82548
500/3984513	0-500	3.0	262.9375	9821975	0.000546	434.95047	20.84533
500/3984899	0-500	4.0	27.1250	10259065	1.030012	423.53595	20.56864
500/3983192	0-500	5.0	264.3750	10324509	0.000016	426.20630	20.63723
500/3982833	0-500	6.0	484.8750	10450328	0.000498	425.76541	20.62319
500/3982115	0-500	7.0	106.8125	10649307	0.000901	422.66776	20.55478
500/3983367	0-500	8.0	83.3125	10579728	0.024415	423.95834	20.58068
500/3982417	0-500	9.0	135.5000	10591295	0.001122	421.97012	20.53260
500/3980853	0-500	10.0	395.3750	10604154	0.008618	426.68729	20.64549
500/3980765	0-500	11.0	51.4375	10579375	0.000016	426.65347	20.64907
500/3979275	0-500	12.0	357.9375	10883819	0.016317	426.56189	20.64445
5000		**9.0**	**135.5000**		**0.000000**	**421.80359**	**20.53260**

Tabela E.9: Experimento i).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/3985630	0-500	1.0	245.5000	8156624	0.037347	436.29852	20.87715
500/3985831	0-500	2.0	275.2500	8746927	0.000907	423.18707	20.55985
500/3986857	0-500	3.0	69.1875	9223085	0.009302	419.24203	20.46542
500/3983486	0-500	4.0	261.9375	9433230	0.001240	419.09695	20.46511
500/3984458	0-500	5.0	98.6875	9756698	0.009668	421.16180	20.51442
500/3983145	0-500	6.0	402.1250	9892247	0.009974	420.73535	20.50286
500/3983676	0-500	7.0	372.3750	10250647	0.000004	419.96661	20.48510
500/3983978	0-500	8.0	105.3125	10074733	0.002362	420.16269	20.48960
500/3985368	0-500	9.0	339.5000	10092824	0.000018	419.22385	20.46511
500/3982192	0-500	10.0	201.1875	10013884	0.040518	419.51154	20.46812
500/3986338	0-500	11.0	251.9375	9910900	0.026599	419.30133	20.46610
500/3983668	0-500	12.0	377.5625	9862871	0.001896	420.35635	20.49476
5000		**4.0**	**261.9375**		**0.000007**	**419.09668**	**20.46510**

Tabela E.10: Experimento j).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/3991288	0-500	1.0	233.1250	8384342	0.000019	438.71509	20.93751
500/3987786	0-500	2.0	395.5000	9564509	0.071429	428.19196	20.67836
500/3983884	0-500	3.0	12.6875	10015261	0.031789	436.60104	20.88638
500/3981782	0-500	4.0	362.3125	10513853	0.128507	426.88544	20.65365
500/3985373	0-500	5.0	171.7500	10646629	0.056733	423.80722	20.57678
500/3979826	0-500	6.0	74.7500	10555363	0.101308	429.99704	20.73178
500/3983323	0-500	7.0	75.8125	10885678	0.002925	432.07309	20.77644
500/3979442	0-500	8.0	260.0625	10601705	0.027828	420.83649	20.50277
500/3983180	0-500	9.0	319.5625	10798866	0.000005	425.90701	20.62879
500/3982275	0-500	10.0	168.5000	10845825	0.053614	422.59705	20.54402
500/3980347	0-500	11.0	138.7500	10972450	0.008224	422.63278	20.55284
500/3980326	0-500	12.0	265.2500	11250703	0.000005	429.41824	20.71595
5000		**8.0**	**260.0625**		**0.000003**	**420.82587**	**20.50251**

Centros e Raios

a)	20.5194	b)	20.4953	c)	20.4859	d)	20.4651	e)	20.5054
69.1297	51.0301	69.1589	50.9417	68.0311	50.4459	68.4609	50.8818	68.4942	50.6108
21.8068	74.4825	21.2818	69.2627	31.9788	51.4066	31.0642	51.8719	31.4973	51.6889
31.9639	51.3790	31.9651	51.3266	21.8240	73.5952	21.8359	72.1892	20.6722	74.1818
24.2600	22.7874	25.8235	23.8120	25.0446	23.3015	24.9366	23.1927	24.4816	22.9325
52.0794	20.8901	52.2029	20.7090	52.3132	20.4741	52.2599	20.6316	52.1242	20.8259
f)	20.4834	**g)**	20.4953	**h)**	20.53260	**i)**	20.4651	**j)**	20.5025
69.3313	51.0050	67.8229	50.8607	69.0206	50.9746	68.0661	50.8578	68.0959	50.4930
31.9337	51.3760	31.4846	52.0878	32.2601	50.9650	31.0556	51.8647	31.7182	51.9128
21.3320	69.1264	21.9191	71.9116	21.4575	70.2236	21.5081	71.4785	21.7789	74.9546
25.2904	23.4626	25.8229	23.8117	26.0781	23.9583	26.2403	23.9990	24.5869	22.9987
52.3068	20.5505	52.1575	20.7783	52.6539	19.7303	52.2599	20.6316	52.8139	19.5416

Figura E.1: Raios de Cobertura e Coordenadas das Antenas do Pará.

E.2 Raios de Cobertura

A figura E.2 mostra os melhores resultados (Raios de Cobertura) encontrados em cada um dos conjuntos de experimentos das tabelas acima.

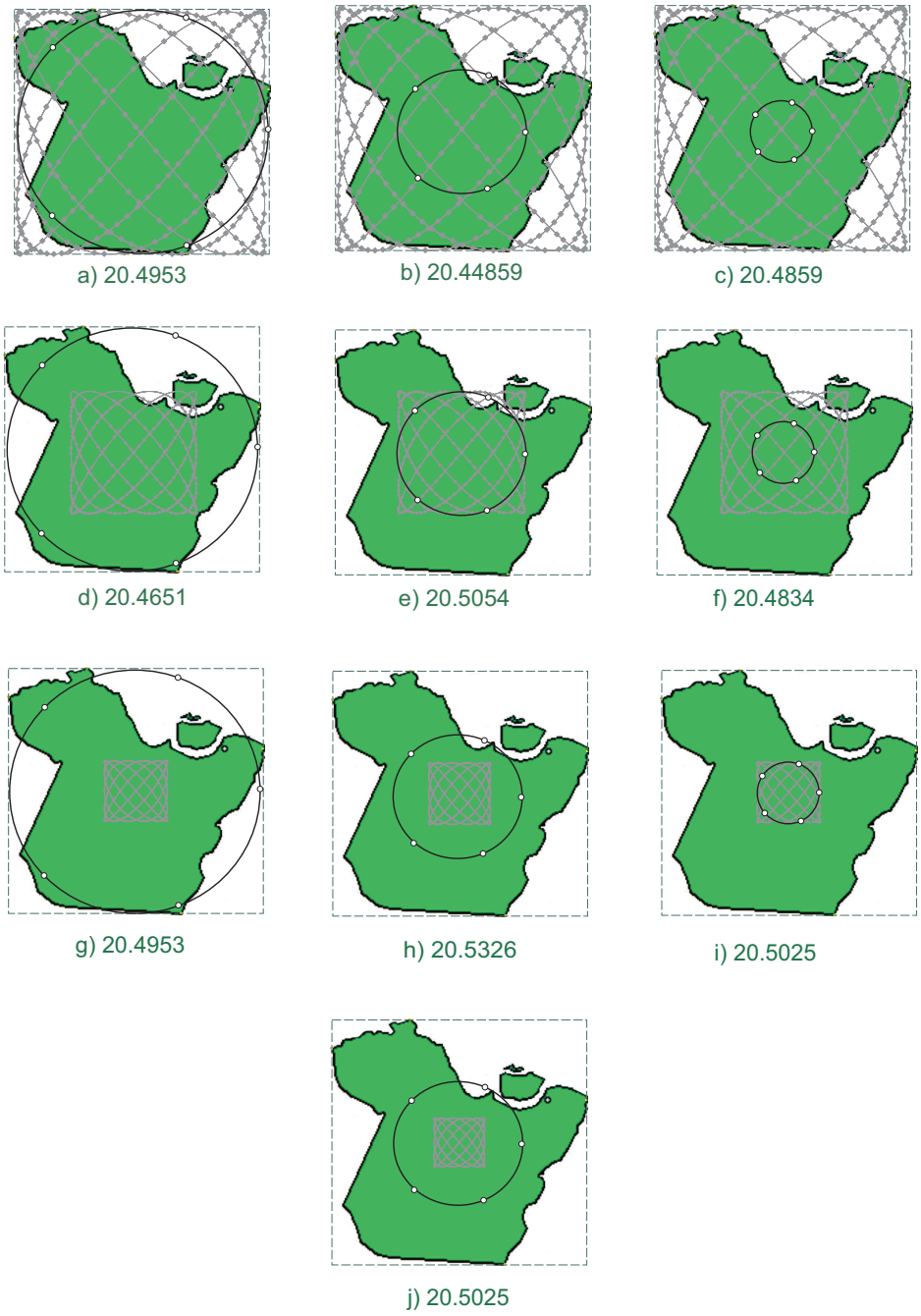

Figura E.2: Raios de Recobrimento dos experimentos do Pará.

Apêndice F

Recobrimento do Brasil

F.1 Experimentos

Os experimentos do Brasil foram feitos com 17255 pontos para o recobrimento com 10 centros e Retângulo Mínimo medindo 18.0 x 18.0.

As amplitudes das Curvas de Lissajous foram $\omega_1 = 5.89$, $\omega_2 = 5.0$.

A tolerância usada foi $\epsilon = 0.000001$

Tabela F.1: Experimento a).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	466.7500	5079225	0.094173	10.594689	3.242100
500/0000000	0-500	2.0	60.3125	5297716	0.063978	9.690176	3.111204
500/0000000	0-500	3.0	459.3125	5458590	0.128106	9.193336	3.031614
500/0000000	0-500	4.0	405.9375	5584093	0.049466	8.770828	2.960940
500/0000000	0-500	5.0	324.2500	5678301	0.015863	8.748992	2.957358
500/0000000	0-500	6.0	229.3125	5769794	0.014137	8.724978	2.953198
500/0000000	0-500	7.0	346.1875	5828260	0.040776	8.611546	2.933961
500/0000000	0-500	8.0	324.8750	5873918	0.060691	8.921191	2.986162
500/0000000	0-500	9.0	420.9375	5926971	0.035395	8.886561	2.980249
500/0000000	0-500	10.0	140.7500	5983120	0.019295	8.911998	2.984907
500/0000000	0-500	11.0	108.7500	6015989	0.008552	8.931631	2.987881
500/0000000	0-500	12.0	107.5625	6035945	0.010778	8.757662	2.957243
5000		**7.0**	**346.1875**		**0.000004**	**8.469968**	**2.910321**

Tabela F.2: Experimento b).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/4000500	0-500	1.0	88.0000	5209685	0.033203	9.253683	3.038524
500/4000500	0-500	2.0	176.5625	5403426	0.057091	8.993527	2.998446
500/4000500	0-500	3.0	35.8125	5545413	0.056275	8.690205	2.947203
500/4000500	0-500	4.0	416.0000	5657698	0.059280	8.922259	2.986590
500/4000500	0-500	5.0	17.6250	5740367	0.003051	8.916010	2.985346
500/4000500	0-500	6.0	339.8750	5795527	0.065894	9.075962	3.012149
500/4000500	0-500	7.0	170.2500	5853567	0.182254	8.808909	2.967534
500/4000500	0-500	8.0	493.7500	5903324	0.126094	8.934935	2.988349
500/4000500	0-500	9.0	229.3125	5964606	0.178172	8.944900	2.990058
500/4000500	0-500	10.0	405.9375	6010122	0.044216	8.834673	2.958053
500/4000500	0-500	11.0	155.5000	6026478	0.033961	8.963293	2.993368
500/4000500	0-500	12.0	116.8750	6031797	0.074365	8.720196	2.952569
5000		**3.0**	**35.8125**		**0.000004**	**8.565260**	**2.926646**

Tabela F.3: Experimento c).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	24.5000	5256370	0.615054	10.464809	3.23396
500/0000000	0-500	2.0	35.7500	5440363	0.073473	9.299410	3.048841
500/0000000	0-500	3.0	127.5000	5538022	0.037924	9.227595	3.037220
500/0000000	0-500	4.0	32.0000	5611169	0,041886	9.230752	3.033414
500/0000000	0-500	5.0	57.1250	5703190	0.041325	9.152866	3.004672
500/0000000	0-500	6.0	109.3750	5758672	0.118909	9.223331	3.031638
500/0000000	0-500	7.0	148.2500	5792434	0.027536	9.288019	3.047234
500/0000000	0-500	8.0	62.8750	5841280	0.078399	8.995636	2.998669
500/0000000	0-500	9.0	347.2500	5857486	0.926504	9.179484	3.029241
500/0000000	0-500	10.0	134.4375	5910335	0.011914	9.368484	3.060213
500/0000000	0-500	11.0	311.0000	5932807	0.079591	9.112167	3.018040
500/0000000	0-500	12.0	39.0625	5959693	0.061819	9.420957	3.068789
5000		**8.0**	**62.8750**		**0.000004**	**8.931348**	**2.988009**

Tabela F.4: Experimento d).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	28.0000	5230579	0.725125	9.838962	3.131353
500/0000000	0-500	2.0	43.2500	5504249	0.502752	9.086272	3.013552
500/0000000	0-500	3.0	214.0625	5691478	0.031703	9.089653	3.014374
500/0000000	0-500	4.0	373.3125	5823290	0.011962	8.655253	2.941376
500/0000000	0-500	5.0	105.5625	5899979	0.017985	8.723151	2.953076
500/0000000	0-500	6.0	370.7500	5959838	0.031987	8.804530	2.966812
500/0000000	0-500	7.0	102.3125	6012486	0.027185	8.718700	2.952288
500/0000000	0-500	8.0	175.8750	6026819	0.151876	8.721094	2.952727
500/0000000	0-500	9.0	40.1875	6061856	0.041550	8.636157	2.938114
500/0000000	0-500	10.0	99.8750	6103538	0.106308	8.771644	2.961172
500/0000000	0-500	11.0	338.0625	6132153	0.074556	8.696943	2.948510
500/0000000	0-500	12.0	110.8125	6153882	0.045777	8.480412	2.911593
5000		**12.0**	**110.8125**		**0,000005**	**8.479123**	**2.911389**

Tabela F.5: Experimento e).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	176.5000	5503844	0.105063	9.154099	3.006310
500/0000000	0-500	2.0	228.8125	5736345	0.015792	8.862593	2.976485
500/0000000	0-500	3.0	31.9375	5908350	0.042481	8.582693	2.925032
500/0000000	0-500	4.0	297.1875	6017884	0.013572	8.826181	2.970227
500/0000000	0-500	5.0	248.0625	6058846	0.094482	8.778692	2.962312
500/0000000	0-500	6.0	248.0625	6058846	0.094482	8.778692	2.962312
500/0000000	0-500	7.0	490.6875	6153421	0.015718	8.911264	2.984389
500/0000000	0-500	8.0	74.1875	6188703	0.166961	8.487123	2.912537
500/0000000	0-500	9.0	204.7500	6210029	0.008454	8,668647	2.943037
500/0000000	0-500	10.0	470.0000	6222118	0.070766	8.641841	2.939209
500/0000000	0-500	11.0	236.7500	6211378	0.039109	9,051678	3.007803
500/0000000	0-500	12.0	240.6250	6200897	0.031409	8.924824	2.984829
5000		**8.0**	**74.1875**		**0.000009**	**8.478990**	**2.911387**

Tabela F.6: Experimento f).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	316.8125	5526221	0.073866	9.292390	3.047698
500/0000000	0-500	2.0	455.0000	5766666	0.038515	9.256233	3.041695
500/0000000	0-500	3.0	304.6250	5879163	0.049998	9.238019	3.023820
500/0000000	0-500	4.0	384.6250	5947551	0.094238	9.213746	3.034633
500/0000000	0-500	5.0	478.8750	5993333	0.061907	8.956819	2.992202
500/0000000	0-500	6.0	279.3125	6067131	0.121002	8.823395	2.969515
500/0000000	0-500	7.0	387.6875	6095464	0.034157	9.047977	3.007595
500/0000000	0-500	8.0	278.3750	6107536	0.114024	8.970227	2.994475
500/0000000	0-500	9.0	257.3750	6123550	0.125419	8.976521	2.995478
500/0000000	0-500	10.0	499.9375	6148729	0.153718	9.094526	3.015182
500/0000000	0-500	11.0	220.6250	6168149	0.010148	9.240638	3.039556
500/0000000	0-500	12.0	490.5625	6191446	0.039238	9.216367	3.034523
5000		**9.0**	**257.3750**		**0.000006**	**8.912318**	**2.984525**

Tabela F.7: Experimento g).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	40.5625	5318988	0.532906	9.383993	3.062180
500/0000000	0-500	2.0	398.4375	5596683	0.084127	9.030010	3.001402
500/0000000	0-500	3.0	419.8125	5795120	0.041984	9.036823	3.005532
500/0000000	0-500	4.0	208.8750	5929830	0.039968	8.484255	2.912207
500/0000000	0-500	5.0	270.6250	6026267	0.015787	8.365842	2.891708
500/0000000	0-500	6.0	300.5000	6093687	0.026085	8.553709	2.921899
500/0000000	0-500	7.0	34.0000	6142904	0.031041	8.552575	2.924004
500/0000000	0-500	8.0	415.8750	6144900	0.014099	8.681412	2.946074
500/0000000	0-500	9.0	72.3750	6170871	0.022180	8.478549	2.910982
500/0000000	0-500	10.0	249.3750	6201481	0.033285	8.896152	2.981982
500/0000000	0-500	11.0	105.5000	6224426	0.035256	8.850760	2.974627
500/0000000	0-500	12.0	237.4375	6233047	0.154528	9.048509	3.007472
5000		**5.0**	**270.6250**		**0.000002**	**8.365658**	**2.891694**

Tabela F.8: Experimento h).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	309.3750	5676440	0.050082	8.830877	2.971394
500/0000000	0-500	2.0	264.3125	5915537	0.013126	8.583604	2.929262
500/0000000	0-500	3.0	25.4375	6098273	0.009073	8.566201	2.926465
500/0000000	0-500	4.0	145.6875	6226845	0.013862	8.672448	2.944455
500/0000000	0-500	5.0	285.8125	6200377	0.003318	8.641374	2.938759
500/0000000	0-500	6.0	138.5000	6214156	0.179602	8.628575	2.936862
500/0000000	0-500	7.0	55.3750	6247497	0.016209	8.548202	2.922591
500/0000000	0-500	8.0	229.2500	6301386	0.045849	8.500091	2.913951
500/0000000	0-500	9.0	264.3750	6306277	0.011240	8.726623	2.953664
500/0000000	0-500	10.0	4.0000	6285115	0.024447	8.826551	2.970265
500/0000000	0-500	11.0	188.5625	6282041	0.053252	8.770814	2.952784
500/0000000	0-500	12.0	431.5625	6262760	0.020119	8.752582	2.942827
5000		**8.0**	**229.2500**		**0.000007**	**8.482328**	**2.911386**

Tabela F.9: Experimento i).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	194.2500	5625651	0.109680	9.496888	3.081114
500/0000000	0-500	2.0	460.4375	5877218	0.325692	9.157663	3.024998
500/0000000	0-500	3.0	272.7500	5971656	0.052808	8.805547	2.966685
500/0000000	0-500	4.0	349.3125	6014462	0.037090	8,788041	2.963950
500/0000000	0-500	5.0	186.4375	6048876	0.047967	8.902101	2.982926
500/0000000	0-500	6.0	113.0625	6086690	0.022734	8.909786	2.984203
500/0000000	0-500	7.0	267.8125	6131510	0.012198	8.746079	2.956926
500/0000000	0-500	8.0	258.5625	6174859	0.035545	8.916140	2.985243
500/0000000	0-500	9.0	299.5625	6211807	0.032874	8.831491	2.971262
500/0000000	0-500	10.0	327.8750	6228733	0.064247	8.652266	2.938523
500/0000000	0-500	11.0	483.1875	6225178	0.127253	8.939241	2.989110
500/0000000	0-500	12.0	469.1250	6224667	0.012715	8.912706	2.984743
5000		**10.0**	**327.8750**		**0.000005**	**8.649082**	**2.938039**

Tabela F.10: Experimento j).

Iterações	t Lissajous	Passo	t final	Av. f	ϵ atingido	Valor de f	Raio
500/0000000	0-500	1.0	425.1875	5688508	0.076569	8.944206	2.990085
500/0000000	0-500	2.0	250.3125	5945182	0.012444	8.517368	2.917899
500/0000000	0-500	3.0	485.2500	6102896	0.037401	8.321062	2.884046
500/0000000	0-500	4.0	16.6250	6218890	0.020479	8.591154	2.928974
500/0000000	0-500	5.0	29.4375	6223863	0.017591	8.611145	2.933918
500/0000000	0-500	6.0	272.8750	6210455	0.039124	8.644853	2.939864
500/0000000	0-500	7.0	290.8125	6275036	0.006390	8.496910	2.914342
500/0000000	0-500	8.0	344.4375	6364166	0.012927	8.390786	2.879917
500/0000000	0-500	9.0	254.0000	6356412	0.091440	8.666189	2.943270
500/0000000	0-500	10.0	397.7500	6307425	0.062332	8.633183	2.937692
500/0000000	0-500	11.0	361.1875	6261210	0.152257	8.413002	2.899803
500/0000000	0-500	12.0	437.4375	6241159	0.103663	8.880583	2.976224
5000		**8.0**	**334.4375**		**0.0000005**	**8.390786**	**2.877809**

Centros e Raios

a)	2.9334	b)	2.9470	c)	2.9880	d)	2.9113	e)	2.9113
16.4529	12.8737	14.4592	8.1563	16.4409	12.7385	16.6743	12.7600	16.3839	12.7345
11.6726	12.4095	17.0876	12.5989	10.8021	6.4530	11.7928	12.2530	12.1510	12.7203
12.3038	15.7792	13.1795	16.2506	8.2873	11.2387	12.8906	15.1033	12.7518	16.2890
7.3303	16.0852	7.8885	16.2597	12.2230	15.6591	9.0354	17.0410	8.5750	16.5532
3.9223	17.3491	4.6012	16.3796	4.6673	16.4312	4.5709	16.5963	4.6655	16.3186
3.8714	12.4398	2.8031	12.2915	7.9383	16.1635	3.3635	11.7812	3.4329	13.1129
8.4808	10.7027	6.7526	12.0043	3.7537	12.6852	7.5214	12.1550	8.0106	11.6269
10.6607	3.5284	11.6553	12.1549	11.2777	12.3511	9.8776	7.6281	10.2724	8.3328
11.1842	6.7181	11.0418	3.8522	10.6688	3.6442	11.1063	3.7975	11.1063	3.7975
14.5077	8.3661	9.1977	8.1639	14.5119	8.1455	14.7141	8.0633	14.9171	7.9134
f)	2.9845	**g)**	2.8916	**h)**	2.9113	**i)**	2.9380	**j)**	2.8778
14.9842	11.7408	16.6464	12.3599	16.3480	12.8298	2.3316	16.4889	16.4938	12.5862
3.3066	13.3065	13.8955	15.4498	12.3163	13.0296	10.5353	8.4747	12.4540	12.9987
12.4590	14.0514	11.5750	12.3929	12.4607	16.5498	12.0944	15.7559	13.3267	16.4569
4.6595	16.1933	8.9635	16.9447	8.4107	16.5387	16.3382	12.9869	8.4310	16.6827
8.9496	17.0503	4.3852	16.5376	4.7028	16.4503	7.5673	16.7948	4.6778	16.5341
11.2894	3.4290	2.4710	12.1003	3.8371	12.5851	11.1249	3.8199	3.7815	12.9363
16.3153	12.8760	6.8171	12.2520	7.5578	11.6742	12.8218	11.2992	7.8910	11.7807
14.6561	7.9139	9.7311	8.0831	10.4447	8.2590	15.1685	7.9354	10.5688	8.4126
7.4239	11.7862	11.0603	3.6224	11.1063	3.7975	8.2097	11.7183	10.9968	3.7888
10.1608	8.7495	14.4534	8.0300	14.7853	8.0579	3.8424	12.8817	14.7680	7.8834

Figura F.1: Raios de Cobertura e Coordenadas das Antenas do Brasil.

F.2 Raios de Cobertura

A figura F.2 mostra os melhores resultados (Raios de Cobertura) encontrados em cada um dos conjuntos de experimentos das tabelas acima.

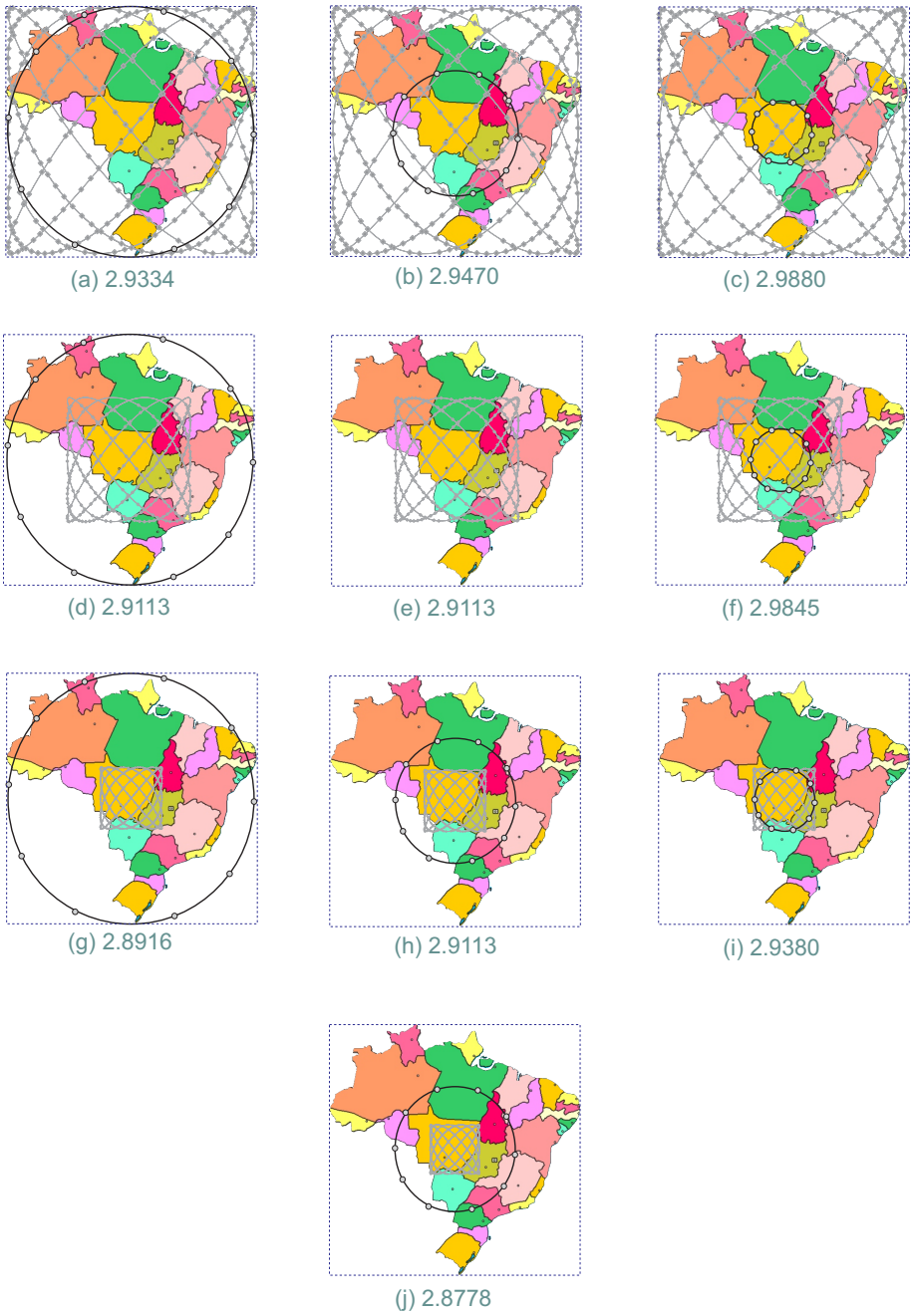

Figura F.2: Raios de Recobrimento dos experimentos do Brasil.