Universidade Federal de Minas Gerais

Pós–Graduação em Matemática

Mestrado em Matemática

José Luis Almendras Montero

UM MÉTODO TRUST-REGION PARA OTIMIZAÇÃO COM RESTRIÇÕES FAZENDO USO DO MÉTODO DO GRADIENTE PROJETADO

Belo Horizonte

2014

José Luis Almendras Montero

UM MÉTODO TRUST-REGION PARA OTIMIZAÇÃO COM RESTRIÇÕES FAZENDO USO DO MÉTODO DO GRADIENTE PROJETADO

Dissertação apresentada ao Programa de Pós–graduação em Matemática da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre, na área de matemática.

Orientador : Dr. Ricardo Hiroshi Caldeira Takahashi.

Belo Horizonte

2014

José Luis Almendras Montero

UM MÉTODO TRUST-REGION PARA OTIMIZAÇÃO COM RESTRIÇÕES FAZENDO USO DO MÉTODO DO GRADIENTE PROJETADO

Dissertação aprovada pela Comissão Examinadora abaixo elencada como requisito para a obtenção do título de Mestre em Matemática pelo Mestrado Acadêmico em Matemática do Instituto de Ciências Exatas da Universidade Federal de Gerais.

Banca examinadora:

Prof. Dr. Ricardo Hiroshi Caldeira Takahashi
(Orientador)
Instituto de Ciências Exatas - UFMG

Profa. Dra. Denise Burgarelli Duczmal
Instituto de Ciências Exatas - UFMG

Prof. Dr. Rodrigo Tomás Nogueira Cardoso
CEFET-MG

Belo Horizonte, 15 de Maio de 2014.

A meus pais, Domínica e Alberto, que sempre me ajudaram com seu apoio e conselhos de maneira incondicional.

Agradecimentos

Inicio meus agradecimentos por DEUS, já que Ele colocou pessoas tão especiais a meu lado, sem as quais certamente não teria dado conta!

A meus pais, meu infinito agradecimento. Sempre acreditaram em minha capacidade e me acharam O melhor de todos, mesmo não sendo. Isso só me fortaleceu e me fez tentar, não ser O melhor, mas a fazer o melhor de mim. Obrigado pelo amor incondicional!.

A meus irmãos Carlos, Yesenia e a minha sobrinha Milagros meu agradecimento especial, pois sempre se orgulharam de mim e confiaram em meu trabalho. Obrigado pela confiança!

Agradeço ao meu orientador Ricardo Takahashi, pela dedicação, orientação e paciência dispensadas à minha pessoa.

Aos meus amigos Carlos, Edwin, Julio, Roy, Miguel, Farley, Renato, Mario, Aislan, Carlos G, Juliano e todos os colegas do mestrado pelo companherismo e atenção.

RESUMO

Neste trabalho, estudaremos o método Trust-region para resolver um problema de otimização com restrições simples, ou seja estamos interessados em construir um algoritmo para resolver o seguinte problema: Encontrar $x^* \in \Omega$ tal que $f(x^*) \leq f(x)$, para todo $x \in \Omega$, onde $\Omega = \{x \in \mathbb{R}^n / L_i \leq x_i \leq U_i, L_i, U_i \in \mathbb{R}\}$, e a função f é suposta ser duas vezes continuamente diferenciável no conjunto factível Ω, precisamos achar x^* tal que: $x^* = \arg\min_{x\in\Omega} f(x)$. A partir de um ponto inicial e fazendo uso do método Trust-Region, geraremos uma sequência de pontos $\{x\}^k$ tal que $\lim_{k\to\infty} x^k = x^*$, cada ponto é obtido da seguinte maneira, $x^{k+1} = x^k + s^k$ e s^k é solução do seguinte subproblema:

$$s^k = \arg \min_{\substack{\|x-x_k\|\leq\Delta_k \\ L\leq x\leq U}} \left(f(x^k) + \left\langle \nabla f(x^k), x - x^k \right\rangle + \frac{1}{2}\left\langle x - x^k, H^k(x - x^k) \right\rangle \right)$$

H^k é uma aproximação da matriz hessiana no ponto x^k. O método do gradiente projetado vai ser usado para resolver o subproblema, isso vai garantir que as iterações sejam sempre pontos factíveis.

ABSTRACT

In this work, we study a trust-region method for solving optimization problems with simple constraints. We are interested in building an algorithm for the following problem: find $x^* \in \Omega$ such that $f(x^*) \leq f(x)$, $\forall x \in \Omega$, in which $\Omega = \{x \in \mathbb{R}^n / L_i \leq x_i \leq U_i; L_i, U_i \in \mathbb{R}\}$, and f is twice differentiable within the feasible set Ω. Starting from an initial point, the trust-region method generates a sequence $\{x\}^k$ such that $\lim_{k\to\infty} x^k = x^*$. The sequence is generated by the recursion $x^{k+1} = x^k + s^k$, in which s^k is the solution of the following subproblem:

$$s^k = \arg \min_{\substack{\|x - x_k\| \leq \Delta_k \\ L \leq x \leq U}} \left(f(x^k) + \left\langle \nabla f(x^k), x - x^k \right\rangle + \frac{1}{2} \left\langle x - x^k, H^k (x - x^k) \right\rangle \right)$$

In this expression, H^k is an approximation of the Hessian matrix on the point x^k. The projected gradient method is used in order to solve the subproblem, in this way ensuring that all iterations generate feasible solutions.

LISTA DE FIGURAS

SUMÁRIO

INTRODUÇÃO

A otimização não linear desempenha um papel muito importante nos campos da ciência e da engenharia, na indústria, e num grande número de problemas práticos. Frequentemente os parâmetros de otimização são dados pela natureza do problema em questão e fazendo uso dessas informações construimos algoritmos que não são outra coisa que processos iterativos; ou seja, são iniciados por meio da estimativa das variáveis de decisão, seguida do cálculo da função objetivo e então por uma sequência de estimativas até se encontrar a solução ótima (valor extremo da função objetivo). A estratégia usada para se mover de uma iteração para outra se diferencia de um algoritmo para outro. Muitas estratégias usam o valor da função objetivo, restrições e possivelmente primeira e segunda derivadas da função objetivo. Um método é chamado de determinístico se for possível prever todos os seus passos conhecendo seu ponto de partida, este tipo de método gera uma seqüência determinística de possíveis soluções requerendo algumas vezes, o uso da derivada da função objetivo em relação às variáveis.
Um problema de otimização não linear irrestrita consiste em minimizar uma função objetivo não linear $f : \mathbb{R}^n \to \mathbb{R}$ sobre todos os vetores $x \in \mathbb{R}^n$. Em geral, a estrutura dos problemas de otimização é a partir de um ponto inicial x^0, gerar uma sequência de pontos $\{x_n\}$ tal que $\lim_{n\to\infty} x_n = x^*$ onde x^* é um minimizador local da função f. Os novos pontos gerados a partir do ponto inicial são obtidos a partir da informação local da função objetivo no ponto atual. No presente trabalho estamos interessados em resolver o problema otimização com restrições formulado da forma seguinte.

Sejam $\{L_i\}_{i\in\mathbb{N}}$ e $\{U_i\}_{i\in\mathbb{N}}$, sequências finitas de números reais tais que

$$-\infty < L_i < U_i < +\infty \tag{1}$$

e um conjunto factível Ω

$$\Omega = \left\{x \in \mathbb{R}^n / L_i \leq x_i \leq U_i\right\} \tag{2}$$

Então o problema de otimização fica do modo seguinte.

Achar $x^* \in \Omega$ tal que $f(x^*) \leq f(x) \quad \forall x \in \Omega$, é dizer:

$$x^* = \arg\min_{x \in \Omega} f(x) \tag{3}$$

A função f, é suposta ser duas vezes continuamente diferenciavel no conjunto factível Ω. O presente trabalho de dissertação tem como objetivo construir um algoritmo capaz de resolver o problema (3).
O trabalho está organizado em quatro capítulos. Este primeiro capítulo é a introdução e o último é a conclusão.

No segundo capítulo faremos a revisão bibliográfica. Como queremos apresentar o método trust-region (região de confiança) para resolver um problema de otimização com restrições simples, esse capítulo está subdividido como segue: falaremos de métodos de otimização para problemas de otimização sem restrições, faremos uma descrição dos métodos direção de busca e métodos trust-region, demonstraremos também a existência e unicidade da projeção de um ponto sobre um conjunto fechado e convexo e para finalizar o capítulo falaremos sobre critérios de otimalidade, condições que tem de cumprir o ponto ótimo.

No terceiro capítulo vamos apresentar o método do gradiente o qual é necessário para resolver os subproblemas gerados em cada iteração, apresentaremos também um novo algoritmo trust-region para problemas de otimização com restrições simples e finalmente faremos uma adaptação do algoritmo construído para o caso de otimização sem restrições.

1 Revisão Bibliográfica

Muitos métodos foram desenvolvidos para resolver problemas de otimização não linear, por exemplo, o método Trust-Region (Região de confiança) é um método que permite resolver este tipo de problemas. Os clásicos metodos Trust-Region fazem uso da norma euclideana para resolver o subproblema

$$s^k = \arg \min_{\|s\|_2 \leq \Delta_k} m^{(k)}(s)$$

onde $m^k(s)$ é uma função quadrática (ver [5] ou ver [3]) e obter assim a nova iteração. Em nosso caso faremos uso da norma $\|.\|_\infty$, o qual é outra maneira de trabalhar dos clásicos métodos Trust-Region.

Definição 1.1. *Seja $f : \mathbb{R}^n \to \mathbb{R}$. Uma direção $d \in \mathbb{R}^n$ é dita direção de descida para f em x^* se existe $\lambda > 0$ tal que $f(x^* + td) < f(x^*), \forall t \in (0, \lambda)$.*

Teorema 1.1. *Sejam $d \in \mathbb{R}^n$ e $f : k \subset \mathbb{R}^n \to \mathbb{R}$ uma função diferenciável em x^*. Se*

$$\langle d, \nabla f(x^*) \rangle < 0$$

então d é uma direção de descida para f em x^*.

Demonstração.

Dado que a função f é diferenciável em x^* então

$$f(x^* + td) = f(x^*) + t\langle \nabla f(x^*), d \rangle + r(td) \Rightarrow \frac{f(x^* + td) - f(x^*)}{t} = \langle \nabla f(x^*), d \rangle + \frac{r(td)}{t}$$

$$\frac{f(x^* + td) - f(x^*)}{t} = \langle \nabla f(x^*), d \rangle + \frac{r(td)}{|td|}(\pm|d|)$$

$$\text{Como} \quad \langle d, \nabla f(x^*) \rangle < 0 \quad \text{e} \quad \frac{r(td)}{|td|} \to 0, \quad \text{quando} \quad t \to 0$$

Então existe $\delta > 0$ tal que $f(x^* + td) < f(x^*)$, $\forall t \in (0, \delta)$.

Definição 1.2. (Hiperplano) *Seja $p \in \mathbb{R}^n$ um vetor diferente de 0 e α um escalar qualquer. Então o conjunto*

$$H = \{x \in \mathbb{R}^n : \langle p, x\rangle = \alpha\}$$

define um hiperplano.

A cada hiperplano correspondem os semi-espaços

$$H^+ = \{x \in \mathbb{R}^n : \langle p, x\rangle \geq \alpha\}$$

$$H^- = \{x \in \mathbb{R}^n : \langle p, x\rangle \leq \alpha\}$$

Definição 1.3. (Hiperplano Suporte) *Seja $C \subset \mathbb{R}^n$ um subconjunto não vazio, $p \in \mathbb{R}^n$ e $\overline{x} \in \partial C$,um ponto na fronteira de C se*

$$C \subset H^+ = \{x \in C : \langle p, x - \overline{x}\rangle \geq 0\} \quad ou$$

$$C \subset H^- = \{x \in C : \langle p, x - \overline{x}\rangle \leq 0\}$$

Então o hiperplano $H = \{x \in C : \langle p, x - \overline{x}\rangle = 0\}$ é chamado hiperplano suporte de C em $\overline{x}$.

Todo subconjunto convexo tem um hiperplano suporte em cada ponto da fronteira.

Teorema 1.2. *Seja $C \subset \mathbb{R}^n$ um subconjunto convexo e $\overline{x} \in \partial C$, então existe um hiperplano suporte em $\overline{x}$ isto é, existe um vetor não nulo p tal que*

$$\langle p, x - \overline{x}\rangle \leq 0 \quad \forall x \in \overline{C}$$

Definição 1.4. (Cone) *Um subconjunto $C \subset \mathbb{R}^n$ é um cone se:*

$$x \in C \Rightarrow (\alpha x) \in C \quad \forall \alpha \geq 0$$

Exemplo 1.0.1. *O conjunto $C = \{x = (\xi_1, \xi_2, ...\xi_n) : \xi_i \geq 0\}$, é um cone.*

Definição 1.5. *O cone gerado pelos vetores x e y é o conjunto definido por*

$$C = \{z : z = \alpha x + \beta y, \quad \forall \alpha, \beta \geq 0\}$$

Definição 1.6. (Cone Normal) *Seja C um subconjunto de $\mathbb{R}^n$ fechado e convexo, o cone normal de C em $\overline{x} \in \partial C$ é definido por:*

$$N(\overline{x}) = \{y \in \mathbb{R}^n : \langle y, x - \overline{x}\rangle \leq 0 \quad \forall x \in C\}$$

Teorema 1.3. *O cone normal $N(\overline{x})$ é fechado e convexo.*

Demonstração. Sejam $(d_k) \subset N(\overline{x})$ tal que $d_k \to d$. Suponhamos $d \notin N(\overline{x})$ então, existe K tal que $d_k \notin N(\overline{x})$, $\forall k \geq K$. $(\Rightarrow\Leftarrow)$.
Por tanto $N(\overline{x})$ é fechado.
Seja $d_1, d_2 \in N(\overline{x})$ e $t \in (0,1)$ então

$$\langle td_1 + (1-t)d_2, x - \overline{x}\rangle = t\langle d_1, x - \overline{x}\rangle + (1-t)\langle d_2, x - \overline{x}\rangle \leq 0 \quad \forall x \in C$$

Assim, $N(\overline{x})$ é um conjunto convexo.

1.1 Métodos de Direção de Busca

Os métodos de direção de busca trabalham como segue: a partir de um ponto inicial, com informações locais da função ao redor do ponto escolhemos uma direção na qual a função decresce, alí tomamos um múltiplo da direção escolhida para obter o passo ao novo ponto.

1.1.1 Estrutura Básica

Dado o problema irrestrito

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

e dado um ponto inicial $x_0 \neq x^*$, obtém-se uma sequência x_k tal que $x_k \to x^*$ a partir do algoritmo de otimização. A família dos algoritmos de direção de busca possui a estrutura

1.1.2 Algoritmo

Algoritmo 1: Algoritmo de Direção de Busca

inicio
  $k \leftarrow 0$
  enquanto *não criterio de parada* **faça**
    $d^k \leftarrow h(x^1, x^2, \cdots, x^k, f(x^1), f(x^2), \cdots, f(x^k))$
    $t_k \leftarrow \arg\min_t f(x^k + td^k)$
    $x^{k+1} \leftarrow x^k + t_k d^k$
    $k \leftarrow k + 1$
  fim
fin

Um algoritmo irá diferir do outro essencialmente pela maneira como é calculada a direção de busca d_k. Por exemplo no algoritmo gradiente tem-se simplesmente que $d_k = -\nabla f(x_k)$, no caso do algoritmo de Newton, tem-se que $d_k = -(\nabla^2 f(x_k))^{-1}\nabla f(x_k)$.

1.1.3 Método de descida mais rápida (Algoritmo gradiente)

A origem deste método é atribuído ao matemático françês Cauchy (1789-1857) e tem base no seguinte teorema.

Teorema 1.4. *Suponha que f seja uma função diferenciável. O valor máximo da derivada direcional $D_u f(x^*)$ é $\|\nabla f(x^*)\|$ e ocorre quando u tem a mesma direção e sentido que o vetor gradiente $\nabla f(x^*)$.*

Demonstração.
Seja $u \in \mathbb{R}^n$ tal que $\|u\| = 1$, então temos

$$D_u f(x^*) = \langle \nabla f(x^*), u \rangle = \|\nabla f(x^*)\| \cos\theta$$

onde θ é o ângulo entre $\nabla f(x^*)$ e u. O valor máximo de $\cos\theta$ é 1, e isso ocorre quando $\theta = 0$. Portanto o valor máximo de $D_u f(x^*)$ é $\|\nabla f(x^*)\|$ e ocorre quando $\theta = 0$, ou seja quando u tem a mesma direção e sentido que $\nabla f(x^*)$.

■

O método de descida mais rápida considera como direção de descida $d = -\nabla f(x)$ pois dentre as direções ao longo das quais f decresce, a direção oposta ao gradiente é a de decrescimento mais acentuado. De fato se $d = -\nabla f(x)$ é tal que $\|v\| = \|d\|$, então temos

$$D_d f(x) = \langle \nabla f(x), d \rangle = -\|\nabla f(x)\|^2 = -\|\nabla f(x)\|\|\nabla f(x)\|$$

$$-\|\nabla f(x)\|\|\nabla f(x)\| = -\|\nabla f(x)\|\|v\| \leq \nabla f(x)v = D_v f(x)$$

Portanto

$$D_d f(x) \leq D_v f(x)$$

Teorema 1.5. *O método de descida mais rápida move-se segundo direções perpendiculares de passo para passo. Mais precisamente, sendo $\{x^k\}$ uma sequência de pontos obtidos pelo método para cada $k \in \mathbb{N}$ tem-se*

$$\langle x^k - x^{k+1}, x^{k+1} - x^{k+2} \rangle = 0$$

Demonstração. Uma vez que $x^{k+1} = x^k - t_k \nabla f(x^k)$, vem que

$$\langle x^{k+1} - x^k, x^{k+2} - x^{k+1} \rangle = t_k t_{k+1} \langle \nabla f(x^k), \nabla f(x^{k+1}) \rangle$$

e dado que t_k minimiza a função unidimensional $\varphi_k(t) = f(x^k - t\nabla f(x^k))$, então

$$0 = \varphi_k^{'}(t_k) = -\langle \nabla f(x^k - t_k \nabla f(x^k)), \nabla f(x^k) \rangle = -\langle \nabla f(x^{k+1}), \nabla f(x^k) \rangle$$

assim :

$$\langle x^k - x^{k+1}, x^{k+1} - x^{k+2} \rangle = 0 \quad \forall k \in \mathbb{N}$$

■

Teorema 1.6. *Se $\{x^k\}$ é uma sequência de pontos obtidas por aplicação do método do gradiente à minimização da função diferenciável $f : \mathbb{R}^n \to \mathbb{R}$, e se para algum $k \in \mathbb{N}$, $\nabla f(x^k) \neq 0$, então $f(x^{k+1}) < f(x^k)$.*

Demonstração. Dado que $x^{k+1} = x^k - t_k \nabla f(x^k)$, onde t_k é o minimizante da função

$$\varphi_k(t) = f(x^k - t\nabla f(x^k))$$

Então, tem-se que $f(x^{k+1}) = \varphi_k(t_k) \leq \varphi_k(t)$, $\forall t \geq 0$. Por outro lado temos que $\varphi_k^{'}(t) = -\langle \nabla f(x^k - t\nabla f(x^k)), \nabla f(x^k) \rangle$, então

$$\varphi_k^{'}(0) = -\langle \nabla f(x^k - 0\nabla f(x^k)), \nabla f(x^k) \rangle = -\|\nabla f(x^k\|^2 < 0$$

e, consequentemente, $\exists \epsilon > 0$ tal que $\forall t \in (0, \epsilon)$, $\varphi_k(t) < \varphi_k(0) = f(x^k)$. Logo

$$f(x^{k+1}) = \varphi_k(t_k) \leq \varphi_k(t) < \varphi_k(0) = f(x^k)$$

■

1.2 Otimização Unidimensional

A seguinte linha do algoritmo do gradiente agora é examinada

$$t_k \leftarrow \arg\min_{t \geq 0} f\big(x^k - t\nabla f(x^k)\big) \tag{1.2}$$

O cálculo de t_k é feito fixando o ponto atual x^k e uma direção de busca $d^k = -\nabla f(x^k)$. Isso faz que a função objetivo, f, que originalmente seria de n variáveis torne-se agora uma função de uma única variavel real t.

1.2.1 Método da Seção Áurea-Busca exata

Definição 1.7. *Uma função contínua $\varphi : [0, \infty) \to \mathbb{R}$ é dita unimodal quando admite um conjunto de minimizadores $[t_1, t_2] \subset [0, \infty)$, é estritamente decrescente em $[0, t_1]$ e estritamente crescente em $[t_2, \infty)$.*

Veja os exemplos de funções unimodais, o intervalo de minimizadores $[t_1, t_2]$ pode ser degenerado $t_1 = t_2$.(ver a figura 1.1).

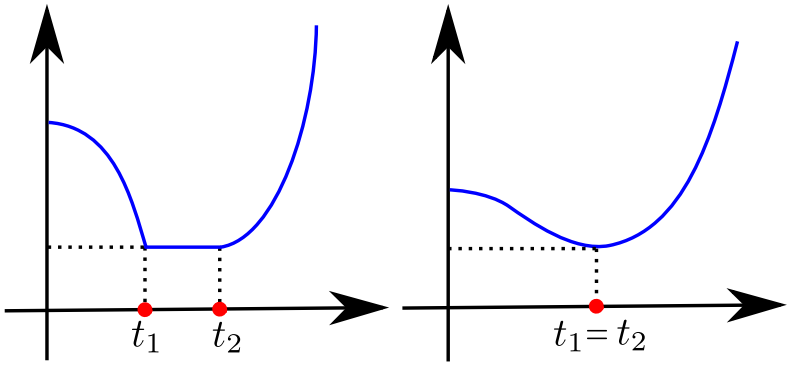

Figura 1.1: Funções Unimodais.

Para a descrição do algoritmo da seção aurea vamos considerar a função φ com um único minizador.

Descrição do Algoritmo

Suponha que o minimizador da função $\varphi : [a, b] \to \mathbb{R}$ pertence ao intervalo $[a, b]$

i Considere $a < u < v < b$

ii Se $\varphi(u) < \varphi(v)$, então o trecho $(v, b]$ não pode conter o minimizador e pode ser descartado.

iii Se $\varphi(u) > \varphi(v)$ então o trecho $[a, u)$ pode ser descartado

iv Particione o intervalo que ficou e repita o processo.

Agora vamos analizar como o intervalo $[a, b]$ deve ser particionado.

Definição 1.8. *Um ponto c divide o segmento $[a, b]$ na razão áurea quando a razão entre o maior segmento e o segmento todo é igual à razão entre o menor e o maior dos segmentos. Tal razão é conhecida como o número de ouro e vale $\frac{\sqrt{5}-1}{2}$.*

Desta forma, temos que u e v devem satisfazer

$$\frac{b-u}{b-a} = \frac{u-a}{b-u} \quad e \quad \frac{v-a}{b-a} = \frac{b-v}{v-a}$$

Considerando $\theta_1 \quad e \quad \theta_2$ tais que

$$u = a + \theta_1(b-a) \quad v = a + \theta_2(b-a) \quad \theta_1, \theta_2 \in [0,1]$$

Substituindo temos:

$$\frac{b-(a+\theta_1(b-a))}{b-a} = \frac{a+\theta_1(b-a)-a}{b-(a+\theta_1(b-a))}$$

Desta forma, obtemos:

$$1-\theta_1 = \frac{\theta_1}{1-\theta_1}$$

Analogamente

$$\theta_2 = \frac{1-\theta_2}{\theta_2}$$

Encontramos assim $\theta_1 = \frac{3-\sqrt{5}}{2} =\approx 0.382$ e $\theta_2 = \frac{\sqrt{5}-1}{2} =\approx 0.618$.
Podemos observar que $\theta_1 + \theta_2 = 1$ e $\theta_2^2 = \theta_1$, assim podemos escolher $u \quad e \quad v$ da seguinte maneira

$$u = a + 0.382(b-a) = b - 0.618(b-a)$$

$$v = a + 0.618(b-a)$$

Uma das vantagens da divisão na razão áurea é que podemos aproveitar o ponto u ou v após termos descartado o intervalo $[v,b]$ ou $[a,u]$ na iteração anterior. Indicamos por $[a^+, b^+]$ o novo intervalo que será particionado pelos pontos u^+ e v^+. Conforme veremos no próximo resultado, o ponto v é aproveitado na próxima etapa e passa a ser u^+ quando descartamos $[a,u)$ e assim o valor da função $\varphi(v)$ é aproveitado para a próxima etapa.

Lema 1. *Na seção áurea se $(v,b]$ é descartado então $v^+ = u$.*

Com efeito, como $(v,b]$ é descartado então $a^+ = a$ e $b^+ = v$. Logo temos

$$\begin{aligned} v^+ &= a^+ + \theta_2(b^+ - a^+) = a + \theta_2(v-a) \\ &= a + \theta_2\big(a + \theta_2(b-a) - a\big) \\ &= a + \theta_2^2(b-a) \\ &= a + \theta_1(b-a) = u \end{aligned}$$

■

Lema 2. *Na seção áurea se $[a, u)$ é descartado então $u^+ = v$.*

Com efeito, como $[a, u)$ é descartado então $a^+ = u$ e $b^+ = b$. Logo, temos

$$\begin{aligned} u^+ &= a^+ + \theta_1(b^+ - a^+) = u + \theta_1(b - u) \\ &= a + \theta_1(b - a) + \theta_1\big(b - (a + \theta_1(b - a))\big) \\ &= a + 2\theta_1(b - a) - \theta_1^2(b - a) \\ &= a + (2\theta_1 - \theta_1^2)(b - a) \\ &= a + (2\theta_1 - 3\theta_1 + 1)(b - a) \\ &= a + (1 - \theta_1)(b - a) = a + \theta_2(b - a) = v \end{aligned}$$

■

1.2.2 Algoritmo

Algoritmo 2: Seção Áurea

inicio

Entrada: $\rho > 0$ e $\epsilon > 0$

Etapa 1 Obtenção do intervalo $[a, b]$

$a^0 = 0, s^0 = \rho, b^0 = 2\rho$

$k = 0$

enquanto $\varphi(b^k) < \varphi(s^k)$ **faça**

$a^{k+1} = s^k$

$s^{k+1} = b^k$

$b^{k+1} = 2b^k$

$k = k + 1$

fim

$a = a^k$ e $b = b^k$

Etapa 2 Obtenção de $t^* \in [a, b]$

$a^0 = a, b^0 = b$

$u^0 = a^0 + \theta_1(b^0 - a^0)$

$v^0 = a^0 + \theta_2(b^0 - a^0)$

$k = 0$

enquanto $b^k - a^k > \epsilon$ **faça**

se $\varphi(u^k) < \varphi(v^k)$ **então**

$a^{k+1} = a^k$

$b^{k+1} = v^k$

$v^{k+1} = u^k$

$u^{k+1} = a^{k+1} + \theta_1(b^{k+1} - a^{k+1})$

senão

$a^{k+1} = u^k$

$b^{k+1} = b^k$

$u^{k+1} = v^k$

$v^{k+1} = a^{k+1} + \theta_2(b^{k+1} - a^{k+1})$

fim

$k = k + 1$

fim

Defina $\boldsymbol{t^* = \frac{u^k + v^k}{2}}$

fin

O teorema seguinte mostra que após um número finito de etapas é possível encontrar um intervalo $[a, b]$ que contém pelo menos um minimizador.

Teorema 1.7. *O algoritmo da Seção Áurea, na Etapa 1, encontra um intervalo $[a, b]$ que contém pelo menos um minimizador da função em um número finito de iterações.*

Demonstração. Mostremos que a etapa 1 do algoritmo é finita.
Suponhamos por absurdo que $\varphi(b^k) < \varphi(s^k)$, para todo $k \in \mathbb{N}$, então $s^k \to \infty$ pois $s^{k+1} = b^k = 2b^{k-1} = 2s^k$. Assim, existe $k \in \mathbb{N}$ tal que $s^k \geq t_1$, $t_1 > 0$. Como a função é unimodal ela é não decrescente no intervalo $[t_1, \infty)$. Logo $\varphi(b^k) \geq \varphi(s^k)$, obtendo assim uma contradição. Portanto fase 1 do algoritmo termina em um certo $k \in \mathbb{N}$. Ainda está faltando mostrar que o intervalo obtido contém um minimizador de φ.

i Se $s^k < t_1$, temos $b^k > t_2$, pois do contrário teríamos $\varphi(b^k) < \varphi(s^k)$ e assim

$$[t_1, t_2] \subset [s^k, b^k] \subset [a^k, b^k]$$

ii Se $s_k \geq t_1$ então $a^k < t_1$.
Com efeito, $a_k = s_{k-1}$ para $k > 0$ e suponhamos que $a_k = s_{k-1} \geq t_1$, então teríamos $\varphi(b_{k-1}) \geq \varphi(s_{k-1})$ e assim a fase 1 teria terminado na iteração $k - 1$ ao invés da iteração k. Tendo assim $a^k < t_1$ e portanto $[t_1, t_2] \subset [a_k, b_k]$.

■

O teorema a seguir analisa a Etapa 2 do algoritmo ao descartar um dos intervalos com a premissa que a função objetivo possua um único mínimo local no domínio em questão.

Teorema 1.8. *Seja $\varphi : \mathbb{R} \to \mathbb{R}$, uma função contínua unimodal. Seja o conjunto $[a, b] \subset \mathbb{R}$ no qual φ possui um único mínimo local x^*, sejam ainda dois pontos u e v, tais que $a < u < v < b$. Se ocorrer*

$$\varphi(u) < \varphi(v) \tag{1.3}$$

então a solução minimizante x^* não se encontra no intervalo $(v, b]$ e se ocorrer

$$\varphi(u) > \varphi(v) \tag{1.4}$$

então a solução minimizante x^*, não se encontra no intervalo $[a, u)$.

*Demonstração.*Tome-se o intervalo $[a, v]$, pelo fato que a φ é contínua nesse intervalo vai existir x_0 para o qual $\varphi(x_0) \leq \varphi(x), \forall x \in [a, v]$ e pela hipótese (1.3) temos $x_0 \neq v$,

logo x_0 é um mínimo local no segmento $[a, v]$. Como $x_0 \neq v$ tem-se que no intervalo $[a, b]$, x_0 permanece ainda sendo mínimo local e como há um único mínimo local no intervalo $[a, b]$, obtém-se que $x^* = x_0$, para o outro lado do segmento o argumento é análogo.

■

Teorema 1.9. *Seja $[a_k, b_k]$ o intervalo obtido pelo algoritmo da seção áurea, então $b_k - a_k \to 0$*

Demonstração. Seja o intervalo inicial $[a_0, b_0]$ e suponhamos que o intervalo $[a_0, u_0)$ é descartado, então na primera iteração conseguimos achar $[a_1, b_1]$ e assim temos que

$$b_1 - a_1 = b_0 - u_0 = b_0 - (a_0 + \theta_1(b_0 - a_0)) = (1 - \theta_1)(b_0 - a_0) = \theta_2(b_0 - a_0)$$

Na segunda iteração obtemos o intervalo$[a_2, b_2]$ e assim

$$\begin{aligned} b_2 - a_2 = b_1 - u_1 &= b_1 - (a_1 + \theta_1(b_1 - a_1)) \\ &= (1 - \theta_1)(b_1 - a_1) = \theta_2(b_1 - a_1) \\ &= (b_0 - a_0)\theta_2^2 \end{aligned}$$

Repetindo o processo com o intervalo de cada iteração, obtemos

$$b_k - a_k = (b_0 - a_0)\theta_2^k$$

Como $\theta_2 \in (0, 1)$ então

$$\lim_{k\to\infty}(b_k - a_k) = 0$$

.

■

1.3 Descrição do Método Trust-Region

O método de Trust-Region (região de confiança) é uma estratégia iterativa para solução de problemas de otimização com o objetivo de obter convergência global e aproveitar, ao máximo, as propriedades de convergência local dos métodos de busca linear. Para o caso de minimização irrestrita, a estratégia consiste na definição de uma região de raio Δ^k em torno do ponto corrente x^k e então encontrar o minimizador (aproximado) s^k de um modelo quadrático que vai aproximar à função objetivo em tal região. O minimizador encontrado define então um passo para um novo ponto $x^{k+1} = x^k + s^k$ que objetiva

o decrescimento da função a partir de x^k. Se isso realmente acontecer, então o modelo representou bem a função objetivo na região definida. Caso não haja o decrescimento da função objetivo no novo ponto x^{k+1}, isso significa que o modelo não representou bem à função objetivo na região dada; esta região é então reduzida para a próxima iteração. A região é definida de acordo com a boa representação do modelo com relação à função objetivo: seu tamanho varia de acordo com o descréscimo da função objetivo nas iterações anteriores. Por isso, os métodos de região de confiança são comparados aos métodos de busca linear, visto que ambos visam gerar um passo que minimize dada função objetivo a partir de um ponto corrente. A principal diferença é que a estratégia de busca linear encontra uma direção e, a partir desta direção, concentra seus esforços em encontrar um tamanho ótimo para o passo, enquanto a estratégia de região de confiança estabelece a direção e o tamanho do passo simultaneamente a partir da minimização do modelo quadrático sujeito à região de raio Δ^k; de fato, a direção e o passo mudam sempre que o tamanho da região é alterado. Para ilustrar o caso, ver a figura (1.2).

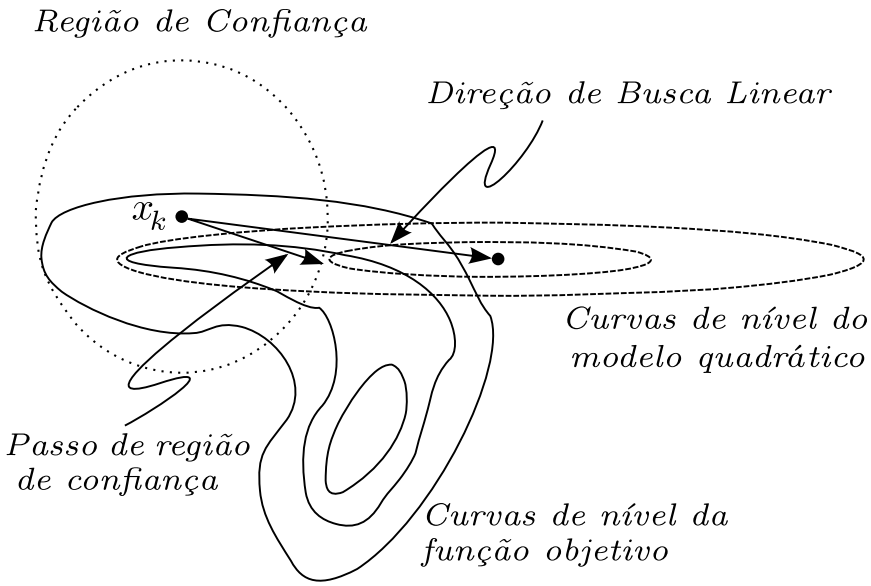

Figura 1.2: Passo da região de confiança e direção de busca linear.

O modelo local de $f(x)$ na $k-$ésima iteração pode ser escrito como uma aproximação quadrática

$$f(x) \approx m(x) = f(x^k) + \langle g^k, x - x^k \rangle + \frac{1}{2}\langle x - x^k, H^k(x - x^k)\rangle$$

onde $g^k = \nabla f(x^k)$ e H^k é uma aproximação da matriz hessiana no ponto x^k. A região de confiança pode ser definida pela expressão:

$$T^k = \left\{ x \in \mathbb{R}^n / \|x - x^k\| \leq \Delta^k \right\}$$

É óbvio que as diferentes escolhas para a norma conduzem a formas diferentes de região de confiança. A norma euclidiana $\|.\|_2$ corresponde a um hiper-esfera, enquanto que a norma $\|.\|_\infty$ define uma hiper-caixa.

1.3.1 Algoritmo

Algoritmo 3: Algoritmo Básico Trust-Region

inicio

Entrada: x^0 e raio $\Delta^0 > 0$, constantes $0 < \mu < \eta < 1$

para *k=0,1...* **hacer**

i. **Se** x^k é ótimo, parar

ii. **Resolver**

$$s^k = \arg \min_{\|s\| \leq \Delta_k} m^{(k)}(s) \tag{1.5}$$

iii. **Calcular**

$$\rho^k = \frac{f(x^k) - f(x^k + s^k)}{m(x^k) - m(x^k + s^k)} \tag{1.6}$$

iv. **se** $\rho^k \leq \mu$ **então**
$x^{k+1} = x^k$

senão
$x^{k+1} = x^k + s^k$

fim

v. **Atualizar** $\boldsymbol{\Delta^k}$

se $\rho^k \leq \mu$ **então**
$\Delta^{k+1} = \frac{1}{2}\Delta^k$

senão
se $\mu < \rho^k < \eta$ **então**
$\Delta^{k+1} = \Delta^k$

senão
$\Delta^{k+1} = 2\Delta^k$

fim

fim

fin

fin

O algoritmo descrito anteriormente é usado para encontrar uma aproximação da solução de um problema de otimização sem restrições, mas o trabalho nosso é adaptar o algoritmo básico para encontrar uma aproximação da solução ao problema com restrições da forma $L_i \leq x_i \leq U_i \quad i = 1 \cdots n$ (problema (3)), já que a partir de um ponto x^k na região factível o novo ponto x^{k+1} poderia ficar fora da região factível. Para garantir isso vamos apresentar algumas definições.

Definição 1.9. *Sejam $x \in \mathbb{R}^n$, e os vetores L , U vetores dados em (1), definimos*

$$I(x, L, U) = \Big\{ i \in \mathbb{N} / x(i) = L(i) \quad ou \quad x(i) = U(i) \Big\} \tag{1.7}$$

O conjunto de índices definido anteriormente é chamado conjunto de índices ativos de x. Se for o caso que $i \notin I(x, L, U)$, então diz-se que a $i-$ésima restrição é inativa.

1.4 Projeção Sobre um Conjunto Convexo e Fechado

Teorema 1.10. *Seja $K \subset \mathbb{R}^n$ um subconjunto não vazio fechado e convexo. Então para cada $x \in \mathbb{R}^n$ existe um único $u \in K$ tal que*

$$\|x - u\| = \min_{v \in K} \|x - v\| = \inf_{v \in K} \|x - v\| \tag{1.8}$$

Além disso u é caracterizado pela seguinte propriedade:

$$u \in K \text{ e } \langle x - u, y - u \rangle \leq 0, \ \forall y \in K \tag{1.9}$$

Prova:
Seja $\{v_n\}$ uma sequência que minimiza (1.8), ou seja $v_n \in K$ e

$$d_n = \|x - v_n\| \to d = \inf_{v \in K} \|x - v\|$$

Demonstraremos que $\{v_n\}$ é uma sequência de Cauchy em K.
Tomando $a = x - v_n$, $b = x - v_m$ e fazendo uso da lei do paralegramo

$$\|\frac{a+b}{2}\|^2 + \|\frac{a-b}{2}\|^2 = \frac{1}{2}\Big(\|a\|^2 + \|b\|^2\Big)$$

Obtemos

$$\|x - \frac{v_n + v_m}{2}\|^2 + \|\frac{v_n - v_m}{2}\|^2 = \frac{1}{2}\Big(\|d_n\|^2 + \|d_m\|^2\Big)$$

Como $\frac{v_n+v_m}{2} \in K$ então $\|x - \frac{v_n+v_m}{2}\| \geq d$

$$\|\frac{v_n - v_m}{2}\|^2 = \frac{1}{2}\Big(\|d_n\|^2 + \|d_m\|^2\Big) - \|x - \frac{v_n + v_m}{2}\|^2$$
$$\leq \frac{1}{2}(d_n^2 + d_m^2) - d^2$$

e assim obtemos que $\lim_{n,m\to\infty} \|v_n - v_m\| = 0$. Ou seja, v_n é uma sequência de Cauchy em K, pela completude de $\mathbb{R}^n$, e como K é fechado, existe $u \in K$ tal que $v_n \to u$, além disso

$$\|x - u\| = \|x - \lim_{n\to\infty} v_n\| = \lim_{n\to\infty} \|x - v_n\| = \inf_{v\in K} \|x - v\|$$

Para unicidade, basta supor que existem u_1 e u_2 que cumprem

$$\langle x - u_1, v - u_1\rangle \leq 0, \ \forall v \in K \tag{1.10}$$

$$\langle x - u_2, v - u_2\rangle \leq 0, \ \forall v \in K \tag{1.11}$$

tomando $v = u_2$ em (1.10) e $v = u_1$ em (1.11) obtemos

$$\langle x - u_1, u_2 - u_1\rangle \leq 0$$

$$\langle x - u_2, u_1 - u_2\rangle \leq 0$$

assim

$$\|u_1 - u_2\|^2 \leq 0 \Rightarrow u_1 = u_2$$

Para demonstrar a desigualdade (1.9) usaremos o fato de que K é convexo. Como

$$\|x - u\| \leq \|x - z\| \ \forall z \in K$$

para qualquer $t \in (0, 1)$, tomamos $z = (1 - t)u + ty$, para qualquer $y \in K$

$$\|x - u\| \leq \|x - (1 - t)u + ty\| = \|(x - u) - t(y - u)\|$$

assim

$$\|x - u\|^2 \leq \|(x - u) - t(y - u)\|^2 = \|x - u\|^2 - 2\langle x - u, t(y - u)\rangle + t^2\|y - u\|^2$$

$$2t\langle x - u, y - u\rangle \leq t^2\|y - u\|^2$$

Fazendo a divisão por t na desigualdade anterior e fazendo que $t \to 0$ obtemos

$$u \in K \text{ e } \langle x - u, y - u\rangle \leq 0, \ \forall y \in K$$

■

Lema 3. *A desigualdade (1.9) é equivalente a (1.8).*

Prova:
Só falta mostrar que se $\langle x-u, y-u\rangle \leq 0$, para qualquer $y \in K$, então $\|x-u\| \leq \|x-y\|,\ \forall y \in K$.
Para qualquer $y \in K$, tem-se

$$\|x-u\|^2 = \|x-y+y-u\|^2 = \|x-y\|^2 + 2\langle x-y, y-u\rangle + \|y-u\|^2$$

$$= \|x-y\|^2 + 2\langle x-u+u-y, y-u\rangle + \|y-u\|^2$$

$$= \|x-y\|^2 + 2\langle x-u, y-u\rangle - \|y-u\|^2 \leq \|x-y\|^2$$

$$\Longrightarrow \|x-u\|^2 \leq \|x-y\|^2$$

assim

$$\|x-u\| \leq \|x-y\|, \forall y \in K.$$

■

O elemento $u \in K$, é chamado projeção de x sobre K e é denotado por $P_K[x]$, o elemento u vai estar dado por

$$u = P_K[x] = \arg\min\left\{\|v-x\| : v \in K\right\}$$

Teorema 1.11. *O operador projeção $P : \mathbb{R}^n \longrightarrow K$ (fechado e convexo) é um operador não expansivo,*

$$\|P_K[x] - P_K[y]\| \leq \|x-y\|,\ \forall x, y \in \mathbb{R}^n. \tag{1.12}$$

Prova:
Como

$$\langle x - P_K[x], w - P_K[x]\rangle \leq 0, \forall w \in K$$

Então

$$\langle x - P_K[x], P_K[y] - P_K[x]\rangle \leq 0$$

$$\langle y - P_K[y], P_K[x] - P_K[y]\rangle \leq 0$$

assim obtemos

$$\langle x - P_K[x], P_K[y] - P_K[x]\rangle + \langle y - P_K[y], P_K[x] - P_K[y]\rangle \leq 0$$

$$\langle x - P_K[x] - y + P_K[y], P_K[y] - P_K[x] \rangle \leq 0$$

$$\langle x - y, P_K[y] - P_K[x] \rangle + \langle P_K[y] - P_K[x], P_K[y] - P_K[x] \rangle \leq 0$$

$$\|P_K[y] - P_K[x]\|^2 \leq \langle y - x, P_K[y] - P_K[x] \rangle \leq \|y - x\| \|P_K[y] - P_K[x]\|$$

$$\|P_K[y] - P_K[x]\| \leq \|y - x\|, \; \forall x, y \in \mathbb{R}^n.$$

■

No caso em que o conjunto K for o conjunto Ω dado em (2), o operador $P_\Omega[.]$ no ponto $x = (x_1, x_2, \cdots, x_n)$, ficará do modo seguinte:

$$P_\Omega[x] = \begin{cases} L_i & se \quad x_i \leq L_i \\ x_i & se \quad L_i < x_i < U_i \\ U_i & se \quad x_i \geq U_i \end{cases} \tag{1.13}$$

Para ilustrar o caso em que $K = \Omega$, observar a figura (1.3).

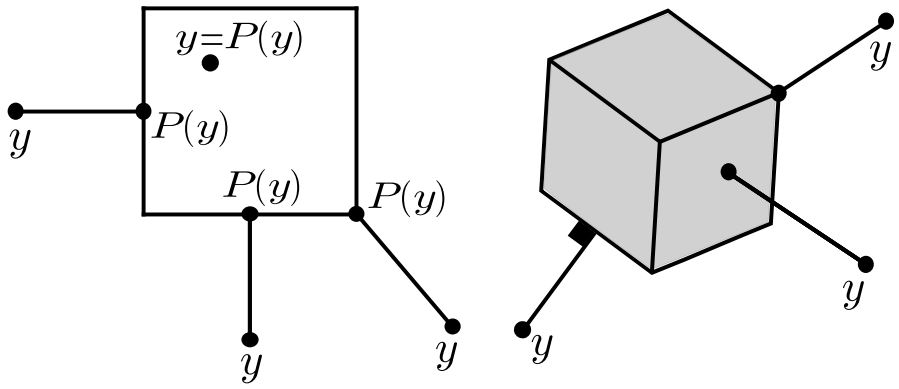

Figura 1.3: Projeção em duas e três dimensões.

A condição de otimalidade $\nabla f(x) = 0$ já não é válida quando se tenta resolver um problema com restrições, agora precisamos encontrar outras condições de otimalidade aplicáveis ao problema (3) e assim detectar os pontos candidatos para ser a solução ótima.

1.5 Condição de otimalidade

Definição 1.10. *Um ponto $x^* \in \Omega$, é chamado estacionário para o problema (3) se*

$$\langle \nabla f(x^*), x - x^* \rangle \geq 0, \; \forall x \in \Omega \tag{1.14}$$

Teorema 1.12. *Seja K um conjunto convexo, $f : K \subset \mathbb{R}^n \to \mathbb{R}$ uma função continuamente diferenciavel, se x^* é um mínimo local da função f em K, então*

$$\langle \nabla f(x^*), x - x^* \rangle \geq 0, \ \forall x \in K. \tag{1.15}$$

Prova:

Suponhamos que

$$\langle \nabla f(x^*), x - x^* \rangle < 0, \text{ para algúm } x \in K$$

Como o conjunto K é convexo definamos a função $g : [0, 1] \to \mathbb{R}$

$$g(t) = f(x^* + t(x - x^*))$$

g é contínua em $[0, 1]$, e diferenciável em $(0, 1)$, então pelo teorema do valor médio

$$g(1) - g(0) = g'(\theta), \text{ para algum } \theta \in (0, 1)$$

$$f(x) - f(x^*) = \langle \nabla f(x^* + \theta(x - x^*))(x - x^*) \rangle$$

Como

$$\langle \nabla f(x^* + \theta(x - x^*))(x - x^*) \rangle < 0$$

temos que

$$f(x) < f(x^*)$$

e obtemos a contradição desejada pelo fato que x^* é mínimo. ■

Para ilustrar o teorema (ver figura 1.4).

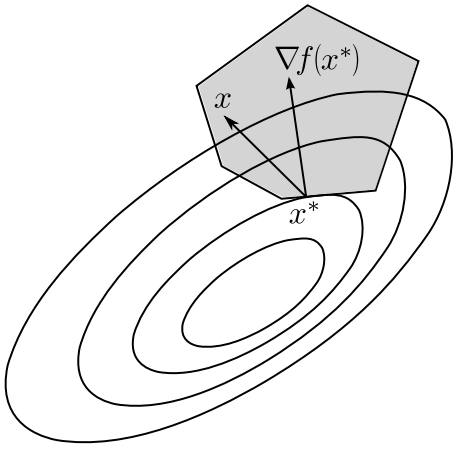

Figura 1.4: O gradiente $\nabla f(x^*)$ forma um ângulo menor ou igual a 90 com todas as variações factíveis $x - x^*$, $x \in K$.

Observação: O teorema acima apresentado não é válido se o conjunto não for convexo.

Para corroborar isso basta observar a figura 1.5 .

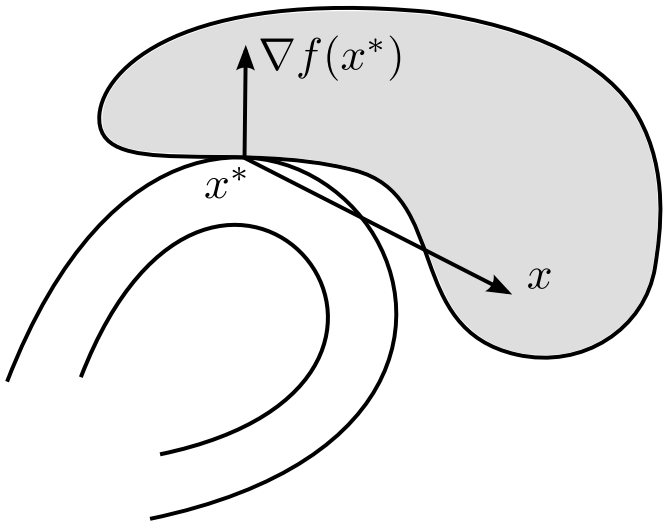

Figura 1.5: Ilustração da condição de otimalidade quando K não é convexo neste caso o teorema acima apresentado não é verdade.

Teorema 1.13. *Seja f uma função continuamente diferenciável, então um ponto x^* é estacionário para (3) se e somente se*

$$x^* = P_\Omega(x^* - \lambda \nabla f(x^*)), \ \forall \lambda > 0. \tag{1.16}$$

Prova:

Seja x^* um ponto estacionário e $\lambda > 0$ um escalar qualquer, então

$$\langle \nabla f(x^*), x - x^* \rangle \geq 0 \iff$$

$$\lambda \langle \nabla f(x^*), x - x^* \rangle \geq 0 \iff$$

$$\langle (x^* - \lambda \nabla f(x^*)) - x^*, x - x^* \rangle \leq 0 \iff$$

$$x^* = P_\Omega(x^* - \lambda \nabla f(x^*)), \ \forall \lambda > 0.$$

■

2 *Método Trust-Region fazendo uso do gradiente projetado*

2.1 Método Gradiente Projetado

O método do gradiente projetado é um bom método para problemas de otimização com restrições convexas e fechadas já que ele vai garantir que o novo ponto obtido a partir de um ponto inicial continue ainda sendo factível.
Seja $P : \mathbb{R}^n \to \Omega$ a projeção sobre o conjunto factível Ω e seja x^k um ponto no interior do conjunto factível, então o novo será dado por

$$x^{k+1} = P_\Omega[x^k - s^k \nabla f(x^k)] \tag{2.1}$$

onde s^k é um escalar positivo (ver figura 2.1).

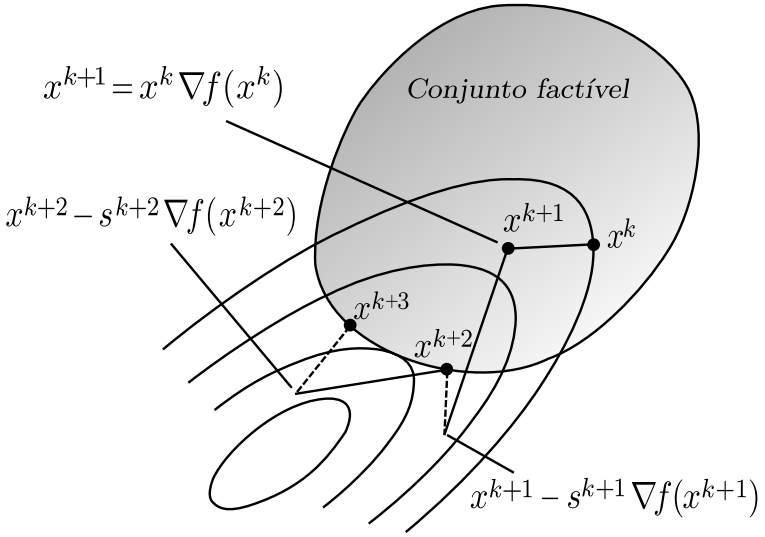

Figura 2.1: Método gradiente projetado.

Note que se $x^k - s^k \nabla f(x^k) \in \Omega$, a iteração converte-se no algoritmo do gradiente para

otimização sem restrições.

Há muitas variantes do método gradiente projetado, que utilizam diferentes maneiras de escolher a constante s^k.
Para nosso caso o gradiente projetado será achado a cada vez que uma das componentes atinge uma das cotas e vai ser obtido projetando o gradiente da função objetivo sobre o plano gerado pelas restrições ativas.

Suponhamos que estamos na iteração k, ou seja, temos o ponto x^k, o gradiente $g^k = \nabla f(x^k)$ e uma aproximação H^k da matriz hessiana da função f no ponto atual. Além disso, temos o raio Δ^k o qual é uma cota sobre os deslocamentos do novo ponto x^{k+1}. Este novo ponto é obtido como uma aproximação da solução do problema

$$\min_{\substack{\|x-x^k\|\leq\Delta^k \\ L\leq x\leq U}} m^k(x) \tag{2.2}$$

onde $\|.\|$ é uma norma, neste trabalho é conveniente escolher a norma infinito pelo fato de que as restrições $\|x - x^k\|_\infty \leq \Delta^k$ são lineares e assim é mais fácil verificar se um ponto é factível ou não, além disso como o problema também tem a restrição $L \leq x \leq U$, então para garantir que o novo ponto é factível, basta dizer que ele se encontra na interseção das duas caixas, ou seja cada x_i deve cumprir

$$\max\left(L_i, x_i^k - \Delta^k\right) \leq x_i \leq \min\left(U_i, x_i^k + \Delta^k\right), \ \forall i = 1, 2, \ldots, n \tag{2.3}$$

A aproximação da solução do problema (2.2) será feita da seguinte maneira: a partir de um ponto x^k no interior da caixa (2.3), trabalhamos como no caso sem restrições e fazemos uso do método do gradiente, se o novo ponto obtido x^{k+1} está no interior da caixa e não é ótimo, então continuamos como se fosse otimização sem restrições.
Se x^{k+1} está na fronteira da caixa, então não podemos usar o vetor gradiente como direção de busca, pois o novo ponto poderia ser não factível, neste caso faremos uso do gradiente projetado e para fazer isso definamos

$$x(t) = P_\Omega(x^k - tg^k)$$

O minimizador é obtido através da análise de cada pedaço de segmento $x(t)$. Para realizar a busca deste minimizador é preciso determinar os valores de $\bar{t}_i$ em que cada componente atinge uma cota quando estamos nos movendo na direção $-g$. Os valores de

$\bar{t}_i$ podem ser calculados do modo seguinte

$$\bar{t}_i = \begin{cases} \frac{x_i - U_i}{g_i} & se \quad g_i < 0 \\ \\ \frac{x_i - L_i}{g_i} & se \quad g_i > 0 \end{cases} \tag{2.4}$$

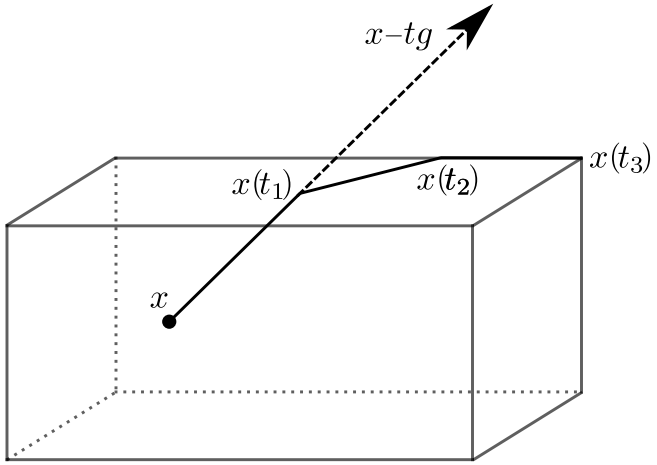

Figura 2.2: Projeção de $x - t\nabla f(x)$, sobre a caixa.

assim o caminho de busca torna-se em cada componente.

$$x_i(t) = \begin{cases} x_i - tg_i & se \qquad t < \bar{t}_i \\ \\ x_i - \bar{t}_i g_i & \text{outro caso} \end{cases} \tag{2.5}$$

Os valores g_i são os gradientes projetados e serão calculados cada vez que uma das componentes atinge uma das cotas.

Suponhamos que para $\bar{t}_j$, a variável b atinge uma das faces da caixa, seja x_j o ponto na fronteira da caixa, então o gradiente projetado neste ponto será dado por

$$g_p = \nabla f(x_j) - \Big(\nabla f(x_j)\Big)_b e_b \tag{2.6}$$

onde e_b é o vetor dado por

$$\Big(e_b\Big)_i = \begin{cases} 1 \quad se \qquad i = b \\ \\ 0 \qquad \text{outro caso} \end{cases}$$

Na busca do minizador através de $x(t)$, nas faces da caixa, nós eliminamos os valores

duplicados e valores zero do conjunto $\left\{\bar{t}_1, \bar{t}_2, \bar{t}_3, \ldots \bar{t}_l\right\}$ e obtemos o conjunto $\left\{t_1, t_2, t_3, \ldots t_s\right\}$ onde $0 < t_i < t_j, \ \ \forall i \neq j$.

O critério de aceitação para escolher o novo ponto e a modificação do raio Δ^k é a mesma dada no algoritmo básico.

2.1.1 Algoritmo

Algoritmo 4: Algoritmo Trust-Region Projetado

inicio

Entrada: $g_k \leftarrow \nabla f(x^k)$, raio $\Delta^k > 0$, constantes $0 < \mu < \eta < 1$, restrições L, U.

Se x^k é ótimo, parar

para *i=1:n* **hacer**

Encontrar

$l(i) = \max[L(i), x^k - \Delta^k]$, $u(i) = \min[L(i), x^k + \Delta^k]$, $\Omega = \{x / l \leq x \leq u\}$

fin

enquanto *não critério de parada* **faça**

$l \leftarrow l$, $u \leftarrow u$

enquanto *não critério de parada* **faça**

Calcular

$t_k \leftarrow \arg\min m^k_{t>0}(x^k - t g^k)$

$x^{k+1} \leftarrow x^k - t_k g_k$

$P_\Omega[x^{k+1}]$,

$I(P_\Omega[x^{k+1}], l, u)$

se $I(P_\Omega[x^{k+1}], l, u) == \emptyset$ **então**

$x^{k+1} \leftarrow x^{k+1}$

$g^{k+1} \leftarrow \nabla m^k(x^{k+1})$

$k \leftarrow k + 1$

senão

Calcular $\forall i \in I(P_\Omega[x^k_C], l, u)$

$$\bar{t}_i = \begin{cases} \frac{x^i - U^i}{g^i} & se \quad g^i < 0 \\ \\ \frac{x^i - L^i}{g^i} & se \quad g^i > 0 \end{cases}$$

$\bar{t}_s \leftarrow \min\left\{\bar{t}_i > 0 / i \in I(P_\Omega[x^{(k)}_C], l, u)\right\}$

$x^{k+1} \leftarrow x^k - \bar{t}_s g^k$ (ponto na fronteira da caixa)

$g_p \leftarrow \nabla f(x^{k+1}) - (\nabla f(x^{k+1}))_s e_s$

$g^{k+1} \leftarrow g_p$

$k \leftarrow k + 1$

fim

fim

Faça (iii), (iv) **e** (v) **do algoritmo básico (Algoritmo 3).**

Encontrar l , u **e** Ω **no novo ponto** x^{k+1}.

Atualizar $\nabla f(x^{k+1})$, $\nabla^2 f(x^{k+1})$ **e** m^{k+1} **no novo ponto** x^{k+1}.

fim

fin

Para testar o algoritmo acima construído consideremos os seguintes exemplos

Exemplo 2.1.1. *Considere-se a função:*

$$f(x_1, x_2, x_3, x_4, x_5) = 2 - \frac{1}{120} x_1 x_2 x_3 x_4 x_5$$

e as restrições

$$0 \leq x_1 \leq 1$$

$$0 \leq x_2 \leq 2$$

$$0 \leq x_3 \leq 3$$

$$0 \leq x_4 \leq 4$$

$$0 \leq x_5 \leq 5$$

O ponto de mínimo para o problema com restrições é $x^* = (1, 2, 3, 4, 5)$.

A tabela mostra a sequência de valores obtidos com o algoritmo. O algoritmo faz quatro iterações para obter a solução desejada.

k	x_1	x_2	x_3	x_4	x_5	$f(x)$	$raio$
0	0.5	1	2	3	4	1.9	0.1
1	0.6000	1.1000	2.1000	3.1000	4.0755	1.8541	0.2
2	0.8000	1.3000	2.3000	3.3000	4.2755	1.7188	0.4
3	1.0000	1.7000	2.7000	3.7000	4.6755	1.3383	0.8
4	1.0000	2.0000	3.0000	4.0000	5.0000	1.0000	1.6

Exemplo 2.1.2. *Considere-se a função:*

$$f(x_1, x_2) = (4 - 2.1x_1^2 + \frac{1}{3}x_1^4)x_1^2 + x_1 x_2 + 4(x_2^2 - 1)x_2^2$$

$$-3 \leq x_1 \leq 3$$

$$-2 \leq x_2 \leq 2$$

O ponto de mínimo para o problema com restrições é $x^* = (-0.0898, 0.7126)$. A tabela mostra que o algoritmo faz dois iterações para obter a solução desejada.

k	x_1	x_2	$f(x)$	g_1	g_2	$raio$
0	-1	1	1.2333	-0.6040	7.0220	0.7
1	-0.3	0.7247	-0.8716	-1.4531	-0.0076	1.4
2	0.-0.0897	0.7127	-1.0316	0.0013	0.0019	2.8

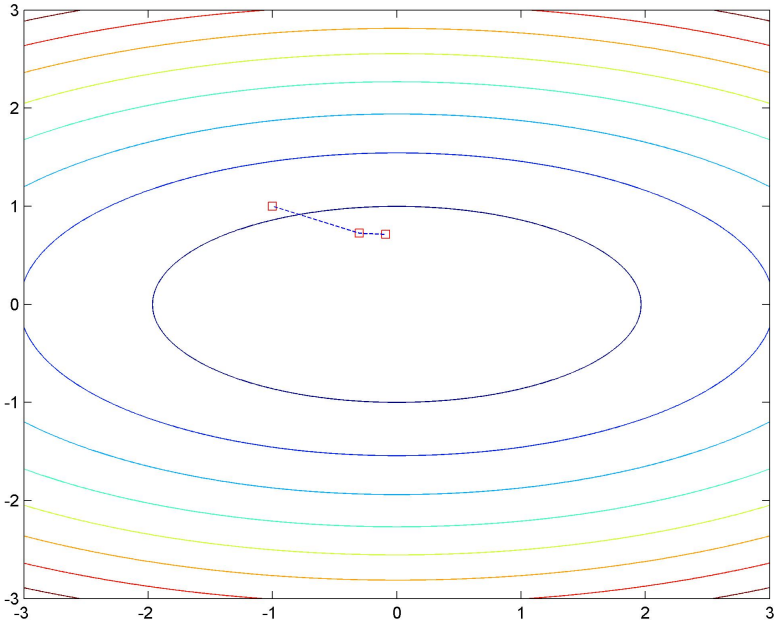

Figura 2.3: Curvas de nível da função $f(x_1, x_2) = (4 - 2.1x_1^2 + \frac{1}{3}x_1^4)x_1^2 + x_1x_2 + 4(x_2^2 - 1)x_2^2$.

2.2 Uma adaptação para o caso sem restrições

O algoritmo acima apresentado pode também ser usado para resolver problemas de otimização do tipo

$$\min_{x \in \mathbb{R}^n} f(x) \tag{2.7}$$

No algoritmo acima apresentado podemos tirar a restrições L e U, então as restrições ficariam

$$x_i^{(k)} - \Delta^{(k)} \leq x_i \leq x_i^{(k)} + \Delta^{(k)} \ , \forall i = 1, \ldots n \tag{2.8}$$

2.3 Exemplos Numéricos Obtidos com o Algoritmo

Para investigar o comportamento do algoritmo apresentado para o caso de otimização sem restrições tentamos resolver alguns problemas.

Exemplo 2.3.1. *Considere-se a seguinte função quadrática de três variáveis reais:*

$$f(x_1, x_2, x_3) = x_1^2 + x_2^2 + 2x_3^2 + x_1x_2 + x_1x_3 - 7x_1 - 5x_2 - 3x_3 + 13$$

O ponto de mínimo da função é $x^* = (3, 1, 0)$.

A tabela seguinte mostra a sequência de valores das coordenadas do vetor de otimização x_1, x_2 e x_3 da função objetivo $f(x)$ e das coordenadas do gradiente g_1,g_2 e g_3. O índice de iteração é dado por k.

k	x_1	x_2	x_3	$f(x)$	g_1	g_2	g_3	$raio$
0	-1	2	0	13	-6.9988	-1.9988	-3.9976	0.4
1	-0.6000	2.4000	0.4000	8.7600	-5.3999	-0.7999	-1.9998	0.8
2	0.2	2.4168	0.7000	4.9003	-3.4831	0.0336	0.0003	1.6
3	1.8000	1.6243	0.3000	0.9006	-1.4757	0.0487	0.0003	3.2
4	2.9717	1.0255	0.0116	0.0006	-0.0193	0.0229	0.0182	6.4

Exemplo 2.3.2. *Seja a função de duas variáveis:*

$$f(x_1, x_2) = x_2^2\big(1 - cos(x_1)\big) + 1 - cos(x_2) + e^{x_1^2}$$

O ponto de mínimo da função é $x^* = (0, 0)$.

A tabela seguinte mostra a sequência de valores obtidos com o algoritmo, observamos que o algoritmo faz oito iterações para aproximar ao ponto de mínimo.

k	x_1	x_2	$f(x)$	g_1	g_2	$raio$
0	2	1	56.4740	219.8432	3.6756	0.5
1	2	1	56.4740	219.8432	3.6756	0.25
2	1.7500	0.7500	22.3120	75.4020	2.4492	0.5
3	1.7500	0.7500	22.3120	75.4020	2.4492	0.25
4	1.5000	0.5000	9.8425	28.7178	1.4088	0.5
5	1.000	0.0000	2.7183	5.4374	0.0001	1
6	1.000	0.0000	2.7183	5.4374	0.0001	0.5
7	0.5000	-0.0001	1.2840	1.2842	-0.0000	1
8	0.0046	-0.0003	1.0000	0.0092	-0.0002	2

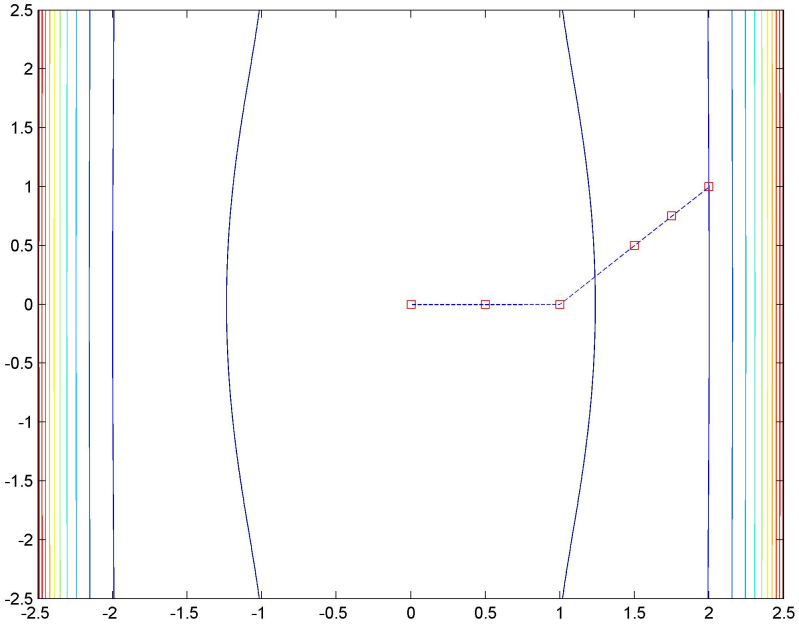

Figura 2.4: Curvas de nível da função $f(x_1, x_2) = x_2^2\big(1 - cos(x_1)\big) + 1 - cos(x_2) + e^{x_1^2}$.

2.3.1 Aplicação do algoritmo para problemas não diferenciáveis

A não diferencibilidade gera uma série de dificuldades que acarretam na inviabilidade de aplicação de vários algoritmos para problemas diferenciáveis em problemas não diferenciáveis. A primeira dificuldade aparece na etapa que determina a direção de busca, pois a direção obtida não é necessariamente de descida, e consequentemente, a busca linear não pode ser aplicada. Outra dificuldade é estabelecer um critério de parada que na maioria de vezes não fica bem definido. Por exemplo o critério de parada que utiliza $\nabla f(x^k) \leq \epsilon$, para algum $\epsilon > 0$ não pode ser aplicado em problemas não diferenciáveis.

Exemplo 2.3.3. *Seja a função* $f : \mathbb{R} \to \mathbb{R}$ *definida por* $f(x) = |x|$. *Tem-se* $\|f'(x^k)\| = 1$, $\forall x^k \neq 0$ *é dizer, não é possível gerar uma sequência de gradientes que tende para zero.*

Nosso algoritmo poderia também ser aplicado em algums problemas onde a função não é diferenciável. Neste caso, o algoritmo pára no instante em que a interseção do cone $N(x)$ com a caixa é vazia, fato que é trivial no caso em que a região de confiança é uma caixa (ver figura(2.5)).

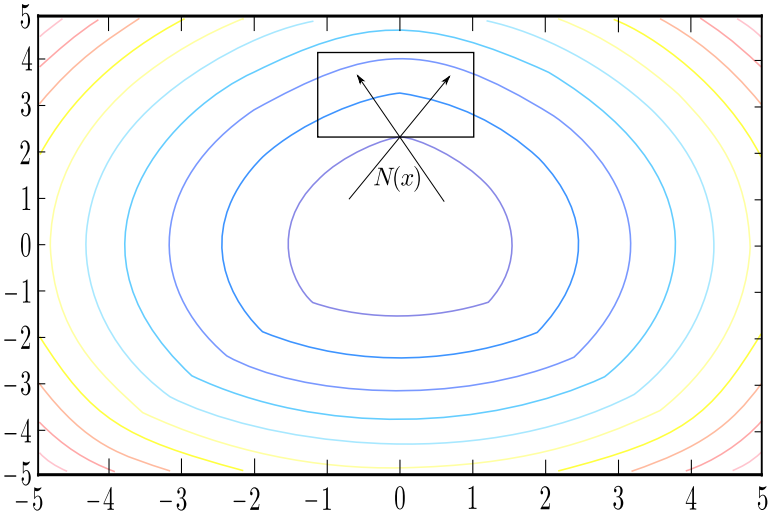

Figura 2.5: Comportamento do algoritmo no caso de uma função não diferenciável.

Exemplo 2.3.4. *Seja a função de duas variáveis:*

$$f(x_1, x_2) = 3|x_1 - 1| + |x_2 - 2|$$

Um mapa de curvas de nível da função (ver figura 2.6) mostra que a função é não diferenciável, e a região de não diferenciabilidade corresponde as retas $x_1 = 1$ e $x_2 = 2$. O mínimo desta função ocorre para $x^* = (1, 2)$. Nosso algoritmo aplicado para esta função com ponto inicial $x_0 = (3, 1)$ pára no ponto $x = (0.9998, 1.9064)$.

k	x_1	x_2	$f(x)$	g_1	g_2	$raio$
0	3	1	7.0000	-3.000	-1.000	0.7
1	2.3000	1.7000	4.2000	-3.000	-1.000	1.4000
2	0.9000	0.3000	2.000	-3.000	-1.000	2.8000
3	0.9000	0.3000	2.000	-3.000	-1.000	1.4000
...	...	...	...	...	...	...
183	0.9998	1.7533	0.2473	-3.000	-1.000	0.0109
184	1.0107	1.7643	0.2680	-3.000	-1.000	0.0109
185	0.9998	1.7752	0.2254	-3.000	-1.000	0.0219
186	0.9998	1.7752	0.2254	-3.000	-1.000	0.0109
187	1.0107	1.7861	0.2461	-3.000	-1.000	0.0109
188	0.9998	1.7971	0.2035	-3.000	-1.000	0.0219
189	0.9998	1.7971	0.2035	-3.000	-1.000	0.0109
190	1.0107	1.8080	0.2242	-3.000	-1.000	0.0109
191	0.9998	1.8189	0.1816	-3.000	-1.000	0.0219
192	0.9998	1.8189	0.1816	-3.000	-1.000	0.0109
193	1.0107	1.8299	0.2023	-3.000	-1.000	0.0109
194	0.9998	1.8408	0.1598	-3.000	-1.000	0.0219
195	1.0107	1.8518	0.1805	-3.000	-1.000	0.0109
196	0.9998	1.8627	0.1379	-3.000	-1.000	0.0219
197	0.9998	1.8627	0.1379	-3.000	-1.000	0.0109
198	1.0107	1.8736	0.1586	-3.000	-1.000	0.0109
199	0.9998	1.8846	0.1160	-3.000	-1.000	0.0219
200	1.0107	1.8955	0.1367	-3.000	-1.000	0.0109
201	0.9998	1.9064	0.0941	-3.000	-1.000	0.0219

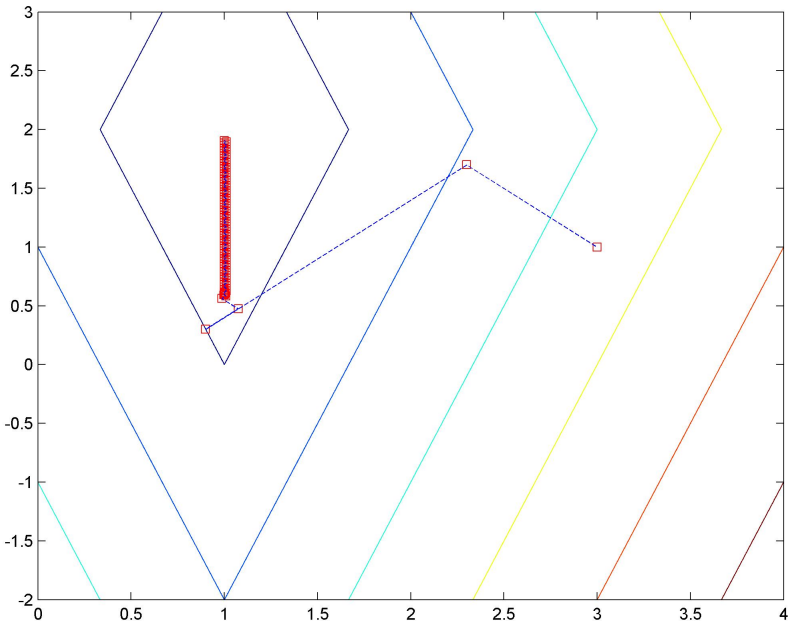

Figura 2.6: curvas de nível da função $f(x_1, x_2) = 3|x_1 - 2| + |x_2 - 2|$.

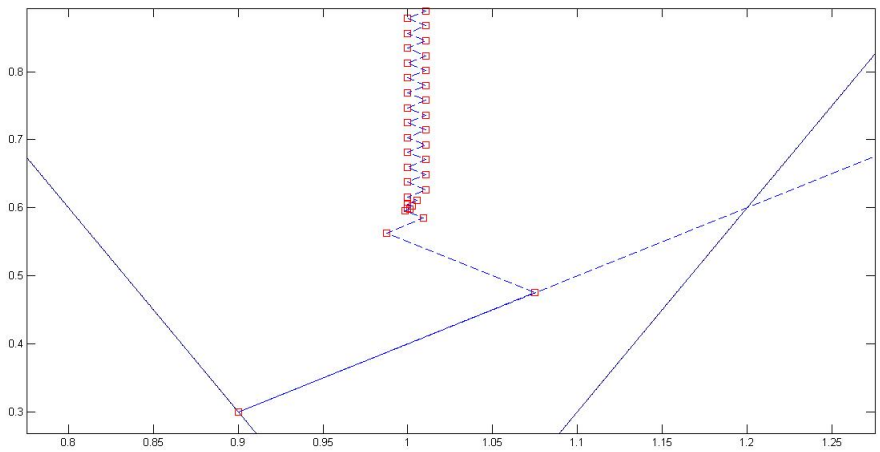

Figura 2.7: Neste exemplo a aproximação para a solução é lenta isso é causado pela não diferenciabilidade da função.

3 Conclusões

Neste trabalho, propomos um novo método para otimização com restrições simples, o algoritmo é uma proposta diferente dos já conhecidos métodos Trust-Region. O algoritmo procura minimizadores da função sobre a caixa definida pela norma $\|.\|_\infty$, nosso método utiliza uma estratégia de busca unidimensional para percorrer a direção de descida dada pelo gradiente projetado e encontrar assim o ponto que minimiza a função na região de confiança. Os resultados obtidos com algumas funções não determinam a superioridade do método, com respeito a outros algoritmos mas mostram que o algoritmo consegue resolver problemas.

REFERÊNCIAS

[1] CONN, A.; GOULD, N.; TOINT, P. *Trus-Region methods*. MPS-SIAM Series on Optimization(2000).

[2] BERTSEKAS, P. *Nonlinear programming*. Massachusetts Institute of technology (1995).

[3] ERWAY, J.; PHILIP, E. An interior-point subspace minimization method for the trust region step. *Physica*, vol 13, 1984, p. 181-194.

[4] RADOSLAW, P. *Nonconvex optimization and its aplication*. Springer-Velarg Berlin Heidelberg (2009).

[5] SORENSEN, D. *Newton's method with a model trust region modification. Vol 19*. SIAM J. Numerycal. Analysis , 1982).