

Luiz Carlos Matioli

**UMA NOVA METODOLOGIA PARA
CONSTRUÇÃO DE FUNÇÕES DE
PENALIZAÇÃO PARA ALGORITMOS
DE LAGRANGEANO AUMENTADO**

**Florianópolis
2001**

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**UMA NOVA METODOLOGIA PARA CONSTRUÇÃO DE FUNÇÕES
DE PENALIZAÇÃO PARA ALGORITMOS DE LAGRANGEANO
AUMENTADO**

Tese submetida como parte dos requisitos para obtenção do grau de Doutor em Engenharia de Produção, no Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina.

Luiz Carlos Matioli

Orientador

Clóvis Caesar Gonzaga

FLORIANÓPOLIS
DEZEMBRO-2001

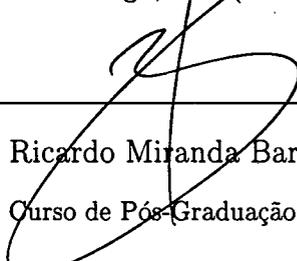
**Uma Nova Metodologia Para Construção de Funções de Penalização
Para Algoritmos de Lagrangeano Aumentado**

Luiz Carlos Matioli

Esta Tese foi julgada para a obtenção de Título de Doutor em Engenharia de Produção, Área de Concentração Engenharia de Produção, e aprovada em sua forma final pelo curso de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina.

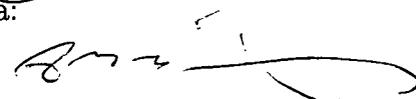


Clóvis Caesar Gonzaga, Dr. (Orientador - MTM-UFSC)

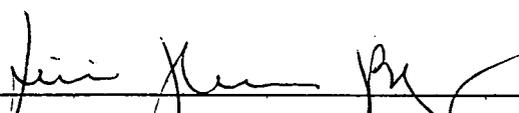


Ricardo Miranda Barcia, Ph.D.
Coordenador do Curso de Pós-Graduação em Engenharia de Produção

Banca Examinadora:



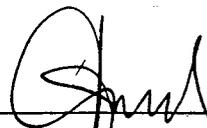
José Mario Martinez Pérez, Dr. (UNICAMP)



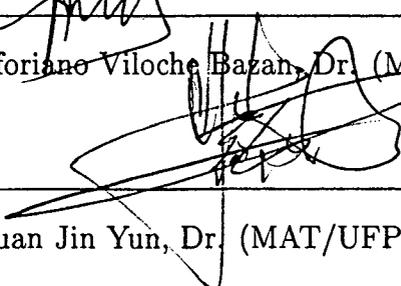
Lício Hernanes Bezerra, Dr. (MTM/UFSC)



Mario César Zambaldi, Dr. (MTM/UFSC)



Fermín Sinfiorano Viloche Bazan, Dr. (MTM/UFSC)



Yuan Jin Yun, Dr. (MAT/UFPR)

A minha esposa, Doni
Aos meus filhos, Lorena e Breno
Aos meus pais, Aparecido e Luzia

AGRADECIMENTOS

A todas as pessoas que contribuíram no desenvolvimento da tese. Em especial ao professor Clóvis Caesar Gonzaga, orientador da tese e grande amigo. Como amigos, tomamos alguns drinques, saboreamos alguns pratos e “jogamos conversa fora”. Como orientador, sempre respeitou minhas limitações. Com muita eficácia e sutileza conduzia-me a entender melhor tópicos complicados e, com toda a sua experiência (de vida, de matemático) sabia o momento certo para intervir. A convivência diária com ele ensinou-me que devemos ser humildes em nossa profissão, haja vista a aprendizagem ser sempre contínua, nunca acabada e completa. Certamente, levo na bagagem, muita experiência para aprender e trabalhar novos conteúdos.

À Universidade Federal do Paraná que possibilitou afastar-me das atividades de professor para cursar o doutorado. Aos professores do departamento de matemática desta Universidade por assumirem minhas atividades durante o tempo em que estive afastado.

À Universidade Federal de Santa Catarina, especialmente ao curso de Engenharia de Produção, onde realizei o doutorado e ao departamento de matemática, por me alojar, ceder computadores e espaço para trabalhar.

À Capes por financiar este projeto e pagar bolsa de estudo pelos quatro anos.

Aos amigos Elizabeth, Marcia, Rafael e Rômulo pela troca de experiências, pelos seminários e apoio moral nos momentos críticos. Discutimos artigos, programas de computadores, sempre contando com a ajuda do Rafael quando não sabíamos alguns comandos do latex e do matlab e sempre que os computadores, pela própria natureza deles, não funcionavam.

Aos meus amores: Doni, Lorena e Breno que abriram mão de algumas de suas aspirações, aceitaram minha ausência durante grande parte dos quatro anos e compartilharam momentos de alegria e apoiaram nos momentos de crise.

Sumário

Lista de Figuras	iv
Resumo	v
Abstract	vi
Introdução	1
1 Conceitos de análise convexa	5
1.1 Convexidade	5
1.2 Função Conjugada	8
1.3 Quase-distâncias	11
2 Pontos Proximais e Lagrangeanos Aumentados	12
2.1 Ponto Proximal	13
2.2 Algoritmo de Ponto Proximal	14
2.3 Ponto Proximal e Região de Confiança	20
2.4 Funções de Penalidades Coercivas Pela Direita	23
2.5 Propriedades da Conjugada de Uma Penalidade da Família \mathcal{P}	23
2.6 Construindo Penalidades da Família \mathcal{P}	26
2.7 Métodos de Lagrangeano Aumentado	28
2.8 A Função Lagrangeano Aumentado	29
2.9 O Algoritmo de Lagrangeano Aumentado	29
2.10 Relação Primal Dual	30
3 A Penalidade M^2B e Resultados de Convergência	35
3.1 Definição da Penalidade M^2B	36
3.2 A Penalidade M^2B Com Quadrática Ajustada	38
3.3 Potencialidades das Penalidades Tipo 1	42

4	Programação Linear e Equivalência Entre Lagrangeano Aumentado e Afim Escala no Caso Linear	44
4.1	O Problema Linear e Seu Dual	45
4.2	A Função Quadrática	46
4.3	Ponto Proximal Para o Caso Linear Com Quase-Distância Quadrática .	47
4.4	Lagrangeano Aumentado Para o Problema Linear e Penalidade Quadrática	50
4.5	Algoritmo Afim Escala	51
4.6	Relação Entre Afim Escala e Lagrangeano Aumentado, no Caso Linear	51
5	Ajuste do Parâmetro de Penalidade	57
6	Testes Numéricos	64
	Considerações Finais	78
	Referências Bibliográficas	79

Lista de Figuras

1.1	Função barreira logarítmica	9
1.2	Função conjugada	9
1.3	Função exponencial	9
1.4	Função conjugada	9
2.1	Função g e função regularizada g_k	15
2.2	Distância de Bregman	17
3.1	Gráfico da penalidade M^2B para três valores diferentes de μ	37
3.2	Quadrática ajustada à barreira modificada	39
3.3	Curvas de nível das aproximações quadráticas de quase-distâncias	43
4.1	Passo do método de ponto proximal	48
4.2	Passo do método de ponto proximal	49
4.3	μ dual não viável	49
4.4	Direções duais geradas pelo alg. de lagrangeano aumentado	54
4.5	Elipsóide N para algum valor de c positivo	55

RESUMO

Propomos uma nova metodologia para construção de funções de penalização para algoritmos de lagrangeano aumentado aplicados a problemas de programação convexa e com restrições.

Métodos de lagrangeano aumentado partem normalmente de funções de penalização $\theta : \mathbb{R} \rightarrow \mathbb{R}$ estritamente convexas e crescentes, que são combinadas com multiplicadores de lagrange μ para compor termos de penalização com o formato

$$(y, \mu) \in \mathbb{R} \times \mathbb{R}_{++} \mapsto p(y, \mu) = \mu\theta(y).$$

Neste trabalho construímos as penalidades por

$$(y, \mu) \in \mathbb{R} \times \mathbb{R}_{++} \mapsto p(y, \mu) = \theta(\mu y).$$

No caso não linear, mostramos que todo ponto de acumulação da seqüência de multiplicadores gerada pelo algoritmo de lagrangeano aumentado com penalidade p acima e θ , a função barreira logarítmica modificada de Polyak, é uma solução ótima dual do problema. Mostramos ainda que, ajustando uma quadrática à função de Polyak, o resultado anterior continua verdadeiro.

No caso linear, mostramos que o algoritmo de lagrangeano aumentado com a penalidade p acima e θ quadrática gera direções duais equivalentes às geradas pelo algoritmo afim escala aplicado ao dual do problema linear. Mostramos também que este resultado não se verifica com a função de penalidade usual

ABSTRACT

We propose a new methodology for constructing penalization functions for augmented lagrangian algorithms, applied to convex programming problems with constraints.

Augmented lagrangian methods normally are built from penalization functions $\theta : \mathcal{R} \rightarrow \mathcal{R}$ strictly convex and increasing, which are combined with Lagrange multipliers μ to compose terms of penalization with the shape

$$(y, \mu) \in \mathcal{R} \times \mathcal{R}_{++} \mapsto p(y, \mu) = \mu\theta(y).$$

In this work we construct the penalty by

$$(y, \mu) \in \mathcal{R} \times \mathcal{R}_{++} \mapsto p(y, \mu) = \theta(\mu y).$$

In the nonlinear case, we show that all the limit points of the sequence of multipliers generated by the augmented lagrangian algorithm with the penalty function p above and with θ given by Polyak's modified barrier function is an optimal solution of the problem. We also show that adjusting a quadratic function to Polyak's barrier, the previous remains true.

In the linear case, we show that the augmented lagrangian algorithm with the penalty function p above and quadratic θ generates dual directions equivalent to those generated by the affine scaling algorithm applied to the dual problem. We also show that this result is not true when the usual penalty function is used.

Introdução

Motivação

O foco central de nosso trabalho são os métodos de lagrangeano aumentado os quais são baseados em penalizações. São usados para resolver problemas de programação não linear com restrições. Em sua forma original, foram aplicados em problemas com restrições de igualdade e introduzidos, em meados da década de 60, por Hestenes [13] e Powell [26], em trabalhos independentes. Neste trabalho serão aplicados a problemas com restrições de desigualdade, ou seja, ao seguinte problema

$$(P) \quad \begin{array}{ll} \text{minimizar} & f_0(x) \\ \text{sujeito a} & f_i(x) \leq 0, \quad i = 1, 2, \dots, m \\ & x \in \mathbb{R}^n \end{array}$$

onde $f_i : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$, para $i = 0, 1, \dots, m$ são funções convexas, próprias e fechadas. A função f_0 é chamada de função objetivo e as funções f_i para $i = 1, 2, \dots, m$ são chamadas de restrições. O conjunto formado pelas restrições é chamado de conjunto viável e passaremos a denotá-lo por X . Assim, se $f = (f_1, \dots, f_m)^T$ tem-se

$$X = \{x : f(x) \leq 0\}.$$

A metodologia dos métodos de lagrangeano aumentado ou, mais geral, dos métodos de penalização é bastante simples. Consiste em transformar o problema restrito em uma seqüência de subproblemas irrestritos. São iterativos e cada iteração resolve um subproblema irrestrito e atualiza os parâmetros (multiplicadores e parâmetro de penalização). Os subproblemas irrestritos resolvidos geram uma seqüência, cujo nome é primal. Uma outra seqüência gerada pelo método é a de multiplicadores, a qual é uma seqüência dual. Assim, são geradas duas seqüências, (x^k) a primal e (μ^k) a dual. A seqüência primal, em geral, não é viável (não satisfaz $f(x^k) \leq 0$), mas espera-se

que vá se tornando viável à medida que k cresce. Já a seqüência dual é sempre viável, no sentido da viabilidade dual, $\mu^k \geq 0$, para todo $k \geq 0$.

O grande charme dos métodos de lagrangeano aumentado é a elegante relação de dualidade que exibem com os métodos de ponto proximal, como veremos no decorrer do trabalho. Um dos grandes resultados alcançados durante a época em que estes métodos foram sendo resolvidos foi exatamente esta conexão entre primal e dual ou, em termos de métodos, lagrangeano aumentado e ponto proximal. Assim, este resultado garante que, sob hipóteses adequadas, a seqüência de multiplicadores gerada pelo algoritmo de lagrangeano aumentado é a mesma obtida por um certo algoritmo de ponto proximal aplicado ao problema dual.

Toda teoria de convergência da seqüência de multiplicadores gerada pelos algoritmos de lagrangeano aumentado é desenvolvida no dual, via métodos proximais [28, 29]. A parte de computação é feita toda no primal.

Os métodos de ponto proximal fazem uma espécie de regularização, ou seja, soma-se à função objetivo um termo positivo, geralmente chamado de núcleo. Quando são aplicados ao problema dual do problema de programação convexa, funcionam como uma espécie de barreira, pois forçam os pontos gerados a ficarem no ortante positivo.

A primeira relação alcançada entre os métodos de ponto proximal e lagrangeano aumentado foi obtida usando núcleos quadráticos nos métodos de ponto proximal [28, 29]. A função de penalização usada no primal, para obter tal relação, tem a peculiaridade de não ser de classe C^2 , que é uma restrição considerável, pois vários métodos de minimização irrestrita aplicam-se somente a funções de classe C^2 .

O grande desafio, passa a ser a substituição dos núcleos quadráticos por núcleos de classe pelo menos C^2 . Neste sentido, muitas contribuições foram feitas por vários pesquisadores [8, 17, 22, 31]. Atualmente, duas classes de métodos não quadráticos são muito utilizadas: distâncias de Bregman [3] e φ -divergências [31].

Façamos, por questão de informação, um pequeno retrospecto dos avanços conseguidos. Censor e Zenios [5] propuseram um algoritmo de ponto proximal usando distância de Bregman. Eckstein [8], inspirado no trabalho de Censor e Zenios, generaliza o trabalho de Rockafellar [29], usando distâncias de Bregman no lugar de núcleos quadráticos. Teboulle [31], baseado na regularização de Moreau-Yoshida, generaliza o método de ponto proximal usando φ -divergências, além de distância de Bregman.

Muitas outras informações sobre estas classes de métodos podem ser obtidas em [1, 4, 6, 9, 19, 20, 21, 24, 30]. Uma abordagem bastante geral e histórica, com muitas referências, é apresentada por Iusem em [18] (ver também o livro de Bertsekas [2]).

Informações com respeito à convergência da seqüência primal podem ser encontradas em [4, 17, 25, 30, 32]

A seguir descrevemos o que será tratado em cada capítulo da tese.

Descrição por Capítulos

O trabalho está dividido em 6 capítulos. No capítulo 1 apresentamos conceitos de análise convexa que formarão a parte teórica básica para o restante dos capítulos. Tais conceitos são baseados nos livros [14, 15] e [27].

No capítulo 2 apresentamos os métodos de ponto proximal e de lagrangeano aumentado sendo, o principal resultado existente, o de equivalência das seqüências duais geradas por estes dois métodos. Outra equivalência importante tratada neste capítulo é a entre métodos de ponto proximal e de região de confiança.

O resultado mais relevante que estamos propondo neste trabalho está apresentado no capítulo 3. Propomos uma nova metodologia de função de penalidade para os métodos de lagrangeano aumentado. Através dessa nova metodologia e do resultado de equivalência entre ponto proximal e lagrangeano aumentado, mostramos que todo ponto de acumulação da seqüência dual, gerada pelo algoritmo de lagrangeano aumentado, é uma solução ótima dual do problema (P), definido no início desta introdução.

Uma segunda contribuição é apresentada no capítulo 4. Mostramos que o algoritmo de lagrangeano aumentado, aplicado a um problema de programação linear, com penalização quadrática, gera direções duais equivalentes às geradas pelo algoritmo afim escala, aplicado ao dual do problema linear. Com o que foi desenvolvido nesse capítulo, juntamente com a seção final do capítulo 3, tem-se que essa equivalência entre lagrangeano aumentado e afim escala pode ser estendida também para o caso não linear.

No capítulo 5 apresentamos questões referentes à implementação dos algoritmos de lagrangeano aumentado. Exploramos de forma sistemática a questão da atualização do parâmetro de penalização nos métodos de lagrangeano aumentado, coisa que, em geral, não é feita na maioria dos trabalhos encontrados na literatura. Acreditamos que esse parâmetro tem uma função essencial. Quando o valor deste fica muito pequeno os subproblemas gerados pelos algoritmos de lagrangeano aumentado ficam muito difíceis de resolver, como será percebido neste capítulo. Dentro dessa visão, propomos um algoritmo que utiliza uma aproximação quadrática da função lagrangeano aumentado e da função penalidade para determinar o parâmetro de penalidade para o

algoritmo de lagrangeano aumentado.

No capítulo 6 apresentamos testes numéricos do algoritmo de lagrangeano aumentado com a metodologia proposta. Aplicamos também o mesmo algoritmo com uma penalidade muito utilizada nos trabalhos encontrados na literatura e fazemos uma análise comparativa entre os resultados encontrados pelas duas metodologias.

Notações

Listamos a seguir algumas notações que serão utilizadas com mais frequência.

Estas podem ser encontradas em qualquer livro sobre análise convexa.

$\text{dom } f$ - domínio efetivo da função f

$\text{intdom } f$ - interior do domínio efetivo da função f

$\text{ri } X$ - interior relativo do conjunto X

$f'(x, d)$ - derivada direcional de f no ponto x e na direção d

$\partial f(\cdot)$ - subdiferencial da função f

∂X - fronteira do conjunto X

$f'_\infty(\cdot)$ - função de recessão da função $f(\cdot)$

$\text{cl}(X)$ - fecho do conjunto X

$\mathbb{R}_{++}^m = \{x \in \mathbb{R}^m : x_i > 0, i = 1, \dots, m\}$

$\mathbb{R}_+^m = \{x \in \mathbb{R}^m : x_i \geq 0, i = 1, \dots, m\}$

Usamos ainda uma notação simplificada para seqüências, escrevendo (\mathbf{x}^k) ao invés de $(\mathbf{x}^k)_{k \in \mathbb{N}}$.

Capítulo 1

Conceitos de análise convexa

Neste capítulo, apresentaremos definições e propriedades básicas sobre análise convexa e quase-distâncias. A maioria destas propriedades estão demonstradas nos livros [14, 15] e [27], nos quais estamos nos baseando. Para o caso linear basear-nos em [11, 12].

1.1 Convexidade

Definição 1.1 *O conjunto $C \subset \mathbb{R}^n$ é chamado convexo se para todo $x_1, x_2 \in C$ e $\alpha \in [0, 1]$, $\alpha x_1 + (1 - \alpha)x_2 \in C$.*

Geometricamente, isto significa que o segmento de reta $[x_1, x_2] = \{\alpha x_1 + (1 - \alpha)x_2, \alpha \in [0, 1]\}$ está inteiramente contido em C .

Exemplo 1.2 *O conjunto viável do problema (P), $X = \{x \in \mathbb{R}^n : f(x) \leq 0\}$, definido na introdução, é convexo.*

Definição 1.3 *Considere C um subconjunto não vazio e convexo do \mathbb{R}^n . A função $f : C \rightarrow \mathbb{R}$ é dita convexa sobre C , se para todo $x_1, x_2 \in C$ e todo $\alpha \in [0, 1]$ se verifica*

$$f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2). \quad (1.1)$$

Considere $C \subset \mathbb{R}^n$ um conjunto convexo. Muitas vezes é conveniente definir f fora do conjunto C (pontos que pertencem ao \mathbb{R}^n e não pertencem a C), ou seja, estender f para todo \mathbb{R}^n e neste caso f pode assumir valor $+\infty$. Um exemplo bastante

utilizado em análise convexa é a função indicadora, definida sobre um conjunto convexo $C \subset \mathbb{R}^n$, da seguinte forma

$$I_C(x) = \begin{cases} 0 & , \text{ se } x \in C \\ +\infty & , \text{ c.c.} \end{cases} \quad (1.2)$$

Estendemos a definição 1.3 para todo $x \in \mathbb{R}^n$.

Definição 1.4 Dizemos que $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$, não identicamente $+\infty$, é convexa se, para todo $x_1, x_2 \in \mathbb{R}^n$ e todo $\alpha \in [0, 1]$, se verifica

$$f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2) \quad (1.3)$$

considerada como desigualdade em $\mathbb{R} \cup \{+\infty\}$.

Se a desigualdade na definição 1.4 é estrita e $\alpha \in (0, 1)$, dizemos que f é estritamente convexa.

Definição 1.5 O domínio efetivo de uma função convexa é o conjunto não vazio

$$\text{dom } f = \{x \in \mathbb{R}^n : f(x) < \infty\}. \quad (1.4)$$

Exemplo 1.6 Considere $C \subset \mathbb{R}^n$ um conjunto convexo e I a função indicadora dada na relação (1.2). Então $\text{dom } I_C(\cdot) = C$.

Definição 1.7 Dizemos que uma função convexa $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ é própria se é finita em pelo menos um ponto.

Definição 1.8 Considere $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ convexa e própria. O vetor $s \in \mathbb{R}^n$ é um subgradiente de f no ponto $x \in \mathbb{R}^n$ se satisfaz:

$$\forall y \in \mathbb{R}^n, \quad f(y) \geq f(x) + s^T(y - x). \quad (1.5)$$

O conjunto de todos os subgradientes de f no ponto x é chamado de subdiferencial de f em x e é denotado por

$$\partial f(x) = \{s \in \mathbb{R}^n : f(y) \geq f(x) + s^T(y - x), \forall y \in \mathbb{R}^n\}. \quad (1.6)$$

Definição 1.9 A função $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ é dita fechada, ou semicontínua inferior, se para todo $x \in \mathbb{R}^n$ satisfaz

$$\liminf_{y \rightarrow x} f(y) \geq f(x). \quad (1.7)$$

Definição 1.10 Dado $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ própria. O epigrafo de f é o conjunto não vazio

$$\text{epi} f = \{(x, r) \in \mathbb{R}^n \times \mathbb{R} : r \geq f(x)\}. \quad (1.8)$$

e, para cada $r \in \mathbb{R}$, os conjuntos de nível de f são definidos como

$$S_r(f) = \{x \in \mathbb{R}^n : f(x) \leq r\}. \quad (1.9)$$

Proposição 1.11 Para $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$, as três propriedades seguintes são equivalentes:

- (i) f é semicontínua inferior em \mathbb{R}^n ;
- (ii) $\text{epi} f$ é um conjunto fechado em $\mathbb{R}^n \times \mathbb{R}$;
- (iii) os conjuntos de nível $S_r(f)$ são fechados (possivelmente vazios) para todo $r \in \mathbb{R}$.

Definição 1.12 Considere uma função $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ convexa, própria, fechada e x um ponto arbitrário no $\text{dom} f$. A função de recessão de f é definida por:

$$d \in \mathbb{R}^n \mapsto f'_\infty(d) = \lim_{t \rightarrow +\infty} \frac{f(x + td) - f(x)}{t}. \quad (1.10)$$

Na definição anterior, a função de recessão é independente da variável x , a dependência é em relação a direção d (ver proposição 3.2.2 pag.179 em [14]).

Definição 1.13 Considere $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ e $x \in \text{dom} f$. A derivada direcional de f em x na direção $d \in \mathbb{R}^n$ é definida por

$$f'(x, d) = \lim_{t \downarrow 0} \frac{f(x + td) - f(x)}{t}. \quad (1.11)$$

Na relação (1.11) $t \downarrow 0$ significa que t tende a zero e $t > 0$.

Teorema 1.14 Considere f uma função convexa e própria. Para $x \notin \text{dom} f$, $\partial f(x)$ é vazio. Para $x \in \text{ri}(\text{dom} f)$, $\partial f(x)$ é não vazio, $f'(x, d)$ é fechada e própria como uma

função de d , e

$$f'(x, d) = \sup\{\gamma^t d : \gamma \in \partial f(x)\}.$$

Finalmente, $\partial f(x)$ é um conjunto não vazio e limitado se, e somente se, $x \in \text{intdom } f$, em qual caso $f'(x, d)$ é finita para todo d .

Teorema 1.15 Para qualquer função convexa e qualquer α real, os conjuntos de nível $\{x : f(x) \leq \alpha\}$ são convexos.

1.2 Função Conjugada

A partir da função $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$, define-se uma outra função, a qual é chamada de função conjugada, e esta será muito importante quando formos discutir, mais adiante, a relação entre os métodos de lagrangeano aumentado e de ponto proximal.

Definição 1.16 Considere $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ uma função convexa. A convexa conjugada de f é a função f^* dada por:

$$s \in \mathbb{R}^n \rightarrow f^*(s) = \sup\{x^t s - f(x), x \in \mathbb{R}^n\}. \quad (1.12)$$

Prova-se que a conjugada f^{**} de f^* é a própria f e que f^* é fechada, própria e convexa se, e somente se, f é própria (ver [27], teorema 12.2, pag. 104).

A transformação $f \rightarrow f^*$ é, frequentemente, chamada de operação conjugação, ou ainda, transformada de Legendre-Fenchel.

Exemplo 1.17 Considere a função $f : \mathbb{R} \rightarrow \mathbb{R} \cup \{+\infty\}$ definida por

$$f(x) = \begin{cases} -\log(1-x) & , \text{ se } x < 1 \\ +\infty & , \text{ se } x \geq 1. \end{cases} \quad (1.13)$$

A conjugada de f tem a forma:

$$s \in \mathbb{R} \mapsto f^*(s) = \begin{cases} s - \log s - 1 & , \text{ se } s > 0 \\ +\infty & , \text{ se } s \leq 0. \end{cases} \quad (1.14)$$

A figura 1.1 mostra o gráfico da função f dada no exemplo anterior, e a figura 1.2 mostra o gráfico da sua conjugada.

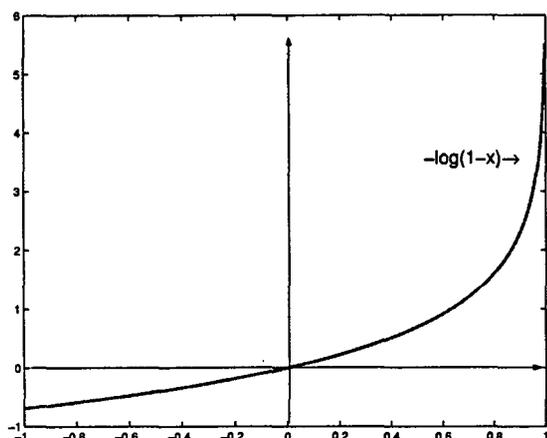


Figura 1.1: Função barreira logarítmica

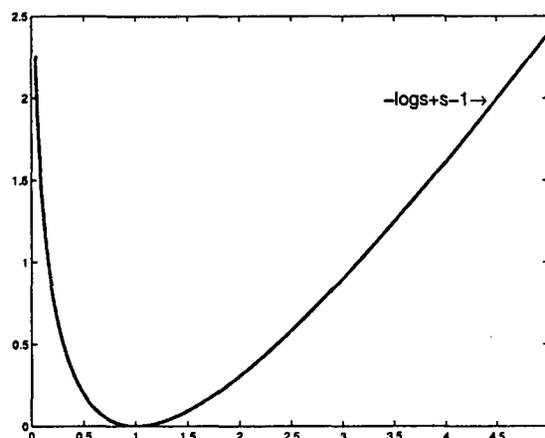


Figura 1.2: Função conjugada

Exemplo 1.18 A função $f : \mathbb{R} \rightarrow \mathbb{R} \cup \{+\infty\}$ definida por $f(x) = e^x - 1$, tem como conjugada

$$s \in \mathbb{R} \mapsto f^*(s) = \begin{cases} s \log s - s + 1 & , \text{ se } s \geq 0 \\ +\infty & , \text{ se } s < 0 . \end{cases} \quad (1.15)$$

Observe que estamos usando a convenção $0 \log 0 = 0$.

A figura 1.3 mostra o gráfico da função f dada no exemplo anterior, e a figura 1.4 mostra o gráfico da sua conjugada.

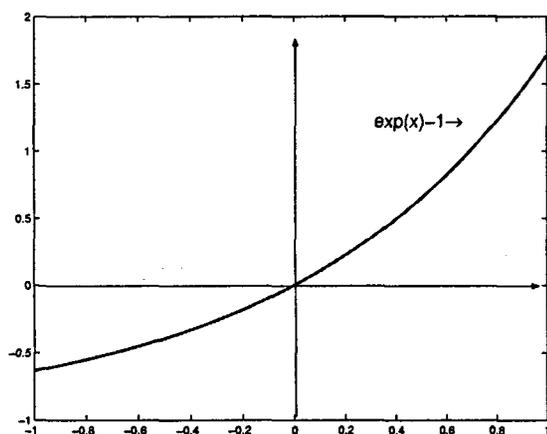


Figura 1.3: Função exponencial

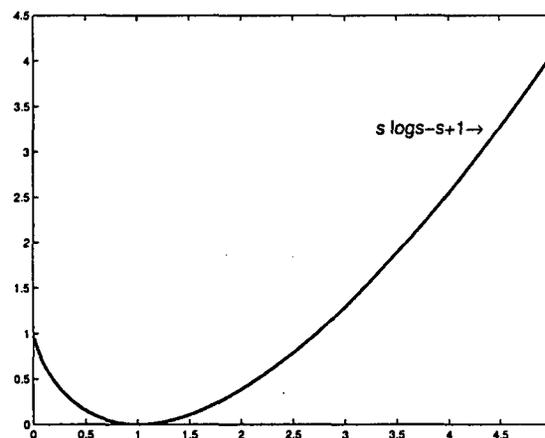


Figura 1.4: Função conjugada

A função dada no exemplo (1.17) é conhecida como barreira logarítmica modificada de Polyak. A função do exemplo (1.18) foi utilizada por Tseng e Bertsekas em [32] para mostrar a convergência do método de multiplicadores exponencial, para programação convexa.

A seguir apresentaremos algumas propriedades sobre funções conjugadas. Que serão utilizadas nos próximos capítulos.

Proposição 1.19 *Considere as funções $f, f_1, f_2 : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$, a seguir, convexas e próprias. As seguintes afirmações são verdadeiras:*

- (i) A conjugada da função $g(x) = f(x) + \alpha$ é $g^*(s) = f^*(s) - \alpha, \forall \alpha \in \mathbb{R}$;
- (ii) A conjugada da função $g(x) = \alpha f(x), \alpha > 0$, é a função $g^*(s) = \alpha f^*(\frac{s}{\alpha})$;
- (iii) A conjugada da função $g(x) = f(\alpha x), \alpha \neq 0$, é a função $g^*(s) = f^*(\frac{s}{\alpha})$;
- (iv) Se A é um operador linear inversível, então $(f \circ A)^* = f^* \circ (A^{-1})$;
- (v) A conjugada da função $g(x) = f(x - x_0)$ é $g^*(s) = f^*(s) + s^t x_0$;
- (vi) A conjugada da função $g(x) = f(x) + s_0^t x$ é $g^*(s) = f^*(s - s_0)$;
- (vii) Se $f_1 \leq f_2$ então $f_1^* \geq f_2^*$;
- (viii) Se $\text{dom} f_1 \cap \text{dom} f_2 \neq \emptyset$ e $\alpha \in (0, 1)$, então $(\alpha f_1 + (1 - \alpha) f_2)^* \leq \alpha f_1^* + (1 - \alpha) f_2^*$;
- (ix) Se $f(x) = \sum_{i=1}^n f_i(x_i)$, com $f_i : \mathbb{R} \rightarrow \mathbb{R} \cup \{+\infty\}$, então $f^*(s) = \sum_{i=1}^n f_i^*(s_i)$.

Exemplo 1.20 *Considere f a função dada no exemplo (1.17) e sejam $\alpha > 0$ e $x \in \mathbb{R}^n \mapsto g(x) = \sum_{i=1}^n f(\alpha x_i)$. Então, pelos itens (iii) e (ix) da proposição (1.19), juntamente com a conjugada da função f dada na relação (1.14), obtemos a conjugada de g*

$$s \in \mathbb{R}^n \mapsto g^*(s) = \begin{cases} \sum_{i=1}^n (\frac{s_i}{\alpha} - \log \frac{s_i}{\alpha} - 1) & , \text{ se } s > 0 \\ +\infty & , \text{ c.c.} \end{cases} \quad (1.16)$$

Note que estamos usando a notação vetorial componente a componente. Quando denotamos $s \in \mathbb{R}^n$ e $s > 0$, estamos querendo dizer que cada componente de s é positiva, ou seja, $s_i > 0$, para todo $i = 1, \dots, n$.

Outra propriedade que aparece com muita frequência em análise convexa, que envolve uma função f , sua conjugada, e o subdiferencial (ou derivada - no caso em que f é diferenciável) é a seguinte

Proposição 1.21 *Se $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função convexa e fechada, então as seguintes equivalências se verificam*

$$f(x) + f^*(s) - \langle s, x \rangle = 0 \text{ (ou } \leq 0) \iff s \in \partial f(x) \iff x \in \partial f^*(s). \quad (1.17)$$

1.3 Quase-distâncias

As quase-distâncias definidas a seguir fornecem uma espécie de distância entre pontos de um conjunto e serão muito úteis quando trabalharmos com o método de ponto proximal.

Definição 1.22 *Considere os conjuntos $S \subset \mathbb{R}^n$ convexo e não vazio e \hat{S} tais que $S \subset \hat{S} \subset \mathbb{R}^n$. Uma função contínua $D : \hat{S} \times S \rightarrow \mathbb{R}$ é uma quase-distância se, e somente se, as seguintes condições são satisfeitas*

$$(a) \quad D(x, y) \geq 0, \quad \forall (x, y) \in (\hat{S} \times S);$$

$$(b) \quad D(x, y) = 0 \text{ se, e somente se, } x = y, \quad \forall (x, y) \in (\hat{S} \times S).$$

Note que a função D não define necessariamente uma métrica, uma vez que não satisfaz a propriedade de simetria nem tampouco a desigualdade triangular. Abaixo apresentaremos um exemplo de quase-distância que define uma métrica.

Exemplo 1.23 *Considere $\hat{S} = \mathbb{R}^n$ na definição (1.22). A função $D : \mathbb{R}^{2n} \rightarrow \mathbb{R}$ dada por $D(x, y) = \|x - y\|^2$ define uma quase-distância, onde $\|\cdot\|$ é a norma euclidiana.*

A quase-distância do exemplo (1.23) é o termo que faz a regularização da função objetivo no método de ponto proximal clássico. Este é o núcleo quadrático que mencionamos na introdução.

Proposição 1.24 *Considere $D : \hat{S} \times S \rightarrow \mathbb{R}$ uma quase-distância. Se, para todo $y \in S$, $D(\cdot, y)$ é convexa, então os conjuntos de nível $\Gamma(y, \alpha) = \{x \in \hat{S} : D(x, y) \leq \alpha\}$ são limitados, para todo $y \in S$ e todo $\alpha \in \mathbb{R}$.*

Prova. Seja $\alpha = 0$. Então, do item (b) da definição 1.22, tem-se que o conjunto $\Gamma(x, 0) = \{x \in \hat{S} : D(x, y) \leq 0\} = \{y\}$, o qual é limitado, para todo $y \in S$.

Por hipótese $D(\cdot, y)$ é convexa, para todo y em S . Pelo teorema 1.15 segue que $\Gamma(\cdot, y)$ é limitado, para todo α em \mathbb{R} . ■

Capítulo 2

Pontos Proximais e Lagrangeanos Aumentados

Neste capítulo, apresentamos os algoritmos de pontos proximais, as relações destes com região de confiança e lagrangeano aumentado.

Iniciamos com o algoritmo de ponto proximal aplicado a um problema convexo restrito ao ortante não negativo do \mathbb{R}^n e utilizando uma quase-distância coerciva. Mostramos que o algoritmo está bem definido, no sentido de que dado um ponto inicial é sempre possível determinar o próximo ponto.

Em seguida, apresentamos a definição de distância de Bregman e algumas de suas propriedades mais usuais. Adaptamos um teorema mostrado em [4] ao nosso caso. Este teorema garante que todo ponto de acumulação da seqüência gerada pelo algoritmo de ponto proximal com distância de Bregman é uma solução ótima do problema considerado.

Na parte em que relacionamos os algoritmos de ponto proximal e região de confiança, mostramos que existe uma certa relação de equivalência entre as seqüências geradas por estes métodos e ainda mostramos que aumentando (ou diminuindo) um certo parâmetro, conhecido na literatura como parâmetro de regularização, no método de ponto proximal é equivalente a diminuir (ou aumentar) o tamanho da região no método de região de confiança. Estes resultados não são inéditos, no entanto, não encontramos na literatura a demonstração dos mesmos. Posteriormente, utilizamos estes resultados para obter uma equivalência entre os algoritmos de lagrangeano aumentado e afim escala.

Afim de apresentarmos o algoritmo de lagrangeano aumentado, definimos uma família de funções de penalização e metodologias de construção destas penalidades. Em

seguida, apresentamos um dos resultados que consideramos mais importantes sobre estes métodos, alcançados nos últimos anos, o qual fornece uma equivalência entre as seqüências duais geradas pelo algoritmos de lagrangeano aumentado e de ponto proximal.

No último resultado do capítulo, mostramos que as seqüências geradas pelo algoritmo de lagrangeano aumentado, com a metodologia que propomos para este, satisfazem a condição de complementaridade para o problema primal, definido na introdução.

2.1 Ponto Proximal

Nesta seção, introduzimos o algoritmo de ponto proximal, aplicado a problemas definidos em subconjuntos convexos do \mathbb{R}^n . Mostramos que o algoritmo está bem definido e no caso de distância de Bregman, mostramos que todo ponto de acumulação da seqüência gerada é uma solução ótima do problema considerado. No final da seção, mostramos que os algoritmos de ponto proximal e de região de confiança, aplicados a problemas irrestritos do \mathbb{R}^n exibem uma certa relação de equivalência.

Considere os conjuntos $S \subset \mathbb{R}^n$ aberto e convexo e \hat{S} tal que $S \subset \hat{S} \subset clS$. Definimos o problema

$$(P1) \quad \begin{array}{ll} \text{minimizar} & g(x) \\ \text{sujeito a} & x \in clS \end{array}$$

onde $g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ é uma função convexa, própria e fechada.

Nota 2.1 Apesar da inclusão $S \subset \hat{S} \subset clS$, estaremos particularmente interessados nos casos em que $\hat{S} = S$ e $\hat{S} = clS$. No caso do problema dual de programação convexa, teremos $\hat{S} = \mathbb{R}_{++}^n$ ou $\hat{S} = \mathbb{R}_+^n$.

Vamos supor que o interior relativo do domínio da função g e o conjunto S se interceptam e que a função g é limitada inferiormente, ou seja:

Hipótese H1 : $\text{ri dom } g \cap S \neq \emptyset$

Hipótese H2 : O problema (P1) tem solução ótima

Definição 2.2 Considere $x \in \mathbb{R}^n \mapsto g(x) \in \mathbb{R} \cup \{+\infty\}$ uma função convexa, própria, fechada e $S \subset \mathbb{R}^n$ aberto tal que $S \subset \text{dom } g$. Dizemos que g é coerciva na fronteira de

S se, dados $x \in \partial S$, $y \in S$ e $h = y - x$ ($y \neq x$), então

$$\lim_{\substack{z \rightarrow x \\ z \in S}} g'(z, h) = -\infty$$

onde $g'(z, h)$ é a derivada direcional de g em z e na direção h , dada na definição (1.13).

No lema a seguir, mostraremos que se g é coerciva na fronteira de S então $\partial g(x) = \emptyset$ para todo $x \in \partial S$.

Lema 2.3 *Considere $S \subset \mathbb{R}^n$ aberto e convexo e $g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ uma função convexa, própria, fechada e coerciva na fronteira de S . Se $x \in \partial S$ então $\partial g(x) = \emptyset$.*

Prova. Considere $x \in \partial S$, $y \in S$, e a direção $h = y - x$, com $y \neq x$. Suponhamos por absurdo que $\partial g(x) \neq \emptyset$, então existe $\gamma \in \partial g(x)$ e do teorema 1.14 segue que

$$g'(x, h) \geq \gamma^T h, \quad (2.1)$$

segue ainda, da convexidade da função g que

$$g'(x + \lambda h, h) \geq g'(x, h), \quad \forall \lambda \in [0, 1]. \quad (2.2)$$

Tomando o limite em (2.2) com λ tendendo a zero, tem-se

$$\lim_{\lambda \rightarrow 0} g'(x + \lambda h, h) \geq g'(x, h)$$

e da relação (2.1), segue que

$$\lim_{\lambda \rightarrow 0} g'(x + \lambda h, h) \geq \gamma^T h$$

o que contradiz a definição 2.2 de coercividade na fronteira. ■

2.2 Algoritmo de Ponto Proximal

A seguir apresentaremos o algoritmo de ponto proximal aplicado ao problema (P1) com quase-distância $D : \hat{S} \times S \rightarrow \mathbb{R}$, dada na definição 1.22, tal que para todo $z \in S$, $D(\cdot, z)$ é coerciva na fronteira de seu domínio \hat{S} .

Algoritmo 2.1

Dados $x^0 \in S$ e (r^k) uma seqüência de termos positivos

$k = 0$

Repita

$$x^{k+1} = \operatorname{argmin}_{x \in \hat{S}} \{g(x) + r^k D(x, x^k)\}$$

$$k = k + 1$$

Na figura 2.1 a seguir, g é dada por $x \in \mathbb{R} \mapsto g(x) = (x - 4)^2$ e a função regularizada é $x \in \mathbb{R} \mapsto g_k(x) = (x - 4)^2 + 7 \left(\frac{x}{2} - \log \frac{x}{2} - 1\right)$. Observe que $x^k = 2$ e $r^k = 7$, para algum valor de k dado.

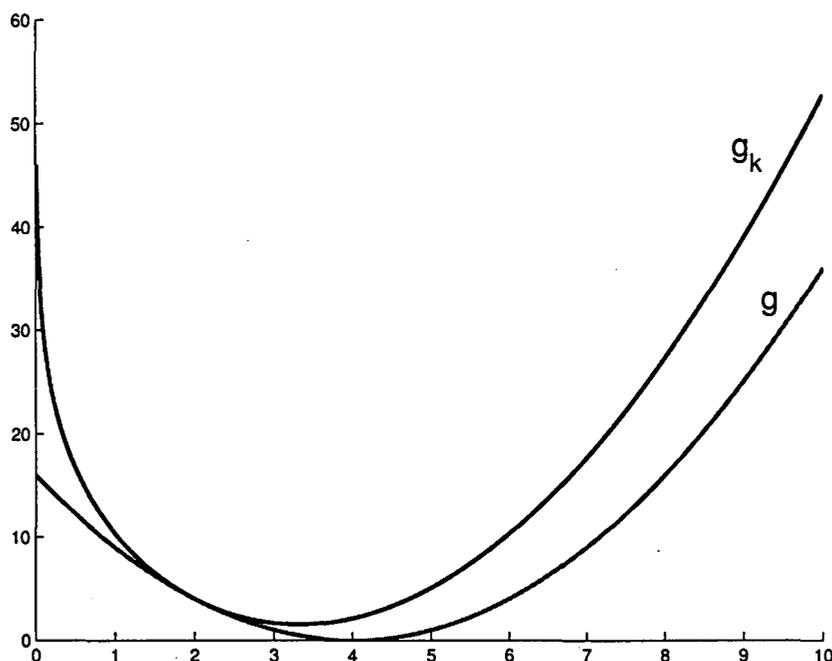


Figura 2.1: Função g e função regularizada g_k

No próximo teorema mostraremos que a seqüência gerada pelo algoritmo de ponto proximal 2.1, está bem definida no conjunto S .

Teorema 2.4 A seqüência (x^k) gerada pelo algoritmo de ponto proximal 2.1, aplicado ao problema (P1) está bem definida e contida em S .

Prova. Considere na iteração k do algoritmo 2.1, $y = x^k$, $r = r^k$ e o conjunto de nível

$$C = \{x \in \hat{S} : g(x) + rD(x, y) \leq g(y)\}.$$

Definimos a função

$$x \in \hat{S} \mapsto g_k(x) = g(x) + rD(x, y). \quad (2.3)$$

Dado que problema (P1) tem solução (hipótese (H2)), sabemos que g é limitada inferiormente em \hat{S} . Logo,

$$\inf\{g(x) : x \in \hat{S}\} > -\infty.$$

Fazendo $\bar{g} = \inf\{g(x) : x \in \hat{S}\}$ tem-se

$$C \subset \{x \in \hat{S} : rD(x, y) \leq g(y) - \bar{g}\},$$

segue da proposição 1.24 que C é limitado.

Como $g_k(\cdot) \geq g(\cdot)$, existe $\bar{z} = \inf\{g_k(x) : x \in \hat{S}\} > -\infty$ e sendo $\bar{z} \leq g(y)$ tem-se

$$\bar{z} = \inf\{g_k(x) : x \in C\}. \quad (2.4)$$

Queremos provar que existe $\bar{x} \in S$ tal que $g_k(\bar{x}) = \bar{z}$.

Por propriedade do ínfimo em (2.4), existe uma seqüência (x^i) em C tal que $g_k(x^i) \rightarrow \bar{z} > -\infty$, e $(g_k(x^i))$ é decrescente. Além disso, a seqüência (x^i) tem ponto de acumulação, pois C é limitado. Tomando subsequências, se necessário, podemos supor que existe $\bar{x} \in \text{cl}C \subset \text{cl}S$ tal que $x^i \rightarrow \bar{x}$.

Basta provar que $\bar{x} \in S$. Suponhamos por absurdo que $\bar{x} \in \partial S$, definimos $h = y - \bar{x}$. Então para $\lambda \in (0, 1]$, $\bar{x} + \lambda h \in S$. Como $D(\cdot, y)$ é coerciva na fronteira de S , $\lim_{\lambda \rightarrow 0} g'_k(\bar{x} + \lambda h, h) = -\infty$.

Portanto, para λ pequeno $g_k(\bar{x} + \lambda h)$ decresce com λ e existe $\bar{\lambda} \in (0, 1)$ e $\delta > 0$ tais que

$$g_k(\bar{x} + \bar{\lambda}h) < \lim_{\lambda \rightarrow 0} g_k(\bar{x} + \lambda h) - \delta.$$

Mas $\lim_{\lambda \rightarrow 0} g_k(\bar{x} + \lambda h) = \bar{z}$ e, portanto, $(\bar{x} + \bar{\lambda}h) \in S$ e

$$g_k(\bar{x} + \bar{\lambda}h) < \inf_{x \in S} g_k(x),$$

o que é absurdo. ■

Na seqüência apresentaremos a definição de distância de Bregman. Mostraremos algumas de suas propriedades e que todo ponto de acumulação da seqüência gerada pelo algoritmo de ponto proximal com Bregman é uma solução ótima do problema (P1).

Em [3] L. Bregman define as D -funções, hoje mais conhecidas como distâncias de Bregman. A definição dada por Bregman para as D -funções é menos geral do que a que faremos aqui, pois ele exigia outras propriedades, além da relação (2.5), a seguir.

Definição 2.5 Considere $S \subset \mathbb{R}^n$ um conjunto aberto e convexo e \hat{S} tal que $S \subset \hat{S} \subset clS$. Seja $h : \hat{S} \rightarrow \mathbb{R}$ uma função estritamente convexa e fechada em \hat{S} e diferenciável em S . Definimos a distância de Bregman induzida pela função h como

$$(x, y) \in \hat{S} \times S \mapsto D_h(x, y) = h(x) - h(y) - \langle \nabla h(y), x - y \rangle \quad (2.5)$$

onde $\nabla h(\cdot)$ é o gradiente da função h e $\langle \cdot, \cdot \rangle$ é o produto interno usual em \mathbb{R}^n .

Relembramos a nota 2.1 que apesar da inclusão $S \subset \hat{S} \subset clS$, estaremos interessados apenas nos casos em que $\hat{S} = S$ ou $\hat{S} = clS$, onde clS é o fecho do conjunto S .

A figura 2.2 mostra geometricamente a distância de Bregman para uma função h definida em \mathbb{R} . Note, neste caso, que o valor de $D_h(x, y)$ é igual ao valor de $h(x)$ menos a aproximação linear da função h no ponto y .

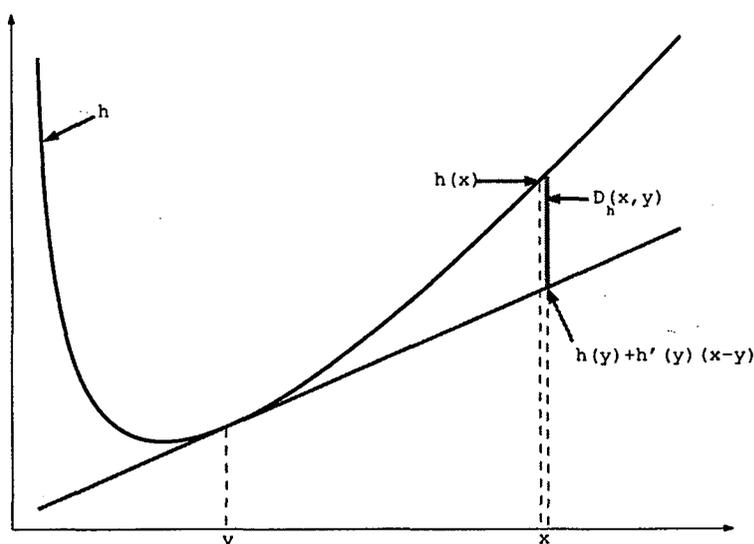


Figura 2.2: Distância de Bregman

Na proposição a seguir mostraremos algumas propriedades satisfeitas pela função D_h definida em (2.5) as quais serão usadas posteriormente.

Proposição 2.6 *Sejam h e D_h dadas como na definição (2.5), então*

- (i) $D_h(x, y) \geq 0$ e $D_h(x, y) = 0$ se, e somente se $x = y$, $\forall x \in \hat{S}$ e $\forall y \in S$;
- (ii) $D_h(\cdot, y)$ é estritamente convexa, $\forall y \in S$;
- (iii) Dado $\alpha \in \mathbb{R}$, os conjuntos de nível $\Gamma(x, y) = \{x \in \hat{S} : D(x, y) \leq \alpha\}$ são limitados, para todo $y \in S$;
- (iv) Para quaisquer $x \in \hat{S}$ e $y, z \in S$, tem-se $D_h(x, y) - D_h(x, z) - D_h(z, y) = \langle \nabla h(y) - \nabla h(z), z - x \rangle$;
- (v) Para quaisquer $x, y \in S$, $\nabla_x D_h(x, y) = \nabla h(x) - \nabla h(y)$.

Prova. (i) Como h é estritamente convexa, segue da desigualdade do gradiente, dada na definição 1.8, que

$$h(x) - h(y) - \langle \nabla h(y), x - y \rangle > 0$$

e, portanto, da relação (2.5), $D_h(x, y) > 0, \forall x \in \hat{S}$ e $\forall y \in S$.

(ii) Fixando $y \in \mathbb{R}^n$ e sendo h estritamente convexa, então D_h é escrita como soma de função estritamente convexa com função convexa e, portanto, é estritamente convexa.

(iii) Dos itens (i) e (ii) anteriores segue que D_h dada pela relação (2.5) é uma quase-distância e então este item está provado na proposição 1.24.

(iv) Da relação (2.5) tem-se para todo $x \in \hat{S}$ e $y, z \in S$:

$$D_h(x, y) = h(x) - h(y) - \langle \nabla h(y), x - y \rangle$$

$$D_h(x, z) = h(x) - h(z) - \langle \nabla h(z), x - z \rangle$$

$$D_h(z, y) = h(z) - h(y) - \langle \nabla h(y), z - y \rangle.$$

Subtraindo equações fica

$$D_h(x, y) - D_h(x, z) - D_h(z, y) = -\langle \nabla h(y), x - y \rangle + \langle \nabla h(z), x - z \rangle + \langle \nabla h(y), z - y \rangle = \langle \nabla h(y) - \nabla h(z), z - x \rangle, \quad \forall x \in \hat{S} \text{ e } y, z \in S.$$

(v) Derivando D_h dada na relação (2.5) em relação a variável x tem-se $\nabla_x D_h(x, y) = \nabla h(x) - \nabla h(y)$. ■

A propriedade (iv) da proposição anterior é conhecida como equação dos três pontos e apareceu pela primeira vez em [6].

O próximo teorema foi provado por Castilho-Gonzaga em [4]. Faremos a demonstração, readaptando ao nosso caso.

Teorema 2.7 *Considere os conjuntos $S \subset \mathbb{R}_{++}^n$ aberto e convexo e \hat{S} tal que $S \subset \hat{S} \subset \text{cl}S$. Se a função $h : \hat{S} \rightarrow \mathbb{R}$ é coerciva na fronteira de S , então em relação à seqüência (x^k) gerada pelo algoritmo de ponto proximal 2.1, aplicado ao problema (P1) com distância de Bregman, dada na definição 2.5, e (r^k) uma seqüência limitada superiormente, tem-se*

(i) *Se $\bar{x} \in \hat{S}$ e na iteração k do algoritmo $g(\bar{x}) < g(x^{k+1})$, então*

$$D_h(\bar{x}, x^{k+1}) \leq D_h(\bar{x}, x^k) - D_h(x^{k+1}, x^k) - \frac{1}{r^k}[g(x^{k+1}) - g(\bar{x})].$$

(ii) *A seqüência de valores $(g(x^k))$ converge ao valor ótimo g^* de (P1) e, portanto, todo ponto limite da seqüência (x^k) é solução ótima do problema (P1).*

Prova.

(i) Na iteração k do algoritmo de ponto proximal 2.1, tem-se

$$0 \in \partial[g(\cdot) + r^k \nabla D(\cdot, x^k)](x^{k+1}),$$

e utilizando propriedades de subdiferencial e soma de conjuntos fica

$$0 \in [\partial g(x^{k+1}) + r^k \nabla D(x^{k+1}, x^k)].$$

A última relação pode ser reescrita como $-r^k \nabla D(x^{k+1}, x^k) \in \partial g(x^{k+1})$ e pelo item (v) da proposição 2.6 tem-se

$$r^k [\nabla h(x^k) - \nabla h(x^{k+1})] \in \partial g(x^{k+1}).$$

Denotando $y^k = r^k [\nabla h(x^k) - \nabla h(x^{k+1})]$, segue do item (iv) da proposição 2.6 que

$$D_h(\bar{x}, x^k) - D_h(\bar{x}, x^{k+1}) - D_h(x^{k+1}, x^k) = \langle \nabla h(x^k) - \nabla h(x^{k+1}), x^{k+1} - \bar{x} \rangle =$$

$$\frac{1}{r^k} \langle y^k, x^{k+1} - \bar{x} \rangle \geq \frac{1}{r^k} [g(x^{k+1}) - g(\bar{x})].$$

Na última desigualdade usamos a definição de subgradiente. Transpondo termos, segue a prova do item (i).

(ii) Faremos por contradição. Sabendo que $(g(x^k))$ é uma seqüência decrescente e limitada, suponhamos que $\lim_{k \rightarrow \infty} g(x^k) > g(x^*)$, onde x^* é uma solução ótima do problema (P1). Logo, para algum $\varepsilon > 0$, tem-se $\lim_{k \rightarrow \infty} g(x^k) = g(x^*) + \varepsilon$.

Podemos escolher $\tilde{x} \in S$ tal que $g(\tilde{x}) < g(x^*) + \frac{\varepsilon}{2}$ e desta forma, para qualquer $k \in \mathbb{N}$, $g(x^k) > g(\tilde{x}) + \frac{\varepsilon}{2}$. Agora, do item (i) deste teorema juntamente com o item (i) da proposição 2.6, segue para cada $k \in \mathbb{N}$ que

$$D_h(\tilde{x}, x^{k+1}) \leq D_h(\tilde{x}, x^k) - \frac{1}{r^k} [g(x^{k+1}) - g(\tilde{x})] \leq D_h(\tilde{x}, x^k) - \frac{\varepsilon}{2r^k},$$

como (r^k) é uma seqüência limitada superiormente, segue que $D_h(\tilde{x}, x^k) \rightarrow -\infty$, contradizendo o fato de que D_h é uma quase-distância. ■

2.3 Ponto Proximal e Região de Confiança

Um método de região de confiança, a cada iteração, constrói um modelo local da função, que é minimizado em uma região. Nesta seção, consideraremos o caso em que o modelo é exato, no sentido que minimizamos a própria função e não o modelo, isto porque nosso interesse, nesse momento, é um estudo teórico relativo a esse método. Mostraremos que “a menos de uma constante” os subproblemas resolvidos são “equivalentes” (num sentido a ser definido a seguir) aos subproblemas resolvidos pelo método de ponto proximal.

O problema a ser tratado nesta seção tem a forma

$$(I) \quad \begin{array}{ll} \text{minimizar} & g(x) \\ \text{sujeito a} & x \in \mathbb{R}^n \end{array}$$

onde $g : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função convexa.

Uma iteração k do método de região de confiança aplicado ao problema (I), consiste em determinar, a partir de $x^k \in \mathbb{R}^n$ e $\Delta_k > 0$ dados, uma solução do subproblema

$$x^{k+1} \in \operatorname{argmin}_{x \in \mathbb{R}^n} \{g(x) : D(x, x^k) \leq \Delta_k\} \quad (2.6)$$

onde $D(\cdot, x^k)$ é uma quase-distância, definida em (1.22), por exemplo, $x \in \mathbb{R}^n \mapsto D(x, x^k) = \|x - x^k\|^2$, onde $\|\cdot\|$ é a norma euclidiana, muito utilizada nos métodos de

região de confiança e nos métodos de pontos proximais tradicionais (núcleos quadráticos). $\{x : D(x, x^k) \leq \Delta_k\}$ é a região de confiança e Δ_k é o “raio” da região de confiança.

Da mesma forma, uma iteração k do método de ponto proximal, aplicado ao problema (I), consiste em determinar, a partir de $x^k > 0$ e $r^k > 0$ dados e $D(., x^k)$ uma quase-distância dada, uma solução do subproblema

$$x^{k+1} \in \operatorname{argmin}_{x \in \mathbb{R}^n} \{g(x) + r^k D(x, x^k)\}. \quad (2.7)$$

Note que no algoritmo de ponto proximal 2.1 $x \in \hat{S}$. Na relação (2.7) $x \in \mathbb{R}^n$, isto porque o problema (I) tratado nesta seção é irrestrito.

Fato 2.8 *Os problemas (2.6) e (2.7) são equivalentes, no sentido que sendo x^* uma solução do problema (2.6), existe $r^k \geq 0$ tal que x^* é uma solução do problema (2.7). Da mesma forma, se x^* é uma solução do problema (2.7) existe $\Delta_k \geq 0$ tal que x^* é uma solução do problema (2.6).*

Prova. Dados $x^k \in \mathbb{R}^n$ e $\Delta_k \geq 0$ e x^* uma solução do problema (2.6), mostraremos que x^* é uma solução do problema (2.7).

Sendo D uma quase-distância, $D(x^k, x^k) = 0 \leq \Delta_k$ e a condição de Slater está satisfeita para o problema (2.6). Dado que x^* é uma solução do problema (2.6), existe $\lambda \geq 0$ tal que x^* é uma solução do seguinte problema

$$\operatorname{minimizar} \{\ell(x, \lambda) : x \in \mathbb{R}^n\} \quad (2.8)$$

onde

$$(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}_+ \mapsto \ell(x, \lambda) = g(x) + \lambda(D(x, x^k) - \Delta_k) \quad (2.9)$$

é a função lagrangeana associada ao problema (2.6).

Portanto, a primeira parte está demonstrada, pois uma solução do problema (2.8) também é uma solução do problema (2.7).

Para a segunda parte, consideraremos $x^k \in \mathbb{R}^n$ e $r^k \geq 0$ dados, $D(., x^k)$ uma quase-distância dada e x^* uma solução do problema (2.7). Devemos mostrar que x^* é uma solução do problema (2.6).

Se x^* uma solução do problema (2.7), também o será do seguinte problema

$$\operatorname{minimizar}_{x \in \mathbb{R}^n} \{g(x) + r^k D(x, x^k) - r^k \Delta_k\} \quad (2.10)$$

onde $\Delta_k \in \mathbb{R}$.

Observe que a função do problema (2.10) é a função lagrangeana

$$x \in \mathbb{R}^n \mapsto \ell(x, r^k) = g(x) + r^k(D(x, x^k) - \Delta_k). \quad (2.11)$$

Consideraremos

$$r \in \mathbb{R}_+ \mapsto d(r) = \inf\{\ell(x, r) : x \in \mathbb{R}^n\} \quad (2.12)$$

onde ℓ é a função lagrangeana dada na relação (2.9), com r no lugar de λ .

Como x^* é uma solução do problema (2.10), segue que

$$d(r^k) = g(x^*) + r^k(D(x^*, x^k) - \Delta_k). \quad (2.13)$$

Fazendo $\Delta_k = D(x^*, x^k)$, na relação (2.13) obtemos $d(r^k) = g(x^*)$, e portanto, pelo teorema forte de dualidade, x^* é uma solução do problema (2.6). ■

No lema seguinte, mostraremos que aumentando-se (diminuindo-se) o parâmetro r no problema (2.7), então diminui-se (aumenta-se) a região de confiança no problema (2.6), ou seja,

Lema 2.9 *Considere x_1 e x_2 soluções do problema (2.7), r_1 e r_2 as respectivas constantes associadas a estas soluções, tal que $r_1 > r_2 > 0$, então*

$$D(x_1, x^k) \leq D(x_2, x^k). \quad (2.14)$$

Prova. Considere $x^k \in \mathbb{R}^n$ e x_1, x_2 soluções do problema (2.7), tais que, r_1 e $r_2 > 0$ são as constantes relacionadas a estas soluções e $r_1 > r_2 > 0$. Então,

$$g(x_1) + r_1 D(x_1, x^k) \leq g(x_2) + r_1 D(x_2, x^k) \quad (2.15)$$

$$g(x_2) + r_2 D(x_2, x^k) \leq g(x_1) + r_2 D(x_1, x^k). \quad (2.16)$$

Somando as desigualdades (2.15) e (2.16) e agrupando termos fica

$$(r_1 - r_2)D(x_1, x^k) \leq (r_1 - r_2)D(x_2, x^k)$$

e, desde que $r_1 > r_2 > 0$, segue o resultado. ■

Apresentamos até aqui os resultados referentes ao método de ponto proximal. Na seqüência trataremos do método de lagrangeano aumentado e, no final, a relação entre os dois métodos.

2.4 Funções de Penalidades Coercivas Pela Direita

Definimos uma família de funções de penalidades que são coercivas à direita e posteriormente serão necessárias para definir o algoritmo de lagrangeano aumentado.

Usamos $p'(y, \mu)$ para denotar a derivada da função p em relação à primeira variável.

Definição 2.10 *Definimos a família \mathcal{P} de funções de penalidades*

$$y \in \mathbb{R}, \mu \in \mathbb{R}_{++} \mapsto p(y, \mu) \in \mathbb{R} \cup \{+\infty\}$$

com $\text{dom } p(\cdot, \mu) = (-\infty, b)$, $b > 0$, possivelmente $+\infty$ e p satisfazendo as seguintes propriedades: Para todo $\mu \in \mathbb{R}_{++}$,

$$(P1) \quad p(0, \mu) = 0;$$

$$(P2) \quad p(\cdot, \mu) \text{ é estritamente convexa e diferenciável em } (-\infty, b);$$

$$(P3) \quad p'(0, \mu) = \mu;$$

$$(P4) \quad p'_\infty(+1, \mu) = +\infty, \text{ ou, equivalentemente, } \lim_{y \rightarrow b} p'(y, \mu) = +\infty \text{ (coercividade pela direita);}$$

$$(P5) \quad p'_\infty(-1, \mu) = 0, \text{ ou, equivalentemente, } \lim_{y \rightarrow -\infty} p'(y, \mu) = 0.$$

Observe que para $\mu > 0$ as funções de penalidades $p(\cdot, \mu)$ da família \mathcal{P} , como definimos, passam pela origem, têm derivada μ na origem, são coercivas pela direita e são estritamente convexas no interior de seu domínio.

A propriedade (P1) não é necessária para o desenvolvimento teórico, pois se esta não for satisfeita, basta fazer uma translação. No entanto, por facilidade na exposição dos resultados admitimo-la satisfeita.

2.5 Propriedades da Conjugada de Uma Penalidade da Família \mathcal{P}

Se p é uma função de penalidade da família \mathcal{P} e $\mu \in \mathbb{R}_{++}$ está fixo, então a conjugada de p , segundo a definição (1.16), é da forma

$$s \in \mathbb{R} \mapsto p^*(s, \mu) = \sup\{sx - p(x, \mu) : x \in \mathbb{R}\}. \quad (2.17)$$

Proposição 2.11 *Considere $\mu \in \mathbb{R}_{++}$ e $p \in \mathcal{P}$. Se p é limitada inferiormente então $\text{dom } p^*(\cdot, \mu) = [0, +\infty)$ e se p é ilimitada inferiormente então $\text{dom } p^*(\cdot, \mu) = (0, +\infty)$.*

Prova. Das propriedades

$$p'(0, \mu) = \mu, \lim_{y \rightarrow -\infty} p'(y, \mu) = 0 \text{ e } \lim_{y \rightarrow b} p'(y, \mu) = +\infty,$$

dadas na definição de p , tem-se $p'(y, \mu) > 0$, para todo $y \in (-\infty, b)$. Logo, $p'((-\infty, b), \mu) = (0, +\infty)$.

Sabemos que p e p^* estão relacionadas (ver proposição 1.21) da seguinte forma

$$s = p'(y, \mu) \Leftrightarrow y = p^*(s, \mu). \quad (2.18)$$

Assim, de (2.18) e de $p'((-\infty, b), \mu) = (0, +\infty)$, tem-se que $s = p'(y, \mu) > 0$, para todo $y \in (-\infty, b)$.

Para $s = 0$,

$$\begin{aligned} p^*(0, \mu) &= \sup\{0 - p(y, \mu) : y \in (-\infty, b)\} \\ &= -\inf\{p(y, \mu) : y \in (-\infty, b)\}, \end{aligned}$$

segue que, se p é ilimitada inferiormente então

$$p^*(0, \mu) = -\inf\{p(y, \mu) : y \in (-\infty, b)\} = +\infty$$

e se p é limitada inferiormente por uma constante $c < 0$, então

$$p^*(0, \mu) = -\inf\{p(y, \mu) : y \in (-\infty, b)\} \geq 0.$$

Finalmente, se $s < 0$, $p^*(s, \mu) = +\infty$, pois p é crescente.

Portanto,

$$\text{dom } p^*(\cdot, \mu) = (0, +\infty),$$

para o caso em que p é ilimitada em $(-\infty, b)$ e

$$\text{dom } p^*(\cdot, \mu) = [0, +\infty),$$

para o caso em que p é limitada inferiormente por uma constante $c < 0$. ■

Da convexidade e diferenciabilidade da função $p(., \mu)$ em $(-\infty, b)$ segue que, $p^*(., \mu)$ é estritamente convexa em $[0, +\infty)$ para o caso limitado inferiormente e $(0, +\infty)$ para o caso ilimitado inferiormente (ver teorema 4.1.3 pag. 81 em [15]).

Da estrita convexidade de $p(., \mu)$ segue, para todo $y \in (-\infty, b)$ e $\mu \in \mathbb{R}_{++}$, que $p^*(., \mu)$ é continuamente diferenciável em $(0, +\infty)$ (ver teorema 4.1.1, pag. 79 em [15]).

Nota 2.12 Até o final do capítulo usaremos $\hat{S} = \mathbf{R}_+^m$ para o caso em que p é limitada inferiormente e $\hat{S} = \mathbf{R}_{++}^m$ para o caso em que p é ilimitada inferiormente.

Definição 2.13 Uma função $h : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ é separável se $h(x) = \sum_{i=1}^m h_i(x_i)$, onde $h_i : \mathbb{R} \rightarrow \mathbb{R} \cup \{+\infty\}$, para todo $i = 1, 2, \dots, m$.

Proposição 2.14 Considere p uma penalidade da família \mathcal{P} . Então a função separável D , definida em função da conjugada de p pela fórmula

$$s \in \hat{S}, \mu \in S \mapsto D(s, \mu) = \sum_{i=1}^m p^*(s_i, \mu_i), \quad (2.19)$$

é uma quase-distância em $\hat{S} \times S$.

Prova. Primeiramente, mostraremos que $D(\mu, \mu) = 0$, para cada $\mu \in \hat{S}$.

Considere $\mu \in \hat{S}$ e $p \in \mathcal{P}$. Então, do item (P3) da definição de p , $p'(0, \mu_i) = \mu_i$, para todo $i = 1, 2, \dots, m$, e usando a proposição 1.21 segue que

$$p^*(\mu_i, \mu_i) = \langle \mu_i, 0 \rangle - p_i(0, \mu_i), \quad \text{para todo } i = 1, 2, \dots, m. \quad (2.20)$$

Somando as m equações na igualdade (2.20) segue que $D(\mu, \mu) = 0$, pois $\langle \mu, 0 \rangle = 0$ e $p(0, \mu) = 0$.

Considere agora $s \in \hat{S}$ e $\mu \in S$, tais que $s \neq \mu$. Mostraremos que $D(s, \mu) > 0$.

Do item (P3) da definição de p , isto é, $p'(0, \mu_i) = \mu_i$, juntamente com a proposição 1.21, segue que $0 \in \partial p^*(\mu_i, \mu_i)$. Da definição de subdiferencial

$$\langle 0, s_i - \mu_i \rangle \leq p^*(s_i, \mu_i) - p^*(\mu_i, \mu_i), \quad \forall i = 1, 2, \dots, m. \quad (2.21)$$

Sabemos que $p^*(., \mu)$, é estritamente convexa em \hat{S} , então

$$p^*(s_i, \mu_i) > p^*(\mu_i, \mu_i) + \langle 0, s_i - \mu_i \rangle, \quad \text{para todo } i = 1, 2, \dots, m \text{ tal que } s_i \neq \mu_i. \quad (2.22)$$

Somando as m inequações em (2.22) e usando $D(\mu, \mu) = 0$ e $\langle 0, s - \mu \rangle = 0$, segue que $D(s, \mu) > 0$. ■

2.6 Construindo Penalidades da Família \mathcal{P}

Vimos, na última seção, como são as funções de penalidades que vamos utilizar na definição do algoritmo de lagrangeano aumentado. Além disso, vimos como são suas conjugadas que serão depois utilizadas no algoritmo de ponto proximal para fazer a ponte com o algoritmo de lagrangeano aumentado.

Em seguida veremos metodologias de construir penalidades da família \mathcal{P} . Basicamente, todas usam uma função auxiliar de uma variável real.

Definição 2.15 Considere $\theta : \mathbb{R} \rightarrow \mathbb{R} \cup \{+\infty\}$ uma função satisfazendo as seguintes propriedades

- (a) $\text{intdom } \theta = (-\infty, b)$, $b > 0$, possivelmente $b = +\infty$;
- (b) $\theta(\cdot)$ é estritamente convexa e diferenciável em $(-\infty, b)$;
- (c) $\lim_{y \rightarrow b} \theta'(y) = +\infty$ (coercividade pela direita);
- (d) $\lim_{y \rightarrow -\infty} \theta'(y) = 0$;
- (e) $\theta(0) = 0$ e $\theta'(0) = 1$.

A função exponencial, $y \in \mathbb{R} \mapsto \theta(y) = \exp(y) - 1$, vista anteriormente no exemplo (1.18), satisfaz as propriedades (a) – (e) acima e este é um caso em que $b = +\infty$, isto é, $\text{intdom } \theta = (-\infty, +\infty)$. A função barreira logaritmica modificada, $y \in \mathbb{R} \mapsto \theta(y) = -\log(y - 1)$ se $y < 1$, $\theta(y) = +\infty$ se $y \geq 1$, vista anteriormente no exemplo (1.17), também satisfaz as propriedades (a) – (e) acima, com $b = 1$.

Tipo 1: Considere $\theta : \mathbb{R} \rightarrow \mathbb{R} \cup \{+\infty\}$ satisfazendo as propriedades (a)-(e) dadas acima. Então, p dada por

$$(y, \mu) \in \mathbb{R} \times \mathbb{R}_{++} \mapsto p(y, \mu) = \theta(\mu y) \in \mathbb{R} \cup \{+\infty\} \quad (2.23)$$

é uma penalidade da família \mathcal{P} .

A conjugada de p tipo 1, segundo o item (iii) da proposição 1.19 é dada por

$$(s, \mu) \in \mathbb{R} \times \mathbb{R}_{++} \mapsto p^*(s, \mu) = \theta^*\left(\frac{s}{\mu}\right) \in \mathbb{R} \cup \{+\infty\}. \quad (2.24)$$

Exemplo 2.16 Considere $\theta : \mathbb{R} \rightarrow \mathbb{R} \cup \{+\infty\}$ definida por

$$\theta(y) = \begin{cases} -\log(1-y) & , \text{ se } y < 1 \\ +\infty & , \text{ se } y \geq 1 \end{cases} \quad (2.25)$$

exibida anteriormente no exemplo 1.17. Assim, p obtida a partir de θ fica

$$y \in \mathbb{R}, \mu \in \mathbb{R}_{++} \mapsto p(y, \mu) = \begin{cases} -\log(1-\mu y) & , \text{ se } \mu y < 1 \\ +\infty & , \text{ c.c.} \end{cases} \quad (2.26)$$

A conjugada de p , para este caso, é da forma

$$s \in \mathbb{R}, \mu \in \mathbb{R}_{++} \mapsto p^*(s, \mu) = \begin{cases} \left(\frac{s}{\mu}\right) - \log\left(\frac{s}{\mu}\right) - 1 & , \text{ se } s > 0 \\ +\infty & , \text{ c.c.} \end{cases} \quad (2.27)$$

A penalidade tipo 1 é a que estamos propondo neste trabalho. Em termos de lagrangeano aumentado ela é nova e não encontramos na literatura prova de convergência deste algoritmo utilizando esta penalidade.

Tipo 2: Considere $\theta : \mathbb{R} \rightarrow \mathbb{R} \cup \{+\infty\}$ satisfazendo as propriedades (a)-(e) dadas no início da seção. Então, p dada por

$$(y, \mu) \in \mathbb{R} \times \mathbb{R}_{++} \mapsto p(y, \mu) = \mu\theta(y) \in \mathbb{R} \cup \{+\infty\} \quad (2.28)$$

é uma penalidade da família \mathcal{P} .

A conjugada de p tipo 2, segundo o item (ii) da proposição 1.19, é dada por

$$(s, \mu) \in \mathbb{R} \times \mathbb{R}_{++} \mapsto p^*(s, \mu) = \mu\theta^*\left(\frac{s}{\mu}\right) \in \mathbb{R} \cup \{+\infty\}. \quad (2.29)$$

Exemplo 2.17 Considere θ a função definida no exemplo anterior. Então,

$$y \in \mathbb{R}, \mu \in \mathbb{R}_{++} \mapsto p(y, \mu) = \begin{cases} -\mu \log(1-y) & , \text{ se } y < 1 \\ +\infty & , \text{ c.c.} \end{cases} \quad (2.30)$$

é uma penalidade do tipo 2. Sua conjugada é da forma

$$s \in \mathbb{R}, \mu \in \mathbb{R}_{++} \mapsto p^*(s, \mu) = \begin{cases} s - \mu \log\left(\frac{s}{\mu}\right) - \mu & , \text{ se } s > 0 \\ +\infty & , \text{ c.c.} \end{cases} \quad (2.31)$$

A penalidade tipo 2 da relação (2.28) é a mais utilizada na literatura. No caso em que θ é a função exponencial, a conjugada da penalidade (2.28) é a entropia de Kullback-Leibler. Por ser a mais utilizada na literatura, é a escolhida para comparar com a que estamos propondo.

Note que a diferença entre as penalidades tipo 1 e 2 está na posição em que o parâmetro $\mu \in \mathbb{R}_{++}$ ocupa em cada uma destas. Nas penalidades tipo 1, μ multiplica o argumento da função θ , já nas do tipo 2, μ multiplica a função θ . Veremos, mais adiante, como isto influi no método.

Uma outra metodologia de construir penalidades, baseada na função real θ , é utilizar deslocamentos. Em [4], Cástilho-Gonzaga constroem a partir da função $\theta: \mathbb{R} \rightarrow \mathbb{R} \cup \{+\infty\}$ e do parâmetro $\mu \geq 0$ penalidades da forma

$$y \in \mathbb{R}, \mu \in \mathbb{R}_+ \mapsto p(y, \mu) = \theta(y + y_\mu) - \theta(y_\mu) \in \mathbb{R} \cup \{+\infty\} \quad (2.32)$$

onde y_μ é tal que $\theta'(y_\mu) = \mu$.

Neste caso, é permitido que o parâmetro μ se anule, usando funções $\theta(\cdot)$ constantes para y muito negativo, o que não pode ocorrer nas penalidades do tipo 1 e 2, como será visto mais adiante. Para mais detalhes sobre este tipo de penalidades ver [4, 22].

Finalizando a parte de construção de penalidades, comentaremos uma metodologia que segue o caminho inverso ao que seguimos, mas que conduz aos mesmos resultados. Tal metodologia parte de uma função de Bregman h no método de ponto proximal, aplicado ao problema dual (este a ser definido adiante), e depois usa a conjugada da função h no método de lagrangeano aumentado, aplicado ao problema primal [8, 31].

Apresentaremos em seguida os métodos de lagrangeano aumentado.

2.7 Métodos de Lagrangeano Aumentado

Esboçamos na introdução várias informações sobre a classe de métodos lagrangeano aumentado. Lembramos que na sua forma original foram introduzidos para resolver problemas com restrições de igualdade e posteriormente generalizados para problemas com restrições de desigualdade. Neste trabalho, iremos aplicá-los a proble-

mas com restrições de desigualdade, ou seja

$$(P) \quad \begin{array}{ll} \text{minimizar} & f_0(x) \\ \text{sujeito a} & f_i(x) \leq 0, \quad i = 1, 2, \dots, m \\ & x \in \mathbb{R}^n \end{array}$$

onde $f_i : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$, para $i = 0, 1, \dots, m$ são funções convexas, próprias e fechadas.

Usamos Ω^* para denotar o conjunto de soluções ótimas de (P) e consideraremos as seguintes hipóteses:

Hipótese H3: Ω^* é não vazio e compacto.

Hipótese H4: Condição de qualificação de Slater: existe \bar{x} no domínio de f_0 tal que $f_i(\bar{x}) < 0$, para todo $i = 1, \dots, m$.

A hipótese (H3) é usada para assegurar a boa definição do algoritmo de lagrangeano aumentado, que será definido adiante, e a hipótese (H4) garante que as condições de Karush-Kuhn-Tucker são necessárias e suficientes para o problema (P).

2.8 A Função Lagrangeano Aumentado

Em relação ao problema (P), definimos a função lagrangeano aumentado como

$$(x, \mu, r) \in \mathbb{R}^n \times \mathbb{R}_+^m \times \mathbb{R}_{++} \mapsto L(x, \mu, r) \in \mathbb{R} \cup \{+\infty\}$$

onde

$$L(x, \mu, r) = f_0(x) + r \sum_{i=1}^m p\left(\frac{f_i(x)}{r}, \mu_i\right) \quad (2.33)$$

e p é uma função de penalidade da família \mathcal{P} .

Note que a função lagrangeano aumentado é formada pela função objetivo f_0 somada a uma penalização das restrições.

2.9 O Algoritmo de Lagrangeano Aumentado

O algoritmo de lagrangeano aumentado aplicado ao problema (P) com penalidade pertencente a família \mathcal{P} , gera seqüências (x^k) e (μ^k) , a partir de $\mu^0 > 0$ fixo e (r^k) uma seqüência de termos positivos e limitada, da seguinte forma:

Algoritmo 2.2

Dados $\mu^0 > 0$, $\bar{r} > 0$ e (r^k) uma seqüência em $(0, \bar{r})$.

$k = 0$

Repita

Encontre

$$x^{k+1} \in \operatorname{argmin}\{L(x, \mu^k, r^k) : x \in \mathbb{R}^n\}$$

Atualize

$$\mu_i^{k+1} = p'\left(\frac{f_i(x^{k+1})}{r^k}, \mu_i^k\right), \quad i = 1, 2, \dots, m.$$

$$k = k + 1$$

onde $L(\cdot, \mu^k, r^k)$ é a função lagrangeano aumentado definida na relação (2.33), r^k é um parâmetro de penalidade e μ^k é um vetor cujas componentes são multiplicadores de Lagrange.

A seqüência (x^k) gerada pelo algoritmo esta bem definida (ver afirmação 4.21, pag. 41 em [10]).

A seqüência (x^k, μ^k) é chamada de primal-dual. Em geral (μ^k) é sempre viável (dual) para $k > 0$ e espera-se que (x^k) vá se tornando viável à medida que k cresce.

Observe que a tarefa pesada do algoritmo 2.2 é a minimização da função lagrangeano aumentado, ao que, em geral chamamos de subproblema gerado pelo algoritmo. A metodologia do algoritmo é bastante simples. Cada iteração consiste em formar os subproblemas irrestritos, que podem ser resolvidos por qualquer método de minimização irrestrita, e atualizar os parâmetros.

2.10 Relação Primal Dual

A relação primal dual é uma relação envolvendo o algoritmo de lagrangeano aumentado, aplicado ao problema primal e o algoritmo de ponto proximal, aplicado ao problema dual com quase-distância dada pela conjugada da penalidade utilizada no primal. Assim, definimos o problema dual de (P) e em seguida apresentaremos o resultado de equivalência.

Definição 2.18 Associado ao problema (P) definimos o problema dual, da seguinte forma

$$(D) \quad \begin{array}{l} \text{minimizar} \quad -d(\mu) \\ \text{sujeito a} \quad \mu \geq 0 \end{array}$$

onde

$$\mu \in \mathbb{R}_+^m \mapsto d(\mu) = \inf\{\ell(x, \mu) : x \in \mathbb{R}^n\} \quad (2.34)$$

é a função dual lagrangeana e

$$x \in \mathbb{R}^n, \mu \in \mathbb{R}^m \mapsto \ell(x, \mu) = f_0(x) + \sum_{i=1}^m \mu_i f_i(x) \quad (2.35)$$

é a função lagrangeana.

Optamos por $-d(\cdot)$ no problema dual, para manter o padrão de problema convexo em todo trabalho.

Em seguida, reescrevemos o algoritmo de ponto proximal 2.1, agora aplicado ao problema (D) e usando a quase-distância dada na relação (2.19).

Considere S e \hat{S} definidos na nota 2.1 e $(\mu, \mu^k) \in \mathbb{R}_+^m \times \mathbb{R}_{++}^m \mapsto D(\mu, \mu^k) = \sum_{i=1}^m p^*(\mu_i, \mu_i^k)$, então

Algoritmo 2.3

Dados $\mu^0 \in S$, $\bar{r} > 0$ e (r^k) uma seqüência em $(0, \bar{r})$,

$k = 0$

Repita

Encontre

$$\mu^{k+1} = \operatorname{argmin}\{-d(\mu) + r^k D(\mu, \mu^k) : \mu \in \hat{S}\}$$

$k = k + 1$

Se (μ^k) é uma seqüência gerada pelo algoritmo 2.3 então a desigualdade a seguir sai naturalmente

$$-d(\mu^{k+1}) + r^k D(\mu^{k+1}, \mu^k) \leq -d(\mu^k) \quad (2.36)$$

pois, D é uma quase-distância e pela definição 1.22, $D(\mu^k, \mu^k) = 0$, além disso $(-d(\mu^k))$ é uma seqüência decrescente.

É importante destacar como esta questão de dualidade está presente por todo o nosso trabalho (às vezes implicitamente). Por um lado, está o método de lagrangeano aumentado, por outro o método de ponto proximal. O próximo teorema relaciona os métodos de lagrangeano aumentado e de ponto proximal. Este resultado foi, primeiramente, obtido por Rockafellar [28, 29], para problemas com restrições de desigualdade, via teoria de operador monótono maximal. Mais tarde, generalizou-se o resultado para métodos de ponto proximal utilizando quase-distâncias.

Teorema 2.19 *Considere o problema (P) satisfazendo as hipóteses (H3) e (H4), (x^k, μ^k) seqüências geradas pelo algoritmo de lagrangeano aumentado 2.2, aplicado ao problema (P) com penalidade $p \in \mathcal{P}$, e $(\hat{\mu}^k)$ uma seqüência gerada pelo algoritmo de ponto proximal 2.3, aplicado ao problema (D) com quase-distância dada pela conjugada de p , determinada na relação (2.19). Se $\mu^0 = \hat{\mu}^0$, então $\mu^k = \hat{\mu}^k$, para todo $k \in \mathbb{N}$.*

Prova. (ver teorema 4.33 em [10]) ■

O teorema 2.19 nos diz que a seqüência de multiplicadores gerada pelo algoritmo de lagrangeano aumentado é a gerada pelo algoritmo de ponto proximal, quando este é aplicado ao dual do problema (P) e com conjugada da penalidade utilizada no algoritmo de lagrangeano aumentado.

Mostraremos no próximo teorema que o algoritmo de lagrangeano aumentado 2.2, com penalidade p tipo 1, dada na relação (2.23), gera seqüências (x^k) e (μ^k) que satisfazem a condição de complementaridade para o problema (P).

Teorema 2.20 *Considere (x^k) e (μ^k) seqüências geradas pelo algoritmo de lagrangeano aumentado 2.2, aplicado ao problema (P) com penalidade tipo 1, dada na relação (2.23), e (r^k) uma seqüência em (r_1, r_2) tais que $0 < r_1 < r_2$. Nestas condições, tem-se*

$$\lim_{k \rightarrow \infty} \frac{\mu_i^k y_i^{k+1}}{r^k} = 0, \quad \forall i = 1, 2, \dots, m,$$

onde $y_i^{k+1} = f_i(x^{k+1})$, $i = 1, 2, \dots, m$.

Prova. A prova será feita por absurdo. Suponhamos que existam $\varepsilon > 0$, $\ell \in \{1, 2, \dots, m\}$ e uma seqüência com índices em $\mathcal{K} \subset \mathbb{N}$, tal que

$$\left| \frac{\mu_\ell^k y_\ell^{k+1}}{r^k} \right| > \varepsilon, \quad \text{para todo } k \in \mathcal{K}.$$

Façamos a prova em três partes:

(a) Suponhamos que $\frac{\mu_i^k y_i^{k+1}}{r^k} > \varepsilon, \quad \forall k \in \mathcal{K}$.

Das propriedades que definem a função θ , seção 2.6, sabemos que esta é estritamente crescente em \mathbb{R} e que $\theta'(0) = 1$, segue que $\theta'(\varepsilon) > 1$.

Considere $\theta'(\varepsilon) = \beta$. Da fórmula de atualização dos multiplicadores no algoritmo de lagrangeano aumentado, tem-se que

$$\mu_i^{k+1} = p' \left(\frac{y_i^{k+1}}{r^k}, \mu_i^k \right), \quad \forall i = 1, 2, \dots, m \quad e \quad \forall k \in \mathbb{N}. \quad (2.37)$$

Sendo p uma penalidade do tipo 1, esta última relação fica

$$\mu_i^{k+1} = \mu_i^k \theta' \left(\frac{\mu_i^k y_i^{k+1}}{r^k} \right), \quad \forall i = 1, 2, \dots, m \quad e \quad \forall k \in \mathbb{N}. \quad (2.38)$$

Da hipótese $\frac{\mu_i^k y_i^{k+1}}{r^k} > \varepsilon > 0, \forall k \in \mathcal{K}$, juntamente com $\theta'(\varepsilon) = \beta$, segue que

$$\mu_i^{k+1} > \mu_i^k \beta, \quad \forall k \in \mathcal{K}. \quad (2.39)$$

Por outro lado, se (μ^k) é obtida pelo algoritmo de ponto proximal 2.3 então da relação (2.36), tem-se

$$d(\mu^{k+1}) \geq d(\mu^k) + r^k D(\mu^{k+1}, \mu^k), \quad \forall k \in \mathbb{N}. \quad (2.40)$$

Mostramos na proposição 2.14 que a função D , dada por $s \in \hat{S}, \mu \in S \mapsto D(s, \mu) = \sum_{i=1}^m p^*(s_i, \mu_i)$ é uma quase-distância em $\hat{S} \times S$. Como p^* é a conjugada de p , segue da relação (2.24) juntamente com o item (ix) da proposição 1.19 que

$$D(\mu^{k+1}, \mu^k) = \sum_{i=1}^m \theta^* \left(\frac{\mu_i^{k+1}}{\mu_i^k} \right), \quad \forall k \in \mathbb{N}. \quad (2.41)$$

A soma do lado direito da relação (2.41) é de termos não negativos, pois D é uma quase-distância. Então, das relações (2.39) e (2.41) segue que

$$D(\mu^{k+1}, \mu^k) > \theta^*(\beta), \quad \forall k \in \mathcal{K}. \quad (2.42)$$

Fazendo $\theta^*(\beta) = \gamma > 0$ em (2.42) e substituindo em (2.40),

$$d(\mu^{k+1}) \geq d(\mu^k) + r^k \gamma, \quad \forall k \in \mathcal{K}, \quad (2.43)$$

ou equivalentemente,

$$d(\mu^{k+1}) - d(\mu^k) \geq r^k \gamma, \quad \forall k \in \mathcal{K}, \quad (2.44)$$

que é uma contradição, pois da relação (2.40) ($d(\mu^k)$) é crescente e, da hipótese (H4), é limitada (teorema 2.32, pg. 312, vol. I em [14]). Então, $d(\mu^{k+1}) - d(\mu^k) \rightarrow 0$.

(b) Suponhamos que $\frac{\mu_\ell^k y_\ell^{k+1}}{r^k} < -\varepsilon, \forall k \in \mathcal{K}$ e $\varepsilon > 0$.

Sendo $-\varepsilon < 0$, das propriedades que definem a função θ , dadas na seção 2.6, tem-se que $\theta'(-\varepsilon) < 1$.

Considere $\theta'(-\varepsilon) = \alpha < 1$. Pela proposição 1.21, $\theta^{*'}(\alpha) = -\varepsilon < 0$. Além disso, $\theta'(0) = 1$. Logo, θ^* é decrescente em $(0, 1)$.

Da mesma forma que fizemos na parte (a), a relação 2.39 fica

$$\mu_\ell^{k+1} < \mu_\ell^k \alpha, \quad \forall k \in \mathcal{K}. \quad (2.45)$$

Como $0 < \alpha < 1$ e θ^* é decrescente em $(0, 1)$, segue que

$$\theta^* \left(\frac{\mu_\ell^{k+1}}{\mu_\ell^k} \right) > \theta^*(\alpha) > 0, \quad \forall k \in \mathcal{K}.$$

Logo, a relação (2.41) fica

$$D(\mu^{k+1}, \mu^k) > \theta^*(\alpha) > 0, \quad \forall k \in \mathcal{K}. \quad (2.46)$$

Das relações (2.40) e (2.46) segue que

$$d(\mu^{k+1}) > d(\mu^k) + r^k \theta^*(\alpha), \quad \text{para todo } k \in \mathcal{K}. \quad (2.47)$$

Portanto, a prova segue como na parte (a) deste teorema, a partir da relação (2.43).

(c) Finalmente, pode ocorrer que existe uma infinidade de pontos em que $\frac{\mu_\ell^k y_\ell^{k+1}}{r^k} > \varepsilon$, ou uma infinidade de pontos em que $\frac{\mu_\ell^k y_\ell^{k+1}}{r^k} < -\varepsilon$. Em qualquer um desses dois casos, basta tomar uma subsequência e recaímos ou no caso (a) ou no caso (b) anteriores. ■

Capítulo 3

A Penalidade M^2B e Resultados de Convergência

Neste capítulo, mostramos o resultado principal do nosso trabalho.

Iniciamos definindo uma função de penalidade, que chamamos de M^2B , construída utilizando uma penalidade tipo 1, dada no capítulo anterior, juntamente com a função barreira logarítmica modificada de Polyak, apresentada anteriormente no exemplo 1.17. Esta metodologia é nova em termos de lagrangeano aumentado e não encontramos na literatura nenhum tipo de prova de convergência sobre a seqüência dual gerada por estes. Aqui, mostramos que todo ponto de acumulação da seqüência dual gerada pelos algoritmos de lagrangeano aumentado, com essa nova metodologia, é uma solução ótima dual do problema (P), definido na introdução. Este resultado foi obtido mostrando-se que a conjugada da penalidade M^2B conduz a uma distância de Bregman no dual e, então, como consequência dos resultados mostrados no capítulo anterior, nosso resultado fica imediato.

Apresentamos ainda outro resultado inédito neste capítulo, que será muito útil quando implementarmos o algoritmo de lagrangeano aumentado. Este resultado garante que a inicialização dos subproblemas gerados pelos algoritmos de lagrangeano aumentado com a nova metodologia proposta é sempre possível, ou seja, mostramos que, ajustando uma função quadrática à função barreira logarítmica modificada de Polyak (lembre-se que o domínio desta não é toda a reta, e que em muitos pontos ela pode não estar definida, ver o exemplo 1.17), ainda temos garantia dos resultados já obtidos.

No final do capítulo, mostramos que as curvas de nível da aproximação quadrática da conjugada da penalidade M^2B têm o formato de elipsóides de Dikin. Isto é muito

bom, pois os algoritmos tipo afim escala têm boas propriedades de convergência. Este resultado não vale para penalidades do tipo 2 que são as mais referenciadas na literatura.

3.1 Definição da Penalidade M^2B

Considere a função barreira logarítmica modificada de Polyak, $x \in \mathbb{R} \mapsto \theta(x) \in \mathbb{R} \cup +\{\infty\}$, dada por

$$\theta(x) = \begin{cases} -\log(1-x) & , \text{ se } x < 1 \\ +\infty & , \text{ se } x \geq 1. \end{cases} \quad (3.1)$$

conhecida na literatura como MB .

Utilizando a função de penalidade p tipo 1, dada na relação (2.23), ou seja

$$y \in \mathbb{R}, \mu \in \mathbb{R}_{++} \mapsto p(y, \mu) = \theta(\mu y) \in \mathbb{R} \cup +\{\infty\} \quad (3.2)$$

e θ dada na relação (3.1) definimos M^2B .

Definição 3.1 Considere θ a função dada pela relação (3.1) e p a penalidade dada pela relação (3.2). Chamamos de M^2B a função dada por

$$x \in \mathbb{R}, \mu \in \mathbb{R}_{++} \mapsto p(x, \mu) = \begin{cases} -\log(1-\mu x) & , \text{ se } \mu x < 1 \\ +\infty & , \text{ c.c.} \end{cases} \quad (3.3)$$

Na figura 3.1, traçamos o gráfico da penalidade M^2B para para três valores diferentes de μ .

O nome M^2B sugere alguma coisa como barreira logarítmica duplamente modificada, já que a função (3.1) é conhecida como função barreira logarítmica modificada, e em geral, é referenciada por MB .

Considere p dada na relação (3.2) e P uma função separável (ver definição 2.13) definida da seguinte maneira

$$y \in \mathbb{R}^m, \mu \in \mathbb{R}_{++}^m \mapsto P(y, \mu) = \sum_{i=1}^m p(y_i, \mu_i). \quad (3.4)$$

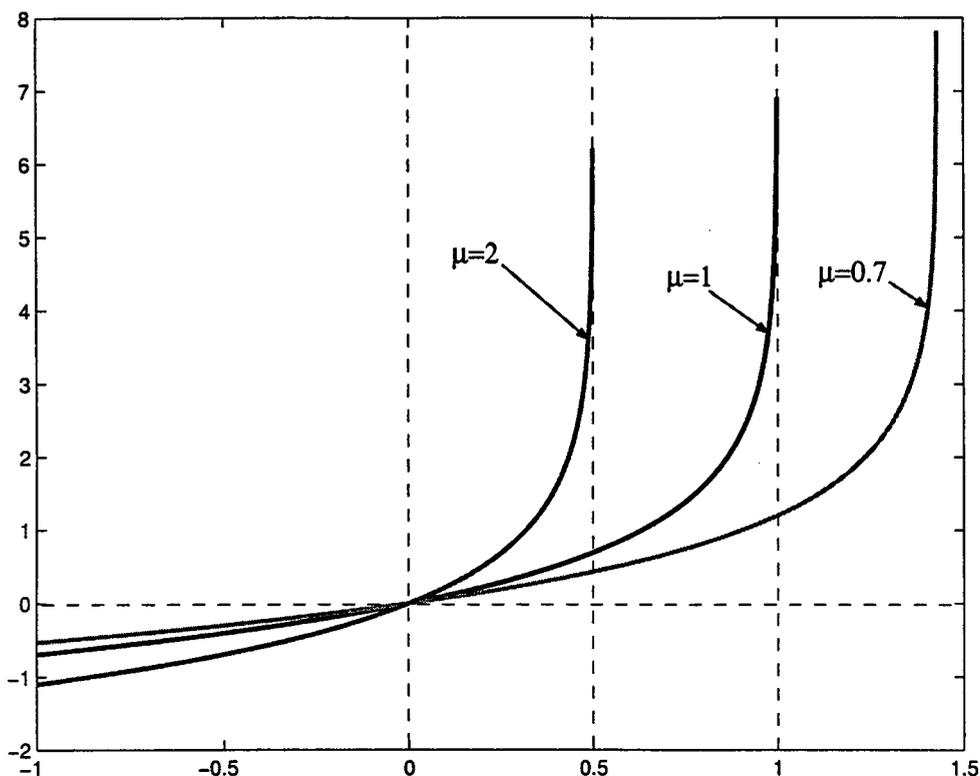


Figura 3.1: Gráfico da penalidade M^2B para três valores diferentes de μ

Vimos que a conjugada de P tem a forma (ver relação (2.41))

$$s, \mu \in S \mapsto D(s, \mu) = \sum_{i=1}^m \theta^* \left(\frac{s_i}{\mu_i} \right). \quad (3.5)$$

Agora, se θ é a função barreira logarítmica modificada (3.1), a relação (3.5) fica

$$s, \mu \in S \mapsto D(s, \mu) = \sum_{i=1}^m \left(\frac{s_i}{\mu_i} - \log \frac{s_i}{\mu_i} - 1 \right). \quad (3.6)$$

No próximo teorema mostraremos que D dada na relação (3.6) define uma distância de Bregman.

Teorema 3.2 *Considere $S = \mathbb{R}_{++}^m$, θ a função barreira logarítmica modificada (3.1) e p a penalidade M^2B dada na relação (3.3). Então, a função $s \in S \mapsto h(s) = \sum_{i=1}^m \theta^*(s_i) \in \mathbb{R}$ gera uma distância de Bregman, que coincide com (3.6)*

Prova. Como θ é a função barreira logarítmica modificada (3.1), sua conjugada foi determinada no exemplo 1.17 e tem a forma $s \in \mathbb{R} \mapsto \theta^*(s) = s - \log s - 1$, se

$s > 0$; e $\theta^*(s) = +\infty$ se $s \leq 0$.

Considere $s \in S$. Então, $h(s) = \sum_{i=1}^m (s_i - \log s_i - 1)$ onde $s_i \in (0, +\infty)$, para todo $i = 1, 2, \dots, m$. Da definição de distância de Bregman, relação (2.5), para cada $s, \mu \in S$, tem-se

$$\begin{aligned}
 D_h(s, \mu) &= h(s) - h(\mu) - \langle \nabla h(\mu), s - \mu \rangle \\
 &= \sum_{i=1}^m \theta^*(s_i) - \sum_{i=1}^m \theta^*(\mu_i) - \sum_{i=1}^m (\theta^*)'(\mu_i)(s_i - \mu_i) \\
 &= \sum_{i=1}^m (s_i - \log s_i - 1) - \sum_{i=1}^m (\mu_i - \log \mu_i - 1) - \sum_{i=1}^m \left(1 - \frac{1}{\mu_i}\right) (s_i - \mu_i) \\
 &= \sum_{i=1}^m \left(\log \mu_i - \log s_i + \frac{s_i}{\mu_i} - 1 \right) \\
 &= \sum_{i=1}^m \left(\frac{s_i}{\mu_i} - \log \frac{s_i}{\mu_i} - 1 \right).
 \end{aligned}$$

■

Como consequência do teorema 3.2 e do teorema 2.7 tem-se o seguinte resultado.

Corolário 3.3 *Considere as seqüências (x^k) e (μ^k) geradas pelo algoritmo de lagrangeano aumentado 2.2, aplicado ao problema (P), com penalidade M^2B dada na definição 3.1. Se a seqüência (r^k) é limitada, então a seqüência (μ^k) é limitada e todo ponto de acumulação de (μ^k) é uma solução ótima do problema (D).*

Prova. Pelo teorema 3.2, a conjugada da penalidade M^2B é uma distância de Bregman. Segue do teorema 2.7 que todo ponto de acumulação da seqüência (μ^k) , gerada pelo algoritmo de ponto proximal 2.3, aplicado ao problema dual com distância de Bregman dada pela conjugada de M^2B , é uma solução ótima do problema (D). ■

3.2 A Penalidade M^2B Com Quadrática Ajustada

Em cada iteração do algoritmo de lagrangeano aumentado 2.2, aplicado ao problema (P) com penalidade M^2B , é resolvido o seguinte subproblema

$$x^{k+1} \in \operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ f_0(x) - r^k \sum_{i=1}^m \log \left(1 - \frac{\mu_i^k f_i(x)}{r^k} \right) \right\}, \quad (3.7)$$

o qual chamamos de subproblema interno.

Quando o algoritmo é implementado, usa-se métodos de programação irrestrita para resolver tais subproblemas. Em geral, para inicializar o programa que resolve o subproblema (3.7) é utilizado o ponto x^k gerado pela iteração $k - 1$. Nestas condições, se $\frac{\mu_i^k f_i(x^k)}{r^k} > 1$, para algum $i = 1, 2, \dots, m$, a penalidade M^2B tem valor $+\infty$ e x^k não pode ser usado como ponto inicial.

Modificaremos a penalidade M^2B , ajustando uma quadrática à função barreira logarítmica modificada, dada na relação (3.1), de maneira a garantir a inicialização do algoritmo que resolve os subproblemas internos.

Considere $t \in (0, 1)$ e θ a função barreira modificada definida em (3.1). Definimos a função $\hat{\theta}$ da seguinte forma

$$y \in \mathbb{R} \mapsto \hat{\theta}(y) = \begin{cases} \theta(y) & , \text{ se } -\infty < y \leq t \\ q(y) & , \text{ se } y > t \end{cases} \quad (3.8)$$

onde $q(y)$ é uma função quadrática ajustada a θ , de tal maneira que as propriedades satisfeitas por θ , (a)-(e) da seção 2.6, sejam satisfeitas também por $\hat{\theta}$.

Assim, dado $t \in (0, 1)$ ajustaremos $q(\cdot)$ à função θ tal que, $\hat{\theta}(t) = \theta(t)$, $\hat{\theta}'(t) = \theta'(t)$ e $\hat{\theta}''(t) = \theta''(t)$, o que garante que a nova função é de classe C^2 .

Como exemplo, fazendo $t = \frac{1}{2}$ e tomando $y > t$, tem-se $q(y) = 2y^2 + \log 2 - \frac{1}{2}$. Para este valor de t , o gráfico da função θ está plotado na figura 3.2 a seguir.

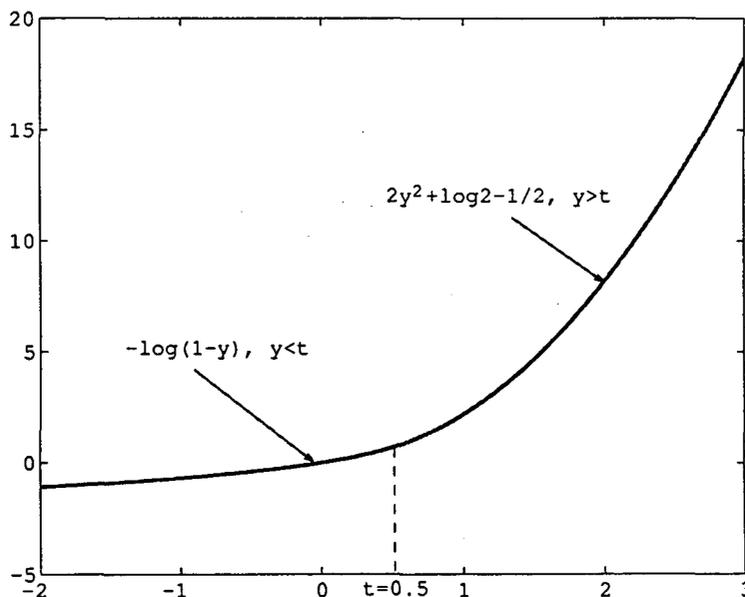


Figura 3.2: Quadrática ajustada à barreira modificada

A fórmula geral de q para um valor $t \in (0, 1)$ dado tem a forma

$$q(y) = \frac{1}{2(1-t)^2}y^2 + \frac{(1-2t)}{(1-t)^2}y + \frac{(4t-3)}{2(1-t)^2} - \log(1-t) + \frac{3}{2}. \quad (3.9)$$

No entanto, se o algoritmo de lagrangeano aumentado 2.2 é aplicado ao problema (P) com a penalidade tipo 1 e $\hat{\theta}$ dada na relação (3.8), ou seja,

$$y \in \mathbb{R}, \mu \in \mathbb{R}_{++} \mapsto p(y, \mu) = \hat{\theta}(\mu y) \quad (3.10)$$

a conjugada de p não será uma distância de Bregman e o teorema 3.2 não pode ser aplicado para este caso.

Mostraremos no próximo lema que, se a seqüência (r^k) no algoritmo de lagrangeano aumentado é limitada inferiormente por uma constante positiva, então a partir de uma iteração \bar{k} deste algoritmo a função $\hat{\theta}(\cdot)$ coincide com a função $\theta(\cdot)$, para todos os pontos em que ela é calculada a partir da iteração \bar{k} .

Lema 3.4 *Considere (x^k) e (μ^k) seqüências geradas pelo algoritmo de lagrangeano aumentado 2.2, aplicado ao problema (P), com penalidade p dada na relação (3.10), $\hat{\theta}$ dada na relação (3.8), (r^k) uma seqüência em (r_1, r_2) satisfazendo $0 < r_1 < r_2 < +\infty$ e $t \in (0, 1)$. Então, existe $\bar{k} \in \mathbb{N}$, tal que, para todo $k \geq \bar{k}$*

$$\frac{\mu_i^k y_i^{k+1}}{r^k} < t, \quad \forall i = 1, 2, \dots, m$$

onde $y^{k+1} = f(x^{k+1})$.

Prova. Como a função $\hat{\theta}$ satisfaz as propriedades (a) – (e), dadas na seção 2.6, as hipóteses do teorema 2.20 estão satisfeitas. Portanto, basta fazer $\varepsilon = t$ no teorema 2.20. ■

Para fechar os resultados sobre a penalidade M^2B , com quadrática ajustada, falta mostrar algum resultado semelhante ao corolário 3.3, que garante que todo ponto de acumulação da seqüência de multiplicadores gerada pelo algoritmo de lagrangeano aumentado é uma solução ótima do problema (D). Isto será feito no próximo teorema.

Teorema 3.5 *Considere (x^k) e (μ^k) seqüências geradas pelo algoritmo de lagrangeano aumentado 2.2, aplicado ao problema (P), com penalidade p dada em (3.10), $\hat{\theta}$ dada na relação (3.8) e (r^k) uma seqüência limitada em (r_1, r_2) , satisfazendo $0 < r_1 < r_2 < +\infty$. Nestas condições, todo ponto de acumulação da seqüência (μ^k) é uma solução ótima do problema (D).*

Prova. Considere \bar{k} definido no lema 3.4. Na iteração k do algoritmo de lagrangeano aumentado, aplicado ao problema (P) com penalidade dada na relação (3.10) e $\hat{\theta}$ dada na relação (3.8), tem-se

$$x^{k+1} \in \operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ f_0(x) + r^k \sum_{i=1}^m \hat{\theta} \left(\frac{f_i(x) \mu_i^k}{r^k} \right) \right\} \quad (3.11)$$

$$\mu_i^{k+1} = \mu_i^k \hat{\theta}' \left(\frac{y_i^{k+1} \mu_i^k}{r^k} \right), \quad \forall i = 1, 2, \dots, m \quad (3.12)$$

onde $y^{k+1} = f(x^{k+1})$.

Queremos mostrar que, para todo $k \geq \bar{k}$,

$$x^{k+1} \in \operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ f_0(x) + r^k \sum_{i=1}^m \theta \left(\frac{f_i(x) \mu_i^k}{r^k} \right) \right\} \quad (3.13)$$

$$\mu_i^{k+1} = \mu_i^k \theta' \left(\frac{y_i^{k+1} \mu_i^k}{r^k} \right), \quad \forall i = 1, 2, \dots, m \quad (3.14)$$

onde $y^{k+1} = f(x^{k+1})$.

A condição de otimalidade para o problema 3.11 é da forma

$$0 \in \left(\partial f_0(x^{k+1}) + \sum_{i=1}^m \hat{\theta}' \left(\frac{y_i^{k+1} \mu_i^k}{r^k} \right) \mu_i^k \nabla f_i(x^{k+1}) \right). \quad (3.15)$$

Do lema 3.4, sabemos que $\frac{y_i^{k+1} \mu_i^k}{r^k} < t$, para todo $i = 1, 2, \dots, m$, e $k \geq \bar{k}$. Segue da definição de $\hat{\theta}$, dada na relação 3.8, que

$$\hat{\theta}' \left(\frac{y_i^{k+1} \mu_i^k}{r^k} \right) = \theta' \left(\frac{y_i^{k+1} \mu_i^k}{r^k} \right), \quad \forall i = 1, 2, \dots, m. \quad (3.16)$$

Logo, das relações (3.15) e (3.16), tem-se a condição de otimalidade para o problema (3.13), isto é,

$$0 \in \left(\partial f_0(x^{k+1}) + \sum_{i=1}^m \theta' \left(\frac{y_i^{k+1} \mu_i^k}{r^k} \right) \mu_i^k \nabla f_i(x^{k+1}) \right). \quad (3.17)$$

Na verdade, o que mostramos neste último teorema é que, depois de algumas iterações do algoritmo 2.2, a parte da função $\hat{\theta}$ que é quadrática não influencia o

resultado das minimizações, pois os pontos gerados por este algoritmo, a partir desta iteração, são determinados por $y \in \mathbb{R} \mapsto \theta(y) = -\log(1 - y)$.

3.3 Potencialidades das Penalidades Tipo 1

Nas seções precedentes mostramos resultados de convergência da seqüência dual gerada pelo algoritmo de lagrangeano aumentado com penalidade tipo 1

$$y \in \mathbb{R}, \mu \in \mathbb{R}_{++} \mapsto p(y, \mu) = \theta(\mu y) \in \mathbb{R} \cup +\{\infty\}, \quad (3.18)$$

onde θ é uma função real satisfazendo as propriedades (a) – (e) dadas na definição 2.15.

Nesta seção, mostraremos e destacaremos qualidades da penalidade (3.18). Faremos sempre comparações desta com a penalidade mais utilizada na literatura que é a do tipo 2

$$y \in \mathbb{R}, \mu \in \mathbb{R}_{++} \mapsto p(y, \mu) = \mu\theta(y) \in \mathbb{R} \cup +\{\infty\}, \quad (3.19)$$

onde θ é uma função real satisfazendo as propriedades (a) – (e) dadas na definição 2.15.

As conjugadas das penalidades (3.18) e (3.19) são, respectivamente, dadas por

$$s \in \mathbb{R}_{++}, \mu \in \mathbb{R}_{++} \mapsto p^*(s, \mu) = \theta^*\left(\frac{s}{\mu}\right) \quad (3.20)$$

e

$$s \in \mathbb{R}_{++}, \mu \in \mathbb{R}_{++} \mapsto p^*(s, \mu) = \mu\theta^*\left(\frac{s}{\mu}\right). \quad (3.21)$$

As quase-distâncias associadas a estas penalidades são, respectivamente, iguais

a

$$s \in \mathbb{R}_{++}^m, \mu \in \mathbb{R}_{++}^m \mapsto D(s, \mu) = \sum_{i=1}^m \theta^*\left(\frac{s_i}{\mu_i}\right) \quad (3.22)$$

e

$$s \in \mathbb{R}_{++}^m, \mu \in \mathbb{R}_{++}^m \mapsto D(s, \mu) = \sum_{i=1}^m \mu_i \theta^*\left(\frac{s_i}{\mu_i}\right). \quad (3.23)$$

Levando em consideração que alguns métodos utilizam um modelo quadrático da função original, como é o caso de região de confiança, olharemos o que acontece com a aproximação quadrática das quase-distâncias (3.22) e (3.23).

Dado $\mu \in \mathbb{R}_{++}^m$, considere a aproximação quadrática de uma quase-distância

D, próximo ao ponto \bar{s} , ou seja,

$$s \in \mathbb{R}_{++}^m \mapsto D(s, \mu) \sim D(\bar{s}, \mu) + \nabla D_s(\bar{s}, \mu)^T (s - \bar{s}) + \frac{1}{2} (s - \bar{s})^T \nabla^2 D_s(\bar{s}, \mu) (s - \bar{s}). \quad (3.24)$$

No caso de (3.22) e (3.23) tem-se, respectivamente,

$$D(s, \mu) \sim \sum_{i=1}^m \theta^* \left(\frac{\bar{s}_i}{\mu_i} \right) + \frac{1}{\mu_i} \theta^{*'} \left(\frac{\bar{s}_i}{\mu_i} \right) (s_i - \bar{s}_i) + \frac{1}{2\mu_i^2} \theta^{*''} \left(\frac{\bar{s}_i}{\mu_i} \right) (s_i - \bar{s}_i)^2 \quad (3.25)$$

e

$$D(s, \mu) \sim \sum_{i=1}^m \mu_i \theta^* \left(\frac{\bar{s}_i}{\mu_i} \right) + \theta^{*'} \left(\frac{\bar{s}_i}{\mu_i} \right) (s_i - \bar{s}_i) + \frac{1}{2\mu_i} \theta^{*''} \left(\frac{\bar{s}_i}{\mu_i} \right) (s_i - \bar{s}_i)^2. \quad (3.26)$$

Se a aproximação quadrática for feita em $\bar{s} = \mu$, (3.25) e (3.26) ficam, respectivamente, iguais a

$$\sum_{i=1}^m \frac{1}{2} \left(\frac{s_i}{\bar{s}_i} - 1 \right)^2 \quad (3.27)$$

e

$$\sum_{i=1}^m \frac{\bar{s}_i}{2} \left(\frac{s_i}{\bar{s}_i} - 1 \right)^2. \quad (3.28)$$

A figura 3.3 a seguir mostra elipsóides gerados por (3.27) e (3.28). O elipsóide externo é referente à função (3.27) e o interno referente à (3.28). Veja que os elipsóides gerados por (3.27) são os elipsóides de Dikin.

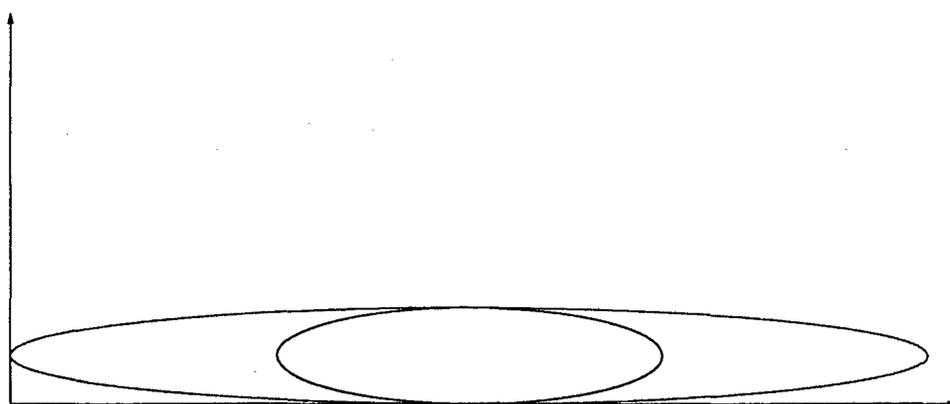


Figura 3.3: Curvas de nível das aproximações quadráticas de quase-distâncias

Capítulo 4

Programação Linear e Equivalência Entre Lagrangeano Aumentado e Afim Escala no Caso Linear

Acreditamos que se um algoritmo apresentar boas propriedades no caso linear terá alguma chance de ter boas propriedades no caso não linear. Agora, se um algoritmo não possui um bom desempenho para problemas de programação linear, também não deverá possuir para problemas de programação não linear. Mostramos que os algoritmos de lagrangeano aumentado, com a metodologia que propomos, possui ótimas propriedades de convergência, em relação à seqüência dual.

Começamos por definir o problema linear, o seu respectivo dual e a função de penalidade que utilizamos neste capítulo, a qual não pertence à família \mathcal{P} , pois a função θ utilizada na sua construção é quadrática e não satisfaz as propriedades (a) – (e) dadas na definição 2.15. Para simplificar nossa explicação vamos chamar tais penalidades de quadráticas.

Neste caso, em que as penalidades são quadráticas, os multiplicadores gerados pelo algoritmo de lagrangeano aumentado podem ser negativos, pois a derivada da função θ não é crescente em toda a reta, como no caso coercivo.

Para contornar o problema de multiplicadores negativos, apresentamos dois novos algoritmos. Um para o método de ponto proximal e o outro para o método de lagrangeano aumentado. Além desses, apresentamos o algoritmo afim escala, este bastante conhecido na literatura referente à programação linear.

No resultado principal do capítulo, mostramos que as direções duais geradas pelo algoritmo de lagrangeano aumentado aplicado ao problema linear e penalidade

tipo 1 quadrática são equivalentes às geradas pelo algoritmo afim escala aplicado ao dual do problema linear. Esse resultado também é muito bom, porque controlando adequadamente o parâmetro de penalidade no algoritmo de lagrangeano aumentado os multiplicadores gerados por este são os mesmos que os gerados pelo algoritmo afim escala. Como antes, este resultado não se aplica para as penalidades tipo 2.

Finalizamos o capítulo mostrando um resultado referente à geometria das curvas de nível das conjugadas das penalidades tipo 1 e 2 neste caso quadrático. Este resultado será posteriormente utilizado quando tratarmos da implementação dos algoritmos.

O método afim escala usado nesta seção é baseado em [11, 12].

4.1 O Problema Linear e Seu Dual

Iniciaremos definindo o problema de programação linear, em formato dual.

Nota 4.1 *Convencionou-se chamar de formato primal ao formato em que as restrições são de igualdade [11].*

O formato dual para o problema primal foi escolhido com o objetivo de manter a notação que vimos fazendo até aqui, a mesma do caso não linear. Assim, definimos o problema

$$(PL) \quad \begin{array}{ll} \text{minimizar} & -b^t x \\ \text{sujeito a} & Ax \leq c \end{array}$$

onde $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^m$ e $m > n$.

O sinal negativo na função objetivo é simplesmente uma convenção, para facilitar o desenvolvimento que faremos, lembrando que, se x^* resolve o problema de maximizar $f(x)$, ele também resolverá minimizar $-f(x)$.

Proposição 4.2 *A função dual lagrangeana associada ao problema (PL) tem a forma*

$$\mu \in \mathbb{R}_+^m \mapsto d(\mu) = \begin{cases} -c^t \mu & , \text{ se } A^t \mu = b \\ -\infty & , \text{ caso contrário.} \end{cases} \quad (4.1)$$

Prova. A função dual lagrangeana foi definida anteriormente na relação (2.34). No caso do problema linear ela tem a forma

$$\mu \in \mathbb{R}_+^m \mapsto d(\mu) = \inf\{-b^t x + \mu^t(Ax - c) : x \in \mathbb{R}^n\} = \inf\{(-b + A^t \mu)^t x - c^t \mu : x \in \mathbb{R}^n\},$$

como $(-b + A^t\mu)^t x - c^t\mu$ é linear em x , se $-b + A^t\mu \neq 0$ o ínfimo é igual $-\infty$. No caso contrário, o ínfimo é igual a $-c^t\mu$. ■

A seguir definiremos o problema dual associado ao problema (PL), o qual tem a forma (ver nota (4.1))

$$(DL) \quad \begin{array}{ll} \text{minimizar} & c^t\mu \\ \text{sujeito a} & A^t\mu = b \\ & \mu \geq 0 \end{array}$$

4.2 A Função Quadrática

A função θ a ser usada no algoritmo de lagrangeano aumentado, aplicado ao problema linear (PL), é definida como

$$y \in \mathbb{R} \mapsto \theta(y) = \frac{1}{2}y^2 + y. \quad (4.2)$$

Note que com esta θ as funções de penalidades construídas a partir desta não serão da família \mathcal{P} . Por exemplo, a propriedade de limitação por baixo da derivada, dada no item d da definição 2.15, não é satisfeita aqui, pois $\lim_{y \rightarrow -\infty} \theta'(y) = -\infty$.

A conjugada de θ dada na relação (4.2) é da forma

$$s \in \mathbb{R} \mapsto \theta^*(s) = \frac{1}{2}(s - 1)^2. \quad (4.3)$$

Se tentarmos imitar o caso não linear, como foi feito anteriormente, aplicando o algoritmo de ponto proximal ao problema dual, (DL), com quase-distância dada pela conjugada da penalidade p , tipo 1 ou 2 e θ^* dada na relação (4.3), os multiplicadores gerados podem ser negativos, o mesmo ocorrendo no primal quando o algoritmo de lagrangeano aumentado é aplicado ao problema (PL) e θ , quadrática dada na relação (4.2).

Para contornar este problema de multiplicadores negativos, recorreremos aos resultados mostrados na seção 2.3, particularmente ao lema 2.9 que fornece uma relação entre os métodos de ponto proximal e de região de confiança.

4.3 Ponto Proximal Para o Caso Linear Com Quase-Distância Quadrática

Apresentaremos nesta seção o algoritmo de ponto proximal aplicado ao problema (DL), utilizando uma quase-distância quadrática. Lembrando que, no primal, a penalidade quadrática é formada por uma penalidade p , tipo 1 ou 2, e função θ quadrática, dada na relação (4.2). No dual a quase-distância é formada pela conjugada da penalidade p e pela conjugada da função θ dada na relação (4.3).

Considere $S = \mathbb{R}_{++}^m$, $\hat{S} = \mathbb{R}_+^m$, p uma penalidade tipo 1 ou 2, θ a função quadrática (4.2) e D uma quase-distância quadrática dada por

$$\mu \in \hat{S}, \mu^k \in S \mapsto D(\mu, \mu^k) = \sum_{i=1}^m p^*(\mu_i, \mu_i^k) \quad (4.4)$$

onde p^* é a conjugada de p .

Para o caso quadrático, necessitaremos de uma nova hipótese, isto é

Hipótese (H5): O ponto inicial μ^0 é dual viável, isto é, $A^t \mu^0 = b$, $\mu^0 \in S$.

Após a apresentação do algoritmo explicaremos porque ela é necessária.

Nestas condições apresentaremos o algoritmo de ponto proximal, aplicado ao problema (DL) e com uma quase-distância quadrática dada na relação (4.4).

Algoritmo 4.1

Dados μ^0 dual viável, $r^0 > 0$ e $0 < \gamma < 1$

$k = 0$, $r = r^0$

Repetir

Encontre

$$\mu^+ = \operatorname{argmin}_{\mu \geq 0} \{-d(\mu) + rD(\mu, \mu^k)\}$$

Calcule

$$\lambda = \min_i \left(\frac{\mu_i^+}{\mu_i^k} \right)$$

Se $\lambda < 0$

$$r = 2r$$

Senão

$$\mu^{k+1} = \mu^+$$

$$\begin{aligned} \text{Se } \lambda > \gamma \\ r &= r/4 \\ k &= k + 1 \end{aligned}$$

Como D é uma quase-distância quadrática, μ^+ gerado na solução do subproblema interno, $\operatorname{argmin}_{\mu \geq 0} \{-d(\mu) + rD(\mu, \mu^k)\}$, pode ser negativo (veja figura 4.1 a seguir).

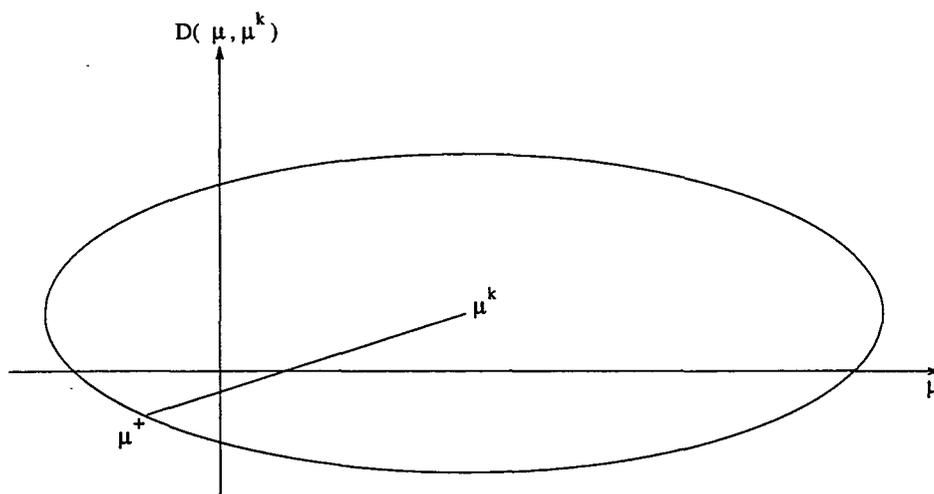


Figura 4.1: Passo do método de ponto proximal

Para entender o mecanismo do algoritmo recorreremos ao que foi feito na seção 2.3, onde mostramos no fato 2.8 que o subproblema

$$\operatorname{argmin}_{\mu \geq 0} \{-d(\mu) + rD(\mu, \mu^k)\}, \quad (4.5)$$

exibe uma certa relação de equivalência com o subproblema de região de confiança

$$\operatorname{argmin}_{\mu \geq 0} \{-d(\mu) : D(\mu, \mu^k) \leq \Delta_k\}. \quad (4.6)$$

Assim, mostramos no fato 2.8 que dados $\mu^k > 0$, $r > 0$ e $\bar{\mu}$ uma solução do problema (4.5), existe $\Delta_k > 0$ tal que $\bar{\mu}$ também é uma solução do problema (4.6). Além disso, mostramos no lema 2.9 que aumentando-se (diminuindo-se) r no problema (4.5) diminui-se (aumenta-se) a região de confiança $\{\mu \in \mathbb{R}_+^n : D(\mu, \mu^k) \leq \Delta_k\}$.

Desta forma, se alguma componente de μ^+ no algoritmo de ponto proximal 4.1 é negativa, como mostra a figura 4.1, aumenta-se o valor de r e diminui-se a região

de confiança, ou seja, μ é forçado a ficar no ortante positivo, conforme a figura 4.2 a seguir.

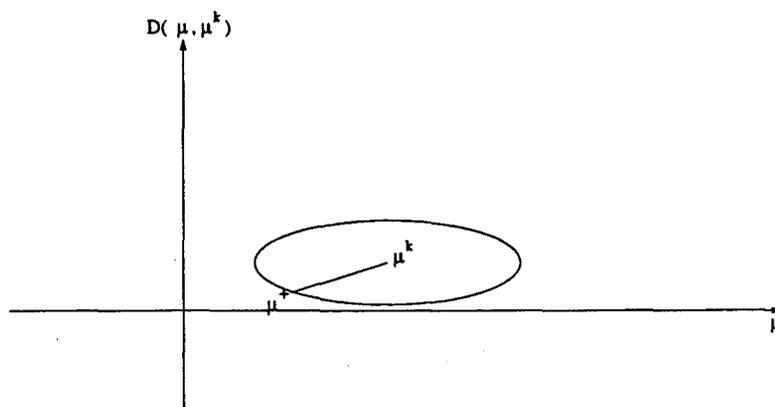


Figura 4.2: Passo do método de ponto proximal

Observe que, no caso em que μ^+ é positivo e a região de confiança é muito pequena, o valor de r é diminuído, para que os passos do algoritmo não sejam muito curtos.

Explicaremos agora porque a hipótese (H5) é necessária. A função dual (4.2), que define o problema (DL), tem valor finito se $A^t\mu = b$, onde A e b são dados (ver definição do problema (DL) na seção 4.1) e valor $-\infty$ no caso contrário. Logo, se o ponto inicial μ^0 não é dual viável, é possível que as regiões de confiança $\{\mu \in \mathbb{R}_+^n : D(\mu, \mu^k) \leq \Delta_k\}$ não contenham solução viável para Δ pequeno. Esta situação está ilustrada na figura 4.3 a seguir. Veja na figura 4.3 que $A^t\mu = b$, mas μ^+ é não positivo para nenhum valor de r^k .

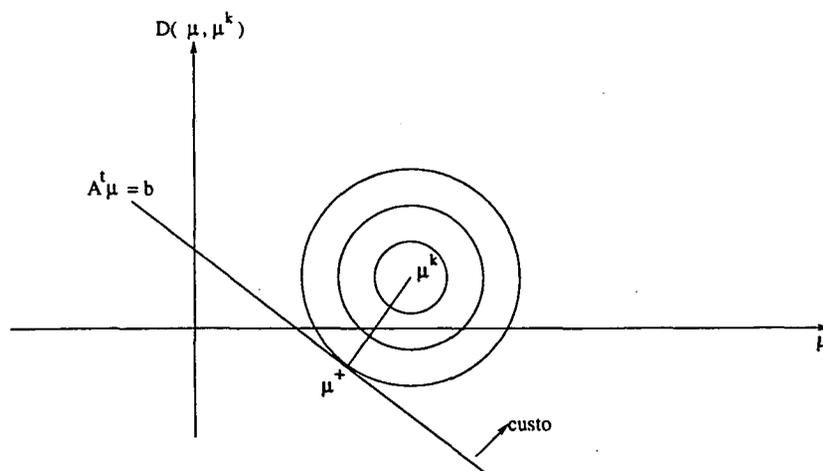


Figura 4.3: μ dual não viável

4.4 Lagrangeano Aumentado Para o Problema Linear e Penalidade Quadrática

Apresentaremos o algoritmo de lagrangeano aumentado, aplicado ao problema linear, (PL), com penalidade p tipo 1 ou 2 dadas, respectivamente, nas relações (2.23) e (2.28) e θ a função quadrática dada na relação (4.2). A hipótese (H5) que considera μ inicial dual viável também é considerada neste caso, pois a penalidade é quadrática.

No algoritmo a seguir, A_i denota a linha i da matriz A .

Algoritmo 4.2

Dados μ^0 dual viável, $r^0 > 0$ e $0 < \gamma < 1$ (tipicamente, $\gamma = 0.7$).

$k = 0$, $r = r^0$

Repetir

$$\mu = \mu^k$$

Encontrar

$$x^+ = \operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ -b^t x + \sum_{i=1}^m r p \left(\frac{A_i x - b_i}{r}, \mu_i \right) \right\}$$

Calcular

$$\mu_i^+ = p' \left(\frac{A_i x^+ - b_i}{r}, \mu_i \right), \quad i = 1, 2, \dots, m$$

$$\lambda = \min_i \left(\frac{\mu_i^+}{\mu_i^k} \right)$$

Se $\lambda < 0$

$$r = 2r$$

Senão

$$x^{k+1} = x^+, \quad \mu^{k+1} = \mu^+$$

Se $\lambda > \gamma$

$$r = r/4$$

$$k = k + 1$$

O mecanismo desse algoritmo é muito parecido com o mecanismo do algoritmo de ponto proximal 4.1, por isso a explicação daquele serve para este. O último passo, após obter $\lambda > 0$, reduz r para a iteração seguinte sempre que o passo foi muito curto, isto é, $\mu_i^{k+1} \geq \gamma \mu_i^k$, para $i = 1, \dots, m$.

4.5 Algoritmo Afim Escala

O algoritmo afim escala que apresentaremos é baseado em [11, 12] e será aplicado ao problema (DL), que conforme nota 4.1 está em formato primal e por isso será chamado de algoritmo afim escala primal. Nosso objetivo é mostrar que as direções duais geradas pelo algoritmo de lagrangeano aumentado com penalidade tipo 1 (2.23) e θ a função quadrática (4.2) são colineares às direções geradas pelo algoritmo afim escala aplicado ao problema dual.

Algoritmo 4.3

Dados $\mu^0 > 0$, $0 < \alpha_1 < \alpha_2 < 1$, $\Delta > 0$ e $\gamma^0 > 0$.

$k = 0$

Repetir

Encontrar

$$\mu^+ = \operatorname{argmin}_{\mu \in \mathbb{R}_+^n} \left\{ c^t \mu : \left\| \frac{\mu}{\mu^k} - e \right\| \leq \Delta, A^t \mu = b \right\}$$

onde $e = (1, 1, \dots, 1)^t \in \mathbb{R}^m$

Calcular

$$h = \mu^+ - \mu^k$$

$$\bar{\lambda} = \max \{ \lambda \in \mathbb{R} : \mu^k + \lambda h \geq 0 \}$$

$$\mu^{k+1} = \mu^k + \gamma^k \bar{\lambda} h \quad (\gamma^k \text{ escolhido em } [\alpha_1, \alpha_2])$$

$k = k + 1$

O maior trabalho do algoritmo (4.3) é o cálculo do novo ponto μ^+ , que consiste em minimizar, em um hiperplano, uma função linear num elipsóide. Este problema é considerado como um dos problemas fáceis em [11] (seções 3.3.3 e 3.3.5 pag. 32), onde o autor fornece uma fórmula explícita para o cálculo deste. Os próximos passos consistem na busca linear e atualização dos parâmetros.

4.6 Relação Entre Afim Escala e Lagrangeano Aumentado, no Caso Linear

Provaremos, nesta seção, o resultado mais importante do capítulo, o que relaciona o método afim escala e lagrangeano aumentado, no caso em que a penalidade é do tipo 1 (2.23) e θ a função quadrática (4.2).

Teorema 4.3 *O algoritmo de lagrangeano aumentado 4.2, aplicado ao problema (PL) com penalidade p do tipo 1, dada na relação (2.23), e θ quadrática dada na relação (4.2), gera direções duais, colineares às geradas pelo algoritmo afim escala 4.3, aplicado ao problema (DL).*

Prova. Considere $\mu^k > 0$ e $r^k > 0$ e a k -ésima iteração do algoritmo de ponto proximal 2.3, aplicado ao problema (DL), com quase-distância

$$\mu \in \mathbb{R}_+^n \mapsto D(\mu, \mu^k) = \sum_{i=1}^m p^*(\mu_i, \mu_i^k),$$

onde p^* é a conjugada de p tipo 1, ou seja,

$$(s, \mu) \in \mathbb{R}_+ \times \mathbb{R}_{++} \mapsto p^*(s, \mu) = \theta^* \left(\frac{s}{\mu} \right),$$

e θ^* é dada por (4.3).

Assim, o ponto μ^{k+1} é determinado por

$$\mu^{k+1} = \operatorname{argmin}_{\mu \in \mathbb{R}_+^n} \left\{ -d(\mu) + \frac{r^k}{2} \sum_{i=1}^m \left(\frac{\mu_i}{\mu_i^k} - 1 \right)^2 \right\}, \quad (4.7)$$

onde d é a função dual lagrangeana dada na relação (4.1).

Reescrevendo a relação (4.7), em termos da norma euclidiana, fica

$$\mu^{k+1} = \operatorname{argmin}_{\mu \in \mathbb{R}_+^n} \left\{ -d(\mu) + \frac{r^k}{2} \left\| \frac{\mu}{\mu^k} - e \right\|^2 \right\}, \quad (4.8)$$

onde $e = (1, 1, \dots, 1)^t \in \mathbb{R}^m$.

Vimos no fato 2.8 que existe $\Delta > 0$ tal que o problema (4.8) é equivalente ao seguinte problema de região de confiança

$$\mu^{k+1} = \operatorname{argmin}_{\mu \in \mathbb{R}_+^n} \left\{ -d(\mu) : \left\| \frac{\mu}{\mu^k} - e \right\| \leq \Delta \right\}. \quad (4.9)$$

Usando a função d dada pela relação (4.1), no problema (4.9) tem-se

$$\mu^{k+1} = \operatorname{argmin}_{\mu \in \mathbb{R}_+^n} \left\{ c^t \mu : \left\| \frac{\mu}{\mu^k} - e \right\| \leq \Delta \text{ e } A^t \mu = b \right\}. \quad (4.10)$$

O problema (4.10) é exatamente o problema resolvido na k -ésima iteração do algoritmo afim escala 4.3, aplicado ao problema (DL). Logo, existe $\Delta > 0$ tal que μ^{k+1} determinado pelo algoritmo afim escala, coincide com μ^{k+1} determinado pelo algoritmo de ponto proximal, ambos aplicados ao problema (DL). Pelo teorema de equivalência 2.19, μ^{k+1} também é determinado pelo algoritmo de lagrangeano aumentado aplicado ao problema (PL) com penalidade p tipo 1 e θ quadrática (4.2). ■

Chamamos a atenção que este resultado só é verdadeiro para a penalidade tipo 1. Para a penalidade tipo 2, este resultado não é verdadeiro, pois sua conjugada é da forma

$$\mu \in \mathbb{R}_+^n, \mu^k \in \mathbb{R}_{++}^n \mapsto D(\mu, \mu^k) = \sum_{i=1}^m \mu_i^k \theta^* \left(\frac{\mu_i}{\mu_i^k} \right). \quad (4.11)$$

Se θ^* é a dada por (4.3), e repetirmos os passos da prova do teorema 4.3, em vez da relação (4.7) teremos

$$\mu^{k+1} = \operatorname{argmin}_{\mu \in \mathbb{R}_+^n} \left\{ -d(\mu) + \frac{r^k}{2} \sum_{i=1}^m \mu_i^k \left(\frac{\mu_i}{\mu_i^k} - 1 \right)^2 \right\}. \quad (4.12)$$

Observe na relação (4.12) o termo μ_i^k que aparece multiplicando o termo quadrático $\left(\frac{\mu_i}{\mu_i^k} - 1 \right)^2$ em cada termo do somatório, para todo $i = 1, 2, \dots, m$. O que impede de obter a equivalência com afim escala, com esta penalidade.

Em seguida olharemos a geometria de curvas das quase-distâncias dadas pela conjugadas das penalidades tipo 1 e 2, respectivamente, no caso em que θ é quadrática.

Para as conjugadas da penalidade tipo 1

$$\mu \in \mathbb{R}_+^2, \mu^k \in \mathbb{R}_{++}^2 \mapsto D(\mu, \mu^k) = \left(\frac{\mu_1}{\mu_1^k} - 1 \right)^2 + \left(\frac{\mu_2}{\mu_2^k} - 1 \right)^2. \quad (4.13)$$

Para as conjugadas da penalidade tipo 2

$$\mu \in \mathbb{R}_+^2, \mu^k \in \mathbb{R}_{++}^2 \mapsto D(\mu, \mu^k) = \mu_1^k \left(\frac{\mu_1}{\mu_1^k} - 1 \right)^2 + \mu_2^k \left(\frac{\mu_2}{\mu_2^k} - 1 \right)^2. \quad (4.14)$$

A figura 4.4 mostra curvas de nível da função D para os casos em que D é dada, respectivamente, por (4.13) e (4.14). O elipsóide externo é referente à função (4.13) e o interno à função (4.14).

Imagine as direções duais d_1 e d_2 geradas pelo algoritmo de lagrangeano aumentado, quando aplicado, respectivamente, com penalidades tipo 1 e 2. Veja que

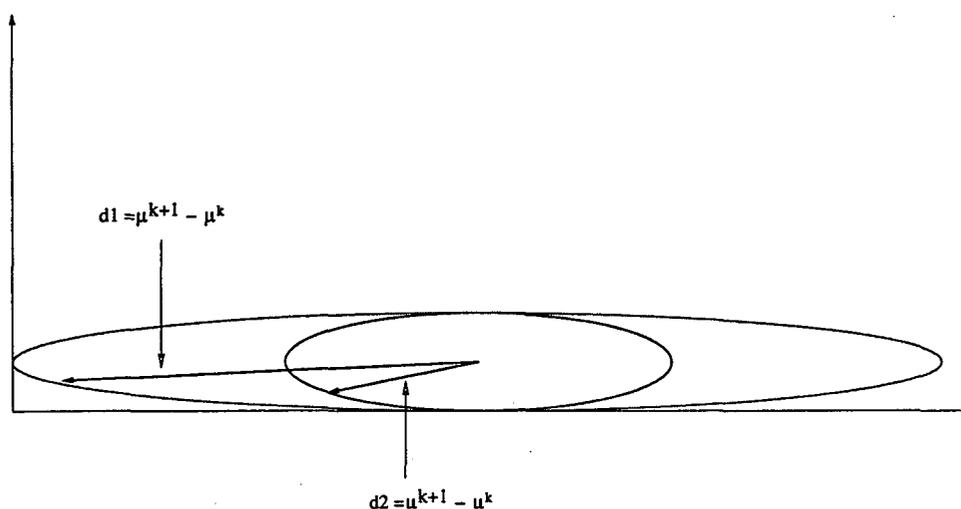


Figura 4.4: Direções duais geradas pelo alg. de lagrangeano aumentado

estamos usando a equivalência entre lagrangeano aumentado e ponto proximal. A representação das direções são duais, uma vez que desenhamos curvas de nível das funções conjugadas.

Uma das grandes vantagens de gerar direções equivalentes às direções geradas pelo algoritmo afim escala é o comprimento do passo dado pelo algoritmo, que é bem mais longo. Veja que o formato do elipsóide gerado pela função (4.13) é bem mais alongado do que o outro gerado pela função (4.14). No próximo teorema destacaremos este fato.

Teorema 4.4 *Considere $\mu \in \mathbb{R}_+^m$ e $D : \mathbb{R}_+^m \times \mathbb{R}_+^m \rightarrow \mathbb{R}$ uma quase-distância quadrática, tais que seus conjuntos de nível, determinados pela constante c , são representados por*

$$N = \{\mu \in \mathbb{R}_+^m : D(\mu, \mu^k) = c\}.$$

Se a função D é dada por

$$\mu \in \mathbb{R}_+^m \mapsto D(\mu, \mu^k) = \sum_{i=1}^m \frac{1}{2} \left(\frac{\mu_i}{\mu_i^k} - 1 \right)^2 \quad (4.15)$$

então o maior valor que a constante c pode assumir tal que $N \subset \mathbb{R}_+^m$ é $c = \frac{1}{2}$. Por outro lado, se a função D é dada

$$\mu \in \mathbb{R}_+^m \mapsto D(\mu, \mu^k) = \sum_{i=1}^m \frac{\mu_i^k}{2} \left(\frac{\mu_i}{\mu_i^k} - 1 \right)^2 \quad (4.16)$$

então o maior valor que a constante c pode assumir tal que $N \subset \mathbb{R}_+^m$ é $c = \frac{1}{2} \min\{\mu^k\}$.

Prova. Dados $\mu \in \mathbb{R}_+^m$ e $D : \mathbb{R}_+^m \times \mathbb{R}_{++}^m \rightarrow \mathbb{R}$ quadrática, queremos determinar o maior valor de c tal que $N = \{\mu \in \mathbb{R}_+^m : D(\mu, \mu^k) = c\}$ esteja inteiramente contido em \mathbb{R}_+^m .

Se D é dada, respectivamente, por (4.15) e (4.16) o conjunto N representa elipsóides com eixos dados pelas coordenadas de μ^k que são paralelos aos eixos coordenados de \mathbb{R}^m .

Desta forma, o menor valor das coordenadas de μ^k ($\min\{\mu^k\}$) determina o valor da constante c onde o elipsóide tangencia um dos eixos coordenados do \mathbb{R}^m , conforme figura 4.5 a seguir.

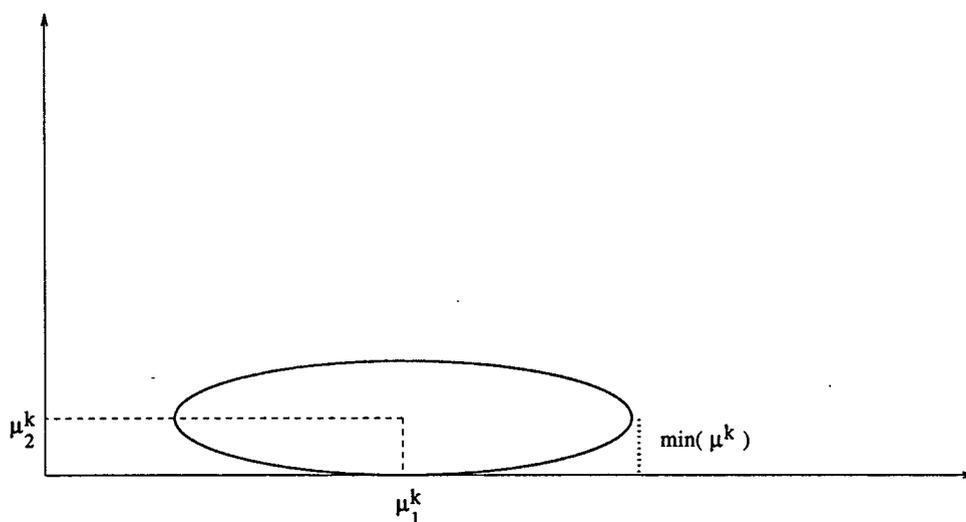


Figura 4.5: Elipsóide N para algum valor de c positivo

Assim, para determinar esse ponto de tangência, isto é, o valor da constante c , considere para algum $\ell \in \{1, 2, \dots, m\}$, $\mu_\ell^k = \min(\mu^k)$ e escolha $\bar{\mu} = (\mu_1^k, \dots, 0, \dots, \mu_m^k)$. Isto quer dizer que escolhemos $\bar{\mu}$ com a coordenada ℓ nula e as demais iguais às de μ^k .

Logo, para D dada por (4.15), tem-se

$$c = \sum_{i=1, i \neq \ell}^m \frac{1}{2} \left(\frac{\mu_i}{\mu_i^k} - 1 \right)^2 + \frac{1}{2} \left(\frac{0}{\mu_\ell^k} - 1 \right)^2$$

e como $\bar{\mu} = \mu_i^k$, para $i \neq \ell$, tem-se $\left(\frac{\mu_i}{\mu_i^k} - 1 \right)^2 = 0$, para todo $i = 1, 2, \dots, \ell - 1, \ell + 1, \dots, m$. Segue que $c = \frac{1}{2}$.

Da mesma forma, se D é dada por (4.16), tem-se

$$c = \sum_{i=1, i \neq \ell}^m \frac{\mu_i^k}{2} \left(\frac{\mu_i}{\mu_i^k} - 1 \right)^2 + \frac{\mu_\ell^k}{2} \left(\frac{0}{\mu_\ell^k} - 1 \right)^2 = \frac{\mu_\ell^k}{2},$$

como $\mu_\ell^k = \min\{\mu^k\}$ segue que $c = \frac{1}{2} \min\{\mu^k\}$. ■

Pelo teorema anterior, o maior elipsóide inteiramente contido em \mathbb{R}_+^m , para as funções dadas em (4.15) e (4.16) são, respectivamente, da forma

$$\sum_{i=1}^m \frac{1}{2} \left(\frac{\mu_i}{\mu_i^k} - 1 \right)^2 = \frac{1}{2} \quad (4.17)$$

e

$$\sum_{i=1}^m \frac{\mu_i^k}{2} \left(\frac{\mu_i}{\mu_i^k} - 1 \right)^2 = \frac{1}{2} \min\{\mu^k\}. \quad (4.18)$$

Para que estes elipsóides fiquem no mesmo nível (chamamos de nível à constante c que determina a curva) é necessário multiplicar (4.17) por $\min\{\mu^k\}$. Usaremos esta informação na implementação do algoritmo de lagrangeano aumentado, que será o assunto para o próximo capítulo.

Capítulo 5

Ajuste do Parâmetro de Penalidade

Neste capítulo, tratamos de questões referentes à implementação dos algoritmos de lagrangeano aumentado, sendo o foco principal a atualização dos parâmetros de penalização. Como o próprio nome diz, a função desse parâmetro é penalizar. Agora, se a penalização for muito rigorosa, os subproblemas gerados pelo algoritmo podem ficar muito difíceis de resolver. Por outro lado, se os pontos gerados estão muito inviáveis e a penalização é muito branda, os subproblemas gerados podem ser fáceis de resolver e o algoritmo pode gastar muitas iterações para resolver o problema. O ideal é ter uma boa fórmula para atualizar este parâmetro, o que, no entanto, não é fácil de conseguir.

Mostramos no principal resultado deste capítulo que os algoritmos de lagrangeano aumentado com penalidades do tipo 1 têm convergência sublinear (péssima) se este parâmetro de penalidade ficar limitado inferiormente por uma constante positiva.

Apresentamos três algoritmos. Um, para aumentar o parâmetro de penalidade r ; e os outros dois, são os algoritmos de lagrangeano aumentado com penalidade tipo 1 e 2. Ambos utilizam o anterior para aumentar r , quando for necessário.

Algoritmos tipo 2.

Os algoritmos tipo 2 são bastante estudados e dão origem a métodos de pontos proximais com φ -divergências no problema dual. Neste caso, o parâmetro de penalidade r^k pode ser mantido constante, com os seguintes resultados:

- Se r^k é grande, os problemas internos serão fáceis mas os passos serão curtos e a convergência lenta.
- Se r^k é pequeno, ocorre o oposto: os passos são longos mas os problemas internos são difíceis.

Com r^k constante, obtém-se normalmente convergência linear, com uma razão que depende de r^k . Para obter convergência superlinear é necessário fazer $r^k \rightarrow 0$, para o que é necessário um critério para reduzir r^k a cada iteração. Espera-se que um bom algoritmo reduza r^k até obter um valor com bom compromisso entre dificuldade do passo interno e velocidade de convergência.

Algoritmos tipo 1.

No fim do capítulo anterior obtivemos uma indicação de que o parâmetro r^k deve estar relacionado ao valor $\min\{\mu^k\}$, para obter regiões de confiança duais de bom tamanho. Assim, faremos por exemplo $r^k = \alpha^k \min\{\mu^k\}$, e controlaremos o valor de α^k . Este método será aperfeiçoado adiante.

Para motivar a necessidade de reduzir r^k em algoritmos tipo 1, mostraremos no lema a seguir que com r^k constante a velocidade de convergência será sublinear.

Lema 5.1 *Considere (μ^k) uma seqüência gerada pelo algoritmo de ponto proximal com quase-distância dada pela conjugada de p tipo 1, ou seja,*

$$(\mu, \mu^k) \in \mathbb{R}_+^m \times \mathbb{R}_{++}^m \mapsto D(\mu, \mu^k) = \sum_{i=1}^m \theta^* \left(\frac{\mu_i}{\mu_i^k} \right),$$

e $r^k > r > 0$. Supomos que $\mu^k \rightarrow \bar{\mu}$ quando $k \rightarrow \infty$ e que exista $\ell \in \{1, 2, \dots, m\}$, tal que $\bar{\mu}_\ell = 0$. A seqüência (μ^k) gerada nestas condições tem convergência sublinear.

Prova. Na k -ésima iteração do algoritmo de ponto proximal é resolvido o seguinte subproblema

$$\mu^{k+1} = \operatorname{argmin} \left\{ -d(\mu) + r^k \sum_{i=1}^m \theta^* \left(\frac{\mu_i}{\mu_i^k} \right) \right\} \quad (5.1)$$

onde d é a função dual lagrangeana, dada na relação (2.34).

De (5.1) tem-se

$$d(\mu^{k+1}) - d(\mu^k) \geq r^k \sum_{i=1}^m \theta^* \left(\frac{\mu_i}{\mu_i^k} \right). \quad (5.2)$$

Como $\sum_{i=1}^m \theta^* \left(\frac{\mu_i}{\mu_i^k} \right) \geq 0$ (é uma quase-distância) e $(-d(\mu^k))$ é uma seqüência decrescente e limitada, segue que

$$d(\mu^{k+1}) - d(\mu^k) \rightarrow 0.$$

Uma vez que $r^k > r > 0$ e $d(\mu^{k+1}) - d(\mu^k) \rightarrow 0$, a soma do lado direito de (5.2) também tende a zero, ou seja, $\sum_{i=1}^m \theta^* \left(\frac{\mu_i}{\mu^k} \right) \rightarrow 0$, quando $k \rightarrow +\infty$. Além disso, $\theta^* \left(\frac{\mu_i}{\mu^k} \right) \geq 0$ e então $\theta^* \left(\frac{\mu_i}{\mu^k} \right) \rightarrow 0$, ou das propriedades que definem θ , $\frac{\mu^{k+1}}{\mu^k} \rightarrow 1$.

Por hipótese $\mu^k \rightarrow \bar{\mu}$ e existe $\ell \in \{1, 2, \dots, m\}$ tal que $\bar{\mu}_\ell = 0$. Segue que $\lim_{k \rightarrow \infty} \frac{\mu_\ell^{k+1} - \bar{\mu}_\ell}{\mu_\ell^k - \bar{\mu}_\ell} = 1$. Portanto, a seqüência (μ^k) tem convergência sublinear. ■

No caso dos multiplicadores gerados pelo algoritmo de lagrangeano aumentado, se $\bar{\mu}$ ótimo não tiver nenhuma componente nula, isto é, a solução dual está no interior do ortante positivo, ou, pensando em termos de primal, todas as restrições estão ativas na solução ótima, em geral, os métodos não devem encontrar dificuldades, pois este é o caso mais simples, a menos que a solução primal seja degenerada. No entanto, o que ocorre com muita frequência é a solução ótima $\bar{\mu}$ ter componentes nulas e este é o caso em que os métodos podem encontrar dificuldades.

Concluimos que com penalidade do tipo 2, r^k deve diminuir com k (ou eventualmente aumentar na primeira iteração se r^0 está muito pequeno); com penalidade tipo 1, é necessário reduzir r^k com k sempre que $\min\{\mu^k\}$ tende a zero. Neste caso, se tomarmos por exemplo $r^k = \alpha^k \min\{\mu^k\}$, é bem possível que devamos reduzir ou aumentar α^k em cada iteração para gerar problemas internos de dificuldade moderada.

A seguir apresentaremos um algoritmo específico para modificar o parâmetro r^k no início de cada iteração do algoritmo 1.

Aumento de r^k .

Este procedimento constitui a novidade de nosso tratamento. O aumento de r^k segue o procedimento desenvolvido para o caso linear com penalidade quadrática do capítulo anterior, procurando garantir que o problema interno gerado na iteração não é muito difícil.

Aqui trataremos somente o caso convexo: tomando a aproximação quadrática do lagrangeano aumentado no ponto x^k , executamos seu minimizador por um passo de Newton. Calculamos os multiplicadores que seriam gerados nesse ponto a partir dessa aproximação quadrática e aumentamos r^k quando o resultado tiver componentes (muito) negativas.

No algoritmo a seguir, consideraremos o parâmetro de penalidade r como vetorial, isto é, $r \in \mathbb{R}^m$, o que será útil adiante.

Algoritmo 5.1 : *Aumento do parâmetro r*

Dados $j = 0$, x , $r > 0$, $\mu > 0$ e $\ell > 0$

Enquanto $j < \ell$ faça

Construa

$m_{f_0}(d) \approx f_0(x + d)$ modelo quadrático de $f_0(x)$ em torno de x

$m_p(d) \approx \sum_{i=1}^m r_i p\left(\frac{f_i(x + d)}{r_i}, \mu_i\right)$ modelo quadrático da penalidade

Calcule $\bar{d} = \operatorname{argmin}\{m_{f_0}(d) + m_p(d)\}$

$\mu_i^+ = \frac{\partial}{\partial d_i} m_p(\bar{d})$, $i = 1, \dots, m$

$\tau = \min_i \frac{\mu_i^+}{\mu_i}$

se $\tau < -\alpha$, faça $r = 2r$, $j = j + 1$

senão, pare.

Este algoritmo somente aumenta r , multiplicando-o por 2^j , no máximo 2^ℓ . Observe que, se $r^k = \alpha^k \min\{\mu^k\}$, deve-se fazer $\alpha^k = 2^j \alpha^k$. É fácil adicionar um dispositivo para reduzir r quando o passo de Newton for muito curto, ou seja, quando o modelo quadrático está muito bom. Decidimos não fazê-lo, usando o critério exposto abaixo para reduzir r^k .

Custo computacional.

Os cálculos executados por este procedimento coincidem com os necessários à primeira iteração de um algoritmo de região de confiança aplicado ao problema interno, e portanto podem ser aproveitados naquele algoritmo. Isto nos fornece também o procedimento a seguir no caso não convexo: basta calcular μ_i^+ a partir do resultado do primeiro passo de região de confiança obtido no algoritmo interno.

Redução de r .

Para reduzir r , usaremos como critério a complementaridade, definindo

$$c^k = \max_i \{|\mu_i^k f_i(x^k)|\}$$

ao final de cada iteração, e utilizaremos o seguinte procedimento: a partir de $k = 2$:

$$\text{se } \frac{c^k}{c^{k-1}} > \gamma, \text{ faça } r^k = r^k/4 \text{ (ou } \alpha^k = \alpha^k/4),$$

onde $\gamma \in (0, 1)$, por exemplo, $\gamma = 0.2$.

Assim, procura-se forçar o pior elemento de $\mu^k f(x^k)$ a melhorar ao menos 80% por iteração. Observe que pode ocorrer que r^k seja reduzido por este critério e em seguida aumentado pelo procedimento anterior, se o efeito de redução foi excessivo.

Uso de parâmetro vetorial.

Com base na análise do problema de programação linear, conclui-se que os algoritmos com penalidade de tipo 1 tendem a variar as componentes de μ de modo harmônico, enquanto que os de tipo 2 podem reduzir algumas componentes de μ muito rapidamente, o que pode causar dificuldades no algoritmo interno. Isto de fato se verifica nos testes computacionais.

Por outro lado, a inviabilidade comporta-se bem com penalidades tipo 2, mas algumas componentes de $f(\cdot)$ podem tender a zero rapidamente com penalidades tipo 1. Isto ocorre quando se usa $r^k = \alpha^k \min\{\mu^k\}$, e os termos de penalidade são

$$\alpha^k \min\{\mu^k\} \theta \left(\frac{\mu_i^k}{\alpha \min\{\mu^k\}} \frac{f_i(x)}{\alpha^k} \right).$$

Os termos $\mu_i^k / \min\{\mu^k\}$ podem tornar-se muito grandes, dando muita ênfase a $f_i(x)$ e provocando problemas computacionais.

Este problema é evitado se usarmos um parâmetro vetorial $\alpha^k r^k \in \mathbb{R}_{++}^m$ com $\alpha^k \in \mathbb{R}_+$ ajustado pelos métodos acima, e

$$r_i^k = \max\{\min\{\mu^k\}, 10^{-3} \mu_i^k\}.$$

Este foi o método usado na implementação, e evitou-se os problemas numéricos citados.

Observe que se utilizássemos $r_i^k = \mu_i^k$, o termo de penalização seria

$$\alpha^k \sum_{i=1}^m \mu_i^k \theta \left(\frac{f_i(x)}{\alpha^k} \right),$$

ou seja, coincidiria com a penalização de tipo 2. O método que propomos pode ser visto como intermediário entre os dois.

O algoritmo implementado.

Enunciaremos agora os dois algoritmos implementados, com penalidades do tipo 1 e 2. No próximo capítulo mostraremos comparações entre eles. Os algoritmos utilizam os procedimentos descritos acima para escolher os parâmetros de penalização.

1. Penalidade tipo 2

Algoritmo 5.2

Dados $\mu^0 > 0$, $r > 0$, $c^0 = +\infty$, $\gamma \in (0, 1)$ ($\gamma = 0.2$)

$k = 0$

Repetir

Obter \bar{r} pelo algoritmo 5.1

Encontrar

$$x^+ \in \operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ f_0(x) + \bar{r} \sum_{i=1}^m \mu_i^k \theta \left(\frac{f_i(x)}{\bar{r}} \right) \right\}$$

Calcular

$$\mu_i^{k+1} = p' \left(\frac{f_i(x^+)}{\bar{r}}, \mu_i \right), \quad i = 1, 2, \dots, m$$

Redução de r :

$$c^k = \max_i \{ |\mu_i^k f_i(x^k)| \}$$

Se $c^k > \gamma c^{k-1}$, faça $r^{k+1} = \bar{r}/4$,

senão, $r^{k+1} = \bar{r}$.

2. Penalidade tipo 1: aqui usaremos parâmetro vetorial

Algoritmo 5.3

Dados $\mu^0 > 0$, $\alpha^0 > 0$, $c^0 = +\infty$, $\gamma \in (0, 1)$

$k = 0$

Repetir

$$\mu_{\min} = \min\{\mu^k\}$$

$$\bar{r}_i = \max\{\mu_{\min}, 10^{-3}\mu_i^k\}, \quad i = 1, \dots, m$$

Calcular $\bar{\alpha} \geq \alpha^k$ pelo algoritmo 5.1 partindo de $r = \alpha^k \bar{r}$.

Encontrar

$$x^{k+1} \in \operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ f_0(x) + \bar{\alpha} \sum_{i=1}^m \bar{r}_i \theta \left(\frac{\mu_i^k f_i(x)}{\bar{\alpha} \bar{r}_i} \right) \right\}$$

Calcular

$$\mu_i^{k+1} = \mu_i^k \theta' \left(\frac{\mu_i^k f_i(x^{k+1})}{\bar{\alpha} \bar{r}_i} \right), \quad i = 1, 2, \dots, m$$

Redução de α :

$$c^k = \max_i \{|\mu_i^k f_i(x^k)|\}$$

Se $c^k > \gamma c^{k-1}$, faça $\alpha^{k+1} = \bar{\alpha}/4$,

senão, $\alpha^{k+1} = \bar{\alpha}$.

Capítulo 6

Testes Numéricos

Para testar o algoritmo de lagrangeano aumentado que estamos propondo, programamos em Matlab, versão 6.0 R12 para Linux, o método de região de confiança para resolver os subproblemas internos gerados em cada iteração do mesmo. Usamos o algoritmo hook descrito em [7] para minimizar a quadrática gerada em cada iteração de região de confiança,

Assim, implementamos o algoritmo 2.2 com penalidades tipo 1 e 2 dadas, respectivamente, por

$$y \in \mathbb{R}, \mu \in \mathbb{R}_{++}^* \mapsto p(y, \mu) = \theta(y\mu)$$

e

$$y \in \mathbb{R}, \mu \in \mathbb{R}_{++} \mapsto p(y, \mu) = \mu\theta(y)$$

que foram apresentados anteriormente na seção 2.6.

Utilizamos duas metodologias para a função θ que define as penalidades tipo 1 e 2 acima. A primeira é a função M^2B dada na definição 3.1, ou seja, é a função barreira logarítmica modificada com quadrática ajustada. Esta é a penalidade que deu origem ao nosso trabalho. A segunda metodologia é a função exponencial quadrática apresentada em [23]

$$y \in \mathbb{R} \mapsto \theta(y) = \begin{cases} e^y & \text{se } y \leq \beta \\ e^\beta + e^\beta(y - \beta) + \frac{e^\beta}{2}(y - \beta)^2 & \text{se } y > \beta \end{cases}$$

$\beta \geq 0$ é o parâmetro onde foi ajustada a quadrática.

Os problemas testados são da coleção CUTE: Constrained and Unconstrained Testing Environment [16], na versão *small* com interface para o Matlab. Por ser um

conjunto com milhares de problemas, escolhemos aqueles que se encaixavam em nossas condições, isto é, problemas com restrições de desigualdade e sem limite nas variáveis.

Testamos 82 (oitenta e dois) problemas que estão apresentados na tabela 6.1. A primeira coluna da tabela é formada pelos nomes dos problemas apresentados na coleção CUTE. As demais colunas são: número de variáveis, restrições, tipo das restrições, e tipo da função objetivo para cada problema.

Tabela 6.1: Problemas do CUTE

Problema	# variáveis	# restrições	Tipo das restrições	tipo da objetivo
CB2	3	3	3 não lineares	linear
CB3	3	3	3 não lineares	linear
CHACONN1	3	3	3 não lineares	linear
CHACONN2	3	3	3 não lineares	linear
COSHFUN	10	3	3 não lineares	linear
DEMYMALO	3	3	2 linear 1 não lineares	linear
DIPIGRI	7	4	4 não lineares	não linear
EXPFITA	5	22	22 lineares	não linear
EXPFITB	5	102	102 lineares	não linear
GIGOMEZ1	3	3	2 lineares 1 não linear	linear
HAIFAS	13	9	9 não lineares	linear
HALDMADS	6	42	42 não lineares	linear
HS10	2	1	não linear	linear
HS11	2	1	não linear	não linear
HS12	2	1	não linear	não linear
HS22	2	2	1 não linear 1 linear	não linear
HS29	3	1	não linear	não linear
HS43	4	3	3 não linear	não linear
HS88	2	1	1 não linear	não linear
HS89	3	1	1 não linear	não linear
HS90	4	1	1 não linear	não linear
HS91	5	1	1 não linear	não linear
HS92	6	1	1 não linear	não linear
HS100	7	4	4 não linear	não linear
HS100MOD	7	4	4 não linear	não linear
HS113	10	8	5 não lineares 3 lineares	não linear
HS268	5	5	5 lineares	não linear
KIWCRESC	3	2	2 não lineares	linear
LISWET1	103	100	100 lineares	não linear
LISWET2	103	100	100 lineares	não linear
LISWET3	103	100	100 lineares	não linear
LISWET4	103	100	100 lineares	não linear
LISWET5	103	100	100 lineares	não linear

continua ...

Tabela 6.1: Problemas do CUTE (continuação)

Problema	# variáveis	# restrições	Tipo das restrições	tipo da objetivo
LISWET6	103	100	100 lineares	não linear
LISWET10	103	100	100 lineares	não linear
MADSEN	3	6	6 não lineares	linear
MAKELA1	3	2	1 linear 1 não linear	linear
MAKELA2	3	3	3 não lineares	linear
MAKELA3	21	20	20 não lineares	linear
MAKELA4	21	40	40 lineares	linear
MIFFLIN1	3	2	1 linear 1 não linear	linear
MIFFLIN2	3	2	2 não lineares	linear
MINMAXBD	5	20	20 não lineares	linear
MINMAXRB	3	4	2 não lineares 2 lineares	linear
OET1	3	6	6 lineares	não linear
OET2	3	202	202 não lineares	linear
OET3	4	6	6 lineares	linear
OET3	4	202	202 lineares	linear
OET4	4	6	6 não lineares	linear
OET5	5	6	6 não lineares	linear
OET6	5	6	6 não lineares	linear
OET7	7	6	6 não lineares	linear
PENTAGON	6	15	15 lineares	não linear
POLAK1	3	2	2 não lineares	linear
POLAK2	11	2	2 não lineares	linear
POLAK3	12	10	10 não lineares	linear
POLAK4	3	3	3 não lineares	linear
POLAK5	3	2	2 não lineares	linear
POLAK6	5	4	4 não lineares	linear
POWELL20	10	10	10 lineares	não linear
PT	2	501	501 lineares	linear
PT	2	3	3 lineares	linear
PT	2	101	101 lineares	linear
ROSENMMX	5	4	4 não lineares	linear
SIAPOW1	2	20	20 lineares	linear
SIAPOW1	2	100	100 lineares	linear
SIAPOW1	2	500	500 lineares	linear
SIAPOW1M	2	20	20 lineares	linear
SIAPOW1M	2	100	100 lineares	linear
SIAPOW1M	2	500	500 lineares	linear
SIAPOW2	2	20	20 lineares	linear
SIAPOW2M	2	20	20 lineares	linear
SIAPOW2M	2	100	100 lineares	linear
SIAPOW2M	2	500	500 lineares	linear
TFI1	3	11	11 não lineares	não linear
TFI1	3	51	51 não lineares	não linear

continua ...

Tabela 6.1: Problemas do CUTE (continuação)

Problema	# variáveis	# restrições	Tipo das restrições	tipo da objetivo
TFI2	3	11	11 lineares	linear
TFI2	3	101	101 lineares	linear
TFI3	3	11	11 lineares	não linear
TFI3	3	51	11 lineares	não linear
TFI3	3	101	11 lineares	não linear
WOMFLET	13	9	9 não linear	linear

Dentre estes 82 problemas existem os não convexos, não lineares e os que são mistos, tendo função objetivo linear com restrições não lineares, ou o contrário, função objetivo não linear com restrições lineares.

Os resultados dos testes estão apresentados na tabela 6.2, cujos dados são especificados na primeira linha de cada coluna, ou seja:

Problema - Nome do problema definido pela coleção CUTE.

Penalidade - Tipo da penalidade e o tipo da função θ , mb é a barreira logarítmica e axp é a exponencial quadrática.

it.int - Número de iterações executadas pelo algoritmo de região de confiança.

it.ext - Número de iterações do lagrangeano aumentado, ou seja, número de vezes que resolvemos o subproblema formado pela função lagrangeano aumentado.

func - Número de avaliações de funções, incluindo todas, função objetivo e restrições.

grad - Número de avaliações dos gradientes, da função objetivo mais os das restrições.

la - Número de avaliações da função lagrangeano aumentado.

fótimo - Valor ótimo da função objetivo encontrado pelo algoritmo.

Tabela 6.2: Resultados Numéricos

Problema	Penalidade	it.int	it.ext	func	grad	la	fótimo	
CB2	mb	Tipo 1	34	10	226	186	142	1.9522
		Tipo 2	31	11	223	179	137	1.9522
	axp	Tipo 1	31	10	214	174	133	1.9522
		Tipo 2	28	9	193	157	120	1.9522

continua ...

Tabela 6.2: Resultados Numéricos (continuação)

Problema	Penalidade		it.int	it.ext	func	grad	la	fótimo
CB3	mb	Tipo 1	34	10	226	186	142	2.0000
		Tipo 2	33	10	222	182	139	2.0000
	axp	Tipo 1	28	10	202	162	124	2.0000
		Tipo 2	29	11	215	171	131	2.0000
CHACONN1	mb	Tipo 1	29	9	205	161	125	1.9522
		Tipo 2	21	7	155	111	89	1.9522
	axp	Tipo 1	26	9	193	141	112	1.9522
		Tipo 2	15	5	113	85	67	1.9522
CHACONN2	mb	Tipo 1	30	8	200	152	120	2.0000
		Tipo 2	29	7	187	143	113	2.0000
	axp	Tipo 1	27	8	188	144	113	2.0000
		Tipo 2	27	6	170	122	99	2.0000
COSHFUN	mb	Tipo 1	21	5	125	97	76	-0.6614
		Tipo 2	24	5	137	101	81	-0.6614
	axp	Tipo 1	18	4	104	84	65	-0.6614
		Tipo 2	19	4	108	88	68	-0.6614
DEMYMALO	mb	Tipo 1	23	2	110	78	65	-3.0000
		Tipo 2	23	2	110	78	65	-3.0000
	axp	Tipo 1	23	3	119	87	71	-3.0000
		Tipo 2	23	3	119	87	71	-3.0000
DIPIGRI	mb	Tipo 1	27	4	179	124	83	680.6301
		Tipo 2	28	8	228	178	112	680.6301
	axp	Tipo 1	23	4	159	129	81	680.6301
		Tipo 2	26	8	218	168	106	680.6301
EXPFITA	mb	Tipo 1	61	4	1499	970	149	1.3915
		Tipo 2	70	13	1922	1439	207	1.3879
	axp	Tipo 1	85	4	2120	1177	194	0.0011
		Tipo 2	68	8	1917	1250	191	0.0011
EXPFITB	mb	Tipo 1	98	4	10510	8553	268	10.8717
		Tipo 2	173	21	20209	13720	462	10.8628
	axp	Tipo 1	87	6	10100	6392	222	0.0050
		Tipo 2	66	10	8765	6190	205	0.0050
GIGOMEZ1	mb	Tipo 1	18	2	98	70	58	-3.0000
		Tipo 2	23	3	127	91	75	-3.0000
	axp	Tipo 1	19	4	120	80	67	-3.0000
		Tipo 2	35	3	175	107	95	-3.0000
HAIFAS	mb	Tipo 1	35	7	427	357	112	-0.4500
		Tipo 2	25	6	316	296	89	-0.4500
	axp	Tipo 1	33	3	363	243	84	-0.4500
		Tipo 2	21	6	276	256	77	-0.4500
HALDMADS	mb	Tipo 1	45	10	2891	2074	163	1.2229e-04
		Tipo 2	47	11	2935	2204	170	1.2188e-04
	axp	Tipo 1	38	10	2547	1902	147	1.2221e-04
		Tipo 2	40	11	2634	2032	155	1.2195e-04

continua ...

Tabela 6.2: Resultados Numéricos (continuação)

Problema	Penalidade		it.int	it.ext	func	grad	la	fótimo
HS10	mb	Tipo 1	30	8	100	82	120	-1.0000
		Tipo 2	30	8	100	82	120	-1.0000
	axp	Tipo 1	27	8	94	78	113	-1.0000
		Tipo 2	27	8	94	78	113	-1.0000
HS11	mb	Tipo 1	18	7	71	57	82	-8.4985
		Tipo 2	18	7	71	57	82	-8.4985
	axp	Tipo 1	20	7	75	61	88	-8.4985
		Tipo 2	20	7	75	61	88	-8.4985
HS100	mb	Tipo 1	29	6	211	146	97	680.6301
		Tipo 2	29	9	244	189	119	680.6301
	axp	Tipo 1	24	5	175	140	88	680.6301
		Tipo 2	27	9	234	179	113	680.6301
HS100MOD	mb	Tipo 1	20	4	144	114	72	678.6796
		Tipo 2	22	8	198	148	94	678.6796
	axp	Tipo 1	18	3	123	93	60	678.6796
		Tipo 2	21	7	182	132	85	678.6796
HS12	mb	Tipo 1	17	5	59	47	69	-30.0000
		Tipo 2	17	5	59	47	69	-30.0000
	axp	Tipo 1	16	5	57	47	68	-30.0000
		Tipo 2	16	5	57	47	68	-30.0000
HS22	mb	Tipo 1	21	8	119	95	95	1.0000
		Tipo 2	21	8	119	95	95	1.0000
	axp	Tipo 1	19	8	113	89	89	1.0000
		Tipo 2	19	8	113	89	89	1.0000
HS29	mb	Tipo 1	15	4	48	42	60	-22.6274
		Tipo 2	15	4	48	42	60	-22.6274
	axp	Tipo 1	21	4	60	44	68	-22.6274
		Tipo 2	21	4	60	44	68	-22.6274
HS43	mb	Tipo 1	27	8	180	140	109	-44.0000
		Tipo 2	23	8	164	124	97	-44.0000
	axp	Tipo 1	22	5	133	109	84	-44.0000
		Tipo 2	24	8	168	132	102	-44.0000
HS88	mb	Tipo 1	72	18	226	180	266	1.3627
		Tipo 2	72	18	226	180	266	1.3627
	axp	Tipo 1	76	18	234	190	280	1.3627
		Tipo 2	76	18	226	180	266	1.3627
HS89	mb	Tipo 1	57	14	178	142	210	1.3627
		Tipo 2	57	14	178	142	210	1.3627
	axp	Tipo 1	71	15	209	161	243	1.3627
		Tipo 2	71	15	209	161	243	1.3627
HS90	mb	Tipo 1	57	14	178	142	210	1.3627
		Tipo 2	57	14	178	142	210	1.3627
	axp	Tipo 1	71	15	209	161	243	1.3627
		Tipo 2	71	15	209	161	243	1.3627

continua ...

Tabela 6.2: Resultados Numéricos (continuação)

Problema	Penalidade		it.int	it.ext	func	grad	la	fótimo
HS91	mb	Tipo 1	65	15	181	165	233	1.3627
		Tipo 2	65	15	181	165	233	1.3627
	axp	Tipo 1	66	15	199	165	242	1.3627
		Tipo 2	66	15	199	165	242	1.3627
HS92	mb	Tipo 1	66	15	181	159	227	1.3627
		Tipo 2	66	15	181	159	227	1.3627
	axp	Tipo 1	67	15	181	153	221	1.3627
		Tipo 2	67	15	181	153	221	1.3627
HS113	mb	Tipo 1	35	9	486	369	133	24.3062
		Tipo 2	32	11	497	389	138	24.3062
	axp	Tipo 1	28	5	347	257	94	24.3062
		Tipo 2	29	11	470	371	131	24.3062
HS268	mb	Tipo 1	26	10	286	226	118	2.9298e-07
		Tipo 2	28	12	324	252	132	1.7079e-07
	axp	Tipo 1	26	8	260	212	110	2.0867e-07
		Tipo 2	27	11	305	239	125	5.1230e-08
KIWCRES	mb	Tipo 1	23	7	124	91	95	1.6463e-07
		Tipo 2	27	5	122	86	93	9.3472e-08
	axp	Tipo 1	22	8	128	95	98	5.3861e-07
		Tipo 2	28	4	118	82	90	-9.2225e-09
LISWET1	mb	Tipo 1	85	10	10615	7383	251	0.2475
		Tipo 2	120	11	14353	9505	330	0.2475
	axp	Tipo 1	117	10	14049	8494	307	0.2475
		Tipo 2	141	11	16474	10414	369	0.2475
LISWET2	mb	Tipo 1	53	12	7789	6072	197	0.2530
		Tipo 2	69	14	9811	7791	251	0.2530
	axp	Tipo 1	55	11	7788	5970	195	0.2530
		Tipo 2	57	13	8396	6679	215	0.2530
LISWET3	mb	Tipo 1	52	12	7688	5971	194	0.2530
		Tipo 2	68	14	9710	7690	248	0.2530
	axp	Tipo 1	55	11	7788	5970	195	0.2530
		Tipo 2	57	13	8396	6679	215	0.2530
LISWET4	mb	Tipo 1	47	9	6574	5160	167	0.2513
		Tipo 2	76	11	9909	7384	244	0.2513
	axp	Tipo 1	58	9	7786	5665	189	0.2513
		Tipo 2	73	11	9606	6778	229	0.2513
LISWET5	mb	Tipo 1	66	10	8696	6373	212	0.2520
		Tipo 2	92	12	11728	8496	284	0.2520
	axp	Tipo 1	82	10	10312	7181	244	0.2520
		Tipo 2	111	11	13444	8697	305	0.2520
LISWET6	mb	Tipo 1	54	12	7890	6173	200	0.2540
		Tipo 2	71	14	10013	7993	257	0.2540
	axp	Tipo 1	55	11	7788	5970	195	0.2540
		Tipo 2	59	13	8598	6881	221	0.2540

continua ...

Tabela 6.2: Resultados Numéricos (continuação)

Problema	Penalidade		it.int	it.ext	func	grad	la	fótimo
LISWET10	mb	Tipo 1	57	9	7584	5564	185	0.2508
		Tipo 2	78	11	10111	7182	242	0.2508
	axp	Tipo 1	65	13	9507	7083	234	0.2508
		Tipo 2	80	14	10922	7993	266	0.2508
MADSEN	mb	Tipo 1	34	9	380	303	137	0.6164
		Tipo 2	30	8	337	267	121	0.6164
	axp	Tipo 1	28	5	278	229	103	0.6164
		Tipo 2	26	7	294	231	105	0.6164
MAKELA1	mb	Tipo 1	24	8	128	101	102	-1.4142
		Tipo 2	23	8	125	98	99	-1.4142
	axp	Tipo 1	23	8	125	98	99	-1.4142
		Tipo 2	23	8	125	98	99	-1.4142
MAKELA2	mb	Tipo 1	26	9	193	149	116	7.2000
		Tipo 2	25	6	162	114	93	7.2000
	axp	Tipo 1	30	4	164	104	90	7.2000
		Tipo 2	20	5	133	101	80	7.2000
MAKELA3	mb	Tipo 1	23	2	611	464	73	-3.8078e-08
		Tipo 2	28	2	716	527	84	9.9853e-09
	axp	Tipo 1	20	2	548	443	68	4.4953e-09
		Tipo 2	30	2	758	611	94	5.7449e-06
MAKELA4	mb	Tipo 1	19	2	945	822	63	1.9507e-08
		Tipo 2	19	2	945	822	63	1.9507e-08
	axp	Tipo 1	20	3	1069	782	64	1.4979e-05
		Tipo 2	20	3	1069	782	64	1.4979e-05
MIFFLIN1	mb	Tipo 1	5	2	29	23	23	-1.0000
		Tipo 2	5	2	29	23	23	-1.0000
	axp	Tipo 1	9	3	48	39	39	-1.0000
		Tipo 2	9	3	48	39	39	-1.0000
MIFFLIN2	mb	Tipo 1	25	8	131	101	103	-1.0000
		Tipo 2	25	8	131	101	103	-1.0000
	axp	Tipo 1	26	7	127	103	104	-1.0000
		Tipo 2	25	8	131	104	105	-1.0000
MINMAXBD	mb	Tipo 1	38	9	1227	891	142	115.7064
		Tipo 2	36	10	1186	934	144	115.7064
	axp	Tipo 1	51	4	1243	802	135	115.7064
		Tipo 2	39	7	1120	889	137	115.7064
MINMAXRB	mb	Tipo 1	35	3	208	183	113	9.9986e-08
		Tipo 2	35	3	208	183	113	9.9986e-08
	axp	Tipo 1	35	5	230	185	117	-3.1854e-14
		Tipo 2	35	5	230	185	117	-3.1898e-14
OET1	mb	Tipo 1	23	8	281	225	101	0.4038
		Tipo 2	22	8	274	218	98	0.4038
	axp	Tipo 1	22	6	244	195	88	0.4038
		Tipo 2	19	5	208	173	77	0.4038

continua ...

Tabela 6.2: Resultados Numéricos (continuação)

Problema	Penalidade		it.int	it.ext	func	grad	la	fótimo
OET2	mb	Tipo 1	56	13	17877	13005	216	0.5382
		Tipo 2	58	14	17878	13615	222	0.5382
	axp	Tipo 1	46	13	15644	11990	195	0.5382
		Tipo 2	48	14	15848	12600	202	0.5382
OET3	mb	Tipo 1	16	4	172	144	64	-6.9481e-11
		Tipo 2	16	4	172	144	64	-6.9481e-11
	axp	Tipo 1	14	4	158	123	56	-5.3691e-08
		Tipo 2	14	4	158	123	56	-5.3693e-08
OET3 m=202	mb	Tipo 1	56	15	18488	13210	221	0.0045
		Tipo 2	60	16	19098	14226	234	0.0045
	axp	Tipo 1	45	15	16255	12195	200	0.0045
		Tipo 2	48	16	16662	13008	210	0.0045
OET4	mb	Tipo 1	29	2	226	149	74	-1.0199e-09
		Tipo 2	29	2	226	149	74	-1.0197e-09
	axp	Tipo 1	14	3	136	108	49	1.9849e-07
		Tipo 2	14	3	136	108	49	1.9849e-07
OET5	mb	Tipo 1	12	3	122	108	47	4.1683e-07
		Tipo 2	12	3	122	108	47	4.1662e-07
	axp	Tipo 1	13	4	137	116	51	-3.1472e-10
		Tipo 2	13	4	137	116	51	-3.1483e-10
OET6	mb	Tipo 1	27	3	220	171	79	9.9082e-09
		Tipo 2	27	3	220	171	79	9.9619e-09
	axp	Tipo 1	23	3	192	164	73	2.0809e-06
		Tipo 2	19	3	164	136	61	1.4253e-06
OET7	mb	Tipo 1	30	3	234	185	85	1.1001e-08
		Tipo 2	33	4	263	207	95	-7.9339e-09
	axp	Tipo 1	25	3	199	164	74	5.0543e-07
		Tipo 2	25	3	199	164	74	5.0530e-07
PENTAGON	mb	Tipo 1	35	3	627	451	95	1.3699e-04
		Tipo 2	44	5	837	645	132	1.3722e-04
	axp	Tipo 1	49	2	818	466	109	1.5280e-04
		Tipo 2	29	6	630	438	93	1.4631e-04
POLAK1	mb	Tipo 1	17	4	82	64	66	2.7183
		Tipo 2	18	2	71	56	59	2.7183
	axp	Tipo 1	17	3	78	63	65	2.7183
		Tipo 2	19	2	77	65	67	2.7183
POLAK2	mb	Tipo 1	14	3	66	51	53	54.5982
		Tipo 2	13	3	63	48	50	54.5982
	axp	Tipo 1	15	4	76	61	62	54.5982
		Tipo 2	15	4	76	61	62	54.5982
POLAK3	mb	Tipo 1	48	12	837	661	193	5.9330
		Tipo 2	51	13	871	684	200	5.9330
	axp	Tipo 1	43	12	793	606	179	5.9330
		Tipo 2	47	13	838	673	195	5.9330

continua ...

Tabela 6.2: Resultados Numéricos (continuação)

Problema	Penalidade		it.int	it.ext	func	grad	la	fótimo
POLAK4	mb	Tipo 1	52	4	252	164	142	1.7560e-07
		Tipo 2	63	5	305	193	169	2.6688e-10
	axp	Tipo 1	32	4	172	124	102	7.8236e-09
		Tipo 2	52	4	252	164	142	-3.3045e-11
POLAK5	mb	Tipo 1	27	2	92	71	76	50.0000
		Tipo 2	31	5	122	98	101	50.0000
	axp	Tipo 1	42	4	145	130	131	50.0000
		Tipo 2	42	4	145	130	131	50.0000
POLAK6	mb	Tipo 1	36	10	290	230	144	-44.0000
		Tipo 2	41	10	315	240	153	-44.0000
	axp	Tipo 1	35	6	241	181	117	-44.0000
		Tipo 2	33	10	275	210	133	-44.0000
POWELL20	mb	Tipo 1	25	12	551	419	123	57.8125
		Tipo 2	26	13	585	442	130	57.8125
	axp	Tipo 1	25	12	551	419	123	57.8125
		Tipo 2	31	15	686	521	153	57.8125
PT	mb	Tipo 1	78	15	57243	41179	278	0.1784
		Tipo 2	79	16	57746	41180	279	0.1784
	axp	Tipo 1	52	16	45698	33650	225	0.1784
		Tipo 2	54	16	44192	35156	228	0.1784
PT m=3	mb	Tipo 1	30	11	219	171	132	0.1429
		Tipo 2	29	12	224	172	133	0.1429
	axp	Tipo 1	26	8	176	144	110	0.1429
		Tipo 2	24	9	177	141	108	0.1429
PT m=101	mb	Tipo 1	56	13	8887	6541	215	0.1784
		Tipo 2	64	14	9704	7052	233	0.1784
	axp	Tipo 1	46	14	8174	6134	200	0.1784
		Tipo 2	49	14	7970	6338	202	0.1784
ROSENMMX	mb	Tipo 1	37	6	251	176	117	-44.0000
		Tipo 2	31	7	232	167	109	-44.0000
	axp	Tipo 1	44	3	253	143	106	-44.0000
		Tipo 2	32	7	237	162	108	-44.0000
SIPOW1	mb	Tipo 1	26	9	933	660	106	-1.0000
		Tipo 2	31	13	1210	916	143	-1.0000
	axp	Tipo 1	23	8	869	617	99	-1.0000
		Tipo 2	26	10	976	745	116	-1.0000
SIPOW1 m=100	mb	Tipo 1	31	12	5870	4254	142	-1.0000
		Tipo 2	35	14	6377	4862	159	-1.0000
	axp	Tipo 1	30	11	5566	4051	135	-1.0000
		Tipo 2	33	12	5769	4456	145	-1.0000

continua ...

Tabela 6.2: Resultados Numéricos (continuação)

Problema	Penalidade		it.int	it.ext	func	grad	la	fótimo
SIPOW1 m=500	mb	Tipo 1	37	14	33581	24062	163	-1.0000
		Tipo 2	40	16	36088	27571	182	-1.0000
	axp	Tipo 1	41	13	35584	25063	171	-1.0000
		Tipo 2	40	14	34583	26567	175	-1.0000
SIPOW1M	mb	Tipo 1	25	9	912	639	103	-1.0125
		Tipo 2	27	11	1040	788	123	-1.0125
	axp	Tipo 1	21	7	742	553	87	-1.0125
		Tipo 2	24	9	891	681	106	-1.0125
SIPOW1M m=100	mb	Tipo 1	27	11	5162	3748	125	-1.0005
		Tipo 2	34	14	6276	4761	156	-1.0005
	axp	Tipo 1	29	11	5465	3950	132	-1.0005
		Tipo 2	30	11	5263	4051	132	-1.0005
SIPOW1M m=500	mb	Tipo 1	37	14	34082	24062	164	-1.0000
		Tipo 2	39	16	35587	27070	179	-1.0000
	axp	Tipo 1	38	12	33078	23058	158	-1.0000
		Tipo 2	37	13	31576	24562	161	-1.0000
SIPOW2	mb	Tipo 1	41	10	1291	955	151	-1.0515
		Tipo 2	31	12	1167	894	139	-1.0515
	axp	Tipo 1	34	10	1186	871	138	-1.0515
		Tipo 2	29	11	1082	809	127	-1.0515
SIPOW2M	mb	Tipo 1	38	10	1228	892	142	-1.0000
		Tipo 2	31	13	1210	916	143	-1.0000
	axp	Tipo 1	28	7	889	679	106	-1.0000
		Tipo 2	30	9	1017	786	122	-1.0000
SIPOW2M m=100	mb	Tipo 1	45	12	7284	5365	178	-1.0000
		Tipo 2	40	14	6882	5266	172	-1.0000
	axp	Tipo 1	46	12	7587	5567	185	-1.0000
		Tipo 2	38	12	6274	4759	156	-1.0000
SIPOW2M m=500	mb	Tipo 1	52	14	42098	31076	208	-1.0000
		Tipo 2	47	16	39595	30577	201	-1.0000
	axp	Tipo 1	51	14	41597	30575	205	-1.0000
		Tipo 2	47	14	38090	28571	190	-1.0000
TFI1	mb	Tipo 1	26	7	463	367	98	5.3347
		Tipo 2	32	9	585	441	120	5.3347
	axp	Tipo 1	24	4	388	268	76	5.3347
		Tipo 2	26	9	525	405	109	5.3347
TFI1 m=51	mb	Tipo 1	86	6	4842	3178	215	5.3347
		Tipo 2	45	11	3339	2403	156	5.3347
	axp	Tipo 1	24	4	1564	1252	78	5.3347
		Tipo 2	32	8	2556	1932	123	5.3347
TFI2	mb	Tipo 1	38	11	731	551	150	0.6479
		Tipo 2	38	13	781	601	162	0.6479
	axp	Tipo 1	27	9	549	441	117	0.6479
		Tipo 2	30	11	635	503	134	0.6479

continua ...

Tabela 6.2: Resultados Numéricos (continuação)

Problema	Penalidade	it.int	it.ext	func	grad	la	fótimo	
TFI2 m=101	mb	Tipo 1	54	14	8684	6542	213	0.6490
		Tipo 2	56	15	8991	6849	222	0.6490
	axp	Tipo 1	44	13	7459	5623	183	0.6490
		Tipo 2	48	15	8175	6339	204	0.6490
TFI3	mb	Tipo 1	37	11	731	587	156	4.3011
		Tipo 2	38	13	781	625	166	4.3011
	axp	Tipo 1	33	10	670	526	141	4.3011
		Tipo 2	35	11	695	563	149	4.3011
TFI3 m=51	mb	Tipo 1	41	13	3653	2821	178	4.3011
		Tipo 2	50	15	4227	3395	211	4.3011
	axp	Tipo 1	45	14	4018	3082	195	4.3011
		Tipo 2	43	13	3653	2873	180	4.3011
TFI3 m=101	mb	Tipo 1	44	13	7561	5827	188	4.3012
		Tipo 2	52	15	8583	6849	218	4.3012
	axp	Tipo 1	45	13	7765	5827	190	4.3012
		Tipo 2	50	15	8379	6543	210	4.3012
WOMFLET	mb	Tipo 1	37	4	180	120	102	1.5299e-06
		Tipo 2	39	5	197	129	110	-1.4953e-11
	axp	Tipo 1	38	3	179	123	104	1.1141e-07
		Tipo 2	38	4	184	128	107	1.8987e-07

Para simplificar a análise dos resultados, construiremos outras tabelas menores que darão informações específicas sobre alguns dados da mesma.

O principal objetivo dos testes numéricos é avaliar o desempenho do algoritmo de lagrangeano aumentado utilizando a metodologia que estamos propondo. Começaremos pela tabela 6.3 em que cada número apresentado é referente a quantas vezes cada metodologia, por exemplo, tipo 1 com mb, executou menos iterações em cada um dos critérios, em relação à outra metodologia, tipo 2 com mb. A linha chamada iguais é o número de vezes que houve empate.

	Penalidade	it.int	it.ext	func	grad	la
mb	Tipo 1	41	48	43	45	45
	Tipo 2	20	7	20	18	18
	iguais	21	27	19	19	19
axp	Tipo 1	38	44	38	47	44
	Tipo 2	21	8	24	15	17
	iguais	23	30	20	20	21

Tabela 6.3: Comparação entre Tipo 1 e 2 para as funções mb e axp

O resultado é o que prevíamos, a penalidade tipo 1 ganha em todos os critérios. Veja na tabela 6.3 que a penalidade tipo 1 foi superior com as duas funções, mb e axp.

Na próxima tabela 6.4 computaremos quais das duas funções mb ou axp, tem um melhor desempenho tanto com as penalidades tipo1 como as de tipo 2. Os números listados na tabela foram calculados da mesma forma que na tabela 6.3. Ficamos surpresos, pois prevíamos que a função logaritmo quadrática tivesse um desempenho superior à função exponencial quadrática. Veja que a exponencial quadrática bateu a logaritmo quadrática em todos os critérios exceto do critério grad em que houve empate.

Penalidade		it.int	it.ext	func	grad	la
Tipo 1	mb	33	17	29	39	29
	axp	44	32	51	39	52
	iguais	5	33	2	4	1
Tipo 2	mb	23	13	21	19	21
	axp	53	36	57	59	59
	iguais	6	33	4	4	2

Tabela 6.4: Comparação entre mb e axp para os Tipos 1 e 2

Por último, apresentaremos na tabela 6.5 a soma das iterações executadas em cada um dos critérios, para cada uma das metodologias. Por exemplo, a primeira linha é referente ao total de iterações executadas pelo algoritmo com a penalidade tipo 1 e função mb, para cada um dos critérios.

Penalidade		it.int	it.ext	func	grad	la
Tipo 1	mb	3102	693	346459	254019	11280
	axp	3010	649	333171	241190	10795
Tipo 2	mb	3358	795	381379	284859	12306
	axp	3075	735	336937	255908	11380

Tabela 6.5: Total de iterações e avaliações

Conclusão sobre os testes numéricos :

Pelos testes que fizemos e os resultados mostrados nas tabelas desta seção, vimos que o algoritmo de lagrangeano aumentado com a metodologia proposta neste trabalho teve um desempenho superior, em relação ao mesmo algoritmo, quando aplicado com a metodologia mais utilizada na literatura.

Assim, oferecemos um novo algoritmo com boas propriedades de convergência. Uma opção a mais para as pessoas interessadas nos métodos tipo lagrangeano aumentado.

Futuramente, deveremos fazer novos testes com problemas de grande porte e utilizando algum programa escrito em Fortran. Por exemplo, o BOX-QUACAN desenvolvido por professores da Unicamp com a supervisão do professor Martinez.

Considerações Finais

A metodologia proposta aqui mostrou-se promissora. Vimos nos testes numéricos que esta metodologia se sobressaiu em relação à mais utilizada na literatura. Mostramos também, no caso de programação linear, que as direções duais geradas pelo algoritmo de lagrangeano aumentado com essa nova metodologia são equivalentes às geradas pelo algoritmo afim escala aplicado ao problema dual. Estes resultados são inéditos, pois nenhuma prova dos mesmos foi encontrada na literatura.

No capítulo 4 apresentamos um algoritmo com penalidade quadrática, mas não o implementamos. Acreditamos que possa ser um bom algoritmo e então deixamos como uma tarefa a ser continuada para depois do doutorado. Além da implementação pensamos em voltar a estudar melhor seu comportamento teórico, em relação à convergência. Outra tarefa que deixamos para depois do doutorado é o estudo da convergência da sequência primal gerada pelos algoritmos utilizando esta nova metodologia.

Referências Bibliográficas

- [1] A. Ben-Tal and M. Zibulevsky. Penalty-barrier multiplier methods for convex programming problems. *SIAM Journal on Optimization*, 7:347–366, 1997.
- [2] D. P. Bertsekas. *D. Constrained Optimization and Lagrange Multipliers*. Academic Press, Ney York, 1992.
- [3] L. Bregman. The relaxation method for finding the common point of convex sets and its applcation to the solution of proglems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7:200–217, 1967.
- [4] R. Castillo. *Penalidades Generalizadas e Métodos de Multiplicadores para Programação Não Linear*. PhD thesis, COPPE - Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, 1998. In Portuguese.
- [5] Y. Censor and S. Zenios. The proximal minimization algorithm with d-functions. *Journal Optmization Theory and Aplications*, 73:451–464, 1992.
- [6] C. Chen and M. Teboulle. Convergence analysis of a proximal-like minimization algorithm using Bregman function. *SIAM J. Optimization*, 3:538–543, 1993.
- [7] J. E. Dennis and R. B. Schnabeel. *Numerical Mhetdos For Unconstranined Optimization and Nonlinear Equation*. SIAM, Philadelphia - United States, 1996.
- [8] J. Eckstein. Nonlinear proximal point algorithms using Bregman functions with aplications to convex programming. *Mathematics of Operations Resaerchs*, 18(1):202–226, 1993.
- [9] P. P. B. Eggermont. Multiplicative iterative algorithms for convex programming. *Linear Algebra and Its Applications*, 130:25–42, 1990.

- [10] Y. C. R. Espinoza. Um teorema de equivalência entre métodos lagrangeano aumentado e algoritmos de pontos proximais. Master's thesis, Universidade Federal de Santa Catarina, Santa Catarina, Brazil, 1998. In Portuguese.
- [11] C. C. Gonzaga. Algoritmos de pontos interiores para programação linear. 17^o Colóquio Brasileiro de Matemática, Rio de Janeiro, Brazil, 1989.
- [12] C. C. Gonzaga. Path following methods for linear programming. *SIAM Review*, 34(2):167–227, 1992.
- [13] M. Hestenes. Multiplier and gradient methods. *Journal Optimization Theory and Applications*, 4:303–320, 1969.
- [14] J-B. Hiriart-Urruty and C. Lemarechal. *Convex Analysis and Minimization Algorithms I*. Springer-Verlag, New York, 1993.
- [15] J-B. Hiriart-Urruty and C. Lemarechal. *Convex Analysis and Minimization Algorithms II*. Springer-Verlag, New York, 1993.
- [16] A. R. Conn I. Bongartz, N. I. M. Gould and Ph. L. Toint. CUTE: Constrained and unconstrained testing environment. *ACM Transactions on Mathematical Software*, 21:123–160, 1995.
- [17] A. Iusem. Métodos de ponto proximal em otimização. 20^o Colóquio Brasileiro de Matemática, Rio de Janeiro, Brazil, 1995. In Portuguese.
- [18] A. Iusem. Augmented lagrangian methods and proximal point methods for convex optimization. *Minicurso I Workshop em Otimização*, pages 1–38, 1997.
- [19] A. Iusem and M. Teboulle. On the convergence rate of entropic proximal optimization methods. *Computational and Applied Mathematics*, 12:153–168, 1993.
- [20] A. Iusem and M. Teboulle. Convergence rate of nonquadratic proximal point methods for convex and linear programming. *Mathematics of Operations Research*, 20:657–677, 1995.
- [21] A. Iusem, M. Teboulle, and B. Svaiter. Entropy-like proximal methods in convex programming. *Mathematics of Operations Research*, 19(4):790–814, 1994.
- [22] K. Kiwiel. Proximal minimization methods with generalized Bregman functions. *SIAM J. Control Optimization*, 35(4):1142–1168, July 1997.

- [23] J. M. Martínez. Box-quacan and the implementation of augmented lagrangian algoritmos for minimization with inequality constraints. *Computational & Applied Mathematics*, 19:31–56, 2000.
- [24] A. De Pierro and A. Iusem. A relaxed version of Bregman’s method for convex programming. *Journal Optmization Theory and Aplications*, 5:421–440, 1986.
- [25] R. Polyak and M. Teboulle. Nonlinear rescaling and proximal-like methods in convex optimization. *Mathematical Programming*, 76:265–284, 1997.
- [26] M. J. D. Powell. A method for nonlinear constraints in minimizations problems in optimization. *R . Fletcher*, pages 283–298, 1969.
- [27] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, New Jersey, 1970.
- [28] R. T. Rockafellar. Augmented lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Researchs*, 1(2):97–116, 1976.
- [29] R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control Optmization*, 14(5):877–898, 1976.
- [30] P. Silva. *Tópicos em Métodos de Ponto Proximal*. PhD thesis, USP - Universidade de São Paulo, São Paulo, Brazil, 2000. In Portuguese.
- [31] M. Teboulle. Entropic proximal mappings with applications to nonlinear programming. *Mathematics of Operations Research*, 17:97–116, 1992.
- [32] P. Tseng and D. Bertsekas. On the convergence of exponential multiplier method for convex programming. *Mathematical Programming*, 60:1–19, 1993.