

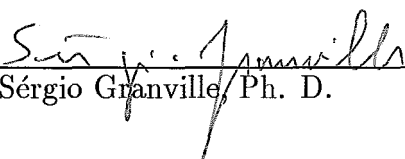
UM ALGORITMO DE MINIMIZAÇÃO USANDO UM MÉTODO DE REGIÃO
DE CONFIANÇA COM FORMATO ELIPSOIDAL PARA PROBLEMAS DE
PROGRAMAÇÃO NÃO LINEAR COM VARIÁVEIS CANALIZADAS

Fernanda Maria Pereira Raupp

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PRO-
GRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FE-
DERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁ-
RIOS PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGE-
NHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

Prof. Clóvis ~~Caesar~~ Gonzaga, Ph. D.
(presidente)



Prof. Sérgio Granville, Ph. D.



Prof. Paulo Roberto Oliveira, D. Sc.

RIO DE JANEIRO, RJ – BRASIL
OUTUBRO DE 1991

PEREIRA RAUPP, FERNANDA MARIA

Um Algoritmo de Minimização Usando um Método de Região de Confiança com Formato Elipsoidal para Problemas de Programação Não Linear com Variáveis Canalizadas [Rio de Janeiro] 1991

VIII, 92 p., 29.7 cm, (COPPE/UFRJ, M. Sc., ENGENHARIA DE SISTEMAS E COMPUTAÇÃO, 1991)

TESE – Universidade Federal do Rio de Janeiro, COPPE

1 – Algoritmo de Ponto Interior 2 – Programação Não Linear

3 – Região de Confiança 4– Variáveis Canalizadas

I. COPPE/UFRJ II. Título(Série).

*Aos meus pais
ao Augusto
e aos meus filhos*

Agradecimentos

Ao prof. Clóvis C. Gonzaga, pela sugestão do tema, pela orientação segura e constante, e pela dedicação e incentivo demonstrados durante o desenvolvimento desta tese.

Aos colegas de Otimização, em especial a Sérgio e Ricardo, e demais colegas que frequentaram o Laboratório de Sistemas da COPPE, pela ajuda e companheirismo.

Aos funcionários técnicos e administrativos do Programa de Engenharia de Sistemas e Computação, pela atenção e prestatividade.

A Capes, pela bolsa de estudos concedida.

Aos colegas do LNCC, em especial a Jaime E. Muñoz Rivera, Carlos Alberto Quadros Coimbra e Antônio Carlos Salgado Guimarães, pelas indispensáveis colaborações, e Ana, Regina, Miriain e Eduardo pelo grande incentivo.

Aos demais colegas do LNCC, que no convívio diário contribuíram de diversas formas para este trabalho.

Ao LNCC, pelas facilidades computacionais e pelo apoio administrativo.

Aos meus irmãos.

Resumo da Tese apresentada à COPPE como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M. Sc.)

Um Algoritmo de Minimização Usando Método de Região de Confiança com Formato Elipsoidal para Problemas de Programação Não Linear com Variáveis Canalizadas

Fernanda Maria Pereira Raupp

Outubro de 1991

Orientador: Clóvis Caesar Gonzaga

Programa: Engenharia de Sistemas e Computação

Neste trabalho, desenvolvemos um Algoritmo de Ponto Interior para Programação não Linear. Usamos o Método de Região de Confiança com formato elipsoidal para resolver problemas de minimização não lineares com Variáveis Canalizadas. Estuda-se as características da região de confiança com formato elipsoidal. A solução é obtida pelo uso do método de Levenberg-Marquardt considerando apenas soluções viáveis. Um tratamento é feito utilizando uma técnica de conjuntos ativos.

Os algoritmos método de região de confiança com formato elipsoidal, método de gradiente projetado e método de máximo declive foram implementados e testados para um conjunto de problemas não lineares com restrições simples elaborados através de uma bibliografia. Fizemos comparações entre os métodos e chegamos a conclusão de que o algoritmo desenvolvido é robusto.

Palavras-chave: Algoritmo de Ponto Interior, Método de Região de Confiança, Programação não Linear, Variáveis Canalizadas.

Abstract of Thesis presented to COPPE as partial fulfillment of the requirements for degree of Master of Science (M. Sc.)

A Trust Region Minimization Algorithm with Ellipsoidal Shape for Non Linear Programs with Simple Bounds.

Fernanda Maria Pereira Raupp

October 1991

Thesis Supervisor: Clóvis Caesar Gonzaga

Department: Programa de Engenharia de Sistemas e Computação

In this work an Interior Point Algorithm for Non Linear Programming is developed. A Trust Region Method with an ellipsoidal shape is used to solve non linear minimization problems with lower and upper bounds in each variable. The properties of the trust region with an ellipsoidal shape are studied. The solution is obtained by Levenberg-Marquardt's method considering only feasible points. An active set technique is used.

The Trust Region Method with ellipsoidal shape, projected gradients and steepest descent algorithms were implemented and tested for a set of non linear problems with Simple Bounds found in the literature. A comparative numerical study shows the robustness of the proposed algorithm.

Key Words: Interior Point Algorithm, Non Linear Programming, Trust Region Method, Simple Bounds.

Índice

I	Introdução	1
II	Introdução ao Método de Região de Confiança sem Restrições	3
II.1	Introdução	3
11.2	Método de Newton-Raphson	4
11.3	Método de Região de Confiança	6
III	Introdução ao Método de Gradiente Projetado com Variáveis Canalizadas	15
III.1	Introdução	15
111.2	Resolvendo o Problema Simplificado	17
III.3	Extensão para o caso de limites superiores e inferiores das variáveis .	23
IV	Introdução ao Método de Regiões de Confiança Elipsoidais com Variáveis Canalizadas	26
IV.1	Introdução	26
IV.2	A Região de Confiança	27
IV.3	O Mínimo da Função Quadrática no Elipsóide	28

IV.4 O Algoritmo de Minimização de Funções não Lineares usando Método de Região de Confiança Elipsoidal com Variáveis' Canalizadas	30
V Implementação e Testes	40
V.1 Introdução	40
V.2 Implementação Computacional	40
V.3 Uma Abordagem Geral dos Programas	41
V.4 Os Testes Realizados	44
VI Conclusões	82

Capítulo I

Introdução

Os métodos de otimização não linear baseados em buscas unidirecionais têm sido muito desenvolvidos ao longo de décadas. Em tais métodos, cada iteração principia pela determinação de uma direção de busca, seguida de um cálculo de tamanho do passo a ser executado. Assim, o tamanho do deslocamento é determinado fazendo uma busca na direção conhecida.

Os métodos de otimização não linear classificados como métodos de região de confiança se caracterizam pelo conhecimento prévio do tamanho máximo do passo e pela determinação da direção segundo um algoritmo. O tamanho do passo está relacionado com o tamanho de uma vizinhança em torno de um ponto conhecido. A essas vizinhanças damos o nome de regiões de confiança. Normalmente usamos bolas n -dimensionais como formato das regiões de confiança nos problemas não lineares sem restrições. O formato circular é simples e não favorece nenhuma direção em especial.

Nosso problema de otimização é a minimização de uma função não linear com restrições do tipo simples, o que também é conhecido como problemas com variáveis canalizadas. São ditas restrições simples as restrições que fixam limites inferiores e superiores para as variáveis. Decidimos usar o método de região de confiança na resolução do nosso problema de otimização. Pensou-se num formato de região de confiança adequado ao conjunto viável, definido pelas restrições às nossas variáveis, e escolhemos o formato elipsoidal. Com isso estaremos permitindo que reduções melhores da nossa função objetivo sejam feitas. O método de região de

confiança com formato elipsoidal é um método interior e a descrição de seu algoritmo é o assunto principal desta tese.

A seguir descrevemos o conteúdo dos capítulos seguintes.

No capítulo II, introduzimos o método de região de confiança para resolver problemas de minimização não lineares sem restrições. O método de Newton-Raphson é apresentado como motivação para a criação do método de região de confiança. A dificuldade do método de região de confiança é apresentada e solucionada pelo método chamado Levenberg-Marquardt.

No capítulo III, introduzimos o método de gradiente projetado para resolver problemas não lineares que apresentam restrições simples, ou seja, suas variáveis são do tipo canalizadas. Esse método foi proposto por Bertsekas e é um método baseado em buscas unidirecionais.

No capítulo IV, apresentamos um algoritmo que usa o método de região de confiança com formato elipsoidal para resolver problemas de minimização não lineares com variáveis canalizadas. Estuda-se as características da região de confiança de formato elipsoidal. A solução é obtida pelo uso do método de Levenberg-Marquardt considerando apenas soluções viáveis. E, como na maioria dos problemas práticos as soluções ótimas estão na fronteira da região viável, podendo ocasionar grande instabilidade numérica quando os elipsóides ficam cada vez mais finos a medida que vão se aproximando da fronteira, um tratamento é feito através da utilização de uma técnica de conjuntos ativos.

No capítulo V, abordamos as implementações computacionais, os testes realizados e as comparações entre três algoritmos de minimização: método de gradiente projetado, método de região de confiança com formato elipsoidal usando o método de Levenberg-Marquardt, e o método de máximo declive tendo como base o método de gradiente projetado.

Por fim, no capítulo VI, apresentamos as conclusões do nosso trabalho.

Capítulo II

Introdução ao Método de Região de Confiança sem Restrições

II.1 Introdução

Considerando o problema

$$\begin{array}{ll} \text{(P)} & \min f(z) \\ & \text{s. a } x \in \mathbb{R}^n \end{array}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é não linear, limitada inferiormente em \mathbb{R}^n , $f(\cdot) \in C^2$, e portanto o conjunto $\{x : f(x) \leq x_0\}$ é fechado e limitado. Iremos resolvê-lo usando um método de minimização chamado de método de região de confiança. Ele é aplicado, em geral, em funções não lineares e está baseado no método de Newton-Raphson para funções sem restrições.

Modelo Geral de Algoritmos de Minimização Usando Buscas Unidirecionais

Em geral, os métodos de minimização de uma função $f(\cdot) \in C^2$, sem restrições, são iterativos, obedecendo à condição de decréscimo da função expressa por

$$f(x^{k+1}) < f(x^k), \forall k \geq 0,$$

como característica da boa definição do algoritmo. Sendo x^k a estimativa atual de um ponto de mínimo x^* , o algoritmo segue basicamente os passos:

1. determina-se uma direção de busca p^k , que é uma direção de decréscimo da função, isto é,

$$p^{k^t} \nabla f(x^k) < 0;$$

2. calcula-se o tamanho do passo α^k nessa direção de modo que

$$f(x^k + \alpha^k p^k) < f(x^k);$$

3. atualiza-se a estimativa do ponto de mínimo e a iteração

$$x^{k+1} \leftarrow x^k + \alpha^k p^k,$$

$$k \leftarrow k + 1;$$

4. se x^{k+1} satisfaz condições de convergência, então x^{k+1} é o resultado do algoritmo; senão volta-se ao passo em 1.;

Vale introduzir agora o método de Newton-Raphson no qual o nosso algoritmo de região de confiança está baseado.

II.2 Método de Newton-Raphson

Vamos considerar, além das hipóteses mencionadas anteriormente, que a função $f(\cdot)$ seja estritamente convexa em \mathbb{R}^n . O método de Newton-Raphson aproxima a função $f(\cdot)$, cujas derivadas de primeira e segunda ordens são definidas e contínuas, por uma expansão dos três primeiros termos da série de Taylor em torno de um ponto conhecido x^k , isto é,

$$f(x^k + d) \approx f(x^k) + \nabla f(x^k)^t d + \frac{1}{2} d^t \nabla^2 f(x^k) d$$

onde

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x) \right)^t$$

e

$$[\nabla^2 f(x)]_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}(x).$$

Função Quadrática

Definimos, então, a função quadrática que aproxima $f(\cdot)$ em torno de x^k como sendo

$$q^k(d) = f(x^k) + \nabla f(x^k)^t d + \frac{1}{2} d^t \nabla^2 f(x^k) d.$$

Notação

A seguir apresentamos a notação que usaremos, neste e nos demais capítulos, para efeito de simplificação:

$$f^k = f(x^k),$$

$$g^k = \nabla f(x^k)$$

$$H^k = \nabla^2 f(x^k).$$

Passo de Newton-Raphson

A minimização de $f(\cdot)$ em \mathbb{R}^n a partir de x^k , quando esse estiver próximo do ponto ótimo de $f(\cdot)$, é obtida fazendo

$$x^{k+1} = x^k + d^k,$$

onde d^k é o ponto de mínimo global da função quadrática $q^k(\cdot)$. Tal ponto d^k existe e é único pelo fato de H^k ser definida positiva por hipótese, o que implica na aproximação quadrática também ser estritamente convexa. Ademais, pela condição necessária e suficiente de mínimo global, d^k satisfaz

$$\nabla q^k(d) = 0,$$

ou, equivalentemente,

$$d^k = - [H^k]^{-1} g^k.$$

O método de Newton-Raphson está perfeitamente definido pois H^k é não singular. Nesse método o tamanho do passo será sempre unitário ao longo da direção d^k . Na figura II.1, ilustramos o passo do método de Newton-Raphson- d^k .

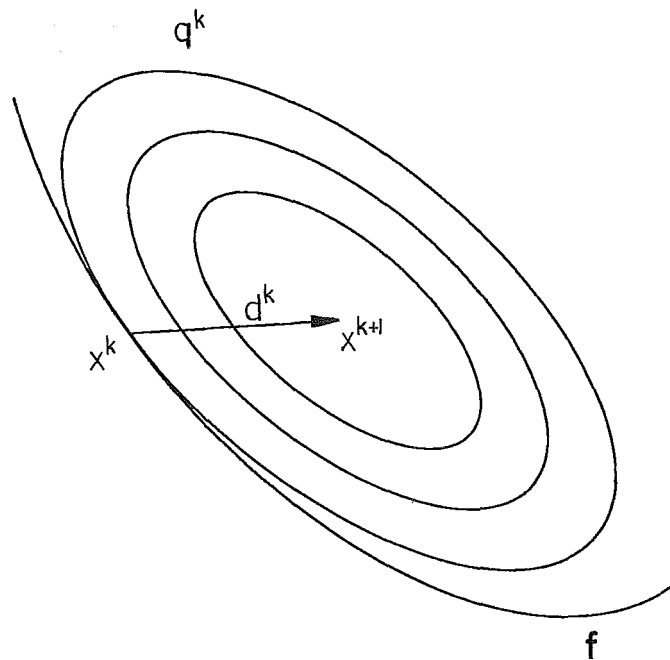


Figura 11.1: O deslocamento do método de Newton-Raphson.

O método de Newton-Raphson, para funções estritamente convexas, converge localmente, (no sentido de que o método depende da escolha do ponto inicial), com ordem de convergência quadrática. (Veja Teorema 3.1.1 em [6].)

Observamos, então, que o método de Newton-Raphson se aplica somente a funções estritamente convexas. Quando este não for o caso, pode não ser possível definir uma direção de decréscimo para a função. O método de região de confiança permitirá a construção de uma direção de decréscimo para qualquer função não linear nos moldes de (P).

II.3 Método de Região de Confiança

O método de região de confiança é um método baseado no modelo quadrático da função não linear. A vantagem do uso do modelo quadrático, característica também do método de Newton-Raphson, se dá pela sua simplicidade e eficiência na prática. A maioria dos métodos de minimização costumam escolher a direção do passo para então decidir sobre o seu tamanho. O método de região de confiança determina um limite superior para o tamanho do passo a ser dado para depois encontrar a direção a ser tomada. Tivemos como referências para o estudo deste método Fletcher[6],

Moré[21] e Dennis e Schnabel[13]

Passo da Região de Confiança

O tamanho do passo do método de região de confiança está restrito, a cada iteração, por um parâmetro h^k , raio da bola n-dimensional centrada em x^k onde o deslocamento a partir do ponto x^k estará limitado. Veja a figura 11.2.

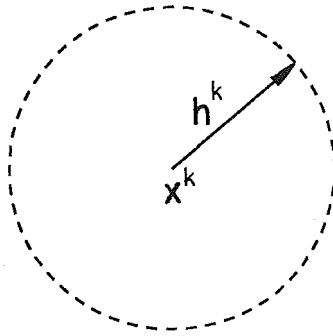


Figura 11.2: Limite superior para o tamanho do passo do método de região de confiança.

Consequentemente, a direção do passo ficará determinada quando minimizamos a aproximação quadrática $q^k(\cdot)$ na bola de centro x^k e raio h^k . Por isso, fazemos de maneira análoga

$$x^{k+1} = x^k + d^k$$

onde

$$d^k \in \operatorname{argmin}\{q^k(d) : \|d\| \leq h^k\}. \quad (11.1)$$

Vemos a seguir, na figura 11.3, o passo d^k como sendo o mínimo da aproximação quadrática $q^k(\cdot)$ na bola de centro x^k e raio h^k .

Vimos que o passo do método de região de confiança depende do parâmetro h^k . Dado h^k , o passo resultante d^k será eficiente se a aproximação quadrática satisfizer

$$q^k(d^k) \approx f(x^k + d^k),$$

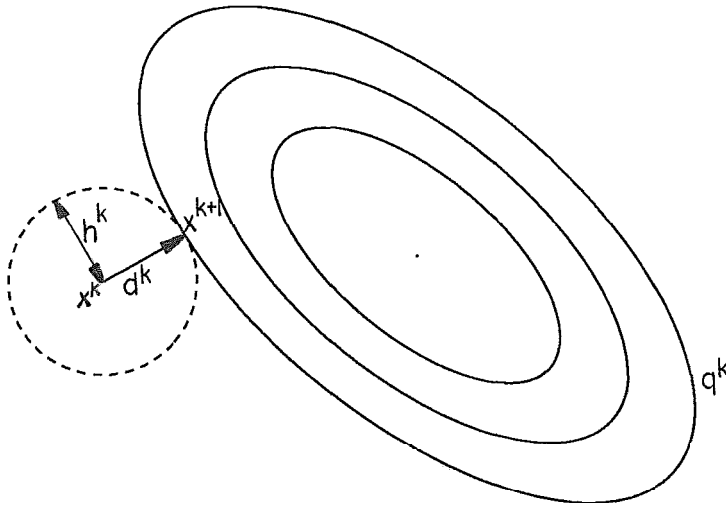


Figura 11.3: A direção do passo do método de região de confiança.

isto é, se $q^k(\cdot)$ é uma aproximação confiável de $f(\cdot)$ na região – daí o nome de "método de região de confiança". A adequação da aproximação quadrática em $\mathbf{x}^k + \mathbf{d}^k$ será verificada como mostraremos a seguir, e \mathbf{h}^k será modificado dependendo dessa verificação.

A Estimativa do Tamanho do Passo

Vamos quantificar a aproximação entre a função $f(\cdot)$ e seu modelo quadrático $q^k(\cdot)$ por uma razão r^k . Supondo a redução real dada por

$$\Delta f^k = f^k - f(\mathbf{x}^k + \mathbf{d}^k)$$

e a redução estimada por

$$\Delta q^k = q^k(0) - q^k(\mathbf{d}^k) = f^k - q^k(\mathbf{d}^k).$$

Tomemos

$$r^k = \frac{\Delta f^k}{\Delta q^k}.$$

Quanto mais perto de 1 r^k estiver, melhor será a aproximação de $f(\cdot)$ por $q^k(\cdot)$ em $\mathbf{x}^k + \mathbf{d}^k$. Se r^k estiver muito próximo de 1, podemos aceitar

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{d}^k$$

como próximo ponto e aumentar o tamanho do passo antes fixado por h^k . Se r^k estiver longe de 1, constata-se que a aproximação é ruim e deve-se refazer o processo com h^k menor. As modificações de h^k serão feitas pelas seguintes regras heurísticas:

- a. se $r^k > 0.75$ então $h^{k+1} = 2h^k$ e $x^{k+1} = x^k + d^k$;
- b. se $0.25 \leq r^k < 0.75$ então $h^{k+1} = h^k$ e $x^{k+1} = x^k + d^k$;
- c. se $r^k < 0.25$ então $h^{k+1} = h^k/4$ e $x^{k+1} = z^k$. Neste caso, $x^k + d^k$ não é aceito como próximo ponto.

A limitação do tamanho do passo dada por h^k evita que cálculos sejam feitos desnecessariamente. No caso de se ter vizinhanças de x^k com raios pequenos, o número de iterações do algoritmo pode aumentar muito. No caso contrário, quando os raios das vizinhanças de x^k são muito grandes, as aproximações da função não serão muito boas. Antes de apresentarmos o algoritmo do método de região de confiança, mostramos o próximo ponto dado pelo algoritmo na figura 11.4.

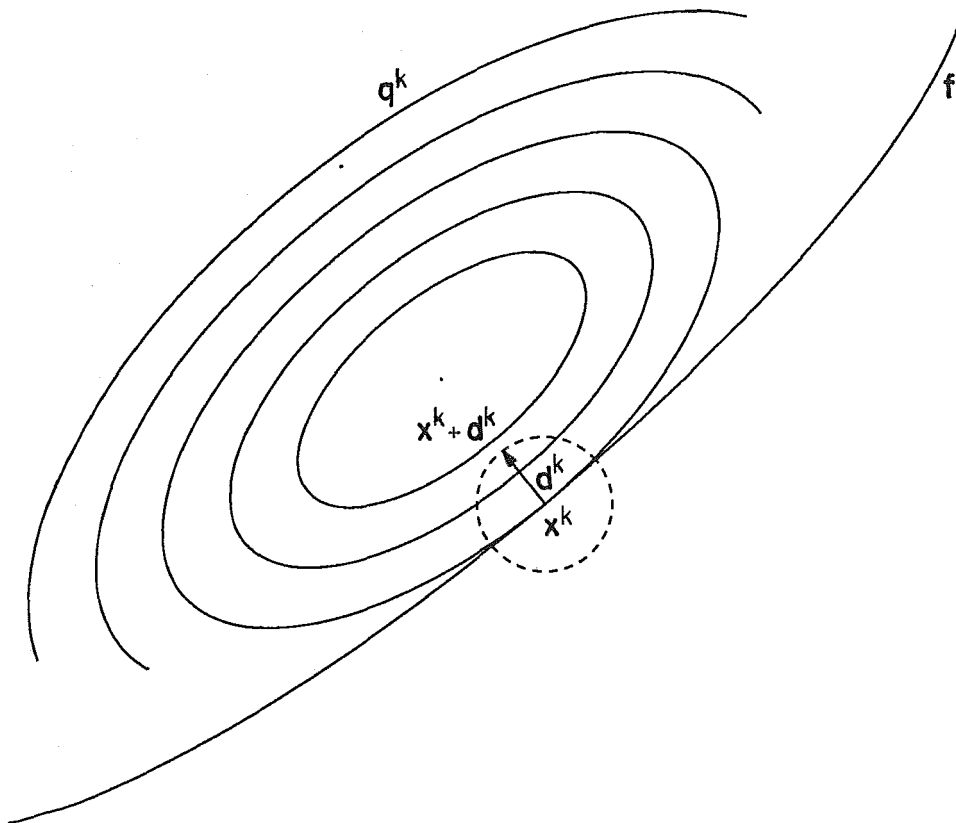


Figura 11.4: Método de região de confiança – o próximo ponto na iteração k .

O Algoritmo Geral do Método Região de Confiança

Definimos o algoritmo de região de confiança como segue. Em nossos algoritmos não explicitaremos a regra de parada, usando o teste "até convergir". Testes de parada práticos baseiam-se em número de iterações, variações da função objetivo ou em norma do gradiente dessa função.

Algoritmo II.1 Algoritmo Geral do Método de Região de Confiança: Da-

dos x^0 e h^0 .

$k = 0$

Repita

calcule g^k e H^k ;

ache $d^k \in \operatorname{argmin}\{q^k(d) \mid \|d\| \leq h^k\}$;

avalie $f(x^k + d^k)$ e r^k ;

se $r^k < 0.25$ faça $h^{k+1} = h^k/4$;

se $r^k \geq 0.75$ faça $h^{k+1} = 2h^k$;

se $0.25 \leq r^k < 0.75$ faça $h^{k+1} = h^k$;

se $r^k \geq 0.25$ faça $x^{k+1} = x^k + d^k$;

senão $x^{k+1} = x^k$;

$k = k + 1$;

Até convergir .

O algoritmo II.1 acima nos fornece pontos de acumulação x^∞ que satisfazem as condições necessárias de primeira e segunda ordens de otimalidade. (Veja Teorema 5.1.1 em [6])

Se um ponto de acumulação satisfizer a condição suficiente de segunda ordem de otimalidade, então $r^k \rightarrow 1$ e a convergência do algoritmo é de segunda ordem. (Veja Teorema 5.1.2 em [6])

O teorema anterior nos garante que para k suficientemente grande a base do algoritmo de região de confiança passa a ser o método de Newton-Raphson que apresenta convergência de segunda ordem. A heurística da restrição do passo, $\|d\| \leq h^k$, assegura que reduções significativas em $f(\cdot)$ foram realizadas antes que x^k penetrasse na região de convergência quadrática do método de Newton-Raphson.

Método de Levenberg-Marquardt

Resolver o subproblema posto em II.1 :

$$d^k \in \operatorname{argmin}\{g^k(d) \mid \|d\| \leq h^k\},$$

básico no método de região de confiança, não é uma tarefa simples. Não se garante que a solução global seja encontrada em um número finito de iterações. Existem métodos eficientes que encontram soluções aproximadas. (Ver Dennis e Schnabel em [13]). Para contornar tal dificuldade, a nível de implementação, utilizamos um método usualmente chamado Levenberg-Marquardt, desenvolvido por Moré e Sorensen [19], baseado em técnicas utilizadas inicialmente por Levenberg[16] e Marquardt[18]. Ele se sustenta no seguinte teorema:

Teorema II.1 *A correção d^k é a solução global do subproblema II.1 se, e somente se, existe $\mu \geq 0$ tal que*

$$\begin{aligned} (H^k + \mu I) d^k &= -g^k, \\ \mu (\|d^k\| - h^k) &= 0 \end{aligned} \tag{11.2}$$

e $(H^k + \mu I)$ é semi-definida positiva. E, ainda, se $(H^k + \mu I)$ for definida positiva, então d^k é a solução única do subproblema dado em II.1. (Veja Teorema 5.2.1 em [6].)

Essencialmente, o método de Levenberg-Marquardt se caracteriza por determinar $\mu > 0$ de modo que a matriz $(H^k + \mu I)$ seja definida positiva permitindo que o sistema em 11.2 seja resolvido a fim de obtermos uma solução única do subproblema 11.1, determinando, assim, o passo a ser dado no método de região de confiança. Notamos que, o passo d^k é encontrado sem o uso da restrição $\|d\| \leq h^k$. O valor μ é usado como parâmetro de controle nas iterações, e, assim, d^k é determinado a partir dele.

Este método possibilita resolver o caso geral, mesmo quando a função a ser minimizada apresenta autovalores negativos.

O Algoritmo de Região de Confiança usando o Método de Levenberg-Marquardt

Vamos apresentar um algoritmo similar ao modelo geral de algoritmo de região de confiança apresentado em 11.1. Devemos observar que mudanças em h^k foram substituídas por mudanças inversas em μ^k . Verificamos que o crescimento de μ^k causa decréscimo em $\|d^k\|$, e vice-versa. Segue abaixo o algoritmo de Levenberg-Marquardt:

Algoritmo II.2 Algoritmo do Método de Levenberg-Marquardt: *Dados* x^0 e μ^0 .

$k = 0$

Repita

calcule g^k e H^k ;

faça, caso necessário, $(H^k + \mu^k I)$ ser definida positiva aumentando μ^k ;

resolva o sistema $(H^k + \mu^k I)d^k = -g^k$ para achar d^k ;

avale $f(x^k + d^k)$ e r^k ;

se $r^k < 0.25$ então $\mu^{k+1} = 4\mu^k$;

se $r^k \geq 0.75$ então $\mu^{k+1} = \mu^k/2$;

se $0.25 \leq r^k < 0.75$ então $\mu^{k+1} = \mu^k$;

se $r^k < 0.25$ faça $x^{k+1} = x^k$;

senão $x^{k+1} = x^k + d^k$;

$k = k + 1$;

Até convergir.

Uma vez que o algoritmo foi apresentado, vamos visualizar seus passos através da trajetória descrita pelos mesmos.

Seja o passo do método de Levenberg-Marquardt da iteração k dado por

$$d(\mu) = - (H^k + \mu I)^{-1} g^k.$$

Temos para $\mu = 0$, quando $\|d\|$ é máxima,

$$d_0 = d(0) = - (H^k)^{-1} g^k, \quad (\text{II.3})$$

sendo exatamente o passo do método de Newton-Raphson. Para μ muito grande, ou $\|d\|$ próxima de zero, temos:

$$d_\mu = d(\mu) \approx -\frac{1}{\mu}g^k, \quad (\text{II.4})$$

representando um passo na direção de máximo declive da função $f(\cdot)$. Na figura 11.5, estão os dois passos 11.3 e 11.4.

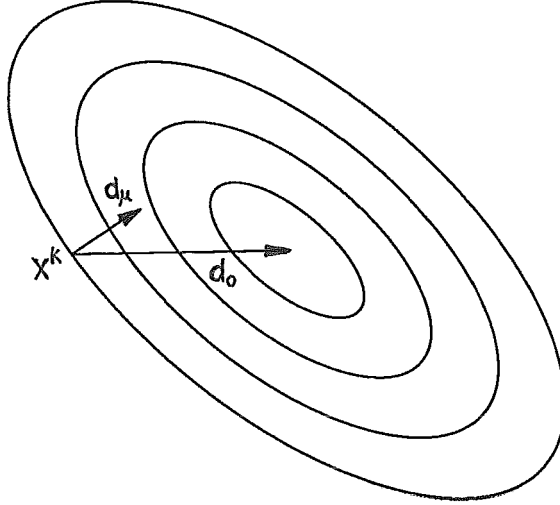


Figura 11.5: Passos d_0 e d_μ .

Se para uma mesma aproximação quadrática da função $f(\cdot)$ em x^k ; $q^k(\cdot)$, variamos o valor de μ de 0 a um valor muito grande, então descreveremos a trajetória $d^k(\mu)$ de todos os possíveis deslocamentos na iteração k . Veja na figura II.6 a trajetória $d^k(\mu)$.

A trajetória do método de região de confiança, cujo algoritmo está apresentado em 11.1, é igual à trajetória do método de Levenberg-Marquardt, porém a parametrização é diferente. Fazendo variar o raio da bola de centro em x^k de 0 a um valor muito grande, a trajetória $d^k(h)$ de todos os possíveis deslocamentos a partir de x^k é descrita no sentido oposto. Observe a figura 11.7.

A resolução do subproblema II.1 pelo método de Levenberg-Marquardt não alterou as condições de convergência do método de região de confiança, o qual apresenta convergência quadrática de segunda ordem.

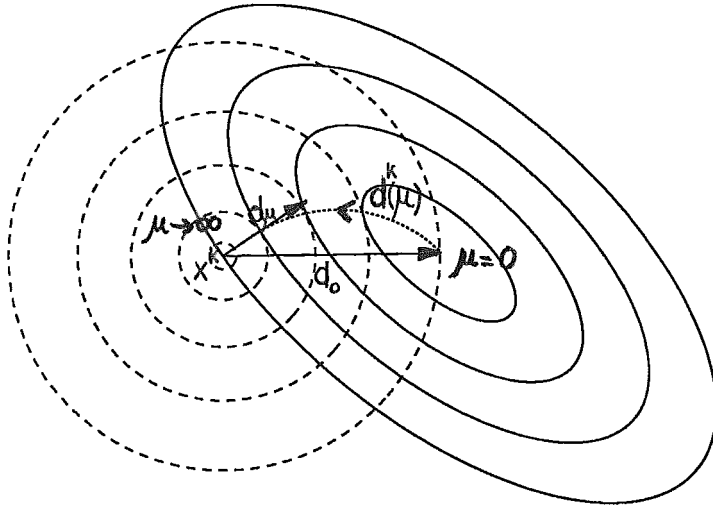


Figura 11.6: A trajetória do método de Levenberg-Marquardt para $\mu \in [0, \infty]$.

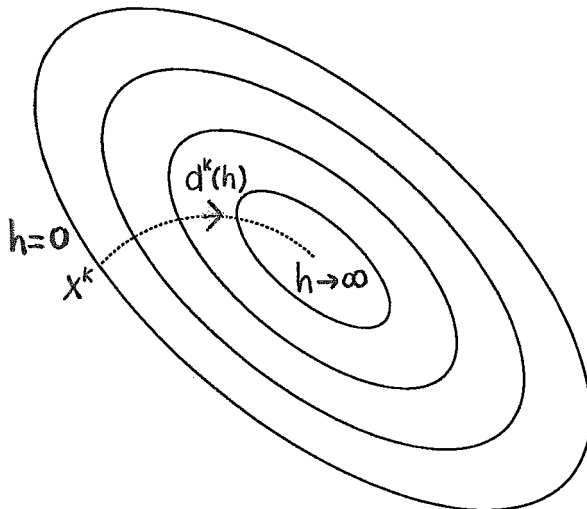


Figura 11.7: A trajetória do método de região de confiança para $h \in [0, \infty]$

Capítulo III

Introdução ao Método de Gradiente Projetado com Variáveis Canalizadas

III.1 Introdução

Considere o problema

$$(PR1) \quad \min f(x) \\ \text{s. a } l \leq x \leq u$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função não linear, convexa, $f(\cdot) \in C^2$ e os vetores l e u são conhecidos com $l \leq u$. Vamos considerar que $\nabla^2 f(x)$ seja definida positiva para $l \leq x \leq u$ e, ainda, que o conjunto viável seja definido por $S_1 = \{x \in \mathbb{R}^n \mid l \leq x \leq u\}$. Bertsekas em [3] sugere que a resolução deste problema (PR1) com algumas modificações pode ser obtida resolvendo o seguinte problema para a mesma função:

$$(PR2) \quad \min f(x) \\ \text{s. a } x \geq 0,$$

para o conjunto viável definido como $S_2 = \{x \in \mathbb{R}^n \mid x \geq 0\}$. Em geral, os problemas (PR1) e (PR2) são ditos problemas não lineares com restrições simples, ou, mais recentemente, ditos problemas não lineares com variáveis canalizadas. Muitos problemas encontrados na prática são formulados como (PR1) e/ou (PR2).

Vamos estudar primeiramente a resolução do problema em (PR2). A condição necessária de primeira ordem para que um vetor \bar{x} seja mínimo local do

problema (PR2) é que

$$\frac{\partial f}{\partial x_i}(\bar{x}) \geq 0 \text{ para } \bar{x}_i = 0, \forall i = 1, \dots, n, \quad (\text{III.1})$$

considerando a possibilidade de \bar{x} ser um ponto da fronteira, e

$$\frac{\partial f}{\partial x_i}(\bar{x}) = 0 \text{ para } \bar{x}_i > 0, \forall i = 1, \dots, n. \quad (\text{III.2})$$

Um primeiro algoritmo de minimização do problema (PR2) é feito de maneira análoga ao método de máximo declive da função $f(\cdot)$ em cada iteração. O algoritmo utiliza a direção de máximo declive a partir de x^k e projeta os pontos $x^k - \alpha^k g^k$ sobre a região viável. Dado um vetor $z \in \mathbb{R}^n$, definimos sua projeção sobre S_2 como

$$[z]^+ = \begin{bmatrix} \max\{0, z_1\} \\ \max\{0, z_2\} \\ \vdots \\ \max\{0, z_n\} \end{bmatrix}.$$

Apresentamos, então, o primeiro algoritmo com segue:

$$x^{k+1} = x^k(\alpha^k) \quad (\text{III.3})$$

$$x^k(\alpha) = [x^k - \alpha g^k]^+, \alpha \geq 0, \forall k = 0, 1, \dots,$$

onde α^k , o tamanho do passo na iteração k , será o resultado da busca tipo Armijo dada por

$$\alpha^k = \beta^{m^k} s$$

onde m^k é o primeiro inteiro não negativo m que satisfaz

$$f(x^k) - f[x^k(\beta^m s)] \geq \sigma g^{k^t} [x^k - x^k(\beta^m s)]^+, \quad (\text{III.4})$$

e os escalares s, β, σ são constantes escolhidas tais que $s > 0$, $\beta \in (0, 1)$ e $\sigma \in (0, \frac{1}{2})$.

Geometricamente podemos interpretar 111.3 como na figura 111.1.

Um segundo e principal algoritmo, que será apresentado na próxima seção, utiliza o esquema de 111.3 conjuntamente com informações de segunda ordem da função $f(\cdot)$ de modo a combinar a simplicidade do método de máximo declive da função com a sofisticada e rápida convergência do método de Newton com restrições. O tamanho do passo α^k será determinado a cada iteração por uma regra de Armijo modificada para problemas com restrições, como veremos.

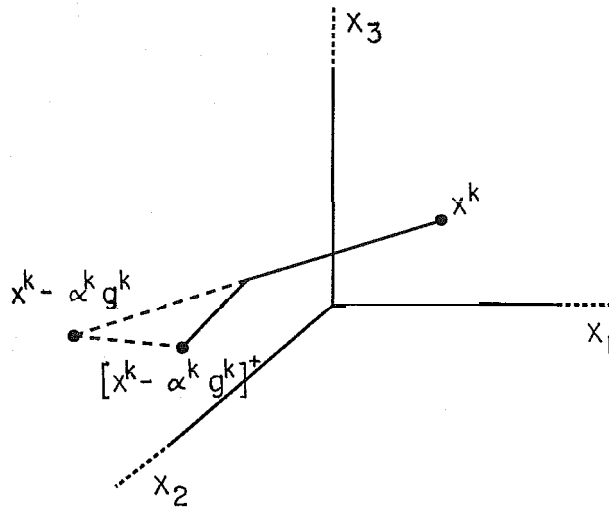


Figura 111.1: A projeção do próximo ponto sobre o conjunto viável S_2 .

III.2 Resolvendo o Problema Simplificado

Nosso objetivo é resolver o problema posto em (PR2) para tentar resolver facilmente o problema em (PR1). Dado o problema (PR2) :

$$\begin{aligned} \min \quad & f(x) \\ \text{s. a} \quad & x \geq 0 \end{aligned}$$

se \bar{x} satisfaz a condição necessária de primeira ordem III.1 e 111.2 então \bar{x} será chamado ponto crítico com respeito ao problema (PR2).

Direção e Tamanho do Passo

Modificando a sugestão apresentada, alterando a direção de decréscimo da função apresentada em III.3, em cada iteração, fazendo

$$x^{k+1} = [x^k - \alpha^k D^k g^k]^+, \quad (111.5)$$

e escolhendo para D^k uma matriz simétrica, definida positiva e parcialmente diagonal, cujos elementos da diagonal $d_{ii}^k, i = 1, \dots, n$, satisfazem

$$\underline{d} \leq d_{ii}^k \leq \bar{d}, \forall i = 1, \dots, n; k = 0, 1, \dots,$$

onde \underline{d} e \bar{d} são escalares positivos, a direção de busca fica então definida por

$$p^k = D^k g^k.$$

E o escalar α^k , que define o tamanho do passo, será o resultado da busca de Armijo modificada, para problemas com restrições, feita ao longo da curva

$$x^k(\alpha) = [x^k - \alpha D^k g^k]^+, \alpha \geq 0,$$

como veremos.

A Escolha da Matriz D^k

Nem sempre será possível reduzir a função $f(\cdot)$ na iteração $k + 1$ se temos para D^k somente as propriedades mencionadas acima. Veja um exemplo na figura 111.2.

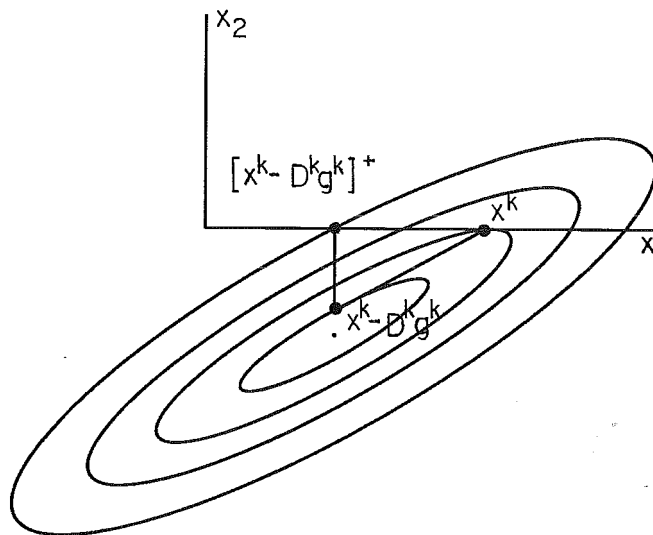


Figura 111.2: A escolha imprópria da matriz D^k

Identificaremos a classe de matrizes D^k , na qual será sempre possível reduzir a função objetivo. Vamos introduzir, para isto, a definição

$$I_+(x) = \left\{ i \mid x_i = 0, \frac{\partial f}{\partial x_i}(x) > 0 \right\}, \forall x \geq 0.$$

$I_+(x)$ é o conjunto dos índices das componentes nulas de x com variação positiva da função objetivo, isto é, o conjunto dos índices das variáveis ativas na iteração k .

Dizemos que uma matriz simétrica, D , $n \times n$, com elementos d_{ij} é diagonal com respeito a um subconjunto de índices $I \subset \{1, 2, \dots, n\}$ se

$$d_{ij} = 0 \quad \forall i \in I, j = 1, 2, \dots, n, j \neq i.$$

Exemplificando, supondo $n = 5$ e $\mathbf{I} = \{1, 3\}$, então D é da forma:

$$D = \begin{bmatrix} d_{11} & 0 & 0 & 0 & 0 \\ 0 & d_{22} & 0 & d_{24} & d_{25} \\ 0 & 0 & d_{33} & 0 & 0 \\ 0 & d_{24} & 0 & d_{44} & d_{45} \\ 0 & d_{25} & 0 & d_{45} & d_{55} \end{bmatrix}.$$

Vamos introduzir também o lema:

Lema 111.1 *Sejam $x \geq 0$ e D uma matriz simétrica definida positiva que é diagonal com respeito a $I_+(x)$. Considere, ainda,*

$$x(\alpha) = [x - \alpha Dg]^+, \forall \alpha \geq 0.$$

Então temos:

(a) *o vetor x é um ponto crítico com respeito ao problema (PR2) se, e somente se,*

$$x = x(\alpha), \forall \alpha \geq 0;$$

(b) *se x não é ponto crítico com respeito ao problema (PR2) então existe um escalar $\bar{\alpha} > 0$ tal que*

$$f[x(\alpha)] < f(x), \forall \alpha \in (0, \bar{\alpha}].$$

Consequentemente, a matriz D^k na iteração

$$x^{k+1} = [x^k - \alpha^k D^k g^k]^+$$

deve ser escolhida de maneira a ser diagonal com respeito a um subconjunto de índices que contém

$$I_+(x^k) = \{i \mid x_i^k = 0, g_i^k > 0\}.$$

(Veja Proposição 1 em [3])

Com o objetivo de superar certas dificuldades causadas pelo fato de $I_+(x^k)$ exibir algumas discontinuidades na fronteira do conjunto de restrições, o que

pode provocar um engarrafamento do algoritmo, é necessário aumentar o conjunto $I(x^k)$ relaxando a igualdade

$$x_i^k = 0$$

para

$$0 \leq x_i^k \leq e^k$$

onde

$$e^k = \min\{\varepsilon, w^k\}.$$

O escalar e é uma constante dada geralmente pequena e o escalar w^k é obtido fazendo

$$w^k = \|x^k - [x^k - Mg^k]^+\|.$$

A matriz M é uma matriz diagonal, definida positiva, também dada, que pode ser, por exemplo, a matriz identidade com dimensão $n \times n$. O escalar w^k nos dá a noção do que é 'pequeno' em cada iteração, enquanto ε é algo pequeno à todas iterações do algoritmo. Assim para efeito de notação o novo conjunto fica definido como

$$I_+^k = \{i \mid 0 \leq x_i^k \leq e^k, g_i^k > 0\}, \quad \forall x \geq 0.$$

Regra de Armijo Modificada

A regra de Armijo será expressa, com a introdução do conjunto I_+^k , por

$$\alpha^k = \beta^{m^k} \tag{III.6}$$

onde m^k é o primeiro inteiro não negativo m tal que

$$f(x^k) - f[x^k(\beta^m)] \geq \sigma \left\{ \beta^m \sum_{i \notin I_+^k} g_i^k p_i^k + \sum_{i \in I_+^k} g_i^k [x_i^k - x_i^k(\beta^m)]^+ \right\}. \tag{III.7}$$

A regra para o tamanho do passo III.6- III.7 está próxima daquela para problemas sem restrições. Quando $I_+^k = \emptyset$, a expressão do lado direito de III.7 fica

$$\sigma \beta^m g^{k^t} p^k,$$

que é idêntica à correspondente na regra de Armijo para problemas sem restrições. No caso de $I_+^k = \{1, 2, \dots, n\}$, então a desigualdade III.7 é idêntica à III.4.

Pelo lema 111.1, temos que a expressão à direita de 111.7 é não negativa e, é somente positiva se, e somente se, x^k não for ponto crítico. Quando x^k não for um ponto crítico o tamanho do passo α^k será sempre determinado em um número finito de iterações. Se x^k é um ponto crítico, então, temos que ambos os termos do lado direito da desigualdade 111.7 são nulos para todo $\alpha \geq 0$ e conseqüentemente 111.7 será satisfeita por $\alpha = 0$.

O Algoritmo

Vamos mostrar a seguir o algoritmo de Gradiente Projetado para variáveis positivas.

Algoritmo 111.1 Algoritmo de Gradiente Projetado para Variáveis Positivas: *Dados x^0 , $\epsilon > 0, M, \beta \in (0, 1)$ e $\alpha \in (0, 0.5)$.*

$k = 0$

Repita

calcule g^k e H^k ;

ache ϵ^k ;

determine o conjunto ativo I_+^k ;

monte D^k diagonal com respeito ao conjunto I_+^k ;

determine a direção $p^k = D^k g^k$;

faça a busca de Armijo com relação a I_+^k obtendo α^k ;

avalie $x^{k+1} = x^k(\alpha^k) = [x^k - \alpha^k p^k]^+$;

$k = k + 1$;

Até **convergir**.

Concluindo, o algoritmo 111.1 está bem definido, decresce o valor da função objetivo a cada iteração k para a qual x^k não é ponto crítico e essencialmente termina quando x^k é ponto crítico.

Agora falaremos à respeito da convergência e taxa de convergência do algoritmo proposto em III.1.

O algoritmo apresentado acima nos fornece um ponto de acumulação que satisfaz as propriedades de um ponto crítico para o problema em (PR2). (Veja

Proposicao 2 em [3].)

Veremos que além do algoritmo ser facilmente implementável e convergente ele apresenta mais uma vantagem. Definindo o conjunto

$$B(x) = \{i \mid x_i = 0\} \forall x \geq 0,$$

que caracteriza as restrições ativas em x , temos para x^* mínimo local, satisfazendo a segunda condição suficiente de otimalidade, que o algoritmo identificará as restrições ativas em x^* em um número finito de iterações, ou seja, existirá \bar{k} tal que

$$B(x^*) = B(x^k) \forall k > \bar{k}.$$

(Veja Proposição 3 em [3])

Temos como propriedade do algoritmo 111.1 o fato do método de minimização com restrições se reduzir a um método de minimização sem restrições em um subconjunto, T , dito das restrições ativas em x^* :

$$T = \{x \mid x_i = 0, \forall i \in B(x^*)\}$$

e, ainda, para algum índice \bar{k} , teremos

$$I_+^k = B(x^*), \forall k > \bar{k}.$$

Isto mostra que se a parte da matriz D^k , correspondente aos índices $i \in I_+^k$ for escolhida para ser a inversa da matriz Hessiana de $f(\cdot)$ com relação a esses índices, então o algoritmo efetivamente se reduz ao método de Newton restrito ao subespaço T .

Considerando em 111.5, que

$$D^k = [J^k]^{-1}$$

onde J^k é a matriz cujos elementos J_{ij}^k são

$$[J^k]_{ij} = \begin{cases} 0 & \text{se } (i \neq j) \text{ e } (i \in I_+^k \text{ ou } j \in I_+^k), \\ [H^*], & \text{caso contrário.} \end{cases}$$

Então o algoritmo III.1 converge para x^* , mínimo local, e a taxa de convergência de $\{ \|x^k - x^*\| \}$ é superlinear. (Veja Proposição 4 em [3])

III.3 Extensão para o caso de limites superiores e inferiores das variáveis

O algoritmo III.1 descrito anteriormente pode ser facilmente estendido para compreender problemas do tipo (PR1)

$$\begin{aligned} \min \quad & f(x) \\ \text{s. a} \quad & l \leq x \leq u \end{aligned}$$

onde l e u são vetores dados para serem os limites inferior e superior de x com $l \leq u$.

Para isto, precisamos modificar o conjunto I_+^k para

$$I_{\#}^k = \left\{ i \mid \left(l_i \leq x_i^k \leq l_i + \varepsilon^k \text{ e } g_i^k > 0 \right) \text{ ou } \left(u_i - \varepsilon^k \leq x_i^k \leq u_i \text{ e } g_i^k < 0 \right) \right\}$$

e a definição $x^k(er)$ fica

$$x^k(a) = [x^k - \alpha D^k g^k]_{\#},$$

onde para todo $z \in \mathbb{R}^n$, definimos sua projeção sobre S_1 como

$$[z]_{\#} = \begin{cases} u_i & \text{se } u_i < z_i, \\ z_i & \text{se } l_i \leq z_i \leq u_i, \\ l_i & \text{se } z_i < l_i. \end{cases}$$

Veja na figura III.3 a interpretação geométrica de $[x^k - \alpha^k D^k g^k]_{\#}$.

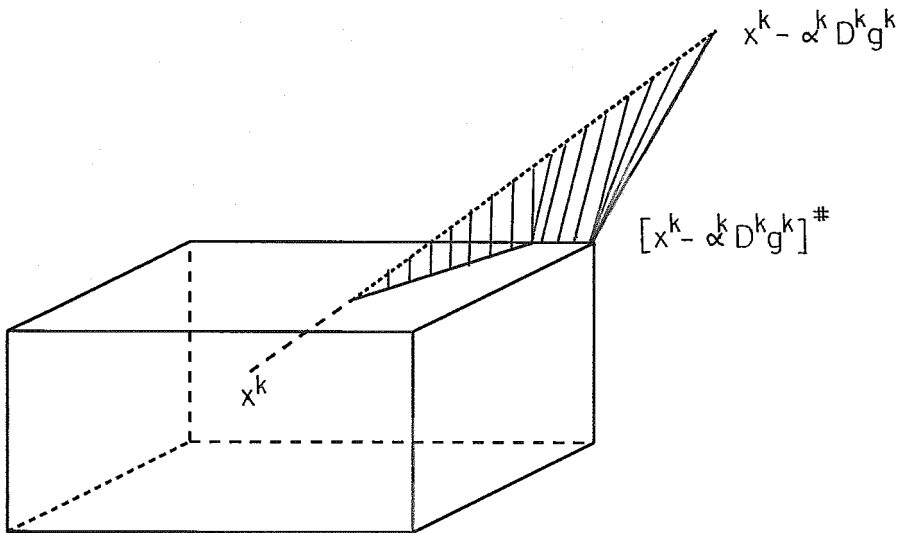


Figura III.3: A projeção do próximo ponto sobre o conjunto viável S_1 .

Analogamente, ε^k será dado por

$$\varepsilon^k = \min\{\varepsilon, w_{\#}^k\},$$

onde

$$w_{\#}^k = \|x^k - [x^k - Mg^k]_{\#}\|.$$

A matriz D^k é escolhida sendo definida positiva e diagonal com respeito a $I_{\#}^k$, e M é uma matriz definida positiva diagonal constante. A iteração é dada por

$$x^{k+1} = x^k(\alpha^k)$$

onde α^k é escolhido pela regra de Armijo 111.6- III.7 com $[x_i^k - x_i^k(\beta^m)]_{\#}$ no lugar de $[x_i^k - x_i^k(\beta^m)]^+$.

Este algoritmo considera o fato de algum limite superior u ; ser $+\infty$ e/ou algum limite inferior l ; ser $-\infty$, ou seja, quando apenas um limite superior e/ou um limite inferior é imposto a alguma variável x ; do problema. O algoritmo estendido às variáveis canalizadas é apresentado a seguir.

Algoritmo III.2 Algoritmo de Gradiente Projetado para Variáveis Canalizadas: Dados $x^0, l, u, e > 0, M, \beta \in (0, 1)$ e $\sigma \in (0, 0.5)$.

$k = 0$

Repita

calcule g^k e H^k ;

ache $\varepsilon_{\#}$;

determine o conjunto ativo $I_{\#}^k$;

monte D^k diagonal com respeito ao conjunto $I_{\#}^k$;

determine a direção $p^k = D^k g^k$;

faça a busca de Armijo com relação a $I_{\#}^k$ obtendo u^k ;

avalie $x^{k+1} = x^k(\alpha^k) = [x^k - \alpha^k p^k]_{\#}$;

$k = k + 1$;

Até convergir.

Mostramos a seguir um exemplo de minimização sobre um conjunto ativo. Observe a figura 111.4.

Este outro exemplo, na figura III.5, apresenta uma busca e projeção para um problema não linear $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ com variáveis canalizadas.

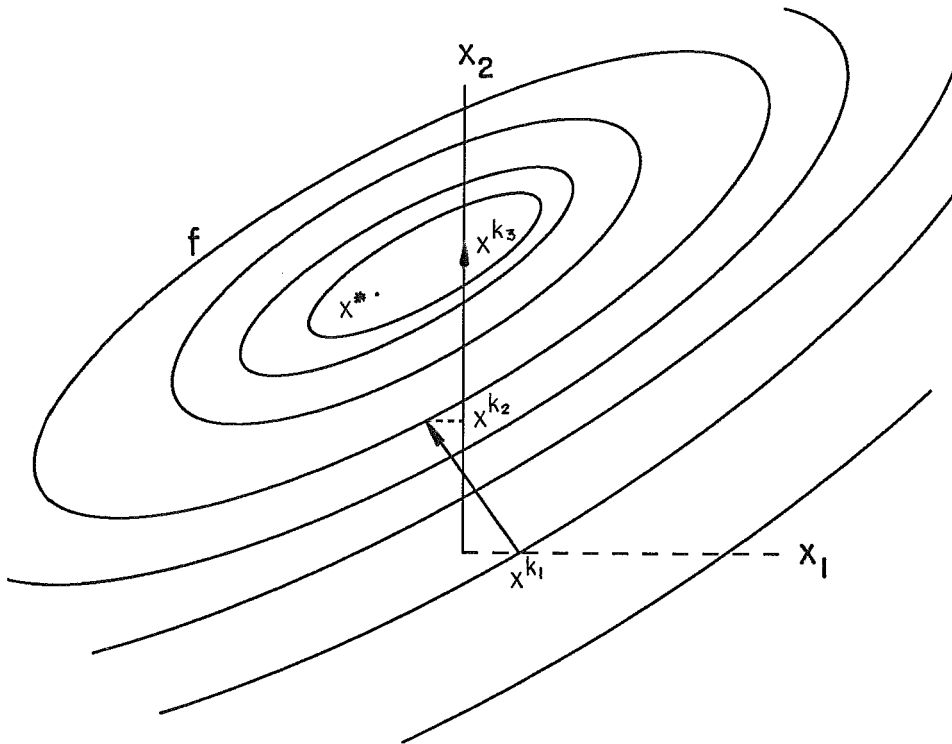


Figura III.4: A minimização sobre um conjunto ativo.

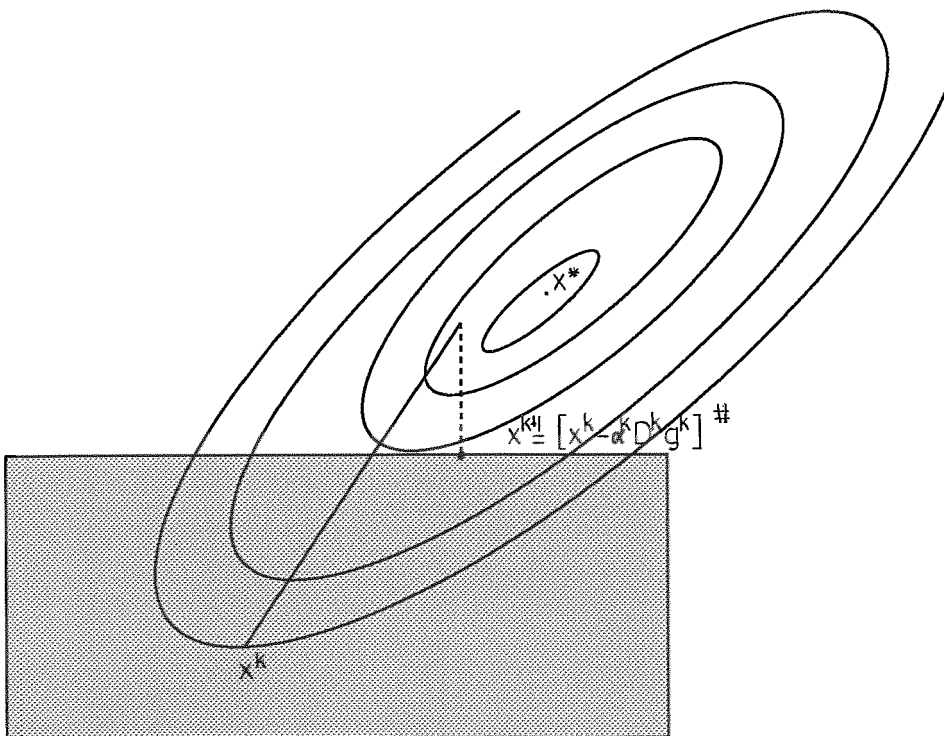


Figura III.5: O próximo ponto no método de gradiente projetado.

Capítulo IV

Introdução ao Método de Regiões de Confiança Elipsoidais com Variáveis Canalizadas

IV.1 Introdução

Considere o problema

$$(PR1) \quad \min f(x) \\ \text{s. a } l \leq x \leq u$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função não linear, $f(\cdot) \in C^2$, e os vetores l e u são conhecidos com $l \leq u$. Sejam os conjuntos $S_1 = \{x \in \mathbb{R}^n \mid l \leq x \leq u\}$, dito conjunto viável, e $S_1^0 = \{x \in \mathbb{R}^n \mid l < x < u\}$, dito conjunto interior de S_1 . Chamamos o problema $(PR1)$ de problema não linear com variáveis canalizadas ou com restrições simples. Iremos resolvê-lo usando um método de ponto interior. Nosso algoritmo de minimização usa o método de regiões de confiança elipsoidais centradas em pontos no interior da região viável. Mais uma vez iremos minimizar o modelo quadrático da função não linear $f(\cdot)$, $q(d)$, expresso por

$$q(d) = f(x) + \nabla f(x)^t d + \frac{1}{2} d^t \nabla^2 f(x) d, \quad (IV.1)$$

com o objetivo de encontrar, com repetidas tentativas, um ponto de mínimo local da função não linear $f(\cdot)$ em S_1^0 .

Usaremos o algoritmo de Levenberg-Marquardt (ver capítulo II) adaptado ao nosso método de região de confiança com formato elipsoidal levando

em conta que nossas variáveis são do tipo canalizadas. Acrescentamos ao método de Levenberg-Marquardt um algoritmo para o tratamento de variáveis ativas feito por Sagara[23], que resolve problemas de mínimos quadrados com restrições simples. Esse algoritmo também sofreu alguns ajustes para ficar de acordo com o nosso problema original. A convergência global de um algoritmo semelhante ao nosso foi demonstrada por Sagara[23], para problemas de mínimos quadrados. Para problemas gerais, admitindo-se funções não convexas, não há ainda uma demonstração de convergência. O estudo de restrições simples é um caso particular do estudo de problemas de programação não linear com restrições lineares, realizado por Gonzaga[11]. Nessa referência indicam-se resultados de convergência bastante gerais, mas ainda sem demonstrações completas.

IV.2 A Região de Confiança

A aproximação quadrática da função não linear $f(\cdot)$ será minimizada sobre um elipsóide simples dentro da região viável determinada pelo vínculo

$$l \leq x \leq u.$$

Entende-se por um elipsóide simples um elipsóide cujos eixos são paralelos aos eixos coordenados, como mostra a figura IV.1.

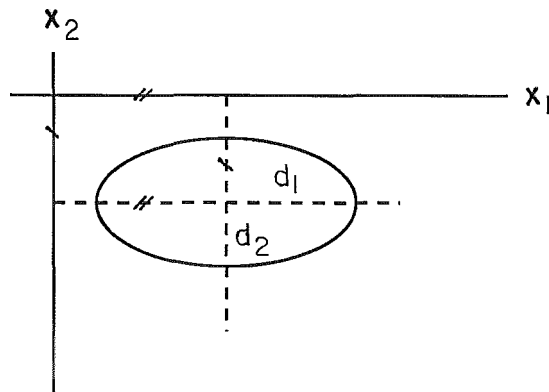


Figura IV.1: Um elipsóide simples

O elipsóide simples é representado pela equação

$$d^t S^2 d = 1 \text{ ou } \|Sd\| = 1$$

onde S é uma matriz diagonal cujos elementos são

$$[s]_{ii} = \frac{1}{\bar{d}_i}, \bar{d}_i > 0, \forall i = 1, \dots, n,$$

onde \bar{d}_i é o tamanho do eixo do elipsóide paralelo ao eixo da variável x_i .

O formato da região de confiança do algoritmo na iteração k é determinado pelo maior elipsóide simples centrado em x^k dentro da região viável. Como a região viável está determinada pelo vínculo $l \leq x \leq u$, que geometricamente representa uma 'caixa' no caso do problema ser de dimensão 3 ou um retângulo no caso de ser bi-dimensional, então o elipsóide será construído achando

$$\bar{d}_i^k = \min\{x_i^k - l_i, u_i - x_i^k\}.$$

A região em torno de x^k definida por $\|Sd\| \leq 1$ define o maior elipsóide simples centrado em x^k e inteiramente contido na região viável.

Ilustramos, na figura IV.2, a construção de elipsóides simples $\|Sd\| \leq h$, com $h = 0.5, 1, 2$, centrado em x^k para um problema bi-dimensional.

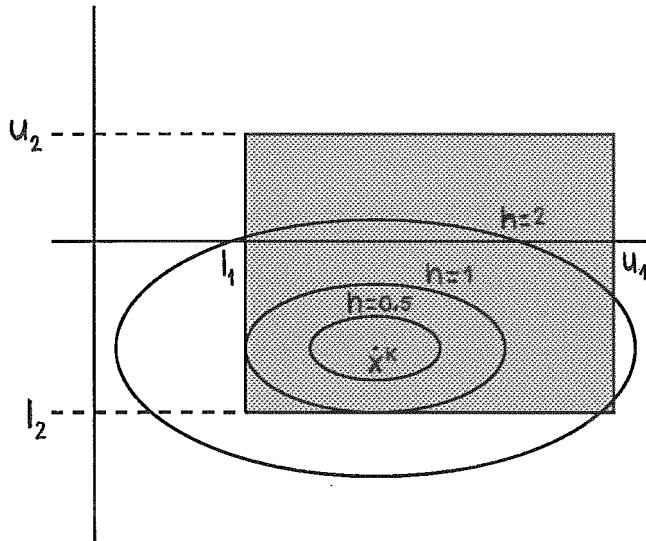


Figura IV.2: Elipsóides simples de centro x^k com $h = 0.5, 1, 2$.

IV.3 O Mínimo da Função Quadrática no Elipsóide

O algoritmo efetivamente resolve a cada iteração

$$\begin{aligned} \min q^k(d) & \tag{IV.2} \\ \text{s. a } d^t S^2 d & \leq h^{k^2} \end{aligned}$$

onde $q^k(\cdot)$ é a aproximação quadrática da função objetivo $f(\cdot)$ e $d^t S^2 d \leq h^{k^2}$ é a região de confiança com formato adaptado ao conjunto viável do problema original (PR1). Se fizermos $h^k < 1$, o resultado de IV.2 será sempre um ponto interior. No entanto, essa limitação não é conveniente em geral: basta garantir que o resultado de IV.2 seja interior, o que pode ocorrer com $h^k \gg 1$. A escolha da região de confiança será estudada na próxima seção.

As condições de otimalidade para este problema IV.2 são as condições de Karush-Kuhn-Tucker já apresentadas no capítulo II. Definindo

$$c(d) = d^t S^2 d - (h^k)^2,$$

vamos apresentar o seguinte teorema:

Teorema IV.1 Se \hat{d} é uma solução Ótima para o problema IV.2, então existe o escalar μ tal que

$$\begin{aligned} \nabla q^k(\hat{d}) + \mu \nabla c(\hat{d}) &= 0, & \tag{IV.3} \\ \mu &\geq 0 \text{ e} \\ \mu c(\hat{d}) &= 0, \end{aligned}$$

para \hat{d} viável. (Veja Teorema K. K. T. em [1].)

O gradiente da aproximação quadrática, apresentada em IV.1, é

$$\nabla q^k(d) = g^k + H^k d, \tag{IV.4}$$

e o gradiente da função $c(d)$ é

$$\nabla c(d) = 2S^2 d. \tag{IV.5}$$

Introduzindo IV.4 e IV.5 em IV.3 obtemos

$$\begin{aligned} (H^k + \mu S^2) \hat{d} &= -g^k & \tag{IV.6} \\ \mu (\hat{d}^t S^2 \hat{d} - h^{k^2}) &= 0 \end{aligned}$$

Concluindo, teremos que resolver o sistema expresso em IV.6 para encontrarmos a solução global \hat{d} em $q^k(\cdot)$ sujeito a $d^t S^2 d \leq h^{k^2}$.

IV.4 O Algoritmo de Minimização de Funções não Lineares usando Método de Região de Confiança Elipsoidal com Variáveis Canalizadas

A condição de otimalidade desenvolvida acima estende o resultado do teorema 11.1 do cap. II para o caso de regiões não esféricas, e sugere o uso do algoritmo de Levenberg-Marquardt que resolve o subproblema do método de região de confiança. Nosso subproblema está determinado por IV.2. Porém o algoritmo de Levenberg-Marquardt minimiza funções não lineares sem restrições. Algumas modificações serão feitas no sentido de que usemos tal algoritmo.

O Tamanho do Passo

Em geral, o tamanho do passo no método de região de confiança está determinado antes de escolhermos a direção do deslocamento em uma iteração. Usando o algoritmo de Levenberg-Marquardt o tamanho do passo fica implicitamente definido por μ .

A Direção do Passo

O método de Levenberg-Marquardt, fundamentado em IV.6, nos faz crer que para um certo $\mu > 0$ tal que $(H^k + \mu S^2)$ seja definida positiva encontraremos d^k satisfazendo $(H^k + \mu S^2)d^k = -g^k$. O próximo ponto será

$$x^{k+1} = x^k + d^k$$

somente se a aproximação entre a função quadrática e a função não linear for considerada 'boa' pelo teste feito através dos valores da razão r^k . (Veja cap. II.) Como no método de Levenberg-Marquardt para o caso desvinculado, cada iteração vai partir de um valor μ^k e obter um passo d^k satisfazendo $(H^k + \mu^k S^2)d^k = -g^k$. Agora, a adequação de d^k deverá ser testada para satisfazer dois requisitos: viabilidade e precisão da aproximação quadrática. A figura IV.3 mostra a variação de d^k sobre

a trajetória $d(\mu)$ quando μ varia continuamente (compare com a figura 11.6 no cap. II).

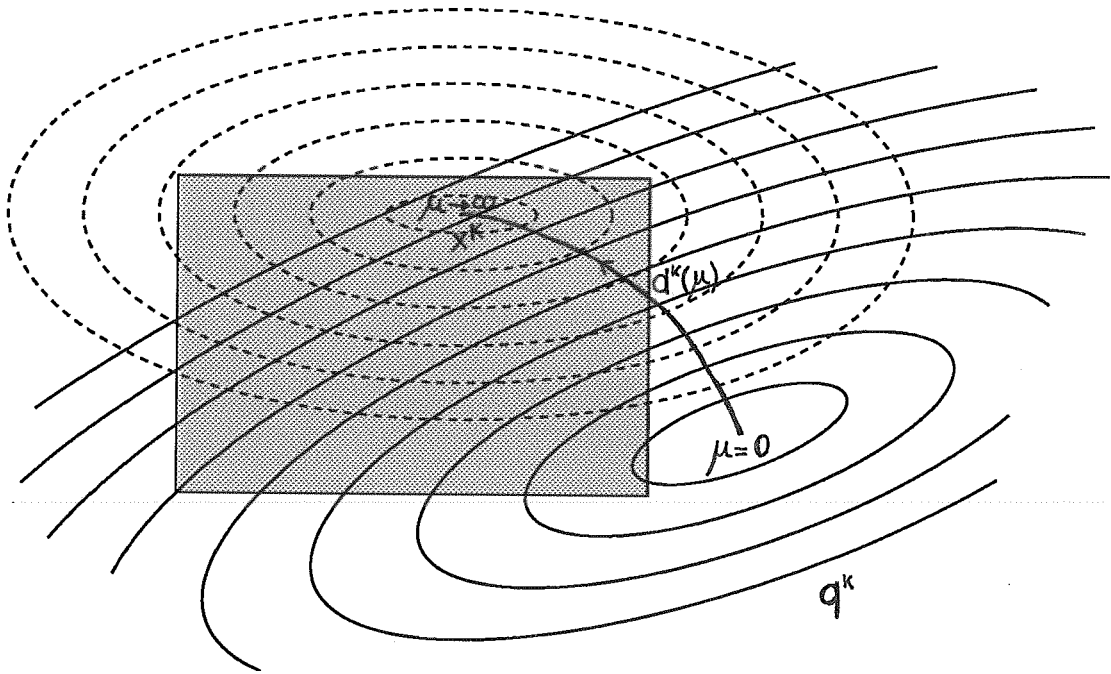


Figura IV.3: A trajetória do método de regiões de confiança elipsoidais para $\mu \in [0, \infty]$.

O Ponto Resultante na Iteração k Pode Não Ser Viável

Dado x^k na iteração k conhecemos de imediato o maior elipsóide centrado em x^k dentro da região viável, o qual chamaremos de E^k . O valor de μ^k definido na iteração anterior pode gerar um elipsóide tal que alguma componente de $x^k + d^k$ esteja fora de seus limites quando minimizamos a quadrática sobre o elipsóide. Observe a figura IV.4.

Nesse caso, para contornarmos tal dificuldade no método de Levenberg-Marquardt, preparamos o nosso algoritmo de maneira que ele ache o ponto de mínimo da quadrática sobre o elipsóide e depois teste se o próximo ponto é viável. Sendo viável, passamos então para o teste r^k , senão fazemos duas considerações.

Definindo a máxima violação como

$$t = \max_{i=1, \dots, n} \max \left\{ \frac{d_i^k}{u_i - x_i^k}, \frac{-d_i^k}{x_i^k - l_i} \right\},$$

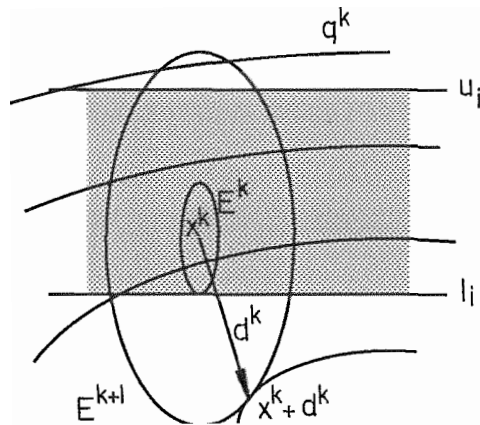


Figura IV.4: O próximo ponto pode não ser viável.

vamos calcular t e executar os seguintes testes:

- a. se $t < 1$, então $x^k + d^k$ é ponto interior;
- b. se $t \in [1, 1.5]$, então reduzimos $\|d^k\|$ mantendo a direção de modo a obrigar $x^k + d^k$ a ser viável;
- c. se $t > 1.5$, então rejeitamos o passo e aumentamos μ^k de maneira a reduzir o raio da região de confiança, ou seja, a fazer o tamanho do elipsóide centrado em x^k diminuir e resolvemos novamente o sistema $(H^k + \mu^k S^2)d^k = -g^k$ a fim de obtermos outro ponto. Na verdade, estamos fazendo, com isto, que o ponto seja deslocado ao longo da trajetória da figura IV.3 até se tornar um ponto viável.

O teste de viabilidade é feito para $x^k + t d^k$ dando nos a garantia de $x^k + d^k$ ser um ponto interior. Vamos apresentar a seguir o algoritmo propriamente dito.

Algoritmo IV.1 Algoritmo do Método de Regiões de Confiança Elipsoidais com Variáveis Canalizadas: Dados μ^0, l, u, x^0 .

$k = 0$

Repita

calcule g^k e H^k ;

ache o elipsóide simples centrado em x^k segundo os limites impostos por

l e u ;

faça, caso necessário, $(H^k + \mu^k S^{2k})$ ser definida positiva aumentando μ^k ;
determine o deslocamento d^k resolvendo o sistema

$$(H^k + \mu^k S^{2k})d^k = -g^k;$$

determine $x^{k+1} = x^k + d^k$ viável testando a viabilidade para $x^k + l.l d^k$,
enquanto $x^k + l.l d^k$ não for viável, faça:

determine a maior violação calculando t

$$t = \max_{i=1, \dots, n} \max \left\{ \frac{1.l d_i^k}{u_i - x_i^k}, \frac{-1.l d_i^k}{x_i^k - l_i} \right\};$$

se $t < 1$ então $x^k + l.l d^k$ é viável;

se $t \in [1, 1.5]$ então $x^k + l.l d^k$ não é viável, faça $d^k = d^k/t$;

se $t > 1.5$ então $x^k + l.l d^k$ não é viável, faça $\mu^k = 4\mu^k$, e ache
novo d^k resolvendo novamente o sistema

$$(H^k + \mu^k S^{2k})d^k = -g^k;$$

avaliar $f(x^k + d^k)$ e r^k ;

teste r^k segundo a heurística :

se $r^k < 0.25$ então faça $\mu^k = 4\mu^k$;

senão se $r^k \geq 0.75$ então faça $\mu^k = \mu^k/2$;

se $r^k \geq 0.25$ então $x^{k+1} = x^k + d^k$;

senão $x^{k+1} = x^k$;

$k = k + 1$

Até convergir.

O Conjunto Ativo

Frequentemente as soluções dos problemas como em (PR1) estão na fronteira da região viável. O algoritmo constrói regiões de confiança elipsoidais que se tornarão cada vez mais finas, a medida que o algoritmo vai gerando pontos de mínimo, causando grande instabilidade numérica. Veja a figura IV.5.

O conjunto ativo é criado para sobrepor tal dificuldade aliado ao fato de que o nosso método é interior.

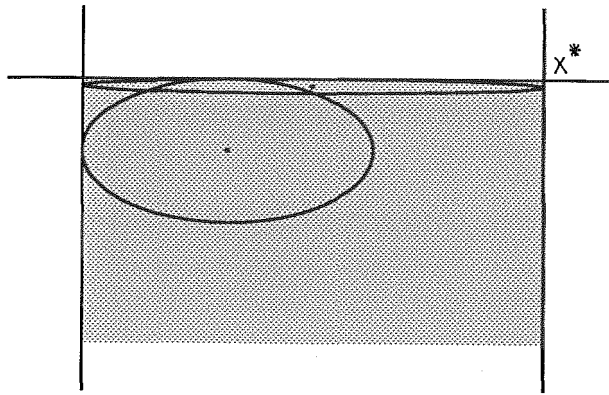


Figura IV.5: Um elipsóide mal-condicionado.

Vamos definir para uma iteração k os seguintes conjuntos ε -ativos, para um dado $\varepsilon > 0$:

$$A_u^\varepsilon = \{i \mid x_i^k > u_i - \varepsilon\}$$

e

$$A_l^\varepsilon = \{i \mid x_i^k < l_i + \varepsilon\}.$$

Definimos o conjunto ativo como

$$A'' = A_l^\varepsilon \cup A_u^\varepsilon$$

na iteração k . Diz-se que x_i é ativa se $i \in A^\varepsilon$. Como exemplo, no caso bidimensional, na figura IV.6, se \mathbf{x}^k apresenta alguma componente na área hachurada, o conjunto ativo deixa de ser vazio na iteração k .

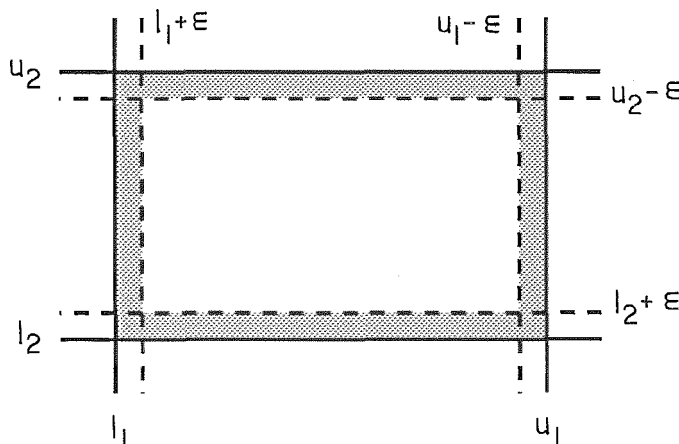


Figura IV.6: Representação dos pontos ε -perto da fronteira do conjunto viável.

Conhecendo o subproblema do método de região de confiança IV.2 expresso por

$$\begin{aligned} \min \quad & q^k(d) \\ \text{s. a} \quad & d^T S^2 d \leq h^{k2}, \end{aligned}$$

a introdução da técnica de conjuntos ativos propõe o mesmo subproblema com mais um vínculo – o passo na iteração k tem componentes nulas para as variáveis ativas. Traduzindo, temos um novo subproblema

$$\begin{aligned} & \min q^k(d) && \text{(IV.7)} \\ \text{s. a} & d^t S^2 d \leq h^{k^2} \\ & d_i = 0 \quad \forall i \in A^\varepsilon \end{aligned}$$

No subespaço gerado pelas variáveis que não são ativas o elipsóide não será mal-condicionado, pois todos os eixos têm comprimento maior ou igual a ε , e, assim, podemos continuar com o algoritmo a fim de melhorar a sua solução.

Ponto E-Ótimo

Supondo que as seguintes condições sejam satisfeitas para x^k na iteração k :

a.

$$|g_i^k| < \tilde{\varepsilon} \quad \forall i \notin A^\varepsilon;$$

b.

$$g_i^k > -\tilde{\varepsilon} \quad \forall i \in A_l^\varepsilon;$$

c.

$$g_i^k < \tilde{\varepsilon} \quad \forall i \in A_u^\varepsilon;$$

onde $\tilde{\varepsilon} > 0$ é dado com $\varepsilon > \tilde{\varepsilon}$. Neste caso, podemos dizer que x^k é um ponto e-ótimo, ou seja, x^k pode ser considerado como a solução aproximada do nosso problema de minimização (PR1). De fato, ao longo de qualquer direção viável d que se afaste das restrições ε -ativas, teremos

$$g^{k,t} \frac{h}{\|h\|} \leq n\tilde{\varepsilon},$$

o que corresponde à condição

$$\|g^k\|_\infty \leq \tilde{\varepsilon}$$

para o caso desvinculado. No caso da condição a. não ser satisfeita, podemos prosseguir com o algoritmo resolvendo o subproblema IV.7, o qual está bem condicionado. Contudo, se a condição a. é satisfeita, porém se b. ou c. não são,

podemos, ainda, assim, melhorar o ponto de mínimo na iteração. Para isso, relaxamos a restrição no subproblema IV.7 retirando um índice i^* do conjunto ativo que causa violação da condição $b.$ ou $c.$.

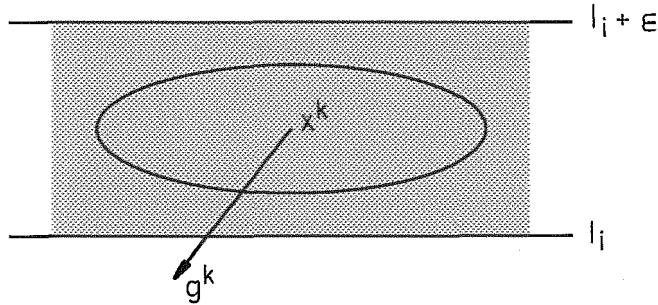


Figura IV.7: Podemos tornar livre a variável x_i^k ativa.

Na figura IV.7, apresentamos um exemplo que evidencia que apesar da variável x_i^k estar fixa, pois temos i no conjunto ativo, podemos torná-la livre uma vez que a direção que esta componente tomará será contrária àquela que causaria instabilidade ao algoritmo.

Como relaxar a restrição referente ao conjunto ativo na iteração k ? escolhendo o índice i^* da variável $x_{i^*}^k$ em A'' tal que

$$|g_{i^*}^k| = \max\{|g_i^k| \mid (i \in A_l^\varepsilon \text{ e } g_i^k < 0) \text{ ou } (i \in A_u^\varepsilon \text{ e } g_i^k > 0)\}. \quad (\text{IV.8})$$

Vejam os porquês. Se $b.$ ou $c.$ não são satisfeitas, significa que

$$\exists i \in A_l^\varepsilon \mid g_i^k \leq -\tilde{\varepsilon}$$

ou

$$\exists i \in A'' \mid g_i^k \geq \tilde{\varepsilon},$$

que implica em

$$|g_i^k| \geq \tilde{\varepsilon} \quad \forall i \in A_l^\varepsilon \cup A_u^\varepsilon = A''.$$

Nesse caso iremos escolher i^* que satisfaz IV.8. Novamente resolvemos o problema IV.7 para o novo conjunto ativo na iteração k com a certeza que estaremos melhorando a solução do problema (PR1).

O Algoritmo **para** o Conjunto Ativo

A seguir apresentamos o algoritmo para o tratamento das variáveis ativas.

Algoritmo IV.2 Algoritmo do Conjunto Ativo:

Se x^k não é ε -ótimo, faça:

determine $A_l^\varepsilon, A_u^\varepsilon$ e A'' ;

se (**a.** é satisfeita) e (**b.** ou **c.** não são satisfeitas) então

encontre $i^* = \operatorname{argmax}\{|g_i^k| \mid i \in A''\}$;

faça $A'' = A'' - \{i^*\}$;

resolva o problema IV.7;

Conseguimos, portanto, com esse algoritmo IV.2, tratar as variáveis ativas em um número finito de iterações sem que problemas de mal-condicionamento aconteçam com o nosso algoritmo de minimização.

O Algoritmo Final

Apresentamos o algoritmo do método de regiões de confiança elipsoidais com variáveis canalizadas introduzindo a técnica de conjuntos ativos a seguir:

Algoritmo IV.3 Algoritmo do Método de Regiões de Confiança Elipsoidais com Variáveis Canalizadas com Conjuntos Ativos: Dados $e, \tilde{\varepsilon}, \mu^0, l, u, x^0$.

$k = 0$

Repita

calcule g^k e H^k ;

ache o elipsóide simples centrado em x^k segundo os limites impostos por l e u ;

determine o conjunto ativo em x^k ;

faça, caso necessário, $(H^k + \mu^k S^{2^k})$ ser definida positiva para as variáveis não ativas aumentando μ^k ;

determine o deslocamento d^k :

para a variável x_i^k ativa faça $d_i^k = 0$;

para as variáveis não ativas resolva o sistema

$$(H^k + \mu^k S^{2^k})d^k = -g^k;$$

determine $x^{k+1} = x^k + d^k$ viável testando a viabilidade para $x^k + 1.1d^k$, enquanto $x^k + 1.1d^k$ não for viável, faça:

determine a maior violação calculando t

$$t = \max_{i=1, \dots, n} \max \left\{ \frac{1.1d_i^k}{u_i - x_i^k}, \frac{-1.1d_i^k}{x_i^k - l_i} \right\};$$

se $t < 1$ então $x^k + 1.1d^k$ é viável;

se $t \in [1, 1.5]$ então $x^k + 1.1d^k$ não é viável, faça $d^k = d^k/t$;

se $t > 1.5$ então $x^k + 1.1d^k$ não é viável, faça $\mu^k = 4\mu^k$, e ache novo d^k resolvendo novamente o sistema

$$(H^k + \mu^k S^{2k})d^k = -g^k;$$

avale $f(x^k + d^k)$ e r^k ;

teste r^k segundo a heurística :

se $r^k < 0.25$ então faça $\mu^k = 4\mu^k$;

senão se $r^k \geq 0.75$ então faça $\mu^k = \mu^k/2$;

se $r^k \geq 0.25$ então $x^{k+1} = x^k + d^k$;

senão $x^{k+1} = x^k$;

$k = k + 1$

Até **convergir**.

No próximo capítulo, falaremos melhor a respeito da implementação do teste de convergência do algoritmo que acabamos de descrever.

Apresentamos um exemplo, na figura IV.8, onde mostramos as construções dos elipsóides simples centrados em pontos viáveis que caminham em direção do ponto de mínimo da função $f(\cdot)$.

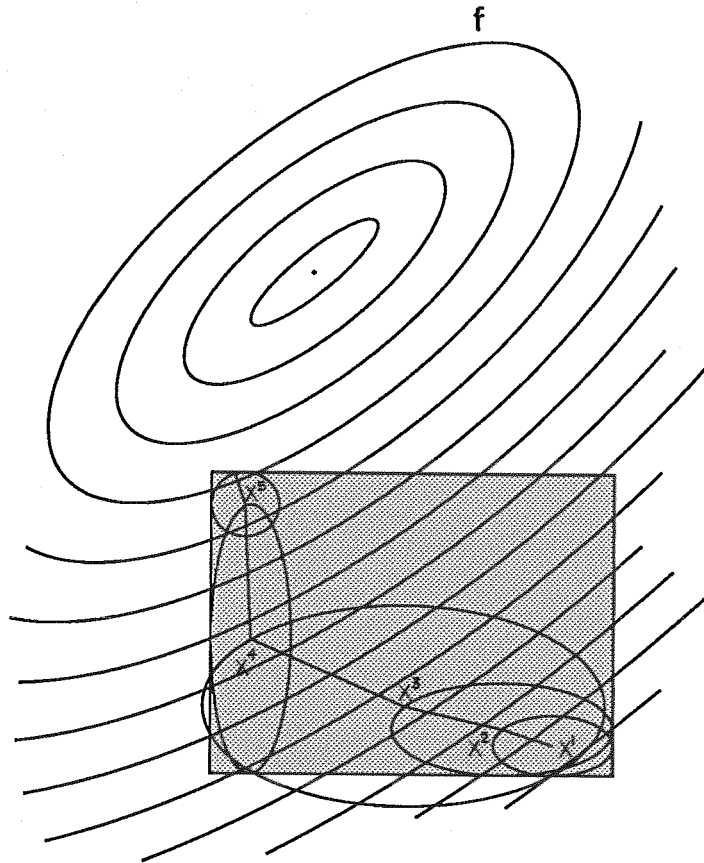


Figura IV.8: Passos do algoritmo de regiões de confiança elipsoidais com variáveis canalizadas para uma função no \mathbb{R}^2 .

Capítulo V

Implementação e Testes

V.1 Introdução

Neste capítulo, vamos abordar a implementação computacional dos algoritmos de minimização com variáveis canalizadas apresentados nos capítulos anteriores. Também, serão mostradas algumas características dos algoritmos implementados e alguns testes realizados para problemas de minimização variados. Finalmente faremos algumas comparações entre os algoritmos.

V.2 Implementação Computacional

A linguagem computacional escolhida foi a linguagem C, por ser uma linguagem moderna, dispor de mais recursos de memória e apresentar maior portabilidade. O equipamento utilizado foi um microcomputador PC-XT. Nosso código computacional certamente não é o mais eficaz, visto que estamos preocupados somente com a eficiência dos algoritmos matemáticos. Contudo, uma elaboração melhor a nível de programação reduziria o tempo de execução de cada iteração dos algoritmos. Foi observado, em especial, que o algoritmo do conjunto ativo IV.2, apresentado no capítulo IV, pode ser aprimorado usando regras heurísticas. Notou-se que certas variáveis ficaram fixas por muitas iterações desnecessariamente, atrasando a resolução do algoritmo. Deixamos isto como sugestão de um futuro trabalho!

V.3 Uma Abordagem Geral dos Programas

Os algoritmos de minimização de funções não lineares com variáveis canalizadas implementados computacionalmente foram:

- método de gradiente projetado,
 - método de região de confiança com formato elipsoidal usando o método de Levenberg-Marquardt, e
- método de máximo declive tendo como base o método de gradiente projetado.

Os programas para estes algoritmos descritos acima foram feitos para resolverem problemas de minimização com dimensão no máximo 20. Essa característica define nossos programas como programas de pequeno porte.

O Programa de Gradiente Projetado

O programa de Gradiente Projetado segue basicamente o algoritmo 111.2 apresentado no capítulo III. Essa implementação foi fácil no sentido de que não houve necessidade de se acrescentar algoritmos externos na sua composição, simplesmente o traduzimos para a nossa linguagem computacional.

O Programa de Região de Confiança com Formato Elipsoidal usando o Método de Levenberg-Marquardt

Esse programa está fundamentado no algoritmo IV.3 do capítulo IV. Ele foi de todos o mais trabalhoso. A primeira dificuldade surgiu na determinação da matriz da condição de otimalidade de K.K.T. - $(H^k + \mu^k S^{2k})$. A propriedade da matriz $(H^k + \mu^k S^{2k})$ ser definida positiva foi averiguada pela não negatividade de seus auto-valores, que, por conseguinte, foram determinados pela decomposição LDL^t da matriz cuja referência foi [9]. Além disso, linhas e colunas da matriz de K.K.T. cujos índices eram iguais aos índices das variáveis ativas deveriam ser eliminadas, uma

vez que para essas variáveis o deslocamento deveria ser nulo. Conseqüentemente, a matriz de otimalidade de K.K.T. poderia apresentar dimensão menor que a matriz de otimalidade do problema de minimização original.

A segunda dificuldade diz respeito a determinação do conjunto ativo das variáveis do problema de minimização. O algoritmo proposto por Sagara[23], apresentado em IV.2 no capítulo IV, foi ligeiramente modificado no sentido de 'enfraquecer' o conjunto das variáveis ativas. Apresentamos, a seguir, a modificação. Considerando λ definido como

$$\lambda = \begin{cases} \infty & \text{se } |A^\varepsilon| = n \\ \frac{\sum_{j \notin A^\varepsilon} |g_j|}{n - |A^\varepsilon|} & \text{caso contrário} \end{cases}$$

$$\text{Se } (i \in A_i^\varepsilon \text{ e } g_i < -2\lambda) \text{ ou } (i \in A_u^\varepsilon \text{ e } g_i > 2\lambda),$$

então fazemos

$$\mathbf{A}'' = \mathbf{A}'' - \{i\}.$$

Outra modificação foi necessária fazer a fim de 'fortalecer' o conjunto ativo: no caso de ser possível melhorar o ponto de mínimo \mathbf{x}^k na iteração k do algoritmo, retirando do conjunto ativo aquele índice i^* descrito em IV.8, então só o fazemos, se

$$\bar{d}_i^k > 10^{-10},$$

onde \bar{d}_i^k é o eixo do maior elipsóide simples centrado em \mathbf{x}^k dentro da região viável paralelo ao eixo da variável x_i na iteração k .

A terceira dificuldade apareceu na determinação do próximo ponto viável calculado pelo algoritmo. O sistema linear $(\mathbf{H}^k + \mu S^{2k})d = -g^k$ foi resolvido invertendo-se a matriz definida positiva $(\mathbf{H}^k + \mu^k S^{2k})$. Na inversão usamos o método de Gauss-Jordan por colunas descrito em [7]. Sendo o nosso problema de pontos interiores, fizemos o teste de viabilidade para $\mathbf{x}^{k+1} = (\mathbf{x}^k + l \cdot \mathbf{l}d^k)$ obedecendo exatamente o algoritmo IV.3 : fazendo o teste da razão para escolher a maior violação entre os limites impostos a cada variável, e escolhendo a maior violação t entre as variáveis. Pelo valor de t , sabemos se o próximo ponto torna-se-á interior apenas reduzindo o deslocamento, dividindo-o pela maior violação, ou se calculando novo deslocamento pela minimização da quadrática em outro elipsóide menor decorrente da quadruplicação do valor de μ^k .

A última dificuldade está relacionada com o teste de convergência do algoritmo realizado a cada iteração. Se todas as variáveis são ativas na iteração, então nosso algoritmo chegou ao seu fim e o resultado, o ponto de mínimo local, é exatamente o último ponto do algoritmo. No caso de se ter todas as variáveis livres na iteração, então, como primeira condição de convergência, sendo g^k o gradiente e $\epsilon > 0$ um número pequeno escolhido, devemos ter

$$\|g^k\|_\infty < \epsilon.$$

Se temos variáveis ativas e não ativas em uma mesma iteração, escolhemos somente as componentes do vetor gradiente cujos índices não são os índices das variáveis ativas e achamos a norma desse novo vetor gradiente reduzido recaindo no teste da primeira condição de parada do algoritmo. Como segunda condição de convergência do algoritmo, supondo τ_f o número de algarismos significativos em f^k e $\theta_k = \tau_f \cdot 10^{-k} + |f^k|$ a acurácia absoluta, estabelecemos que

$$f^{k-1} - f^k < \theta_k$$

deva acontecer. E para a terceira condição de convergência do algoritmo, afirmamos que a desigualdade a seguir deva ser verdadeira

$$\|x^{k-1} - x^k\| < \sqrt{\tau_f}(1 + \|x^k\|).$$

Finalmente, se as três condições são verdadeiras simultaneamente então consideramos x^k o ponto de mínimo local fornecido pelo algoritmo.

Com a intenção de uniformizar os algoritmos implementados, usamos esse teste de convergência, totalizando três condições, nos outros dois algoritmos.

O Programa de Máximo Declive

O programa de máximo declive para funções não lineares com variáveis canalizadas foi feito com o objetivo de servir de instrumento de nossas comparações e conclusões dos testes dos algoritmos de minimização as quais serão apresentadas na seção final deste capítulo. Esse programa segue basicamente o algoritmo de gradiente projetado III.2 do capítulo III excluindo as informações de segunda ordem da função

objetivo. A direção do deslocamento é exatamente a direção oposta ao gradiente o que ocasionou pequenas mudanças no algoritmo de gradiente projetado. A seguir apresentamos o algoritmo de máximo declive.

Algoritmo V.1 Algoritmo de Máximo Declive para Variáveis Canalizadas: *Dados $x^0, l, u, \varepsilon > 0, M, \beta \in (0, 1), \sigma \in (0, 0.5)$.*

$k = 0$

Repita

calcule g^k , e H^k ;

ache ε^k ;

determine o conjunto ativo $I_{\#}^k$;

faça $p^k = -g^k$;

faça a busca de Armijo com relação a $I_{\#}^k$ obtendo α^k ;

avalie $x^{k+1} = [x^k + \alpha^k p^k]_{\#}$;

$k = k + 1$;

Até convergir.

V.4 Os Testes Realizados

Para efeito de comparação realizamos os testes para os seguintes programas

- gradiente projetado,
- região de confiança com formato elipsoidal, e
- máximo declive.

Os Problemas de Minimização

Dentre as referências: Fletcher[6]; Dennis e Schnabel[13]; Gill, Murray e Wright[8]; Hock e Schittkowski[12], preparamos 14 problemas de minimização não lineares com variáveis canalizadas de dimensão variando de 2 a 20. Entre as funções objetivo, estão funções convexas e não convexas. Tivemos o cuidado de escolher o conjunto

viável de cada problema de minimização de modo a não conter o ponto de mínimo global, exceto para o problema (P_{10}). Determinamos limites inferiores e superiores para as variáveis de tal maneira que essas variáveis ao longo das iterações se apropriassem de todas as características possíveis envolvendo a técnica de conjuntos ativos. Por último, temos os pontos iniciais de cada problema interiores aos respectivos conjuntos viáveis.

Na próxima seção, apresentamos os problemas de minimização com seus pontos iniciais, x^0 , e seus pontos de mínimo global, x^* .

Os Testes

Para um mesmo problema de minimização executamos os **3** programas com o mesmo ponto inicial e o mesmo conjunto viável. Os resultados mostramos em um gráfico que apresenta os códigos: f_E , f_B e f_G , que significam os valores das funções objetivo dos pontos de mínimo ao longo das iterações dos algoritmos região de confiança com formato elipsoidal, gradiente projetado e máximo declive respectivamente. Esses mesmos resultados podem ser vistos em tabelas. Como a dimensão das funções-objetivo é multivariada, apresentamos somente para as funções-objetivo de dimensão 2 mais um resultado onde fazemos um gráfico ilustrando suas curvas de nível e os pontos de mínimo do algoritmo de região de confiança com formato elipsoidal.

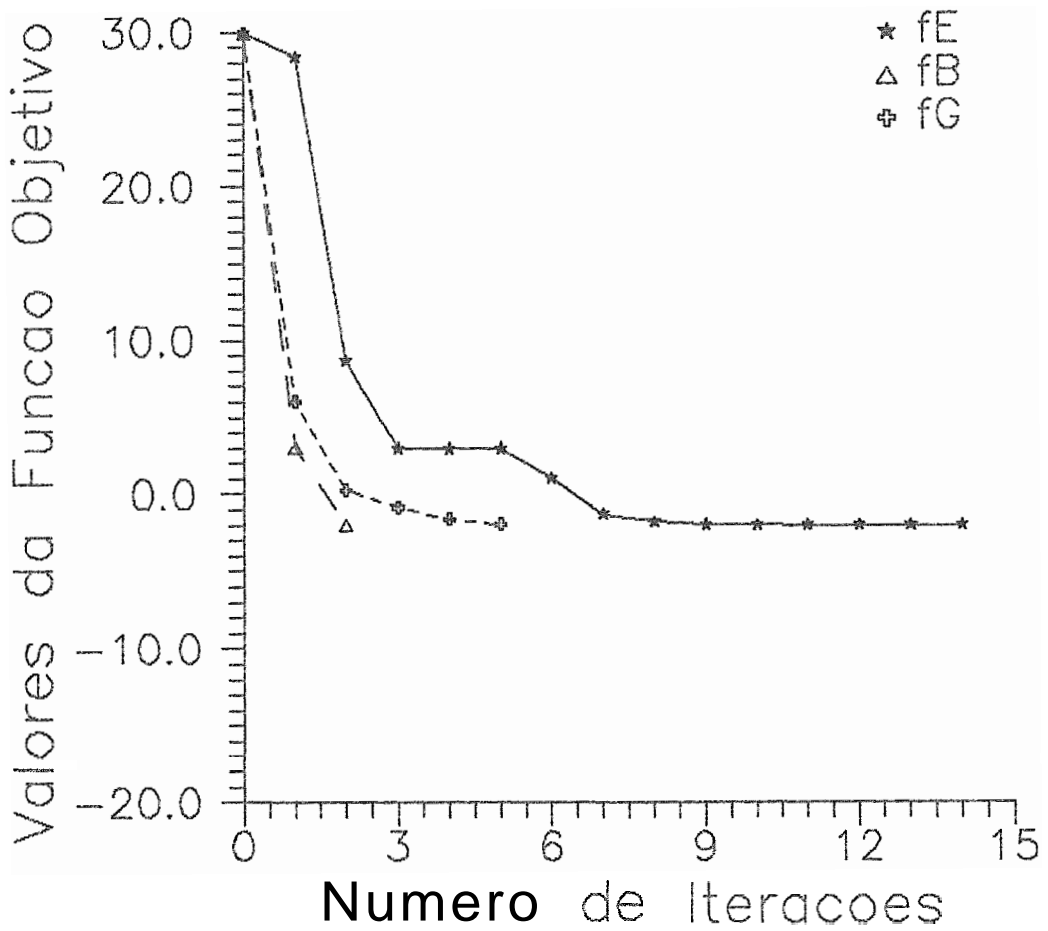
$$(P1) \quad \min \quad f_1 = 2x_1^2 + 3x_2^2 - 4x_1x_2 - 3$$

$$s.a \quad \begin{array}{l} 1 \leq x_1 \leq 3 \\ 1 \leq x_2 \leq 5 \end{array}$$

$$x^0 = (2.995, 4.995)^t$$

$$x^* = (0, 0)^t$$

Problema (P1)



Problema ($P1$)			
k	f_E	f_B	f_G
0	29.950025	29.950025	29.950025
1	3.530425	3.000033	6.000000
2	2.983274	-2.000000	0.288390
3	2.980022		-0.821199
4	2.980017		-1.597773
5	2.980017		-1.926728
6	-1.437252		
7	-1.608919		
8	-1.685928		
9	-1.722602		
10	-1.784742		
11	-1.860803		
12	-1.935926		
13	-1.980999		
14	-1.994355		

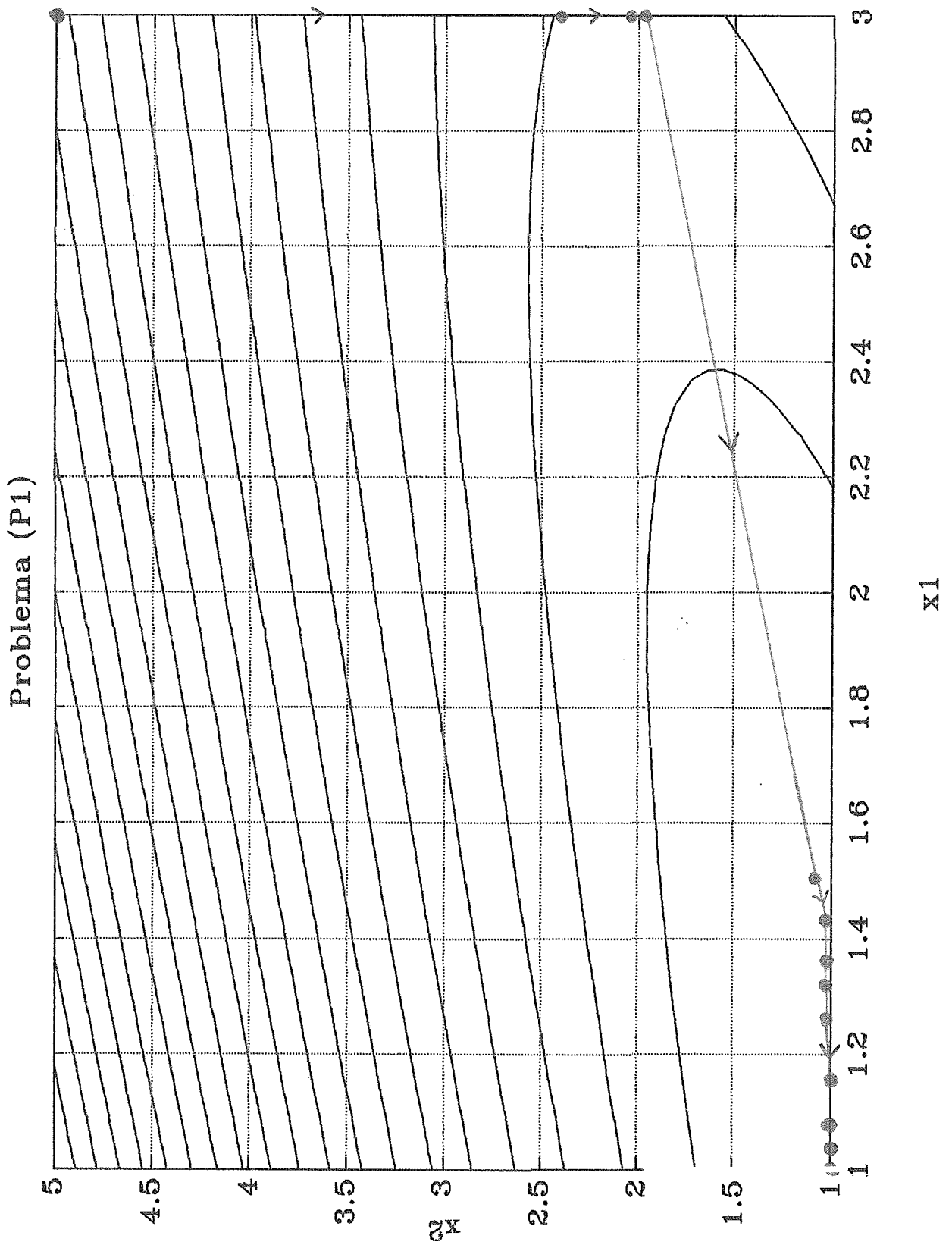


Figura V.1: Curvas de nível da função objetivo de (P1) e os pontos de mínimo do algoritmo região de confiança com formato elipsoidal.

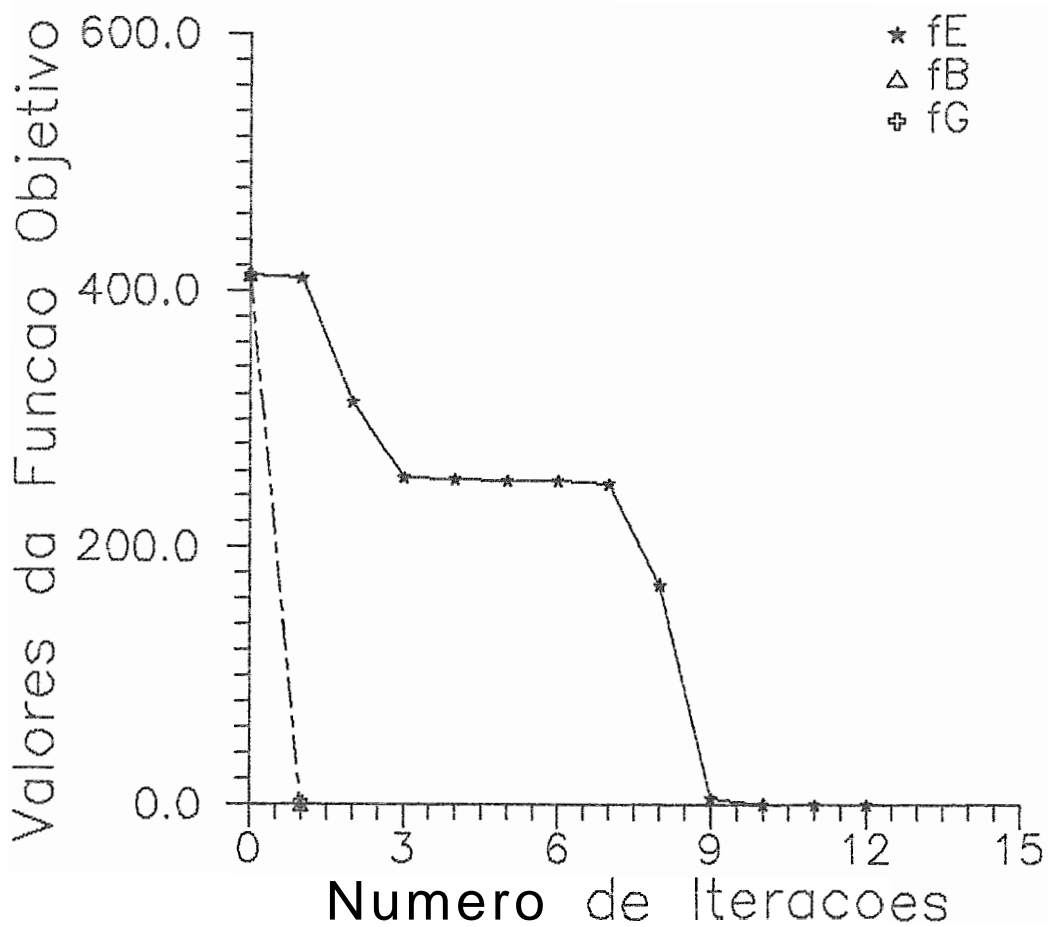
$$(P2) \quad \min \quad f_2 = 2x_1^2 + 4x_2^2 - 4x_1 - 8x_2$$

$$s.a \quad \begin{matrix} -8 & \leq & x_1 & \leq & 0 \\ 2 & \leq & x_2 & \leq & 9 \end{matrix}$$

$$x^0 = (-7.998, 8.997)^t$$

$$x^* = (1, 1)^t$$

Problema (P2)



Problema (P2)			
k	f_E	f_B	f_G
0	411.736044	411.736044	411.736044
1	256.285156	0.000000	2.204593
2	252.686756		2.203993
3	251.881326		2.203393
4	251.849306		
5	251.833452		
6	244.022982		
7	229.220556		
8	202.592737		
9	159.239102		
10	100.482175		
11	42.450642		
12	7.631722		
13	1.183774		
14	0.594758		
15	0.322024		
16	0.190837		
17	0.126507		
18	0.034554		

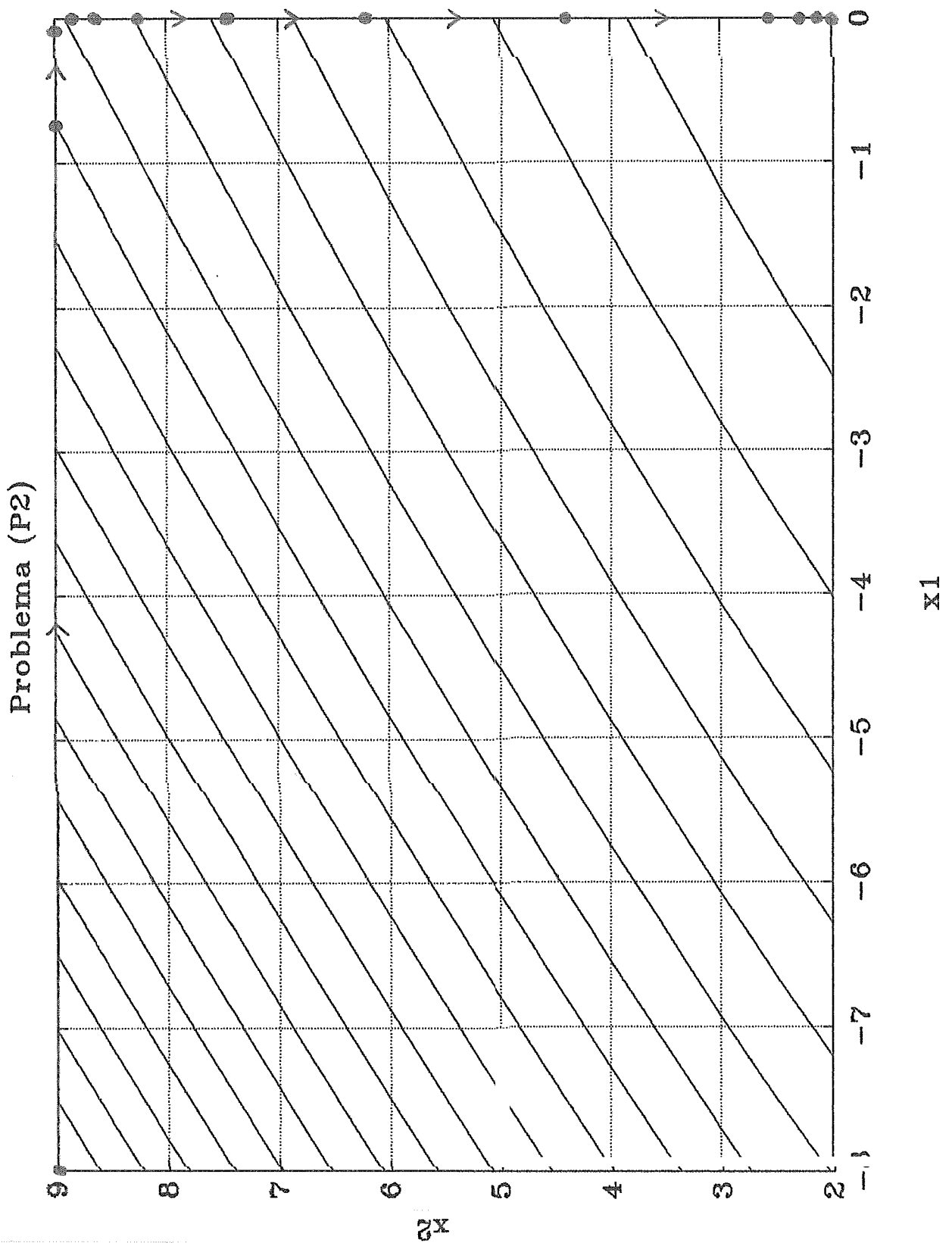
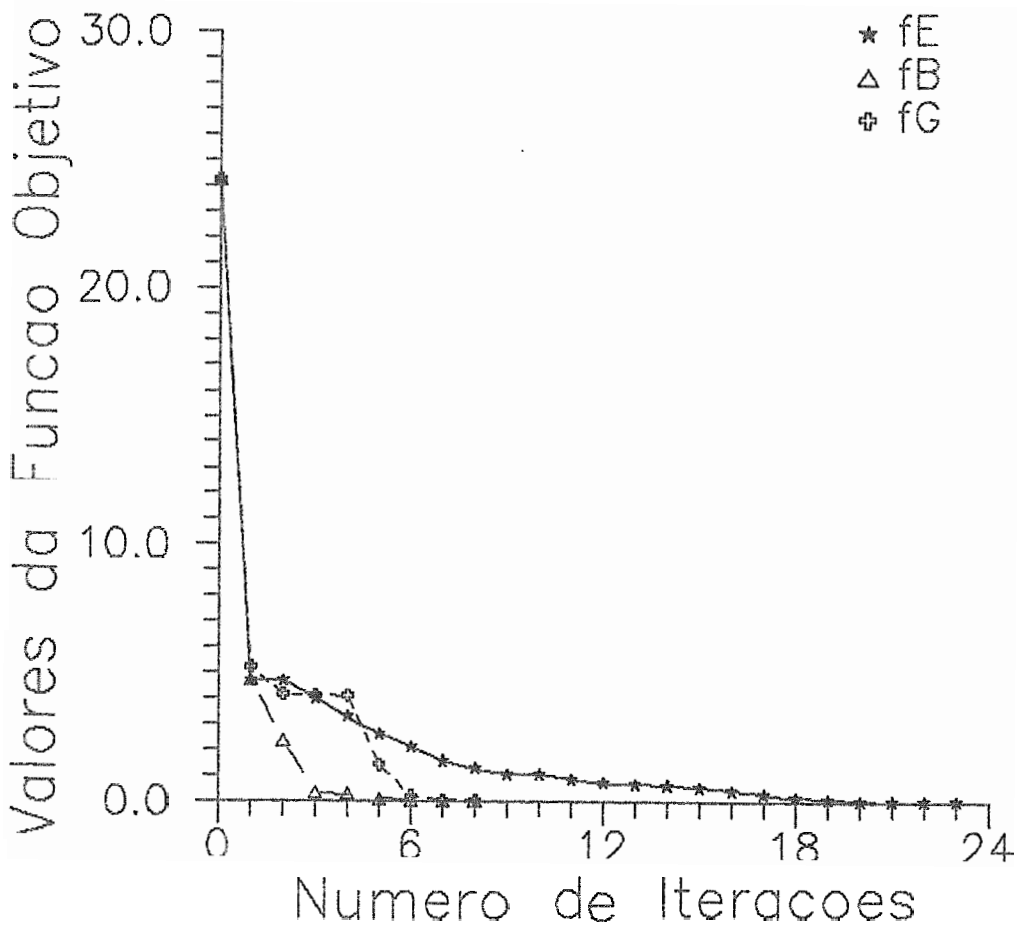


Figura V.2: Curvas de nível da função objetivo de (P2) e os pontos de mínimo do algoritmo região de confiança com formato elipsoidal.

$$\begin{aligned}
 (P3) \quad \min \quad & f_3 = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \\
 \text{s.a} \quad & -2 \leq x_1 \leq 0.8 \\
 & 0 \leq x_2 \leq 2
 \end{aligned}$$

$$\begin{aligned}
 x^0 &= (-1.2, 1)^t \\
 x^* &= (1, 1)^t
 \end{aligned}$$

Problema (P3)



Problema (P3)			
k	f_E	f_B	f_G
0	24.200000	24.200000	24.200000
1	4.711776	4.731884	5.245890
2	4.711776	2.376827	4.179193
3	4.066688	0.363833	4.118821
4	3.215388	0.287685	4.110282
5	3.215388	0.135409	1.446880
6	2.661409	0.048672	0.220461
7	2.128956	0.044819	0.072339
8	1.654326	0.040000	0.065065
9	1.311095		0.064527
10	1.066478		0.064226
11	0.799784		0.063984
12	0.519752		0.063738
13	0.386595		0.063494
14	0.261380		0.063252
15	0.178057		0.063011
16	0.101768		0.062773
17	0.059206		0.062568
18	0.043531		0.062276
19	0.043394		0.043867
20			0.040126

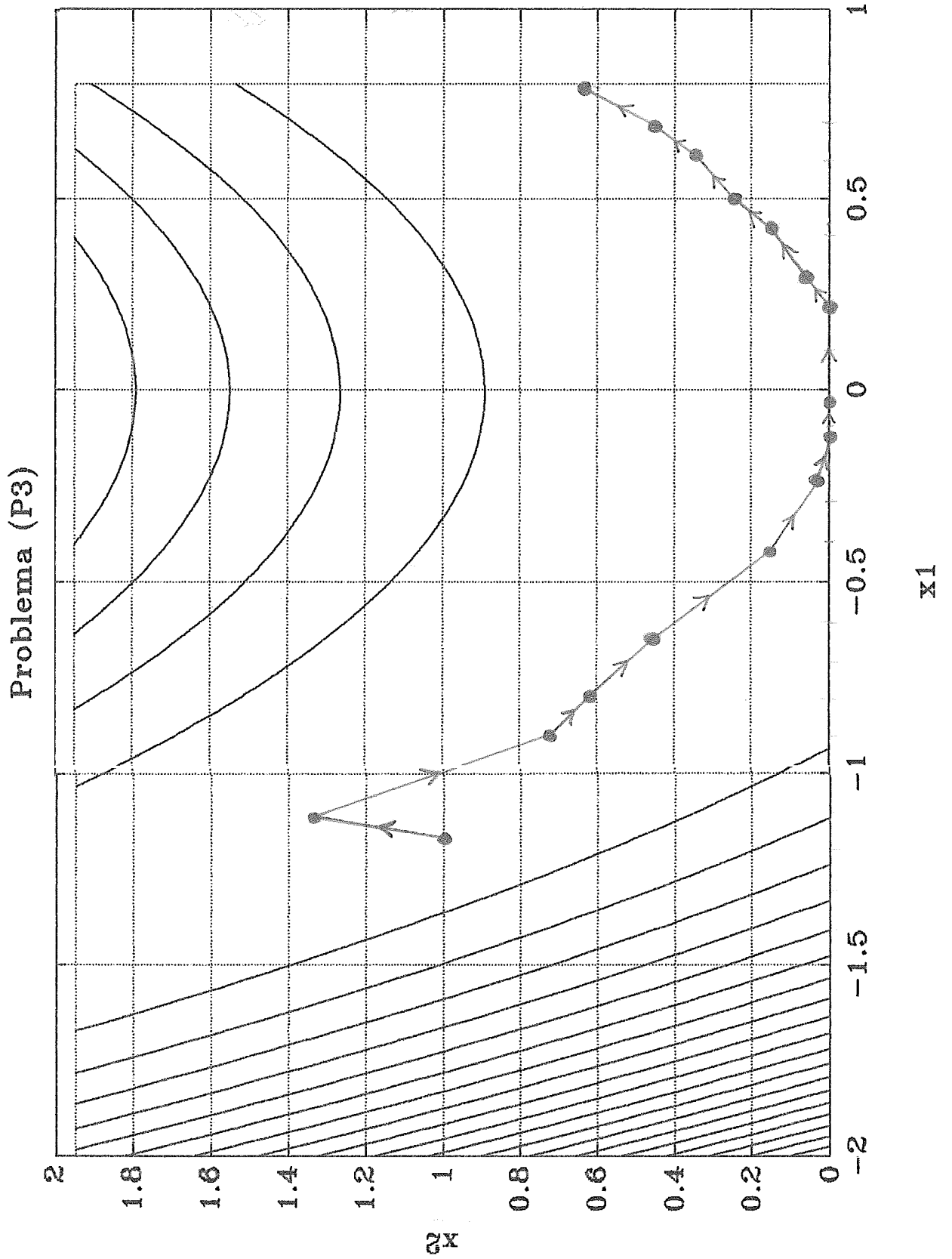
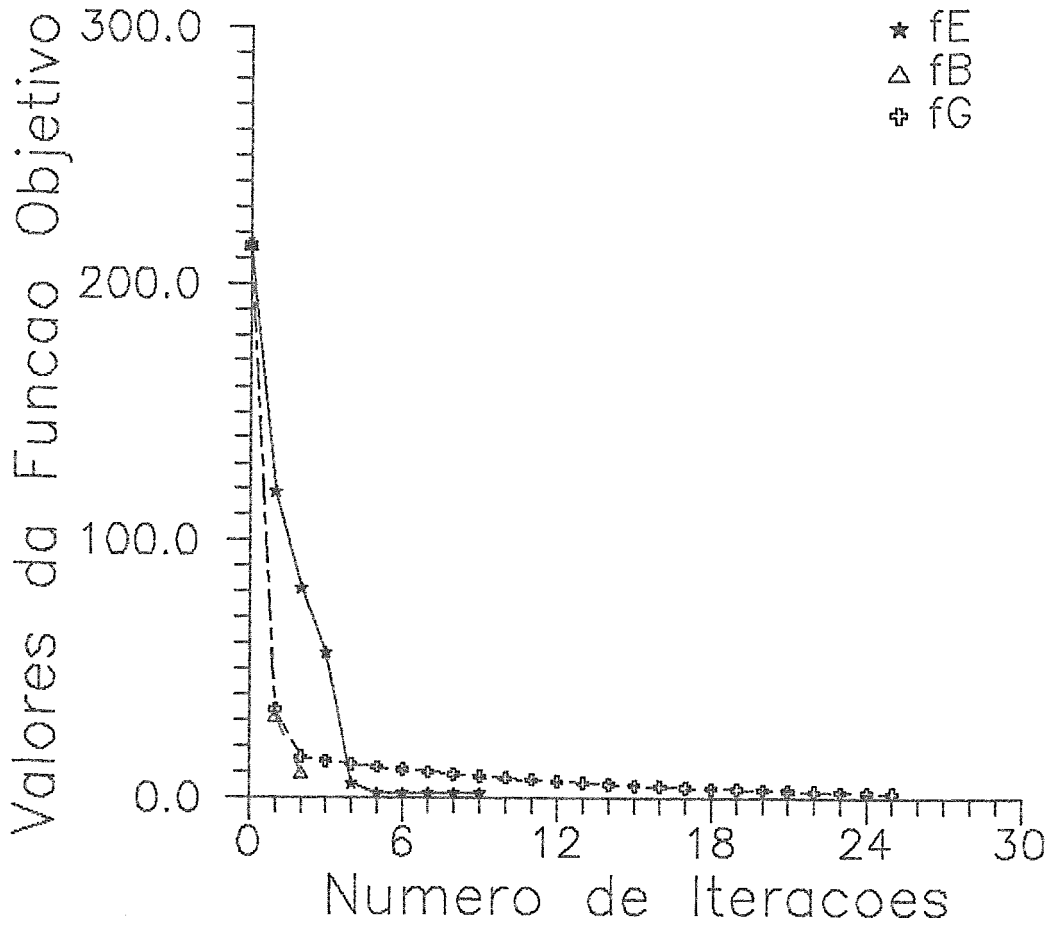


Figura V.3: Curvas de nível da função objetivo de (P3) e os pontos de mínimo do algoritmo região de confiança com formato elipsoidal.

$$\begin{aligned}
 (P4) \quad \min \quad & f_4 = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4 \\
 \text{s.a} \quad & \begin{aligned}
 1 & \leq x_1 \leq 4 \\
 -1.001 & \leq x_2 \leq 2 \\
 -1 & \leq x_3 \leq 0.01 \\
 0 & \leq x_4 \leq 2
 \end{aligned} \\
 & x^0 = (3, -1, 0, 1)^t \\
 & x^* = (0, 0, 0, 0)^t
 \end{aligned}$$

Problema (P4)



Problema ($P4$)			
k	f_E	f_B	f_G
0	215.000000	215.000000	215.000000
1	140.501588	31.825697	34.222954
2	100.271315	9.613382	15.949126
3	82.969342	9.613154	14.509386
4	27.455218		13.228867
5	14.782414		12.074746
6	10.581100		11.179081
7	6.457772		10.168560
8	2.755999		9.268193
9	2.004502		8.459297
10	1.972219		7.732797
11	1.953863		7.158448
12	1.929672		6.523262
13	1.907290		5.954540
14	1.895717		5.444954
15	1.893088		4.985722
16	1.892869		4.613557
17	1.892863		4.212549
18			3.853995
19			3.531456
20			3.280253
21			2.996590
22			2.743428
23			2.516059
24			2.311293
25			2.144416
26			2.143363

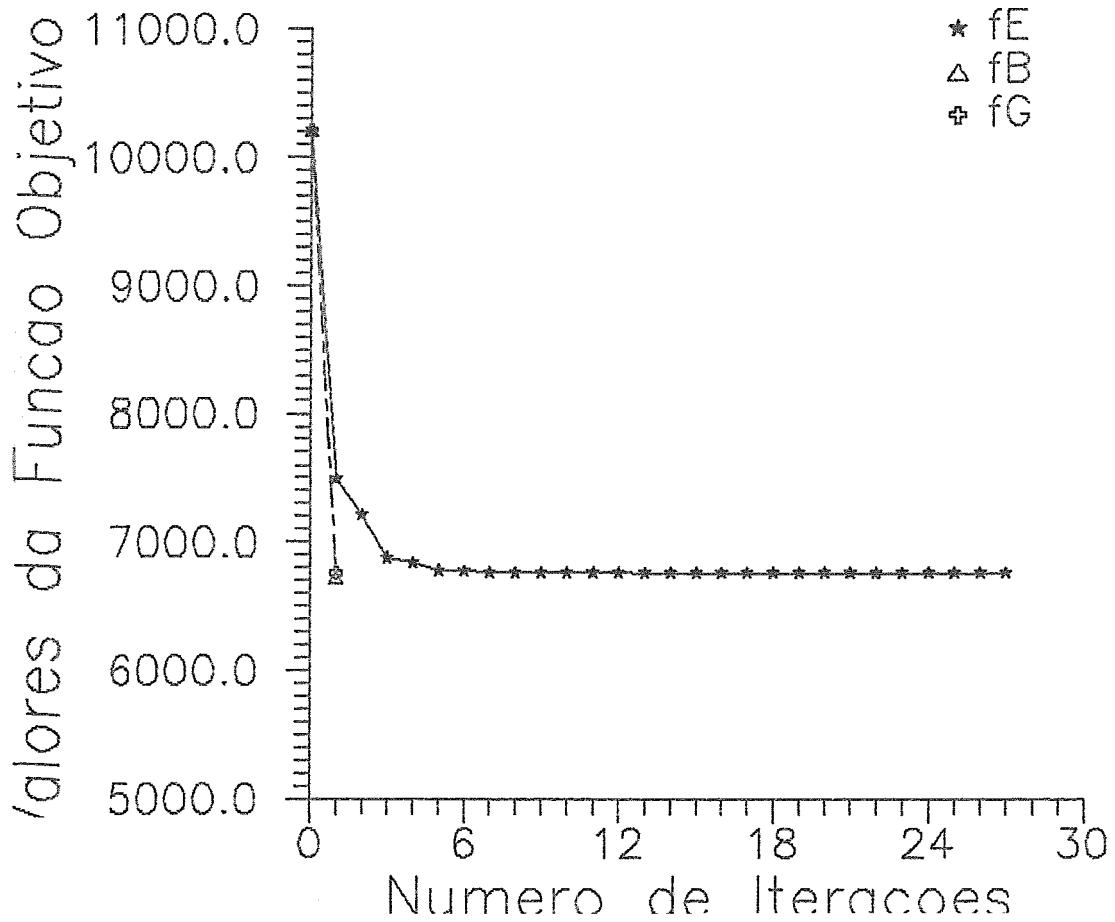
$$(P5) \min f_5 = x_1^4 + x_1^2 + x_2^2 + 5$$

$$s.a \quad \begin{array}{l} 9.005 \leq x_1 \leq 12 \\ -10.008 \leq x_2 \leq -8 \end{array}$$

$$x^0 = (10, -10)^t$$

$$x^* = (0, 0)^t$$

Problema (P5)



Problema (<i>P5</i>)			
<i>k</i>	f_E	f_B	f_G
0	10205.000000	10205.000000	10205.000000
1	7483.595123	6725.682180	6761.428810
2	7189.752853		6761.421692
3	6799.772811		6761.414574
4	6783.203571		6761.407457
5	6774.938202		
6	6762.886454		
7	6762.371555		
8	6762.114124		
9	6761.721446		
10	6761.705359		
11	6761.697315		
12	6761.683555		
13	6761.683053		
14	6761.682259		
15	6761.682227		
16	6761.682184		
17	6761.682182		
18	6761.682181		
19	6761.682180		

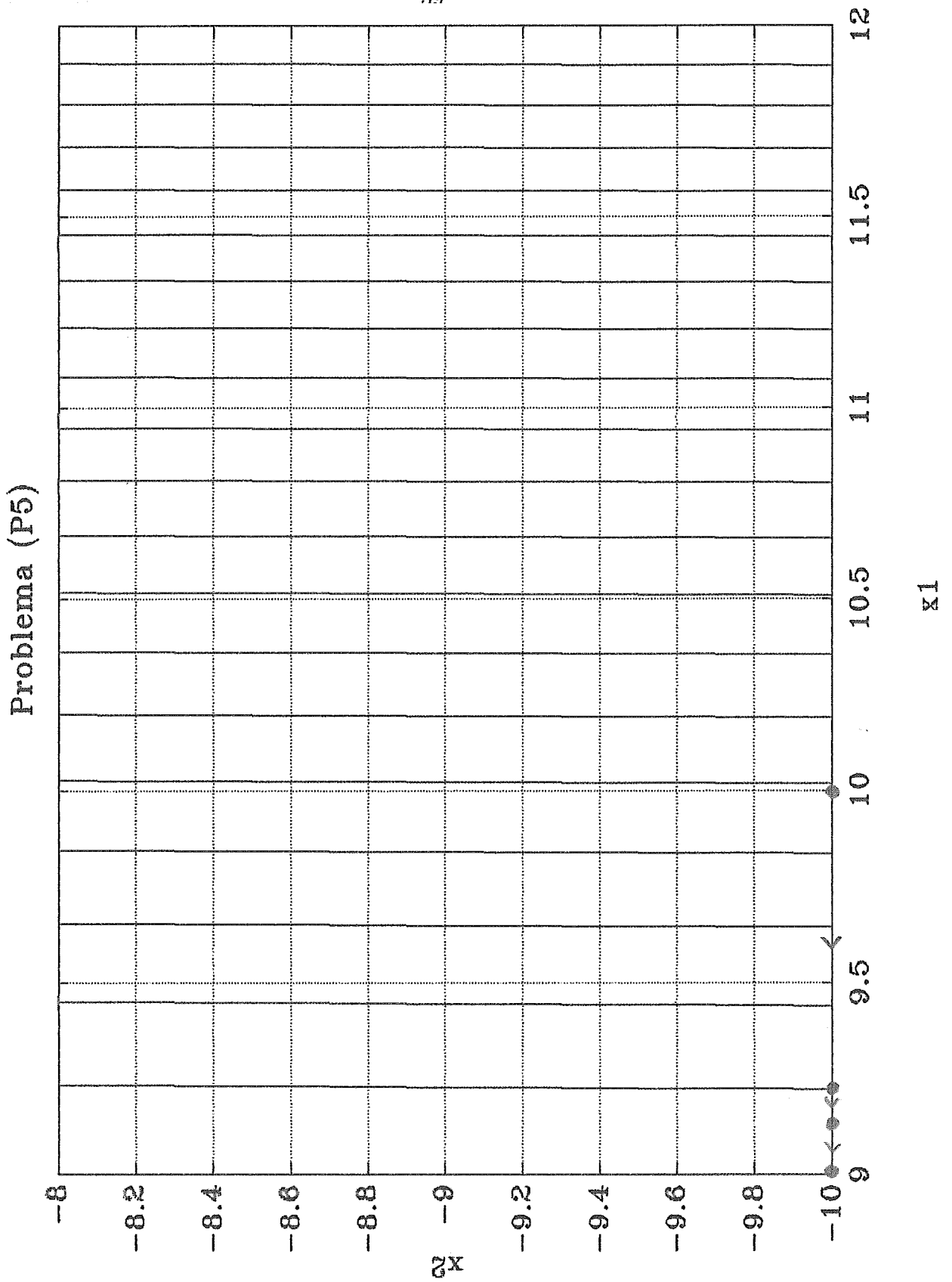


Figura V.4: Curvas de nível da função objetivo de (P5) e os pontos de mínimo do algoritmo região de confiança com formato elipsoidal.

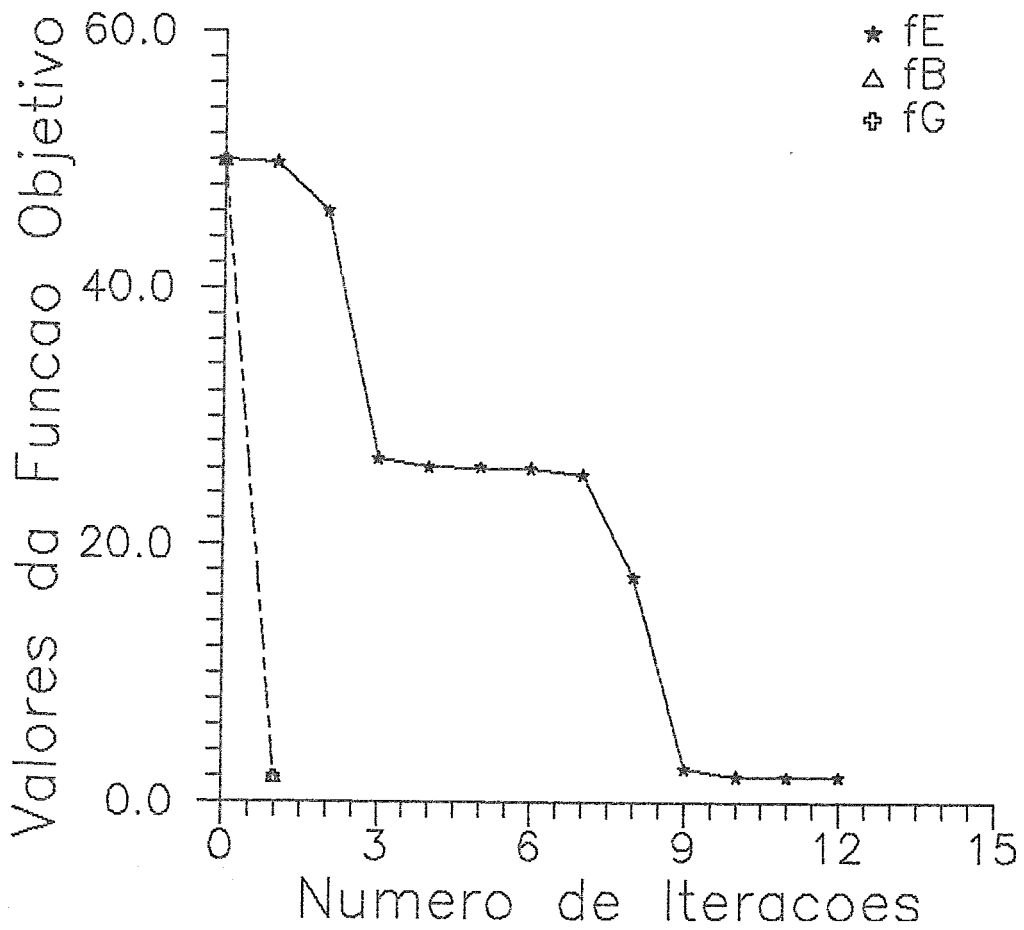
$$(P6) \quad \min f_6 = x_1^2 + x_2^2$$

$$s.a \quad \begin{array}{l} 1 \leq x_1 \leq 5 \\ 1 \leq x_2 \leq 5 \end{array}$$

$$x^0 = (4.995, 4.998)^t$$

$$x^* = (0, 0)^t$$

Problema (P6)



<i>Problema (P6)</i>			
k	f_E	f_B	f_G
0	49.930029	49.930029	49.930029
1	27.725581	2.000000	2.000000
2	26.074782		
3	26.007647		
4	25.975356		
5	25.959521		
6	25.197733		
7	23.753990		
8	21.156868		
9	16.928407		
10	11.197599		
11	5.537542		
12	2.225086		
13	2.028226		

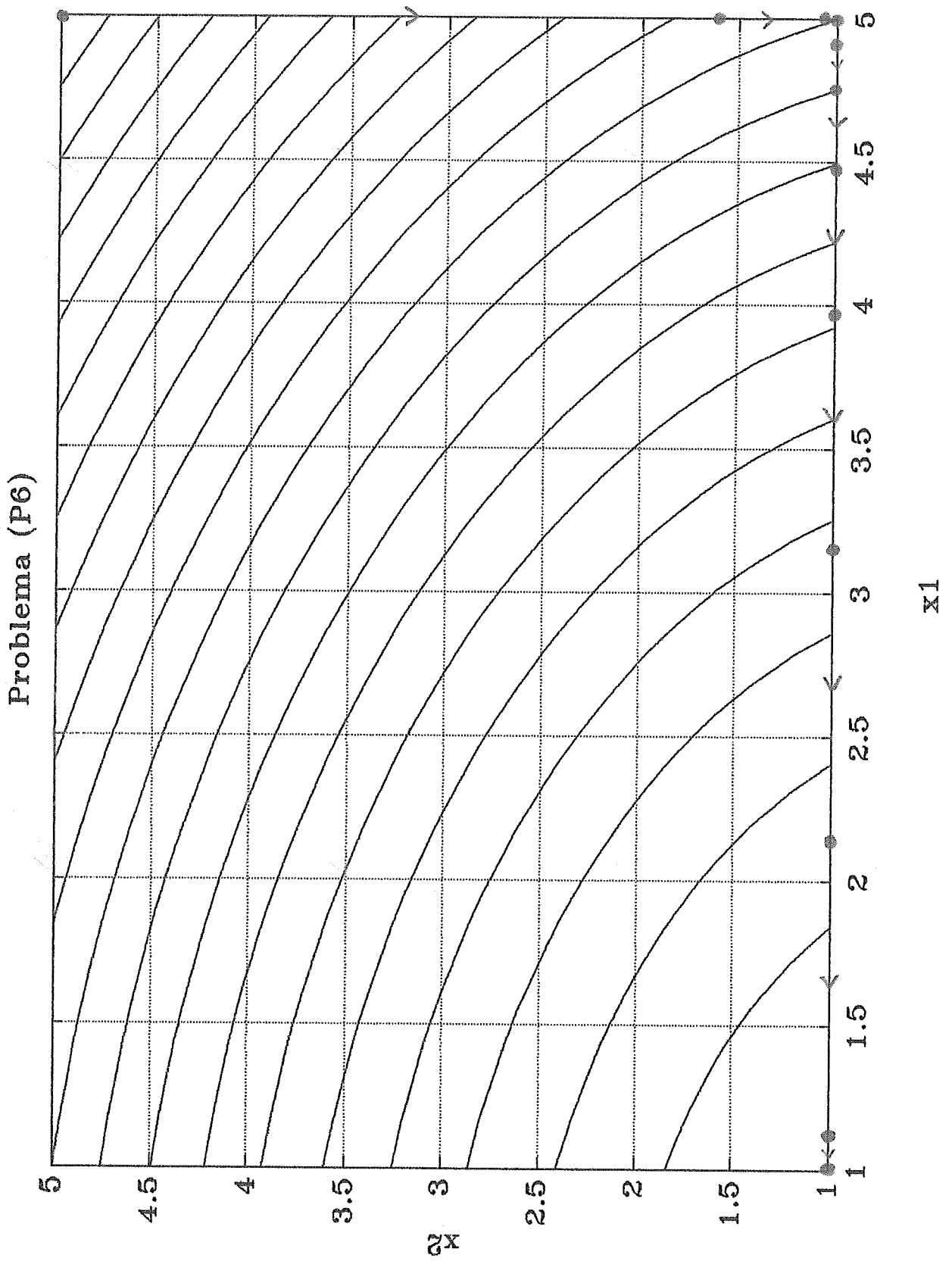
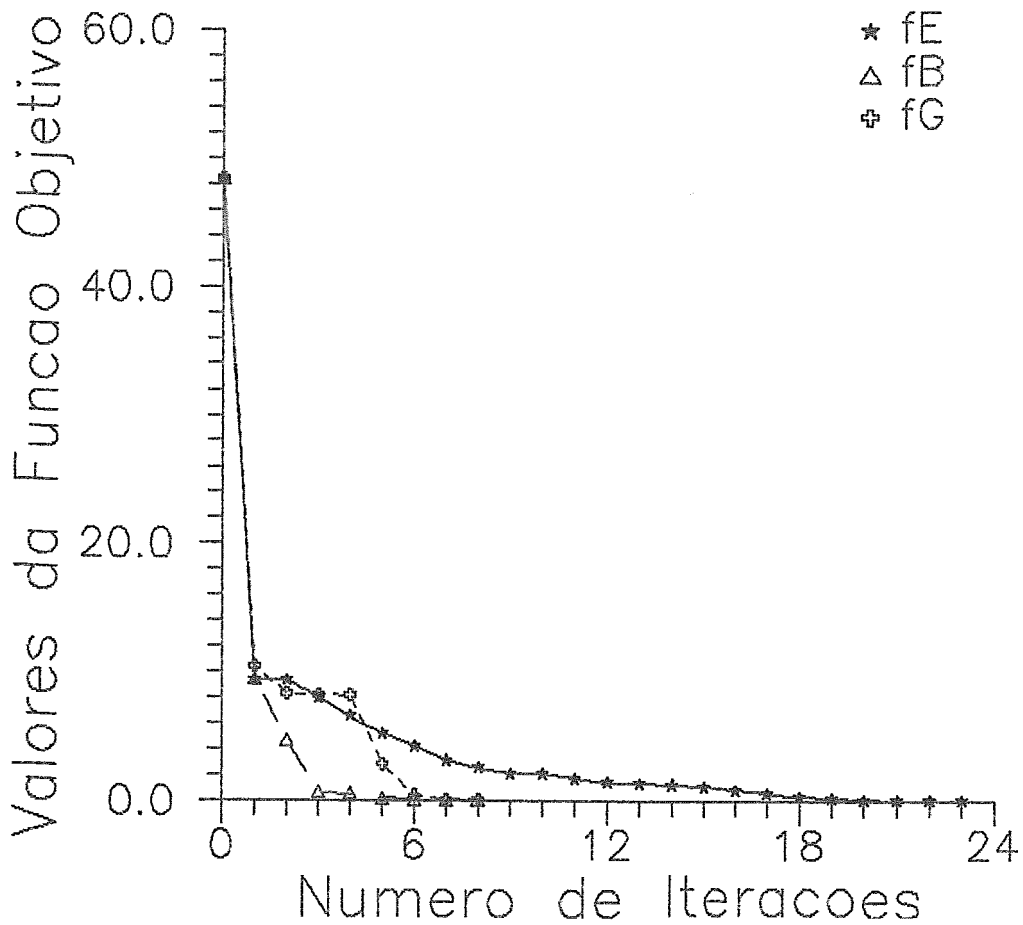


Figura V.5: Curvas de nível da função objetivo de (P6) e os pontos de mínimo do algoritmo região de confiança com formato elipsoidal.

$$\begin{aligned}
 (P7) \quad & \min f_7 = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 100(x_4 - x_3^2)^2 + (1 - x_3)^2 \\
 & -2 \leq x_1 \leq 0.8 \\
 \text{s.a} \quad & 0 \leq x_2 \leq 2 \\
 & -2 \leq x_3 \leq 0.8 \\
 & 0 \leq x_4 \leq 2 \\
 & x^0 = (-1.2, 1, -1.2, 1)^t \\
 & x^* = (1, 1, 1, 1)^t
 \end{aligned}$$

Problema (P7)

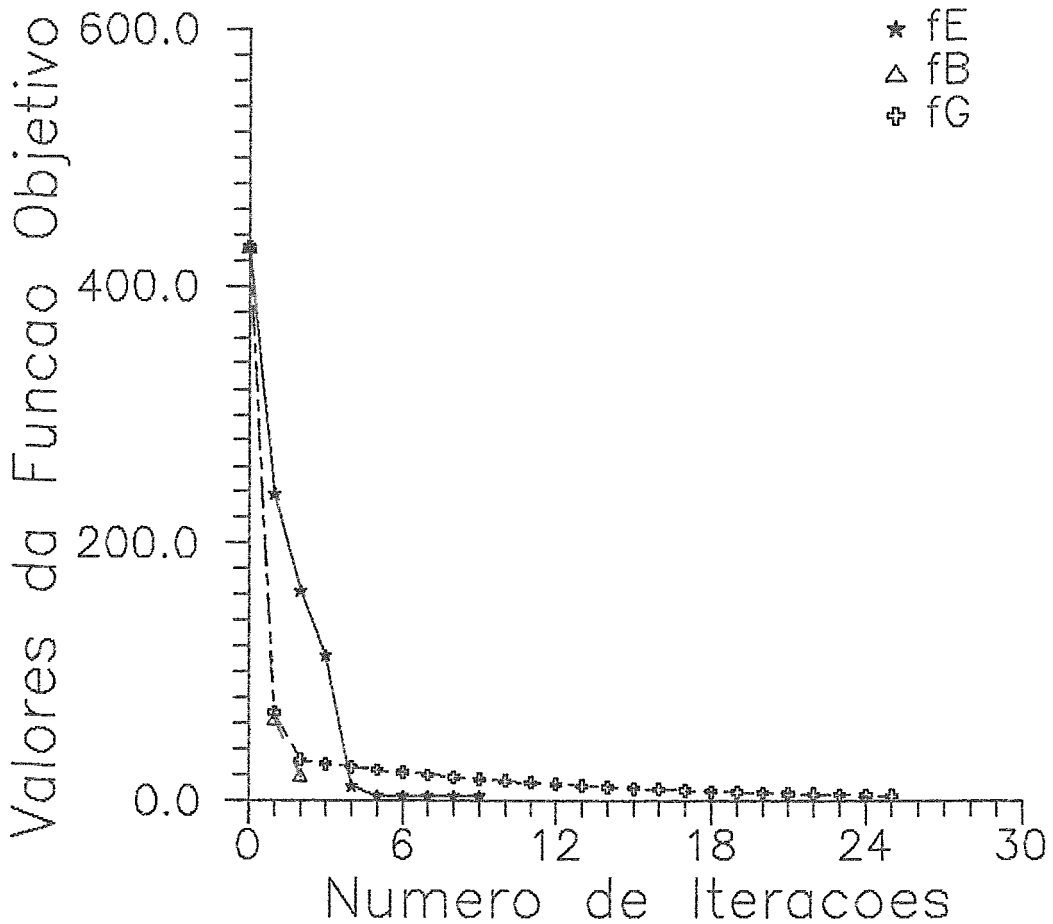


Problema ($P7$)			
k	f_E	f_B	f_G
0	48.400000	48.400000	48.400000
1	9.423551	9.463769	10.491780
2	9.423551	4.753654	8.358386
3	8.133376	0.727665	8.237641
4	6.430776	0.575370	8.220565
5	6.430776	0.270818	2.893760
6	5.322819	0.097344	0.440921
7	4.257913	0.089638	0.144678
8	3.308652	0.080000	0.130130
9	2.622190		0.129054
10	2.132957		0.128452
11	1.599569		0.127967
12	1.039504		0.127475
13	0.773190		0.126987
14	0.522760		0.126503
15	0.356115		0.126022
16	0.203537		0.125545
17	0.118412		0.125136
18	0.087063		0.124551
19	0.086787		0.087735
20			0.080252

$$\begin{aligned}
 (P8) \quad \min \quad & f_8 = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4 + \\
 & (x_5 + 10x_6)^2 + 5(x_7 - x_8)^2 + (x_6 - 2x_7)^4 + 10(x_5 - x_8)^4 \\
 \text{s.a} \quad & 1 \leq x_1 \leq 4 \\
 & -1.001 \leq x_2 \leq 2 \\
 & -1 \leq x_3 \leq 0.01 \\
 & 0 \leq x_4 \leq 2 \\
 & 1 \leq x_5 \leq 4 \\
 & -1.001 \leq x_6 \leq 2 \\
 & -1 \leq x_7 \leq 0.01 \\
 & 0 \leq x_8 \leq 2
 \end{aligned}$$

$$\begin{aligned}
 x^0 &= (3, -1, 0, 1, 3, -1, 0, 1)^t \\
 x^* &= (0, 0, 0, 0, 0, 0, 0, 0)^t
 \end{aligned}$$

Problema (P8)



Problema (P8)			
k	f_E	f_B	f_G
0	430.000000	430.000000	430.000000
1	281.003175	63.651395	68.445909
2	200.542629	19.226764	31.898252
3	165.938684	19.226309	29.018771
4	54.910436	19.225854	26.457734
5	29.564828	19.225398	24.149493
6	21.162200	19.22494	22.358162
7	12.915544		20.337120
8	5.511998		18.536386
9	4.009004		16.918593
10	3.944438		15.465594
11	3.907725		14.316895
12	3.859344		13.046524
13	3.814580		11.909079
14	3.791434		10.889909
15	3.786176		9.971444
16	3.785738		9.227114
17	3.785726		8.425099
18			7.707990
19			7.062913
20			6.560505
21			5.993180
22			5.486856
23			5.032117
24			4.622586
25			4.288832

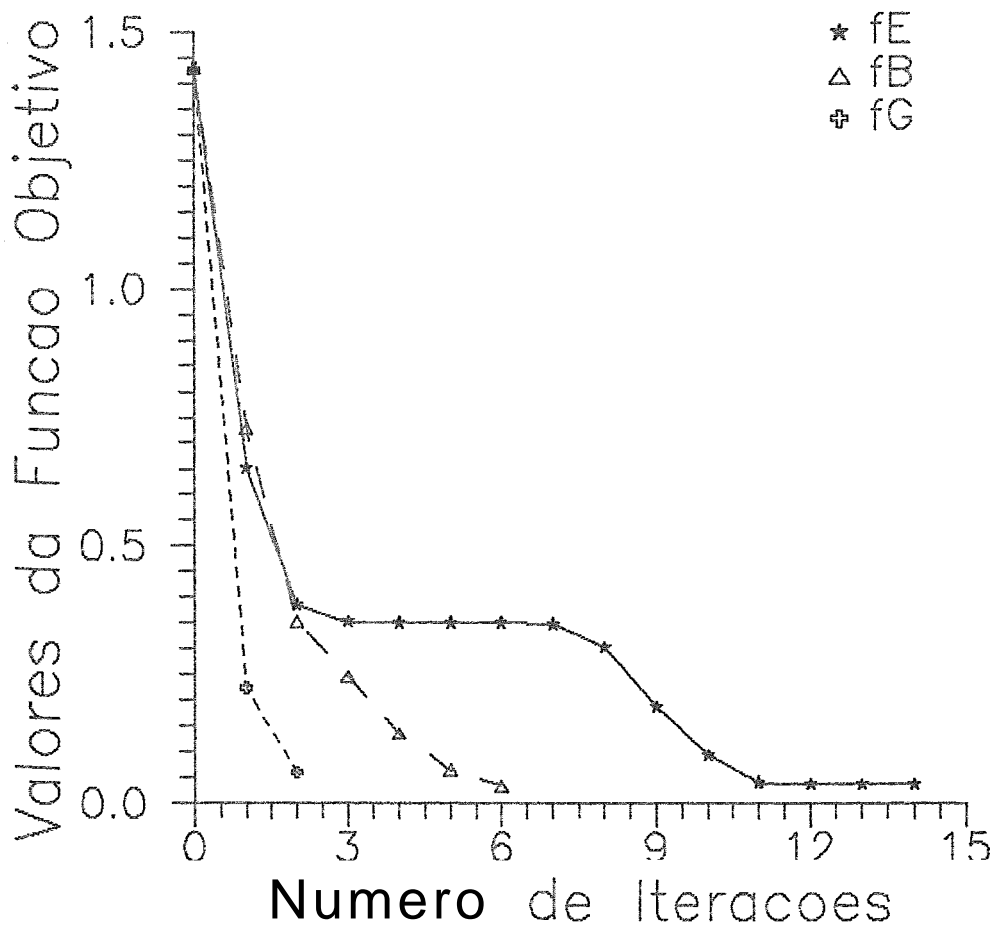
$$(P9) \quad \min \quad f_9 = (\cos x_1 - \cos x_2 + 2\sin x_1)^2 + (3\cos x_2 + 2\sin x_2 - \cos x_1 - 2)^2$$

$$s.a \quad \begin{array}{l} -0.5 \leq x_1 \leq 0.9 \\ 0.1 \leq x_2 \leq 0.5 \end{array}$$

$$x^0 = (0.5, 0.4995)^t$$

$$x^* = (0, 0)^t$$

Problema (P9)



Problema (P9)			
k	f_E	f_B	f_G
0	1.428528	1.428528	1.428528
1	1.428528	0.729439	0.224364
2	0.354003	0.353628	0.060351
3	0.351889	0.246400	0.060344
4	0.351690	0.135519	0.060337
5	0.351684	0.063826	0.060329
6	0.351684	0.034130	0.060322
7	0.262105		
8	0.150036		
9	0.072422		
10	0.037144		
11	0.037142		

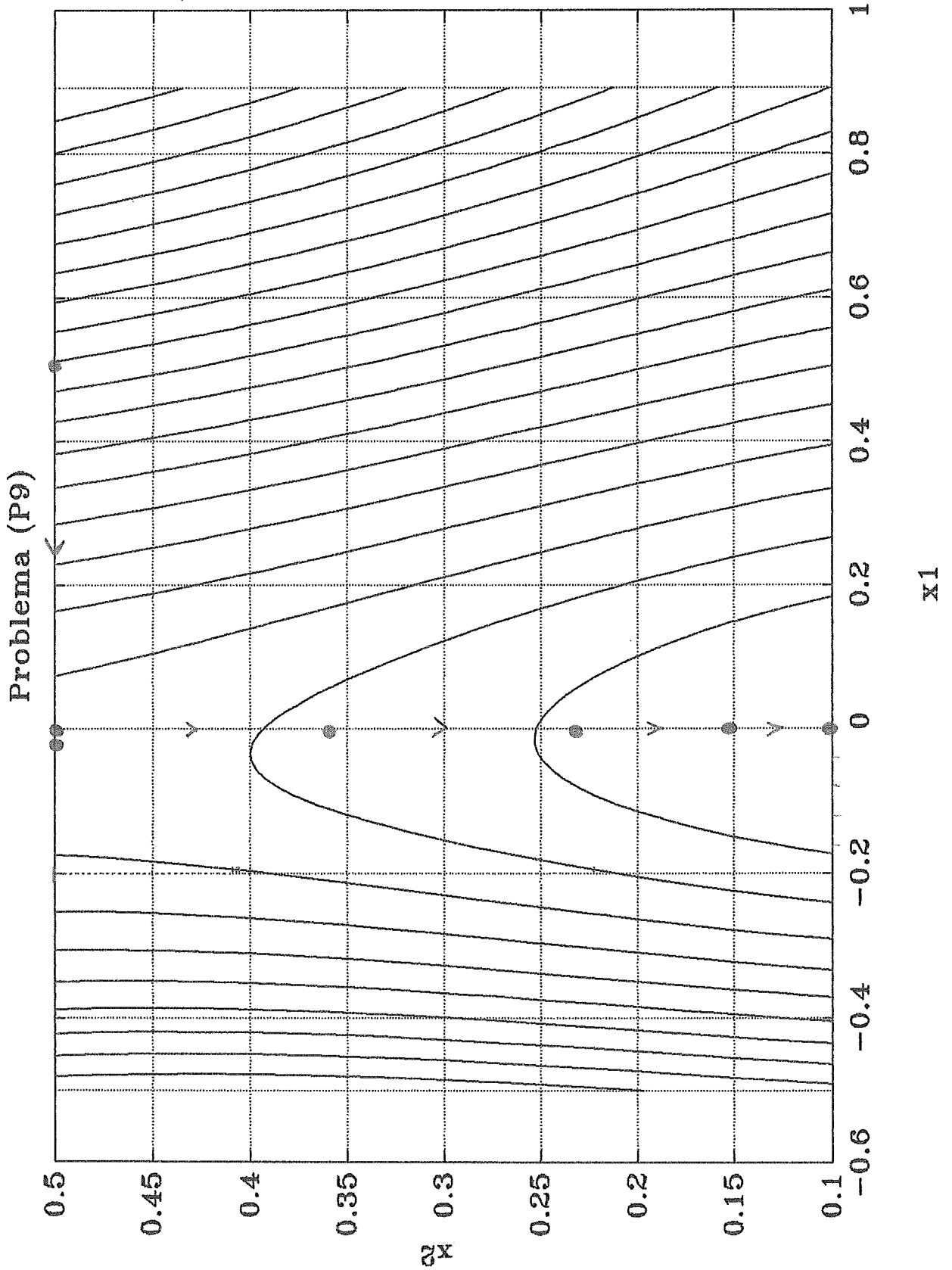


Figura V.6: Curvas de nível da função objetivo de (P9) e os pontos de mínimo do algoritmo região de confiança com formato elipsoidal.

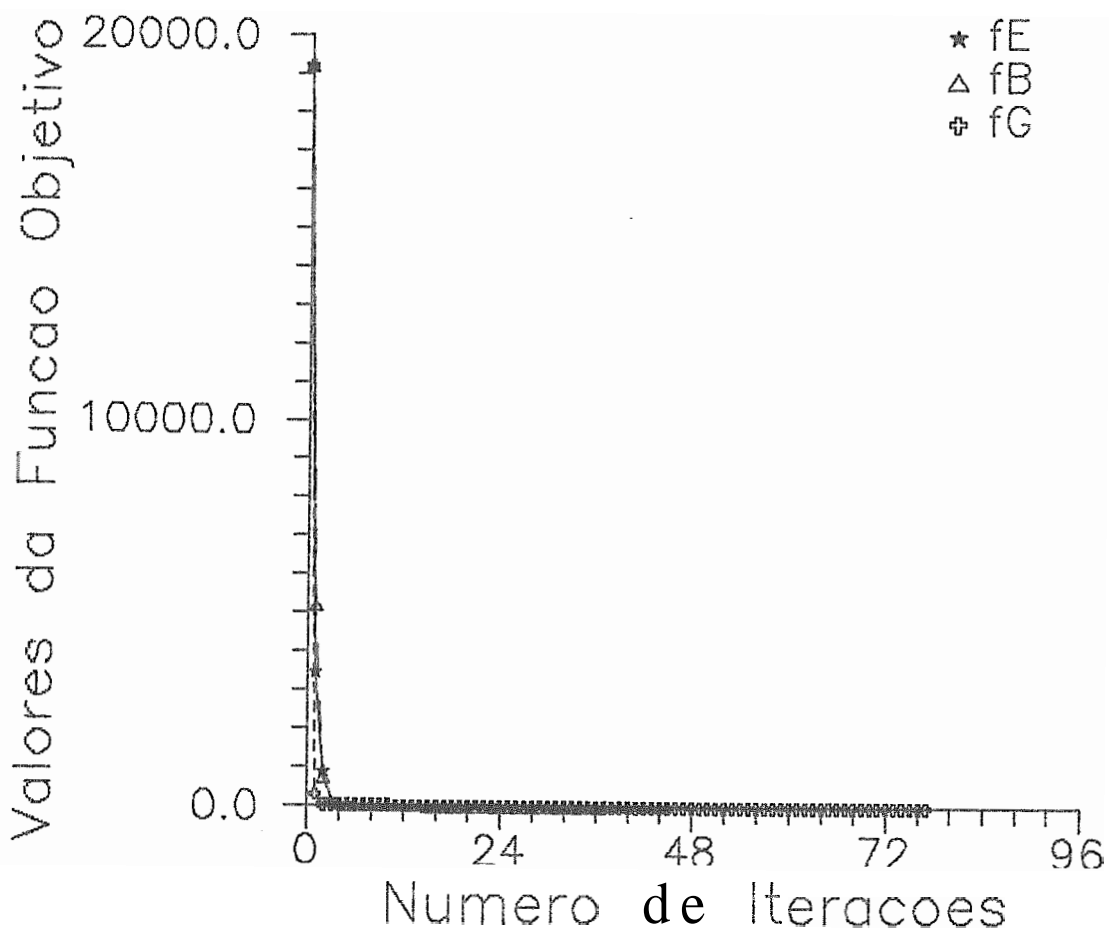
$$(P10) \quad \min \quad f_{10} = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 + 90(x_3^2 - x_4)^2 + (1 - x_3)^2 + \\ 10.1[(1 - x_2)^2 + (1 - x_4)^2] + 19.8(1 - x_2)(1 - x_4)$$

$$s.a \quad \begin{aligned} -5 &\leq x_1 \leq 2 \\ -3 &\leq x_2 \leq 2 \\ -5 &\leq x_3 \leq 2 \\ -3 &\leq x_4 \leq 2 \end{aligned}$$

$$x^0 = (-3, -1, -3, -1)^t$$

$$x^* = (1, 1, 1, 1)^t$$

Problema (P10)



<i>Problema (P10)</i>			
k	f_E	f_B	f_G
0	19192.000000	19192.000000	19192.000000
1	2477.100092	5182.287704	299.534657
2	465.449539	728.050628	33.134845
3	118.188287	102.623389	30.238208
4	50.961049	52.587378	26.328291
5	40.423836	12.187626	23.137610
6	33.150972	8.027711	20.942161
7	23.994990	7.877223	18.817182
8	15.631740	7.876402	16.904324
9	10.888280	7.868592	15.078245
10	8.952067	7.862885	13.558457
11	8.101061	7.823287	12.149745
12	7.636751	7.797370	10.846359
13	7.287447	7.693235	9.777259
14	6.957804	7.611869	8.587096
15	6.753238	7.326293	7.693426
16	6.709752	7.146684	6.794774
17	6.708082	6.589707	6.174950
18	6.708079	6.251867	5.429213
19	6.708079	5.400577	4.792006
20	6.708079	4.969018	4.321042
21	6.708079	4.001204	3.830276
22	6.708079	3.659489	3.430069
23	5.844764	2.959363	3.104533
24	5.197224	2.776989	2.787180
25	4.568200	2.245220	2.493450
26	3.837842	2.027562	2.248267
27	3.509427	1.640163	2.022333
28	2.821746	1.445733	1.798996
29	2.821746	0.999399	1.630079
30	2.821746	0.798725	1.453989
31	2.502275	0.447886	1.327890
32	2.159512	0.299089	1.183136
33	1.900375	0.100811	1.075741
34	1.512241	0.041741	0.983551
35	1.512241	0.004461	0.886054
36	1.512241	0.000231	0.818818
37	1.375039	0.000000	0.733712
⋮	⋮		⋮
52	0.000000		0.388433
⋮			⋮
80			0.136756

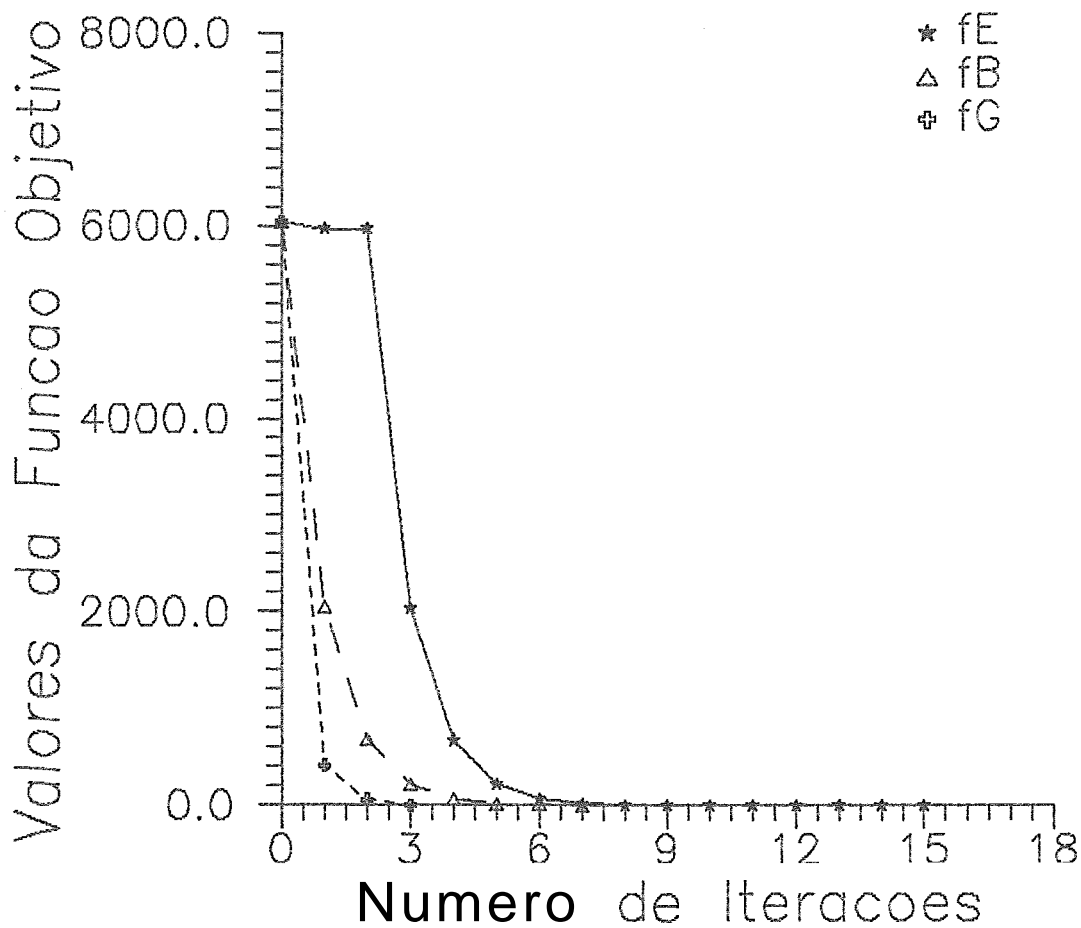
$$(P11) \quad \min \quad f_{11} = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$$

$$s.a \quad \begin{array}{l} 1 \leq x_1 \leq 5 \\ -10 \leq x_2 \leq -1 \end{array}$$

$$x^0 = (4.995, -5)^t$$

$$x^* = (0.5, -1)^t$$

Problema (P11)



Problema (P11)			
k	f_E	f_B	f_G
0	6039.838233	6039.838233	6039.838233
1	5967.471313	2051.983616	409.243970
2	5967.471107	680.882871	56.940008
3	5967.471107	218.227202	2.728940
4	2028.398152	66.320061	
5	673.309536	18.494520	
6	215.845797	4.446417	
7	65.594400	1.419381	
8	18.284397		
9	4.390898		
10	1.540373		
11	1.377716		
12	1.377303		
13	1.376677		
14	1.375925		
15	1.375315		
16	1.375041		
17	1.374985		
18	1.374981		

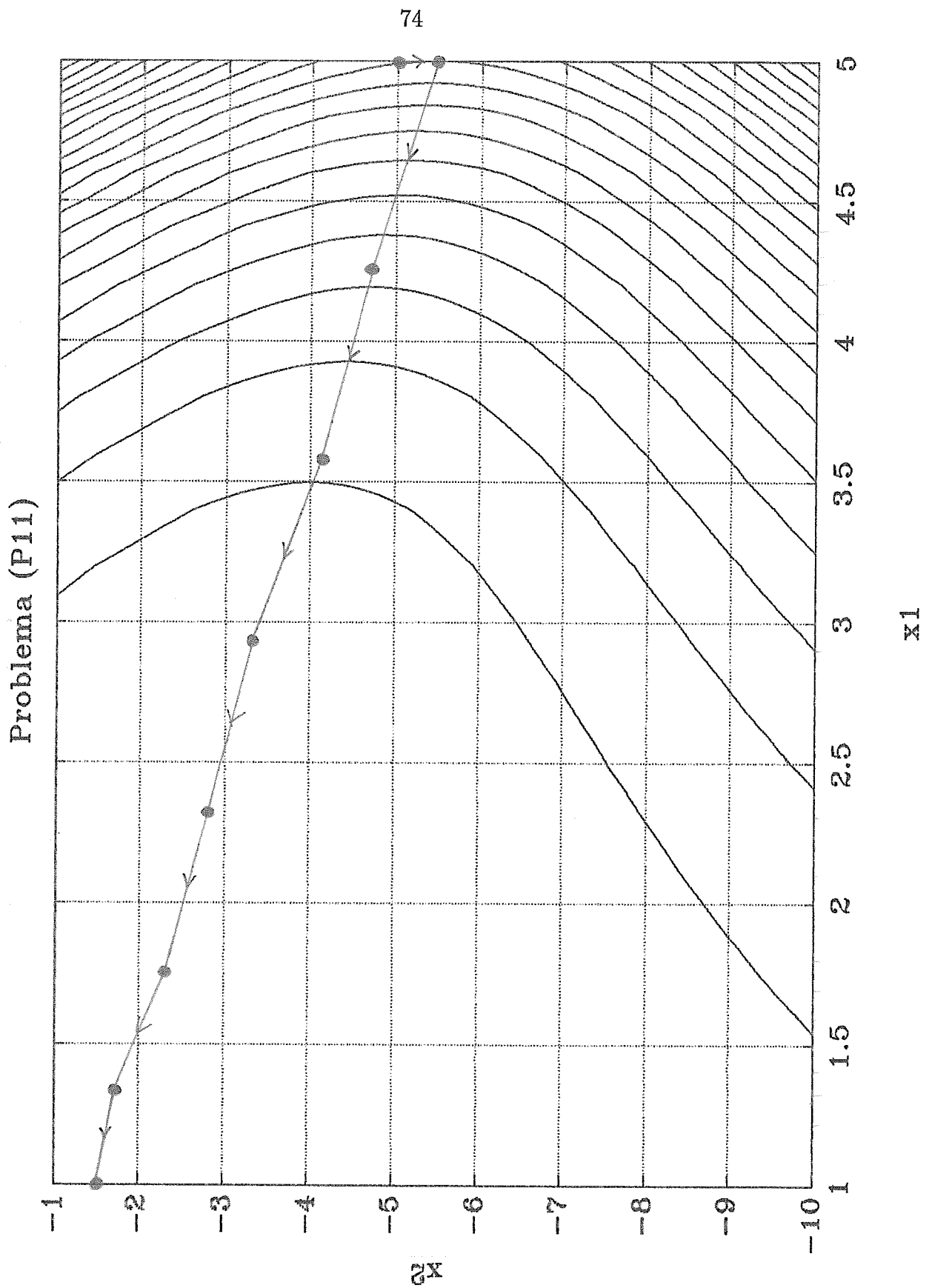


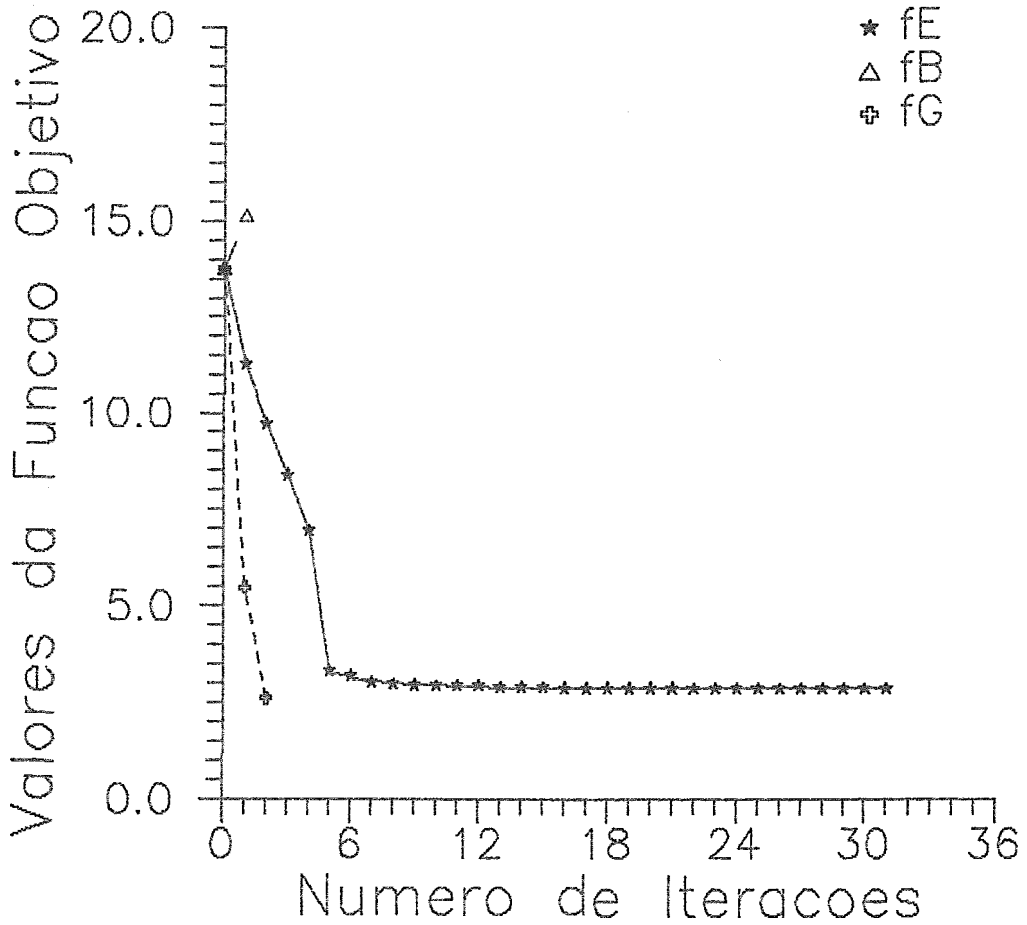
Figura V.7: Curvas de nível da função objetivo de (P11) e os pontos de mínimo do algoritmo região de confiança com formato elipsoidal.

$$\begin{aligned}
 (P12) \quad \min \quad & f_{12} = (5 - \cos x_1 - \cos x_2 - \cos x_3 - \cos x_4 - \cos x_5 + 5 \operatorname{sen} x_1 - 5(1 - \cos x_1))^2 + \\
 & (5 - \cos x_1 - \cos x_2 - \cos x_3 - \cos x_4 - \cos x_5 + 5 \operatorname{sen} x_2 - 10(1 - \cos x_2))^2 + \\
 & (5 - \cos x_1 - \cos x_2 - \cos x_3 - \cos x_4 - \cos x_5 + 5 \operatorname{sen} x_3 - 15(1 - \cos x_3))^2 + \\
 & (5 - \cos x_1 - \cos x_2 - \cos x_3 - \cos x_4 - \cos x_5 + 5 \operatorname{sen} x_4 - 20(1 - \cos x_4))^2 + \\
 & (5 - \cos x_1 - \cos x_2 - \cos x_3 - \cos x_4 - \cos x_5 + 5 \operatorname{sen} x_5 - 25(1 - \cos x_5))^2 \\
 \text{s.a} \quad & 0.2 \leq x_1 \leq 0.9 \\
 & 0.4 \leq x_2 \leq 0.9 \\
 & -0.8 \leq x_3 \leq 0.8 \\
 & -0.9 \leq x_4 \leq 0.9 \\
 & -1 \leq x_5 \leq 0.2
 \end{aligned}$$

$$x^0 = (0.8, 0.8, 0.2, 0.2, 0.1995)^t$$

$$x^* = (0, 0, 0, 0, 0)^t$$

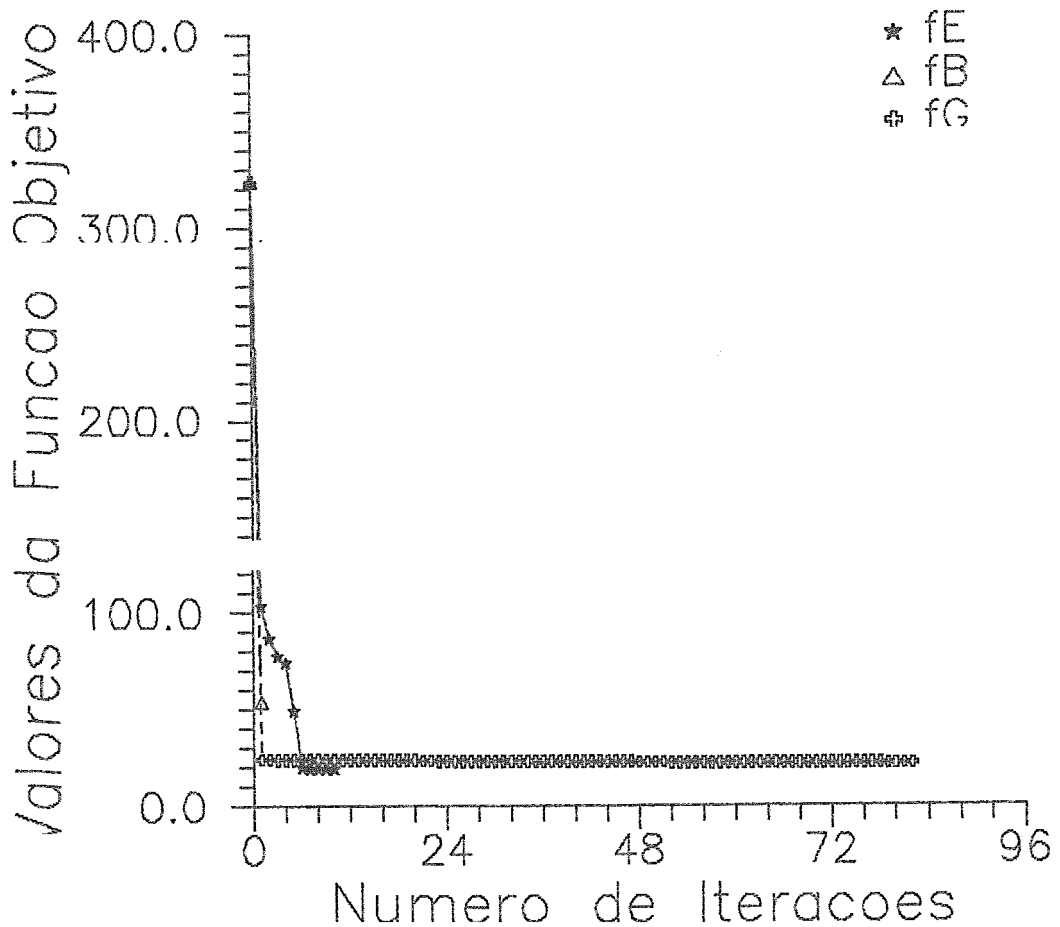
Problema (P12)



Problema (P12)			
k	f_E	f_B	f_G
0	13.767760	13.767760	13.767760
1	8.698268	15.121756	5.478695
2	6.724525		2.634163
3	3.478733		
4	3.203970		
5	3.169052		
6	3.118953		
7	3.056783		
8	3.001242		
9	2.989401		
10	2.988995		
11	2.988958		
12	2.988957		

$$\begin{aligned}
 (P13) \quad \min \quad & f_{13} = 100(x_3 - 10\theta)^2 + 100[(x_1^2 + x_2^2)^{\frac{1}{2}} - 1]^2 + x_3^2 \\
 & \theta = \begin{cases} (1/2\pi) \arctan(x_2/x_1) & \text{se } x_1 > 0 \\ (1/2\pi) \arctan(x_2/x_1) + 0.5 & \text{se } x_1 < 0 \end{cases} \\
 \text{s.a} \quad & 1 \leq x_1 \leq 3 \\
 & 1 \leq x_2 \leq 3 \\
 & 1 \leq x_3 \leq 2 \\
 & x^0 = (2, 1.5, 1.995)^t \\
 & x^* = (1, 0, 0)^t
 \end{aligned}$$

Problema (P13)



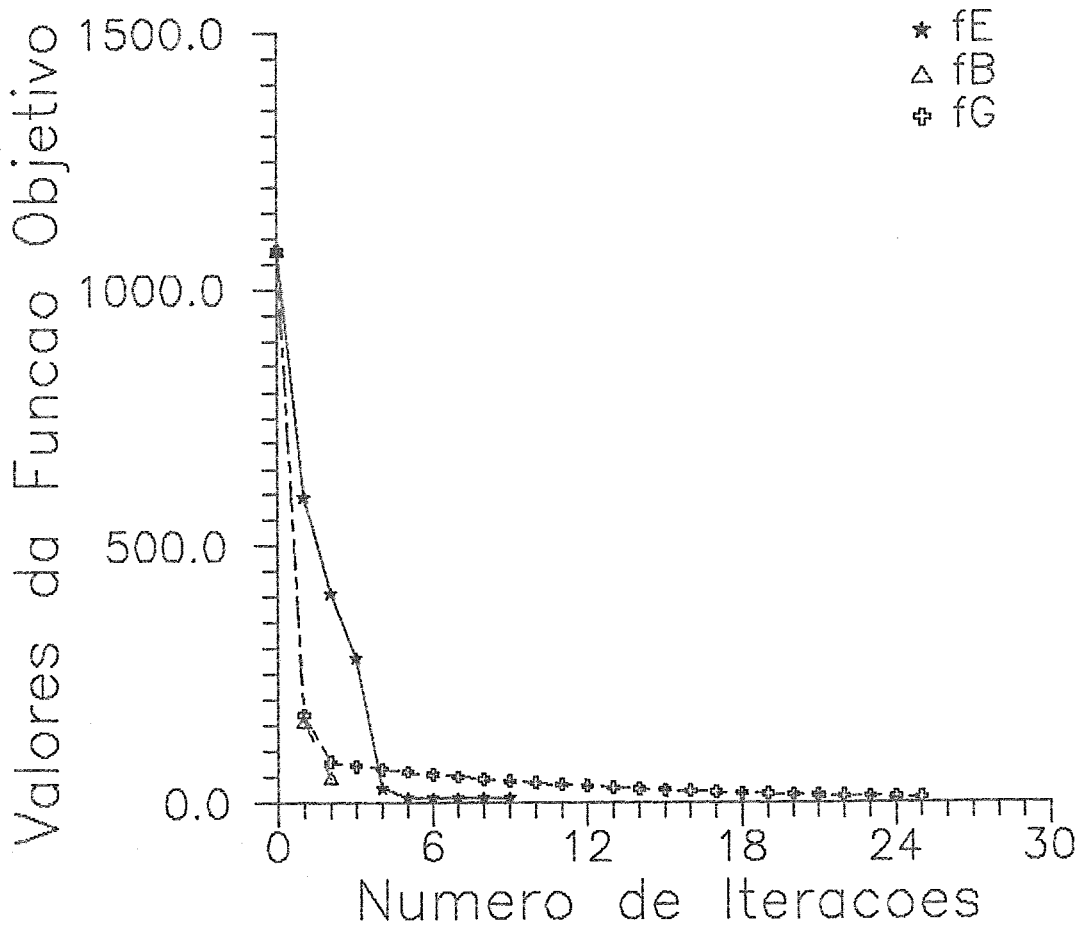
<i>Problema (P13)</i>			
k	f_E	f_B	f_G
0	323.232313	323.232313	323.232313
1	153.351689	54.196397	24.407288
2	143.721871		24.366257
3	123.653355		24.325521
4	94.418350		24.285079
5	82.720593		24.244927
6	77.449199		24.205065
7	59.878132		24.165489
8	40.185237		24.126198
9	25.852594		24.087190
10	20.319786		24.048462
11	19.396681		24.010013
12	19.341021		23.971840
13	19.339966		23.933943
14	19.339960		23.896317
15			23.858963
16			23.821877
17			23.785058
18			23.748504
19			23.712213
20			23.676183
21			23.640413
22			23.604899
23			23.569642
24			23.534637
25			23.499885
26			23.465383
27			23.431129
28			23.397121
29			23.363358
30			23.329838
31			23.296559
32			23.263520
33			23.230718
34			23.198152
35			23.165821
36			23.133722
37			23.101854
38			23.070216
39			23.038805
40			23.007620
⋮			⋮
83			21.859025

$$\begin{aligned}
(P14) \quad \min \quad & f_{14} = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4 + \\
& (x_5 + 10x_6)^2 + 5(x_7 - x_8)^2 + (x_6 - 2x_7)^4 + 10(x_5 - x_8)^4 + \\
& (x_9 + 10x_{10})^2 + 5(x_{11} - x_{12})^2 + (x_{10} - 2x_{11})^4 + 10(x_9 - x_{12})^4 + \\
& (x_{13} + 10x_{14})^2 + 5(x_{15} - x_{16})^2 + (x_{14} - 2x_{15})^4 + 10(x_{13} - x_{16})^4 + \\
& (x_{17} + 10x_{18})^2 + 5(x_{19} - x_{20})^2 + (x_{18} - 2x_{19})^4 + 10(x_{17} - x_{20})^4 \\
s.a \quad & \begin{array}{llll}
1 & \leq & x_1 & \leq & 4 \\
-1.001 & \leq & x_2 & \leq & 2 \\
-1 & \leq & x_3 & \leq & 0.01 \\
0 & \leq & x_4 & \leq & 2 \\
1 & \leq & x_5 & \leq & 4 \\
-1.001 & \leq & x_6 & \leq & 2 \\
-1 & \leq & x_7 & \leq & 0.01 \\
0 & \leq & x_8 & \leq & 2 \\
1 & \leq & x_9 & \leq & 4 \\
-1.001 & \leq & x_{10} & \leq & 2 \\
-1 & \leq & x_{11} & \leq & 0.01 \\
0 & \leq & x_{12} & \leq & 2 \\
1 & \leq & x_{13} & \leq & 4 \\
-1.001 & \leq & x_{14} & \leq & 2 \\
-1 & \leq & x_{15} & \leq & 0.01 \\
0 & \leq & x_{16} & \leq & 2 \\
1 & \leq & x_{17} & \leq & 4 \\
-1.001 & \leq & x_{18} & \leq & 2 \\
-1 & \leq & x_{19} & \leq & 0.01 \\
0 & \leq & x_{20} & \leq & 2
\end{array}
\end{aligned}$$

$$x^0 = (3, -1, 0, 1, 3, -1, 0, 1, 3, -1, 0, 1, 3, -1, 0, 1, 3, -1, 0, 1)^t$$

$$x^* = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^t$$

Problema (P14)



<i>Problema (P14)</i>			
k	f_E	f_B	f_G
0	1075.000000	1075.000000	1075.000000
1	702.507938	159.128487	171.114771
2	501.356573	48.066910	79.745630
3	414.846709		72.546928
4	137.276090		66.144334
5	73.912069		60.373732
6	52.905499		55.895406
7	32.288861		50.842800
8	13.779996		46.340966
9	10.022510		42.296483
10	9.861096		38.663985
11	9.769313		35.792238
12	9.648360		32.616311
13	9.536449		29.772698
14	9.478585		27.224771
15	9.465441		24.928609
16	9.464346		23.067784
17	9.464316		21.062747
18			19.269975
19			17.657282
20			16.401263
21			14.982950
22			13.717140
23			12.580293
24			11.556465
25			10.722079

Capítulo VI

Conclusões

Os algoritmos método de região de confiança com formato elipsoidal, método de gradiente projetado e método de máximo declive foram implementados nesta tese sem a preocupação quanto a eficiência numérica. Nosso objetivo foi analisar o desempenho dos algoritmos com restrições simples do ponto de vista de convergência.

Resolvemos 14 problemas não lineares de minimização de pequeno porte usando os três algoritmos citados acima. Os problemas foram selecionados com funções convexas e não convexas e de maneira que pudéssemos observar a técnica de conjuntos ativos introduzida no método de região de confiança.

O método de região de confiança com formato elipsoidal apresentou-se robusto. Ele foi capaz de resolver todos os problemas. Porém foi muito eficiente para os problemas com funções não convexas, veja os resultados comparativos de P4, P8, P10, P13 e P14. Os problemas não lineares quadráticos, P1, P2, P5 e P6, mostraram-se triviais para os métodos gradiente projetado e máximo declive.

Observou-se uma certa lentidão na liberação das variáveis pertencentes ao conjunto ativo de alguns problemas, como P1, P2, P5, P6, P8, P9, P11, P12 e P14, o que prejudicou o desempenho do algoritmo do método de região de confiança com formato elipsoidal. O uso de estratégias arrojadas no algoritmo que trata o conjunto ativo pode melhorar bastante o seu desempenho, podendo até competir com os métodos que foram eficientes nos problemas quadráticos.

Referências

- [1] M. S. Bazaraa and C. M. Shetty. *Nonlinear Programming Theory and Algorithms*. John Wiley & Sons, New York, N. Y., 1979.
- [2] D. P. Bertsekas. On the Golgstein-Levitin-Poljac gradient projection method. *IEEE Transactions on Automatic Control*, 21:174–184, 1976.
- [3] D. P. Bertsekas. Projected Newton methods for optimization problems with simple constraints. *SIAM Journal of Control and Optimization*, 20(2):221–246, 1982.
- [4] A. Conn, N. Gould, and Ph. Toint. *Global Convergence of a Class of Trust Region Algorithms for Optimization Problems with Simple Bounds*. Report, Dept. of Computer Sciences, University of Waterloo, Waterloo, Canada, 1988.
- [5] Mahmoud Mahmoud El-Alem. *A Global Convergence Theory for a Class of Trust Region Algorithms for Constrained Optimization*. PhD thesis, Rice University, Houston, Texas, 1988.
- [6] R. Fletcher. *Practical Methods of Optimization*. John Wiley and Sons, New York, N.Y., second edition, 1987.
- [7] M. A. Malcom G. E. Forsythe and C. B. Moler. *Computer Methods for Mathematical Computations*. Prentice-Hall, Englewood Cliffs, N. J., 1977.
- [8] P. Gill, W. Murray, and M. Wright. *Practical Optimization*. Princeton University Press, New York, N.Y., 1981.
- [9] G. H. Golub and C. F. Van Loan. *Matriz Computations*. The John Hopkins University Press, Baltimore, Maryland, 1983.
- [10] C. C. Gonzaga. *Algoritmos de Pontos Interiores para Programação Linear*. Instituto de Matemática Pura e Aplicada, Rio de Janeiro, R.J., 1989.
- [11] C. C. Gonzaga. *An interior trust region method for linearly constrained nonlinear programming*. Internal Report, COPPE - Federal University of Rio de Janeiro, 1991.
- [12] W. Hock and K. Schittkowski. *Test Examples for Nonlinear Programming Codes*. Springer Verlag, Beilin, 1981.
- [13] Jr. J. Dennis and R. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, N.J., 1983.
- [14] B. W. Kernighan and D. M. Ritchie. *The C Programming Language*. Prentice-Hall, Englewood Cliffs, N. J., 1978.
- [15] R. Lafore. *The Waite Group's Turbo C Programrning Language for the PC*. Howard W. Sans & Company, Indianapolis, Indiana, second edition, 1989.

- [16] K. Levenberg. A method for the solution of certain problems in least-squares. *Quarterly Journal of Applied Mathematics*, 12:164–168, 1944.
- [17] D. G. Luenberger. *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, Reading, Massachusetts, 1973.
- [18] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11:431–441, 1963.
- [19] J. Moré and D. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.
- [20] J. Moré and G. Toraldo. *On the Solution of Large Quadratic Programming Problems with Bound Constraints*. Report ANL/MCS-P77-0589, Argonne National Laboratory, Argonne, Illinois, 1989.
- [21] J. J. Moré. Recent developments in algorithms and software for trust region methods. In M. Gotschel A. Bachem and B. Korte, editors, *Mathematical Programming – The State of the Art, Bonn 1982*, Springer Verlag, Berlin, 1983.
- [22] E. Hille S. L. Salas and J. T. Anderson. *Calculus, One and Several Variables*. John Wiley & Sons, New York, N. Y., 1986.
- [23] N. Sagara. A hybrid method for the nonlinear least squares problem with simple bounds. *Journal of Computational and Applied Mathematics*, 36:149–157, 1991.
- [24] D. Sorensen. Newton’s method with a model trust region modification. *SIAM Journal on Numerical Analysis*, 19:409–426, 1982.
- [25] D. Sorensen. Trust region methods for unconstrained minimization. In M. J. D. Powell, editor, *Nonlinear Optimization 1981*, Academic Press, London, 1982.