



PAULA APARECIDA KIKUCHI

Métodos de Pontos Interiores Aplicados à Basis Pursuit

CAMPINAS
2013



Universidade Estadual de Campinas

**Instituto de Matemática, Estatística
e Computação Científica**

Paula Aparecida Kikuchi

Métodos de Pontos Interiores Aplicados à Basis Pursuit

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestra em matemática aplicada.

Orientadora: Daniela Renata Cantane

Coorientador: Aurelio Ribeiro Leite de Oliveira

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA PELA ALUNA PAULA APARECIDA KIKUCHI, E ORIENTADA PELA PROFA. DRA. DANIELA RENATA CANTANE.

Assinatura da Orientadora

Assinatura do Coorientador

**Campinas
2013**

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

K55m Kikuchi, Paula Aparecida, 1987-
Métodos de pontos interiores aplicados à basis pursuit / Paula Aparecida Kikuchi. – Campinas, SP : [s.n.], 2013.

Orientador: Daniela Renata Cantane.
Coorientador: Aurelio Ribeiro Leite de Oliveira.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Programação linear. 2. Métodos de pontos interiores. 3. Processamento de sinais. I. Cantane, Daniela Renata. II. Oliveira, Aurelio Ribeiro Leite de, 1962-. III. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Interior point methods applied to basis pursuit

Palavras-chave em inglês:

Linear programming

Interior point methods

Signal processing

Área de concentração: Matemática Aplicada

Titulação: Mestra em Matemática Aplicada

Banca examinadora:

Daniela Renata Cantane [Orientador]

Francisco de Assis Magalhães Gomes Neto

Silvana Bocanegra

Data de defesa: 23-08-2013

Programa de Pós-Graduação: Matemática Aplicada

Dissertação de Mestrado defendida em 23 de agosto de 2013 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.



Prof.(a). Dr(a). DANIELA RENATA CANTANE



Prof.(a). Dr(a). FRANCISCO DE ASSIS MAGALHÃES GOMES NETO



Prof.(a). Dr(a). SILVANA BOCANEIRA

Abstract

There are many proposed methods for signal reconstruction. However, our focus is on the Basis Pursuit method. When working with overcomplete dictionaries, there exist countless possible combinations to represent the signal. Basis Pursuit finds the sparsest, because it minimizes the sum of the combination coefficients absolute values, i.e., it minimizes the coefficients on norm 1. We will see that the problem in question can be rewritten as a linear programming problem. An existing method is shown for the solution of this problem, the Primal-Dual Logarithmic Barrier Method. Initially, we will apply the Logarithmic Barrier Method, and seeking higher efficiency, we will include the affine scaling direction, the centering direction and the nonlinear correction direction in the same method, obtaining the Predictor-Corrector Primal-Dual Logarithmic Barrier Method, of which a variation is also implemented. Computational results with real life problems show the efficiency of the proposed method.

Keywords: linear programming, interior point methods, signal processing.

Resumo

Vários são os métodos propostos para reconstrução de sinal. Nosso enfoque é o método *Basis Pursuit*. Trabalhando com dicionários overcomplete, são inúmeras as combinações possíveis para a representação do sinal. *Basis Pursuit* encontra a mais esparsa, porque minimiza a soma dos módulos dos coeficientes da combinação, ou seja, minimiza os coeficientes na norma 1. Veremos que podemos reescrever o problema em questão como um problema de programação linear. Apresentaremos um método já existente para a resolução deste problema, o Método Primal-Dual Barreira Logarítmica. Em um primeiro momento, vamos aplicar o Método Barreira Logarítmica, e buscando maior eficiência, iremos incluir a direção afim-escala, a direção de centragem e a direção de correção no mesmo método, obtendo o Método Primal-Dual Barreira Logarítmica Predictor-Corretor, além de implementar uma variação deste. Resultados computacionais com problemas reais comprovam a eficiência do método proposto.

Palavras-chave: programação linear, métodos de pontos interiores, processamento de sinais.

Sumário

Dedicatória	ix
Agradecimentos	x
1 Motivação	1
2 <i>Basis Pursuit</i>	6
2.1 Representação de sinais	6
2.2 O Método <i>Basis Pursuit</i>	7
2.2.1 Basis Pursuit Denoising	8
2.2.2 Resolução de BP e BPDN	10
3 Método de Pontos Interiores	11
3.1 Otimização Linear	11
3.2 Otimização Não Linear	13
3.2.1 Método de Newton Aplicado as Condições de Otimalidade	15
3.3 Método Primal-Dual	15
3.3.1 Método Primal-Dual Afim-Escala	15
3.3.2 Método Primal-Dual Seguidor de Caminho	18
3.4 Método de Pontos Interiores Barreira Logarítmica	19
3.5 Método de Pontos Interiores Barreira Logarítmica Preditor-Corretor	21
4 Método Primal-Dual Barreira Logarítmica Aplicado a BP e BPDN	24
4.1 Formulação do Problema BP	24
4.2 Aplicação do Método à BP	25
4.3 Basis Pursuit Denoising	28
4.4 BP_Interior	29
5 Desenvolvimento dos Métodos Propostos	30
5.1 Método Barreira Logarítmica	30
5.2 Método Primal-Dual Barreira Logarítmica Preditor-Corretor	33
5.3 Variante do Método Primal-Dual Barreira Logarítmica Preditor-Corretor	34
5.4 Características dos Métodos Apresentados	37

6	Testes Computacionais	39
7	Conclusões e Perspectivas Futuras	50
7.1	Conclusões	50
7.2	Perspectivas Futuras	51
A	Método dos Gradientes Conjugados	52
A.2	Método da Máxima Descida	53
A.3	Método dos Gradientes Conjugados	54
B	Método de Newton	57
	Referência Bibliográfica	61

À minha família.

Agradecimentos

À Deus, por me conduzir e dar forças para enfrentar cada etapa da vida.

Aos meus pais, Adelia e Paulo, pela paciência e ajuda em todos os momentos que precisei, em especial minha mãe, que sempre incentivou e apoiou os meus estudos.

Ao meu irmão Lincoln e minha cunhada Camila, pelo incentivo e disposição nas horas de desânimo. Agradeço ao meu irmão pela sua paciência nos meus anos iniciais de escolaridade, em que as “contas com letras” que ele estudava anos a frente, me despertavam curiosidade e interesse na matemática, curiosidade e interesse que se mantêm até hoje.

À minha orientadora Daniela, pelos ensinamentos, paciência, dedicação, disposição e pela oportunidade da orientação.

Ao meu coorientador Aurelio, pela oportunidade, paciência, ensinamentos, sempre estar disposto quando precisei e por despertar meu interesse em Programação Linear.

Aos meus amigos, Julianna, Josimar, Jorge, Denis, Karina, Ingrid, Karla, os quais sempre estiveram presentes quando necessitei, sendo nas horas de desânimo ou dúvidas referentes ao projeto.

À CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, pelo apoio financeiro.

Capítulo 1

Motivação

Algo que muitas vezes não percebemos, mas faz parte constante de nossas vidas, são os sinais. Seja na imagem reproduzida no televisor, em fotos, na música que ouvimos de um rádio, no uso cotidiano do celular..., os sinais são os responsáveis pela reprodução e uso destes.



Figura 1.1: Gold Hill (Extraído de [1]).

Ao olharmos uma foto, por exemplo a Figura 1.1, notamos que as partes que não correspondem à região de contorno são muito similares às adjacentes. Como podemos ver na Figura 1.2, que corresponde a uma aproximação de uma parte da Figura 1.1, as regiões correspondentes ao telhado e a parede rosa da casa são muito parecidas. Sendo que a transição do telhado à parede, correspondente à região de contorno, contém pontos bem distintos.



Figura 1.2: Aproximação da Imagem Gold Hill (Extraído de [1]).

Assim se armazenarmos uma imagem através de uma matriz na qual cada coeficiente corresponde à sua intensidade, teremos muitos valores redundantes. A fim de evitar o armazenamento desses elementos similares, procurando economizar memória, vamos aplicar a compressão de sinal.

Considere um vetor coluna $x \in \mathbb{R}^N$, tendo seus elementos representados por $x(n)$, com n variando de 1 a N . Qualquer sinal em \mathbb{R}^N pode ser representado em termos dos elementos de uma base $\{\psi_i\}_{i=1}^N$. Supondo que a base seja ortonormal, o vetor x poderá ser expresso como:

$$x = \sum_{i=1}^N \alpha_i \psi_i \text{ ou } x = \Psi \alpha, \quad (1.1)$$

onde $\alpha_i = \langle x, \psi_i \rangle$.

Definimos x como um sinal K – *esparso*, se é combinação linear de K vetores da base em (1.1). O sinal é dito compressível se a representação em (1.1) tem poucos coeficientes de módulo grande e muitos coeficientes de módulo pequeno. Sinais compressíveis são bem aproximados por formas de representação K – *esparsas*.

Obter uma representação esparsa para o sinal nos dá um número menor de coeficientes armazenados ou transmitidos. Para tanto temos a *Codificação por Transformada* [23, 3], que corresponde a técnicas de compressão de imagem. As três operações básicas são:

- **Transformação**, que transforma a imagem em um conjunto de coeficientes com pouca redundância.
- **Quantização**, cujo objetivo é representar uma grande faixa de valores através de um conjunto pequeno de símbolos.
- **Codificação**, que visa mapear os símbolos com o objetivo de obter o menor número possível de bits.

Como consta em [2], este quadro de amostragem-compressão possui três desvantagens. A primeira é que o número de amostras pode ser muito alto, tendo em vista o teorema de amostragem Shannon-Nyquist [12], que nos diz, de uma forma genérica, que um sinal analógico limitado em banda que foi amostrado, pode ser recuperado se a taxa de amostragem for maior que $2 \times F$

amostras por segundo, sendo F a maior frequência do sinal original. A segunda desvantagem corresponde ao fato que os N coeficientes devem ser calculados, embora saibamos que apenas K não serão descartados. E por fim a terceira, pelo fato que a localização dos maiores coeficientes deve ser codificada.

O método *Compressive Sensing* contorna as desvantagens descritas, obtendo diretamente uma representação do sinal em uma forma comprimida, sem a necessidade de adquirir N amostras, como feito pela *Codificação por Transformada*. Nesse método, considera-se um processo que calcula $M < N$ produtos internos, entre x e um conjunto de vetores $\{\phi_j\}_{j=1}^M$, ou seja, $s_j = \langle x, \phi_j \rangle$. Sendo s o vetor coluna formado pelos elementos s_j , e ϕ_j^T a linha j da matriz $\Phi : M \times N$; podemos reescrever (1.1) como:

$$s = \Phi x = \Phi \Psi \alpha = \Theta \alpha,$$

onde $\Theta = \Phi \Psi$ é uma matriz $M \times N$. Na Figura (1.3), podemos visualizar esse procedimento.

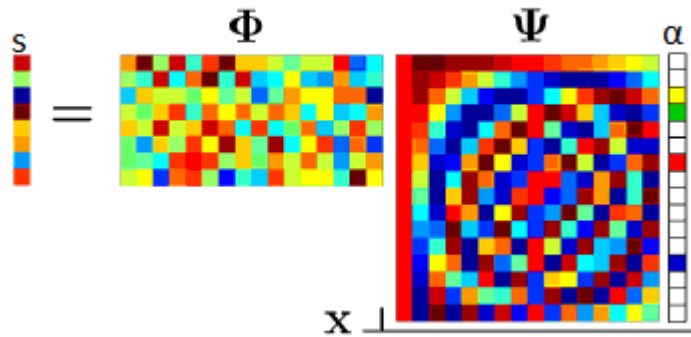


Figura 1.3: Matriz de aquisição (Extraída de [2]).

O método *Compressive Sensing* consiste em:

- a) Obter uma matriz Φ tal que as informações salientes em qualquer sinal K – *esparso* ou compressivo não sejam comprometidas pela redução de dimensão.
- b) Um algoritmo de reconstrução para recuperar x das M medidas de s .

O artigo [2] descreve como podemos obter a matriz Φ do item (a).

O objetivo do item (a) relaciona-se com Processamento de Sinais, que não é área que focamos nessa dissertação, estamos interessados em obter o algoritmo de reconstrução referente ao item (b). Note que, sendo um sinal compressível, x terá uma boa representação como vetor K – *esparso*. Dessa forma, sendo $M \approx K$, gostaríamos de obter um sinal α com aproximadamente $N - K$ elementos iguais a zero, tendo em vista que buscamos a sua representação esparsa. Sendo assim,

minimizar a soma dos elementos em módulo do vetor α é uma excelente opção para alcançar nosso objetivo. Ao problema a seguir:

$$\begin{aligned} &\text{minimizar} && \|\alpha\|_1 \\ &\text{sujeito a} && \Theta\alpha = s \end{aligned} \quad (1.2)$$

é dado o nome *Basis Pursuit*.

A aplicação que destacaremos de *Basis Pursuit* é a representação de sinais através de dicionários, como feito em [6], que veremos no Capítulo 3. Trabalharemos com dicionários que possibilitam infinitas formas de representar o sinal e, com a finalidade de encontrar a mais esparsa, determinaremos a representação através de *Basis Pursuit*.

Em [16] é dada uma motivação geométrica para escolha de tal procedimento que minimiza a norma 1 dos coeficientes.

Dado o problema (1.2), considere:

$$H = \{\alpha \in \mathbb{R}^N : \Theta\alpha = s\}.$$

Sabemos que *Basis Pursuit* encontra o elemento α^* de H que tem a norma 1 mínima. Isto pode ser determinado aumentando o parâmetro τ da bola B_τ referente a norma 1:

$$B_\tau = \{\alpha \in \mathbb{R}^N : \|\alpha\|_1 \leq \tau\},$$

até interceptar H .

Como podemos ver na Figura 1.4 (a), a bola B_τ referente à norma 1 permanece mais próxima dos eixos do que a bola determinada pela norma 2 (Figura 1.4 (b)). Assim, a solução determinada pela norma 1 é propensa a ter mais coeficientes nulos ou próximos de zero.

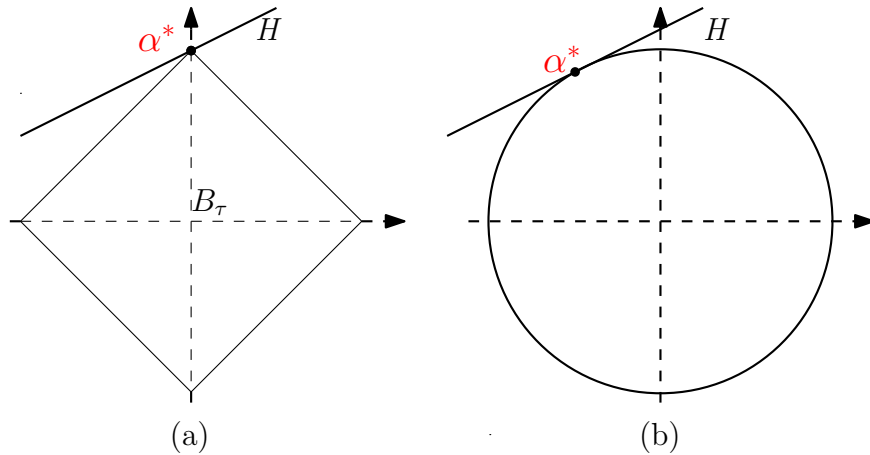


Figura 1.4: Esparsidade e as normas 1 e 2.

Como ilustrado na Figura (1.4), a solução α^* determinada por *Basis Pursuit* é justamente a mais esparsa.

No Capítulo 2, apresentamos uma abordagem superficial a respeito da representação de sinais. Neste capítulo, também descrevemos com maiores detalhes o método *Basis Pursuit*. Veremos que, através de transformações, podemos reescrever nosso problema como um problema de programação linear.

A fim de solucionar o problema *Basis Pursuit*, no Capítulo 3, introduzimos os Métodos de Pontos Interiores, bem como definimos alguns conceitos básicos de programação linear e não-linear.

No Capítulo 4, é descrito o Método Primal-Dual Barreira Logarítmica e suas aplicações aos problemas *Basis Pursuit* e *Basis Pursuit Denoising*. Esse último, como veremos, é utilizado quando queremos obter a representação do sinal sem ruído. Para tanto, reescreveremos nosso problema de programação linear, como feito em [10].

No Capítulo 5, é descrita a implementação BP_Interior, feita em [6], que implementa o Método Primal-Dual Barreira Logarítmica para resolver o problema *Basis Pursuit*; bem como do Método Barreira Logarítmica, do Método Primal-Dual Barreira Logarítmica Preditor-Corretor e de uma variação deste último, desenvolvidos a fim de buscar uma maior eficiência do problema.

Apresentamos no Capítulo 6 os resultados obtidos nos testes computacionais para determinados sinais, utilizados em [6].

No Capítulo 7, concluimos fazendo uma análise dos resultados obtidos e perspectivas futuras do nosso trabalho.

Capítulo 2

Basis Pursuit

Neste capítulo, apresentaremos a forma de representação de sinal pelo Método *Basis Pursuit*, que é o foco deste trabalho. Antes porém, vamos introduzir alguns conceitos fundamentais para o entendimento do assunto em questão.

2.1 Representação de sinais

Definimos dicionário como uma coleção de *waveforms* parametrizadas $D = (\phi_\gamma : \gamma \in \Gamma)$, sendo *waveforms* ϕ_γ sinais discretos de tempo chamados átomos, que nessa dissertação serão vistos como vetores pertencentes a \mathbb{R}^n , e Γ o conjunto de todos os parâmetros do dicionário. Representação de sinais por meio de dicionários é uma alternativa que surge, além das representações usuais que utilizam superposição de senóides, como mencionado em [8].

Considerando um sinal s como sendo um vetor pertencente a \mathbb{R}^n , sua decomposição em um dicionário é dada por:

$$s = \sum_{\gamma \in \Gamma} \alpha_\gamma \phi_\gamma. \quad (2.1)$$

Considerando que temos um dicionário discreto de p *waveforms* e uma matriz Φ cujas colunas correspondem às p *waveforms* ($\Phi : n \times p$), podemos reescrever (2.1) como:

$$\Phi \alpha = s, \quad (2.2)$$

onde $\alpha = (\alpha_\gamma)$ é o vetor dos coeficientes em (2.1).

Veja que, se as *waveforms* são linearmente independentes, teremos uma matriz Φ não singular, o que implica que α terá representação única. Em particular, se as *waveforms* forem ortonormais $\Phi^{-1} = \Phi^T$.

Dicionários são definidos como completos quando possuem exatamente n átomos, *overcomplete* quando possuem mais que n átomos, ou *undercomplete* quando possuem menos de n átomos. Exemplos de dicionários dados em [6] são: *stationary wavelets*, *wavelet packets*, *cosine packets*, *chirples*, *warplets*, *Gabor dictionaries*, *wavelets* e *steerable wavelets*.

Note que, em (2.1), se trabalhamos com um dicionário completo, s terá uma representação única, pois todos α_γ serão linearmente independentes. Já quando fazemos uso de dicionários *overcomplete*, sua representação não será única (teremos átomos que são combinações lineares de outros), permitindo que escolhamos a que nos for mais conveniente. Como um dos objetivos que queremos alcançar nas representações de sinais é esparsidade (obter o menor número de coeficientes significativos), já podemos ter em mente que uma boa ideia seria minimizar $\sum_{\gamma \in \Gamma} |\alpha_\gamma|$.

2.2 O Método *Basis Pursuit*

Não são poucos os métodos propostos para representar sinais em dicionários *overcomplete*. Alguns exemplos, descritos em [6] são: *método de frames* (MOF), que escolhe a representação cujos coeficientes tenham a norma 2 mínima, *matching pursuit* (MP), *the best orthogonal basis* (BOB) e *basis pursuit* (BP). Diferentemente do *método de frames*, BP irá escolher a representação cujos coeficientes tenham a norma 1 mínima. Chamamos essa decomposição do sinal de superposição “ótima” de elementos do dicionário. Chen, Donoho e Saunders [8] mostram várias vantagens de BP sobre MOF, MP e BOB, dentre elas, a esparsidade. Note que BP baseia-se na resolução do seguinte problema:

$$\begin{aligned} &\text{minimizar} \quad \|\alpha\|_1 \\ &\text{sujeito a} \quad \Phi\alpha = s \end{aligned} \quad (2.3)$$

Podemos reformular o problema (2.1) por meio de um programa linear na forma padrão (3.1).

De fato, considere e o vetor pertencente a \mathbb{R}^p , cujos elementos são compostos por 1, escrevendo α em (2.3) como a diferença de dois vetores não-negativos u e v , pertencentes a \mathbb{R}^p , e sendo $\|\alpha\|_1$ limitante superior de $e^T u + e^T v$, pois estamos minimizando $\|\alpha\|_1$ e, nesse caso, $u_i v_i = 0$ para i variando de 1 a p , teremos $\Phi\alpha = \Phi(u - v)$ e $e^T u + e^T v = \|\alpha\|_1$, esta última segue do seguinte fato:

$$\sum_{\gamma=1}^p |\alpha_\gamma| = \sum_{\gamma=1}^p |u_\gamma - v_\gamma| \leq \sum_{\gamma=1}^p |u_\gamma| + \sum_{\gamma=1}^p |v_\gamma| = \sum_{\gamma=1}^p u_\gamma + v_\gamma = e^T u + e^T v \leq \sum_{\gamma=1}^p |\alpha_\gamma| \Rightarrow \sum_{\gamma=1}^p |\alpha_\gamma| = e^T u + e^T v.$$

Dessa forma obtemos o seguinte problema equivalente:

$$\begin{aligned} &\text{minimizar} \quad e^T u + e^T v \\ &\text{sujeito a} \quad \Phi(u - v) = s, \\ &\quad \quad \quad (u, v) \geq 0 \end{aligned}$$

que pode ser reformulado como um problema de programação linear na forma padrão por meio de transformações. Como visto em [6], essa reformulação é obtida por meio das seguintes associações:

$$\begin{aligned} m &\Leftrightarrow 2p \\ x &\Leftrightarrow (u; v) \\ c &\Leftrightarrow (e; e) \\ A &\Leftrightarrow (\Phi, -\Phi) \\ b &\Leftrightarrow s. \end{aligned} \quad (2.4)$$

Resolveremos o problema linear na forma padrão, equivalente a (2.3), por meio do método de pontos interiores, que será o enfoque desse trabalho.

Até o momento foi discutido como representar um sinal por meio de dicionários *overcomplete*. Agora vamos analisar uma aplicação de processamento de sinal, como feito em [5], quando levamos em conta o ruído do sinal.

2.2.1 Basis Pursuit Denoising

Através da tomografia axial computadorizada é diagnosticada a presença de tumores e alterações celulares. Isso é feito emitindo raios (sinais) em diferentes ângulos nas regiões onde tais presenças são suspeitas, a fim de produzir uma imagem em três dimensões da parte interna do corpo. O problema é descrito da seguinte forma:

Sendo $R \subset \mathbb{R}^3$ a região de interesse, assuma que R_i , $1 \leq i \leq p$, correspondam a pequenas células que formam R e que α_i seja a densidade da célula R_i . Nosso objetivo é determinar o valor dos α_i .

Supondo que um raio (sinal) L é transmitido para nossa região de interesse R , várias células de R serão interceptadas pelo raio. Definimos $I(L)$ o conjunto dessas células. Essas células interceptadas pelo sinal absorvem energia do mesmo, sendo a energia absorvida por cada célula proporcional à densidade e ao comprimento do caminho percorrido pelo raio na respectiva célula. O comprimento $\phi_i(L)$ do caminho percorrido pelo raio no interior da célula é conhecido para todo $i \in I(L)$. Assim a quantidade de energia absorvida é dada por:

$$y(L) = \sum_{i \in I(L)} \alpha_i \phi_i(L).$$

A energia absorvida $y(L)$ é estimada subtraindo a energia final pela energia inicial emitida (note que não passa de uma estimativa, o valor de y não é exato).

Esse processo é repetido enviando os raios n vezes, obtendo dessa forma:

$$y(L_j) = \sum_{i \in I(L_j)} \alpha_i \phi_i(L_j), 1 \leq j \leq n. \quad (2.5)$$

Como α_i representa a densidade da célula, temos que $\alpha_i \geq 0$, $1 \leq i \leq p$.

Devido ao custo dos raios-x e ao seu perigo, n geralmente é diferente do número de células p , tendo o sistema poucas equações ($n < p$).

Observe que podemos reescrever (2.5) como:

$$\Phi \alpha = y, \quad \alpha \geq 0, \quad (2.6)$$

com $\Phi : n \times p$, sendo que cada linha da matriz corresponde a um processo de emissão do raio (como mencionado anteriormente, tal processo é repetido n vezes), e o número de colunas corresponde ao número de células que pertencem ao conjunto R .

Sabemos que o valor de y não é exato. Dessa forma, consideramos $y = s + \sigma z$, onde z é um ruído branco Gaussiano padrão, σ é o nível de ruído, e s é o sinal puro. Assim, quando resolvemos

o problema:

$$\begin{aligned} & \text{minimizar} && \|y - \Phi\alpha\|_2^2 \\ & \text{sujeito a} && \alpha \geq 0 \end{aligned},$$

não estamos determinando os valores de α que desejamos (que correspondem à representação do sinal puro s). Além disso, tal problema possui infinitas soluções, considerando que a matriz em questão possui mais variáveis que equações. Em [5], é atribuído o coeficiente $\frac{1}{2}$ à norma anterior e, com a finalidade de encontrar α esparso, minimiza-se conjuntamente $\|\alpha\|_1$, obtendo:

$$\begin{aligned} & \text{minimizar} && \frac{1}{2} \|y - \Phi\alpha\|_2^2 + \lambda \|\alpha\|_1, \\ & \text{sujeito a} && \alpha \geq 0 \end{aligned}, \quad (2.7)$$

onde $\lambda \geq 0$ é um parâmetro escalar. Como podemos ver em [6], a solução desse problema refere-se à *Basis Pursuit Denoising* (BPDN), que é um caso BP adaptado para s perturbado:

$$y = s + \sigma z,$$

onde o objetivo não é a decomposição exata de y (o que é obtido aplicando-se BP a y). Neste, queremos uma representação do tipo:

$$y = \sum_{i=1}^m \alpha_{\gamma_i} \phi_{\gamma_i} + R^{(m)},$$

onde $R^{(m)}$ é o resíduo e estamos considerando \mathbb{D} um dicionário, onde as *waveforms* $(\phi_\gamma)_{\gamma \in \Gamma}$ são os seus elementos.

Utilizando a mesma notação que usamos para BP, teremos o problema equivalente a (2.7):

$$\begin{aligned} & \text{minimizar} && \|y - \Phi(u - v)\|_2^2 + \lambda e^T u + \lambda e^T v \\ & \text{sujeito a} && (u, v) \geq 0 \end{aligned}. \quad (2.8)$$

Definindo $x = (u; v)$, $c = \lambda(e; e)$, $b = y$ e $A = (\Phi, -\Phi)$, (2.8) é escrito como:

$$\begin{aligned} & \text{minimizar} && \|Ax - b\|_2^2 + c^T x \\ & \text{sujeito a} && x \geq 0 \end{aligned}. \quad (2.9)$$

Por sua vez, Michael Saunders e Shaobing Scott Chen [6] constataram que (2.9) é equivalente ao problema linear perturbado:

$$\begin{aligned} & \text{minimizar} && c^T x + \frac{1}{2} \|p\|_2^2 \\ & \text{sujeito a} && Ax + \delta p = b \\ & && x \geq 0 \\ & && \delta = 1 \end{aligned}. \quad (2.10)$$

2.2.2 Resolução de BP e BPDN

Como vimos, os dois problemas analisados (BP e BPDN) podem ser reformulados. BP é reescrito como um problema linear e BPDN como um problema quadrático (problema linear perturbado). Nosso enfoque será trabalhar com o problema referente à BP aplicando Métodos de Pontos Interiores.

No Capítulo 4, vamos analisar o método proposto em [6] para o problema BP, que veremos ser adaptável a BPDN. Nele resolvemos o problema linear perturbado:

$$\begin{aligned} &\text{minimizar} && c^T x + \frac{1}{2} \|\gamma x\|_2^2 + \frac{1}{2} \|p\|_2^2 \\ &\text{sujeito a} && Ax + \delta p = b \\ &&& x \geq 0 \end{aligned} \tag{2.11}$$

onde γ e δ são parâmetros de perturbação pequenos, da ordem de (10^{-4}) , utilizando o método primal-dual barreira logarítmica.

Um código referente a tal método (BP_Interior), que faz parte de um pacote denominado *Atomizer*, foi implementado em Matlab e pode ser encontrado em [7]. No Capítulo 5, apresentaremos maiores detalhes da implementação.

Capítulo 3

Método de Pontos Interiores

Em 1984, um novo algoritmo para resolução de problemas de programação linear foi proposto. Este algoritmo apresentado por Karmarkar, despertou um grande interesse para esse tipo de problema, já que segundo ele, tal método resolveria problemas de programação linear de grande porte até cinquenta vezes mais rápido que o Método Simplex. Neste capítulo, definiremos alguns conceitos sobre problemas de otimização, apresentando os métodos de pontos interiores de nosso interesse.

3.1 Otimização Linear

Um problema de otimização linear consiste em minimizar ou maximizar uma função objetivo linear, sujeito a um conjunto finito de restrições lineares. As restrições podem ser de igualdade ou desigualdade. A forma padrão de um problema de otimização linear é denominada problema primal e é dada por:

$$\begin{aligned} &\text{minimizar} && c^T x \\ &\text{sujeito a} && Ax = b, \\ &&& x \geq 0 \end{aligned} \tag{3.1}$$

sendo A uma matriz de restrições pertencente a $\mathbb{R}^{m \times n}$, x um vetor coluna pertencente a \mathbb{R}^n , cujas componentes são denominadas variáveis primais, e b e c vetores coluna pertencentes a \mathbb{R}^m e \mathbb{R}^n , respectivamente, sendo c os custos associados aos elementos de x .

Definimos um vetor \bar{x} tal que $A\bar{x} = b$, $\bar{x} \geq 0$, como uma solução factível, e ao conjunto de todas as soluções factíveis damos o nome de conjunto factível. Uma solução x^* é ótima quando, além de ser uma solução factível, admite o menor valor possível para a função objetivo.

No caso das restrições serem inconsistentes, não teremos solução factível. Tal problema é chamado infactível. No caso em que existe solução factível, a função objetivo pode ser ilimitada ou limitada no domínio, sendo assim chamado de problema ilimitado e limitado, respectivamente. Não teremos solução ótima se e somente se o problema é infactível ou ilimitado.

Para um dado problema primal, sempre podemos construir um problema associado, ao qual chamamos problema dual, que consiste dos mesmos componentes dados, arranjos de uma forma

diferente. O problema dual de (3.1) é dado por:

$$\begin{array}{ll} \text{maximizar} & b^T y \\ \text{sujeito a} & A^T y \leq c, \\ & y \text{ livre} \end{array} \quad (3.2)$$

que é equivalente a

$$\begin{array}{ll} \text{maximizar} & b^T y \\ \text{sujeito a} & A^T y + z = c, \\ & z \geq 0, \\ & y \text{ livre} \end{array} \quad (3.3)$$

sendo y um vetor coluna pertencente a \mathbb{R}^m , denominado vetor de variáveis duais, e z um vetor coluna pertencente a \mathbb{R}^n , um vetor de variáveis de folga.

A seguir são dadas algumas relações entre os problemas primal e dual.

Lema 3.1 (Lema Fraco da Dualidade). Se x e y são soluções factíveis para os problemas primal e dual, respectivamente, então $c^T x \geq b^T y$.

Demonstração:

Sabemos que $Ax = b$, assim $b^T y = x^T A^T y \leq x^T c = c^T x$, pois $x \geq 0$, ou seja, a função objetivo do problema dual é um limitante inferior da função objetivo do problema primal.

Corolário 3.1.1. Se $c^T x = b^T y$ e x e y são factíveis, então x é solução ótima do problema primal e y é solução ótima do problema dual.

Teorema 3.1.1. Sejam x e (y, z) soluções factíveis dos problemas primal e dual, respectivamente. Uma condição necessária e suficiente para que ambas soluções sejam ótimas é que:

Se $x_j > 0$, então $z_j = 0$.

Se $z_j > 0$, então $x_j = 0$.

Ou seja, $x_j \times z_j = 0$, para j variando de 1 a n .

Combinando os resultados acima, determinamos as Condições de Otimalidade (condições algébricas que devem ser satisfeitas pelas soluções dos problemas de programação linear).

Condições de Otimalidade

Dado um ponto (x, y, z) , ele será ótimo para os problemas primal e dual se e somente se as seguintes condições forem satisfeitas:

$$\begin{cases} Ax = b \\ A^T y + z = c \\ XZe = 0 \\ (x, z) \geq 0 \end{cases}, \quad (3.4)$$

sendo X e Z matrizes diagonais formadas pelos elementos dos vetores x e z , respectivamente.

3.2 Otimização Não Linear

Seja um conjunto $\Omega \subset \mathbb{R}^n$ e sejam $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $h := (h_1; h_2; \dots; h_m)$ e $g : \mathbb{R}^n \rightarrow \mathbb{R}^p$, $g := (g_1; g_2; \dots; g_p)$, contínuas e pertencentes a C^2 . Um problema de otimização não linear geral tem a seguinte forma:

$$\begin{aligned} &\text{minimizar} && f(x) \\ &\text{sujeito a} && h(x) = 0 \\ &&& g(x) \leq 0 \text{ ,} \\ &&& x \in \Omega \end{aligned}$$

sendo $h(x) = 0$ as restrições de igualdade, $g(x) \leq 0$ as restrições de desigualdade e $x \in \Omega$ a restrição do conjunto Ω .

Nesse trabalho, consideramos apenas o problema com restrições de igualdade, ou seja:

$$\begin{aligned} &\text{minimizar} && f(x) \\ &\text{sujeito a} && h(x) = 0 \text{ .} \end{aligned} \tag{3.5}$$

Chamaremos de S o conjunto formado pelos vetores $x \in \mathbb{R}^n$ tais que $h(x) = \mathbf{0}$.

Definição 3.2.1. A função Lagrangeana $L : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ associada ao problema (3.5) é dada por:

$$L(x, y) := f(x) + y^T h(x) ,$$

onde $y \in \mathbb{R}^m$ é o vetor dos multiplicadores de Lagrange.

Definição 3.2.2 (Ponto Regular). Um ponto $x^* \in S$ é dito ponto regular se $\{\nabla h_1(x^*), \dots, \nabla h_m(x^*)\}$ é um conjunto linearmente independente.

As condições de otimalidade para um problema de otimização não linear com restrição de igualdade (3.5) são dadas por:

Teorema 3.2.1. (Condições necessárias de primeira ordem)

Sejam $f, g \in C^1$ e x^* um mínimo local de f satisfazendo as restrições de igualdade. Então existe um vetor $y \in \mathbb{R}^m$ tal que

$$\nabla f(x^*) + \sum_{i=1}^m y_i \nabla h_i(x^*) = \nabla f(x^*) + \nabla h(x^*)^T y = 0.$$

Tais condições podem ser obtidas derivando a função Lagrangeana com relação a x e y :

$$\nabla L(x, y) = \begin{cases} \nabla f(x^*) + \nabla h(x^*)^T y = 0 \\ h(x) = \mathbf{0} \\ x \in \mathbb{R}^n \end{cases} ,$$

Teorema 3.2.2. Seja x^0 um ponto regular. O subespaço tangente a este ponto é representado por:

$$\mathcal{M}(x^0) = \left\{ u \in \mathbb{R}^n \mid \nabla h(x^0)^T u = 0 \right\},$$

onde $\nabla h(x^0)$ é a matriz Jacobiana de h em x^0 .

Teorema 3.2.3. (Condições necessárias de segunda ordem)

Sejam $f, h \in C^2$ e x^* um ponto regular das restrições de igualdade $h(x) = 0$ que é um mínimo local da função f . Então existe um vetor $y \in \mathbb{R}^m$ tal que:

$$\nabla f(x^*) + \sum_{i=1}^m y_i \nabla h_i(x^*) = \mathbf{0},$$

e a matriz Hessiana da função Lagrangeana, $\nabla^2 L(x^*) = \nabla^2 f(x^*) + \sum_{i=1}^m y_i \nabla^2 h_i(x^*)$ é tal que $u^T \nabla^2 L(x^*) u \geq 0$ para todo $u \in \mathcal{M}(x^*)$, ou seja, $\nabla^2 L(x^*)$ é semi-definida positiva no conjunto considerado.

Teorema 3.2.4. (Condições suficientes de segunda ordem)

Sejam $f, h \in C^2$ e x^* um ponto regular do conjunto S tal que $\nabla f(x^*) + y^T \nabla h(x^*) = 0$. Suponha que $\nabla^2 L(x^*)$ seja definida positiva em \mathcal{M} , de modo que x^* satisfaz as condições necessárias de segunda ordem, então x^* é um mínimo local estrito de f em S .

As demonstrações das três condições enunciadas anteriormente podem ser encontradas em [13].

Convexidade

Sabendo que nem sempre é fácil determinar se um mínimo local é global, apresentaremos as definições de convexidade, bem como um teorema.

Definição 3.2.3. Um subconjunto $\Omega \subset \mathbb{R}^n$ é convexo se e somente se, para todo $x, y \in \Omega, \lambda \in [0, 1]$, temos que $\lambda x + (1 - \lambda)y \in \Omega$. Ou seja, para quaisquer dois pontos em Ω , o segmento que os une está contido em Ω .

Definição 3.2.4 (Funções Convexas). Dada uma função f definida em um conjunto convexo Ω , tal função é convexa se e somente se, para todo $x, y \in \Omega, \lambda \in [0, 1]$, verifica-se

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y).$$

Se para todo $\lambda \in (0, 1)$ e $x \neq y$ vale a desigualdade estrita, dizemos que f é estritamente convexa. Se $-f$ é convexa, f é côncava.

Teorema 3.2.5. Seja f uma função convexa definida em um conjunto convexo $\Omega \in \mathbb{R}^n, f : \Omega \rightarrow \mathbb{R}$, então qualquer mínimo local é também um mínimo global de f .

3.2.1 Método de Newton Aplicado às Condições de Otimalidade

Note que podemos reescrever as condições de otimalidade (3.4) para um problema de programação linear de uma forma ligeiramente diferente, definindo uma aplicação F de \mathbb{R}^{2n+m} a \mathbb{R}^{2n+m} :

$$F(x, y, z) = \begin{pmatrix} F_p \\ F_d \\ F_a \end{pmatrix} = \begin{pmatrix} A^T y - c + z \\ Ax - b \\ XZe \end{pmatrix} = 0,$$

supondo que $(x, z) \geq 0$.

Para resolver esse sistema não linear, aplicamos o Método de Newton às Condições de Otimalidade (a descrição do Método de Newton encontra-se no Apêndice B), aproximando F pela série de Taylor truncada e prosseguindo de forma análoga ao processo descrito no Apêndice B, referente ao Método de Newton para várias variáveis. Assim, obtemos:

$$\begin{aligned} F(x^{k+1}) &\approx F(x^k) + \nabla F(x^k)^T (x^{k+1} - x^k) = 0 \\ \Rightarrow -F(x^k) &= \nabla F(x^k)^T (x^{k+1} - x^k) \\ \Rightarrow x^{k+1} &= x^k - (\nabla F(x^k))^{-T} F(x^k) \\ \Rightarrow x^{k+1} &= x^k + d, d = -(\nabla F(x^k))^{-T} F(x^k). \end{aligned}$$

O método convergirá se as condições de convergência forem satisfeitas. Maiores detalhes encontram-se no Apêndice B.

3.3 Método Primal-Dual

3.3.1 Método Primal-Dual Afim-Escala

Os métodos de pontos interiores Primais-Duais encontram uma solução ótima (x^*, y^*, z^*) do problema de programação linear aplicando o Método de Newton às condições de otimalidade (desconsiderando as desigualdades: $x \geq 0, z \geq 0$) e modificando o tamanho dos passos das direções encontradas, fazendo com que sejam tais que $(x, z) \geq 0$ para todas as iterações. Assim, resolvemos os problemas primal e dual simultaneamente.

Iniciaremos o método com x^0, y^0 e z^0 , não sendo exigido que sejam factíveis, mas apenas que $(x^0, z^0) > 0$ ou seja, que (x^0, z^0) seja um ponto interior. As condições de otimalidade no ponto inicial serão dadas por $F(x^0, y^0, z^0)$, que podemos escrever como:

$$F(x^0, y^0, z^0) = \begin{pmatrix} Ax^0 - b \\ A^T y^0 + z^0 - c \\ X^0 Z^0 e \end{pmatrix} = - \begin{pmatrix} r_p^0 \\ r_d^0 \\ r_a^0 \end{pmatrix} = r(x_0, y_0, z_0),$$

onde r_p refere-se ao resíduo do problema primal, e r_d ao do dual. Agora, vamos aplicar o Método de Newton para várias variáveis às condições de otimalidade.

$$(x^1, y^1, z^1) = (x^0, y^0, z^0) - [J(x^0, y^0, z^0)]^{-1} F(x^0, y^0, z^0),$$

sendo que $J(x^0, y^0, z^0) = \begin{pmatrix} \nabla F_p^T \\ \nabla F_d^T \\ \nabla F_a^T \end{pmatrix} = \begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z^0 & 0 & X^0 \end{pmatrix}.$

Como visto na seção anterior, $d_0 = -[J(x^0, y^0, z^0)]^{-1} F(x^0, y^0, z^0) = [J(x^0, y^0, z^0)]^{-1} r(x^0, y^0, z^0):$

$$d^0 = \begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z^0 & 0 & X^0 \end{pmatrix}^{-1} \begin{pmatrix} r_p^0 \\ r_d^0 \\ r_a^0 \end{pmatrix} = \begin{pmatrix} \Delta x^0 \\ \Delta y^0 \\ \Delta z^0 \end{pmatrix}.$$

Tendo em vista que resolveremos tal sistema a cada iteração, desconsideremos o índice 0. Multiplicando ambos os lados da igualdade pela matriz Jacobiana, obtemos:

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} r_p \\ r_d \\ r_a \end{pmatrix},$$

o que nos dá o sistema:

$$A\Delta x = r_p, \tag{3.6}$$

$$A^T \Delta y + \Delta z = r_d, \tag{3.7}$$

$$Z\Delta x + X\Delta z = r_a. \tag{3.8}$$

Note que, de (3.8), obtemos $\Delta z = X^{-1}(r_a - Z\Delta x)$. Assim podemos reescrever a equação (3.7) como:

$$A^T \Delta y + X^{-1}(r_a - Z\Delta x) = r_d \Rightarrow A^T \Delta y - X^{-1}Z\Delta x = r_d - X^{-1}r_a.$$

Definindo $D = Z^{-1}X$, vamos substituir na equação anterior, obtendo:

$$A^T \Delta y - D^{-1}\Delta x = r_d - X^{-1}r_a.$$

Logo,

$$\Delta x = D(A^T \Delta y - r_d + X^{-1}r_a).$$

Substituindo a última equação em (3.6), chegamos a:

$$\Delta y = (ADA^T)^{-1}(r_p + AD r_d - ADX^{-1}r_a).$$

Dado que a matriz A tem posto m e que ADA^T tem dimensão $m \times m$, temos que esta é definida positiva. Assim podemos calcular a decomposição de Cholesky de ADA^T , resolvendo dois sistemas triangulares para obter Δy .

A seguir, apresentamos como é determinado o ponto inicial, quais são os critérios de convergência do método, e apresentamos um resumo do mesmo.

Ponto Inicial

O ponto inicial será determinado como em [18]. Para o problema primal:

$$\begin{aligned}\tilde{x} &= A^T (AA^T)^{-1} b \Rightarrow A\tilde{x} = b, \\ x_i^0 &= \max \{ \tilde{x}_i, \epsilon_1 \}, \\ \epsilon_1 &= \max \left\{ -\min \tilde{x}_i, \epsilon_2, \frac{\|b_1\|}{\epsilon_2 \|A\|_1} \right\}, \\ \epsilon_2 &= 100.\end{aligned}$$

Para o problema dual:

$$\begin{aligned}y^0 &= 0, \\ z_i^0 &= \begin{cases} c_i + \epsilon_3 & \text{se } c_i \geq 0; \\ -c_i & \text{se } c_i \leq -\epsilon_3; \\ \epsilon_3 & \text{se } -\epsilon_3 \leq c_i \leq 0, \end{cases} \\ \epsilon_3 &= 1 + \|c\|_1.\end{aligned}$$

Critério de Convergência

O algoritmo é interrompido quando as seguintes condições são satisfeitas:

- **Factibilidade Primal:** $\frac{\|b - Ax\|}{\|b\| + 1} \leq \epsilon;$
- **Factibilidade Dual:** $\frac{\|c - A^T y - z\|}{\|c\| + 1} \leq \epsilon;$
- **Otimidade:** $\frac{|c^T x - b^T y|}{1 + c^T x + b^T y} \leq \epsilon$ ou $\left| \frac{x^T z}{1 + c^T x + b^T y} \right| \leq \epsilon.$

Método Primal-Dual Afim-Escala

1. Ponto inicial (x^0, y^0, z^0) interior, $(x^0, z^0) > 0$, sem exigência de factibilidade e $\tau \in (0, 1)$.

2. Repita até convergir:

$$\begin{aligned}r_p^k &= b - Ax^k; \\ r_d^k &= c - A^T y^k - z^k; \\ r_a^k &= -X^k Z^k e; \\ \Delta y^k &= [AD^k A^T]^{-1} \left[r_p^k + AD^k r_d^k - AD^k (X^k)^{-1} r_a^k \right]; \\ \Delta x^k &= D^k \left[A^T \Delta y^k - r_d^k + (X^k)^{-1} r_a^k \right]; \\ \Delta z^k &= (X^k)^{-1} \left[r_a^k - Z^k \Delta x^k \right];\end{aligned}$$

$$\begin{aligned}
\rho_p &= \min_{\Delta x_i^k < 0} \left\{ -\frac{x_i^k}{\Delta x_i^k} \right\}; \\
\rho_d &= \min_{\Delta z_i^k < 0} \left\{ -\frac{z_i^k}{\Delta z_i^k} \right\}; \\
\alpha_p^k &= \min \{1, \tau \rho_p^k\}; \\
\alpha_d^k &= \min \{1, \tau \rho_d^k\}; \\
x^{k+1} &= x^k + \alpha_p^k \Delta x^k \left(\alpha_p^k \text{ é tal que } x^{k+1} > 0 \right); \\
y^{k+1} &= y^k + \alpha_d^k \Delta y^k; \\
z^{k+1} &= z^k + \alpha_d^k \Delta z^k \left(\alpha_d^k \text{ é tal que } z^{k+1} > 0 \right).
\end{aligned}$$

3.3.2 Método Primal-Dual Seguidor de Caminho

O método primal-dual afim escala, apesar de ser melhor que os métodos primal e dual de pontos interiores, pois não precisa de ponto inicial factível, não é eficiente, visto que permite que alguns produtos $x_i z_i$ aproximem de zero rapidamente, ou seja, aproximem da fronteira da região factível, fazendo com que as direções calculadas sejam distorcidas, com isso o método pode demorar a convergir, ou inclusive, não convergir.

A fim de eliminar tal problema, acrescentamos uma perturbação μ à condição de complementaridade. Assim, $XZe = \mu e$, e as novas condições de otimalidade são:

$$\begin{cases} Ax = b \\ A^T y + z = c \\ XZe = \mu e \\ (x, z) \geq 0 \end{cases},$$

onde μ é tal que $\lim_{k \rightarrow \infty} \mu^k = 0$.

Dependendo da escolha de τ e σ obtemos resultados teóricos e práticos com respeito a eficiência do método.

O valor estimado de μ^k , na maioria das implementações, é dado por:

$$\mu^k = \sigma^k \left(\frac{\gamma^k}{n} \right),$$

sendo $\gamma^k = (x^k)^T z^k$, e $\sigma^k \in (0, 1)$, o parâmetro de centragem.

Observe que quando $\sigma^k = 0$, temos o método afim-escala.

Quando $\sigma^k = 1 \Rightarrow \mu^k = \frac{\gamma^k}{n} \Rightarrow XZe = \frac{x^T z e}{n}$, ou seja, $x_j \times z_j$ tem o mesmo valor para todo j .

A direção assim obtida é definida como “*direção de centragem*”.

Agora, aplicando o método de Newton para as condições de otimalidade, obtemos o seguinte sistema linear:

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} r_p \\ r_d \\ r_c \end{pmatrix}.$$

E prosseguimos como no método primal-dual afim-escala para encontrar as direções. A diferença dos dois métodos são: a troca de r_a por r_c e o cálculo de μ .

Método Primal-Dual Seguidor de Caminho

1. Ponto inicial (x^0, y^0, z^0) interior, $(x^0, z^0) > 0$, sem exigência de factibilidade e $\tau, \sigma \in (0, 1)$.

2. Repita até convergir:

$$\begin{aligned} \mu^k &= \sigma \frac{\gamma^k}{n}; \\ r_p^k &= b - Ax^k; \\ r_d^k &= c - A^T y^k - z^k; \\ r_c^k &= \mu^k e - X^k Z^k e; \\ \Delta y^k &= [AD^k A^T]^{-1} \left[r_p^k + AD^k r_d^k - A (Z^k)^{-1} r_c^k \right]; \\ \Delta x^k &= D^k \left[A^T \Delta y^k - r_d^k + (X^k)^{-1} r_c^k \right]; \\ \Delta z^k &= (X)^{-1} \left[r_c^k - Z^k \Delta x^k \right]; \\ \rho_p &= \min_{\Delta x_i^k < 0} \left\{ -\frac{x_i^k}{\Delta x_i^k} \right\}; \\ \rho_d &= \min_{\Delta z_i^k < 0} \left\{ -\frac{z_i^k}{\Delta z_i^k} \right\}; \\ \alpha_p^k &= \min \{1, \tau \rho_p^k\}; \\ \alpha_d^k &= \min \{1, \tau \rho_d^k\}; \\ x^{k+1} &= x^k + \alpha_p^k \Delta x^k \left(\alpha_p^k \text{ é tal que } x^{k+1} > 0 \right); \\ y^{k+1} &= y^k + \alpha_d^k \Delta y^k; \\ z^{k+1} &= z^k + \alpha_d^k \Delta z^k \left(\alpha_d^k \text{ é tal que } z^{k+1} > 0 \right). \end{aligned}$$

3.4 Método de Pontos Interiores Barreira Logarítmica

Descreveremos nessa seção o método de pontos interiores barreira logarítmica o qual é utilizado no método proposto por Chen [6] para solucionar o problema *Basis Pursuit*, assim como também é utilizado no desenvolvimento dos nossos métodos.

Considere o problema de programação não linear:

$$\begin{aligned} &\text{minimizar} && f(x) \\ &\text{sujeito a} && h(x) = 0 \text{ .} \\ &&& g(x) \leq 0 \end{aligned} \tag{3.9}$$

A fim de trabalharmos apenas com restrições de igualdade, vamos inserir a varável de folga $s \geq 0$. Dessa forma, (3.9) pode ser escrito como:

$$\begin{array}{ll} \text{minimizar} & f(x) \\ \text{sujeito a} & h(x) = 0 \\ & g(x) + s = 0 \\ & s \geq 0 \end{array} .$$

Agora, vamos encontrar uma aproximação do problema anterior por meio da função barreira logarítmica. Obtemos dessa forma:

$$\begin{array}{ll} \text{minimizar} & f(x) - \mu \sum_{i=1}^n \ln(s_i) \\ \text{sujeito a} & h(x) = 0 \\ & g(x) + s = 0 \end{array} . \quad (3.10)$$

Note que o domínio deste problema é o conjunto de pontos factíveis de (3.9), onde μ é o parâmetro de barreira, com μ tendendo a zero.

Considere o problema de programação não linear (3.9), com a relaxação da restrição $s \geq 0$ (3.10). A função Lagrangeana associada a tal problema, é dada por:

$$L = f(x) - \mu \sum_{i=1}^n \ln(s_i) + \pi^T (g(x) + s) + \lambda^T h(x),$$

sendo π e λ os vetores dos multiplicadores de Lagrange.

Como podemos ver em [4], para resolver esse problema aplicando as condições necessárias de primeira ordem e utilizando o método de Newton, chegamos a:

$$\begin{pmatrix} \nabla^2 f(x) + \pi^T \nabla^2 g(x) + \lambda^T \nabla^2 h(x) & \nabla g(x) & \nabla h(x) & 0 \\ \nabla g(x) & 0 & 0 & I \\ \nabla h(x) & 0 & 0 & 0 \\ 0 & S & 0 & \pi \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \pi \\ \Delta \lambda \\ \Delta s \end{pmatrix} =$$

$$= - \begin{pmatrix} \nabla f(x) + \pi^T \nabla g(x) + \lambda^T \nabla h(x) \\ g(x) + s \\ h(x) \\ -\mu e + S\pi \end{pmatrix},$$

sendo S a matriz diagonal formada pelos elementos do vetor s .

Dessa forma, determinamos as direções Δx , $\Delta \pi$, $\Delta \lambda$ e Δs .

Assim, depois de calcular o vetor gradiente da função Lagrangeana L , calcular a matriz Hessiana e resolver o sistema anterior determinando as direções. Vamos também calcular o tamanho de passo primal e dual α para que s e π mantenham-se estritamente positivas. Dessa forma, o tamanho de passo α será dado por:

$$\alpha = \min \{1, \tau \rho_p, \tau \rho_d\},$$

$$\text{onde } \rho_p = \min_{\Delta s_i < 0} \left\{ -\frac{s_i}{\Delta s_i} \right\} \text{ e } \rho_d = \min_{\Delta \pi_i < 0} \left\{ -\frac{\pi_i}{\Delta \pi_i} \right\}.$$

Prosseguimos atualizando as variáveis, sendo que a atualização do parâmetro barreira é dada por:

$$\mu^{k+1} = \frac{\mu^k}{\beta}, \text{ onde } \beta > 1.$$

Um resumo do método e maiores detalhes podem ser encontrados em [4] e [20]. Um exemplo numérico é dado em [11].

3.5 Método de Pontos Interiores Barreira Logarítmica Preditor-Corretor

Em nosso trabalho aplicaremos o Método Barreira Logarítmica Preditor-Corretor ao problema *Basis Pursuit* a fim de buscar uma maior eficiência no programa BP_Interior, contido em um pacote disponível em [7] para o Matlab. Tal método é baseado em três componentes:

- Direção afim-escala, que corresponde ao passo preditor, que consiste em encontrar uma direção do problema de otimização não-linear barreira logarítmica, que em um primeiro momento consideramos sem perturbação (ou seja, o parâmetro de barreira μ é igual a zero).
- Direção de centragem, definida pelo σ que foi definido no método primal-dual seguidor de caminho, recordando que a direção de centragem evita que as soluções ao longo das iterações se aproximem das arestas do politopo dado pelas restrições, mantendo dessa forma, o produto $x_i z_i$ estritamente positivo. Essa direção de centragem permite-nos tomar passos maiores da direção de Newton, pois a condição de positividade somente é violada com um tamanho de passo maior do que quando consideramos μ igual a zero, visto que obtemos uma direção voltada ao interior do politopo.
- Direção de correção, que corresponde ao passo corretor, em que calculamos a correção não linear, tentando compensar a aproximação linear do Método de Newton.

Para uma maior compreensão deste passo, considere o exemplo dado em [11]. Dado o problema

$$\begin{aligned} &\text{minimizar} && x^2 \\ &\text{sujeito a} && y = 1 + x, \\ &&& x \geq 0 \end{aligned} \tag{3.11}$$

definimos o Lagrangeano do problema com barreira logarítmica como

$$L(x, y, \lambda) = x^2 - \mu \ln(x) + \lambda(y - 1 - x).$$

As condições de otimalidade associadas ao problema são dadas por:

$$\nabla L(x, y, \lambda) = \begin{pmatrix} 2x^2 - \lambda x \\ \lambda \\ y - 1 - x \end{pmatrix} = \begin{pmatrix} \mu \\ 0 \\ 0 \end{pmatrix}. \tag{3.12}$$

Quando aplicamos o Método de Newton ao sistema anterior, aproximamos uma equação polinomial do segundo grau a uma equação polinomial do primeiro grau (referindo à primeira equação do sistema (3.12)). Assim, nosso objetivo nessa etapa será compensar tal aproximação. No exemplo anterior, o Método de Newton aplicado à primeira equação nos dará a seguinte equação:

$$(4x - \lambda) \Delta x - x \Delta \lambda = -2x^2 + \lambda x + \mu.$$

Assim, vamos calcular o resíduo de $2x^2 - \lambda x - \mu$ para o ponto $(x + \Delta x)$ e $(\lambda + \Delta \lambda)$ que é a raiz da função linear $R(\bar{x}, \bar{\lambda}) = -(4x - \lambda) \Delta x + x \Delta \lambda - 2x^2 + \lambda x + \mu$, sendo x e λ os valores definidos no passo anterior, $\Delta x = \bar{x} - x$ e $\Delta \lambda = \bar{\lambda} - \lambda$. Ou seja, temos os seguintes dados:

$$(4x - \lambda) \Delta x - x \Delta \lambda = -2x^2 + \lambda x + \mu \quad (3.13)$$

e

$$2(x + \Delta x)^2 - (\lambda + \Delta \lambda)(x + \Delta x) - \mu = 4x\Delta x - \lambda\Delta x - x\Delta\lambda + 2x^2 - \lambda x - \mu + 2\Delta x^2 - \Delta x\Delta\lambda. \quad (3.14)$$

Da equação (3.13) concluímos que os termos $4x\Delta x - \lambda\Delta x - x\Delta\lambda + 2x^2 - \lambda x - \mu$ da segunda equação é igual a zero, o que nos dá como resíduo: $2\Delta x^2 - \Delta x\Delta\lambda$.

Resumidamente, o método preditor-corretor consiste em aplicar o método de Newton duas vezes, utilizando a mesma Hessiana. Diferente do método barreira logarítmica, que desconsidera o resíduo dos termos não lineares e sua aproximação linear pelo Método de Newton, neste método vamos introduzir as correções destas equações.

No exemplo anterior, a equação não linear é dada por $2x^2 - \lambda x$, e o resíduo é dado por $2\Delta x^2 - \Delta x\Delta\lambda$.

No caso de um problema de programação linear na forma padrão, o termo não linear é dado por Xz , e o resíduo é dado por $\Delta X \Delta Z e$, sendo ΔX e ΔZ matrizes diagonais cujos elementos são as entradas dos vetores Δx e Δz , respectivamente.

Então, primeiramente resolvemos o sistema buscando o passo preditor, considerando μ igual a zero; em seguida, no passo corretor, consideramos os dados obtidos no passo afim, resolvendo o segundo sistema com a correção não linear e com perturbação.

Dessa forma, para o problema (3.9), primeiramente resolvemos o sistema para μ igual a zero:

$$\begin{pmatrix} \nabla^2 f(x) + \pi^T \nabla^2 g(x) + \lambda^T \nabla^2 h(x) & \nabla g(x) & \nabla h(x) & 0 \\ \nabla g(x) & 0 & 0 & I \\ \nabla h(x) & 0 & 0 & 0 \\ 0 & S & 0 & \pi \end{pmatrix} \begin{pmatrix} \bar{\Delta x} \\ \bar{\Delta \pi} \\ \bar{\Delta \lambda} \\ \bar{\Delta s} \end{pmatrix} =$$

$$= - \begin{pmatrix} \nabla f(x) + \pi^T \nabla g(x) + \lambda^T \nabla h(x) \\ g(x) + s \\ h(x) \\ S\pi \end{pmatrix}.$$

E, com as direções preditoras, resolvemos o segundo sistema:

$$\begin{pmatrix} \nabla^2 f(x) + \pi^T \nabla^2 g(x) + \lambda^T \nabla^2 h(x) & \nabla g(x) & \nabla h(x) & 0 \\ \nabla g(x) & 0 & 0 & I \\ \nabla h(x) & 0 & 0 & 0 \\ 0 & S & 0 & \pi \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \pi \\ \Delta \lambda \\ \Delta s \end{pmatrix} =$$

$$= - \begin{pmatrix} \nabla f(x) + \pi^T \nabla g(x) + \lambda^T \nabla h(x) \\ g(x) + s \\ h(x) \\ -\mu e + S\pi + \overline{\Delta S \Delta \Pi} e \end{pmatrix}.$$

Como mencionado em [4], apesar do método barreira logarítmica preditor-corretor exigir que se resolvam dois sistemas lineares em cada iteração, o número de iterações em relação ao método barreira logarítmica é reduzido. Como a Hessiana é a mesma nos dois sistemas, os cálculos utilizados para a decomposição são feitos uma única vez.

O tamanho de passo α e as atualizações das variáveis são equivalentes às do método barreira logarítmica, que vimos na seção anterior.

Um resumo do método encontra-se em [4], um exemplo numérico pode ser encontrado em [11]. Para maiores detalhes, ver [24].

Capítulo 4

Método Primal-Dual Barreira Logarítmica Aplicado a BP e BPDN

Neste capítulo, vamos aplicar o Método Primal-Dual Barreira Logarítmica a BP, mas antes reescreveremos o problema como um problema linear perturbado. Veremos que, com essa modificação, o mesmo método aplicado para a resolução de BP pode ser adaptado para a resolução do problema BPDN.

4.1 Formulação do Problema BP

Em [10] é discutido o método de barreira para a resolução de programas lineares e quadráticos (QP), expressos na forma padrão como:

$$\begin{aligned} &\text{minimizar} && c^T x + \frac{1}{2} x^T Q x \\ &\text{sujeito a} && Ax = b \\ &&& l \leq x \leq u \end{aligned} \quad (4.1)$$

onde A é uma matriz $m \times n$ ($m \leq n$) e Q é uma matriz simétrica semidefinida positiva. Note que, se $Q = 0$, (4.1) é um problema de programação linear (LP).

No artigo em questão, é descrito um algoritmo Primal-Dual QP para resolver (4.1). Para tal resolução, os autores reapresentam o problema da seguinte forma:

$$\begin{aligned} &\text{minimizar} && c^T x + \frac{1}{2} x^T Q x + \frac{1}{2} \|\gamma x\|_2^2 + \frac{1}{2} \|p\|_2^2 \\ &\text{sujeito a} && Ax + \delta p = b \\ &&& x - s_1 = l \\ &&& x + s_2 = u \end{aligned} \quad (4.2)$$

onde $s_1 \geq 0$, $s_2 \geq 0$ e os escalares γ e δ são parâmetros de regularização “pequenos”. Denota-se por π , z e $-y$ as variáveis duais associadas às três equações de restrições. Na solução, z e y são não negativos.

O termo $\frac{1}{2} \|\gamma x\|_2^2$ é incluído para garantir que $\|\gamma x^*\|$ seja limitado (sendo x^* uma solução ótima), e o termo δp permite que $Ax = b$ seja satisfeito no sentido de mínimos quadrados, caso a restrição não tenha solução factível (podemos associar ao caso do sinal ser perturbado).

Para mais detalhes desse problema, ver [10].

4.2 Aplicação do Método à BP

Em nosso trabalho, o objetivo principal é a busca da representação do sinal s através de *Basis Pursuit*. Dessa forma, como queremos a representação exata do sinal, sendo ele perturbado ou não, vamos resolver o problema de programação linear (3.1), que vimos ser equivalente a (2.3). Para resolver tal problema, vamos fazer uso da formulação proposta por Gill, Murray e Saunders [10], reescrevendo o problema como o seguinte programa linear perturbado:

$$\begin{aligned} &\text{minimizar} && c^T x + \frac{1}{2} \|\gamma x\|_2^2 + \frac{1}{2} \|p\|_2^2 \\ &\text{sujeito a} && Ax + \delta p = b \\ &&& x \geq 0 \end{aligned} \quad , \quad (4.3)$$

considerando os valores dos parâmetros γ e δ pequenos (10^{-4}); e resolvendo-o assim como feito em [6].

Associamos ao problema linear perturbado (4.3) o subproblema barreira logarítmica:

$$\begin{aligned} &\text{minimizar} && c^T x + \frac{1}{2} \|\gamma x\|_2^2 + \frac{1}{2} \|p\|_2^2 - \mu \sum_{i=1}^m \ln(x_i) \\ &\text{sujeito a} && Ax + \delta p = b \end{aligned}$$

A restrição de desigualdade $x \geq 0$ fica implícita. Como $\mu \rightarrow 0$ a solução converge para a solução do problema linear perturbado.

Definimos o Lagrangeano como:

$$L(x, y, p) = c^T x + \frac{1}{2} \|\gamma x\|_2^2 + \frac{1}{2} \|p\|_2^2 - \mu \sum_{i=1}^m \ln(x_i) + y^T (b - Ax - \delta p).$$

Agora vamos calcular as condições necessárias de primeira ordem do problema de minimização da função Lagrangeana:

$$\text{minimizar} \quad c^T x + \frac{1}{2} \|\gamma x\|_2^2 + \frac{1}{2} \|p\|_2^2 - \mu \sum_{i=1}^m \ln(x_i) + y^T (b - Ax - \delta p) ,$$

obtendo,

$$-\nabla_x L = A^T y - \gamma^2 x - c + \mu X^{-1} e = 0,$$

onde e é o vetor de dimensão apropriada, cujos elementos são compostos por 1.

No artigo de McShane, Monma e Shanno [17] vemos que quando o parâmetro μ do problema primal e do dual são iguais, $z = \mu X^{-1} e$. Assim, as condições necessárias de primeira ordem são dadas por:

$$\begin{aligned}
-\nabla_x L &= A^T y - \gamma^2 x - c + z = 0 \\
-\nabla_p L &= \delta y - p = 0 \Rightarrow p = \delta y \\
-\nabla_y L &= Ax + \delta p - b = Ax + \delta^2 y - b = 0
\end{aligned}
,$$

e por [17] novamente:

$$ZXe - \mu e = 0.$$

Rearranjando as expressões, obtemos

$$F(x, y, z) = \begin{pmatrix} A^T y - \gamma^2 x - c + z \\ Ax + \delta^2 y - b \\ ZXe - \mu e \end{pmatrix} = \mathbf{0},$$

onde z é um vetor dual, e X e Z são as matrizes diagonais formadas pelos elementos dos vetores x e z , respectivamente.

Apliquemos o Método de Newton a fim de encontrar a solução do sistema não linear acima. As direções de Newton $(\Delta x, \Delta y, \Delta z)$ devem satisfazer:

$$F(x, y, z) + (\Delta x, \Delta y, \Delta z) \begin{pmatrix} \frac{\partial F}{\partial x}(x, y, z) \\ \frac{\partial F}{\partial y}(x, y, z) \\ \frac{\partial F}{\partial z}(x, y, z) \end{pmatrix} = \mathbf{0},$$

o que corresponde a:

$$\begin{pmatrix} -\gamma^2 \Delta x + A^T \Delta y + \Delta z \\ A \Delta x + \delta^2 \Delta y \\ Z \Delta x + X \Delta z \end{pmatrix} = \begin{pmatrix} c + \gamma^2 x - z - A^T y \\ b - Ax - \delta^2 y \\ \mu e - Zx \end{pmatrix} = \begin{pmatrix} t \\ r \\ v \end{pmatrix},$$

que fica reduzido ao seguinte sistema:

$$(ADA^T + \delta^2 I) \Delta y = r - AD(X^{-1}v - t), \quad (4.4)$$

$$\Delta x = DA^T \Delta y + D(X^{-1}v - t), \quad (4.5)$$

$$\Delta z = X^{-1}v - X^{-1}Z \Delta x. \quad (4.6)$$

sendo $D = (X^{-1}Z + \gamma^2 I)^{-1}$. Para o cálculo do tamanho do passo (ρ_p, ρ_d) , vamos sempre escolher o maior possível, contanto que o ponto $x^{(k+1)}$ e $z^{(k+1)}$ sejam pontos interiores. A cada passo de Newton vamos decrescer o parâmetro de barreira μ monotonicamente e mais rapidamente se largos passos são tomados:

$$\mu \leftarrow (1 - \min(\rho_p, \rho_d, 0, 99)) \bar{\mu}. \quad (4.7)$$

sendo $\bar{\mu}$ o valor de μ do passo anterior.

O método converge quando $\|b - Ax - \delta^2 y\|_2$, $\|c + \gamma^2 x - z - A^T y\|_2$ e $z^T x$ são suficientemente pequenos.

Algoritmo (como descrito por Chen, Donoho e Saunders em [8]) :

1- Conjunto de parâmetros:

- Tolerância de factibilidade: FeaTol,
- Tolerância *gap* de dualidade: PDGapTol,
- Dois parâmetros de perturbação: γ e δ .

2- Inicialize x , y , z e μ , $k = 0$.

3- Loop.

- a) Resolva (4.4) para Δy , calcule Δx por (4.5) e Δz por (4.6).
- b) Calcule o comprimento dos passos primal e dual ρ_p , ρ_d :

$$\rho_p = 0,99 \times \max \{ \rho : x + \rho \Delta x \geq 0, \rho \leq 1 \};$$

$$\rho_d = 0,99 \times \max \{ \rho : z + \rho \Delta z \geq 0, \rho \leq 1 \}.$$

c) Atualize as variáveis:

$$x \leftarrow x + \rho_p \Delta x;$$

$$y \leftarrow y + \rho_d \Delta y;$$

$$z \leftarrow z + \rho_d \Delta z;$$

$$\mu \leftarrow (1 - \min(\rho_p, \rho_d, .99)) \bar{\mu}.$$

d) $k \leftarrow k + 1$.

4) Termine se as três condições seguintes são satisfeitas:

- Infactibilidade Primal = $\frac{\|b - Ax - \delta^2 y\|_2}{1 + \|x\|_2} < \text{FeaTol};$
- Infactibilidade Dual = $\frac{\|c + \gamma^2 x - z - A^T y\|_2}{1 + \|y\|_2} < \text{FeaTol};$

- $\text{Gap de Dualidade} = \frac{z^T x}{1 + \|x\|_2} < \text{PDGapTol}.$

Na resolução de sistemas do tipo:

$$Ax = b, \quad (4.8)$$

sendo A uma matriz simétrica e definida-positiva, frequentemente utilizamos a fatoração de Cholesky e determinamos o valor da variável x resolvendo dois sistemas triangulares. Como a fatoração de Cholesky tem complexidade de $O(n^3)$, e tendo em vista que estamos interessados em matrizes A de grande dimensão, essa fatoração acaba não sendo conveniente.

Em nosso trabalho, consideramos a existência de um “algoritmo implícito rápido”, o que significa que, para vetores arbitrários x , as operações Ax e $A^T x$ são calculadas em $O(n \log(n))$ ou $O(n)$, sem mesmo armazenar as matrizes A e A^T , e utilizando propriedades especiais da matriz para acelerar o cálculo da operação. Dessa forma, temos maior eficiência usando métodos iterativos, sendo o mais popular o Método dos Gradientes Conjugados (ver Apêndice A).

4.3 Basis Pursuit Denoising

O problema *Basis Pursuit Denoising* (BPDN) é uma adaptação de *Basis Pursuit* (BP) no caso de dados perturbados, cujo objetivo não seja a representação exata dos mesmos. Como já foi discutido, podemos considerar s (sinal puro), z (ruído branco gaussiano padrão) e σ (nível do ruído), escrevendo o sinal perturbado como:

$$y = s + \sigma z.$$

Basis Pursuit Denoising refere-se a solução de:

$$\text{minimizar } \frac{1}{2} \|y - \Phi\alpha\|_2^2 + \lambda \|\alpha\|_1.$$

Note que a solução $\alpha^{(\lambda)}$ é uma função do parâmetro λ . Dessa forma, o resíduo $r^\lambda = y - \Phi\alpha$ também é função de λ . O sinal $s^{(\lambda)}$ é representado por $\Phi\alpha^{(\lambda)}$, ou seja, $\Phi\alpha^{(\lambda)} = s^{(\lambda)}$. Quando λ tender a zero, o resíduo irá para zero e teremos a solução de BP aplicado a y , uma vez que $\Phi\alpha = y$. Quando λ tender a infinito, iremos obter $r^{(\lambda)}$ tendendo a y , já que os elementos de α serão muito pequenos e $s^{(\lambda)}$ tende a zero.

Assumindo que o dicionário em questão é normalizado, isto é, $\|\Phi_\gamma\|_2 = 1, \forall \gamma$, definimos o valor de λ como:

$$\lambda_p = \sigma \sqrt{2 \log(p)},$$

onde p é a cardinalidade do dicionário. A motivação para tal escolha pode ser encontrada em [8].

Já vimos que BPDN é equivalente a resolver o problema linear perturbado:

$$\begin{aligned} &\text{minimizar} && c^T x + \frac{1}{2} \|p\|_2^2 \\ &\text{sujeito a} && Ax + \delta p = b \\ &&& x \geq 0 \\ &&& \delta = 1 \end{aligned} \quad (4.9)$$

Como podemos notar, (4.3) com $\delta = 1$, resolve o problema (4.9).

O Método Primal-Dual Barreira Logarítmica com gradientes conjugados implementado por Chen, pode resolver aproximadamente todos os problemas de otimização BP e BPDN, em um tempo quase-linear [6].

Assim, o objetivo de considerar o problema linear perturbado (4.3) deve-se ao fato de, como mencionado por Chen [6], além de obter equações bem condicionadas para resolver o problema *Basis Pursuit*, o problema *Basis Pursuit Denoising* pode ser reformulado como um problema linear perturbado, fazendo assim com que a mesma implementação trabalhe para os dois tipos de problema.

Até agora, apresentamos os conceitos básicos do Método Primal-Dual Barreira Logarítmica aplicado ao problema não-linear (4.3). O programa BP_Interior desenvolvido por Chen, contido em *Atomizer* [7], resolve o problema *Basis Pursuit* (2.3) seguindo o mesmo procedimento descrito no Método Primal-Dual Barreira Logarítmica. Na seção seguinte, vamos apresentá-lo com maiores detalhes.

4.4 BP_Interior

O objetivo é resolver o problema de minimização (2.3) por meio do problema equivalente (4.3). Portanto, vamos fazer as seguintes associações:

$$\begin{aligned} \alpha &\iff u - v \\ x &\iff (u; v) \\ c &\iff (e; e) \\ A &\iff (\Phi, -\Phi) \\ b &\iff s \\ m &\iff 2p \end{aligned} \tag{4.10}$$

Dado que o sinal s possui algum tipo de perturbação, assumimos que o nível de ruído Gaussiano branco padrão é dado por 1.

O método inicializará pela solução do Método de *Frames* (MOF), ou seja, por encontrar um α tal que $\Phi\alpha = s$ e $\|\alpha\|_2$ seja mínimo. O programa MOF, assim como BP_Interior, encontra-se em *Atomizer* [7].

Inicializa-se com

$$\mu_0 = 0,01 \frac{\|f\|}{\sqrt{2m}},$$

onde f é o vetor cujas entradas são dadas pelas normas dos resíduos:

$$\begin{aligned} &\|b - Ax - \delta^2 y\|_2, \\ &\|c + \gamma^2 x - z - A^T y\|_2, \\ &z^T x. \end{aligned}$$

Os valores μ das outras iterações são calculados como descrito anteriormente (4.7).

Através do Método dos Gradientes Conjugados determinamos as direções do Método de Newton, como já comentado.

Por fim, projeta-se $dx = x - x_0$ no espaço nulo de Φ por gradientes conjugados a fim de garantir factibilidade.

No próximo capítulo, apresentaremos modificações no Método Primal-Dual Barreira Logarítmica, utilizado em BP_Interior, com o objetivo de melhorar a eficiência computacional.

Capítulo 5

Desenvolvimento dos Métodos Propostos

Como mencionado anteriormente, estamos interessados em dicionários que são *overcomplete*, já que a não-unicidade da solução do sistema nos dá a possibilidade de escolher a melhor representação para nossos propósitos. Teremos como objetivo, alcançar simultaneamente:

- Esparsidade: visto que queremos obter poucos coeficientes significativos, a fim de economizar memória;
- Super-resolução: obtemos uma resolução com objetos esparsos, que é melhor se comparada com abordagens tradicionais não adaptativas, ou seja, que não buscam uma maior eficiência utilizando o fato que os objetos em questão são esparsos;
- Estabilidade: ou seja, pequenas perturbações no sinal puro não devem causar grandes alterações;
- Rapidez.

Dessa forma, vamos modificar o Método Primal-Dual Barreira Logarítmica, do programa BP_Interior, visando obter uma melhor eficiência. Aplicaremos o Método Barreira Logarítmica, o Método Primal-Dual Barreira Logarítmica Preditor-Corretor, e uma variação deste último a (4.3).

5.1 Método Barreira Logarítmica

Consideremos novamente o problema (4.3). Associamos a ele o subproblema barreira logarítmica:

$$\begin{aligned} \text{minimizar} \quad & c^T x + \frac{1}{2} \|\gamma x\|^2 + \frac{1}{2} \|p\|^2 - \mu \sum_{i=1}^m \ln(x_i) \\ \text{sujeito a} \quad & Ax + \delta p = b \end{aligned}$$

Definimos o Lagrangeano como:

$$L(x, y, p) = c^T x + \frac{1}{2} \|\gamma x\|^2 + \frac{1}{2} \|p\|^2 - \mu \sum_{i=1}^m \ln(x_i) + y^T (b - Ax - \delta p).$$

O processo até aqui é equivalente ao do método anterior.

Agora, calculando as condições necessárias de primeira ordem do problema de minimização da função Lagrangeana:

$$\text{minimizar } c^T x + \frac{1}{2} \|\gamma x\|^2 + \frac{1}{2} \|p\|^2 - \mu \sum_{i=1}^m \ln(x_i) + y^T (b - Ax - \delta p),$$

obtemos,

$$\nabla_x L = A^T y - \gamma^2 x - c + \mu X^{-1} e = \mathbf{0},$$

onde e é o vetor de dimensão apropriada, cujos elementos são compostos por 1,

$$\nabla_p L = \delta y - p = 0 \Rightarrow p = \delta y$$

$$\nabla_y L = Ax + \delta p - b = Ax + \delta^2 y - b = \mathbf{0}$$

Rearranjando as expressões e fazendo a devida substituição, obtemos

$$F(x, y) = \begin{pmatrix} A^T y - \gamma^2 x - c + \mu X^{-1} e \\ Ax + \delta^2 y - b \end{pmatrix} = \mathbf{0},$$

onde X é a matriz diagonal formada pelos elementos do vetor x .

Aplicamos o Método de Newton a fim de encontrar a solução do sistema linear acima. As direções de Newton $(\Delta x, \Delta y)$ devem satisfazer:

$$F(x, y) + (\Delta x, \Delta y) \begin{pmatrix} \frac{\partial F}{\partial x}(x, y) \\ \frac{\partial F}{\partial y}(x, y) \end{pmatrix} = \mathbf{0},$$

o que corresponde a:

$$\begin{pmatrix} (-\gamma^2 - \mu X^{-2}) \Delta x + A^T \Delta y \\ A \Delta x + \delta^2 \Delta y \end{pmatrix} = \begin{pmatrix} c + \gamma^2 x - \mu X^{-1} e - A^T y \\ b - Ax - \delta^2 y \end{pmatrix} = \begin{pmatrix} t \\ r \end{pmatrix},$$

que fica reduzido ao seguinte sistema:

$$\Delta y = \delta^{-2} (r - A \Delta x), \quad (5.1)$$

$$\Delta x = (-\gamma^{-2} - \mu X^{-2})^{-1} (t - A^T \Delta y). \quad (5.2)$$

Vamos considerar $D = (-\gamma^2 - \mu X^{-2})^{-1}$.

Substituindo (5.2) em (5.1) chegamos a:

$$(I - \delta^{-2} A D A^T) \Delta y = \delta^{-2} (r - (A D t)). \quad (5.3)$$

O comprimento do passo de x e y é dado apenas por (ρ_p) , uma vez que a variável y é livre. Escolhemos o maior passo possível, contanto que o ponto $x^{(k+1)}$ seja interior. A cada passo de Newton decrescemos o parâmetro de barreira μ monotonicamente e mais rapidamente se largos passos são tomados:

$$\mu \leftarrow (1 - \min(\rho_p, 0, 99)) \bar{\mu}.$$

sendo $\bar{\mu}$ o valor de μ do passo anterior.

O método converge quando $\|b - Ax - \delta^2 y\|_2$, $\|c + \gamma^2 x - z - A^T y\|_2$ ou a diferença relativa do valor da função objetivo do passo atual com o passo anterior é pequeno (menor que 10^{-4}), em mais de três iterações.

Algoritmo :

1- Conjunto de parâmetros:

- Tolerância de factibilidade: FeaTol,
- Tolerância *gap* de dualidade: PDGapTol,
- Dois parâmetros de perturbação: γ e δ .

2- Inicialize x , y , e μ , $k = 0$.

3- Loop.

- a) Resolva (5.3) para Δy pelo Método dos Gradientes Conjugados, e calcule Δx por (5.2).
- b) Calcule o comprimento do passo primal ρ_p :

$$\rho_p = 0,99 \times \max \{ \rho : x + \rho \Delta x \geq 0, \rho \leq 1 \}.$$

c)Atualize as variáveis:

$$x \leftarrow x + \rho_p \Delta x;$$

$$y \leftarrow y + \rho_p \Delta y.$$

d) $k \leftarrow k + 1$.

4) Termine se as condições seguintes são satisfeitas:

- Infactibilidade Primal = $\frac{\|b - Ax - \delta^2 y\|_2}{1 + \|x\|_2} < \text{Featol}$;
- Infactibilidade Dual = $\frac{\|c + \gamma^2 x - \mu X^{-1} e - A^T y\|_2}{1 + \|y\|_2} < \text{Featol}$.

ou se:

- a diferença relativa do valor da função objetivo do passo atual com o passo anterior é menor que 10^{-4} em mais de três iterações.

5.2 Método Primal-Dual Barreira Logarítmica Preditor-Corretor

Na Seção 2.6, intitulada Método de Pontos Interiores Barreira Logarítmica Preditor-Corretor, introduzimos o conceito das três componentes: Direção Afim-Escala, Direção de Centragem e Direção de Correção, a fim de melhorar o Método Primal-Dual Barreira Logarítmica. Tendo em vista esses conceitos, e em busca de melhorar o desempenho de BP_Interior, façamos as correções dos termos não lineares do método descrito na seção anterior. Note que o termo não linear corresponde à condição de complementaridade. Assim, no primeiro passo consideraremos v tal que $v = -Zx$, e no segundo passo $v = \mu e - Zx - \Delta Z \Delta X e$, onde ΔZ e ΔX são as matrizes diagonais cujos elementos são as entradas dos vetores Δz e Δx respectivamente, que foram obtidos no primeiro passo.

O algoritmo é dado por:

1- Conjunto de parâmetros:

- Tolerância de factibilidade: FeaTol,
- Tolerância *gap* de dualidade: PDGapTol,
- Dois parâmetros de perturbação: γ e δ .

2- Inicialize x , y , z e μ , $k = 0$.

3- Loop.

a) Determine os dois passos seguintes:

1º Passo: Considere $v = -Zx$ e resolva (4.4) para Δy , calcule Δx por (4.5) e Δz por (4.6).

2º Passo: Considere $v = \mu - Zx - \Delta Z \Delta X e$, sendo ΔZ e ΔX matrizes diagonais formadas pelos elementos dos vetores Δz e Δx respectivamente, que foram obtidos no passo anterior, e resolva (4.4) para Δy , calcule Δx por (4.5) e Δz por (4.6) para obter novas direções.

b) Calcule o comprimento dos passos primal e dual ρ_p , ρ_d :

$$\rho_p = 0,99 \times \max \{ \rho : x + \rho \Delta x \geq 0, \rho \leq 1 \};$$

$$\rho_d = 0,99 \times \max \{ \rho : z + \rho \Delta z \geq 0, \rho \leq 1 \}.$$

c) Atualize as variáveis:

$$x \leftarrow x + \rho_p \Delta x;$$

$$y \leftarrow y + \rho_d \Delta y;$$

$$z \leftarrow z + \rho_d \Delta z;$$

$$\mu \leftarrow (1 - \min(\rho_p, \rho_d, 0, 99)) \bar{\mu}.$$

$$d) k \leftarrow k + 1.$$

4) Termine se as três condições seguintes são satisfeitas:

- Infactibilidade Primal = $\frac{\|b - Ax - \delta^2 y\|_2}{1 + \|x\|_2} < \text{Featol};$
- Infactibilidade Dual = $\frac{\|c + \gamma^2 x - z - A^T y\|_2}{1 + \|y\|_2} < \text{Featol};$
- Gap de Dualidade = $\frac{z^T x}{1 + \|x\|_2} < \text{PDGapTol}.$

5.3 Variante do Método Primal-Dual Barreira Logarítmica Predictor-Corretor

No segunda seção do Capítulo 4, associamos ao programa linear perturbado (4.3) o subproblema barreira logarítmica e, tendo definido o Lagrangeano, calculamos as condições necessárias de primeira ordem do problema de minimização da função Lagrangeana, obtendo:

$$\begin{aligned} \nabla_x L &= A^T y - \gamma^2 x - c + z = 0 \\ \nabla_p L &= \delta y - p = 0 \\ \nabla_y L &= Ax + \delta p - b = Ax + \delta^2 y - b = 0 \end{aligned}, \quad (5.4)$$

e da condição de complementaridade:

$$ZXe - \mu e = 0.$$

Prosseguimos, no capítulo em questão, substituindo a segunda equação de (5.4) na terceira equação; agora, vamos propor uma variação do Método Primal-Dual Barreira Logarítmica Predictor-Corretor, onde não realizaremos tal substituição. Dessa forma, rearranjaremos as expressões das condições de primeira ordem como:

$$F(x, y, z, p) = \begin{pmatrix} A^T y - \gamma^2 x - c + z \\ \delta y - p \\ Ax + \delta p - b \\ ZXe - \mu e \end{pmatrix} = \mathbf{0},$$

onde z é um vetor dual, X e Z são as matrizes diagonais formadas pelos elementos dos vetores x e z , respectivamente.

Aplicando o Método de Newton a $F(x, y, z, p) = \mathbf{0}$, obtemos:

$$F(x, y, z, p) + (\Delta x, \Delta y, \Delta z, \Delta p) \begin{pmatrix} \frac{\partial F}{\partial x}(x, y, z, p) \\ \frac{\partial F}{\partial y}(x, y, z, p) \\ \frac{\partial F}{\partial z}(x, y, z, p) \\ \frac{\partial F}{\partial p}(x, y, z, p) \end{pmatrix} = \mathbf{0},$$

que corresponde a:

$$\begin{pmatrix} -\gamma^2 \Delta x + A^T \Delta y + \Delta z \\ \delta \Delta y - I \Delta p \\ A \Delta x + \delta \Delta p \\ Z \Delta x + X \Delta z \end{pmatrix} = \begin{pmatrix} c + \gamma^2 x - z - A^T y \\ -\delta y + p \\ b - Ax - \delta p \\ \mu e - Zx \end{pmatrix} = \begin{pmatrix} t \\ r \\ v \\ w \end{pmatrix},$$

que fica reduzido ao seguinte sistema:

$$(ADA^T + \delta^2 I) \Delta y = r - AD(X^{-1}w - t) + \delta r, \quad (5.5)$$

$$\Delta x = DA^T \Delta y + D(X^{-1}v - t), \quad (5.6)$$

$$\Delta z = X^{-1}v - X^{-1}Z \Delta x, \quad (5.7)$$

$$\Delta p = \delta \Delta y - r, \quad (5.8)$$

sendo $D = (X^{-1}Z + \gamma^2 I)^{-1}$.

Adicionando a direção afim-escala, a direção de centragem, e a direção de correção dadas pelo Método Preditor-Corretor, vamos considerar no primeiro passo $w = -Zx$ e no segundo passo $w = \mu - Zx - \Delta Z \Delta X e$.

Como vimos na primeira seção deste capítulo, o método inicializa-se pela solução do Método de Frames (MOF). Assim, nosso p inicial será dado por: $p = \frac{(b - Ax)}{\delta}$. Para a escolha do comprimento do passo (ρ_p, ρ_d) , vamos fazer do mesmo modo que para o Método Primal-Dual Barreira Logarítmica Preditor-Corretor, onde sempre escolhemos o maior passo possível, contanto que $x^{(k+1)}$ e $z^{(k+1)}$ sejam pontos interiores. A cada passo do Método de Newton, vamos decrescer o parâmetro de barreira μ monotonicamente e mais rapidamente se passos largos são tomados:

$$\mu \leftarrow (1 - \min(\rho_p, \rho_d, 0, 99)) \bar{\mu}.$$

sendo $\bar{\mu}$ o valor de μ do passo anterior.

O método converge quando $\|b - Ax - \delta p\|_2$, $\|c + \gamma^2 x - z - A^T y\|_2$, $\|p - \delta y\|_2$ e $z^T x$ são suficientemente pequenos.

Algoritmo:

1- Conjunto de parâmetros:

- Tolerância de factibilidade: FeaTol,
- Tolerância *gap* de dualidade: PDGapTol,
- Dois parâmetros de perturbação: γ e δ .

2- Inicialize x, y, z, p e $\mu, k = 0$.

3- Loop.

a) Determine os dois passos seguintes:

1º Passo: Considere $v = -Zx$ e resolva (5.5) para Δy , calcule Δx por (5.6), Δz por (5.7) e Δp por (5.8).

2º Passo: Considere $v = \mu - Zx - \Delta Z \Delta X e$, sendo ΔZ e ΔX matrizes diagonais formadas pelos elementos dos vetores Δz e Δx respectivamente, que foram obtidos no passo anterior, e resolva (5.5) para Δy , calcule Δx por (5.6), Δz por (5.7) e Δp por (5.8) para obter novas direções.

b) Calcule o comprimento dos passos primal e dual ρ_p, ρ_d :

$$\rho_p = 0,99 \times \max \{ \rho : x + \rho \Delta x \geq 0, \rho \leq 1 \};$$

$$\rho_d = 0,99 \times \max \{ \rho : z + \rho \Delta z \geq 0, \rho \leq 1 \}.$$

c)Atualize as variáveis:

$$x \leftarrow x + \rho_p \Delta x;$$

$$y \leftarrow y + \rho_d \Delta y;$$

$$z \leftarrow z + \rho_d \Delta z;$$

$$p \leftarrow p + \rho_d \Delta p;$$

$$\mu \leftarrow (1 - \min(\rho_p, \rho_d, 0,99)) \bar{\mu}.$$

d) $k \leftarrow k + 1$.

4) Termine se as quatro condições seguintes são satisfeitas:

- Infactibilidade Primal = $\frac{\|b - Ax - \delta^2 y\|_2}{1 + \|x\|_2} < \text{FeaTol};$

- Infactibilidade Dual = $\frac{\|c + \gamma^2 x - z - A^T y\|_2}{1 + \|y\|_2} < \text{FeaTol};$

- $\frac{\|p - \delta y\|_2}{1 + \|p\|_2} < \text{FeaTol};$

- *Gap* de Dualidade = $\frac{z^T x}{1 + \|x\|_2} < \text{PDGapTol}.$

5.4 Características dos Métodos Apresentados

Considere o Método Barreira Logarítmica, o Método Primal-Dual Barreira Logarítmica Preditor-Corretor, e nossa variação do Método Primal-Dual Barreira Logarítmica Preditor-Corretor, como BP_BL, BP_InteriorPC e BP_InteriorPC1, respectivamente. Sendo Df a diferença relativa do valor da função objetivo do passo atual com o passo anterior e MPC referente ao Método Preditor-Corretor. Apresentamos nas tabelas (5.1) e (5.2) um resumo dos métodos apresentados.

Tabela 5.1: Resumo dos Métodos.

Método	BP_Interior	BP_BL
Conjunto de variáveis	x, y e z	x e y
Propriedades	Com condição de complementaridade e sem direções de correções do MPC	Sem condição de complementaridade e sem direções de correções do MPC
Critérios de Parada	Infactibilidade Primal: $\frac{\ b - Ax - \delta^2 y\ _2}{1 + \ x\ _2} < \text{Featol};$ Infactibilidade do Dual: $\frac{\ c + \gamma^2 x - z - A^T y\ _2}{1 + \ y\ _2} < \text{Featol};$ Gap de Dualidade: $\frac{z^T x}{1 + \ x\ _2} < \text{PDGapTol}.$	Infactibilidade Primal: $\frac{\ b - Ax - \delta^2 y\ _2}{1 + \ x\ _2} < \text{Featol};$ Infactibilidade do Dual: $\frac{\ c + \gamma^2 x - \mu X^{-1} e - A^T y\ _2}{1 + \ y\ _2} < \text{Featol}.$ ou se: Df é menor que 10^{-4} em mais de três iterações.

Tabela 5.2: Resumo dos Métodos.

Método	BP_InteriorPC	BP_InteriorPC1
Conjunto de variáveis	x, y e z	x, y, z e p
Propriedades	Com condição de complementaridade e com direções de correções do MPC	Com condição de complementaridade e com direções de correções do MPC
Critérios de Parada	Infactibilidade Primal: $\frac{\ b - Ax - \delta^2 y\ _2}{1 + \ x\ _2} < \text{Featol};$ Infactibilidade do Dual: $\frac{\ c + \gamma^2 x - z - A^T y\ _2}{1 + \ y\ _2} < \text{Featol};$ Gap de Dualidade: $\frac{z^T x}{1 + \ x\ _2} < \text{PDGapTol}.$	Infactibilidade Primal: $\frac{\ b - Ax - \delta^2 y\ _2}{1 + \ x\ _2} < \text{Featol};$ Infactibilidade do Dual: $\frac{\ c + \gamma^2 x - z - A^T y\ _2}{1 + \ y\ _2} < \text{Featol};$ $\frac{\ p - \delta y\ _2}{1 + \ p\ _2} < \text{Featol};$ Gap de Dualidade: $\frac{z^T x}{1 + \ x\ _2} < \text{PDGapTol}.$

No próximo capítulo, vamos aplicar os métodos desenvolvidos até agora, com o objetivo de obter representações eficientes de sinais, em dicionários específicos.

Capítulo 6

Testes Computacionais

A seguir, apresentamos os resultados obtidos para os sinais *TwinSine-1*, *WernerSorrows*, *Carbon*, *TwinSine-2*, *FM-Cosine*, *Gong*, *Dynamic-0*, *Dynamic-2* e *MultiGong*, que foram utilizados para o Método BP_Interior de Chen [6], e para os nossos métodos modificados de BP_Interior: Método Barreira Logarítmica, Método Primal-Dual Barreira Logarítmica Preditor-Corretor, e nossa variação do Método Primal-Dual Barreira Logarítmica Preditor-Corretor, os quais serão chamados BP_BL, BP_InteriorPC e BP_InteriorPC1, respectivamente.

Os experimentos numéricos foram implementados em Matlab R2010a, com o sistema operacional Linux, distribuição Ubuntu 11.04, processador Intel[®] core i7 2600, 3.4 Ghz, 4 GB de memória DDR3, RAM clock 1333Mhz.

Na tabela a seguir listamos os sinais utilizados, com seus respectivos parâmetros, que são números ou vetores linha, o nome do dicionário que vamos utilizar para a representação e o tamanho do problema.

Tabela 6.1: Dados do problema.

Sinal	Tamanho do Problema	Dicionário	par1	par2	par3
TwinSine-1	256	DCT	4	0	0
WernerSorrows	1024	CP	6	seno	0
Carbon	1024	WP	10	qmf	0
TwinSine-2	256	DCT	4	0	0
FM-Cosine	1024	CP	6	seno	0
Gong	1024	CP	10	seno	0
Dynamic-0	256	DCT e DIRAC	MekeList(4,0)	0	0
Dynamic-2	256	DCT e DIRAC	MekeList(4,0)	0	0
MultiGong	256	MDC	8	1	0

As Tabelas 6.2, 6.3 e 6.4 mostram os resultados obtidos para BP_Interior, BP_BL, BP_InteriorPC e BP_InteriorPC1.

Nas Tabelas 6.2 e 6.3 temos os valores da função objetivo, ou seja, o mínimo valor de $|\alpha|_1$ em (2.3), e o tempo decorrido até a obtenção da convergência, sendo que FO corresponde ao valor da função objetivo e o tempo é dado em segundos.

Tabela 6.2: Resultados computacionais para a função objetivo e tempo de processamento.

Sinal	BP_Interior		BP_BL	
	<i>FO</i>	<i>Tempo</i>	<i>FO</i>	<i>Tempo</i>
TwinSine-1	2,00933e+00	0,1	2,01365e+00	0,8
WernerSorrows	5,07482e+02	249,2	5,17296e+02	78,0
Carbon	6,00247e+00	19,8	6,01948e+00	707,3
TwinSine-2	2,01150e+00	0,1	2,01425e+00	1,4
FM-Cosine	2,52872e+02	237,3	2,54593e+02	102,5
Gong	4,73171e+00	1924,5	4,73566e+00	2507,2
Dynamic-0	6,01964e+00	0,3	6,01029e+00	3,8
Dynamic-2	4,03672e+02	0,4	4,88308e+02	28,7
MultiGong	2,43810e+01	5,6	2,54105e+01	15,8

Tabela 6.3: Resultados computacionais para a função objetivo e tempo de processamento.

Sinal	BP_InteriorPC		BP_InteriorPC1	
	<i>FO</i>	<i>Tempo</i>	<i>FO</i>	<i>Tempo</i>
TwinSine-1	2,00934e+00	0,1	2,00934e+00	0,1
WernerSorrows	5,07587e+02	130,0	5,07576e+02	124,1
Carbon	6,00003e+00	19,6	6,00003e+00	20,2
TwinSine-2	2,01108e+00	0,1	2,01108e+00	0,1
FM-Cosine	2,52885e+02	210,9	2,52896e+02	208,5
Gong	4,73088e+00	3501,3	4,72132e+00	11587,9
Dynamic-0	6,01902e+00	0,4	6,01907e+00	0,5
Dynamic-2	4,02187e+02	0,7	4,02187e+02	0,7
MultiGong	2,44009e+01	7,2	2,43895e+01	7,3

Na Tabela 6.4 são apresentados o número de iterações realizadas por cada método, e o número de iterações realizadas pelo Método dos Gradientes Conjugados. Temos que ItGC corresponde ao número de iterações do Método dos Gradientes Conjugados.

Tabela 6.4: Número de iterações dos métodos.

Sinal	BP_Interior		BP_BL		BP_InteriorPC		BP_InteriorPC1	
	<i>It</i>	<i>ItGC</i>	<i>It</i>	<i>ItGC</i>	<i>It</i>	<i>ItGC</i>	<i>It</i>	<i>ItGC</i>
TwinSine-1	11	59	24	1745	9	90	9	90
WernerSorrows	18	20883	91	6687	11	10850	11	10520
Carbon	8	85	35	4722	6	99	6	99
TwinSine-2	9	43	24	2950	9	97	9	98
FM-Cosine	17	20098	67	8862	12	17891	12	18026
Gong	21	11510	29	15377	19	21681	23	71812
Dynamic-0	7	40	24	3910	5	54	5	54
Dynamic-2	7	52	220	29022	7	91	7	91
MultiGong	19	853	57	11088	15	1097	17	1457

A seguir ilustramos a relação entre o número de iterações e o valor da função objetivo, obtidos nos métodos implementados, com os sinais da Tabela 6.1.

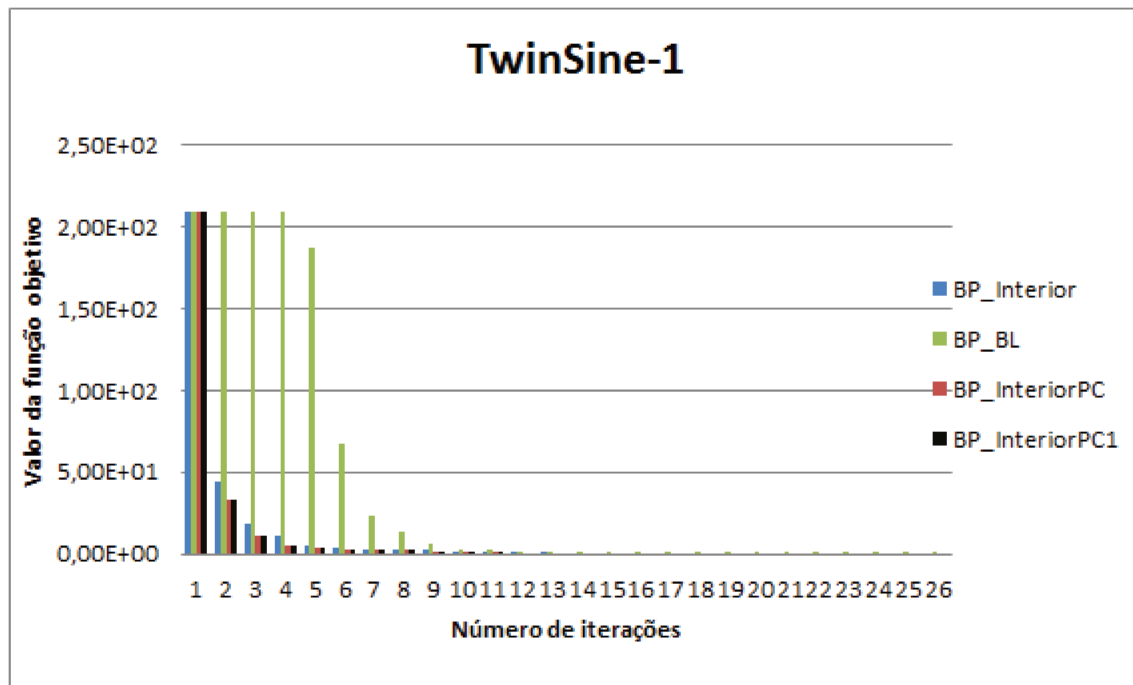


Figura 6.1: Valores da função objetivo referentes ao sinal TwinSine-1.

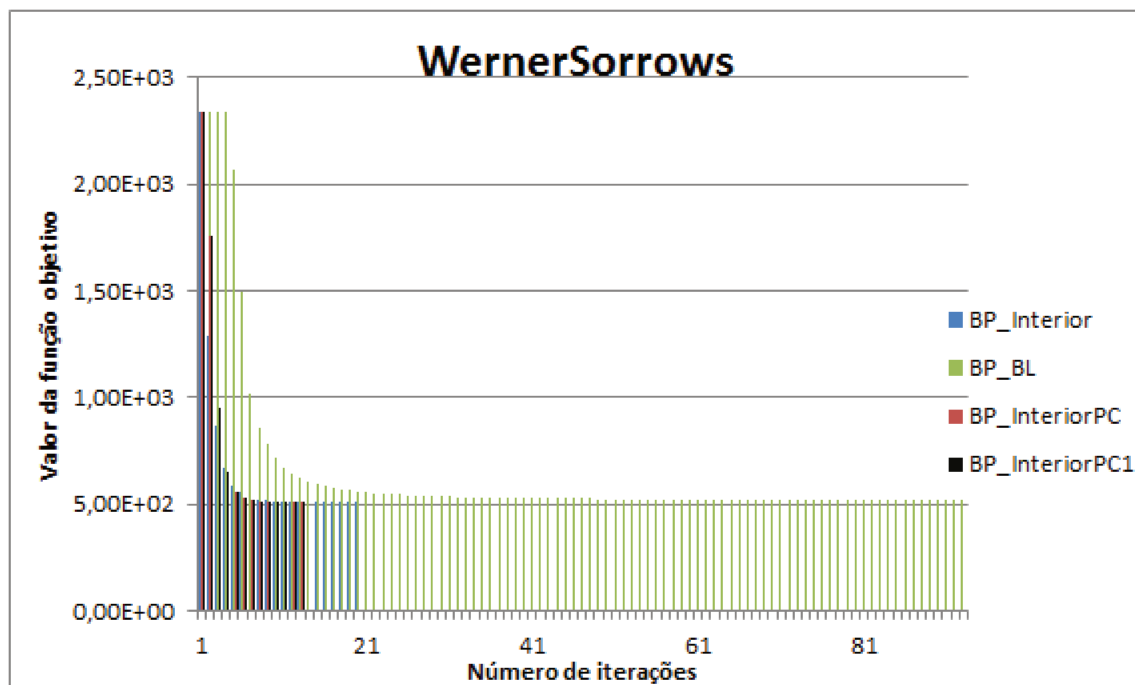


Figura 6.2: Valores da função objetivo referentes ao sinal WernerSorrows.

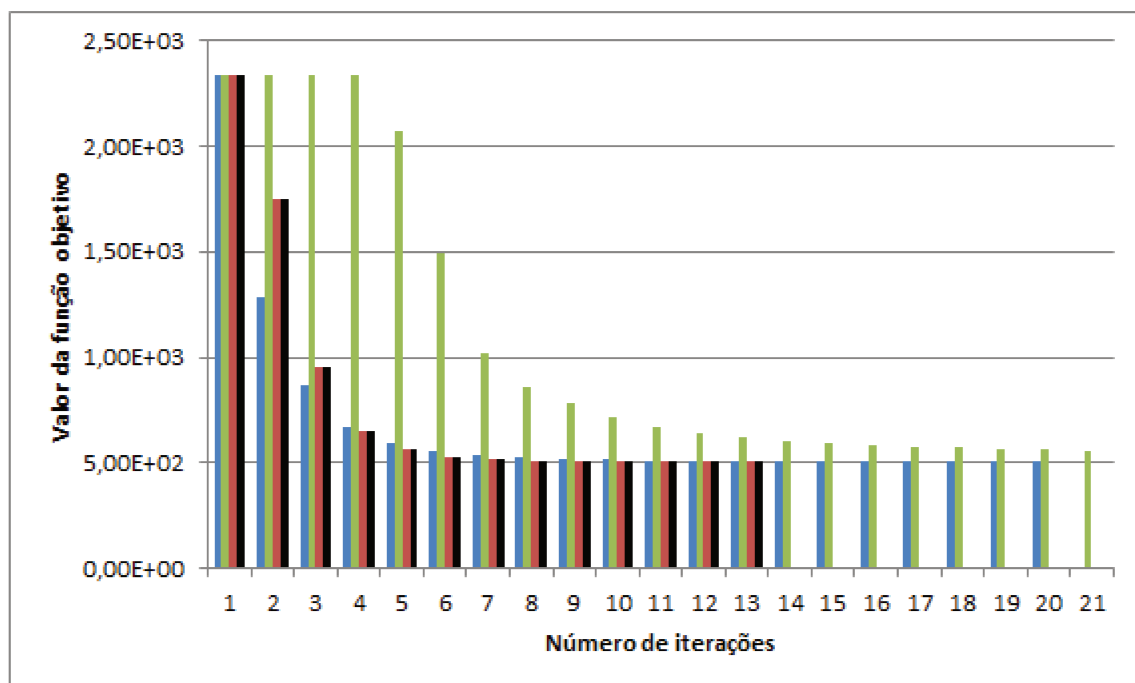


Figura 6.3: Valores do sinal WernerSorrows referentes até a 21^a iteração.

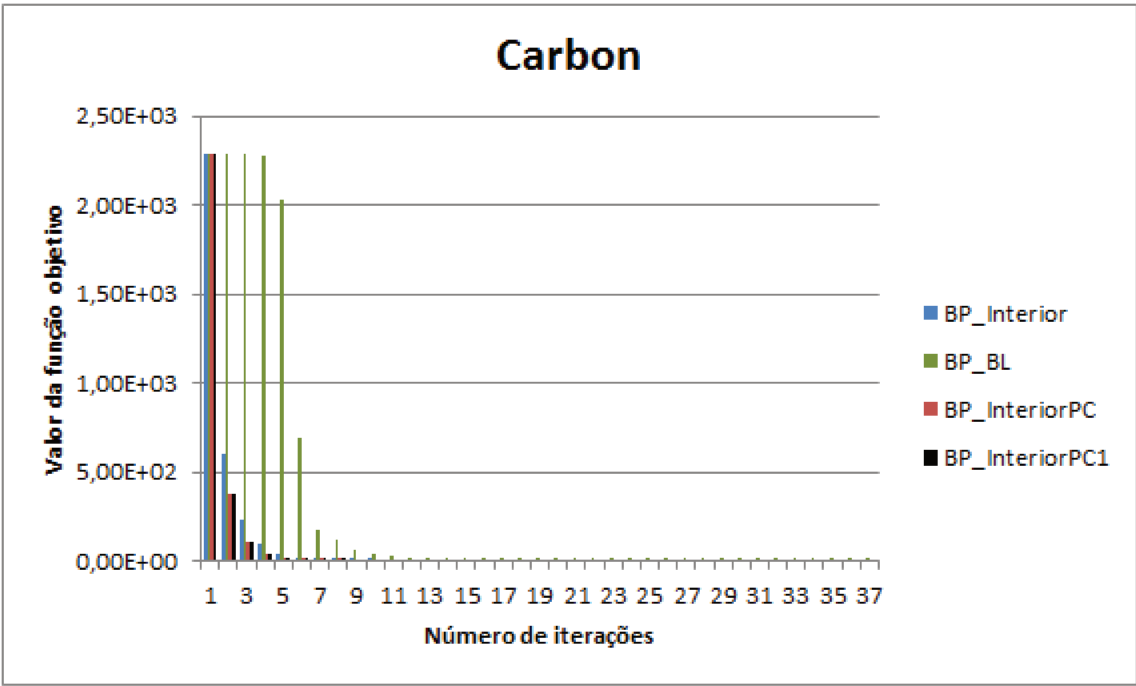


Figura 6.4: Valores da função objetivo referentes ao sinal Carbon.

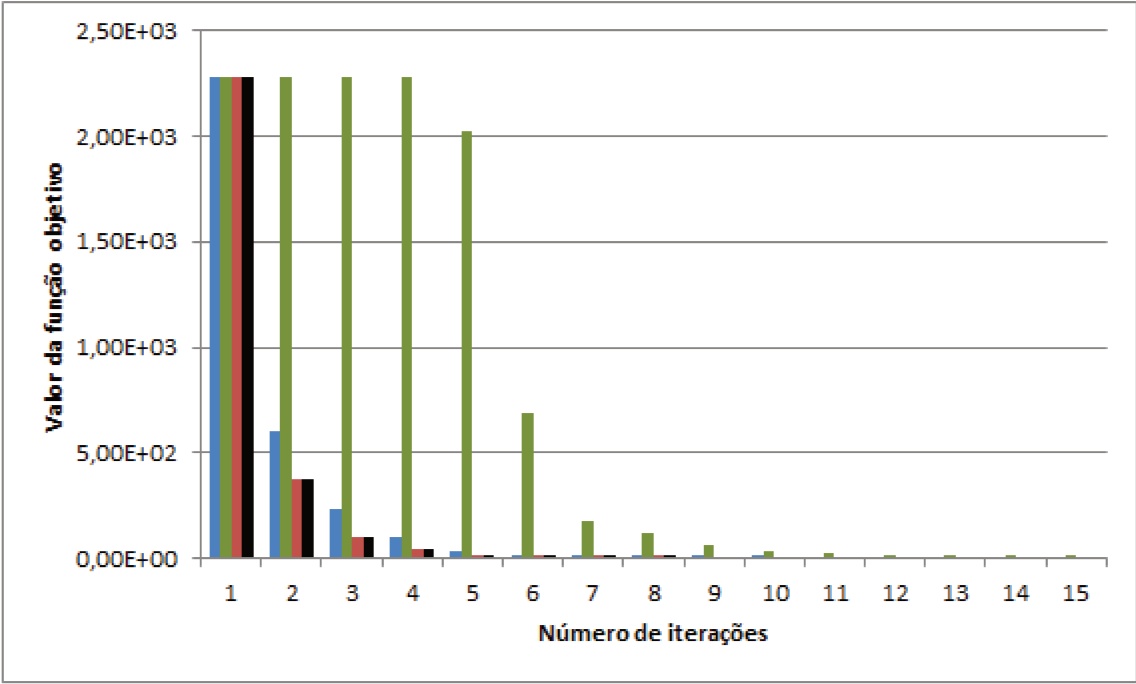


Figura 6.5: Valores do sinal Carbon referentes até a 15ª iteração.

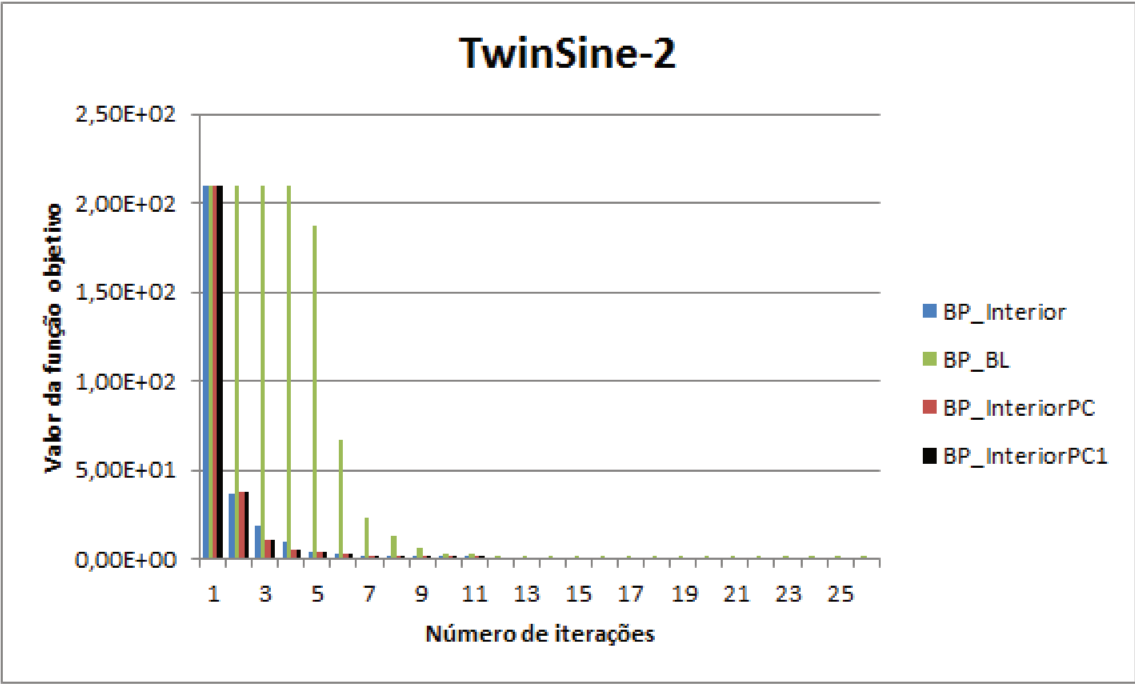


Figura 6.6: Valores da função objetivo referentes ao sinal TwinSine-2.

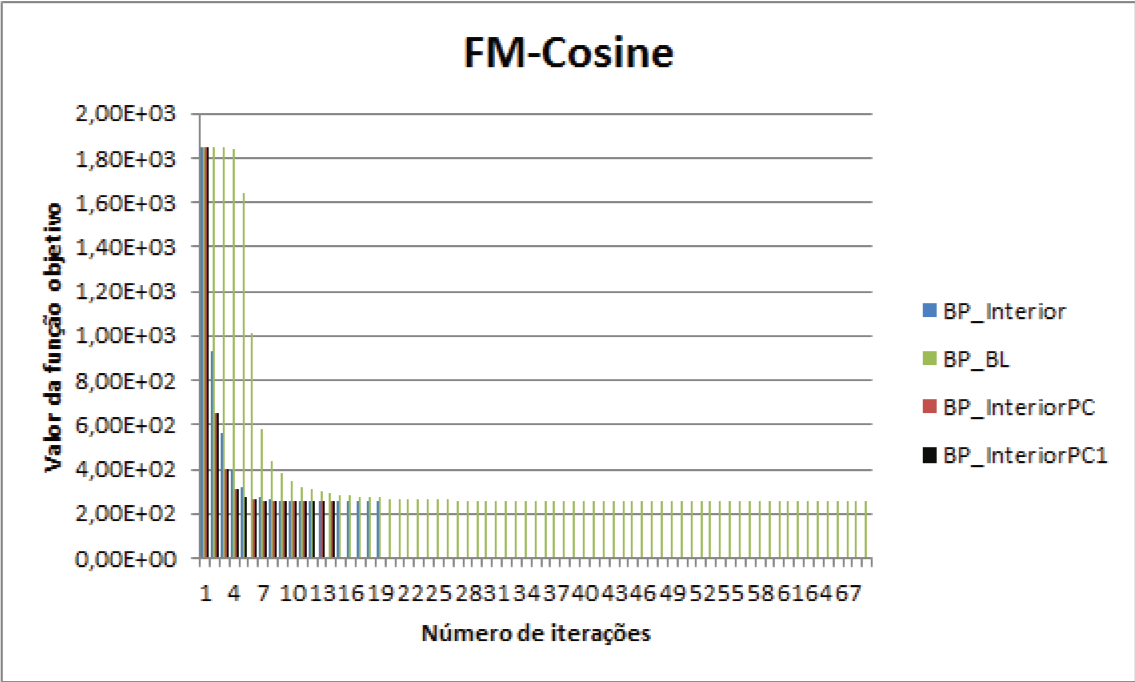


Figura 6.7: Valores da função objetivo referentes ao sinal FM-Cosine.

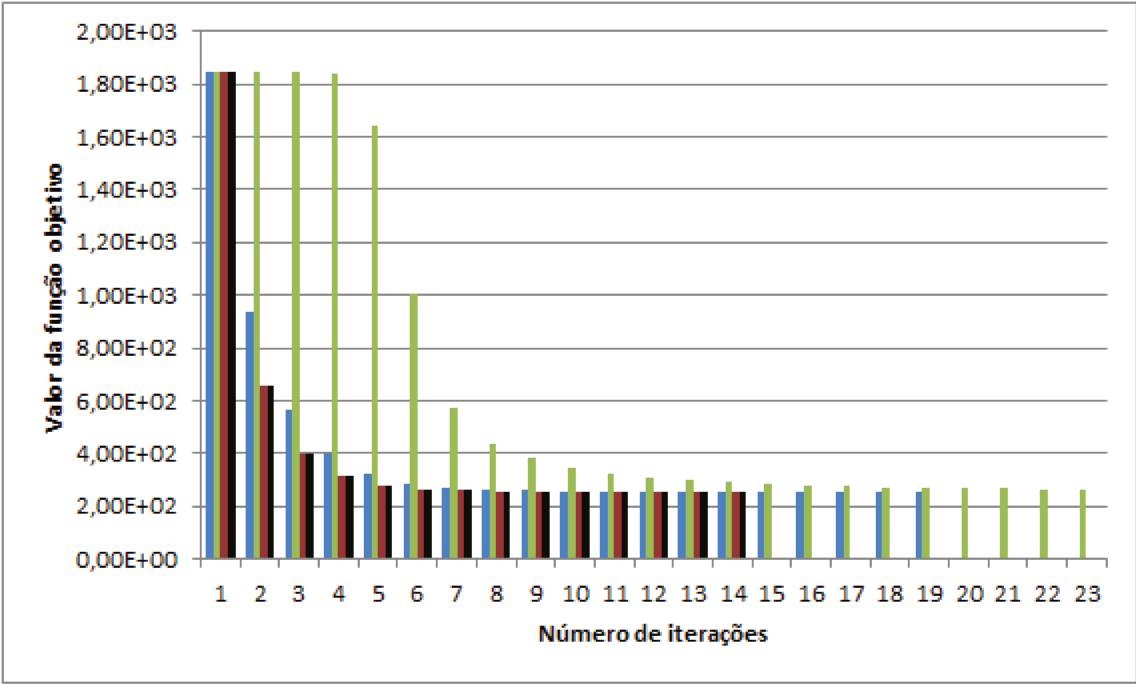


Figura 6.8: Valores do FM-Cosine referentes até a 23ª iteração.

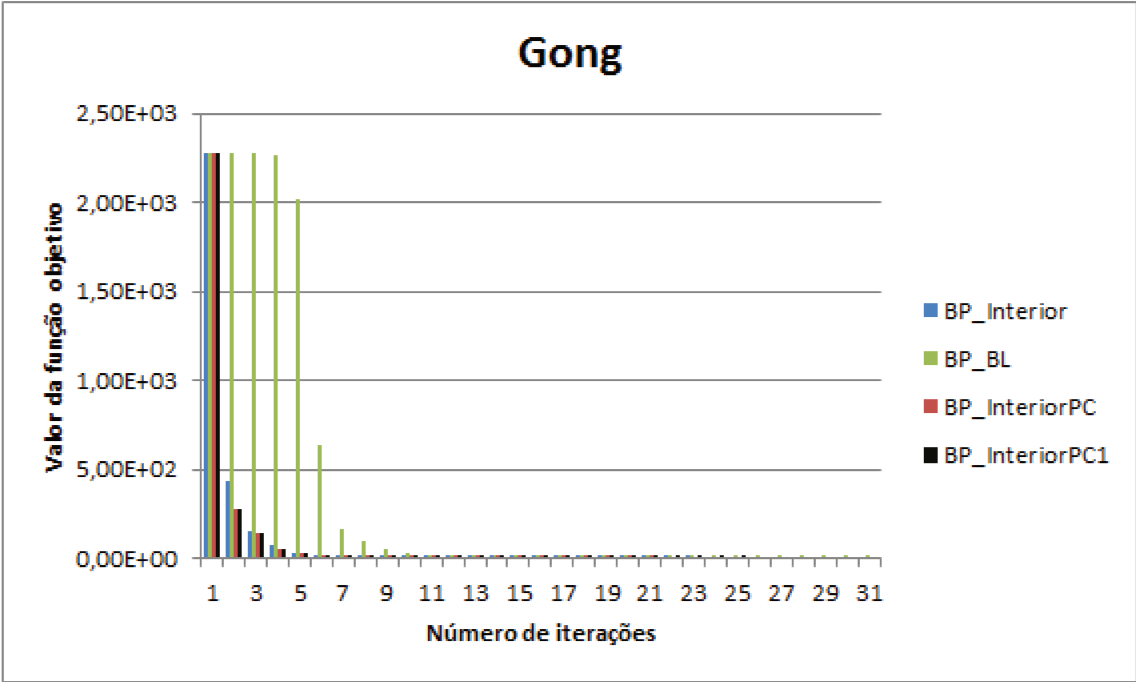


Figura 6.9: Valores da função objetivo referentes ao sinal Gong.

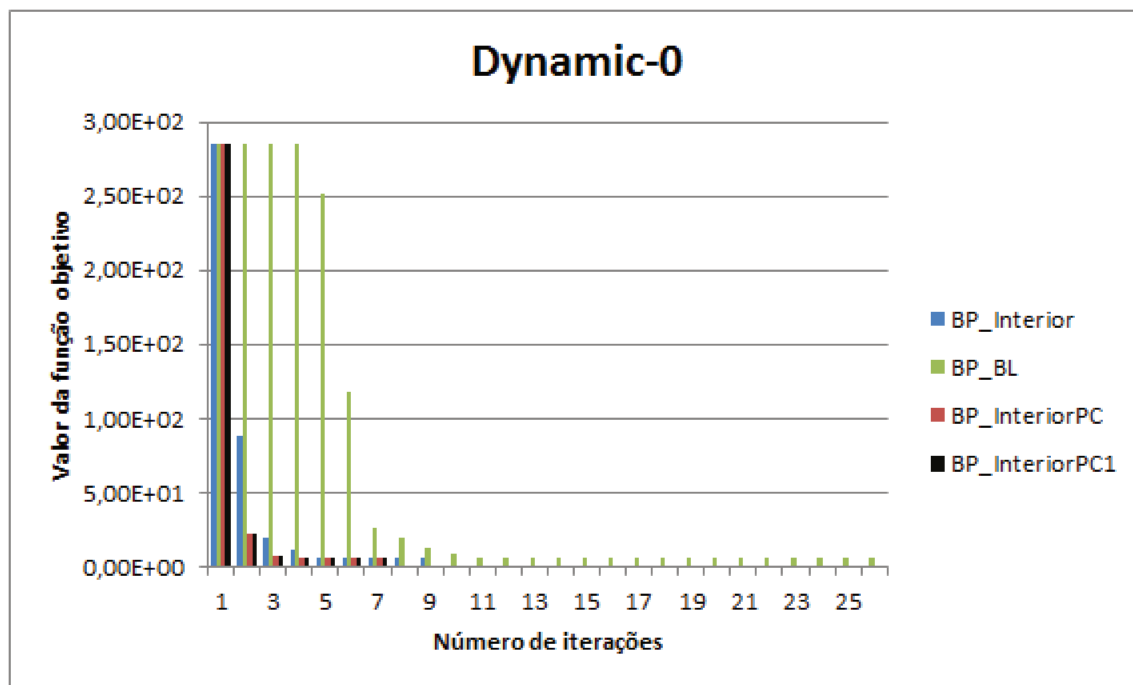


Figura 6.10: Valores da função objetivo referentes ao sinal Dynamic-0.

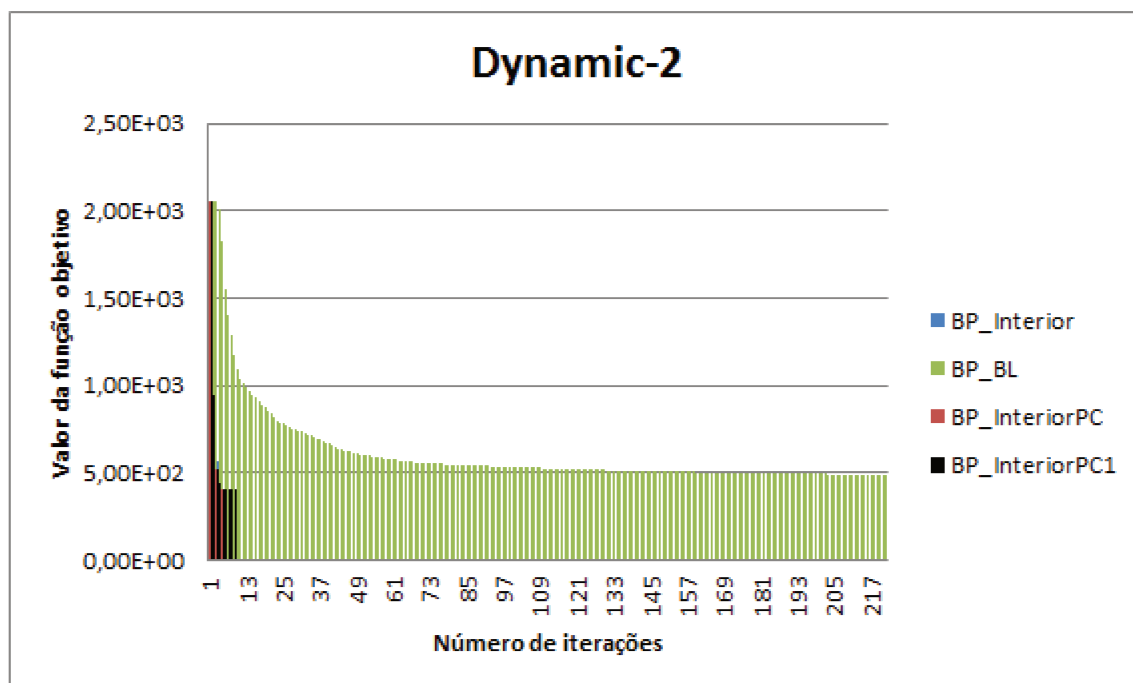


Figura 6.11: Valores da função objetivo referentes ao sinal Dynamic-2.

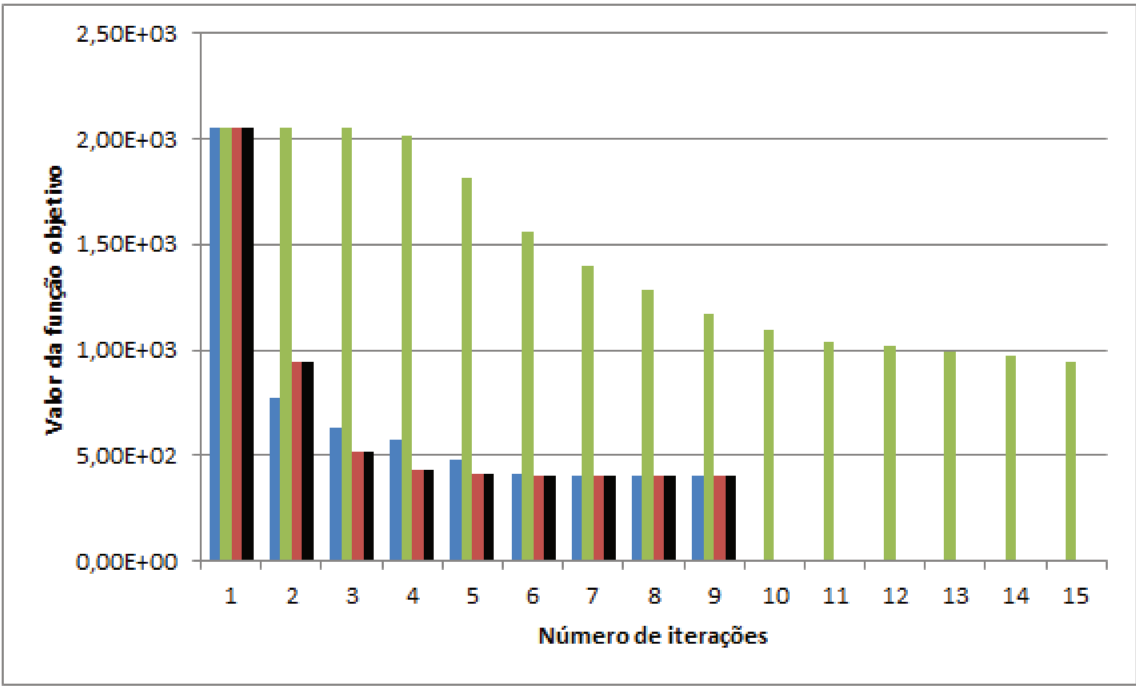


Figura 6.12: Valores do sinal Dynamic-2 referentes até a 15ª iteração.

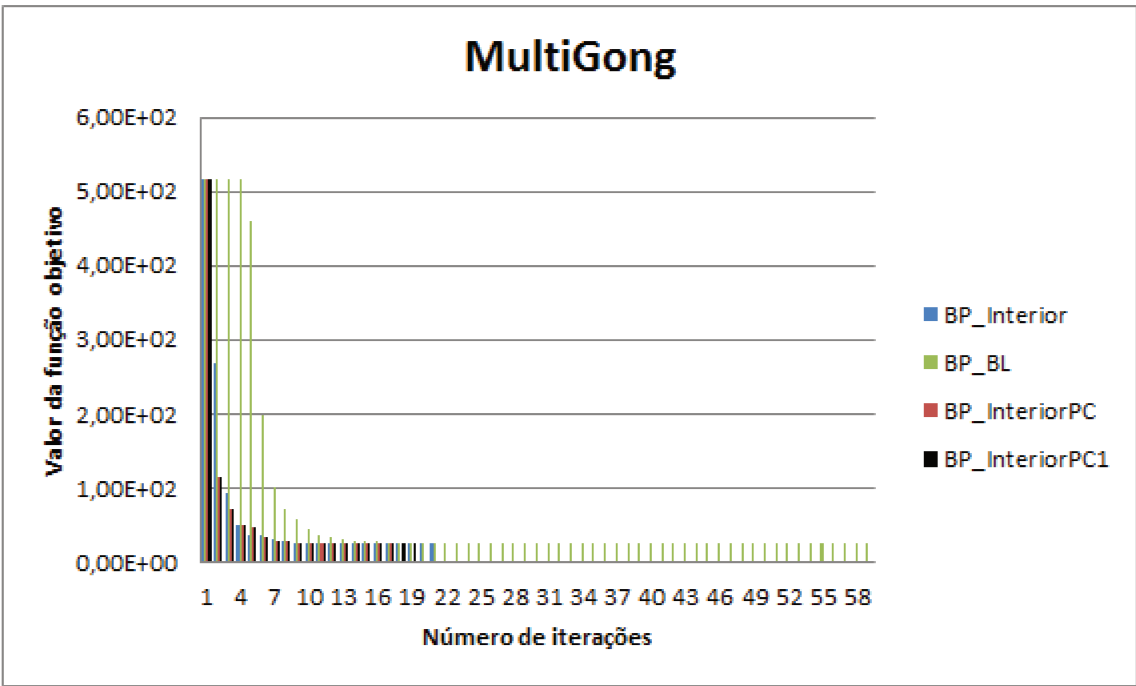


Figura 6.13: Valores da função objetivo referentes ao sinal MultiGong.

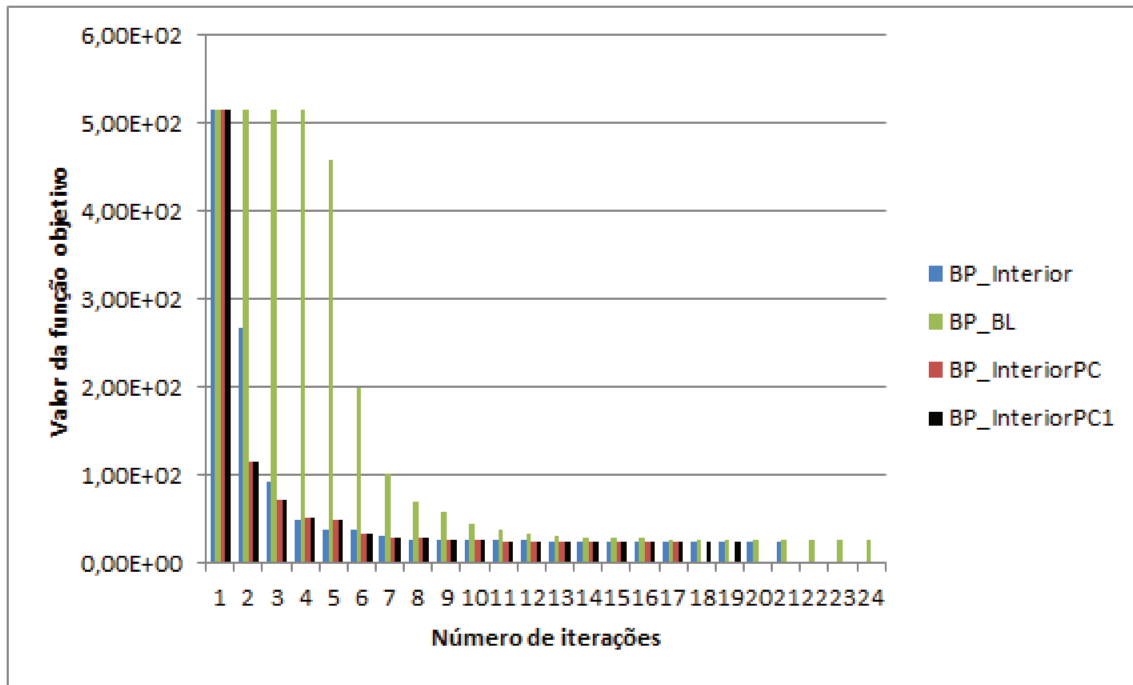


Figura 6.14: Valores do sinal MultiGong referentes até a 24^a iteração.

Nota-se que BP_Interior, BP_InteriorPC e BP_InteriorPC1 possuem desempenho computacional equivalente em relação ao valor da função objetivo. BP_BL não teve um resultado satisfatório em relação aos outros, obtendo um valor da função objetivo melhor apenas em *Dynamic-0*, mas tendo um tempo computacional bem maior. A diferença dos valores alcançados para a função objetivo deve-se ao fato do critério de convergência ser diferente para BP_BL.

O tempo de convergência para BP_Interior, BP_InteriorPC e BP_InteriorPC1 foram similares, sendo as diferenças mais significativas obtidas para o sinal *WernerSorrows*, onde BP_Interior obteve 249,2 segundos para a convergência, praticamente o dobro de tempo que BP_InteriorPC (130,0 segundos) e BP_InteriorPC1 (124,1 segundos); e para o sinal *Gong*, obtendo tempos bem distintos para os três métodos em questão, para BP_Interior obtemos 1924,5 segundos para a convergência, para BP_InteriorPC 3501,3 segundos, e BP_InteriorPC1 11587,9 segundos, um número bem maior em relação aos anteriores. O tempo obtido por BP_BL foi menor que dos outros métodos apenas para *WernerSorrows* e *FM-Cosine*, contudo o valor da função objetivo obtida, não foi satisfatória.

Observamos claramente nas figuras anteriores de (6.1) à (6.13), que a função objetivo tem um decréscimo mais lento em BP_BL e um maior decréscimo em BP_InteriorPC e BP_InteriorPC1. Isso é constatado na Tabela 6.4.

Quando o número de iterações realizadas por BP_InteriorPC não foi menor que o número realizado por BP_Interior, essa mostrou-se igual. Já BP_InteriorPC1 obteve mais iterações que BP_Interior apenas para o sinal *Gong*, com uma diferença de apenas duas iterações.

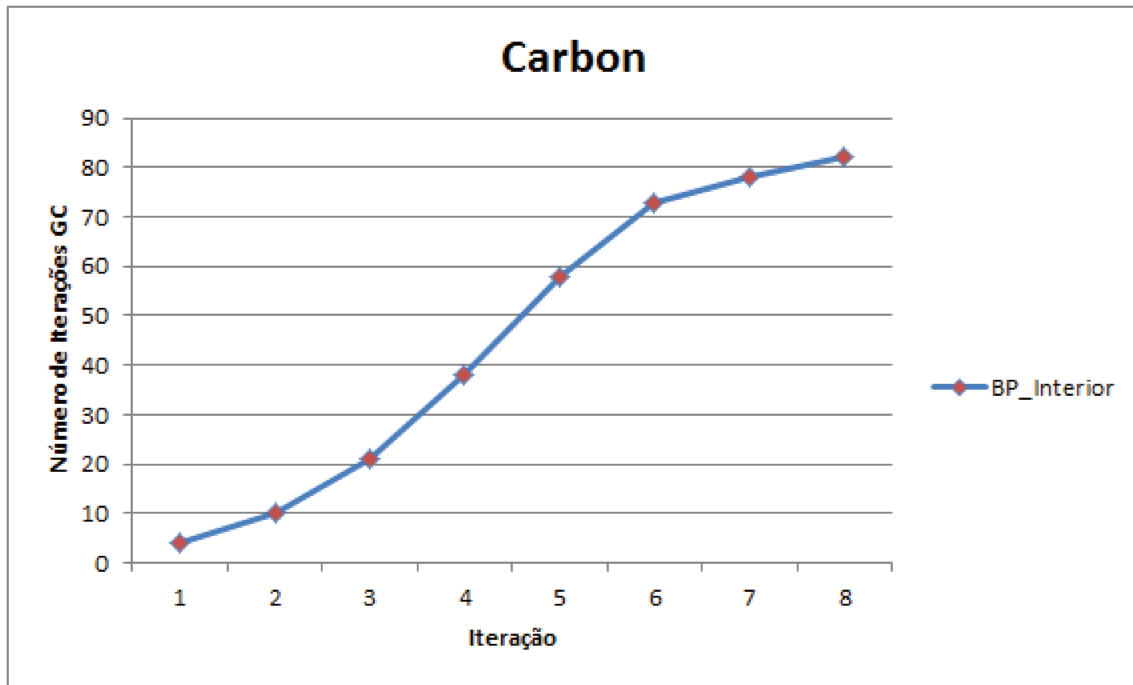


Figura 6.15: Número de iterações do Método dos Gradientes Conjugados a cada iteração do método.

Em relação ao Método dos Gradientes Conjugados, podemos notar pela Figura (6.15), referente ao sinal Gong e sua representação através de BP_Interior, e pelos resultados obtidos em [6], que o número de iterações do Método dos Gradientes Conjugados aumenta constantemente. Isso ocorre porque a solução inicial está próxima do centro da região factível. Assim, nas iterações iniciais, o sistema de equações (4.4) é bem condicionado e o método converge rapidamente. Mas, como $x^T z$ converge para 0, z/x convergirá para infinito ou 0. Assim, a matriz $D = (X^{-1}Z + \gamma^2 I)$ e a matriz $(ADA^T + \delta^2 I)$ tornam-se mais mal-condicionadas, e o método demora para convergir. Nos métodos implementados, não pré-condicionamos a matriz A. Por isso, obtivemos grande número de iterações no Método dos Gradientes Conjugados.

Capítulo 7

Conclusões e Perspectivas Futuras

7.1 Conclusões

Nesse trabalho apresentamos o problema de minimização *Basis Pursuit*, tendo como motivação sua aplicação em problemas envolvendo processamento de sinais.

Reescrevemos o problema *Basis Pursuit* como um problema de programação linear em sua forma padrão por meio de associações.

Consideramos a reformulação do problema como um problema quadrático e introduzimos os métodos Barreira Logarítmica, Primal-Dual Barreira Logarítmica Preditor-Corretor e uma variante deste último, os comparando com o método Primal-Dual Barreira Logarítmica implementado por Chen.

Os métodos Primal-Dual Barreira Logarítmica Preditor-Corretor e sua variante apresentaram resultados satisfatórios em relação ao método implementado por Chen.

O valor da função objetivo foi muito similar para os métodos Primal-Dual Barreira Logarítmica, Primal-Dual Barreira Logarítmica Preditor-Corretor e sua variante. O método Barreira Logarítmica não obteve resultados satisfatórios comparado aos outros métodos. Isso já era esperado, visto que não estamos considerando a condição de complementaridade.

Em relação ao tempo de convergência, também obtivemos tempos similares, com diferenças significativas em apenas dois sinais para os métodos Primal-Dual Barreira Logarítmica, Primal-Dual Barreira Logarítmica Preditor-Corretor e sua variante. Para o método Barreira Logarítmica, quando seu tempo de convergência foi menor em comparação aos outros métodos, o valor da função objetivo não foi satisfatório.

O número de iterações realizadas pelo método Primal-Dual Barreira Logarítmica Preditor-Corretor foi menor ou igual as realizadas pelo método Primal-Dual Barreira Logarítmica. A variante do método Primal-Dual Barreira Logarítmica Preditor-Corretor obteve mais iterações que o método Primal-Dual Barreira Logarítmica em apenas um sinal.

Portanto, obtivemos um melhor desempenho com as direções afim-escala, direção de centragem e direção de correção do Método Preditor-Corretor, e um resultado menos eficiente quando não consideramos

a condição de complementaridade. E apesar do método Primal-Dual Barreira Logarítmica Preditor-Corretor e sua variante terem obtido resultados muito similares, o método Primal-Dual Barreira Logarítmica Preditor-Corretor obteve os melhores resultados, sendo assim o mais eficiente.

7.2 Perspectivas Futuras

Com a finalidade de obter melhor desempenho, pretendemos pré-condicionar a matriz de interesse, a fim de obter um número menor de iterações no Método dos Gradientes Conjugados.

Também pretendemos verificar a aplicação dos métodos para dicionários diferentes e de maior dimensão, verificando se esta abordagem é viável em aplicações reais.

Apêndice A

Método dos Gradientes Conjugados

Sendo A uma matriz simétrica e definida positiva de ordem n , vamos encontrar $x^* \in \mathbb{R}$, solução do seguinte sistema:

$$Ax = b. \quad (\text{A.1})$$

Como A é definida positiva, A é invertível, logo:

$$x^* = A^{-1}b, \quad (\text{A.2})$$

é a única solução de (A.1).

Sabemos que a fatoração de Cholesky fornece uma solução numérica para a equação anterior.

Vamos apresentar os Métodos dos Gradientes, que são convenientes quando a matriz definida positiva do sistema linear é esparsa.

Considere o seguinte funcional:

$$J : \mathbb{R}^n \rightarrow \mathbb{R},$$
$$x \rightarrow J(x) = \frac{1}{2} \left((Ax)^T x \right) - b^T x.$$

Primeiramente vamos mostrar que o problema de minimização:

Encontrar $x^* \in \mathbb{R}^n$ tal que,

$$J(x^*) = \text{minimizar} \{ J(x), x \in \mathbb{R}^n \},$$

é equivalente a resolver o sistema linear definido-positivo:

Encontrar $x^* \in \mathbb{R}^n$ solução de

$$Ax = b.$$

Para mostrar tal equivalência, vamos inicialmente caracterizar o ponto crítico do funcional J , isto é, $\nabla J(x^*) = 0_{\mathbb{R}^n}$. Para isto, vamos calcular a primeira variação de J no ponto $x \in \mathbb{R}^n$ na direção do vetor v .

$$J(x + tv) = \frac{1}{2} (A(x + tv))^T (x + tv) - (b^T (x + tv))$$
$$= \frac{1}{2} \left[(Ax)^T x + t \left((Ax)^T v \right) + t \left((Av)^T x \right) + t^2 \left((Av)^T v \right) \right] - b^T x - t (b^T v).$$

$$J(x + tv) = \frac{1}{2} \left[(Ax)^T x + 2t \left((Ax)^T v \right) + t^2 \left((Av)^T v \right) \right] - b^T x - t \left(b^T v \right).$$

Derivando em relação a t , obtemos:

$$\frac{d}{dt} \{J(x + tv)\} = (Ax)^T v + t \left((Av)^T v \right) - b^T v,$$

fazendo $t = 0$:

$$J'(x)(v) = (Ax)^T v - b^T v,$$

$$J'(x)(v) = (Ax - b)^T v,$$

que corresponde a primeira variação de J no ponto x na direção do vetor v .

Sabemos que x^* é um ponto crítico de J se, e somente se,

$$J'(x^*)(v) = (Ax^* - b)^T v = 0, \forall v \in \mathbb{R}^n,$$

portanto, o ponto crítico x^* de J é a solução da equação

$$\nabla J(x) = Ax - b = 0_{\mathbb{R}^n} \iff Ax = b.$$

Para classificar o ponto crítico x^* de J , devemos calcular a segunda variação de J .

É fácil ver que

$$J''(x, v)(w) = (Aw)^T v,$$

é a segunda variação de J no ponto x na direção de w , uma vez que:

$$J''(x, v)(w) = \frac{d}{dt} \{J'(x + tv)(v)\}_{t=0}.$$

Note que $H(x) = A$ é a matriz Hessiana de J . Como $J''(x, v)(w)$ é independente da variável x , dizemos que J é um Funcional Quadrático.

Portanto x^* é um ponto de mínimo de J , uma vez que $J''(x, v)(w) = (Aw)^T v > 0$, já que A é definida positiva.

Vamos construir métodos numéricos para encontrar o mínimo de J , obtendo assim uma solução numérica para o sistema linear definido positivo $Ax = b$.

A.2 Método da Máxima Descida

A partir de uma aproximação inicial, que denotamos por $x^{(0)} \in \mathbb{R}^n$, construímos uma sequência minimizante $\{x^{(k)}\}$ para o funcional J , tal sequência caminha na direção de $-\nabla J(x^{(k)}) = b - Ax^{(k)} = r^{(k)}$. Dessa forma teremos:

$$x^{(k+1)} = x^{(k)} + \lambda r^{(k)},$$

onde k corresponde a iteração correspondente ao método.

Vamos escolher um λ de modo que $J(x^{(k+1)}) < J(x^{(k)})$ e que

$$J(x^{(k+1)}) = \min_{\lambda} \left\{ J(x^{(k)} + \lambda r^{(k)}) \right\},$$

dessa forma vamos determinar $\lambda_k \in \mathbb{R}$ da seguinte forma:

$$\frac{d}{d\lambda} J(x^{(k)} + \lambda r^{(k)}).$$

Resolvendo a equação acima, chegaremos a:

$$\lambda_k = \frac{(r^{(k)})^T r^{(k)}}{(Ar^{(k)})^T r^{(k)}}.$$

Como A é uma matriz definida positiva, temos que $\lambda_k > 0$. Este fato garante que estamos caminhando sempre na direção correta.

A seguir descrevemos o método da máxima descida, que também é conhecido como Método do Gradiente Otimizado.

Método da máxima descida

Dado $x^0 \in \mathbb{R}^n$ uma aproximação inicial.

Para $k = 0, 1, 2, 3, \dots$

$$r^{(k)} = b - Ax^{(k)};$$

$$\lambda_k = \frac{(r^{(k)})^T r^{(k)}}{(Ar^{(k)})^T r^{(k)}};$$

$$x^{(k+1)} = x^{(k)} + \lambda_k r^{(k)}.$$

fim

Para maiores detalhes do método ver [19].

A.3 Método dos Gradientes Conjugados

Agora, vamos considerar minimizações sucessivas do funcional J ao longo de direções que não são necessariamente as direções dos resíduos, como feito no método anterior.

Definindo as direções por p^k , onde k indica a iteração correspondente, teremos:

$$J(x^{(k+1)}) = \min_{\lambda} \left\{ J(x^{(k)} + \lambda p^{(k)}) \right\},$$

com

$$x^{(k+1)} = x^{(k)} + \lambda_k p^{(k)},$$

e sendo λ_k obtido da mesma do método do gradiente otimizado:

$$\lambda_k = \frac{\left(p^{(k)}\right)^T p^{(k)}}{\left(Ap^{(k)}\right)^T p^{(k)}}.$$

Nosso objetivo é determinar direções $p^{(k)}$, de modo que em n passos possamos obter a convergência do processo, sendo n a dimensão do sistema linear.

Neste método primeiramente vamos buscar direções consecuintes $p^{(k)}$ e $p^{(k+1)}$ A-conjugadas, ou seja,

$$\left(Ap^{(k)}\right)^T p^{(k+1)} = 0,$$

com $p^{(0)} = -\nabla J(x^{(0)}) = r^{(0)}$.

As direções consecuintes são dadas por:

$$p^{(k+1)} = r^{(k+1)} + \beta_k p^{(k)},$$

onde β_k é obtido de forma que $p^{(k)}$ e $p^{(k+1)}$ sejam A-conjugadas, o que ocorre quando:

$$\beta_k = -\frac{\left(Ar^{(k+1)}\right)^T p^{(k)}}{\left(Ap^{(k)}\right)^T p^{(k)}} = -\frac{\left(r^{(k+1)}\right)^T Ap^{(k)}}{\left(Ap^{(k)}\right)^T p^{(k)}}.$$

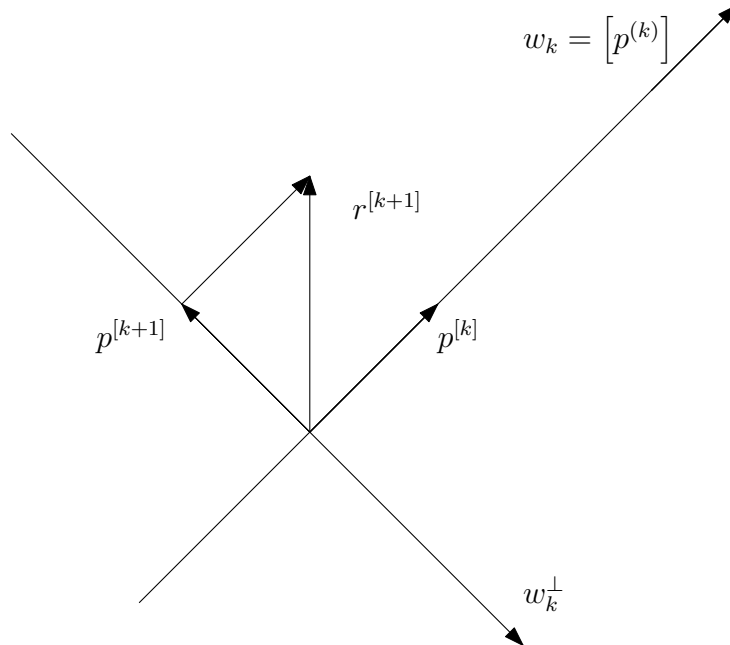


Figura A.1: Projeção ortogonal do resíduo sobre o complemento ortogonal.

Assim, como podemos visualizar na Figura A.1, a direção $p^{(k+1)}$ é a projeção ortogonal do resíduo r^{k+1} sobre o complemento ortogonal w_k^\perp , do subespaço gerado por $p^{(k)}$ (w_k) com relação ao produto interno.

Uma forma mais econômica de escrever os parâmetros λ_k , β_k e as demonstrações que $\{r(0), r(1), \dots, r(k), \dots\}$ é um conjunto ortogonal e $\{p(0), p(1), \dots, p(k), \dots\}$ é um conjunto A-conjugado, com a finalidade de mostrar que o Método dos Gradientes Conjugados converge para a solução exata do sistema linear definido positivo em n passos, encontra-se em [19], assim como mais detalhes do método.

Método dos Gradientes Conjugados

Dado $x^0 \in \mathbb{R}^n$ uma aproximação inicial.

$$r^{(0)} = b - Ax^{(0)};$$

$$p^{(0)} = r^{(0)}.$$

Para $k = 0, 1, 2, 3 \dots$

$$\lambda_k = \frac{(p^{(k)})^T r^{(k)}}{(Ap^{(k)})^T p^{(k)}};$$

$$x^{(k+1)} = x^{(k)} + \lambda_k p^{(k)};$$

$$r^{(k+1)} = r^{(k)} - \lambda_k Ap^{(k)};$$

$$\beta_k = -\frac{(Ap^{(k)})^T r^{(k+1)}}{(Ap^{(k)})^T p^{(k)}};$$

$$p^{(k+1)} = r^{(k+1)} + \beta_k p^{(k)}.$$

fim

O método converge em no máximo n passos para a solução exata somente na ausência de erros de arredondamento. Logo, o método pode falhar ou demorar demais dependendo da matriz.

Apêndice B

Método de Newton

Apresentamos o Método de Newton, que foi utilizado para resolver sistemas de equação não lineares nos métodos desenvolvidos.

Método de Newton para uma variável

Considere $f(x)$ uma função contínua em $[a, b]$, intervalo que contém uma raiz da equação $f(x) = 0$.

Transforma-se esta equação $f(x) = 0$ em uma equação equivalente $x = \varphi(x)$ e a partir de uma aproximação inicial x^0 gera-se a sequência $\{x^k\}$ de aproximações para x^* (raiz da equação $f(x) = 0$) pela relação $x^{k+1} = \varphi(x^k)$, pois a função $\varphi(x)$ é tal que $f(x^*) = 0$ se e somente se $\varphi(x^*) = x^*$.

A forma geral das funções de iteração $\varphi(x)$ é $\varphi(x) = x + A(x)f(x)$, tendo como condição que em x^* , ponto fixo de $\varphi(x)$ (ou seja, $\varphi(x^*) = x^*$), se tenha $A(x^*) \neq 0$.

Escolhe-se para a função de iteração a função $\varphi(x)$ tal que $\varphi'(x^*) = 0$.

Dessa forma, dada a equação $f(x) = 0$ e partindo da forma geral para $\varphi(x)$, queremos obter a função $A(x)$ tal que $\varphi'(x^*) = 0$.

Como feito em [21], temos:

$$\begin{aligned}\varphi(x) &= x + A(x)f(x) \Rightarrow \\ \Rightarrow \varphi'(x) &= 1 + A'(x)f(x) + A(x)f'(x) \\ \Rightarrow \varphi'(x^*) &= 1 + A'(x^*)f(x^*) + A(x^*)f'(x^*) \Rightarrow \varphi'(x^*) = 1 + A(x^*)f'(x^*).\end{aligned}$$

Assim, $\varphi'(x^*) = 0 \Leftrightarrow 1 + A(x^*)f'(x^*) = 0 \Rightarrow A(x^*) = \frac{-1}{f'(x^*)}$, onde tomamos $A(x) = \frac{-1}{f'(x)}$.

A função de iteração $\varphi(x) = x - \frac{f(x)}{f'(x)}$ será tal que $\varphi'(x^*) = 0$, pois como podemos verificar:

$$\varphi'(x) = 1 - \frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2} = \frac{f(x)f''(x)}{(f'(x))^2}$$

e, como $f(x^*) = 0$, $\varphi'(x^*) = 0$, sendo $f'(x^*) \neq 0$.

Dessa forma, a sequência $\{x^{k+1}\}$ será determinada por $x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}$.

A seguir apresentamos uma motivação geométrica do método.

Dado o ponto $(x^k, f(x^k))$ traçamos a reta $R_k(x)$ tangente à curva, que é definida pela equação $y = f(x)$, neste ponto:

$$R_k(x) = f(x^k) + f'(x^k)(x - x^k).$$

$R_k(x)$ é um modelo linear que aproxima a função $f(x)$ em uma vizinhança de x^k .

A raiz de $R_k(x) = 0$ é obtida quando $x = x^k - \frac{f(x^k)}{f'(x^k)}$. Prosseguimos cosiderando $x^{k+1} = x$.

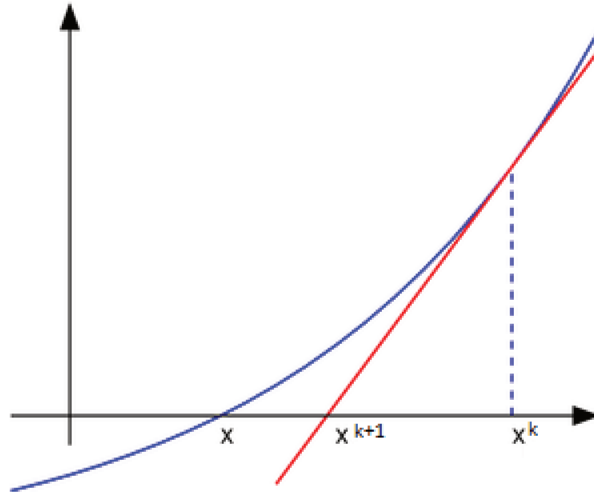


Figura B.1: Método de Newton (Extraída de [21]).

De [21], temos o teorema que nos dá as condições necessárias para tal método convergir. Nesse livro podemos encontrar o algoritmo do método com maiores detalhes.

Teorema B.0.1. Sejam $f(x)$, $f'(x)$ e $f''(x)$ contínuas num intervalo I que contém a raiz $x = x^*$ de $f(x) = 0$. Suponha que $f'(x^*) \neq 0$. Então existe um intervalo $\bar{I} \subset I$, contendo a raiz x^* , tal que, se $x^0 \in \bar{I}$, a sequência $\{x^k\}$ gerada pela fórmula recursiva $x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}$ converge para a raiz.

Método de Newton para várias variáveis

Dada uma função não linear $F : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$, $F = (f_1, \dots, f_n)^T$, o objetivo é determinar as soluções para:

$$F(x) = 0.$$

Conhecida a aproximação $x^k \in D$, para qualquer $x \in D$, existe $c_i \in D$, tal que:

$$f_i(x) = f_i(x^k) + \nabla f_i(c_i)^T (x - x^k)$$

com i variando de 1 a n .

Temos um modelo local linear para $f_i(x)$ em torno de x^k aproximando $\nabla f_i(c_i)$ por $\nabla f_i(x^k)$, i variando de 1 a n .

$$f_i(x) \approx f_i(x^k) + \nabla f_i(x^k)^T (x - x^k).$$

Dessa forma o modelo local linear para $F(x)$ em torno de x^k é:

$$F(x) \approx R_k(x^k) = F(x^k) + J(x^k)(x - x^k).$$

sendo que $J(x^k)$ refere-se a matriz Jacobiana avaliada em x^k .

A nova aproximação x^{k+1} será o valor em que o modelo local linear $R_k(x)$ se anula.

Note que,

$$R_k(x) = 0 \Leftrightarrow J(x^k)(x - x^k) = -F(x^k).$$

Denotando $(x - x^k)$ por v^k temos que $x^{k+1} = x^k + v^k$, onde v^k é solução do sistema linear

$$J(x^k)v = -F(x^k).$$

Determinado o valor de v , consideramos $x^{k+1} = x^k + v^k$.

Sob condições adequadas envolvendo o ponto inicial x^0 , a função $F(x)$ e a matriz Jacobiana $J(x)$, a sequência gerada $\{x^k\}$ converge a x^* a uma taxa quadrática.

Maiores detalhes sobre o teorema de convergência encontra-se em [9] e uma explicação mais detalhada sobre o Método de Newton para várias variáveis pode ser encontrado em [21].

Referências Bibliográficas

- [1] *Test image*, Disponível em: <<http://www.hlevkin.com/TestImages/classic.htm>>, Acesso em: 21 julho 2013.
- [2] Richard G Baraniuk, *Compressive sensing [lecture notes]*, Signal Processing Magazine, IEEE **24** (2007), no. 4, 118–121.
- [3] Leonardo V Batista, *Teoria da Informação*, Disponível em: <www.di.ufpb.br/leonardo/ensino>, Acesso em: 12 dez. 2012 (2005).
- [4] Daniela Renata Cantane, *Métodos de pontos interiores aplicados ao problema de regressão pela norma L_p* , Master's thesis, ICMC-USP, São Carlos, (2004).
- [5] Kristen Michelle Cheman, *Optimization Techniques for Solving Basis Pursuit Problems*, Master's thesis, North Carolina State University, (2006).
- [6] Scott Shaobing Chen, *Basis pursuit*, Ph.D. thesis, Stanford University, (1995).
- [7] Scott Shaobing Chen, David L Donoho, and Michael A Saunders, *Atomizer for matlab5. x*, Disponível em: <<http://www-stat.stanford.edu/~atomizer>>, Acesso em: 15 nov. 2012.
- [8] ———, *Atomic decomposition by basis pursuit*, SIAM review **43** (2001), no. 1, 129–159.
- [9] W. S. Dorn and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, vol. 16, SIAM Classics in Applied Mathematics, (1996).
- [10] Philip E Gill, Walter Murray, Dulce B Pongeleon, and Michael A Saunders, *Solving reduced KKT systems in barrier methods for linear and quadratic programming*, Tech. report, DTIC Document, (1991).
- [11] Eliana Contharteze Grigoletto, *Implementação eficiente dos métodos de pontos interiores especializados para o problema de regressão pela norma L_p* , Master's thesis, IMECC-UNICAMP, Campinas, (2011).
- [12] Simon Haykin and Barry Van Veen, *Sinais e sistemas*, Bookman, (2001).
- [13] Alexey Izmailov and Mikhail Solodov, *Otimização - condições de otimalidade, elementos de análise convexa e de dualidade*, vol. 1, Impa, (2009).
- [14] Irvin J Lustig, Roy E Marsten, and David F Shanno, *Computational experience with a primal-dual interior point method for linear programming*, Linear Algebra and Its Applications **152** (1991), 191–222.

- [15] ———, *On implementing Mehrotra's predictor-corrector interior-point method for linear programming*, SIAM Journal on Optimization **2** (1992), no. 3, 435–449.
- [16] Stephane Mallat, *A wavelet tour of signal processing: the sparse way*, Elsevier, (2008).
- [17] Kevin A McShane, Clyde L Monma, and David Shanno, *An implementation of a primal-dual interior point method for linear programming*, ORSA Journal on computing **1** (1989), no. 2, 70–83.
- [18] Sanjay Mehrotra, *On the implementation of a primal-dual interior point method*, SIAM Journal on optimization **2** (1992), no. 4, 575–601.
- [19] Petronio Pulino, *Algebra Linear e suas Aplicações Notas de Aula*, Disponível em: <www.ime.unicamp.br/pulino/ALESA>, Acesso em: 20 mai 2013 (2012).
- [20] Cornelis Roos, Tamás Terlaky, and Jean-Philippe Vial, *Theory and algorithms for linear optimization: an interior point approach*, Wiley, (1997).
- [21] Márcia A Gomes Ruggiero and Vera Lúcia da Rocha Lopes, *Cálculo numérico: aspectos teóricos e computacionais*, Makron Books do Brasil, (1997).
- [22] Mark Schmidt, *Least squares optimization with L1-norm regularization*, Project Report, University of British Columbia (2005).
- [23] Adriana Schulz, *Compressive Sensing-Novos Paradigmas para Aquisição e Compressão de Imagens*, Projeto apresentado ao Departamento de Engenharia Eletrônica e de Computação da Escola Politécnica da UFRJ como requisito parcial para obtenção do título de Engenheira Eleterônica e de Computação, (2008).
- [24] Stephen J Wright, *Primal-dual interior-point methods*, vol. 54, Society for Industrial and Applied Mathematics, (1987).