



UNIVERSIDADE ESTADUAL DE
CAMPINAS

Instituto de Matemática, Estatística e
Computação Científica

RONALDO LOPES

**Métodos de descenso coordenado por blocos e
identificação das restrições ativas em
otimização de porte enorme**

Campinas

2018

Ronaldo Lopes

**Métodos de descenso coordenado por blocos e
identificação das restrições ativas em otimização de
porte enorme**

Tese apresentada ao Instituto de Matemática,
Estatística e Computação Científica da Uni-
versidade Estadual de Campinas como parte
dos requisitos exigidos para a obtenção do
título de Doutor em Matemática Aplicada.

Orientadora: Sandra Augusta Santos

Coorientador: Paulo José da Silva e Silva

Este exemplar corresponde à versão
final da Tese defendida pelo aluno Ro-
naldo Lopes e orientada pela Profa.
Dra. Sandra Augusta Santos.

Campinas

2018

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

L881m Lopes, Ronaldo, 1988-
Métodos de descenso coordenado por blocos e identificação das restrições
ativas em otimização de porte enorme / Ronaldo Lopes. – Campinas, SP :
[s.n.], 2018.

Orientador: Sandra Augusta Santos.

Coorientador: Paulo José da Silva e Silva.

Tese (doutorado) – Universidade Estadual de Campinas, Instituto de
Matemática, Estatística e Computação Científica.

1. Otimização matemática. 2. Algoritmos. 3. Convergência global. 4.
Processamento paralelo (Computadores). 5. Problemas de grande porte
(Matemática). I. Santos, Sandra Augusta, 1964-. II. Silva, Paulo José da Silva
e, 1973-. III. Universidade Estadual de Campinas. Instituto de Matemática,
Estatística e Computação Científica. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Block-coordinate descent methods and active-set identification for
huge-scale problems

Palavras-chave em inglês:

Mathematical optimization

Algorithms

Global convergence

Parallel processing (Electronic computers)

Large-scale problems (Mathematics)

Área de concentração: Matemática Aplicada

Titulação: Doutor em Matemática Aplicada

Banca examinadora:

Sandra Augusta Santos [Orientador]

Jose Mario Martinez Perez

Lucio Tunes dos Santos

Ernesto Julián Goldberg Birgin

Geovani Nunes Grapiglia

Data de defesa: 23-04-2018

Programa de Pós-Graduação: Matemática Aplicada

**Tese de Doutorado defendida em 23 de abril de 2018 e aprovada
pela banca examinadora composta pelos Profs. Drs.**

Prof(a). Dr(a). SANDRA AUGUSTA SANTOS

Prof(a). Dr(a). JOSE MARIO MARTINEZ PEREZ

Prof(a). Dr(a). LUCIO TUNES DOS SANTOS

Prof(a). Dr(a). ERNESTO JULIÁN GOLDBERG BIRGIN

Prof(a). Dr(a). GEOVANI NUNES GRAPIGLIA

As respectivas assinaturas dos membros encontram-se na Ata de defesa

Agradecimentos

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo auxílio financeiro necessário para execução deste trabalho (processo *n*º. 2014/14228-6).

Aos orientadores Profa. Dra. Sandra Augusta Santos e Prof. Dr. Paulo José da Silva e Silva, pela elaboração do projeto inicial de pesquisa, por todo apoio proporcionado durante este projeto, pelo conhecimento compartilhado que será levado durante toda minha vida e pelo companheirismo, importante ao longo do doutorado.

Aos familiares, em especial minha esposa, companheira fiel, ombro amigo nas horas difíceis e que fez grandes sacrifícios na vida profissional e acadêmica para que eu pudesse alcançar essa conquista.

Aos amigos feitos em Campinas, pessoas que sempre serão lembradas com muito carinho durante a vida.

Resumo

Neste trabalho desenvolvemos estratégias de identificação das restrições ativas para o método de descenso coordenado por blocos aplicado a problemas de otimização irrestritos, ou em caixas, cuja função objetivo é a soma de uma função suave e outra convexa. Mostramos que, em certas situações, o método tem a capacidade intrínseca de identificação e também apresentamos um exemplo de função identificadora compatível com a simplicidade computacional exigida pelos problemas de porte enorme. Combinando essas estratégias, desenvolvemos um método de descenso coordenado por blocos, denominado *Active BCDM*, que busca explorar as restrições ativas do problema com restrições de caixa ou, no caso irrestrito, de uma reformulação auxiliar relacionada que possui variáveis não negativas. Analisamos nosso método em duas classes de problemas com muita relevância no contexto de otimização de porte enorme: *LASSO* e regressão logística com regularização ℓ_1 . Preparamos uma ampla discussão de resultados numéricos utilizando problemas reais extraídos da literatura. Isso permite a comparação do *Active BCDM* com vários métodos bem estabelecidos e do estado da arte para estes problemas, tanto no caso sequencial quanto no paralelo. Em ambas implementações, a proposta de identificação apresentou desempenho computacional superior aos métodos com os quais foi comparada. Além disso, resultados de convergência global acompanham os algoritmos propostos, reforçando sua consistência e relevância teórica.

Palavras-chave: Otimização matemática. Algoritmos. Convergência global. Experimentos numéricos. Processamento paralelo (Computadores). Problemas de grande porte (Matemática).

Abstract

This work is concerned with the development of strategies to identify active constraints for the block-coordinate descent method applied to unconstrained, or box-constrained, optimization problems whose objective function is the sum of a smooth component and a convex one. We show that, under appropriate assumptions, the method has an intrinsic identification capacity. We also present an example of an identification function compatible with the computational simplicity required to address large-scale problems. Combining these strategies, we have developed a block-coordinate descent method, called Active BCDM, which aims to explore the active constraints in box-constrained problems, or, in the unconstrained case, of a related auxiliary reformulation with non negative variables. We analyze the performance of our method for solving two classes of problems with great relevance in the context of huge-scale optimization: LASSO and ℓ_1 -regularized logistic regression. We have prepared an extensive discussion of numerical results using real problems from the literature. This allows the comparison of Active BCDM with several well-established and state-of-the-art methods for such problems, with sequential and parallel implementations. In both implementations, the identification strategy presented better computational performance among the methods under comparison. In addition, global convergence results have been proved for the proposed algorithms, reinforcing their consistency and theoretical relevance.

Keywords: Mathematical optimization. Algorithms. Global convergence. Numerical experiments. Parallel processing (Computers). Large-scale problems (Mathematics).

Lista de ilustrações

Figura 1 – <i>Performance profiles</i> mais significativos entre as 24 variantes do método BCDM+IF para os problemas da Tabela 1	77
Figura 2 – <i>Performance profiles</i> mais significativos entre as 24 variantes do método BCDM+ST para os problemas da Tabela 1	77
Figura 3 – <i>Performance profile</i> entre BCDM+IF e BCDM+ST para os problemas da Tabela 1	78
Figura 4 – <i>Performance profiles</i> mais significativos entre as 24 variantes do método UBCDM para os problemas da Tabela 1	78
Figura 5 – <i>Performance profile</i> entre UBCDM e ActiveBCDM para os problemas da Tabela 1	78
Figura 6 – <i>Performance profiles</i> mais significativos entre as 24 variantes do método ActiveBCDM para 18 problemas das Tabelas 3 e 4	82
Figura 7 – <i>Performance profiles</i> mais significativos entre as 24 variantes do método ActiveBCDM+S01 para 18 problemas das Tabelas 3 e 4	84
Figura 8 – <i>Performance profiles</i> mais significativos entre as 24 variantes do método ActiveBCDM+S02 para 18 problemas das Tabelas 3 e 4	85
Figura 9 – <i>Performance profiles</i> mais significativos entre as 24 variantes do método ActiveBCDM+S03 para 18 problemas das Tabelas 3 e 4	85
Figura 10 – <i>Performance profiles</i> mais significativos entre as 24 variantes do método ActiveBCDM+S04 para 18 problemas das Tabelas 3 e 4	86
Figura 11 – <i>Performance profiles</i> entre as variantes do método ActiveBCDM+S0(\cdot) calibradas para 18 problemas das Tabelas 3 e 4	86
Figura 12 – <i>Performance profile</i> entre ActiveBCDM e ActiveBCDM+S0 para 31 problemas das Tabelas 3 e 4	87
Figura 13 – <i>Performance profiles</i> entre os métodos quase-Newton (esquerda) e entre o melhor método quase-Newton e ActiveBCDM+S0 (direita) para 31 problemas das Tabelas 3 e 4.	88
Figura 14 – <i>Performance profiles</i> entre ActiveBCDM+S0 e FAST-BCD2-E (esquerda) e entre ActiveBCDM+S0 e SpARSA (direita) para 31 problemas das Tabelas 3 e 4.	89
Figura 15 – <i>Performance profiles</i> mais significativos entre as 24 variantes do método ActiveBCDM para 12 problemas das Tabelas 3 e 4	91
Figura 16 – <i>Performance profiles</i> mais significativos entre as 24 variantes do método ActiveBCDM+S01 para 12 problemas das Tabelas 3 e 4	91
Figura 17 – <i>Performance profiles</i> mais significativos entre as 24 variantes do método ActiveBCDM+S02 para 12 problemas das Tabelas 3 e 4	92

Figura 18 – <i>Performance profiles</i> mais significativos entre as 24 variantes do método ActiveBCDM+S03 para 12 problemas das Tabelas 3 e 4	93
Figura 19 – <i>Performance profiles</i> mais significativos entre as 24 variantes do método ActiveBCDM+S04 para 12 problemas das Tabelas 3 e 4	94
Figura 20 – <i>Performance profile</i> entre as variantes do método ActiveBCDM+S0(·) calibradas para 12 problemas das Tabelas 3 e 4	94
Figura 21 – <i>Performance profile</i> entre ActiveBCDM e ActiveBCDM+S0 para 23 problemas das Tabelas 3 e 4	95
Figura 22 – <i>Performance profiles</i> entre os métodos quase-Newton (esquerda) e entre o método quase-Newton com melhor desempenho e ActiveBCDM+S0 (direita) para 23 problemas das Tabelas 3 e 4	95
Figura 23 – <i>Performance profiles</i> entre os métodos FCDv.1 e FCDv.2 (esquerda) e entre o melhor dos métodos FCD e ActiveBCDM+S0 (direita) para 23 problemas das Tabelas 3 e 4	96
Figura 24 – <i>Performance profiles</i> mais significativos entre as 24 variantes do método BCDM+IF para os dados da Tabela 1, em FORTRAN	97
Figura 25 – <i>Performance profiles</i> mais significativos entre as 24 variantes do método BCDM+ST para os problemas da Tabela 1, em FORTRAN	98
Figura 26 – <i>Performance profiles</i> mais significativos entre as 24 variantes do método UBCDM para os problemas da Tabela 1, em FORTRAN	98
Figura 27 – <i>Performance profile</i> entre BCDM+IF e BCDM+ST para os problemas da Tabela 1, em FORTRAN	99
Figura 28 – <i>Performance profile</i> entre UBCDM e ActiveBCDM para os problemas da Tabela 1, em FORTRAN	99
Figura 29 – <i>Performance profiles</i> mais significativos entre as 6 variantes do método ActivePCDM-1TH para 18 problemas das Tabelas 3 e 4, em FORTRAN	101
Figura 30 – <i>Performance profiles</i> entre ActivePCDM-1TH , PUBCDM-1TH e PBCDM1-1TH dois a dois para 31 problemas das Tabelas 3 e 4, em FORTRAN	106
Figura 31 – <i>Performance profiles</i> entre ActivePCDM-2TH , PUBCDM-2TH e PBCDM1-2TH dois a dois para 31 problemas das Tabelas 3 e 4, em FORTRAN	107
Figura 32 – <i>Performance profiles</i> entre ActivePCDM-4TH , PUBCDM-4TH e PBCDM1-4TH dois a dois para 31 problemas das Tabelas 3 e 4, em FORTRAN	108
Figura 33 – <i>Performance profiles</i> entre ActivePCDM-8TH , PUBCDM-8TH e PBCDM1-8TH dois a dois para 31 problemas das Tabelas 3 e 4, em FORTRAN	109
Figura 34 – <i>Performance profiles</i> do método ActivePCDM (figura à esquerda) e o método PUBCDM (figura à direita), variando o número de <i>threads</i> para 15 problemas da Tabela 3, em FORTRAN	109

Figura 35 – <i>Performance profiles</i> do método ActivePCDM (figura à esquerda) e o método PUBCDM (figura à direita), variando o número de <i>threads</i> para 16 problemas da Tabela 4, em FORTRAN	110
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----

Lista de tabelas

Tabela 1 – Dados testados no problema LASSO com restrições não negativas . . .	75
Tabela 2 – Escolhas usadas em todos os algoritmos do texto para calibragem dos parâmetros δ_{DP} e δ_F	76
Tabela 3 – Problemas usados nos testes com matriz possuindo mais linhas do que colunas	80
Tabela 4 – Problemas usados nos testes com matriz possuindo mais colunas do que linhas	81
Tabela 5 – Problemas gerados aleatoriamente	102
Tabela 6 – Desempenho dos métodos em paralelo para o problema <i>AL1</i> , com percentual $\omega/n = 0.0005\%$, para um número fixo de iterações, com 1, 2, 4, e 8 <i>threads</i>	102
Tabela 7 – Desempenho dos métodos em paralelo para o problema <i>AL2</i> , com percentual $\omega/n = 0.00038\%$, para um número fixo de iterações, com 1, 2, 4, e 8 <i>threads</i>	103
Tabela 8 – Desempenho dos métodos em paralelo para o problema <i>AL3</i> , com percentual $\omega/n = 0.0003\%$, para um número fixo de iterações, com 1, 2, 4, e 8 <i>threads</i>	103
Tabela 9 – Desempenho dos métodos em paralelo para o problema <i>SL17</i> , com percentual $\omega/n = 2.86\%$, para um número fixo de iterações, com 1, 2, 4, e 8 <i>threads</i>	103
Tabela 10 – Desempenho dos métodos em paralelo para o problema <i>SL20</i> , com percentual $\omega/n = 92.02\%$, para um número fixo de iterações, com 1, 2, 4, e 8 <i>threads</i>	103
Tabela 11 – Desempenho dos métodos em paralelo para o problema <i>SC20</i> , com percentual $\omega/n = 0.18\%$, para um número fixo de iterações, com 1, 2, 4, e 8 <i>threads</i>	103
Tabela 12 – Desempenho dos métodos em paralelo para o problema <i>SC25</i> , com percentual $\omega/n = 10.58\%$, para um número fixo de iterações, com 1, 2, 4, e 8 <i>threads</i>	103
Tabela 13 – Resultados da aceleração real <i>versus</i> aceleração teórica e aceleração com respeito ao tempo para o problema <i>SL17</i>	105
Tabela 14 – Resultados da aceleração real <i>versus</i> aceleração teórica e aceleração com respeito ao tempo para o problema <i>SL20</i>	105
Tabela 15 – Resultados da aceleração real <i>versus</i> aceleração teórica e aceleração com respeito ao tempo para o problema <i>SC20</i>	105

Tabela 16 – Resultados da aceleração real <i>versus</i> aceleração teórica e aceleração com respeito ao tempo para o problema <i>SC25</i>	105
--------------------------------------------------------------------------------------------------------------------------------------------------------	-----

Lista de Símbolos

$\text{cl}(\mathcal{C})$ representa o fecho do conjunto \mathcal{C} .

$\text{int}(\mathcal{C})$ representa o interior do conjunto \mathcal{C} .

$\text{bdry}(\mathcal{C})$ representa a fronteira do conjunto \mathcal{C} .

$|\mathcal{C}|$ representa a cardinalidade do conjunto \mathcal{C} .

$\mathcal{B}(x, \epsilon)$ representa a bola Euclidiana aberta de centro x e raio ϵ .

$\mathcal{C} + \mathcal{B}_\epsilon$ representa a união de todas as bolas Euclidianas abertas de raio ϵ centradas em cada um dos pontos de \mathcal{C} .

$\text{dist}\{y, \mathcal{C}\}$ representa a distância entre o ponto y e o conjunto convexo \mathcal{C} .

\mathbb{N} representa o conjunto $\mathbb{N} = \{1, 2, 3, \dots\}$.

\mathbb{R}_+ representa o conjunto dos números reais não negativos.

\mathbb{R}_{++} representa o conjunto dos números reais positivos.

\mathbb{S}_{++}^n representa o conjunto das matrizes simétricas definidas positivas de dimensão $n \times n$.

$\text{conv}(\mathcal{C})$ representa o fecho convexo de um conjunto finito de vetores \mathcal{C} , definido por

$$\text{conv}(\mathcal{C}) = \left\{ v = \sum_{i=1}^{|\mathcal{C}|} \lambda_i v_i \mid \sum_{i=1}^{|\mathcal{C}|} \lambda_i = 1, \lambda_i \geq 0, v_i \in \mathcal{C} \right\}.$$

$[n]$ denota o conjunto das partes de 1 a n .

a_{i*} denota a i -ésima linha da matriz $A \in \mathbb{R}^{s \times n}$.

a_{*i} denota a i -ésima coluna da matriz $A \in \mathbb{R}^{s \times n}$.

Sumário

1	INTRODUÇÃO	16
2	MÉTODOS DE DESCENSO COORDENADO POR BLOCOS	19
2.1	Métodos de Descenso Coordenado por Blocos	19
2.2	Métodos de Descenso Coordenado por Blocos em Paralelo	30
3	RESULTADOS DE CONVERGÊNCIA	42
3.1	<i>Active BCDM</i>	42
3.2	<i>Active PCDM</i>	44
4	IDENTIFICAÇÃO DAS RESTRIÇÕES ATIVAS	52
4.1	Identificação das restrições ativas para pontos não degenerados	52
4.2	Funções Identificadoras	62
4.2.1	Exemplo de Função Identificadora	64
5	TESTES COMPUTACIONAIS	70
5.1	Problema 1: LASSO	70
5.2	Problema 2: Regressão logística com regularização ℓ_1	71
5.3	Testes em MATLAB	72
5.3.1	Escolha do Conjunto \mathcal{J}	73
5.3.2	LASSO	79
5.3.2.1	Conjunto de dados e critério de parada	79
5.3.2.2	Otimização extra no subspaço das restrições inativas e seleção de parâmetros	80
5.3.2.3	Comparação com outros métodos	87
5.3.3	Regressão logística com regularização ℓ_1	89
5.3.3.1	Conjunto de dados e critério de parada	89
5.3.3.2	Otimização extra no subspaço das restrições inativas e seleção de parâmetros	90
5.3.3.3	Comparação com outros métodos	90
5.4	Testes em FORTRAN	93
5.4.1	Escolha do conjunto \mathcal{J}	95
5.4.2	Testes <i>Active PCDM</i>	97
5.4.2.1	Testando a qualidade da nossa implementação em paralelo	101
5.4.2.2	Testes de desempenho	104
6	CONSIDERAÇÕES FINAIS	111

REFERÊNCIAS	113
APÊNDICES	117
APÊNDICE A – RESULTADOS SOBRE PROBABILIDADE	118

1 Introdução

Nos últimos anos, com a evolução tecnológica, diversas áreas tiveram o volume de informações disponíveis e necessárias de serem avaliadas significativamente aumentado, isso interferiu também nos problemas de otimização. Em vista disso, a busca por estratégias que consigam resolver problemas de porte enorme, eficientemente, se faz cada vez mais necessária. Um dos métodos que parecem se adequar bem à resolução de problemas desse porte são os Métodos de Descenso Coordenado por Blocos (MDCB).

Apesar dos métodos de descenso coordenado se enquadrarem dentre as classes de métodos mais antigos estudados em otimização, seu estudo foi deixado de lado durante algum tempo devido à reduzida eficiência desse tipo de algoritmo. O interesse no estudo desses métodos tem sido renovado pela crescente demanda de problemas de porte enorme e que se contentam com soluções com baixa precisão, vindos de diferentes áreas como: *matrix completion* [13], *compressed sensing* [18], Biologia [25, 30], *machine learning* [29], Estatística [47, 48], *truss topology design* [35] e *group Lasso* [49], para citar alguns exemplos.

Em problemas de porte enorme, a simples tarefa de calcular o valor da função ou o seu gradiente, em um determinado ponto, requer um alto esforço computacional. Os MDCB se adequam a esse tipo de aplicação, visto que a ideia por trás desses métodos é obter sucessivas direções de descida em espaços de dimensão menor do que a dimensão do problema, diminuindo o custo computacional por iteração.

Traremos um breve panorama geral sobre algumas pesquisas recentes desenvolvidas na área de otimização envolvendo métodos de descenso coordenado.

Em [2], Beck e Tetruashvili apresentam um método com escolhas cíclicas de coordenadas, relevante pelos experimentos numéricos envolvendo reconstrução de imagens. Tseng e Yun em [43], se destacam por trazerem três estratégias de escolhas de blocos de coordenadas determinísticas, porém não cíclicas. Referente à escolha dos blocos de coordenadas de maneira aleatória, citamos dois artigos dos autores Richtárik e Takáč [36, 37], com destaque ao segundo, pois possui uma análise mais profunda, englobando a do primeiro artigo. Tal análise envolve a paralelização dos métodos de descenso coordenado, por meio da introdução de uma nova classe de funções, denominada por eles como parcialmente separáveis, relaxando a ideia de separabilidade por coordenadas de uma função. Os artigos [15, 41] abordam versões inexatas dos métodos de descenso coordenado, o primeiro trata de uma versão em paralelo e com escolha híbrida dos blocos, o segundo considera uma versão serial e a atualização aleatória dos blocos. Também ressaltamos o método apresentado em [38], no qual, além dos blocos serem atualizados de maneira cíclica, são usadas estratégias para identificar as coordenadas nulas do problema *LASSO* [42], acrônimo da expressão em

inglês *Least Absolute Shrinkage and Selection Operator*.

Uma última e importante referência sobre métodos de descenso coordenado pode ser encontrada em [45]. Esse é um *survey*, publicado em 2015, que traz uma visão geral sobre esse assunto, mencionando artigos que apresentam aplicações desses métodos em áreas como: *machine learning* e *compressed sensing*, bem como textos que exploram diferentes variantes desses métodos: Cíclica, Estocástica, Aleatória, Acelerada e Paralela. Além disso, nesse texto são apontados e incluídos alguns resultados de convergência que podem ser encontrados na literatura especializada nessa classe de métodos.

Nosso objetivo nesse trabalho é acelerar os métodos de descenso coordenado por blocos, por meio do uso de estratégias de identificações das restrições ativas, particularmente, para problemas de porte enorme com restrições de caixa.

A justificativa para a possibilidade de aceleração provém da existência de problemas tais que a dimensão da face ótima tem a tendência de ser significativamente menor do que a dimensão do problema, como nos problemas descritos em [25, 30, 47, 48]. Além disso, existem problemas irrestritos, tais como *LASSO* [42] e regressão logística com regularização ℓ_1 [33], cujas coordenadas nulas de uma solução podem ser descritas como as restrições ativas de uma reformulação equivalente, com variáveis não negativas. Os bons resultados práticos obtidos pelos MDCB, aplicados a problemas de minimização irrestrita com estrutura separável, também servem de motivação para o estudo dos MDCB em problemas de minimização com restrições.

As restrições em forma de caixa no problema de interesse são justificadas, pois, fixado um ponto na caixa e uma separação das coordenadas do vetor de variáveis, sempre conseguimos uma direção factível no conjunto, para qualquer um dos blocos coordenados, propriedade essa não garantida se tivéssemos um conjunto de restrições mais geral.

Este texto está dividido em 4 capítulos. Descreveremos, brevemente, o conteúdo de cada um deles, destacando as nossas contribuições, a seguir.

No Capítulo 2, apresentamos duas versões de métodos de descenso coordenado, serial e paralelo. Dentro da literatura dos métodos de descenso coordenado, podemos encontrar uma grande variedade de estratégias para atualização dos blocos, dentre elas citamos alguns artigos que usam estratégias determinísticas: métodos cíclicos [38, 9], quase-cíclicos [43], regra de *Gauss-Southwell-q* [43]; e estratégias não determinísticas: aleatória com distribuição de probabilidade uniforme [36], aleatória com distribuição de probabilidade não uniforme e fixa durante o método [31]. Nossa contribuição encontra-se em construir um método que escolhe os blocos de maneira aleatória, com distribuição de probabilidade não uniforme e variável durante o método. Enquanto os resultados teóricos para o caso serial seguem como adaptações simples de resultados desenvolvidos na literatura, no caso paralelo, conseguimos desenvolver fórmulas para adaptar nosso modelo

para o cálculo das direções de descida em paralelo, Corolário 2.1.

O Capítulo 3 aborda resultados de convergência e complexidade para os métodos de descenso coordenado estudados no texto. Apresentamos um resultado de convergência para nosso método de descenso coordenado, Teorema 3.1, em vista que não encontramos na literatura nenhum resultado de convergência que abrangesse o caso dos blocos serem atualizados de maneira não uniforme e com probabilidade variável durante o método. Outro ponto relevante é que nosso resultado de convergência pode ser trivialmente adaptado para qualquer distribuição de probabilidade que atribua uma probabilidade não nula para todos os blocos de coordenadas durante o método. Nossos resultados de complexidade foram extraídos do artigo [37] e envolvem a probabilidade para a qual, com um certo número de iterações, os nossos métodos atingem uma ϵ -precisão no valor de função, Teorema 3.2 e Corolário 3.1.

A primeira novidade presente no Capítulo 4 envolve a construção da Proposição 4.4, baseada na abordagem de identificação das restrições ativas do artigo [12], garantindo que nosso algoritmo consegue identificar as restrições ativas para nosso problema específico com probabilidade 1. A segunda novidade vem da construção de uma função identificadora, função auxiliar capaz de encontrar as restrições ativas de um problema de minimização com restrições, cuja estrutura do problema é formada pela soma de uma função suave e uma função convexa, possivelmente não suave, de estrutura separável por blocos.

No Capítulo 5, construímos métodos de descenso coordenado por blocos aplicados a problemas de regularização ℓ_1 , explorando a informação das coordenadas nulas de uma solução do problema, visto que essa regularização é introduzida com o intuito de gerar uma solução esparsa, verificando as restrições ativas de uma formulação equivalente com variáveis não negativas. Encontramos alguns artigos na literatura que também buscam informação das coordenadas nulas olhando para a formulação equivalente, por exemplo [38, 39, 44], porém exploramos de maneira diferenciada a forma de classificar e atualizar as coordenadas não nulas. Realizamos diversos experimentos numéricos comparando nossos métodos de descenso coordenado com outros métodos desenvolvidos na literatura recente para os problemas *LASSO* e regressão logística com regularização ℓ_1 .

2 Métodos de Descenso Coordenado por Blocos

Nesse capítulo, apresentaremos os métodos de descenso coordenados, o problema de minimização a que eles se aplicam, hipóteses feitas sobre esse problema para aplicação do método, e algumas propriedades relacionadas aos métodos, tanto para o caso serial (Seção 2.1), quanto para o caso paralelo (Seção 2.2).

2.1 Métodos de Descenso Coordenado por Blocos

Consideremos o seguinte problema

$$\begin{aligned} \min_x \quad & F(x) = f(x) + \psi(x) \\ \text{s.a.} \quad & x \in \mathcal{X} \end{aligned} \quad (2.1)$$

em que $f, \psi : \mathbb{R}^n \rightarrow \mathbb{R}$ e o conjunto \mathcal{X} é dado por

$$\mathcal{X} = \{x \in \mathbb{R}^n \mid l \leq x \leq u\}, \text{ para } l, u \in \mathbb{R}^n, \quad (2.2)$$

permitindo possivelmente alguns $l_i = -\infty$ ou $u_i = +\infty$. Vamos considerar que f é uma função de classe C^1 e que ψ é uma função convexa, possivelmente não suave, com estrutura separável por blocos dada por

$$\psi(x) = \sum_{i=1}^m \psi_i(x_{(i)}),$$

em que $m \leq n$, e $x_{(i)}$ contém as coordenadas do vetor x segundo alguma separação da forma

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \sum_{i=1}^m U_i x_{(i)}, \text{ com } \begin{pmatrix} U_1 & U_2 & \cdots & U_m \end{pmatrix} P = I,$$

sendo $I \in \mathbb{R}^{n \times n}$ a matriz identidade de ordem n , $P \in \mathbb{R}^{n \times n}$ uma matriz de permutação e $U_i \in \mathbb{R}^{n \times p_i}$, com p_i representando a quantidade de coordenadas do vetor $x_{(i)}$.

Vamos exemplificar essa ideia de separação das coordenadas do vetor x .

Exemplo 2.1. Suponhamos que $x \in \mathbb{R}^8$, os blocos sejam de tamanho 2, isto é $m = 4$, $p_i = 2$, para todo $i \in \{1, \dots, 4\}$ e compostos pelos pares de coordenadas $(2, 6), (1, 3), (4, 8), (5, 7)$,

organizados nessa ordem. Assim, temos que

$$x_{(1)} = \begin{pmatrix} x_2 \\ x_6 \end{pmatrix}; \quad x_{(2)} = \begin{pmatrix} x_1 \\ x_3 \end{pmatrix}; \quad x_{(3)} = \begin{pmatrix} x_4 \\ x_8 \end{pmatrix}; \quad x_{(4)} = \begin{pmatrix} x_5 \\ x_7 \end{pmatrix}.$$

Agora vamos representar as matrizes U_i , para todo $i \in \{1, \dots, m\}$ e P ,

$$U_1 = \begin{pmatrix} e_2 & e_6 \end{pmatrix}; \quad U_2 = \begin{pmatrix} e_1 & e_3 \end{pmatrix}; \quad U_3 = \begin{pmatrix} e_4 & e_8 \end{pmatrix}; \quad U_4 = \begin{pmatrix} e_5 & e_7 \end{pmatrix}.$$

$$P = \begin{pmatrix} e_3 & e_1 & e_4 & e_5 & e_7 & e_2 & e_8 & e_6 \end{pmatrix},$$

em que e_i são os vetores canônicos do \mathbb{R}^8 .

Observação 2.1. Tentando diferenciar os índices das variáveis, no sentido de distinguir quando estamos falando de uma coordenada de um vetor ou de um dos blocos associados à variável, convencionaremos: para as variáveis x, y, l, u , que trataremos em alguns momentos de suas coordenadas e em outros de seus blocos, i fará referência às coordenadas e (i) aos blocos. Para aquelas variáveis que durante todo o texto usamos o índice referindo-se somente aos blocos, como é o caso de $h_i, B_i, U_i, \psi_i, w_i, p_i, s_i, L_i, \nabla_i f$, buscando simplificar a notação, faremos referência aos blocos escrevendo apenas i .

O gradiente parcial de f com relação ao bloco de coordenadas dado por $x_{(i)}$ será denotado por

$$\nabla_i f(x) = U_i^T \nabla f(x) \in \mathbb{R}^{p_i}.$$

Para cada espaço \mathbb{R}^{p_i} fixamos as normas

$$\|x_{(i)}\|_{(i)} = \sqrt{x_{(i)}^T B_i x_{(i)}}, \quad \text{com } B_i \in \mathbb{S}_{++}^{p_i}$$

e sua respectiva norma dual

$$\|x_{(i)}\|_{(i)}^* = \sqrt{x_{(i)}^T B_i^{-1} x_{(i)}}.$$

As matrizes B_i definidas anteriormente são importantes, porque podemos usá-las para capturar alguma característica da função objetivo envolvendo informação sobre a hessiana da parte suave. Outra aplicabilidade, é usá-la para fazer um escalamento nos blocos de variáveis do problema, por exemplo, para capturar a geometria do conjunto viável, dando pesos diferentes às coordenadas, baseados na viabilidade primal $|u_i - l_i|$. Caso não se opte por explorar tais elementos, B_i pode ser tomada simplesmente como a identidade da ordem adequada.

Assumiremos que a função gradiente de f é Lipschitz contínua em cada um dos seus blocos de coordenadas, com constantes L_i , ou seja

$$\|\nabla_i f(x + U_i h_i) - \nabla_i f(x)\|_{(i)}^* \leq L_i \|h_i\|_{(i)}, \quad h_i \in \mathbb{R}^{p_i}, \quad i = 1, \dots, m. \quad (2.3)$$

Usando (2.3) podemos mostrar que

$$f(x + U_i h_i) \leq f(x) + \nabla_i f(x)^T h_i + \frac{L_i}{2} \|h_i\|_{(i)}^2, \quad (2.4)$$

uma demonstração dessa desigualdade pode ser encontrada em [32, Teorema 2.1.5]. Aplicando a relação (2.4), obtemos

$$\begin{aligned} F(x + U_i h_i) &= f(x + U_i h_i) + \sum_{j \neq i} \psi_j(x_{(j)}) + \psi_i(x_{(i)} + h_i) \\ &\leq f(x) + \nabla_i f(x)^T h_i + \frac{L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)} + h_i) - \psi_i(x_{(i)}) + \sum_{j=1}^m \psi_j(x_{(j)}) \\ &= F(x) + \nabla_i f(x)^T h_i + \frac{L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)} + h_i) - \psi_i(x_{(i)}). \end{aligned} \quad (2.5)$$

A partir de (2.5), notamos que é possível obter limitantes superiores para o valor do decréscimo $F(x + U_i h_i) - F(x)$, utilizando apenas termos que dependem do vetor h_i , de dimensão p_i . Portanto, uma boa escolha para direção de descida a partir do ponto x para o i -ésimo bloco de coordenadas seria

$$h_i(x) = \underset{x + h \in \mathcal{X}}{\operatorname{argmin}} \left\{ \nabla_i f(x)^T h_i + \frac{L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)} + h_i) - \psi_i(x_{(i)}) \right\}, \quad (2.6)$$

com $h = U_i h_i$. É por meio dessa expressão que diversos outros métodos de descenso coordenado por blocos obtêm direções de descida para seus problemas de interesse, como por exemplo, ver [36] e [43].

Observação 2.2. *Notemos que o termo $\psi_i(x_{(i)})$ é desnecessário para a minimização da função objetivo dada por (2.6). Este termo foi preservado apenas por um detalhe técnico, que ficará claro no decorrer do texto.*

Observação 2.3. *O bom funcionamento do método está condicionado à capacidade de resolver os sucessivos problemas (2.6) com simplicidade e eficiência. Mostraremos a seguir, no Exemplo 2.3, uma aplicação em que essas duas características estão presentes.*

Antes de apresentarmos o exemplo em que a direção de descida (2.6) pode ser facilmente calculada, traremos um exemplo que servirá de suporte para ele.

Exemplo 2.2. (Operador Soft-Thresholding) *Considere o problema de minimização*

$$\min_y \frac{1}{2} \|y - z\|_2^2 + \lambda \|y\|_1, \quad \lambda > 0, \quad (2.7)$$

com $z \in \mathbb{R}^n$ fixo. Então, o vetor y , solução do problema (2.7), tem coordenadas dadas por

$$y_i = \begin{cases} z_i - \lambda, & \text{se } z_i > \lambda; \\ z_i + \lambda, & \text{se } z_i < -\lambda; \\ 0, & \text{caso contrário.} \end{cases}$$

O operador $T(z) = y$, com y dado pela expressão anterior, é chamado Operador Soft-Thresholding.

Para mostrar esse fato, vamos usar o conceito de minimização sem restrições, porém para o leitor familiarizado com o subdiferencial da função $t(y_i) = |y_i|$, as coordenadas do vetor y podem ser facilmente obtidas usando as equações KKT do problema (2.7) que dependem do cálculo do subdiferencial da função módulo.

Como a função objetivo do problema (2.7) é separável por coordenadas, isto é, pode ser escrita como uma soma de n funções dependendo das coordenadas y_i , a saber $1/2(y_i - z_i)^2 + \lambda|y_i|$, podemos encontrar o minimizador do problema (2.7) resolvendo os n problemas $h(y_i) = 1/2(y_i - z_i)^2 + \lambda|y_i|$, $i = 1, \dots, n$, separadamente.

Primeiramente, vamos reescrever as funções $h(y_i)$ anteriores da forma

$$h(y_i) = \begin{cases} h_1(y_i) = 1/2(y_i - z_i)^2 + \lambda y_i, & \text{se } y_i \geq 0; \\ h_2(y_i) = 1/2(y_i - z_i)^2 - \lambda y_i, & \text{se } y_i \leq 0. \end{cases}$$

Para cada problema na variável y_i , $i = 1, \dots, n$, vamos resolvê-lo separando-o em três situações:

Caso 1 ($z_i > \lambda$):

Neste caso, podemos descobrir em quais das duas funções $h_1(y_i)$ ou $h_2(y_i)$ o menor valor da função $h(y_i)$ pertence.

Como $(\lambda - z_i) < 0$ e para $y_i \geq 0$, vemos que

$$\begin{aligned} h(y_i) = h_1(y_i) &= 1/2(y_i - z_i)^2 + \lambda y_i \\ &= 1/2(y_i^2 + z_i^2) + y_i(\lambda - z_i) \\ &\leq 1/2(|y_i|^2 + z_i^2). \end{aligned}$$

Agora, para $y_i \leq 0$, temos que

$$\begin{aligned} h(y_i) = h_2(y_i) &= 1/2(y_i - z_i)^2 + \lambda y_i \\ &= 1/2(y_i^2 + z_i^2) - y_i(\lambda + z_i) \\ &\geq 1/2(|y_i|^2 + z_i^2). \end{aligned}$$

Portanto, sabemos que o minimizador para o problema $h(y_i)$ para $z_i > \lambda$ se encontra entre os valores não negativos de y_i . Logo,

$$\begin{aligned} \min_{y_i} h(y_i) &\Leftrightarrow \min_{y_i} h_1(y_i). \\ &\text{s.a. } y_i \geq 0 \end{aligned}$$

Como o único minimizador global para o problema irrestrito $\min_{y_i} h_1(y_i)$ é $y_i = z_i - \lambda > 0$, vemos que esse também é o único minimizador para o problema com variáveis não negativas e, por consequência, para o problema $\min_{y_i} h(y_i)$.

Caso 2 ($z_i < -\lambda$):

Analogamente ao Caso 1, vemos, para $y_i \geq 0$, que

$$\begin{aligned} h(y_i) = h_1(y_i) &= 1/2(y_i - z_i)^2 + \lambda y_i \\ &= 1/2(y_i^2 + z_i^2) - y_i(z_i - \lambda) \\ &\geq 1/2(|y_i|^2 + z_i^2). \end{aligned}$$

Agora, para $y_i \leq 0$, temos que

$$\begin{aligned} h(y_i) = h_2(y_i) &= 1/2(y_i - z_i)^2 + \lambda y_i \\ &= 1/2(y_i^2 + z_i^2) - y_i(\lambda + z_i) \\ &\leq 1/2(|y_i|^2 + z_i^2). \end{aligned}$$

Portanto, sabemos que o minimizador para o problema $h(y_i)$ para $z_i < -\lambda$ se encontra entre os valores não positivos de y_i . Logo,

$$\begin{aligned} \min_{y_i} h(y_i) &\Leftrightarrow \min_{y_i} h_2(y_i). \\ &\text{s.a. } y_i \leq 0 \end{aligned}$$

Desde que o único minimizador global para o problema irrestrito $\min_{y_i} h_2(y_i)$ é $y_i = z_i + \lambda < 0$, obtemos que esse também é o único minimizador para o problema com variáveis não positivas e, portanto, para o problema $\min_{y_i} h(y_i)$.

Caso 3 ($-\lambda \leq z_i \leq \lambda$):

Se $-\lambda \leq z_i \leq \lambda$, temos que

$$\begin{cases} -v\lambda \leq vz_i \leq v\lambda, & \text{se } v \geq 0; \\ -v\lambda \geq vz_i \geq v\lambda, & \text{caso contrário.} \end{cases} \quad (2.8)$$

Pelas duas desigualdades de (2.8), vemos, para todo $v \in \mathbb{R}$, que

$$\lambda|v| - vz_i \geq 0. \quad (2.9)$$

Usando a expressão (2.9), obtemos que

$$\begin{aligned}
 \frac{1}{2}(v - z_i)^2 + \lambda|v| &= \frac{1}{2}v^2 - vz_i + \frac{1}{2}z_i^2 + \lambda|v| \\
 &= \frac{1}{2}v^2 + \frac{1}{2}z_i^2 + (\lambda|v| - vz_i) \\
 &\stackrel{(2.9)}{\geq} \frac{1}{2}v^2 + \frac{1}{2}z_i^2 \\
 &\geq \frac{1}{2}z_i^2 \\
 &= \frac{1}{2}(0 - z_i)^2 + \lambda|0|.
 \end{aligned} \tag{2.10}$$

Pela expressão (2.10), vemos que $y_i = 0$ é o minimizador global para o problema quando $-\lambda \leq z_i \leq \lambda$, concluindo o desejado.

□

Observação 2.4. O problema (2.7) é muito importante no contexto de métodos proximais, especialmente no método chamado Método Gradiente Proximal, para o leitor interessado no assunto, uma ótima leitura é o survey [34].

Exemplo 2.3. Apresentamos o problema de regressão esparsa sobre uma caixa, cuja função objetivo também é conhecida como LASSO [42], dado por

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1, \quad \lambda > 0, \quad A = [a_{*1} | \cdots | a_{*n}] \in \mathbb{R}^{s \times n}. \tag{2.11}$$

$l \leq x \leq u$

Tomando blocos de tamanho 1, isto é $p_i = 1$ e $B_i = 1$ para todo i , temos que as constantes de Lipschitz para cada bloco são obtidas explicitamente como $L_i = \|a_{*i}\|_2^2$, pois

$$\nabla_i f(x) = a_{*i}^T(Ax - b) \quad \text{e} \quad |\nabla_i f(x + te_i) - \nabla_i f(x)| = |ta_{*i}^T A e_i| = |t| \|a_{*i}\|_2^2,$$

sendo $e_i \in \mathbb{R}^n$ o i -ésimo vetor canônico do \mathbb{R}^n . Para calcular uma direção de descida para o problema (2.11), descrita pela expressão (2.6), devemos encontrar

$$h_i(x) = \underset{l_i \leq x_i + h_i \leq u_i}{\operatorname{argmin}_{h_i}} \left\{ a_{*i}^T(Ax - b)h_i + \frac{L_i}{2}h_i^2 + \lambda|x_i + h_i| \right\}. \tag{2.12}$$

Primeiramente, vamos analisar as soluções do problema irrestrito, isto é,

$$j_i(x) = \operatorname{argmin}_{h_i} \left\{ a_{*i}^T(Ax - b)h_i + \frac{L_i}{2}h_i^2 + \lambda|x_i + h_i| \right\}. \tag{2.13}$$

Para encontrarmos os minimizadores do problema (2.13), notamos que o problema

$$j_i(x) = \operatorname{argmin}_{h_i} \left\{ \frac{L_i}{2}(h_i + \frac{1}{L_i}a_{*i}^T(Ax - b))^2 + \lambda|x_i + h_i| \right\}, \tag{2.14}$$

é equivalente ao problema (2.13). Para ver isso, basta reparar que os dois problemas têm função objetivo diferindo por uma constante, $[a_{*i}^T(Ax - b)]^2 / (2L_i)$.

Notemos que o problema (2.14) pode ser reescrito da forma $j_i(x) = y_i - x_i$, y_i solução do problema de minimização

$$y_i = \operatorname{argmin}_z \left\{ \frac{1}{2} \left(z - \left(x_i - \frac{1}{L_i} a_{*i}^T(Ax - b) \right) \right)^2 + \frac{\lambda}{L_i} |z| \right\}, \quad (2.15)$$

cuja solução foi obtida no Exemplo 2.2, e é dada por

$$y_i = \begin{cases} x_i - \frac{1}{L_i} a_{*i}^T(Ax - b) - \frac{\lambda}{L_i}, & \text{se } x_i - \frac{1}{L_i} a_{*i}^T(Ax - b) > \frac{\lambda}{L_i}; \\ x_i - \frac{1}{L_i} a_{*i}^T(Ax - b) + \frac{\lambda}{L_i}, & \text{se } x_i - \frac{1}{L_i} a_{*i}^T(Ax - b) < -\frac{\lambda}{L_i}; \\ 0, & \text{caso contrário.} \end{cases}$$

Pela expressão anterior, vemos que $j_i(x)$ é descrito pela fórmula

$$j_i(x) = \begin{cases} -\frac{1}{L_i} a_{*i}^T(Ax - b) - \frac{\lambda}{L_i}, & \text{se } x_i - \frac{1}{L_i} a_{*i}^T(Ax - b) > \frac{\lambda}{L_i}; \\ -\frac{1}{L_i} a_{*i}^T(Ax - b) + \frac{\lambda}{L_i}, & \text{se } x_i - \frac{1}{L_i} a_{*i}^T(Ax - b) < -\frac{\lambda}{L_i}; \\ -x_i, & \text{caso contrário.} \end{cases} \quad (2.16)$$

Por meio da solução do problema irrestrito, $j_i(x)$, e pelo fato da função objetivo do problema (2.12) ser uma função quadrática convexa, podemos facilmente obter a solução do problema com restrições a partir da solução do problema irrestrito $j_i(x)$, conforme a descrição a seguir.

Se $l_i \geq 0$ temos que a primeira situação (2.16) ocorre para todo h_i que mantém a viabilidade, assim

$$h_i(x) = \begin{cases} \frac{-a_{*i}^T(Ax - b) - \lambda}{L_i}, & \text{se } l_i < x_i - \frac{a_{*i}^T(Ax - b) + \lambda}{L_i} < u_i; \\ l_i - x_i, & \text{se } x_i - \frac{a_{*i}^T(Ax - b) + \lambda}{L_i} \leq l_i; \\ u_i - x_i, & \text{caso contrário.} \end{cases} \quad (2.17)$$

Quando $u_i \leq 0$, temos que a segunda situação da expressão (2.16) acontece para todo h_i que mantém a viabilidade, logo

$$h_i(x) = \begin{cases} \frac{-a_{*i}^T(Ax - b) + \lambda}{L_i}, & \text{se } l_i < x_i - \frac{a_{*i}^T(Ax - b) - \lambda}{L_i} < u_i; \\ l_i - x_i, & \text{se } x_i - \frac{a_{*i}^T(Ax - b) - \lambda}{L_i} \leq l_i; \\ u_i - x_i, & \text{caso contrário.} \end{cases} \quad (2.18)$$

Caso $l_i < 0 < u_i$, sabemos que

$$h_i(x) = \begin{cases} \frac{-a_{*i}^T(Ax - b) - \lambda}{L_i}, & \text{se } 0 < x_i - \frac{a_{*i}^T(Ax - b) + \lambda}{L_i} < u_i; \\ u_i - x_i, & \text{se } x_i - \frac{a_{*i}^T(Ax - b) + \lambda}{L_i} \geq u_i; \\ \frac{-a_{*i}^T(Ax - b) + \lambda}{L_i}, & \text{se } l_i < x_i - \frac{a_{*i}^T(Ax - b) - \lambda}{L_i} < 0; \\ l_i - x_i, & \text{se } x_i - \frac{a_{*i}^T(Ax - b) - \lambda}{L_i} \leq l_i; \\ -x_i, & \text{caso contrário.} \end{cases} \quad (2.19)$$

□

Dessa forma, as expressões (2.17), (2.18) e (2.19) nos fornecem fórmulas fechadas para a solução do problema (2.12), quando escolhemos blocos de tamanho 1.

Vamos mostrar agora um resultado que quantifica o decréscimo obtido no valor de função a partir do cálculo de uma direção de descida, descrita por meio da expressão (2.6).

Lema 2.1. *Seja x um ponto viável para o problema (2.1). Então,*

$$F(x + U_i h_i(x)) - F(x) \leq -\frac{L_i}{2} \|h_i(x)\|_{(i)}^2. \quad (2.20)$$

Prova: Por simplicidade, denotamos $h_i(x) = h_i$.

Usando a definição de h_i , podemos ver que para todo $\alpha \in (0, 1)$, temos que

$$\nabla_i f(x)^T h_i + \frac{L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)} + h_i) \leq \alpha \nabla_i f(x)^T h_i + \frac{L_i \alpha^2}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)} + \alpha h_i) \quad (2.21)$$

Pela convexidade de $\psi_i(x_{(i)})$, obtemos que

$$\psi_i(x_{(i)} + \alpha h_i) = \psi_i((1 - \alpha)x_{(i)} + \alpha(x_{(i)} + h_i)) \leq (1 - \alpha)\psi_i(x_{(i)}) + \alpha\psi_i(x_{(i)} + h_i) \quad (2.22)$$

Substituindo a expressão (2.22) em (2.21) e agrupando os termos em comum, vemos que

$$\begin{aligned} & \nabla_i f(x)^T h_i + \frac{L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)} + h_i) \\ & \leq \alpha \nabla_i f(x)^T h_i + \frac{L_i \alpha^2}{2} \|h_i\|_{(i)}^2 + \alpha\psi_i(x_{(i)} + h_i) + (1 - \alpha)\psi_i(x_{(i)}) \\ \Leftrightarrow & (1 - \alpha)\nabla_i f(x)^T h_i + \frac{(1 - \alpha)(1 + \alpha)L_i}{2} \|h_i\|_{(i)}^2 + (1 - \alpha)\psi_i(x_{(i)} + h_i) - (1 - \alpha)\psi_i(x_{(i)}) \\ & \leq 0. \end{aligned} \quad (2.23)$$

Dividindo a expressão (2.23) por $(1 - \alpha)$ e passando o resultado ao limite para $\alpha \uparrow 1$, temos que

$$\nabla_i f(x)^T h_i + \frac{L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)} + h_i) - \psi_i(x_{(i)}) \leq -\frac{L_i}{2} \|h_i\|_{(i)}^2. \quad (2.24)$$

Para concluirmos a demonstração, basta substituímos a expressão (2.24) em (2.5). \square

Apresentaremos a seguir algumas definições importantes para a sequência do texto.

Definição 2.1. O **cone tangente** ao conjunto \mathcal{X} no ponto $x^* \in \mathcal{X}$, denotado por $T_{\mathcal{X}}(x^*)$, é o conjunto dos vetores y tais que $\frac{y}{\|y\|} = \lim_{k \rightarrow +\infty} \frac{x^k - x^*}{\|x^k - x^*\|}$, com $x^k \rightarrow x^*$, $x^k \in \mathcal{X}$.

Pela estrutura particular do nosso conjunto de restrições \mathcal{X} ser uma caixa, o cone tangente é equivalente ao conjunto das direções viáveis. Definiremos também a estacionariedade de um ponto do problema (2.1).

Definição 2.2. Um ponto $x^* \in \mathcal{X}$ é um **ponto estacionário** do problema (2.1) se

$$F'(x^*; d) = \lim_{\alpha \downarrow 0} \frac{F(x^* + \alpha d) - F(x^*)}{\alpha} \geq 0, \quad (2.25)$$

para toda direção $d \in T_{\mathcal{X}}(x^*)$.

Até o momento, temos os principais elementos para apresentar um algoritmo de descenso coordenado por blocos. Dado um ponto inicial, sabemos como atualizar os vetores a cada iteração, mas ainda não conhecemos um critério de parada para o mesmo. O próximo resultado busca atender essa necessidade, e foi obtido por meio de pequenas modificações dos resultados de Tseng e Yun [43, Lemas 1 e 2].

Proposição 2.1. Para $h_i(x)$ dado em (2.6), vale a seguinte relação

$$h_i(x) = 0, \forall i \in \{1, \dots, m\} \Leftrightarrow x \text{ é um ponto estacionário do problema (2.1).}$$

Prova: Visando simplificação da notação, chamamos $h_i(x) = h_i$.

(\Leftarrow) Suponha que x é ponto estacionário do problema (2.1) e suponha por absurdo que exista $i \in \{1, \dots, m\}$ tal que $h_i \neq 0$ e $x + U_i h_i \in \mathcal{X}$. Pela minimalidade do vetor h_i em (2.6), temos que

$$\begin{aligned} & \nabla_i f(x)^T h_i + \frac{L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)} + h_i) - \psi_i(x_{(i)}) \leq 0 \\ \Leftrightarrow & \quad \nabla_i f(x)^T h_i + \psi_i(x_{(i)} + h_i) - \psi_i(x_{(i)}) \leq -\frac{L_i}{2} \|h_i\|_{(i)}^2 \end{aligned} \quad (2.26)$$

Temos, pela continuidade da derivada de f , que

$$f(x + \alpha d) = f(x) + \alpha \nabla f(x)^T d + o(\alpha), \quad \forall d \in \mathbb{R}^n, \quad (2.27)$$

em que $\lim_{\alpha \downarrow 0} \frac{o(\alpha)}{\alpha} = 0$.

Utilizando as expressões (2.22), (2.26) e (2.27), para $d = U_i h_i$ e $\alpha \in (0, 1]$, vemos que

$$\begin{aligned} F(x + \alpha d) &= f(x + \alpha U_i h_i) + \sum_{j \neq i} \psi_j(x_{(j)}) + \psi_i(x_{(i)} + \alpha h_i) \\ &= f(x) + \alpha \nabla_i f(x)^T h_i + o(\alpha) + \psi_i(x_{(i)} + \alpha h_i) - \psi_i(x_{(i)}) + \\ &\quad + \sum_{j=1}^m \psi_j(x_{(j)}) \\ &\leq F(x) + \alpha (\nabla_i f(x)^T h_i + \psi_i(x_{(i)} + \alpha h_i) - \psi_i(x_{(i)})) + o(\alpha) \\ &\leq F(x) + \alpha \left(-\frac{L_i}{2} \|h_i\|_{(i)}^2 \right) + o(\alpha). \end{aligned} \quad (2.28)$$

Dividindo a expressão (2.28) por α e passando ao limite para $\alpha \rightarrow 0$, obtemos que

$$F'(x^*; d) = \lim_{\alpha \downarrow 0} \frac{F(x + \alpha d) - F(x)}{\alpha} \leq -\frac{L_i}{2} \|h_i\|_{(i)}^2 + \lim_{\alpha \downarrow 0} \frac{o(\alpha)}{\alpha} = -\frac{L_i}{2} \|h_i\|_{(i)}^2 < 0.$$

contrariando a hipótese de que x é ponto estacionário do problema (2.1), pois mostramos que $U_i h_i$, direção viável, é uma direção de descida para F a partir do ponto x .

(\Rightarrow) Se $h_i = 0$, para todo $i \in \{1, \dots, m\}$. Pela minimalidade dos h_i temos por (2.6) que

$$\begin{aligned} 0 &= \nabla_i f(x)^T h_i + \frac{L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)} + h_i) - \psi_i(x_{(i)}) \\ &\leq \nabla_i f(x)^T d_i + \frac{L_i}{2} \|d_i\|_{(i)}^2 + \psi_i(x_{(i)} + d_i) - \psi_i(x_{(i)}) \\ \Leftrightarrow \psi_i(x_{(i)}) &\leq \nabla_i f(x)^T d_i + \frac{L_i}{2} \|d_i\|_{(i)}^2 + \psi_i(x_{(i)} + d_i), \quad \forall i \in \{1, \dots, m\}, \end{aligned} \quad (2.29)$$

em que d_i são tais que $x + d \in \mathcal{X}$ com $d = \sum_{i=1}^m U_i d_i$. Somando as m desigualdades dadas por (2.29), vemos que

$$\begin{aligned} \sum_{i=1}^m \psi_i(x_{(i)}) &\leq \sum_{i=1}^m \nabla_i f(x)^T d_i + \sum_{i=1}^m \frac{L_i}{2} \|d_i\|_{(i)}^2 + \sum_{i=1}^m \psi_i(x_{(i)} + d_i) \\ \Rightarrow \psi(x) &\leq \nabla f(x)^T d + L \|d\|_B^2 + \psi(x + d), \quad \forall x + d \in \mathcal{X}, \end{aligned} \quad (2.30)$$

em que $L = \max \left\{ \frac{L_i}{2} \mid i \in \{1, \dots, m\} \right\}$ e $\|x\|_B^2 = x^T B x$, com B uma matriz diagonal por blocos com o i -ésimo bloco diagonal dado pela matriz B_i que define a norma $\|\cdot\|_{(i)}^2$.

Para qualquer $d \in \mathbb{R}^n$ tal que $x + d \in \mathcal{X}$, temos, para $\alpha \in (0, 1]$, por (2.30) que

$$\psi(x) \leq \alpha \nabla f(x)^T d + \alpha^2 L \|d\|_B^2 + \psi(x + \alpha d). \quad (2.31)$$

Usando (2.27) e (2.31)

$$\begin{aligned} F'(x; d) &= \lim_{\alpha \downarrow 0} \frac{F(x + \alpha d) - F(x)}{\alpha} = \lim_{\alpha \downarrow 0} \frac{f(x + \alpha d) + \psi(x + \alpha d) - f(x) - \psi(x)}{\alpha} \\ &\geq \lim_{\alpha \downarrow 0} \frac{\alpha \nabla f(x)^T d + o(\alpha) - \alpha \nabla f(x)^T d - \alpha^2 L \|d\|_B^2}{\alpha} \\ &= \lim_{\alpha \downarrow 0} \frac{o(\alpha)}{\alpha} - \alpha L \|d\|_B^2 = 0, \quad \forall d \text{ tal que } x + d \in \mathcal{X}. \end{aligned}$$

Logo, x é um ponto estacionário do problema (2.1). \square

Apresentaremos um algoritmo de descenso coordenado por blocos para o problema (2.1), baseado no desenvolvimento dessa seção.

Algoritmo 1: *Active Block Coordinate Descent Method (Active BCDM)*

Input: $x^0 \in \mathcal{X}$, $\delta_{DP}, \delta_F \in \mathbb{N}$, $\epsilon \in \mathbb{R}_+$, $l_{\max} \in \mathbb{N}_+$

Output: x^k

begin

$\mathcal{I} \leftarrow \{1, \dots, m\};$

$\mathcal{J} \leftarrow \emptyset;$

$\ell \leftarrow 1;$

repeat

for $k = (\ell - 1)\delta_F + 1$ **to** $\ell\delta_F$ **do**

 Escolha um bloco $i \in \{1, \dots, m\}$ com a distribuição de probabilidade;

$$P(i) = \begin{cases} \frac{\delta_{DP}}{\delta_{DP}|\mathcal{I}| + |\mathcal{J}|}, & \text{se } i \in \mathcal{I}, \\ \frac{1}{\delta_{DP}|\mathcal{I}| + |\mathcal{J}|}, & \text{se } i \in \mathcal{J}. \end{cases}$$

 Encontre $h_i(x^k)$ solução do problema (2.6);

$x^{k+1} \leftarrow x^k + U_i h_i(x^k);$

$v_{(i)} \leftarrow h_i(x^k);$

end

 Escolha o conjunto $\mathcal{J} \subset \{1, \dots, m\}$ de alguma maneira;

$\mathcal{I} \leftarrow \{1, \dots, m\} - \mathcal{J};$

$\ell \leftarrow \ell + 1;$

until $\|v\| \leq \epsilon$ **or** $\ell = l_{\max};$

end

Observação 2.5. Vemos, pelo Algoritmo 1, que ele define uma família de métodos de descenso coordenado por meio da variação dos parâmetros δ_F, δ_{DP} e da maneira como é

escolhido o conjunto \mathcal{J} . O parâmetro δ_{DP} e o conjunto \mathcal{J} foram escolhidos para permitir que o algoritmo faça escolhas não uniformes dos blocos de coordenadas que serão atualizados durante o método, desde que \mathcal{J} controle quais blocos de coordenadas terão uma distribuição de probabilidade menor de serem escolhidos e δ_{DP} controle o fator de preferência com que os blocos de coordenadas do conjunto \mathcal{I} têm de serem escolhidos, com respeito à distribuição de probabilidade, com relação aos blocos do conjunto \mathcal{J} . Já o parâmetro δ_F controla a frequência de iterações entre a mudança dos índices pertencentes ao conjunto \mathcal{J} .

Observação 2.6. Fazendo as escolhas $\delta_F = +\infty, \delta_{DP} = 1, \mathcal{J} = \{1, \dots, m\}$, para todo $x \in \mathcal{X}$ no Algoritmo 1, temos um método de descenso coordenado por blocos que atualiza todas as coordenadas com distribuição uniforme, o mesmo apresentado em [36] com o nome de UCDC, do inglês *Uniform Coordinate Descent for Composite functions*.

Observação 2.7. Claramente, o bom funcionamento do Algoritmo 1 está inteiramente relacionado com boas escolhas para os parâmetros δ_F, δ_{DP} e para o conjunto \mathcal{J} . Ao longo do texto, mostraremos algumas escolhas que garantirão uma melhora no desempenho do algoritmo, quando comparado à variante mais simples descrita na Observação 2.6, de forma a justificar e compensar a elaborada caracterização desse algoritmo.

2.2 Métodos de Descenso Coordenado por Blocos em Paralelo

Nesta seção, continuamos com as mesmas hipóteses feitas na Seção 2.1 com respeito à separabilidade por blocos de coordenadas da função $\psi(x)$ e a Lipschitz continuidade por blocos, uniforme em x , do gradiente da função f , acrescentando a hipótese de que f é convexa.

Em [37], os autores começam investigando quais propriedades a função objetivo $F(x) = f(x) + \psi(x)$ deveria possuir para que o problema (2.1), em sua versão irrestrita, pudesse ser acelerado com o uso do paralelismo. Com essa pergunta em mente, eles iniciam analisando o caso em que a parte suave da função objetivo $f(x)$ é separável por blocos, da mesma maneira que $\psi(x)$, isto é, $f(x) = \sum_{i=1}^m f_i(x_{(i)})$ com funções suaves $f_i : \mathbb{R}^{p_i} \rightarrow \mathbb{R}$. Nesse caso, é fácil perceber que atualizando os m blocos de coordenadas em paralelo, devemos obter uma aceleração com fator m no método, comparado ao algoritmo de descenso coordenado que atualiza apenas um bloco de coordenadas por iteração.

Baseados em tal observação, eles apresentaram uma classe de funções que relaxa o conceito de separabilidade por coordenadas (total), representado pela função suave anterior, e mostraram quais propriedades a função f precisa possuir para que exista a possibilidade de aceleração do método de descenso coordenado em paralelo. Este conceito pode ser encontrado em [37, Seção 1.5] e é apresentado a seguir.

Definição 2.3. Diremos que uma função suave f é **parcialmente separável de grau ω** se existe um número finito de funções suaves f_J tal que

$$f(x) = \sum_{J \in \mathcal{T}} f_J(x),$$

em que \mathcal{T} é uma coleção finita de subconjuntos do conjunto $\{1, \dots, m\}$ (possivelmente contendo conjuntos idênticos), f_J são funções convexas e suaves tais que f_J depende somente dos blocos de coordenadas $x_{(i)}$ tais que $i \in J$ e

$$|J| \leq \omega, \quad \forall J \in \mathcal{T}.$$

Apresentaremos um exemplo, visando ilustrar o conceito de separação parcial exposto anteriormente.

Exemplo 2.4. Consideremos a função

$$f(x_1, x_2, x_3, x_4, x_5) = (x_1 + 2x_2) + (x_3 + x_4)^2 - (5x_1 + x_3 + 8x_5) + (x_5 + x_4 + x_1)^4 - (x_1 + x_2)^{16}.$$

Nesse caso temos que

$$\mathcal{T} = \{\{1, 2\}, \{3, 4\}, \{1, 3, 5\}, \{1, 4, 5\}, \{1, 2\}\} \text{ e } \omega = 3.$$

Essa definição de separabilidade parcial tem relevância, pois existem alguns exemplos de funções muito utilizadas nesse contexto de otimização de porte enorme que são parcialmente separáveis, e para as quais pode-se obter as funções suaves f_J de maneira simples. Outro fato, que será mostrado posteriormente, é que a aceleração obtida pelo paralelismo depende do grau ω de separabilidade parcial da função f . Nesse contexto, portanto, é fundamental o conhecimento do valor de ω para aplicação do método. Traremos alguns exemplos de funções parcialmente separáveis e seus respectivos valores de ω .

Exemplo 2.5. Consideremos $g_1(x) = \frac{1}{2} \|Ax - b\|_2^2 = \sum_{i=1}^s \frac{1}{2} (a_{i*}^T x - b_i)^2$, $g_2(x) = \sum_{i=1}^s \log(1 + e^{-b_i a_{i*}^T x})$ e $g_3(x) = \sum_{i=1}^s \frac{1}{2} \max\{0, 1 - b_i a_{i*}^T x\}^2$ com $A \in \mathbb{R}^{s \times n}$, $b \in \mathbb{R}^s$, e $x \in \mathbb{R}^n$.

Nos três casos, vemos que, se a matriz A possui uma linha não nula, então temos que $\omega = m$, pois pelo menos uma componente do somatório dependerá de todos os blocos de coordenadas descritos pela função separável $\psi(x)$. No caso em que a função $\psi(x)$ é separável por blocos de tamanho 1, por exemplo $\psi(x) = \|x\|_1$, então $\omega = \max_{i=1, \dots, s} \{\|a_{i*}\|_0\}$, em que $\|v\|_0$ corresponde ao número de elementos não nulos do vetor v .

No artigo [37], os autores obtiveram fórmulas para descrever a aceleração dos métodos de descenso coordenado em paralelo, para blocos escolhidos com algumas

distribuições de probabilidade, dentre elas cabe ressaltar, a distribuição de probabilidade uniforme.

Nosso objetivo, no decorrer desta seção, é construir uma versão em paralelo para o Algoritmo 1. Claramente, não podemos simplesmente substituir a parte em que atualizamos os blocos de coordenadas de maneira serial por uma atualização em paralelo dos blocos, e esperar que o novo algoritmo funcione, visto que para este novo algoritmo não temos a garantia que as direções obtidas sejam de descida, como no caso do Algoritmo 1.

Uma maneira simples de pensar em um algoritmo em paralelo, usando os argumentos da Seção 2.1, seria, dado um número τ de *threads*¹ em que se deseja paralelizar o algoritmo, separar os m blocos de coordenadas em subconjuntos com τ blocos, e para cada atualização desse novo conjunto de blocos, atualizar em paralelo, um bloco de coordenadas para cada *thread*. Porém, para o nosso objetivo, que é permitir a atualização não uniforme dos blocos de coordenadas, essa estratégia geraria alguns empecilhos, pois a cada ciclo de iterações precisaríamos, possivelmente, agrupar diferentes τ blocos de coordenadas e calcular novamente as constantes de Lipschitz de ∇f para cada um desses novos blocos, prejudicando o desempenho do método.

A ideia que usaremos na sequência seguirá o mesmo desenvolvimento feito em [37]. Buscaremos colocar um peso na constante de Lipschitz por blocos, $\beta \geq 1$, da forma

$$h_i^\beta(x) = \underset{x+h \in \mathcal{X}}{\operatorname{argmin}} \left\{ \nabla_i f(x)^T h_i + \frac{\beta L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)} + h_i) - \psi_i(x_{(i)}) \right\}, \quad (2.32)$$

garantindo que a cada iteração do método de descenso coordenado por blocos em paralelo as direções obtidas por ele sejam, na média, direções de descida para o problema (2.1). Uma vantagem dessa abordagem para o nosso método é que não precisamos nos preocupar com quais blocos de coordenadas estão sendo atualizados em paralelo, basta calcularmos as constantes de Lipschitz do ∇f para os m blocos de coordenadas, como no caso serial, bem como a constante β , definida na sequência do texto, e seremos capazes de calcular direções, em média, de descida para o problema de interesse.

Dado um ponto $x \in \mathcal{X}$, o vetor $h^\beta = (h_1^\beta, \dots, h_n^\beta) \in \mathbb{R}^n$ com h_i^β definido pela expressão (2.32) e $\mathcal{S} \subset \{1, \dots, m\}$, uma variável aleatória contendo τ blocos de coordenadas, em que a probabilidade de cada bloco de coordenadas pertencer ao conjunto \mathcal{S} é descrita pela distribuição de probabilidade presente no Algoritmo 1, buscaremos uma fórmula para o parâmetro β de modo que o valor esperado para $F \left(x + \sum_{j \in \mathcal{S}} U_j h_j^\beta \right)$ gerado pelo método

¹ *thread*: forma de um processo se dividir em duas ou mais tarefas, que podem ser executadas concorrentemente.

a partir de x , satisfaça a expressão

$$\begin{aligned} \mathbb{E} \left[F \left(x + \sum_{j \in \mathcal{S}} U_j h_j^\beta \right) \right] &\leq F(x) + \\ &+ \sum_{i \in \mathcal{I}} \frac{\tau \delta_{DP}}{p} \left(\nabla_i f(x)^T h_i^\beta + \frac{\beta L_i}{2} \|h_i^\beta\|_{(i)}^2 + \psi_i(x_{(i)} + h_i^\beta) - \right. \\ &- \psi_i(x_{(i)}) \left. + \sum_{i \in \mathcal{J}} \frac{\tau}{p} \left(\nabla_i f(x)^T h_i^\beta + \frac{\beta L_i}{2} \|h_i^\beta\|_{(i)}^2 + \right. \right. \\ &\left. \left. + \psi_i(x_{(i)} + h_i^\beta) - \psi_i(x_{(i)}) \right) \right], \end{aligned} \quad (2.33)$$

com $p = \delta_{DP}|\mathcal{I}| + |\mathcal{J}|$.

Para obter uma expressão para β em (2.33), vamos, inicialmente, explorar a estrutura separável da função F , da forma

$$\mathbb{E} \left[F \left(x + \sum_{j \in \mathcal{S}} U_j h_j^\beta \right) \right] \stackrel{\text{Lema A.1(iii)}}{=} \mathbb{E} \left[f \left(x + \sum_{j \in \mathcal{S}} U_j h_j^\beta \right) \right] + \mathbb{E} \left[\psi \left(x + \sum_{j \in \mathcal{S}} U_j h_j^\beta \right) \right] \quad (2.34)$$

Sabemos, pela estrutura de $\psi(x)$, que

$$\begin{aligned} \mathbb{E} \left[\psi \left(x + \sum_{j \in \mathcal{S}} U_j h_j^\beta \right) \right] &= \mathbb{E} \left[\sum_{j \in \mathcal{S}} \psi_j(x_{(j)} + h_j^\beta) + \sum_{j \notin \mathcal{S}} \psi_j(x_{(j)}) \right] \\ &= \mathbb{E} \left[\sum_{j \in \mathcal{S}} (\psi_j(x_{(j)} + h_j^\beta) - \psi_j(x_{(j)})) + \sum_{i=1}^m \psi_i(x_{(i)}) \right] \\ &\stackrel{\text{Def. A.1, Lema A.1(iii)}}{=} \sum_{i=1}^m (\psi_i(x_{(i)} + h_i^\beta) - \psi_i(x_{(i)})) \mathbb{P}(i \in \mathcal{S}) + \sum_{i=1}^m \psi_i(x_{(i)}) \\ &= \sum_{i \in \mathcal{I}} \left(\frac{\tau \delta_{DP}}{p} \psi_i(x_{(i)} + h_i^\beta) + (1 - \frac{\tau \delta_{DP}}{p}) \psi_i(x_{(i)}) \right) + \\ &\quad + \sum_{i \in \mathcal{J}} \left(\frac{\tau}{p} \psi_i(x_{(i)} + h_i^\beta) + (1 - \frac{\tau}{p}) \psi_i(x_{(i)}) \right). \end{aligned} \quad (2.35)$$

Substituindo a expressão (2.35) em (2.34) e comparando com a expressão (2.33), lembrando que $F(x) = f(x) + \psi(x)$, vemos que precisamos encontrar uma constante β satisfazendo

$$\begin{aligned} \mathbb{E} \left[f \left(x + \sum_{j \in \mathcal{S}} U_j h_j^\beta \right) \right] &\leq f(x) + \sum_{i \in \mathcal{I}} \frac{\tau \delta_{DP}}{p} \left(\nabla_i f(x)^T h_i^\beta + \frac{\beta L_i}{2} \|h_i^\beta\|_{(i)}^2 \right) + \\ &+ \sum_{i \in \mathcal{J}} \frac{\tau}{p} \left(\nabla_i f(x)^T h_i^\beta + \frac{\beta L_i}{2} \|h_i^\beta\|_{(i)}^2 \right). \end{aligned} \quad (2.36)$$

Para determinar um valor de β para o qual a desigualdade anterior é válida, primeiramente usaremos o resultado [37, Teorema 12], que estabelece uma expressão para o valor de β para o caso em que os blocos de coordenadas são escolhidos com distribuição de probabilidade uniforme, e depois mostraremos como conectar esse resultado com nosso contexto não uniforme.

Proposição 2.2. Considere $x, h^\beta \in \mathbb{R}^n$ e $\mathcal{M} \subset \{1, \dots, m\}$ uma variável aleatória contendo τ blocos de coordenadas, em que a probabilidade de cada bloco de coordenadas pertencer ao conjunto \mathcal{M} é dada pela distribuição de probabilidade uniforme e $m > 1$. Se f é uma função parcialmente separável de grau ω , então,

$$\mathbb{E} \left[f \left(x + \sum_{k \in \mathcal{M}} U_k h_k^\beta \right) \right] \leq f(x) + \frac{\tau}{m} \sum_{i=1}^m \left(\nabla_i f(x)^T h_i^\beta + \frac{\beta L_i}{2} \|h_i^\beta\|_{(i)}^2 \right), \quad (2.37)$$

$$\text{com } \beta = 1 + \frac{(\omega - 1)(\tau - 1)}{m - 1}.$$

Prova: Primeiramente, para simplificar a notação, usaremos $\hat{h} = \sum_{k \in \mathcal{M}} U_k h_k^\beta$ e definimos as funções

$$\phi(h) = f(x + h) - f(x) - \nabla f(x)^T h,$$

$$\phi_J(h) = f_J(x + h) - f_J(x) - \nabla f_J(x)^T h, \forall J \in \mathcal{T},$$

em que as funções f_J , $J \in \mathcal{T}$ são aquelas definidas pela separabilidade parcial de f (cf. Definição 2.3). Desde que

$$\begin{aligned} \mathbb{E} [\phi(\hat{h})] &= \mathbb{E} [f(x + \hat{h}) - f(x) - \nabla f(x)^T \hat{h}] \\ &\stackrel{\text{Def. A.1, Lema A.1(iii)}}{=} \mathbb{E} [f(x + \hat{h})] - f(x) - \sum_{i=1}^m \nabla_i f(x)^T h_i^\beta \mathbb{P}(i \in \mathcal{M}) \\ &= \mathbb{E} [f(x + \hat{h})] - f(x) - \sum_{i=1}^m \frac{\tau}{m} \nabla_i f(x)^T h_i^\beta, \end{aligned}$$

para mostrar o desejado, basta garantir que

$$\mathbb{E} [\phi(\hat{h})] \leq \frac{\tau}{m} \sum_{i=1}^m \left(\frac{\beta L_i}{2} \|h_i^\beta\|_{(i)}^2 \right), \quad (2.38)$$

$$\text{com } \beta = 1 + \frac{(\omega - 1)(\tau - 1)}{m - 1}.$$

Usando os conjuntos $J \in \mathcal{T}$, descritos na Definição 2.3, denotaremos $\eta_J = |J \cap \mathcal{M}|$. Pela hipótese que f é parcialmente separável de grau ω , podemos assumir que $|J| = \omega$, desde que se $|J| < \omega$, isto é, alguma função f_J depende de um número menor do que ω blocos de coordenadas, poderíamos acrescentar alguns termos envolvendo blocos de coordenadas que não aparecem na função f_J sem alterar o valor da função e completando o número de ω blocos de coordenadas. Com essas notações, temos que

$$\begin{aligned} \mathbb{E} [\phi(\hat{h})] &= \mathbb{E} \left[\sum_{J \in \mathcal{T}} \phi_J(\hat{h}) \right] \\ &\stackrel{\text{Lema A.1(iii)}}{=} \sum_{J \in \mathcal{T}} \mathbb{E} [\phi_J(\hat{h})] \\ &= \sum_{t=0}^m \sum_{J \in \mathcal{T}} \mathbb{E} [\phi_J(\hat{h}) \mid (\eta_J = t)] \mathbb{P}(\eta_J = t) \\ &= \sum_{t=0}^m \mathbb{P}(\eta_J = t) \sum_{J \in \mathcal{T}} \mathbb{E} [\phi_J(\hat{h}) \mid (\eta_J = t)], \end{aligned} \quad (2.39)$$

em que a última igualdade segue do fato que $|J| = \omega$, para todo $J \in \mathcal{T}$ e da uniformidade do conjunto \mathcal{M} , garantida pela escolha uniforme dos índices, e pelo fato do termo $\mathbb{P}(\eta_J = t)$ independer de $J \in \mathcal{T}$.

Para todo $0 \leq t \leq m$ tal que $\mathbb{P}(\eta_J = t) > 0$, podemos escrever o vetor $\sum_{k \in J \cap \mathcal{M}} U_k h_k^\beta$, desde que o somatório possui η_J termos, como a combinação convexa

$$\sum_{k \in J \cap \mathcal{M}} U_k h_k^\beta = \frac{t - \eta_J}{t} \mathbf{0} + \frac{1}{t} \sum_{k \in J \cap \mathcal{M}} t U_k h_k^\beta.$$

Pela convexidade das funções $\phi_J(h)$, $\phi_J(\mathbf{0}) = 0$, para todo $J \in \mathcal{T}$ e usando a combinação convexa anterior, obtemos que

$$\phi_J \left(\sum_{k \in J \cap \mathcal{M}} U_k h_k^\beta \right) \leq \frac{t - \eta_J}{t} \phi_J(\mathbf{0}) + \frac{1}{t} \sum_{k \in J \cap \mathcal{M}} \phi_J(t U_k h_k^\beta) = \frac{1}{t} \sum_{k \in J \cap \mathcal{M}} \phi_J(t U_k h_k^\beta). \quad (2.40)$$

Usando (2.40), vemos que

$$\begin{aligned} \mathbb{E} \left[\phi_J(\hat{h}) \mid (\eta_J = t) \right] &\stackrel{\text{Lema A.1(v)}}{\leq} \mathbb{E} \left[\frac{1}{t} \sum_{k \in J \cap \mathcal{M}} \phi_J(t U_k h_k^\beta) \mid (\eta_J = t) \right] \\ &\stackrel{\text{Lema A.2(ii)}}{=} \frac{1}{t} \sum_{k \in J} \phi_J(t U_k h_k^\beta) (\mathbb{P}(k \in \mathcal{M}) \mid (\eta_J = t)) \\ &= \frac{1}{t} \sum_{k \in J} \frac{t}{|J|} \phi_J(t U_k h_k^\beta) \\ &= \frac{1}{\omega} \sum_{k \in J} \phi_J(t U_k h_k^\beta). \end{aligned} \quad (2.41)$$

Somando as desigualdades (2.41) para todos $J \in \mathcal{T}$, vemos que

$$\begin{aligned} \sum_{J \in \mathcal{T}} \mathbb{E} \left[\phi_J(\hat{h}) \mid (\eta_J = t) \right] &\leq \sum_{J \in \mathcal{T}} \left(\frac{1}{\omega} \sum_{k \in J} \phi_J(t U_k h_k^\beta) \right) \\ &= \frac{1}{\omega} \sum_{J \in \mathcal{T}} \left(\sum_{k=1}^m \phi_J(t U_k h_k^\beta) \right) \\ &= \frac{1}{\omega} \sum_{k=1}^m \left(\sum_{J \in \mathcal{T}} \phi_J(t U_k h_k^\beta) \right) \\ &= \frac{1}{\omega} \sum_{k=1}^m \phi(t U_k h_k^\beta) \\ &\stackrel{(2.3), (2.4)}{\leq} \frac{1}{\omega} \sum_{k=1}^m \frac{L_k}{2} \|t h_k^\beta\|_{(k)}^2 \\ &= \frac{t^2}{\omega} \sum_{k=1}^m \frac{L_k}{2} \|h_k^\beta\|_{(k)}^2, \end{aligned} \quad (2.42)$$

onde a primeira igualdade de (2.42) segue do fato que apesar das funções f_J dependerem apenas de ω blocos de coordenadas, podemos acrescentar todos os blocos de coordenadas às funções sem alterar seus valores de função.

Substituindo a expressão (2.42) em (2.39), temos que

$$\begin{aligned} \mathbb{E}[\phi(\hat{h})] &\leq \sum_{t=0}^m \mathbb{P}(\eta_J = t) \frac{t^2}{\omega} \left(\sum_{k=1}^m \frac{L_k}{2} \|h_k^\beta\|_{(k)}^2 \right) \\ &\stackrel{\text{Lema A.1(ii)}}{=} \frac{\mathbb{E}[|J \cap \mathcal{M}|^2]}{\omega} \left(\sum_{k=1}^m \frac{L_k}{2} \|h_k^\beta\|_{(k)}^2 \right). \end{aligned} \quad (2.43)$$

Calcularemos a esperança $\mathbb{E}[|J \cap \mathcal{M}|^2]$ em (2.43), da forma

$$\begin{aligned} \mathbb{E}[|J \cap \mathcal{M}|^2] &\stackrel{\text{Prop. A.1(i)}}{=} \sum_{i \in J} \sum_{j \in J} \mathbb{P}(i \in \mathcal{M}, j \in \mathcal{M}) \\ &= \sum_{i \in J} \sum_{j \neq i \in J} \mathbb{P}(i \in \mathcal{M}, j \in \mathcal{M}) + \sum_{i \in J} \mathbb{P}(i \in \mathcal{M}) \end{aligned} \quad (2.44)$$

Resolvendo as probabilidades em (2.44), obtemos que

$$\sum_{i \in J} \sum_{j \neq i \in J} \mathbb{P}(i \in \mathcal{M}, j \in \mathcal{M}) = \frac{\omega\tau(\omega-1)(\tau-1)}{m(m-1)}, \quad (2.45)$$

$$\sum_{i \in J} \mathbb{P}(i \in \mathcal{M}) = \frac{\omega\tau}{m}. \quad (2.46)$$

Substituindo as expressões (2.45) e (2.46) em (2.44), temos que

$$\begin{aligned} \mathbb{E}[|J \cap \mathcal{M}|^2] &\stackrel{(2.45), (2.46)}{=} \frac{\omega\tau}{m} + \frac{\omega\tau(\omega-1)(\tau-1)}{m(m-1)} \\ &= \frac{\omega\tau}{m} \left(1 + \frac{(\omega-1)(\tau-1)}{m-1} \right). \end{aligned} \quad (2.47)$$

Substituindo a expressão (2.47) em (2.43), vemos que

$$\begin{aligned} \mathbb{E}[\phi(\hat{h})] &\leq \frac{\frac{\omega\tau}{m} \left(1 + \frac{(\omega-1)(\tau-1)}{m-1} \right)}{\omega} \left(\sum_{k=1}^m \frac{L_k}{2} \|h_k^\beta\|_{(k)}^2 \right) \\ &= \frac{\tau}{m} \left(1 + \frac{(\omega-1)(\tau-1)}{m-1} \right) \left(\sum_{k=1}^m \frac{L_k}{2} \|h_k^\beta\|_{(k)}^2 \right). \end{aligned} \quad (2.48)$$

Como a expressão (2.48) é exatamente a expressão (2.38), concluímos assim a demonstração. □

Para utilizar as ideias expostas na Proposição 2.2 e garantir a existência de uma constante β satisfazendo (2.36), fixaremos o conjunto de índices do bloco de coordenadas \mathcal{I} , definido no Algoritmo 1, tomaremos $p = \delta_{DP}|\mathcal{I}| + |\mathcal{J}|$ e reordenaremos, sem perda de generalidade, os blocos de coordenadas definidos pela variável x como: os $|\mathcal{I}|$ primeiros blocos de coordenadas

serão aqueles relacionados ao conjunto \mathcal{I} , e o restante dos blocos serão descritos pelos índices do conjunto $|\mathcal{J}|$. Também, definiremos a função $\hat{f} : \mathbb{R}^r \rightarrow \mathbb{R}$ da forma

$$\hat{f}(z) = f(\hat{x}), \text{ com } \hat{x} = \sum_{i=1}^{|\mathcal{I}|} \left(U_i z_{(i)} + \sum_{j=1}^{\delta_{DP}-1} U_i z_{(\nu(i,j))} \right) + \sum_{i=1+|\mathcal{I}|}^m U_i z_{(i)}, \quad (2.49)$$

em que as matrizes U_i , $i \in \{1, \dots, m\}$ são as mesmas definidas na Seção 2.1 e a função $\nu : \{1, \dots, |\mathcal{I}|\} \times \{1, \dots, \delta_{DP}-1\} \rightarrow \{m+1, \dots, p\}$ é dada por $\nu(i, j) = m + j + (i-1)(\delta_{DP}-1)$ e $r = n + (\delta_{DP}-1) \sum_{i=1}^{|\mathcal{I}|} p_i$.

Observação 2.8. Semelhante à definição que fizemos no início da Seção 2.1, usaremos as seguintes matrizes

$$\hat{U}_i \in \mathbb{R}^{r \times p_i}, \quad i \in \{1, \dots, m\}$$

e

$$\hat{U}_j \in \mathbb{R}^{r \times p_i}, \quad j = m+1 + (i-1)(\delta_{DP}-1), \dots, m+i(\delta_{DP}-1),$$

com $i \in \{1, \dots, |\mathcal{I}|\}$, obedecendo à expressão a seguir

$$z = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_r \end{pmatrix} = \sum_{i=1}^p \hat{U}_i z_{(i)}, \text{ com } \begin{pmatrix} \hat{U}_1 & \hat{U}_2 & \dots & \hat{U}_p \end{pmatrix} P = I,$$

sendo $I \in \mathbb{R}^{r \times r}$ a matriz identidade de ordem r e $P \in \mathbb{R}^{r \times r}$ uma matriz de permutação, com $r = n + (\delta_{DP}-1) \sum_{i=1}^{|\mathcal{I}|} p_i$.

Observação 2.9. Essa maneira pouco natural de se organizar os índices dos blocos de coordenadas com respeito à variável z se deve ao fato de que com ela, podemos facilmente, dados os vetores $x, h \in \mathbb{R}^n$, encontrar um par de candidatos $z, \hat{h} \in \mathbb{R}^r$, tal que

$$\hat{f}(z) = f(x),$$

$$\hat{f}(z + \hat{U}_i \hat{h}_i) = f(x + U_i h_i), \quad \forall i \in \{1, \dots, m\}, \quad \hat{h}_i \in \mathbb{R}^{p_i},$$

e

$$\hat{f}(z + \hat{U}_j \hat{h}_j) = f(x + U_i h_i), \quad \forall j = m+1 + (i-1)(\delta_{DP}-1), \dots, m+i(\delta_{DP}-1),$$

com $i \in \{1, \dots, |\mathcal{I}|\}$.

A saber, podemos conseguir as expressões anteriores fazendo

$$z = (x, 0, \dots, 0)^T$$

e

$$\hat{h} = (h, \underbrace{h_1, \dots, h_1}_{(\delta_{DP}-1) \text{ vezes}}, \dots, \underbrace{h_{|\mathcal{I}|}, \dots, h_{|\mathcal{I}|}}_{(\delta_{DP}-1) \text{ vezes}})^T.$$

Observação 2.10. Pela forma como a função \hat{f} foi definida, claramente, se a função f é parcialmente separável com grau de separabilidade igual a ω , vemos que a função \hat{f} também é parcialmente separável, com grau de separabilidade não maior do que $\min\{|\mathcal{I}|, \omega\}(\delta_{DP} - 1) + \omega$.

Esse valor vem da análise que, no pior caso, a função $f(x)$ possui uma função f_J que depende de ω blocos de coordenadas, e além disso, possui o maior número possível de índices pertencentes ao conjunto \mathcal{I} . Portanto, o valor do grau de separabilidade parcial de \hat{f} será

$$\delta_{DP}|J \cap \mathcal{I}| + \omega - |J \cap \mathcal{I}| = |J \cap \mathcal{I}|(\delta_{DP} - 1) + \omega.$$

Substituindo, $\min\{|\mathcal{I}|, \omega\} = |J \cap \mathcal{I}|$, obtemos o desejado.

Alertamos que, estimando o grau de separabilidade da função \hat{f} pelo pior caso, podemos estar calculando um valor muito pior do que o real grau de separabilidade de \hat{f} em algumas situações. Porém, essa abordagem elimina o esforço computacional do nosso método de descenso coordenado de precisar calcular o novo valor para o grau de separabilidade de \hat{f} a cada momento em que o conjunto \mathcal{I} é modificado.

Pela maneira como a função \hat{f} foi definida, e usando a regra da cadeia, vemos que

$$\nabla_i \hat{f}(z) = U_i^T \nabla f(\hat{x}) = \nabla_i f(\hat{x}) \quad (2.50)$$

para todo $i \in \{1, \dots, m\}$ e

$$\nabla_j \hat{f}(z) = U_j^T \nabla f(\hat{x}) = \nabla_j f(\hat{x}) \quad (2.51)$$

para todo $j = m + 1 + (i - 1)(\delta_{DP} - 1), \dots, m + i(\delta_{DP} - 1)$, com $i \in \{1, \dots, |\mathcal{I}|\}$ e \hat{x} definido por (2.49).

Usando as expressões (2.50), (2.3) e (2.4), sabemos que

$$\|\nabla_i \hat{f}(z + \hat{U}_i \hat{h}_i) - \nabla_i \hat{f}(z)\|_{(i)}^* = \|\nabla_i f(\hat{x} + U_i \hat{h}_i) - \nabla_i f(\hat{x})\|_{(i)}^* \leq L_i \|\hat{h}_i\|_{(i)}, \hat{h}_i \in \mathbb{R}^{p_i}, \quad (2.52)$$

e

$$\hat{f}(z + \hat{U}_i \hat{h}_i) \leq \hat{f}(z) + \nabla_i f(\hat{x})^T \hat{h}_i + \frac{L_i}{2} \|\hat{h}_i\|_{(i)}^2, \hat{h}_i \in \mathbb{R}^{p_i}, \quad (2.53)$$

para todo $i \in \{1, \dots, m\}$, com \hat{x} definido por (2.49).

Pelas expressões (2.51), (2.3) e (2.4), vemos que

$$\|\nabla_j \hat{f}(z + \hat{U}_j \hat{h}_j) - \nabla_j \hat{f}(z)\|_{(j)}^* = \|\nabla_j f(\hat{x} + U_j \hat{h}_j) - \nabla_j f(\hat{x})\|_{(j)}^* \leq L_j \|\hat{h}_j\|_{(j)}, \hat{h}_j \in \mathbb{R}^{p_j}, \quad (2.54)$$

e

$$\hat{f}(z + \hat{U}_j \hat{h}_j) \leq \hat{f}(z) + \nabla_j f(\hat{x})^T \hat{h}_j + \frac{L_j}{2} \|\hat{h}_j\|_{(j)}^2, \quad (2.55)$$

para todo $j = m + 1 + (i - 1)(\delta_{DP} - 1), \dots, m + i(\delta_{DP} - 1)$, com $i \in \{1, \dots, |\mathcal{I}|\}$ e \hat{x} definido por (2.49).

Pela separabilidade parcial da função \hat{f} descrita na Observação 2.10, devido às expressões (2.52), (2.53), (2.54) e (2.55), podemos repetir os mesmos passos da Proposição 2.2, garantindo que, se $\mathcal{M} \subset \{1, \dots, p\}$ for uma variável aleatória, contendo τ blocos de coordenadas,

onde os blocos são escolhidos com distribuição de probabilidade uniforme, sabemos, para quaisquer $z, \hat{h} \in \mathbb{R}^r$, que

$$\begin{aligned} \mathbb{E} \left[\hat{f} \left(z + \sum_{k \in \mathcal{M}} \hat{U}_k \hat{h}_k \right) \right] &\leq \hat{f}(z) + \frac{\tau}{p} \sum_{i=1}^{|\mathcal{I}|} \left(\nabla_i f(\hat{x})^T \hat{h}_i + \frac{\beta L_i}{2} \|\hat{h}_i\|_{(i)}^2 + \sum_{j=1}^{\delta_{DP}-1} \nabla_i f(\hat{x})^T \hat{h}_{\nu(i,j)} + \right. \\ &\quad \left. + \frac{\beta L_i}{2} \|\hat{h}_{\nu(i,j)}\|_{(i)}^2 \right) + \frac{\tau}{p} \sum_{i=1+|\mathcal{I}|}^m \left(\nabla_i f(\hat{x})^T \hat{h}_i + \frac{\beta L_i}{2} \|\hat{h}_i\|_{(i)}^2 \right), \end{aligned} \quad (2.56)$$

com $\beta = 1 + \frac{[\min\{|\mathcal{I}|, \omega\}(\delta_{DP} - 1) + \omega - 1](\tau - 1)}{p - 1}$, e \hat{x} definido por (2.49).

Com a análise feita, podemos apresentar o resultado desejado sobre o valor da constante β que usaremos para garantir um decréscimo no valor esperado da função objetivo F , quando aplicado uma iteração do nosso método de descenso coordenado por blocos.

Corolário 2.1. *Considere $x, h^\beta \in \mathbb{R}^n$, dois conjuntos \mathcal{I}, \mathcal{J} tais que $|\mathcal{I}| + |\mathcal{J}| = m$ e $\mathcal{S} \subset \{1, \dots, m\}$ uma variável aleatória contendo τ blocos de coordenadas, em que a probabilidade de cada bloco de coordenadas pertencer ao conjunto \mathcal{S} é descrita pela distribuição de probabilidade do Algoritmo 1, com $p > 1$. Se f é uma função parcialmente separável de grau ω , então,*

$$\begin{aligned} \mathbb{E} \left[f \left(x + \sum_{k \in \mathcal{S}} U_k h_k^\beta \right) \right] &\leq f(x) + \sum_{i \in \mathcal{I}} \frac{\tau \delta_{DP}}{p} \left(\nabla_i f(x)^T h_i^\beta + \frac{\beta L_i}{2} \|h_i^\beta\|_{(i)}^2 \right) + \\ &\quad + \sum_{i \in \mathcal{J}} \frac{\tau}{p} \left(\nabla_i f(x)^T h_i^\beta + \frac{\beta L_i}{2} \|h_i^\beta\|_{(i)}^2 \right), \end{aligned} \quad (2.57)$$

com $\beta = 1 + \frac{[\min\{|\mathcal{I}|, \omega\}(\delta_{DP} - 1) + \omega - 1](\tau - 1)}{p - 1}$.

Prova: Sabemos que a expressão (2.56) é válida para todo $z, \hat{h} \in \mathbb{R}^r$ e para toda variável aleatória $\mathcal{M} \subset \{1, \dots, p\}$, contendo τ blocos de coordenadas, onde os blocos são escolhidos com distribuição de probabilidade uniforme. Por isso, vamos escolher $z, \hat{h} \in \mathbb{R}^r$ da forma: $z = (x, 0, \dots, 0)^T$ e

$$\hat{h} = (h^\beta, \underbrace{h_1^\beta, \dots, h_1^\beta}_{(\delta_{DP}-1)\text{vezes}}, \dots, \underbrace{h_{|\mathcal{I}|}^\beta, \dots, h_{|\mathcal{I}|}^\beta}_{(\delta_{DP}-1)\text{vezes}})^T.$$

Pela maneira como a função \hat{f} foi definida e pelas escolhas feitas para z e \hat{h} , sabemos

que

$$\begin{aligned}
\mathbb{E} \left[\hat{f} \left(z + \sum_{k \in \mathcal{M}} \hat{U}_k \hat{h}_k \right) \right] &\stackrel{\text{Def. A.1}}{=} \sum_{i=1}^p \hat{f}(z + \hat{U}_i \hat{h}_i) \mathbb{P}(i \in \mathcal{M}) \\
&= \sum_{i=1}^{|\mathcal{I}|} \left(\hat{f}(z + \hat{U}_i \hat{h}_i) \mathbb{P}(i \in \mathcal{M}) + \right. \\
&\quad \left. + \sum_{j=1}^{\delta_{DP}-1} \hat{f}(z + \hat{U}_i \hat{h}_{\nu(i,j)}) \mathbb{P}(\nu(i,j) \in \mathcal{M}) \right) + \\
&\quad + \sum_{i=1+|\mathcal{I}|}^m \hat{f}(z + \hat{U}_i \hat{h}_i) \mathbb{P}(i \in \mathcal{M}) \\
&= \sum_{i=1}^{|\mathcal{I}|} \left(f(x + U_i h_i^\beta) \mathbb{P}(i \in \mathcal{S}) + \sum_{j=1}^{\delta_{DP}-1} f(x + U_i h_i^\beta) \mathbb{P}(i \in \mathcal{S}) \right) + \\
&\quad + \sum_{i=1+|\mathcal{I}|}^m f(x + U_i h_i^\beta) \mathbb{P}(i \in \mathcal{S}) \\
&= \sum_{i=1}^{|\mathcal{I}|} \left(\frac{\tau}{p} f(x + U_i h_i^\beta) + \sum_{j=1}^{\delta_{DP}-1} \frac{\tau}{p} f(x + U_i h_i^\beta) \right) + \sum_{i=1+|\mathcal{I}|}^m \frac{\tau}{p} f(x + U_i h_i^\beta) \\
&= \sum_{i=1}^{|\mathcal{I}|} \frac{\tau \delta_{DP}}{p} f(x + U_i h_i^\beta) + \sum_{i=1+|\mathcal{I}|}^m \frac{\tau}{p} f(x + U_i h_i^\beta) \\
&= \sum_{i=1}^{|\mathcal{I}|} f(x + U_i h_i^\beta) \mathbb{P}(i \in \mathcal{S}) + \sum_{i=1+|\mathcal{I}|}^m f(x + U_i h_i^\beta) \mathbb{P}(i \in \mathcal{S}) \\
&\stackrel{\text{Def. A.1}}{=} \mathbb{E} \left[f \left(x + \sum_{k \in \mathcal{S}} U_k h_k^\beta \right) \right]. \tag{2.58}
\end{aligned}$$

Usando a expressão (2.58) e as escolhas específicas de z e \hat{h} na expressão (2.56), obtemos que

$$\begin{aligned}
\mathbb{E} \left[f \left(x + \sum_{k \in \mathcal{S}} U_k h_k^\beta \right) \right] &\leq f(x) + \sum_{i \in \mathcal{I}} \frac{\tau \delta_{DP}}{p} \left(\nabla_i f(x)^T h_i^\beta + \frac{\beta L_i}{2} \|h_i^\beta\|_{(i)}^2 \right) \\
&\quad + \sum_{i \in \mathcal{J}} \frac{\tau}{p} \left(\nabla_i f(x)^T h_i^\beta + \frac{\beta L_i}{2} \|h_i^\beta\|_{(i)}^2 \right), \tag{2.59}
\end{aligned}$$

com $\beta = 1 + \frac{[\min\{|\mathcal{I}|, \omega\}(\delta_{DP} - 1) + \omega - 1](\tau - 1)}{p - 1}$, e assim, concluímos a demonstração. \square

Depois desse resultado, apresentaremos a versão em paralelo do Algoritmo 1.

Algoritmo 2: *Active Parallel Coordinate Descent Method (Active PCDM)***Input:** $x^0 \in \mathcal{X}$, $\delta_{DP}, \delta_F \in \mathbb{N}$, $B \in \mathbb{S}_{++}^n$, $\tau \in \{1, \dots, m\}$, $\epsilon \in \mathbb{R}_+$, $l_{max} \in \mathbb{N}_+$ **Output:** x^k **begin** $\mathcal{I} \leftarrow \{1, \dots, m\};$ $\mathcal{J} \leftarrow \emptyset;$ $\gamma \leftarrow \lfloor \frac{\delta_F}{\tau} \rfloor;$ $\ell \leftarrow 1;$ **repeat** $p \leftarrow \delta_{DP}|\mathcal{I}| + |\mathcal{J}|;$ $\beta \leftarrow 1 + \frac{[\min\{|\mathcal{I}|, \omega\}(\delta_{DP} - 1) + \omega - 1](\tau - 1)}{p - 1};$ **for** $k = (\ell - 1)\gamma + 1$ **to** $\ell\gamma$ **do**Escolha um conjunto de τ blocos de coordenadas $S^k \subset \{1, \dots, m\}$ onde os elementos de S^k satisfazem a distribuição de probabilidade;

$$\mathbb{P}(i \in S^k) = \begin{cases} \frac{\delta_{DP}}{\delta_{DP}|\mathcal{I}| + |\mathcal{J}|}, & \text{se } i \in \mathcal{I}, \\ \frac{1}{\delta_{DP}|\mathcal{I}| + |\mathcal{J}|}, & \text{se } i \in \mathcal{J}. \end{cases}$$

for each $i \in S^k$ **in parallel do**Encontre $h_i^\beta(x^k)$ solução do problema (2.32); $x^{k+1} \leftarrow x^k + U_i h_i^\beta(x^k);$ $v_{(i)} \leftarrow h_i^\beta(x^k);$ **end****end**Escolha o conjunto $\mathcal{J} \subset \{1, \dots, m\}$ de alguma maneira; $\mathcal{I} \leftarrow \{1, \dots, m\} - \mathcal{J};$ $\ell \leftarrow \ell + 1;$ **until** $\|v\| \leq \epsilon$ **or** $\ell = l_{max};$ **end**

3 Resultados de Convergência

Neste capítulo apresentamos os resultados de convergência dos algoritmos *Active BCDM* (Algoritmo 1) e *Active PCDM* (Algoritmo 2).

3.1 *Active BCDM*

Iniciaremos a seção fazendo uma hipótese sobre a parte convexa da função objetivo do problema (2.1), $\psi(x)$ de modo que o resultado de convergência estará bem definido.

Hipótese 3.1. *Nessa seção assumiremos que a função $\psi(x)$ é contínua em \mathcal{X} .*

Para demonstração do resultado de convergência dos métodos de descenso coordenado seriais, precisaremos de alguns resultados auxiliares que serão demonstrados na sequência.

Lema 3.1. *Seja x um ponto não estacionário do problema (2.1) e $i \in \{1, \dots, m\}$ um bloco de coordenadas tal que $h_i(x) \neq 0$. Então, existem $r, \eta > 0$ tais que, para todo $y \in \mathcal{B}(x, r)$, temos que*

$$\|h_i(y)\|_{(i)} \geq \eta \|h_i(x)\|_{(i)}.$$

Prova: Vamos supor por absurdo que a tese não é válida. Então, existe uma sequência $y^k \rightarrow x$ factível ao conjunto \mathcal{X} e uma sequência de escalares $\eta_k \downarrow 0$ tais que

$$\|h_i(y^k)\|_{(i)} < \eta_k \|h_i(x)\|_{(i)}.$$

Portanto, $h_i(y^k) \rightarrow 0$.

Por outro lado, como a função que é minimizada na definição de $h_i(\cdot)$ é estritamente convexa, existe $\delta > 0$, tal que

$$\nabla_i f(x)^T h_i(x) + \frac{L_i}{2} \|h_i(x)\|_{(i)}^2 + \psi_i(x_{(i)} + h_i(x)) \leq \psi_i(x_{(i)}) - 2\delta.$$

Como $y^k \rightarrow x$, factível, e ψ é uma função contínua em \mathcal{X} , para k suficientemente grande, obtemos que

$$\nabla_i f(y^k)^T h_i(x) + \frac{L_i}{2} \|h_i(x)\|_{(i)}^2 + \psi_i(y_{(i)}^k + h_i(x)) < \psi_i(y_{(i)}^k) - \delta.$$

Usando o fato que $h_i(y^k) \rightarrow 0$ e novamente a continuidade de $\psi(x)$, temos, para k suficientemente grande, que

$$\psi_i(y_{(i)}^k) - \delta < \nabla_i f(y^k)^T h_i(y^k) + \frac{L_i}{2} \|h_i(y^k)\|_{(i)}^2 + \psi_i(y_{(i)}^k + h_i(y^k)).$$

As duas últimas desigualdades combinadas contradizem a otimalidade de $h_i(y^k)$, concluindo assim a demonstração. \square

Lema 3.2. *Seja $\mathcal{K} \subset \mathbb{N}$ um conjunto infinito de índices dos blocos de coordenadas atualizados nas iterações do método Active BCDM e $i \in \{1, \dots, m\}$ um bloco de coordenadas fixado. A probabilidade de que i pertencerá um número infinito de vezes ao conjunto \mathcal{K} é igual a 1.*

Prova: Para um índice k ser escolhido um número finito de vezes, ele não pode ser escolhido após um índice finito $s \in \mathcal{K}$.

Pela distribuição de probabilidade do método Active BCDM, podemos garantir a existência de uma constante $\varrho = 1/(\delta_{DPM}) > 0$ tal que a probabilidade do bloco i ser escolhido em uma iteração $k \in \mathcal{K}$ é não menor do que ϱ . Portanto, temos que a probabilidade do bloco i não ser escolhido em uma iteração $k \in \mathcal{K}$ é, no máximo, $(1 - \varrho)$. Logo, a probabilidade de que esse índice não é escolhido para $k > s$, $k \in \mathcal{K}$ é, no máximo, igual a

$$\prod_{k>s, k \in \mathcal{K}} (1 - \varrho) = 0.$$

Pela última expressão, vemos que a probabilidade de que o bloco de coordenadas i é escolhido um número finito de vezes em \mathcal{K} é igual a 0, e portanto, a probabilidade de que esse bloco de coordenadas seja escolhido um número infinito de vezes é 1. \square

Para concluir a subseção, mostraremos o resultado de convergência dos métodos de descenso coordenado.

Teorema 3.1. *Seja $\{x^k\}$ uma sequência gerada pelo método de descenso coordenado Active BCDM, x^* um dos seus pontos de acumulação e suponha que F é limitada inferiormente na caixa \mathcal{X} . A probabilidade de que x^* seja um ponto estacionário do problema (2.1) é 1.*

Prova: Seja $\mathcal{K} \subset \mathbb{N}$ um subconjunto de índices associado a uma subsequência de $\{x^k\}$ convergindo para x^* .

Se x^* é um ponto não estacionário do problema (2.1), podemos, sem perda de generalidade, assumir que para todo $k \in \mathcal{K}$, temos que $x^k \in \mathcal{B}(x^*, r)$, vizinhança de raio r dada pelo Lema 3.1. Além disso, em toda iteração em que o bloco i do Lema 3.1 for escolhido pelos métodos de descenso coordenado por blocos, vemos pelo Lema 2.1 que a função objetivo F decresce pelo menos o valor constante de $\frac{\eta^2 L_i}{2} \|h_i(x^*)\|_{(i)}$.

Como a função F é limitada inferiormente em \mathcal{X} e os métodos de descenso coordenado não permitem que o valor de F aumente de uma iteração para a outra, pelo fato que \mathcal{K} é um conjunto infinito, podemos usar o Lema 3.2, garantindo que a probabilidade de que um bloco $i \in \{1, \dots, m\}$ seja atualizado um número finito de vezes é igual a 0.

Portanto, a probabilidade de que x^* seja um ponto não estacionário do problema (2.1) é igual a 0 e, por consequência, a probabilidade de que x^* seja um ponto estacionário do problema (2.1) é igual a 1, concluindo assim a demonstração. \square

3.2 Active PCDM

Mostraremos nessa seção resultados de complexidade presentes em [37], garantindo a convergência do algoritmo *Active PCDM* (Algoritmo 2). Apresentaremos, inicialmente, três lemas técnicos [37, Lemas 14, 15 e 16], que servirão de suporte para a demonstração do resultado de complexidade principal. Nessa seção vamos supor que f é uma função convexa.

Lema 3.3. *Seja*

$$\mathcal{G}(x, h) = f(x) + \nabla f(x)^T h + \frac{\bar{\beta}}{2} \|h\|_B^2 + \psi(x + h), \quad (3.1)$$

em que $\bar{\beta} > 0$, $\|z\|_B^2 = \sum_{i=1}^m L_i z_{(i)}^T B_i z_{(i)}$, com matrizes B_i , $i \in \{1, \dots, m\}$ definidas na Seção 2.1. Então, para todo $x \in \mathcal{X}$ vale

$$\min_h \mathcal{G}(x, h) \leq \min_{y \in \mathbb{R}^n} \left\{ f(y) + \frac{\bar{\beta}}{2} \|y - x\|_B^2 + \psi(y) \right\}. \quad (3.2)$$

Prova: Pela convexidade de f , sabemos que

$$f(y) \geq f(x) + \nabla f(x)^T (y - x), \forall x, y \in \mathbb{R}^n. \quad (3.3)$$

Tomando $h = y - x$, temos que

$$\begin{aligned} \min_{h \in \mathbb{R}^n} \mathcal{G}(x, h) &= \min_{y \in \mathbb{R}^n} \mathcal{G}(x, y - x) \\ &= \min_{y \in \mathbb{R}^n} \left\{ f(x) + \nabla f(x)^T (y - x) + \frac{\bar{\beta}}{2} \|y - x\|_B^2 + \psi(y) \right\} \\ &\leq \min_{y \in \mathbb{R}^n} \left\{ f(y) + \frac{\bar{\beta}}{2} \|y - x\|_B^2 + \psi(y) \right\}. \end{aligned}$$

□

Lema 3.4. *Suponhamos x^* um minimizador do problema (2.1), $x \in \mathcal{X}$, $F^* = F(x^*)$ e $R = \|x - x^*\|_B$. Então*

$$\min_h \mathcal{G}(x, h) - F^* \leq \begin{cases} \left(1 - \frac{F(x) - F^*}{2\bar{\beta}R^2}\right) (F(x) - F^*), & \text{se } F(x) - F^* \leq \bar{\beta}R^2; \\ \frac{1}{2}\bar{\beta}R^2 < \frac{1}{2}(F(x) - F^*), & \text{caso contrário.} \end{cases} \quad (3.4)$$

Prova: Pelo Lema 3.3 e pela convexidade de F , vemos que

$$\begin{aligned} \min_h \mathcal{G}(x, h) &\leq \min_{y \in \mathbb{R}^n} \left\{ f(y) + \frac{\bar{\beta}}{2} \|y - x\|_B^2 + \psi(y) \right\} \\ &\leq \min_{\lambda \in [0,1]} \left\{ f(\lambda x^* + (1 - \lambda)x) + \frac{\bar{\beta}}{2} \|\lambda x^* + (1 - \lambda)x - x\|_B^2 + \psi(\lambda x^* + (1 - \lambda)x) \right\} \\ &\leq \min_{\lambda \in [0,1]} \left\{ \lambda f(x^*) + (1 - \lambda)f(x) + \frac{\bar{\beta}\lambda^2}{2} R^2 + \lambda\psi(x^*) + (1 - \lambda)\psi(x) \right\} \\ &= \min_{\lambda \in [0,1]} \left\{ F(x) + \lambda(F^* - F(x)) + \frac{\bar{\beta}\lambda^2}{2} R^2 \right\}. \end{aligned} \quad (3.5)$$

Sabemos que o minimizador para a expressão (3.5) com respeito a λ irrestrito é $\lambda^* = \frac{F(x) - F^*}{\bar{\beta}R^2}$. Pela minimalidade de F^* , temos que o minimizador para a expressão (3.5) é dado por

$$\lambda^* = \min \left\{ 1, \frac{F(x) - F^*}{\bar{\beta}R^2} \right\}.$$

Portanto, temos dois casos a considerar:

Caso 1: $F(x) - F^* \leq \bar{\beta}R^2$.

Nesse caso, temos que $\lambda^* = \frac{F(x) - F^*}{\bar{\beta}R^2}$. Substituindo esse valor de λ em (3.5), obtemos que

$$\begin{aligned} \min_h \mathcal{G}(x, h) - F^* &\leq F(x) + \frac{F(x) - F^*}{\bar{\beta}R^2} (F^* - F(x)) + \frac{\bar{\beta}R^2 \left[\frac{F(x) - F^*}{\bar{\beta}R^2} \right]^2}{2} - F^* \\ &= (F(x) - F^*) - \frac{(F(x) - F^*)^2}{\bar{\beta}R^2} + \frac{\bar{\beta}R^2 (F(x) - F^*)^2}{2\bar{\beta}^2 R^4} \\ &= (F(x) - F^*) - \frac{(F(x) - F^*)^2}{2\bar{\beta}R^2} \\ &= \left(1 - \frac{(F(x) - F^*)}{2\bar{\beta}R^2} \right) (F(x) - F^*). \end{aligned}$$

Caso 2: $F(x) - F^* > \bar{\beta}R^2$.

Nesse caso, temos que $\lambda^* = 1$. Substituindo em (3.5), vemos que

$$\begin{aligned} \min_h \mathcal{G}(x, h) - F^* &\leq F(x) + (F^* - F(x)) + \frac{\bar{\beta}R^2}{2} - F^* \\ &= \frac{\bar{\beta}R^2}{2} \\ &< (F(x) - F^*). \end{aligned}$$

Com as duas expressões anteriores, concluímos a demonstração. \square

Lema 3.5. Escolha uma precisão $0 < \epsilon < \xi_0$, um nível de confiança $0 < \delta < 1$, e assuma que a sequência $\{\xi_k\}_{k \geq 0}$ é não-crescente e satisfaz uma das propriedades:

- (i) $\mathbb{E}[\xi_{k+1} | \xi_k] \leq \left(1 - \frac{\xi_k}{c} \right) \xi_k$, para todo k , onde $c > \epsilon$ é uma constante;
- (ii) $\mathbb{E}[\xi_{k+1} | \xi_k] \leq \left(1 - \frac{1}{c} \right) \xi_k$, para todo k , tal que $\xi_k > \epsilon$ e $c > 1$ é uma constante.

Se a propriedade (i) ocorre e escolhemos

$$K > 2 + \frac{c}{\epsilon} \left(1 - \frac{\epsilon}{\xi_0} + \log \left(\frac{1}{\delta} \right) \right), \quad (3.6)$$

ou se a propriedade (ii) ocorre e escolhemos

$$K > c \log \left(\frac{\xi_0}{\epsilon \delta} \right), \quad (3.7)$$

então

$$\mathbb{P}(\xi_K \leq \epsilon) \geq 1 - \delta. \quad (3.8)$$

Prova: Note que a sequência $\{\xi_k^\epsilon\}_{k \geq 0}$ definida por

$$\begin{cases} \xi_k, & \text{se } \xi_k \geq \epsilon; \\ 0, & \text{se caso contrário,} \end{cases}$$

satisfaz

$$\xi_k^\epsilon \geq \epsilon \Leftrightarrow \xi_k \geq \epsilon, \quad \forall k \geq 0. \quad (3.9)$$

Pela desigualdade de Markov, temos que

$$\mathbb{P}(\xi_k > \epsilon) = \mathbb{P}(\xi_k^\epsilon > \epsilon) \leq \frac{\mathbb{E}[\xi_k^\epsilon]}{\epsilon},$$

e, portanto, para mostrar (3.8) é suficiente garantir que

$$\theta_K \leq \epsilon \delta, \quad (3.10)$$

em que $\theta_k = \mathbb{E}[\xi_k^\epsilon]$. Se a propriedade (i) ocorre, então, pela definição de ξ_k^ϵ , sabemos que

$$\mathbb{E}[\xi_{k+1}^\epsilon | \xi_k^\epsilon] \leq \xi_k^\epsilon - \frac{(\xi_k^\epsilon)^2}{c}, \quad \forall k. \quad (3.11)$$

e

$$\mathbb{E}[\xi_{k+1}^\epsilon | \xi_k^\epsilon] \leq \left(1 - \frac{\epsilon}{c}\right) \xi_k^\epsilon, \quad \forall k. \quad (3.12)$$

Dada uma função convexa $\kappa(t)$ com $\kappa : \mathbb{R} \rightarrow \mathbb{R}$, podemos mostrar a Desigualdade de Jensen, usando $y = \xi_k^\epsilon$ e $x = \theta_k$ na relação (3.3), da forma

$$\begin{aligned} & \kappa(\xi_k^\epsilon) \geq \kappa(\theta_k) + \nabla \kappa(\theta_k)^T (\xi_k^\epsilon - \theta_k) \\ \Leftrightarrow & \mathbb{E}[\kappa(\xi_k^\epsilon)] \stackrel{\text{Lema A.1}(v)}{\geq} \mathbb{E}[\kappa(\theta_k) + \nabla \kappa(\theta_k)^T (\xi_k^\epsilon - \theta_k)] \\ \Leftrightarrow & \mathbb{E}[\kappa(\xi_k^\epsilon)] \stackrel{\text{Lema A.1}(iii)}{\geq} \mathbb{E}[\kappa(\theta_k)] + \nabla \kappa(\theta_k)^T \mathbb{E}[(\xi_k^\epsilon - \theta_k)] \\ \Leftrightarrow & \mathbb{E}[\kappa(\xi_k^\epsilon)] \stackrel{\text{Lema A.1}(iii)}{\geq} \kappa(\theta_k) + \nabla \kappa(\theta_k)^T (\mathbb{E}[\xi_k^\epsilon] - \mathbb{E}[\theta_k]) \\ \Leftrightarrow & \mathbb{E}[\kappa(\xi_k^\epsilon)] \geq \kappa(\theta_k) + \nabla \kappa(\theta_k)^T (\theta_k - \theta_k) \\ \Leftrightarrow & \mathbb{E}[\kappa(\xi_k^\epsilon)] \geq \kappa(\theta_k). \end{aligned}$$

Tomando a esperança nos dois membros das relações (3.11), (3.12) e usando a Desigualdade de Jensen anterior em (3.11), para $\kappa(t) = t^2$, obtemos que

$$\theta_{k+1} \leq \theta_k - \frac{\mathbb{E}[(\xi_k^\epsilon)^2]}{c} \leq \theta_k - \frac{\theta_k^2}{c}, \quad (3.13)$$

e

$$\theta_{k+1} \leq \left(1 - \frac{\epsilon}{c}\right) \theta_k. \quad (3.14)$$

Notemos que a expressão (3.13) é melhor do que a expressão (3.14) precisamente quando $\theta_k > \epsilon$. Pela monotonicidade da sequência $\{\xi_k^\epsilon\}_{k \geq 0}$, herdada de $\{\xi_k\}_{k \geq 0}$, temos que $\theta_{k+1} \leq \theta_k$, e, portanto, vemos que

$$\begin{aligned} \frac{1}{\theta_{k+1}} - \frac{1}{\theta_k} &= \frac{\theta_k - \theta_{k+1}}{\theta_{k+1}\theta_k} \geq \frac{\theta_k - \theta_{k+1}}{\theta_k^2} \stackrel{(3.13)}{\geq} \frac{1}{c} \\ \Leftrightarrow \frac{1}{\theta_{k+1}} &\geq \frac{1}{\theta_k} + \frac{1}{c}. \end{aligned} \quad (3.15)$$

Aplicando sucessivamente a expressão (3.15), temos que

$$\frac{1}{\theta_k} \geq \frac{1}{\theta_0} + \frac{k}{c} = \frac{1}{\xi_0} + \frac{k}{c}. \quad (3.16)$$

Portanto, se temos $k_1 \geq \frac{c}{\epsilon} - \frac{c}{\xi_0}$, obtemos por (3.16) que $\theta_{k_1} \leq \epsilon$. Finalmente, tomando $k_2 \leq \frac{c}{\epsilon} \log\left(\frac{1}{\delta}\right)$, vemos que

$$\theta_K \stackrel{(3.6)}{\leq} \theta_{k_1+k_2} \stackrel{(3.14)}{\leq} \left(1 - \frac{\epsilon}{c}\right)^{k_2} \theta_{k_1} \leq \left(\left(1 - \frac{\epsilon}{c}\right)^{\frac{1}{\epsilon}}\right)^{c \log\left(\frac{1}{\delta}\right)} \epsilon \leq \left(e^{-\frac{1}{c}}\right)^{c \log\left(\frac{1}{\delta}\right)} \epsilon \leq \epsilon \delta,$$

estabelecendo assim (3.10).

Se a propriedade (ii) ocorre, então $\mathbb{E}[\xi_{k+1}|\xi_k] \leq \left(1 - \frac{1}{c}\right) \xi_k$, para todo k , e portanto

$$\theta_K \leq \left(1 - \frac{1}{c}\right)^K \theta_0 = \left(1 - \frac{1}{c}\right)^K \xi_0 \stackrel{(3.7)}{\leq} \left(\left(1 - \frac{1}{c}\right)^c\right)^{\log\left(\frac{\xi_0}{\epsilon\delta}\right)} \xi_0 \leq (e^{-1})^{\log\left(\frac{\xi_0}{\epsilon\delta}\right)} \xi_0 = \epsilon\delta,$$

novamente, estabelecendo assim (3.10) e concluindo a demonstração. \square

Agora vamos apresentar o teorema principal de complexidade do algoritmo *Active PCDM*, baseado em [37, Teorema 17].

Teorema 3.2. *Dados $x^0 \neq x^* \in \mathcal{X}$ satisfazendo*

$$R_B(x^0, x^*) \stackrel{\text{def}}{=} \max_x \{\|x - x^*\|_B : F(x) \leq F(x^0)\} < +\infty, \quad (3.17)$$

em que x^ é uma solução do problema (2.1). Além disso, escolha um nível de confiança $0 < \delta < 1$, um nível de precisão $\epsilon > 0$ e um contador de iterações K de uma das duas maneiras:*

$$(i) \quad \epsilon < F(x^0) - F^* \text{ e}$$

$$K > 2 + \frac{2\left(\frac{\beta}{\alpha}\right) \max\left\{R_B^2(x^0, x^*), \frac{F(x^0) - F^*}{\beta}\right\}}{\epsilon} \left(1 - \frac{\epsilon}{F(x^0) - F^*} + \log\left(\frac{1}{\delta}\right)\right). \quad (3.18)$$

$$(ii) \quad \epsilon < \min\left\{2\left(\frac{\beta}{\alpha}\right) R_B^2(x^0, x^*), F(x^0) - F^*\right\} \text{ e}$$

$$K > 2 + \frac{2\left(\frac{\beta}{\alpha}\right) R_B^2(x^0, x^*)}{\epsilon} \log\left(\frac{F(x^0) - F^*}{\epsilon\delta}\right), \quad (3.19)$$

em que $\alpha = \frac{\tau}{\delta_{DPM}}$ e $\beta = 1 + \frac{[\min\{|\mathcal{I}|, \omega\}(\delta_{DP} - 1) + \omega - 1](\tau - 1)}{p - 1}$, para *Active PCDM*.

Se $\{x^k\}_{k \geq 0}$ são iterandos gerados pelo *Active PCDM*, assumindo adicionalmente que a sequência $\{F(x^k)\}_{k \geq 0}$ é não-crescente, então $\mathbb{P}(F(x^K) - F^* \leq \epsilon) \geq 1 - \delta$.

Prova: Por hipótese, temos que $F(x^k) \leq F(x^0)$, para todo k . Em vista de (3.17), sabemos que $\|x^k - x^*\|_B \leq R_B(x^0, x^*)$. Tomando $\xi_k = F(x^k) - F^*$, temos para o algoritmo *Active PCDM*, pelo Lema 2.1, que $\min_{h_i} \left\{ \nabla_i f(x^k)^T h_i + \frac{\beta L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)}^k + h_i) - \psi_i(x_{(i)}^k) \right\} \leq 0$. Além disso, usando a desigualdade $p = \delta_{DP}|\mathcal{I}| + |\mathcal{J}| \leq \delta_{DPM}$, vemos que

$$\begin{aligned}
\mathbb{E}[\xi_{k+1} | \xi_k] &\stackrel{\text{Cor. 2.1}}{\leq} \xi_k + \\
&\quad + \min_h \left\{ \sum_{i \in \mathcal{I}} \frac{\delta_{DP}\tau}{p} \left(\nabla_i f(x^k)^T h_i + \frac{\beta L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)}^k + h_i) - \psi_i(x_{(i)}^k) \right) + \right. \\
&\quad \left. + \sum_{i \in \mathcal{J}} \frac{\tau}{p} \left(\nabla_i f(x^k)^T h_i + \frac{\beta L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)}^k + h_i) - \psi_i(x_{(i)}^k) \right) \right\} \\
&= \xi_k + \\
&\quad + \sum_{i \in \mathcal{I}} \frac{\delta_{DP}\tau}{p} \left\{ \min_{h_i} \left(\nabla_i f(x^k)^T h_i + \frac{\beta L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)}^k + h_i) - \psi_i(x_{(i)}^k) \right) \right\} + \\
&\quad + \sum_{i \in \mathcal{J}} \frac{\tau}{p} \left\{ \min_{h_i} \left(\nabla_i f(x^k)^T h_i + \frac{\beta L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)}^k + h_i) - \psi_i(x_{(i)}^k) \right) \right\} \\
&\leq \xi_k + \\
&\quad + \sum_{i \in \mathcal{I}} \frac{\tau}{p} \left\{ \min_{h_i} \left(\nabla_i f(x^k)^T h_i + \frac{\beta L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)}^k + h_i) - \psi_i(x_{(i)}^k) \right) \right\} + \\
&\quad + \sum_{i \in \mathcal{J}} \frac{\tau}{p} \left\{ \min_{h_i} \left(\nabla_i f(x^k)^T h_i + \frac{\beta L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)}^k + h_i) - \psi_i(x_{(i)}^k) \right) \right\} \\
&\leq \xi_k + \\
&\quad + \sum_{i \in \mathcal{I}} \frac{\tau}{\delta_{DPM}} \left\{ \min_{h_i} \left(\nabla_i f(x^k)^T h_i + \frac{\beta L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)}^k + h_i) - \psi_i(x_{(i)}^k) \right) \right\} + \\
&\quad + \sum_{i \in \mathcal{J}} \frac{\tau}{\delta_{DPM}} \left\{ \min_{h_i} \left(\nabla_i f(x^k)^T h_i + \frac{\beta L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)}^k + h_i) - \psi_i(x_{(i)}^k) \right) \right\} \\
&= \xi_k + \frac{\tau}{\delta_{DPM}} \left(\min_h \left\{ \nabla f(x^k)^T h + \frac{\beta}{2} \|h\|_B^2 + \psi(x^k + h) - \psi(x^k) \right\} \right) \\
&= (1 - \frac{\tau}{\delta_{DPM}}) \xi_k + \frac{\tau}{\delta_{DPM}} \left(\min_h \left\{ f(x^k) + \nabla f(x^k)^T h + \frac{\beta}{2} \|h\|_B^2 + \right. \right. \\
&\quad \left. \left. + \psi(x^k + h) \right\} - F^* \right) \\
&= (1 - \frac{\tau}{\delta_{DPM}}) \xi_k + \frac{\tau}{\delta_{DPM}} \left(\min_h \left\{ \mathcal{G}(x^k, h) \right\} - F^* \right). \tag{3.20}
\end{aligned}$$

Por (3.20), conseguimos mostrar que

$$\mathbb{E}[\xi_{k+1} | \xi_k] \leq (1 - \alpha) \xi_k + \alpha \left(\min_h \left\{ \mathcal{G}(x^k, h) \right\} - F^* \right), \text{ com } 0 < \alpha < 1. \tag{3.21}$$

Portanto, vemos que

$$\begin{aligned}
\mathbb{E}[\xi_{k+1}|\xi_k] &\stackrel{(3.21)}{\leq} (1-\alpha)\xi_k + \alpha \left(\min_h \left\{ \mathcal{G}(x^k, h) \right\} - F^* \right) \\
&\stackrel{(3.4)}{\leq} (1-\alpha)\xi_k + \alpha \left(\max \left\{ 1 - \frac{\xi_k}{2\beta\|x^k - x^*\|_B^2}, \frac{1}{2} \right\} \xi_k \right) \\
&= \max \left\{ 1 - \frac{\alpha\xi_k}{2\beta\|x^k - x^*\|_B^2}, 1 - \frac{\alpha}{2} \right\} \xi_k \\
&\leq \max \left\{ 1 - \frac{\alpha\xi_k}{2\beta R_B(x^0, x^*)^2}, 1 - \frac{\alpha}{2} \right\} \xi_k. \tag{3.22}
\end{aligned}$$

Considere o caso (i), e seja $c_1 = 2 \left(\frac{\beta}{\alpha} \right) \max \left\{ R_B^2(x^0, x^*), \frac{\xi_0}{\beta} \right\}$. Por (3.22), temos que

$$\mathbb{E}[\xi_{k+1}|\xi_k] \leq \left(1 - \frac{\xi_k}{c_1} \right) \xi_k, \quad \forall k.$$

Como $0 < \xi_0 < c_1$, para obter o desejado basta aplicar o Lema 3.5 (i). Considere agora o caso (ii), e tome $c_2 = 2 \left(\frac{\beta}{\alpha} \right) \frac{R_B^2(x^0, x^*)}{\epsilon}$. Observe que sempre que $\xi_k \geq \epsilon$, de (3.22), obtemos que

$$\mathbb{E}[\xi_{k+1}|\xi_k] \leq \left(1 - \frac{1}{c_2} \right) \xi_k.$$

Por hipótese, $c_2 > 1$ e, portanto, para se obter o desejado basta aplicar o Lema 3.5 (ii), concluindo assim a demonstração do teorema. \square

Observação 3.1. Alertamos aos leitores que, apesar de não apresentado nesse texto, é possível encontrar em [37] um resultado de complexidade do algoritmo *Active PCDM* que não precisa da hipótese de monotonicidade da sequência $\{F(x^k)\}_{k \geq 0}$, porém deve-se acrescentar a hipótese de que f é fortemente convexa.

Podemos melhorar um pouco a complexidade do resultado de convergência do algoritmo *Active PCDM*, Teorema 3.2, levando em consideração que as coordenadas escolhidas para o \mathcal{J} não desempenham papel no decréscimo do valor de função do problema, dentro de cada ciclo de iterações de tamanho δ_F durante o algoritmo.

Corolário 3.1. Dados $x^0 \neq x^* \in \mathcal{X}$ satisfazendo

$$R_B(x^0, x^*) \stackrel{\text{def}}{=} \max_x \{ \|x - x^*\|_B : F(x) \leq F(x^0) \} < +\infty, \tag{3.23}$$

em que x^* é uma solução do problema (2.1). Além disso, escolha um nível de confiança $0 < \delta < 1$, um nível de precisão $\epsilon > 0$, assumamos que durante todo ciclo de iterações de tamanho δ_F o conjunto de índices \mathcal{J} não altera o valor de função. Tome um contador de iterações K de uma das duas maneiras:

$$\begin{aligned}
&(i) \quad \epsilon < F(x^0) - F^* \text{ e} \\
&K > 2 + \frac{2 \left(\frac{\beta}{\alpha} \right) \max \left\{ R_B^2(x^0, x^*), \frac{F(x^0) - F^*}{\beta} \right\}}{\epsilon} \left(1 - \frac{\epsilon}{F(x^0) - F^*} + \log \left(\frac{1}{\delta} \right) \right). \tag{3.24}
\end{aligned}$$

$$(ii) \quad \epsilon < \min \left\{ 2 \left(\frac{\beta}{\bar{\alpha}} \right) R_B^2(x^0, x^*), F(x^0) - F^* \right\} \quad e$$

$$K > 2 + \frac{2 \left(\frac{\beta}{\bar{\alpha}} \right) R_B^2(x^0, x^*)}{\epsilon} \log \left(\frac{F(x^0) - F^*}{\epsilon \delta} \right), \quad (3.25)$$

em que $\bar{\alpha} = \frac{\tau}{m}$ e $\beta = 1 + \frac{[\min\{|\mathcal{I}|, \omega\}(\delta_{DP} - 1) + \omega - 1](\tau - 1)}{p - 1}$, para Active PCDM.

Se $\{x^k\}_{k \geq 0}$ são iterandos gerados pelo Active PCDM, assumindo adicionalmente que a sequência $\{F(x^k)\}_{k \geq 0}$ é não-crescente, então $\mathbb{P}(F(x^K) - F^* \leq \epsilon) \geq 1 - \delta$.

Prova: Pelo fato que os índices do conjunto \mathcal{J} não melhoram a função objetivo sabemos que

$$\min_{h_i} \left\{ \nabla_i f(x^k)^T h_i + \frac{\beta L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)}^k + h_i) - \psi_i(x_{(i)}^k) \right\} = 0, \quad \forall i \in \mathcal{J}. \quad (3.26)$$

Seguindo os mesmos argumentos da demonstração do Teorema 3.2 e usando a expressão (3.26), vemos que

$$\begin{aligned} \mathbb{E}[\xi_{k+1} | \xi_k] &\stackrel{\text{Cor. 2.1}}{\leq} \xi_k + \\ &+ \min_h \left\{ \sum_{i \in \mathcal{I}} \frac{\delta_{DP}\tau}{p} \left(\nabla_i f(x^k)^T h_i + \frac{\beta L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)}^k + h_i) - \psi_i(x_{(i)}^k) \right) + \right. \\ &\quad \left. + \sum_{i \in \mathcal{J}} \frac{\tau}{p} \left(\nabla_i f(x^k)^T h_i + \frac{\beta L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)}^k + h_i) - \psi_i(x_{(i)}^k) \right) \right\} \\ &= \xi_k + \\ &+ \sum_{i \in \mathcal{I}} \frac{\delta_{DP}\tau}{p} \left\{ \min_{h_i} \left(\nabla_i f(x^k)^T h_i + \frac{\beta L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)}^k + h_i) - \psi_i(x_{(i)}^k) \right) \right\} + \\ &+ \sum_{i \in \mathcal{J}} \frac{\delta_{DP}\tau}{p} \left\{ \min_{h_i} \left(\nabla_i f(x^k)^T h_i + \frac{\beta L_i}{2} \|h_i\|_{(i)}^2 + \psi_i(x_{(i)}^k + h_i) - \psi_i(x_{(i)}^k) \right) \right\} \\ &= \xi_k + \frac{\delta_{DP}\tau}{p} \left(\min_h \left\{ \nabla f(x^k)^T h + \frac{\beta}{2} \|h\|_B^2 + \psi(x^k + h) - \psi(x^k) \right\} \right) \\ &= \xi_k + \frac{\delta_{DP}\tau}{\delta_{DP}m} \left(\min_h \left\{ \nabla f(x^k)^T h + \frac{\beta}{2} \|h\|_B^2 + \psi(x^k + h) - \psi(x^k) \right\} \right) \\ &= \left(1 - \frac{\tau}{m} \right) \xi_k + \frac{\tau}{m} \left(\min_h \left\{ f(x^k) + \nabla f(x^k)^T h + \frac{\beta}{2} \|h\|_B^2 + \right. \right. \\ &\quad \left. \left. + \psi(x^k + h) \right\} - F^* \right) \\ &= \left(1 - \frac{\tau}{m} \right) \xi_k + \frac{\tau}{m} \left(\min_h \left\{ \mathcal{G}(x^k, h) \right\} - F^* \right). \end{aligned} \quad (3.27)$$

Pela equação (3.27), podemos repetir os argumentos utilizados na demonstração do Teorema 3.2 para $\bar{\alpha} = \frac{\tau}{m}$, concluindo assim a demonstração. \square

Observação 3.2. Ressaltamos que a hipótese de que o conjunto de índices \mathcal{J} não altera o valor de função é uma hipótese razoável de ser incorporada, desde que esse conjunto é construído com informação sobre as restrições ativas no final de cada ciclo de iterações. Ela é satisfeita, por

exemplo, se o conjunto \mathcal{J} estiver contido no conjunto das restrições ativas durante todos os ciclos de iterações do método.

Tendo em vista os resultados de complexidade anteriores, vamos apresentar expressões envolvendo a aceleração esperada com respeito ao número de iterações dos métodos de descenso coordenado uniforme, PCDM1 [37], e para nosso algoritmo, **Active PCDM**.

Pelos Teorema 3.2 e Corolário 3.1, sabemos que o número de iterações necessário para os algoritmos PCDM1 e **Active PCDM** atingirem uma ϵ -precisão é proporcional a β_τ/α_τ iterações, onde τ representa o número de *threads* utilizadas pelos algoritmos. Logo, podemos definir a aceleração esperada usando a quantidade de τ *threads*, denotada por \mathcal{AE} , como sendo o quociente entre o número de iterações do método serial e o número de iterações do método em paralelo, para ambos atingirem uma ϵ -precisão, isto é,

$$\mathcal{AE} = \frac{(\beta_1/\alpha_1)}{(\beta_\tau/\alpha_\tau)} = \frac{m}{\beta_\tau m/\tau} = \frac{\tau}{\beta_\tau}. \quad (3.28)$$

Por meio da expressão (3.28), lembrando que o método PCDM1 pode ser derivado do algoritmo *Active PCDM* para $\delta_{DP} = 1$, descrevemos a aceleração esperada com respeito ao número de iterações para PCDM1 com τ *threads* como

$$\mathcal{AE}_{PCDM1} = \frac{\tau}{1 + \frac{(\omega - 1)(\tau - 1)}{m - 1}}. \quad (3.29)$$

Novamente usando a expressão (3.28), temos que a aceleração esperada pelo algoritmo *Active PCDM*, com respeito ao número de iterações, usando τ *threads* é dada pela expressão

$$\mathcal{AE}_{APCDM} = \frac{\tau}{1 + \frac{[\min\{|\mathcal{I}|, \omega\}(\delta_{DP} - 1) + \omega - 1](\tau - 1)}{p - 1}}. \quad (3.30)$$

4 Identificação das restrições ativas

Este capítulo contém as propostas de estratégias de restrições ativas especialmente adequadas para métodos de descenso coordenado por blocos, aplicados a problemas com restrições simples, do tipo caixa.

4.1 Identificação das restrições ativas para pontos não degenerados

Começaremos definindo o conjunto de índices das restrições associado a um ponto x com relação à caixa, reescrita como

$$\mathcal{X} = \{x \mid -x_i + l_i \leq 0, \ i = 1, \dots, n, \text{ e } x_i - u_i \leq 0, \ n + i = n + 1, \dots, 2n\}.$$

Definição 4.1. *O conjunto dos índices das restrições ativas do ponto x , denotado por $\mathcal{A}(x)$, com relação ao conjunto \mathcal{X} é definido como*

$$\mathcal{A}(x) = \{i \mid x_i = l_i\} \cup \{n + i \mid x_i = u_i\},$$

e é tal que $\mathcal{A}(x) \subset \{1, \dots, 2n\}$.

Uma propriedade desejável para um algoritmo de minimização aplicado a um problema com restrições de desigualdades é a identificação das restrições ativas. Descrevemos tal propriedade como: dada uma sequência $\{x^k\} \subset \mathcal{X}$ gerada por um algoritmo tal que $x^k \rightarrow x^*$, um ponto estacionário do problema (2.1), então esse algoritmo tem a propriedade de identificação das restrições ativas se existe $k \in \mathbb{N}$ tal que $\mathcal{A}(x^s) \equiv \mathcal{A}(x^*)$, para todo $s \geq k$.

Vários trabalhos têm sido desenvolvidos com o objetivo de estudar propriedades necessárias para um algoritmo ter a característica de identificação finita, por exemplo [10, 11]. Pesquisas têm sido feitas mostrando efetivamente que um determinado algoritmo está equipado com essa propriedade, dentre eles [10] para programação quadrática sequencial, [12] para o gradiente projetado, [20] para subgradiente projetado, [22] para um algoritmo de região de confiança.

Outro trabalho relacionado com a identificação das restrições ativas é o artigo [19], que define uma classe de funções, chamadas de funções identificadoras, capazes de identificar as restrições ativas numa vizinhança suficientemente pequena de um minimizador local isolado. O texto [19] é interessante, pois com o uso das funções nele descritas, podemos obter a identificação em alguns problemas cujo minimizador satisfaz hipóteses mais fracas do que a complementaridade estrita dos multiplicadores de Lagrange, e até em conjuntos descritos por restrições não lineares.

Em problemas de minimização com restrições, tanto as características da função objetivo quanto das restrições são relevantes para determinação de minimizadores do problema. Definiremos a seguir, o cone polar relativo a um conjunto.

Definição 4.2. O **cone polar** ao conjunto \mathcal{X} , denotado por \mathcal{X}° , é o conjunto dos vetores y tais que

$$\mathcal{X}^\circ = \{y \mid y^T x \leq 0, \forall x \in \mathcal{X}\}.$$

Apesar do nosso problema de otimização ser composto por uma função objetivo não suave, vamos abordar alguns aspectos envolvendo problemas de minimização de funções suaves sobre conjuntos convexos que nos ajudarão a desenvolver a teoria sobre a identificação das restrições ativas para problemas não suaves. Consideremos o problema de otimização

$$\begin{aligned} \min \quad & f(x) \\ \text{s.a.} \quad & x \in \mathcal{X} \end{aligned} \tag{4.1}$$

em que $f \in C^1$. Podemos definir dois vetores que desempenham papéis importantes na verificação da estacionariedade de um vetor associada ao problema (4.1). Mas antes, definiremos a projeção de um vetor sobre um conjunto convexo.

Definição 4.3. Dado um conjunto convexo e fechado \mathcal{S} , a **projeção** de um vetor z sobre o conjunto \mathcal{S} , denotada por $P_{\mathcal{S}}(z)$, é o vetor

$$P_{\mathcal{S}}(z) = \underset{v \in \mathcal{S}}{\operatorname{argmin}} \quad \|v - z\|_2. \tag{4.2}$$

Definição 4.4. Dado um ponto $x \in \mathcal{X}$, definimos o vetor **gradiente projetado** de f em x , como a projeção do vetor $-\nabla f(x)$ sobre o conjunto $T_{\mathcal{X}}(x)$, cone tangente ao conjunto \mathcal{X} (cf. Definição 2.1), denotado por $P_{T_{\mathcal{X}}(x)}(-\nabla f(x))$.

Ressaltamos que a Definição 4.4 é coerente, desde que, para a caixa n -dimensional \mathcal{X} , o cone tangente em qualquer ponto é um conjunto convexo e fechado, e portanto, o vetor gradiente projetado existe e é único para todo $x \in \mathcal{X}$.

Definição 4.5. Dado um ponto $x \in \mathcal{X}$, definimos o vetor **gradiente projetado contínuo** de x , denotado por $g_{\mathcal{X}}(x)$, como

$$g_{\mathcal{X}}(x) = P_{T_{\mathcal{X}}(x)}(x - \nabla f(x)) - x,$$

em que $P_{T_{\mathcal{X}}(x)}(x - \nabla f(x))$ é descrito pela Definição 4.3.

As Definições 4.4 e 4.5 podem ser encontradas respectivamente em [12] e [6]. Mostraremos a seguir, um resultado que relaciona a estacionariedade de um ponto $x \in \mathcal{X}$, associada ao problema (4.1), com os vetores gradiente projetado e gradiente projetado contínuo.

Proposição 4.1. Suponhamos que $x^* \in \mathcal{X}$ seja um ponto estacionário do problema (4.1). Então as seguintes afirmações são equivalentes:

- (a) x^* é ponto estacionário do problema;
- (b) $P_{T_{\mathcal{X}}(x^*)}(-\nabla f(x^*)) = 0$;
- (c) $g_{\mathcal{X}}(x^*) = 0$.

Prova: A equivalência (a) \Leftrightarrow (c) pode ser encontrada em [6, Lema 2.1 (b)]. Será mostrada usando um resultado conhecido envolvendo projeções sobre conjuntos não vazios, convexos e fechados. Dado um vetor $x \in \mathbb{R}^n$ temos que

$$(x - P_{T_{\mathcal{X}}(x)}(x))^T(y - P_{T_{\mathcal{X}}(x)}(x)) \leq 0, \quad \forall y \in T_{\mathcal{X}}(x), \quad (4.3)$$

isto é, o segmento formado pela diferença entre um vetor e sua projeção sobre um conjunto convexo e fechado forma um ângulo obtuso com o segmento formado pela sua projeção e qualquer outro vetor do conjunto, ver por exemplo [50, Lema 1.1a].

(a) \Rightarrow (c) Considere que x^* é um ponto estacionário. Então sabemos que $-\nabla f(x^*)^T(y - x^*) \leq 0$, para todo y tal que $y - x^* \in T_{\mathcal{X}}(x^*)$. Usando a equação anterior vemos que

$$(x^* - \nabla f(x^*) - x^*)^T(y - x^*) \leq 0, \quad \forall y \text{ tal que } y - x^* \in T_{\mathcal{X}}(x^*).$$

Aplicando o resultado (4.3) na expressão anterior obtemos que $P_{T_{\mathcal{X}}(x^*)}(x^* - \nabla f(x^*)) = x^*$, ou equivalentemente, $g_{\mathcal{X}}(x^*) = 0$.

(a) \Leftarrow (c) A recíproca segue exatamente o caminho contrário da demonstração de (a) \Rightarrow (c) usando os mesmos argumentos.

A demonstração da equivalência (a) \Leftrightarrow (b) pode ser encontrada em [12, Lema 3.1 (c)], e será transcrita no próximo lema. \square

Lema 4.1. *Seja $P_{T_{\mathcal{X}}(x)}(-\nabla f(x))$ o gradiente projetado de f em $x \in \mathcal{X}$.*

- (i) $\langle -\nabla f(x), P_{T_{\mathcal{X}}(x)}(-\nabla f(x)) \rangle = \|P_{T_{\mathcal{X}}(x)}(-\nabla f(x))\|_2^2$;
- (ii) $\min\{\langle \nabla f(x), v \rangle \mid v \in T_{\mathcal{X}}(x), \|v\|_2 \leq 1\} = -\|P_{T_{\mathcal{X}}(x)}(-\nabla f(x))\|_2$;
- (iii) x é ponto estacionário do problema (4.1) $\Leftrightarrow P_{T_{\mathcal{X}}(x)}(-\nabla f(x)) = 0$.

Prova: Como $T_{\mathcal{X}}(x)$ é um conjunto convexo e fechado, podemos usar a expressão (4.3) para $x = -\nabla f(x)$ e $y = \lambda P_{T_{\mathcal{X}}(x)}(-\nabla f(x))$, para qualquer $\lambda \geq 0$ e vemos que

$$(-\nabla f(x) - P_{T_{\mathcal{X}}(x)}(-\nabla f(x)))^T((\lambda - 1)P_{T_{\mathcal{X}}(x)}(-\nabla f(x))) \leq 0, \quad \forall \lambda \geq 0.$$

Usando a expressão anterior com $\lambda = 0$ e $\lambda = 2$, obtemos que

$$(-\nabla f(x) - P_{T_{\mathcal{X}}(x)}(-\nabla f(x)))^T(P_{T_{\mathcal{X}}(x)}(-\nabla f(x))) = 0,$$

e a última expressão é exatamente o item (i).

Para provar o item (ii) notemos que se $v \in T_{\mathcal{X}}(x)$ e $\|v\|_2 \leq \|P_{T_{\mathcal{X}}(x)}(-\nabla f(x))\|_2$ então pela definição do vetor $P_{T_{\mathcal{X}}(x)}(-\nabla f(x))$ temos que

$$\begin{aligned} \|\nabla f(x) + P_{T_{\mathcal{X}}(x)}(-\nabla f(x))\|_2^2 &\leq \|v + \nabla f(x)\|_2^2 \\ &= \|v\|_2^2 + 2\langle v, \nabla f(x) \rangle + \|\nabla f(x)\|_2^2 \\ &\leq \|P_{T_{\mathcal{X}}(x)}(-\nabla f(x))\|_2^2 + 2\langle v, \nabla f(x) \rangle + \|\nabla f(x)\|_2^2. \end{aligned}$$

Aplicando o item (i) na expressão anterior, obtemos

$$-\|P_{T_{\mathcal{X}}(x)}(-\nabla f(x))\|_2^2 = \langle \nabla f(x), P_{T_{\mathcal{X}}(x)}(-\nabla f(x)) \rangle \leq \langle v, \nabla f(x) \rangle.$$

Dividindo a última desigualdade por $\|P_{T_{\mathcal{X}}(x)}(-\nabla f(x))\|_2$, vemos que $\|v\|_2 \leq 1$, mostrando assim o item (ii).

Suponhamos que x é um ponto estacionário do problema (4.1). Então $\langle \nabla f(x), y - x \rangle \geq 0$, para todo y tal que $y - x \in T_{\mathcal{X}}(x)$. Usando o item (ii) concluímos que $P_{T_{\mathcal{X}}(x)}(-\nabla f(x)) = 0$.

Reciprocamente, supondo que $P_{T_{\mathcal{X}}(x)}(-\nabla f(x)) = 0$, sabemos pelo item (ii) que $\langle \nabla f(x), y - x \rangle \geq 0$, para todo y tal que $y - x \in T_{\mathcal{X}}(x)$ e $\|y - x\|_2 \leq 1$, o que garante a estacionariedade do ponto x , concluindo a demonstração do item (iii) e do lema. \square

Pela Proposição 4.1, vemos que ambos os vetores que definimos se anulam quando estamos em um ponto estacionário do problema (4.1), porém eles possuem algumas diferenças significativas. Enquanto o gradiente projetado contínuo é um operador contínuo sobre o conjunto \mathcal{X} , o operador gradiente projetado é descontínuo sobre o mesmo conjunto.

Assim, o gradiente projetado contínuo pode ser usado como critério de parada para um algoritmo que busca minimizadores para o problema (4.1). Isso é feito, por exemplo, em [6]. Por sua vez, o gradiente projetado não pode ser usado com esse mesmo propósito, pois devido à sua descontinuidade, podemos ter $x^k \rightarrow x^*$ ponto estacionário de (4.1) mas $P_{T_{\mathcal{X}}(x^k)}(-\nabla f(x^k)) \rightarrow 0$.

Quando a função $f \in C^1$, é fácil notar em quais situações o gradiente projetado pode apresentar suas descontinuidades. A primeira é quando deslocamos um vetor do interior do conjunto para a fronteira, e no momento que passamos para a fronteira do conjunto perdemos alguma componente do gradiente, que pela continuidade do mesmo não se anularia, mas depois de projetado sobre o conjunto viável, tal componente é perdida. A segunda situação ocorre quando trocamos de face, isto é, quando as restrições de igualdade variam de um ponto para o outro, perdendo e/ou acrescentando algumas componentes do gradiente por projetar e/ou deixar de projetá-las sobre o conjunto. Evidentemente, pela suposta continuidade da função gradiente, quando nos movemos apenas no interior do conjunto ou sobre um conjunto de restrições de igualdade fixado temos a continuidade no operador gradiente projetado.

Apesar da descontinuidade do gradiente projetado, Calamai e Moré em [12] conseguiram mostrar, sob a hipótese de não degeneração do ponto estacionário¹, que esse operador

¹ Um ponto estacionário do problema (4.1) é não degenerado quando os gradientes das restrições ativas formam um conjunto L.I. e os multiplicadores de Lagrange associados às restrições ativas são não nulos.

pode ser utilizado para verificar quando uma sequência $x^k \rightarrow x^*$, x^* ponto estacionário de (4.1), tem a propriedade de identificação das restrições ativas, isto é, $\mathcal{A}(x^k) \equiv \mathcal{A}(x^*)$ para todo k suficientemente grande, [12, Teorema 4.1].

No Teorema 4.1 de [12], os autores mostraram que uma sequência tem a propriedade de identificação das restrições ativas se, e somente se, $P_{T_{\mathcal{X}}(x)}(-\nabla f(x^k)) \rightarrow 0$ desde que o ponto limite x^* seja um ponto estacionário não degenerado. Com as ideias apresentadas sobre a continuidade e descontinuidade do operador gradiente projetado e pela Proposição 4.1, conseguimos observar que a única forma do gradiente projetado convergir para zero seria quando x^* pertence ao interior do conjunto \mathcal{X} ou se, a partir de algum índice $\bar{k} \in \mathbb{N}$, os vetores x^k , para $k \geq \bar{k}$, pertencem à face ótima relativa ao conjunto \mathcal{X} .

Notemos também que apesar da continuidade do operador gradiente projetado contínuo, não podemos utilizá-lo com o mesmo propósito do gradiente projetado, isto é, com o intuito de identificar as restrições ativas, desde que independente da maneira como uma sequência $\{x^k\}$ se aproxima de um ponto estacionário x^* , o gradiente projetado contínuo em qualquer situação sempre converge para zero, devido à sua continuidade.

Voltemos ao nosso problema particular. Enunciaremos um resultado encontrado em [4, Proposição 4.7.3] que caracteriza os minimizadores do problema (2.1) a partir da sua estrutura.

Proposição 4.2. *Seja x^* um minimizador local para f sobre um conjunto $\mathcal{X} \subset \mathbb{R}^n$. Assuma que f tem a forma*

$$f(x) = f_1(x) + f_2(x),$$

onde f_1 é convexa e f_2 é suave. Então:

$$-\nabla f_2(x^*) \in \partial f_1(x^*) + T_{\mathcal{X}}^{\circ}(x^*), \quad (4.4)$$

em que $\partial f_1(x^*)$ denota o subdiferencial de f_1 no ponto x^* e $T_{\mathcal{X}}^{\circ}(x^*)$ denota o cone polar ao cone $T_{\mathcal{X}}(x^*)$.

Sabemos que, para o conjunto \mathcal{X} , o conjunto $T_{\mathcal{X}}^{\circ}(x^*)$ pode ser representado da forma

$$T_{\mathcal{X}}^{\circ}(x^*) = \left\{ v \mid v = - \sum_{i \in \mathcal{A}(x^*) \cap \{1, \dots, n\}} \lambda_i e_i + \sum_{n+j \in \mathcal{A}(x^*) \cap \{n+1, \dots, 2n\}} \lambda_j e_j, \text{ com } \lambda_i, \lambda_j \geq 0. \right\} \quad (4.5)$$

Equivalentemente à Proposição 4.2, porém substituindo $f_1 = \psi$ e $f_2 = f$, podemos reescrever a condição de estacionariedade (4.4) utilizando a representação (4.5) como

$$\nabla f(x^*) + d = \sum_{i \in \mathcal{A}(x^*) \cap \{1, \dots, n\}} \lambda_i e_i - \sum_{n+j \in \mathcal{A}(x^*) \cap \{n+1, \dots, 2n\}} \lambda_j e_j,$$

para algum $d \in \partial \psi(x^*)$ e $\lambda_i, \lambda_j \geq 0$.

Mostraremos um resultado similar aos que foram apresentados nos textos [12] e [20], buscando descrever as características que uma sequência de vetores convergente a um ponto estacionário deve possuir para ter a propriedade de identificação procurada. Antes, apresentaremos uma definição de não degeneração para um ponto estacionário do problema (2.1).

Definição 4.6. Diremos que um **ponto estacionário** x^* do problema (2.1) é **não degenerado** quando existe $\bar{d} \in \partial\psi(x^*)$ tal que

$$\nabla f(x^*) + \bar{d} = \sum_{i \in \mathcal{A}(x^*) \cap \{1, \dots, n\}} \lambda_i e_i - \sum_{n+j \in \mathcal{A}(x^*) \cap \{n+1, \dots, 2n\}} \lambda_j e_j, \text{ com } \lambda_i, \lambda_j > 0. \quad (4.6)$$

Proposição 4.3. Suponhamos que x^* é um ponto estacionário não degenerado do problema (2.1) e $\{x^k\} \subset \mathcal{X}$ é uma sequência tal que $x^k \rightarrow x^*$. Adicionalmente, suponha a existência de vetores $d^k \in \partial\psi(x^k)$ tais que $d^k \rightarrow \bar{d}$ com $\nabla f(x^*) + \bar{d}$ satisfazendo (4.6). Se

$$\lim_{k \rightarrow +\infty} P_{T_{\mathcal{X}}(x^k)}[-\nabla f(x^k) - d^k] = 0$$

então $\mathcal{A}(x^k) \equiv \mathcal{A}(x^*)$, para todo k suficientemente grande.

Prova: Como $x^k \rightarrow x^*$, e as restrições são contínuas, temos que $\mathcal{A}(x^k) \subset \mathcal{A}(x^*)$, para k suficientemente grande. Vamos supor, por absurdo, que exista algum subconjunto infinito de índices $\mathcal{K}_1 \subset \mathbb{N}$ tal que $s \in \mathcal{A}(x^*)$ mas $s \notin \mathcal{A}(x^k)$ para todo $k \in \mathcal{K}_1$.

Como $s \notin \mathcal{A}(x^k)$ para todo $k \in \mathcal{K}_1$ temos que $\{e_s, -e_s\} \subset T_{\mathcal{X}}(x^k)$ para todo $k \in \mathcal{K}_1$, pois

$$T_{\mathcal{X}}(x^k) = \left\{ v \mid \begin{array}{ll} e_i^T v \leq 0, & \text{se } i \in \mathcal{A}(x^k) \cap \{1, \dots, n\} \text{ e} \\ e_i^T v \geq 0, & \text{se } n+i \in \mathcal{A}(x^k) \cap \{n+1, \dots, 2n\} \end{array} \right\}.$$

Vamos dividir a prova em dois casos:

1° Caso: $x_s^* = l_s$, isto é, $s \in \{1, \dots, n\}$. Usando o Lema 4.1 (ii), obtemos

$$\begin{aligned} -\langle -\nabla f(x^k) - d^k, e_s \rangle &\leq \|P_{T_{\mathcal{X}}(x^k)}[-\nabla f(x^k) - d^k]\|_2 \\ \Leftrightarrow \langle \nabla f(x^k) + d^k, e_s \rangle &\leq \|P_{T_{\mathcal{X}}(x^k)}[-\nabla f(x^k) - d^k]\|_2. \end{aligned} \quad (4.7)$$

Passando ao limite os dois membros de (4.7), temos $\langle \nabla f(x^*) + \bar{d}, e_s \rangle \leq 0$. Por outro lado, como x^* é um ponto estacionário não degenerado e $s \in \mathcal{A}(x^*)$ vemos que

$$\begin{aligned} \langle \nabla f(x^*) + \bar{d}, e_s \rangle &= \left\langle \sum_{i \in \mathcal{A}(x^*) \cap \{1, \dots, n\}} \lambda_i e_i - \sum_{n+j \in \mathcal{A}(x^*) \cap \{n+1, \dots, 2n\}} \lambda_j e_j, e_s \right\rangle \\ &= \lambda_s \|e_s\|_2^2 \\ &= \lambda_s > 0. \end{aligned}$$

2° Caso: $x_s^* = u_s$, isto é, $n+s \in \{n+1, \dots, 2n\}$. Pelo mesmo argumento usado em (4.7),

$$\begin{aligned} -\langle -\nabla f(x^k) - d^k, e_s \rangle &\leq \|P_{T_{\mathcal{X}}(x^k)}[-\nabla f(x^k) - d^k]\|_2 \\ \Leftrightarrow \langle \nabla f(x^k) + d^k, -e_s \rangle &\leq \|P_{T_{\mathcal{X}}(x^k)}[-\nabla f(x^k) - d^k]\|_2. \end{aligned} \quad (4.8)$$

Passando ao limite os dois membros de (4.8), temos $\langle \nabla f(x^*) + \bar{d}, -e_s \rangle \leq 0$. Analogamente ao caso anterior, temos

$$\begin{aligned} \langle \nabla f(x^*) + \bar{d}, -e_s \rangle &= \left\langle \sum_{i \in \mathcal{A}(x^*) \cap \{1, \dots, n\}} \lambda_i e_i - \sum_{n+j \in \mathcal{A}(x^*) \cap \{n+1, \dots, 2n\}} \lambda_j e_j, -e_s \right\rangle \\ &= \langle -\lambda_s e_s, -e_s \rangle \\ &= \lambda_s \|e_s\|_2^2 \\ &= \lambda_s > 0. \end{aligned}$$

Essas contradições, para os dois casos, provam que $\mathcal{A}(x^k) \equiv \mathcal{A}(x^*)$, para todo k suficientemente grande. \square

A Proposição 4.3 apresenta qual característica devemos procurar em um algoritmo para garantir que este possua a propriedade de identificação das restrições ativas.

Nosso objetivo agora é apresentar um resultado que garanta que os pontos de acumulação de uma sequência gerada pelo Algoritmo 1, sob certas hipóteses, satisfazem a propriedade descrita pela Proposição 4.3, desde que, pela Seção 3.1, sabemos que eles são pontos estacionários para o problema (2.1).

Considere uma sequência $\{x^k\}$ gerada pelo Algoritmo 1. Por simplicidade, vamos supor que a sequência converge para um ponto estacionário $\{x^k\} \rightarrow x^*$ com probabilidade 1. Diferentemente do Teorema 3.1, que garantia a convergência para todo ponto de acumulação da sequência, a hipótese atual simplificará a notação.

Para essa sequência $\{x^k\} \subset \mathcal{X}$, sabemos que o vetor x^{k+1} , obtido a partir de x^k atualizando o bloco de coordenadas $i \in \{1, \dots, m\}$, satisfaz a relação a seguir

$$\begin{aligned} x^{k+1} = \operatorname{argmin} \quad & \left\{ \sum_{j=1}^m \nabla_j f(x^k)^T (x_{(j)} - x_{(j)}^k) + \sum_{j=1}^m \frac{L_j}{2} (x_{(j)} - x_{(j)}^k)^T B_j^k (x_{(j)} - x_{(j)}^k) + \sum_{j=1}^m \psi_j(x_{(j)}) \right\} \\ \text{s.a.} \quad & l \leq x \leq u \\ & x_{(j)} = x_{(j)}^k, \forall j \neq i \in \{1, \dots, m\}, \end{aligned} \quad (4.9)$$

em que a expressão (4.9) vem da expressão (2.6) substituindo $\|h_i\|_{(i)}^2 = h_i^T B_i^k h_i$, o vetor h_i por $x_{(i)} - x_{(i)}^k$ e escrevendo $x_{(i)}^{k+1} = x_{(i)}^k + h_i(x^k)$, para todo $i \in \{1, \dots, m\}$.

Pela expressão (4.9) e pela Proposição 4.2, vemos que

$$-\nabla_j f(x^k) - L_j B_j^k (x_{(j)}^{k+1} - x_{(j)}^k) = p_j^{k+1} + v_j^{k+1} + w_j^{k+1}, \forall k \in \mathbb{N} \text{ e } \forall j \in \{1, \dots, m\}, \quad (4.10)$$

para algum $p_j^{k+1} \in \partial \psi_j(x_{(j)}^{k+1})$ e algum $v_j^{k+1} \in T_{\mathcal{X}_{(j)}}^\circ(x_{(j)}^{k+1})$, com $\mathcal{X}_{(j)} = \{x_{(j)} \mid l_{(j)} \leq x_{(j)} \leq u_{(j)}\}$ e w_j^{k+1} é o multiplicador de Lagrange associado às restrições lineares $x_{(j)} = x_{(j)}^k$ de (4.9), com a convenção que caso o bloco de coordenadas j tenha sido atualizado na iteração k temos $w_j^{k+1} = 0$.

Antes de mostrarmos o principal resultado desta seção, apresentaremos um lema auxiliar.

Lema 4.2. *Sejam $w_j^k \in \mathbb{R}^{p_j}$, $j \in \{1, \dots, m\}$ sequências de vetores dadas por*

$$\begin{cases} w_j^k = 0, & \text{se o bloco de coordenadas } j \text{ foi} \\ & \text{atualizado na iteração } k-1; \\ w_j^k = -\nabla_j f(x^{k-1}) + \nabla_j f(x^s) + L_j B_j^s(x_{(j)}^{s+1} - x_{(j)}^s), & \text{caso contrário,} \end{cases}$$

em que s é a primeira iteração menor do que $k-1$ tal que o j -ésimo bloco foi atualizado e a sequência de vetores $\{x^k\} \rightarrow x^*$ é construída pelo Algoritmo 1. Então, $w_j^k \rightarrow 0$, para todo $j \in \{1, \dots, m\}$ com probabilidade 1.

Prova: Para cada $j \in \{1, \dots, m\}$ fixo, construímos as sequências $\{y^{k,j}\}, \{C^{k,j}\}$ tais que $y^{k,j} = x^k$, $C^{k,j} = B^k$ se o bloco de coordenadas j foi atualizado na iteração k e $y^{k,j} = y^{k-1,j}$, $C^{k,j} = C^{k-1,j}$ caso contrário. Usando o Lema 3.2, sabemos que todos os blocos de coordenadas são atualizados um número infinito de vezes com probabilidade 1, portanto, vemos que $y^{k,j} \rightarrow x^*$ com probabilidade 1.

Considerando a subsequência $\{w_j^k\}_{\mathcal{K}'_j}$, na qual

$$\mathcal{K}'_j = \{k \mid \text{o bloco de coordenadas } j \text{ não foi atualizado na iteração } k\},$$

obtemos que

$$\begin{aligned} \|w_j^k\|_2 &= \|-\nabla_j f(x^{k-1}) + \nabla_j f(y^{k,j}) + L_j C^{k,j}(x_{(j)}^k - y_{(j)}^{k,j})\|_2 \\ &\leq \|-\nabla_j f(x^{k-1}) + \nabla_j f(y^{k,j})\|_2 + L_j \|C^{k,j}\|_2 \|x_{(j)}^k - y_{(j)}^{k,j}\|_2, \quad \forall k \in \mathcal{K}'_j. \end{aligned} \quad (4.11)$$

Pela continuidade de $\nabla_j f$, pela limitação das matrizes $C^{k,j}$, pela convergência dos vetores $y^{k,j}, x^k \rightarrow x^*$, vemos que $\{w_j^k\} \rightarrow 0$, $k \in \mathcal{K}'_j$, com probabilidade 1.

Como a subsequência $\{w_j^k\}_{\mathcal{K}'_j}$ foi obtida a partir da sequência $\{w_j^k\}$ retirando os elementos nulos desta sequência e garantimos que $\{w_j^k\} \rightarrow 0$, $k \in \mathcal{K}'_j$, temos que $w_j^k \rightarrow 0$, para todo $j \in \{1, \dots, m\}$, com probabilidade 1. \square

Proposição 4.4. *Suponhamos que x^* , ponto estacionário da sequência $x^k \rightarrow x^*$ gerada pelo Algoritmo 1, seja um ponto estacionário não degenerado do problema (2.1), segundo a Definição 4.6. Se as matrizes $B_j^k \in \mathbb{S}_{++}^n$ usadas a cada iteração do algoritmo são limitadas e os vetores p^k dados por (4.10) satisfazem $p^k \in \partial\psi(x^k) \rightarrow d^*$, em que $\nabla f(x^*) + d^*$ obedece a relação (4.6), então essa sequência possui a propriedade de identificação das restrições ativas com probabilidade 1.*

Prova: Pela expressão (4.10) e pela hipótese sobre a sequência de vetores $\{p^k\}$, para mostrar o desejado, precisamos garantir inicialmente que a sequência $w_j^k \rightarrow 0$, para todo j , com probabilidade 1.

Como já explicado, se na iteração k o bloco de coordenadas j foi atualizado temos $w_j^{k+1} = 0$. E caso essa coordenada tenha sido atualizada pela última vez em alguma iteração $s < k$ e não foi atualizada na iteração k , obtemos

$$\begin{aligned} -\nabla_j f(x^k) - L_j B_j^k(x_{(j)}^{k+1} - x_{(j)}^k) &= -\nabla_j f(x^k) \\ &= -\nabla_j f(x^k) + \nabla_j f(x^s) + L_j B_j^s(x_{(j)}^{s+1} - x_{(j)}^s) - b_j^s \end{aligned} \quad (4.12)$$

com $b_j^s = \nabla_j f(x^s) + L_j B_j^s(x_{(j)}^{s+1} - x_{(j)}^s)$.

Por (4.10) vemos que $-b_j^s = p_j^{s+1} + v_j^{s+1}$ e como o bloco j não foi atualizado entre as iterações s e k , temos $\partial\psi_j(x_{(j)}^{s+1}) \equiv \partial\psi_j(x_{(j)}^{k+1})$ e $T_{\mathcal{X}_{(j)}}^\circ(x_{(j)}^s) \equiv T_{\mathcal{X}_{(j)}}^\circ(x_{(j)}^k)$. Dessas observações e tomando $w_j^{k+1} = -\nabla_j f(x^k) + \nabla_j f(x^s) + L_j B_j^s(x_{(j)}^{s+1} - x_{(j)}^s)$, obtemos por (4.12)

$$-\nabla_j f(x^k) - L_j B_j^k(x_{(j)}^{k+1} - x_{(j)}^k) = p_j^{s+1} + v_j^{s+1} + w_j^{k+1}, \quad (4.13)$$

em que $p_j^{s+1} \in \partial\psi_j(x_{(j)}^k)$, $v_j^{s+1} \in T_{\mathcal{X}_{(j)}}^\circ(x_{(j)}^{k+1})$. De (4.13), mostramos que a sequência $\{w_j^k\}$ pode ser descrita como

$$\begin{cases} w_j^k = 0, & \text{se o bloco de coordenadas } j \text{ foi} \\ & \text{atualizado na iteração } k-1; \\ w_j^k = -\nabla_j f(x^{k-1}) + \nabla_j f(x^s) + L_j B_j^s(x_{(j)}^{s+1} - x_{(j)}^s), & \text{caso contrário,} \end{cases}$$

em que s é a primeira iteração menor do que $k-1$ tal que o j -ésimo bloco foi atualizado.

Aplicando o Lema 4.2, obtemos que $w_j^k \rightarrow 0$, com probabilidade 1, como desejado.

Pela decomposição de Moreau, veja por exemplo [50, Lema 2.2], sabemos que qualquer vetor $d \in \mathbb{R}^n$ pode ser decomposto como soma direta entre as projeções em um cone convexo e fechado \mathcal{S} e em seu cone polar \mathcal{S}° , isto é

$$d = P_{\mathcal{S}}[d] \oplus P_{\mathcal{S}^\circ}[d].$$

Como para o nosso problema, o conjunto $T_{\mathcal{X}}(x^k)$ sempre é um cone convexo e fechado para qualquer $x^k \in \mathbb{R}^n$, obtemos

$$-\nabla f(x^k) - p^k = P_{T_{\mathcal{X}}(x^k)}[-\nabla f(x^k) - p^k] \oplus P_{T_{\mathcal{X}}^\circ(x^k)}[-\nabla f(x^k) - p^k]$$

e por essa decomposição em soma direta temos que

$$\|P_{T_{\mathcal{X}}(x^k)}[-\nabla f(x^k) - p^k]\|_2 \leq \|-\nabla f(x^k) - p^k - v^k\|_2, \quad \forall v^k \in T_{\mathcal{X}}^\circ(x^k). \quad (4.14)$$

Em particular, $P_{T_{\mathcal{X}}^\circ(x^k)}[-\nabla f(x^k) - p^k] \in T_{\mathcal{X}}^\circ(x^k)$ é o único vetor para o qual a igualdade em (4.14) é válida.

Usando a expressão (4.14), vemos que

$$\begin{aligned} \|P_{T_{\mathcal{X}}(x^k)}[-\nabla f(x^k) - p^k]\|_2^2 &\leq \|-\nabla f(x^k) - p^k - v^k\|_2^2 \\ &= \sum_{j=1}^m \|-\nabla_j f(x^k) - p_j^k - v_j^k\|_2^2, \quad \forall v^k \in T_{\mathcal{X}}^\circ(x^k). \end{aligned} \quad (4.15)$$

Substituindo (4.10) em (4.15), obtemos

$$\begin{aligned} \|P_{T_{\mathcal{X}}(x^k)}[-\nabla f(x^k) - p^k]\|_2^2 &\leq \sum_{j=1}^m \|-\nabla_j f(x^k) - p_j^k - [-\nabla_j f(x^{k-1}) - L_j B_j^{k-1}(x_{(j)}^k - x_{(j)}^{k-1}) - p_j^k - w_j^k]\|_2^2 \\ &= \sum_{j=1}^m \|-\nabla_j f(x^k) + \nabla_j f(x^{k-1}) + L_j B_j^{k-1}(x_{(j)}^k - x_{(j)}^{k-1}) + w_j^k\|_2^2 \\ &\leq \sum_{j=1}^m \|-\nabla_j f(x^k) + \nabla_j f(x^{k-1})\|_2^2 + \sum_{j=1}^m L_j^2 \|B_j^{k-1}\|_2^2 \|x_{(j)}^k - x_{(j)}^{k-1}\|_2^2 \\ &\quad + \sum_{j=1}^m \|w_j^k\|_2^2. \end{aligned} \quad (4.16)$$

Pela continuidade de ∇f , pela limitação das matrizes B_j^k e pela convergência das sequências $y^k, x^k \rightarrow x^*, w^k \rightarrow 0$, com probabilidade 1, vemos por (4.16) que

$$\lim_{k \rightarrow +\infty} P_{T_{\mathcal{X}}(x^k)}[-\nabla f(x^k) - p^k] = 0,$$

é válida, com probabilidade 1.

Pela última expressão, podemos aplicar a Proposição 4.3, garantindo que $\mathcal{A}(x^k) \equiv \mathcal{A}(x^*)$, para todo k suficientemente grande, com probabilidade 1, o que conclui a demonstração. \square

Uma observação importante feita pelos autores Hare e Lewis [23] foi que a propriedade de identificação das restrições ativas só é verificada por uma sequência gerada pelo método do gradiente projetado escalado puro, isto é, sem nenhuma modificação na maneira como se muda de face durante a execução do método, quando o ponto estacionário do problema é não degenerado. Nesse texto, eles mostraram um exemplo desse fato para o caso em que o conjunto de restrições é convexo mas formado por restrições não-lineares [23, Exemplo 4.1], mostraremos aqui um outro exemplo, em que o conjunto de restrições do problema é formado por restrições lineares.

Exemplo 4.1. *Considere o problema*

$$\begin{aligned} \min \quad & \frac{1}{4}x^2 + \frac{1}{4}y^2 \\ \text{s.a.} \quad & -x \leq 0 \\ & -y \leq 0 \end{aligned} \tag{4.17}$$

O conjunto de pontos viáveis com relação ao problema anterior é

$$\mathcal{Y} = \{(x, y) \mid x \geq 0 \text{ e } y \geq 0\}.$$

O minimizador global para este problema é o ponto $(x, y) = (0, 0)$, desde que o mesmo é também minimizador global para o problema irrestrito. Vemos que o ponto $(x^, y^*) = (0, 0)$ é um ponto estacionário degenerado, pois*

$$-\nabla f(x^*, y^*) = \lambda_1 \nabla g_1(x^*, y^*) + \lambda_2 \nabla g_2(x^*, y^*)$$

em que $\nabla f(0, 0) = (0, 0)^T$, $\nabla g_1(0, 0) = (1, 0)^T$ e $\nabla g_2(0, 0) = (0, -1)^T$, $\lambda_1 = \lambda_2 = 0$.

Tomando como ponto inicial $(x^0, y^0) = (1, 1)^T$ e notando que $\nabla f(x, y) = \left(\frac{x}{2}, \frac{y}{2}\right)^T$, para todo $x, y \in \mathbb{R}$, por um argumento de indução finita, podemos mostrar que o método do gradiente projetado puro aplicado a este problema de minimização a partir do ponto (x^0, y^0) gera a sequência de pontos

$$\begin{pmatrix} x^k \\ y^k \end{pmatrix} = P_{T_{\mathcal{Y}}(x^k, y^k)} \left[\begin{pmatrix} \frac{1}{2^k} \\ \frac{1}{2^k} \end{pmatrix} \right] = \begin{pmatrix} \frac{1}{2^k} \\ \frac{1}{2^k} \end{pmatrix}.$$

Notemos que, para todo $k \in \mathbb{N}$, as sequências $\{x^k\}, \{y^k\}$ têm valores positivos e convergem para 0, e portanto, não atingem a face ótima em um número finito de iterações. Logo, nesse problema existem sequências geradas pelo método do gradiente projetado que não identificam as restrições ativas, e portanto, a hipótese de não degeneração é essencial para a identificação.

O texto [23] também mostra um exemplo garantindo que a hipótese de não degeneração do ponto estacionário é essencial para a identificação das restrições ativas de qualquer sequência gerada pelo método de Newton projetado, [23, Exemplo 4.2]. Desses exemplos e da Observação 2.6, vemos que para nosso algoritmo possuir a propriedade de identificação, não podemos abrir mão da hipótese de não degeneração do ponto estacionário do problema (2.1).

4.2 Funções Identificadoras

Como vimos pelo Exemplo 4.1, sem alguma estratégia específica permitindo que se encontre a face ótima com relação a um ponto estacionário do problema (2.1), não conseguimos garantir a propriedade de identificação das restrições ativas do nosso algoritmo quando esse ponto é degenerado. Mesmo para o caso de um ponto estacionário não degenerado, existem algoritmos que na sua forma pura não são capazes de encontrar a face ótima do problema em tempo finito.

Por isso, estratégias que nos permitam encontrar a face ótima independentemente do algoritmo empregado são importantes de serem estudadas, principalmente por melhorarem as propriedades de convergência do algoritmo em questão. Como já citamos, o texto [19] se propõe a apresentar uma estratégia para encontrar a face ótima mesmo quando o ponto estacionário é degenerado. Nosso objetivo é, utilizando as ideias empregadas em [19], estender os resultados dos autores para problemas de minimização com a estrutura presente em (2.1).

A ideia da função identificadora é construir uma função $\rho : \mathbb{R}^p \rightarrow \mathbb{R}_-$ que, numa vizinhança de um ponto estacionário, seja capaz de dizer quais são as restrições ativas e inativas ao problema de interesse. No texto [19], os autores definiram esse conceito a partir do par de vetores primal e dual que satisfazem o sistema de equações KKT do problema de minimização de interesse. Mas como no nosso contexto prático é inviável trabalhar com as variáveis primal e dual, vamos reformular a definição de função identificadora envolvendo apenas a variável primal e visando identificar as restrições ativas dos pontos do conjunto \mathcal{P} ,

$$\mathcal{P} = \{x^* \mid x^* \text{ é um ponto estacionário do problema (2.1)}\}.$$

Apresentaremos, a seguir, uma definição para o conceito de função identificadora, importante para o desenvolvimento da seção. Ela é baseada em [19, Definição 2.1] e depende de um ponto estacionário pré-fixado inicialmente.

Definição 4.7. Diremos que uma função $\rho : \mathbb{R}^n \rightarrow \mathbb{R}_-$ é uma **função identificadora** para um ponto $x^* \in \mathcal{P}$, quando

- (i) ρ é contínua em um aberto contendo x^* ;

- (ii) $\rho(x^*) = 0$;
- (iii) $\rho(x) = 0 \Rightarrow x \in \mathcal{P}$;
- (iv) $\lim_{\substack{x^k \rightarrow x^* \\ x^k \notin \mathcal{P}}} \frac{-\rho(x^k)}{\|x^k - x^*\|_2} = +\infty$, para toda sequência $\{x^k\} \subset \mathcal{X}$ tal que $x^k \rightarrow x^*$, com $x^* \in \mathcal{P}$ e $x^k \notin \mathcal{P}$, para todo k .

Os quatro itens da definição anterior terão sua importância esclarecida no decorrer do texto. A quarta propriedade é a mais exigente e mais difícil de ser verificada na prática, como será observado na Seção 4.2.1. Note que, sequências geradas por qualquer algoritmo que busque pontos estacionários para o problema (2.1), claramente satisfazem parcialmente a Definição 4.7 (iv), isto é, geram pontos x^k com $x^k \notin \mathcal{P}$.

O próximo teorema foi obtido com sutis modificações do Teorema 2.2 de [19]. Porém, para facilitar a demonstração do resultado, definiremos dois conjuntos de índices relacionados com a identificação.

Definição 4.8. Definimos $\hat{\mathcal{A}}(x)$ e $\mathcal{A}(x)$, com $x \in \mathbb{R}^n$, como os subconjuntos de índices dados por

$$\hat{\mathcal{A}}(x) = \{ i \in \{1, \dots, m\} \mid g_i(x) \geq \rho(x) \},$$

$$\mathcal{A}(x) = \{ i \in \{1, \dots, m\} \mid g_i(x) = 0 \}.$$

Teorema 4.1. Sejam ρ uma função identificadora para $x^* \in \mathcal{P}$ e g uma função de classe $C^1(\mathbb{R}^m)$. Então, existe $\epsilon > 0$ tal que

$$\mathcal{A}(x^*) \equiv \hat{\mathcal{A}}(x), \quad \forall x \in \mathcal{B}(x^*, \epsilon), \quad \text{com } x \notin \mathcal{P}.$$

Prova: Como g é de classe C^1 sabemos que g é localmente Lipschitz contínua, assim existe $\epsilon_2 > 0$ tal que

$$-g_i(x) \leq -g_i(x^*) + c\|x - x^*\|_2, \quad i = 1, \dots, m \quad (4.18)$$

para algum $c > 0$ e para todo $x \in \mathcal{B}(x^*, \epsilon_2)$.

Suponhamos que $i \in \mathcal{A}(x^*)$. Por definição, temos que $g_i(x^*) = 0$. Pela Definição 4.7 itens (ii), (iii) e (iv), existe um $\epsilon_3 > 0$ em que

$$c\|x - x^*\|_2 \leq -\rho(x), \quad \forall x \in \mathcal{B}(x^*, \epsilon_3) \text{ e } x \notin \mathcal{P}. \quad (4.19)$$

Fazendo a interseção das vizinhanças de (4.18) e (4.19), vemos que

$$g_i(x) \geq \rho(x),$$

$\forall x \notin \mathcal{P}$ e $x \in \mathcal{B}(x^*, \epsilon_2) \cap \mathcal{B}(x^*, \epsilon_3)$. Logo, $i \in \hat{\mathcal{A}}(x)$.

Para mostrar a inclusão contrária, mostraremos a relação contrapositiva, isto é, $i \notin \mathcal{A}(x^*) \Rightarrow i \notin \hat{\mathcal{A}}(x)$, para todo $x \notin \mathcal{P}$ e pertencente a uma vizinhança de x^* . Se $i \notin \mathcal{A}(x^*)$,

então $g_i(x^*) < 0$. Pelos itens (i), (ii) e (iii) da Definição 4.7, sabemos que ρ é contínua num aberto contendo x^* e só pode se anular em pontos do conjunto \mathcal{P} . Isto, juntamente com a continuidade da função g , garante a existência de uma vizinhança do ponto x^* tal que $g_i(x) < \rho(x)$ para todo x pertencente a essa vizinhança. Logo, $i \notin \hat{\mathcal{A}}(x)$. \square

Na próxima seção, mostraremos um exemplo de função identificadora garantindo a possibilidade da aplicação desses conceitos na identificação das restrições ativas em nosso problema de minimização.

4.2.1 Exemplo de Função Identificadora

Apresentaremos um exemplo de função identificadora que está relacionado com o vetor h de (2.6), quando obtemos uma direção de descida para o problema (2.1). Assim, poderemos buscar outra maneira de encontrar as restrições ativas de pontos estacionários do problema de interesse, sem prejudicar significativamente o custo computacional envolvido.

Vamos definir o vetor $h(x)$ como o minimizador do problema

$$\begin{aligned} \min_h \quad & \sum_{i=1}^m \left(\nabla_i f(x)^T h_i + \frac{\beta L_i}{2} h_i^T B_i h_i + \psi_i(x_{(i)} + h_i) \right) \\ \text{s.a.} \quad & l_{(i)} \leq x_{(i)} + h_i \leq u_{(i)}, \quad i \in \{1, \dots, m\}. \end{aligned} \quad (4.20)$$

Adicionalmente, definiremos o vetor x^+ e $x_{(i)}^+$, como $x^+ = x + h(x)$ e $x_{(i)}^+ = x_{(i)} + h_i(x)$, respectivamente, para α fixo, em que $h_i(x)$ representa o minimizador do problema

$$\begin{aligned} \min_{h_i} \quad & \nabla_i f(x)^T h_i + \frac{\beta L_i}{2} h_i^T B_i h_i + \psi_i(x_{(i)} + h_i) \\ \text{s.a.} \quad & l_{(i)} \leq x_{(i)} + h_i \leq u_{(i)}. \end{aligned} \quad (4.21)$$

Como nosso próximo exemplo de função identificadora dependerá do vetor $h(x)$, precisamos que esse vetor seja unicamente definido a partir de x e α fixos. Pela convexidade estrita da função objetivo do problema (4.20), podemos garantir que, se o problema (4.20) tem um minimizador, ele é único.

Nosso objetivo nesta seção é mostrar que a função $\rho_1 : \mathbb{R}^n \rightarrow \mathbb{R}_-$ definida por

$$\rho_1^\alpha(x) = -\|h(x)\|_2^\alpha, \quad \text{com } \alpha \in (0, 1) \text{ fixo}, \quad (4.22)$$

é uma função identificadora para os pontos do conjunto \mathcal{P} , desde que a parte suave do problema (2.1) satisfaça certas propriedades.

No texto [44], os autores propuseram uma função identificadora semelhante a ρ_1 , para o problema específico $\min_{x \in \mathbb{R}^n} \frac{1}{2} \|y - x\|_2^2 + \lambda \|x\|_1$ com $B_i = I$, para todo $i \in \{1, \dots, m\}$ e conseguiram mostrar que essa função identifica as coordenadas tais que $x_i = 0$ reescrevendo o problema originalmente irrestrito como um problema com restrições. Eles também conseguiram mostrar que a função x^+ , associada a tal problema, satisfaz algumas propriedades similares ao operador projeção sobre um conjunto convexo e fechado. Nosso objetivo é mostrar um resultado com

algumas propriedades semelhantes às encontradas no Lema 2.1 de [44]. Para isso vamos extrair algumas propriedades de um ponto estacionário do problema (2.1) reescrevendo-o de outra forma.

Considere o problema

$$\begin{aligned} \min_{(x,z)} \quad & f(x) + z \\ \text{s.a.} \quad & \psi(x) \leq z \\ & x \in \mathcal{X}, \end{aligned} \quad (4.23)$$

com \mathcal{X} dado em (2.2).

Definiremos o conjunto das pontos viáveis para o problema (4.23) como

$$\Omega = \{(x, z) \mid \psi(x) \leq z \text{ e } x \in \mathcal{X}\}.$$

Consideraremos nessa subseção $g : \mathbb{R}^n \rightarrow \mathbb{R}_-^{2n}$ da forma

$$g(x) = \begin{pmatrix} l - x \\ x - u \end{pmatrix}.$$

Sabemos que as soluções do problema (2.1) e (4.23) estão relacionadas. O vetor x^* é minimizador do problema (2.1) se, e somente se, $(x^*, \psi(x^*))$ é minimizador de (4.23). Pela Definição 2.2, vemos que um ponto estacionário $(x^*, \psi(x^*))$ do problema (4.23) satisfaz a relação

$$\begin{pmatrix} \nabla f(x^*) \\ 1 \end{pmatrix}^T d \geq 0, \quad \forall d \in T_\Omega(x^*, \psi(x^*)). \quad (4.24)$$

Em particular, a expressão (4.24) é válida para toda direção viável $d \in \Omega$ a partir do ponto $(x^*, \psi(x^*))$, isto é

$$\begin{aligned} & \begin{pmatrix} \nabla f(x^*) \\ 1 \end{pmatrix}^T \begin{pmatrix} x - x^* \\ z - \psi(x^*) \end{pmatrix} \geq 0, \quad \forall (x, z) \in \Omega, \\ \Leftrightarrow & \nabla f(x^*)^T (x - x^*) + (z - \psi(x^*)) \geq 0, \quad \forall (x, z) \in \Omega. \end{aligned} \quad (4.25)$$

Equivalentemente à expressão obtida para o problema (4.23) e pela separabilidade do problema (4.20), podemos encontrar uma expressão para as coordenadas por blocos do minimizador $h(x)$ reescrevendo-o como o seguinte conjunto de problemas

$$\begin{aligned} \min_{(h_i, z_i)} \quad & \nabla_i f(x)^T h_i + \frac{\beta L_i}{2} h_i^T B_i h_i + z_i \\ \text{s.a.} \quad & \psi_i(x_{(i)} + h_i) \leq z_i \\ & l_{(i)} \leq x_{(i)} + h_i \leq u_{(i)} \end{aligned} \quad (4.26)$$

para todo $i \in \{1, \dots, m\}$.

Definiremos os conjuntos $\Omega_i, i \in \{1, \dots, m\}$ como

$$\Omega_i = \{(y_{(i)}, z_i) \mid \psi_i(y_{(i)}) \leq z_i \text{ e } l_{(i)} \leq y_{(i)} \leq u_{(i)}\}.$$

Pelo formato do i -ésimo problema (4.26), equivalentemente às relações obtidas em (4.25), se $(h_i(x), \psi_i(x_{(i)} + h_i(x)))$, para todo $i \in \{1, \dots, m\}$ é um minimizador para o problema (4.26), então teremos que

$$\begin{aligned} & (\nabla_i f(x) + \beta L_i B_i h_i(x))^T (h_i - h_i(x)) + (z_i - \psi_i(x_{(i)} + h_i(x))) \geq 0 \\ \Leftrightarrow & \left(\nabla_i f(x) + \beta L_i B_i (x_{(i)}^+ - x_{(i)}) \right)^T (x_{(i)} + h_i - x_{(i)}^+) + (z_i - \psi_i(x_{(i)}^+)) \geq 0, \end{aligned} \quad (4.27)$$

para todo $(x_{(i)} + h_i, z_i) \in \Omega_i$.

Apresentaremos um resultado com algumas propriedades semelhantes às presentes no Lema 2.1 de [44].

Proposição 4.5. *Seja x^* um ponto estacionário do problema (2.1). Suponhamos que a função ∇f seja localmente Lipschitz contínua numa vizinhança de x^* , isto é, existem $L, \bar{\epsilon} > 0$ tais que*

$$\|\nabla f(y) - \nabla f(x)\|_2 \leq L\|y - x\|_2, \quad \forall y, x \in \mathcal{B}(x^*, \bar{\epsilon}),$$

e ∇f satisfaça numa vizinhança de x^* a propriedade

$$(\nabla f(y) - \nabla f(x^*))^T (y - x^*) \geq \sigma \|y - x^*\|^2, \quad \text{para algum } \sigma > 0 \text{ e } \forall y \in \mathcal{B}(x^*, \bar{\epsilon}). \quad (4.28)$$

Então existem $\zeta, \delta > 0$ tais que

- (i) $\|y^+ - x^+\|_2 \leq \zeta \|y - x\|_2, \quad \forall y, x \in \mathcal{B}(x^*, \bar{\epsilon});$
- (ii) $\|x - x^*\|_2 \leq \delta \|h(x)\|_2, \quad \forall x \in \mathcal{B}(x^*, \bar{\epsilon}) \text{ e } \forall x \notin \mathcal{P}.$

Prova: Para mostrar o item (i), primeiramente faremos uso da expressão (4.27), para todo $i \in \{1, \dots, m\}$, para os vetores $x_{(i)}$ e $y_{(i)}$, obtendo que

$$\left(\nabla_i f(x) + \beta L_i B_i (x_{(i)}^+ - x_{(i)}) \right)^T (x_{(i)} + h_i - x_{(i)}^+) + (z_i - \psi_i(x_{(i)}^+)) \geq 0, \quad \forall (x_{(i)} + h_i, z_i) \in \Omega_i, \quad (4.29)$$

$$\left(\nabla_i f(y) + \beta L_i B_i (y_{(i)}^+ - y_{(i)}) \right)^T (y_{(i)} + h_i - y_{(i)}^+) + (z_i - \psi_i(y_{(i)}^+)) \geq 0, \quad \forall (y_{(i)} + h_i, z_i) \in \Omega_i.$$

Denotaremos $(h_i^1, z_i^1) = (y_{(i)}^+ - x_{(i)}, \psi_i(y_{(i)}^+))$ e $(h_i^2, z_i^2) = (x_{(i)}^+ - y_{(i)}, \psi_i(x_{(i)}^+))$. Como $(x_{(i)} + h_i^1, z_i^1) \in \Omega_i$ e $(y_{(i)} + h_i^2, z_i^2) \in \Omega_i$, podemos utilizá-los na primeira e segunda equações de (4.29), respectivamente, obtendo

$$\left(\nabla_i f(x) + \beta L_i B_i (x_{(i)}^+ - x_{(i)}) \right)^T (y_{(i)}^+ - x_{(i)}^+) + (\psi_i(y_{(i)}^+) - \psi_i(x_{(i)}^+)) \geq 0, \quad (4.30)$$

$$\left(\nabla_i f(y) + \beta L_i B_i (y_{(i)}^+ - y_{(i)}) \right)^T (x_{(i)}^+ - y_{(i)}^+) + (\psi_i(x_{(i)}^+) - \psi_i(y_{(i)}^+)) \geq 0.$$

Somando as expressões em (4.30), vemos que

$$\begin{aligned} & (\nabla_i f(x) - \nabla_i f(y))^T (y_{(i)}^+ - x_{(i)}^+) + \left(\beta L_i B_i (x_{(i)}^+ - y_{(i)}^+) + y_{(i)} - x_{(i)} \right)^T (y_{(i)}^+ - x_{(i)}^+) \geq 0 \\ \Leftrightarrow & (\nabla_i f(x) - \nabla_i f(y))^T (y_{(i)}^+ - x_{(i)}^+) + (\beta L_i B_i (y_{(i)} - x_{(i)}))^T (y_{(i)}^+ - x_{(i)}^+) \\ & \geq \beta L_i (y_{(i)}^+ - x_{(i)}^+)^T B_i (y_{(i)}^+ - x_{(i)}^+) \end{aligned} \quad (4.31)$$

Usando a Desigualdade de Cauchy-Schwarz, a Lipschitz continuidade do ∇f e o fato das matrizes B_i serem definidas positivas na expressão (4.31), temos para $x, y \in \mathcal{B}(x^*, \bar{\epsilon})$ que

$$\begin{aligned}
& \|\nabla_i f(x) - \nabla_i f(y)\|_2 \|y_{(i)}^+ - x_{(i)}^+\|_2 + \|\beta L_i B_i(y_{(i)} - x_{(i)})\|_2 \|y_{(i)}^+ - x_{(i)}^+\|_2 \\
& \geq \beta L_i (y_{(i)}^+ - x_{(i)}^+)^T B_i (y_{(i)}^+ - x_{(i)}^+) \\
\Leftrightarrow & \|\nabla f(x) - \nabla f(y)\|_2 \|y_{(i)}^+ - x_{(i)}^+\|_2 + \beta L_i \|B_i\|_2 \|y_{(i)} - x_{(i)}\|_2 \|y_{(i)}^+ - x_{(i)}^+\|_2 \\
& \geq \beta L_i \lambda_i^{\min} \|y_{(i)}^+ - x_{(i)}^+\|_2^2 \\
\Leftrightarrow & L \|x - y\|_2 \|y_{(i)}^+ - x_{(i)}^+\|_2 + \beta L_i \|B_i\|_2 \|y_{(i)} - x_{(i)}\|_2 \|y_{(i)}^+ - x_{(i)}^+\|_2 \\
& \geq \beta L_i \lambda_i^{\min} \|y_{(i)}^+ - x_{(i)}^+\|_2^2,
\end{aligned} \tag{4.32}$$

em que λ_i^{\min} denota o menor autovalor de B_i . Somando as m desigualdades (4.32) para todo i , tomando $c_1 = \min_{1 \leq i \leq m} \{\beta L_i \lambda_i^{\min}\}$ e $c_2 = \max_{1 \leq i \leq m} \{\beta L_i \|B_i\|_2\}$, vemos que

$$\begin{aligned}
& \sum_{i=1}^m \frac{L}{c_1} \|x - y\|_2 \|y_{(i)}^+ - x_{(i)}^+\|_2 + \sum_{i=1}^m \frac{c_2}{c_1} \|y_{(i)} - x_{(i)}\|_2 \|y_{(i)}^+ - x_{(i)}^+\|_2 \\
& \geq \sum_{i=1}^m \|y_{(i)}^+ - x_{(i)}^+\|_2^2 \\
\Leftrightarrow & \frac{mL}{c_1} \|x - y\|_2 \|y^+ - x^+\|_2 + \frac{mc_2}{c_1} \|y - x\|_2 \|y^+ - x^+\|_2 \geq \|y^+ - x^+\|_2^2.
\end{aligned} \tag{4.33}$$

Dividindo os dois membros de (4.33) por $\|y^+ - x^+\|_2$, com $y^+ \neq x^+$, vemos que a expressão obtida é igual à do item (i) para $\zeta = \frac{m(L + c_2)}{c_1}$.

Para mostrar o item (ii), notamos que $(x_{(i)}^+, \psi_i(x_{(i)}^+)) \in \Omega_i$ e $(x^*, \psi(x^*)) \in \Omega$. Logo, usando $(x^+, \psi(x^+))$ e $(h_i(x), z_i) = (x_{(i)}^* - x_{(i)}, \psi_i(x_{(i)}^*))$ nas expressões (4.25) e (4.27), respectivamente, obtemos que

$$\nabla f(x^*)^T (x^+ - x^*) + (\psi(x^+) - \psi(x^*)) \geq 0, \tag{4.34}$$

e

$$\left(\nabla_i f(x) + \beta L_i B_i(x_{(i)}^+ - x_{(i)}) \right)^T (x_{(i)}^* - x_{(i)}^+) + (\psi_i(x_{(i)}^*) - \psi_i(x_{(i)}^+)) \geq 0, \forall i \in \{1, \dots, m\}. \tag{4.35}$$

Somando as m expressões em (4.35), usando $x^+ = x + h(x)$ e $x_{(i)}^+ = x_{(i)} + h_i(x)$, temos que

$$\begin{aligned}
& \sum_{i=1}^m \left(\nabla_i f(x) + \beta L_i B_i(x_{(i)}^+ - x_{(i)}) \right)^T (x_{(i)}^* - x_{(i)}^+) + \sum_{i=1}^m (\psi_i(x_{(i)}^*) - \psi_i(x_{(i)}^+)) \geq 0 \\
\Leftrightarrow & \nabla f(x)^T (x^* - x^+) + \sum_{i=1}^m \left(\beta L_i B_i(x_{(i)}^+ - x_{(i)}) \right)^T (x_{(i)}^* - x_{(i)}^+) + \\
& + (\psi(x^*) - \psi(x^+)) \geq 0.
\end{aligned} \tag{4.36}$$

Somando as expressões (4.34) e (4.36), vemos que

$$\begin{aligned}
& (\nabla f(x^*) - \nabla f(x))^T (x^+ - x^*) - \sum_{i=1}^m \left(\beta L_i B_i (x_{(i)}^+ - x_{(i)}) \right)^T (x_{(i)}^+ - x_{(i)}^*) \geq 0 \\
\Leftrightarrow & (\nabla f(x^*) - \nabla f(x))^T (x - x^*) + (\nabla f(x^*) - \nabla f(x))^T h(x) - \\
& - \sum_{i=1}^m (\beta L_i B_i h_i(x))^T (x_{(i)}^+ - x_{(i)}^*) \geq 0 \\
\Leftrightarrow & (\nabla f(x^*) - \nabla f(x))^T (x - x^*) + (\nabla f(x^*) - \nabla f(x))^T h(x) + \\
& + \sum_{i=1}^m (\beta L_i B_i h_i(x))^T (x_{(i)}^* - x_{(i)}) - \sum_{i=1}^m (\beta L_i B_i h_i(x))^T h_i(x) \geq 0 \\
\Leftrightarrow & (\nabla f(x^*) - \nabla f(x))^T h(x) + \\
& + \sum_{i=1}^m (\beta L_i B_i h_i(x))^T (x_{(i)}^* - x_{(i)}) \geq (\nabla f(x) - \nabla f(x^*))^T (x - x^*) \quad (4.37)
\end{aligned}$$

Usando na expressão (4.37), a Desigualdade de Cauchy-Schwarz, a Lipschitz continuidade do ∇f , a expressão (4.28) e a consistência das normas definidas por B_i , vemos que

$$\begin{aligned}
& L \|x^* - x\|_2 \|h(x)\|_2 + \\
& + \sum_{i=1}^m \beta L_i \|B_i\|_2 \|h_i(x)\|_2 \|x_{(i)}^* - x_{(i)}\|_2 \geq \sigma \|x - x^*\|_2^2 \quad (4.38)
\end{aligned}$$

Tomando $c_3 = \max_{1 \leq i \leq m} \{\beta L_i \|B_i\|_2\}$, temos, pela expressão (4.38), que

$$\begin{aligned}
& L \|x^* - x\|_2 \|h(x)\|_2 + \sum_{i=1}^m \beta L_i \|B_i\|_2 \|h_i(x)\|_2 \|x_{(i)}^* - x_{(i)}\|_2 \geq \sigma \|x - x^*\|_2^2 \\
\Leftrightarrow & L \|x^* - x\|_2 \|h(x)\|_2 + m c_3 \|h(x)\|_2 \|x^* - x\|_2 \geq \sigma \|x - x^*\|_2^2. \quad (4.39)
\end{aligned}$$

Pela expressão (4.39), temos a validade do item (ii), para todo $x \in \mathcal{B}(x^*, \bar{\epsilon})$. Dividindo os dois membros da expressão (4.39) por $\|x - x^*\|_2$ com $x \neq x^*$, $\delta = \frac{L + m c_3}{\sigma}$, concluímos a demonstração. \square

Observação 4.1. Notamos que a hipótese (4.28) é razoável de ser pedida, pois podemos exibir algumas situações em que a função f apresenta o comportamento exigido em (4.28).

O primeiro caso é obtido pedindo que $f \in C^2(\mathbb{R}^n)$ e $\nabla^2 f(x) \in \mathbb{S}_{++}^n$, para todo $x \in \mathcal{B}(x^*, \epsilon)$. Nesse caso, considerando a função $\phi : \mathbb{R} \rightarrow \mathbb{R}$ definida por

$$\phi(t) = \nabla f(x^* + t(y - x^*))^T (y - x^*),$$

obtemos pelo Teorema Fundamental do Cálculo que

$$\phi(1) - \phi(0) = \int_0^1 \phi'(t) dt, \quad (4.40)$$

e usando a regra da cadeia em (4.40), vemos que

$$\begin{aligned}
\nabla f(y)^T (y - x^*) - \nabla f(x^*)^T (y - x^*) &= \int_0^1 (y - x^*)^T \nabla^2 f(x^* + t(y - x^*)) (y - x^*) dt \\
&\geq \int_0^1 \lambda_{\min}^* (y - x^*)^T (y - x^*) dt \\
&= \lambda_{\min}^* \|y - x^*\|_2^2, \quad (4.41)
\end{aligned}$$

em que λ_{\min}^* é uma cota inferior para o menor autovalor de todas as matrizes $\nabla^2 f(x)$, para todo $x \in \mathcal{B}(x^*, \epsilon)$. Assim (4.41) é comparável à expressão (4.28).

Outra situação em que a expressão (4.28) ocorre é quando f é uma função fortemente convexa, isto é, existe $\delta > 0$ tal que

$$f(y) - f(x) - \nabla f(x)^T(y - x) \geq \frac{\delta}{2} \|y - x\|_2^2, \quad \forall y, x \in \mathbb{R}^n. \quad (4.42)$$

Para verificar a afirmação anterior, considere a expressão (4.42) e uma segunda expressão substituindo y por x na expressão (4.42). Depois, some as duas expressões obtidas, chegando ao desejado

$$(\nabla f(y) - \nabla f(x))^T(y - x) \geq \delta \|y - x\|_2^2, \quad \forall y, x \in \mathbb{R}^n.$$

Mostraremos que, caso o vetor $x^* \in \mathcal{P}$ satisfaça as hipóteses da Proposição 4.5, é possível garantir que a função ρ_1 dada por (4.22) é uma função identificadora para $x^* \in \mathcal{P}$.

Proposição 4.6. *Se $x^* \in \mathcal{P}$ satisfaz as hipóteses da Proposição 4.5, então, ρ_1^α é uma função identificadora para x^* , para qualquer $0 < \alpha < 1$ fixo.*

Prova: A Proposição 2.1 garante a validade dos itens (ii) e (iii) da Definição 4.7. Com a Proposição 4.5 e a desigualdade $|\|h(x)\|_2 - \|h(y)\|_2| \leq \|h(x) - h(y)\|_2$, existe $\epsilon > 0$ tal que para todo $x, y \in \mathcal{B}(x^*, \epsilon)$

$$\begin{aligned} |\|h(x)\|_2 - \|h(y)\|_2| &\leq \|h(x) - h(y)\|_2 \\ &= \|x + h(x) - y - h(y) + y - x\|_2 \\ &\leq \|x + h(x) - y - h(y)\|_2 + \|y - x\|_2 \\ &\leq \zeta \|x - y\|_2 + \|y - x\|_2 \\ &= (1 + \zeta) \|x - y\|_2. \end{aligned}$$

Usando a expressão anterior, juntamente com o fato que a função $j(t) = -t^\alpha$ é localmente Lipschitz contínua, obtemos que ρ_1^α é uma função contínua numa vizinhança de x^* , mostrando assim o item (i) da Definição 4.7.

Para mostrar o item (iv) da Definição 4.7, usaremos o item (ii) da Proposição 4.5, tomando uma sequência $x^k \rightarrow x^*$ com $x^k \notin \mathcal{P}$ obtendo, para k suficientemente grande, que

$$\frac{1}{\delta \|h(x^k)\|_2} \leq \frac{1}{\|x^k - x^*\|_2} \Leftrightarrow \frac{1}{\delta \|h(x^k)\|_2^{1-\alpha}} \leq \frac{\|h(x^k)\|_2^\alpha}{\|x^k - x^*\|_2}.$$

Passando ao limite a última desigualdade da expressão anterior, vemos que

$$+\infty = \lim_{x^k \rightarrow x^*} \frac{1}{\delta \|h(x^k)\|_2^{1-\alpha}} \leq \lim_{x^k \rightarrow x^*} \frac{-\rho_1^\alpha(x^k)}{\|x^k - x^*\|_2},$$

mostrando assim o item (iii), e concluindo a demonstração. \square

5 Testes computacionais

Neste capítulo apresentaremos alguns testes feitos em FORTRAN 90 e MATLAB. As primeiras implementações feitas foram voltadas para os problemas com a função não suave $\psi(x) = \lambda\|x\|_1$, que foi escolhida por alguns motivos. O primeiro se deve pela versatilidade na escolha dos blocos fornecida por essa função quando aplicada ao nosso subproblema dado por (2.6) e pela possibilidade de resolução exata dos subproblemas para escolhas específicas das matrizes B_i . O segundo vem dos trabalhos encontrados envolvendo a resolução de problemas com tal estrutura, por exemplo nas áreas de *matrix completion* em [13], *compressed sensing* em [18], da Biologia em [25, 30], *machine learning* [29], Estatística em [47, 48], *truss topology design* [35] e *group Lasso* [49], para citar alguns exemplos.

Os problemas testados nesse capítulo são da forma

$$\min F(x), \quad (5.1)$$

a caixa que será explorada usando a teoria de identificação das restrições ativas desenvolvida nesse texto, aparecerá por meio de uma reformulação equivalente dos problemas resolvidos.

Vamos apresentar alguns exemplos de funções suaves testadas nesse trabalho e o cálculo das constantes de Lipschitz para o gradiente dessas funções por blocos, necessário para aplicação do método.

5.1 Problema 1: LASSO

O problema conhecido como *LASSO* [42] envolve a resolução de um problema de minimização com a função objetivo

$$F(x) = \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1. \quad (5.2)$$

Como descrito no Seção 2, para o funcionamento do nosso algoritmo precisamos calcular as constantes de Lipschitz por blocos referentes ao gradiente do termo suave da função (5.2), que denotaremos por f . Supondo que o i -ésimo bloco seja escolhido com tamanho p e contenha as coordenadas $\{i_1, i_2, \dots, i_p\}$, então podemos mostrar que a constante de Lipschitz associada ao i -ésimo bloco é dada por $L_i = \|a_{*(i)}^T a_{*(i)}\|_2$. De fato,

$$\begin{aligned} \|\nabla_i f(x + U_i t_i) - \nabla_i f(x)\|_2 &= \|\nabla_i f(x + U_i t_i) - \nabla_i f(x)\|_2 \\ &= \|a_{*(i)}^T [A(x + U_i t_i) - b] - a_{*(i)}^T (Ax - b)\|_2 \\ &= \|a_{*(i)}^T A U_i t_i\|_2 \\ &= \|a_{*(i)}^T a_{*(i)} t_i\|_2 \\ &\leq \|a_{*(i)}^T a_{*(i)}\|_2 \|t_i\|_2, \end{aligned} \quad (5.3)$$

em que $a_{*(i)} = [a_{*i_1}, \dots, a_{*i_p}]$ é a submatriz de A formada pelas colunas $\{i_1, i_2, \dots, i_p\}$, e as matrizes U_i são as mesmas definidas no Seção 2.1. Ressaltamos que caso o i -ésimo bloco tenha tamanho 1, a desigualdade de (5.3) na verdade é uma igualdade, conforme descrito no Exemplo 2.3 desse texto.

5.2 Problema 2: Regressão logística com regularização ℓ_1

O problema da regressão logística com regularização ℓ_1 [33] envolve a resolução de um problema de minimização com a função objetivo

$$F(x) = \sum_{i=1}^s \log(1 + \exp(-b_i a_{i*} x)) + \lambda \|x\|_1, \quad (5.4)$$

com $b_i \in \{-1, +1\}$ e $\lambda > 0$.

A função conhecida como perda logística, denotada por $f_{pl}(z) = \log(1 + \exp(-z))$, é um dos exemplos de função utilizada no contexto de classificação binária linear. Em classificação binária linear, para um conjunto de dados representados pelos vetores a_{i*} , $i = 1, \dots, s$, cada um associado a um rótulo $\{-1, +1\}$, o objetivo é encontrar um vetor x que classifique da melhor maneira possível os vetores a_{i*} , isto é,

$$\begin{cases} b_i = 1, & \text{se } a_{i*} x \geq 0; \\ b_i = -1, & \text{se } a_{i*} x < 0. \end{cases}$$

Evidentemente, quando o conjunto de dados não admite um classificador linear, a relação anterior não será satisfeita para todos os vetores do conjunto.

A ideia estatística por trás da parte suave do problema (5.4) é tomar a probabilidade condicional a posteriori da variável b_i , dados os vetores x, a_{i*} , como

$$\mathbb{P}(b_i | x, a_{i*}) = \frac{1}{1 + \exp(-b_i a_{i*} x)}.$$

A partir das probabilidades anteriores, pode-se obter a parte suave do problema (5.4) pelo fato que o hiperplano de normal x que melhor classifica os vetores a_{i*} , segundo as probabilidades condicionais, será aquele que maximizar as probabilidades das variáveis b_i , $i = 1, \dots, s$, estarem corretas, e isto pode ser obtido, minimizando o negativo do logaritmo da verossimilhança das probabilidades de cada um dos eventos i , supondo que os eventos são independentes. Destacamos algumas referências de métodos de otimização que lidam com o problema (5.4): um método de descenso coordenado [9] e uma implementação de pontos interiores [26]. O segundo texto, em especial, explora também um pouco do contexto estatístico e das condições de otimalidade do problema.

Vamos descrever quais são as constantes de Lipschitz para o gradiente do termo suave de (5.4), que denotaremos por f , referente ao bloco de coordenadas i contendo p coordenadas descritas por $\{i_1, i_2, \dots, i_p\}$. Podemos mostrar que a j -ésima coordenada do gradiente da função f é dada por

$$\nabla_j f(x) = - \sum_{i=1}^s b_i a_{ij} \frac{\exp(-b_i a_{i*} x)}{(1 + \exp(-b_i a_{i*} x))}, \quad (5.5)$$

em que $\{a_{i*} \in \mathbb{R}^n, i = 1, \dots, s\}$ é um conjunto de s vetores pertencentes ao \mathbb{R}^n .

Pela Desigualdade do Valor Médio sabemos que, fixado o bloco de coordenadas i , temos que

$$\|\nabla_i f(x + U_i t_i) - \nabla_i f(x)\|_2 \leq \|\nabla_{ii}^2 f(x + c)\|_2 \|t_i\|_2 \leq \sup_{x \in \mathbb{R}^n} \|\nabla_{ii}^2 f(x)\|_2 \|t_i\|_2, \quad (5.6)$$

para $x + c \in (x + t_i, x)$.

Calculando a hessiana com relação ao bloco de coordenadas i , vemos que

$$\nabla_{ii}^2 f(x) = \begin{pmatrix} \sum_{j=1}^s b_j a_{ji_1} b_j a_{ji_1} \frac{\exp(-b_j a_{j*} x)}{(1 + \exp(-b_j a_{j*} x))^2} & \cdots & \sum_{j=1}^s b_j a_{ji_1} b_j a_{ji_p} \frac{\exp(-b_j a_{j*} x)}{(1 + \exp(-b_j a_{j*} x))^2} \\ \sum_{j=1}^s b_j a_{ji_2} b_j a_{ji_1} \frac{\exp(-b_j a_{j*} x)}{(1 + \exp(-b_j a_{j*} x))^2} & \cdots & \sum_{j=1}^s b_j a_{ji_2} b_j a_{ji_p} \frac{\exp(-b_j a_{j*} x)}{(1 + \exp(-b_j a_{j*} x))^2} \\ \vdots & & \vdots \\ \sum_{j=1}^s b_j a_{ji_p} b_j a_{ji_1} \frac{\exp(-b_j a_{j*} x)}{(1 + \exp(-b_j a_{j*} x))^2} & \cdots & \sum_{j=1}^s b_j a_{ji_p} b_j a_{ji_p} \frac{\exp(-b_j a_{j*} x)}{(1 + \exp(-b_j a_{j*} x))^2} \end{pmatrix} \quad (5.7)$$

Usando a teoria de minimização com restrições, conseguimos mostrar que o problema

$$\begin{aligned} \max \quad & \frac{z}{(1+z)^2} \\ \text{s.a.} \quad & z \geq 0 \end{aligned}$$

assume valor máximo para $z = 1$, concluindo assim que

$$\frac{\exp(-b_j a_{j*} x)}{(1 + \exp(-b_j a_{j*} x))^2} \leq \frac{1}{4}, \forall j = 1, \dots, m. \quad (5.8)$$

Substituindo (5.8) em (5.7) obtemos que

$$L_i = \sup_{x \in \mathbb{R}^n} \|\nabla_{ii}^2 f(x)\|_2 = \frac{1}{4} \|S_i\|_2,$$

em que S_i é dada por

$$S_i = \begin{pmatrix} \sum_{j=1}^s b_j a_{ji_1} b_j a_{ji_1} & \cdots & \sum_{j=1}^s b_j a_{ji_1} b_j a_{ji_p} \\ \sum_{j=1}^s b_j a_{ji_2} b_j a_{ji_1} & \cdots & \sum_{j=1}^s b_j a_{ji_2} b_j a_{ji_p} \\ \vdots & & \vdots \\ \sum_{j=1}^s b_j a_{ji_p} b_j a_{ji_1} & \cdots & \sum_{j=1}^s b_j a_{ji_p} b_j a_{ji_p} \end{pmatrix}.$$

5.3 Testes em MATLAB

Nossos experimentos numéricos foram realizados em um notebook DELL Latitude E6440 Intel(R) Core(TM) i7-4610M CPU @ 3.00GHz, usando a versão do MATLAB: 9.0.0.341360

(R2016a) e focam nos dois problemas abordados anteriormente, *LASSO* e regressão logística com regularização ℓ_1 .

Esses problemas são relevantes e bem adaptados para serem resolvidos usando métodos de descenso coordenado por blocos por vários motivos. Primeiro, a estrutura da função $\psi(x) = \lambda\|x\|_1$, com $\lambda > 0$, permite a escolha de qualquer separação por blocos das coordenadas do problema. Segundo, os subproblemas (2.6) possuem fórmulas fechadas quando escolhemos as matrizes B_i , $i = 1, \dots, m$ como sendo diagonais, como descrito no Exemplo 2.3. Terceiro, as constantes de Lipschitz para o gradiente da função f por blocos podem ser calculadas facilmente e com pouco esforço computacional, se os blocos têm poucas coordenadas (cf. Seções 5.1 e 5.2). Por fim, o gradiente da parte suave pode ser atualizado com menos esforço computacional quando um único bloco de coordenadas é modificado, comparado à atualização do gradiente quando modificamos todas as coordenadas do vetor. Para mais detalhes, veja por exemplo [36].

Discutiremos nas próximas subseções possibilidades de escolha para o conjunto \mathcal{J} no algoritmo *Active BCDM* que estimam as restrições ativas do problema.

5.3.1 Escolha do Conjunto \mathcal{J}

Uma escolha natural para o conjunto \mathcal{J} , pelo desenvolvimento desse texto, é usar a Definição 4.8, o Teorema 4.1, a Proposição 4.6, e definir

$$\mathcal{J} \equiv \hat{\mathcal{A}}(x). \quad (5.9)$$

Pelo Teorema 4.1 e pela Proposição 4.6, vemos que a função identificadora (4.22) poderia ser usada para construir o conjunto $\hat{\mathcal{A}}(x)$, e portanto, numa vizinhança de um ponto estacionário satisfazendo as hipóteses dessas proposições, o conjunto \mathcal{J} seria igual ao conjunto das restrições ativas no ponto x^* . Essa variante do algoritmo *Active BCDM* será chamada **BCDM+IF**, com IF escrito em alusão ao termo função identificadora escrito em inglês como *Identification Function*.

Para testar a capacidade dessa estratégia de identificação das restrições ativas, precisamos escolher problemas que preencham todas as hipóteses da Proposição 4.5. Portanto, restringiremos nossa atenção, nessa subseção, a problemas do tipo

$$\begin{aligned} \min_x \quad & \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1 \\ \text{s.a.} \quad & x \geq 0 \end{aligned} \quad (5.10)$$

com $\lambda = 0.1\|A^T b\|_\infty$, $A \in \mathbb{R}^{s \times n}$ e $s > n$. O motivo para a escolha desse valor de λ , vem do fato que ele é uma fração do valor que garante a solução nula para o problema *LASSO* (5.2), pois,

pelas condições de otimalidade desse problema, sabemos que

$$\begin{aligned}
& \mathbf{0} \text{ é solução do problema (5.2)} \\
\Leftrightarrow & -\nabla f(\mathbf{0}) \in \partial(\lambda\|x\|_1) \\
\Leftrightarrow & -\nabla f(\mathbf{0}) \in [-\lambda, \lambda]^n \\
\Leftrightarrow & -A^T(A\mathbf{0} - b) \in [-\lambda, \lambda]^n \\
\Leftrightarrow & A^T b \in [-\lambda, \lambda]^n \\
\Leftrightarrow & \max_{i=1,\dots,n} (A^T b)_i \leq \lambda \text{ e } \min_{i=1,\dots,n} (A^T b)_i \geq -\lambda \\
\Leftrightarrow & \max_{i=1,\dots,n} (A^T b)_i \leq \lambda \text{ e } \max_{i=1,\dots,n} (-A^T b)_i \leq \lambda \\
\Leftrightarrow & \max_{i=1,\dots,n} |(A^T b)_i| = \|A^T b\|_\infty \leq \lambda.
\end{aligned} \tag{5.11}$$

Analisando algumas características do Algoritmo 1, usaremos uma segunda escolha para o conjunto \mathcal{J} dada por

$$\mathcal{J} = \{i \mid x_i = 0 \text{ e } h_i(x) = 0\} \subset \hat{\mathcal{A}}(x). \tag{5.12}$$

A inclusão anterior é válida porque qualquer função identificadora classificará como ativa uma restrição que é satisfeita como igualdade no ponto corrente. Logo, o conjunto \mathcal{J} em (5.12) tenta capturar duas informações relevantes: a informação $x_i = 0$, um subconjunto das coordenadas descritas como ativas por qualquer função identificadora, e a informação $h_i = 0$, que tenta evitar que durante o método fixemos uma coordenada como ativa, sendo que o método diz que podemos decrescer o valor da função objetivo nessa direção. Essa última escolha é especialmente importante em nosso contexto, pois começamos o método do vetor nulo, um ponto com todas as coordenadas ativas ao problema (5.10). Essa estratégia está associada com as ideias que já foram usadas em experimentos numéricos [36, Seção 6.1.7]. Essa variante do algoritmo *Active BCDM* será denotada por BCDM+ST, com ST em alusão à estacionariedade do método escrito em inglês como *Stationarity*, desde que a informação $h_i(x) = 0$ está relacionada com a estacionariedade do ponto x , veja Proposição 2.1.

Para analisar o desempenho das variantes do algoritmo *Active BCDM*, selecionamos dados que foram usados nos artigos [24, 40]. Escolhemos aqueles dados com $s > n$, e para os quais (A, b) foram obtidos de maneira determinística, totalizando 12 problemas. Os testes eram originalmente aplicados a problemas de quadrados mínimos com restrições não negativas, isto é,

$$\begin{aligned}
& \min_x \quad \frac{1}{2} \|Ax - b\|_2^2 \\
& \text{s.a.} \quad x \geq 0.
\end{aligned} \tag{5.13}$$

A fim de obter um problema da forma (5.10), adicionamos o termo extra $\lambda\|x\|_1$. A Tabela 1 apresenta a lista dos problemas com respeito ao nome, dimensão, fonte original e outras informações adicionais, descritas mais adiante.

Uma principal dificuldade quando tentamos comparar as variantes dos métodos de descenso coordenado, e outros métodos de primeira ordem, é com respeito ao critério de

parada. Nesse trabalho, usaremos como critério de parada um valor de função alvo que está acima do valor de função mínimo do problema, semelhante ao feito em [38]. Para definir o alvo usamos o seguinte procedimento: calculamos uma solução com alta precisão para cada problema rodando um método padrão, que gera uma sequência de pontos cujo valor de função forma uma sequência monótona não-crescente, como o *UCDC*, descrito na Observação 2.6 da Seção 2.1, para um número de iterações suficientemente grande. Para o menor valor de função obtido por tal procedimento, arredondamos até o quarto dígito significativo e adicionamos uma unidade ao quarto dígito.

O procedimento descrito cria um limitante superior para o valor ótimo de cada problema, que está correto até o terceiro dígito, e com erro em uma unidade no quarto dígito. Portanto, qualquer método que esteja apto a calcular um ponto factível com valor de função menor do que esse limitante, resolverá o problema com erro relativo na função objetivo menor do que 10^{-4} . Essa é uma meta razoável para um método de primeira ordem, que sofrerá muito para atingir uma solução com alta precisão em problemas do mundo real. O alvo para valor de função usado em cada problema é apresentado na Tabela 1, na penúltima coluna, rotulada como F^* . Adotaremos esse critério para construir o valor de função alvo em todos os nossos experimentos numéricos. Na última coluna, descrevemos a porcentagem de coordenadas nulas da solução encontrada pelo método usado para obter o alvo F^* , esse valor é denotado na tabela por $\text{nz}(x^*)$.

Tabela 1 – Dados testados no problema LASSO com restrições não negativas

Rótulo	Nome	(s, n)	F^*	$\text{nz}(x^*)$
NN_1	[7] illc1033	(1033, 320)	1.098×10^7	88.12%
NN_2	[7] illc1850	(1850, 712)	1.771×10^7	94.52%
NN_3	[7] well1033	(1033, 320)	7.093×10^6	93.75%
NN_4	[7] well1850	(1850, 712)	8.295×10^6	96.91%
NN_5	[14] real-sim	(72309, 20958)	3.525×10^4	99.93%
NN_6	[14] mnist	(60000, 717)	3.241×10^5	95.53%
NN_7	[14] webspam-unigram	(350000, 138)	3.241×10^5	98.55%
NN_8	[16] Maragal-3	(1690, 858)	1.009×10^1	99.06%
NN_9	[16] Maragal-4	(1964, 1027)	1.865×10^1	99.22%
NN_{10}	[16] Maragal-5	(4654, 3296)	4.381×10^1	99.72%
NN_{11}	[16] Maragal-6	(21255, 10144)	5.968×10^1	99.90%
NN_{12}	[16] Maragal-7	(46845, 26525)	1.414×10^2	99.95%

Além disso, para completamente definir BCDM+IF e BCDM+ST é necessário selecionar valores para os parâmetros δ_{DP} e δ_F que são usados no Algoritmo 1. Nós fizemos isso testando 24 combinações desses parâmetros, quatro valores para $\delta_F \in \{[0.001n], [0.01n], [0.1n], n\}$ e seis valores para $\delta_{DP} \in \{2, 5, 10, 10^2, 10^3, 10^4\}$. Apresentaremos a Tabela 2 contendo os respectivos valores usados para δ_{DP} e δ_F baseados na terminação da variante de cada método.

Selecionamos a melhor combinação de parâmetros para cada método olhando para o *performance profiles* [17] do tempo total de *CPU* obtido após 10 rodadas de ambos os métodos nos 12 problemas teste. Os resultados mais relevantes encontram-se nas Figuras 1 e 2. Pelas figuras

Tabela 2 – Escolhas usadas em todos os algoritmos do texto para calibragem dos parâmetros δ_{DP} e δ_F

Final da Variante	δ_{DP}	δ_F
1	2	$[0.001n]$
2	2	$[0.01n]$
3	2	$[0.1n]$
4	2	n
5	5	$[0.001n]$
6	5	$[0.01n]$
7	5	$[0.1n]$
8	5	n
9	10	$[0.001n]$
10	10	$[0.01n]$
11	10	$[0.1n]$
12	10	n
13	100	$[0.001n]$
14	100	$[0.01n]$
15	100	$[0.1n]$
16	100	n
17	1000	$[0.001n]$
18	1000	$[0.01n]$
19	1000	$[0.1n]$
20	1000	n
21	10000	$[0.001n]$
22	10000	$[0.01n]$
23	10000	$[0.1n]$
24	10000	n

mencionadas anteriormente, vemos que as melhores combinações foram $\delta_{DP} = 10$ e $\delta_F = [0.1n]$ para BCDM+IF e $\delta_{DP} = 10^4$ e $\delta_F = [0.01n]$ para BCDM+ST.

Na sequência, comparamos as duas escolhas para \mathcal{J} usando novamente um *performance profile* confrontando as médias dos tempo de *CPU* das 10 rodadas entre BCDM+IF e BCDM+ST (Figura 3). Está claro por essa figura que a melhor escolha para o conjunto \mathcal{J} é aquela usada pelo método BCDM+ST. De agora em diante, usaremos BCDM+ST como implementação padrão do Algoritmo 1 e a chamaremos simplesmente de **ActiveBCDM**.

Finalmente, comparamos o método **ActiveBCDM** com o método BCDM original, que inspirou o método de descenso coordenado desse texto, sem nenhuma identificação. Isso é equivalente a considerar $\mathcal{J} = \emptyset$ em todas as iterações. Chamamos essa variante de **UBCDM**, pois ela usa distribuição de probabilidade uniforme para selecionar os blocos. Equivalentemente às variantes anteriores, precisamos ajustar o parâmetro δ_F para obter uma variante o mais qualificada possível. Testamos os mesmos quatro valores usados anteriormente $\delta_F \in \{[0.001n], [0.01n], [0.1n], n\}$ e comparamos seus respectivos *performance profiles* usando o tempo de *CPU* obtido por meio de uma média de 10 rodadas nos 12 problemas da Tabela 1, os resultados estão na Figura 4.

Figura 1 – *Performance profiles* mais significativos entre as 24 variantes do método BCDM+IF para os problemas da Tabela 1

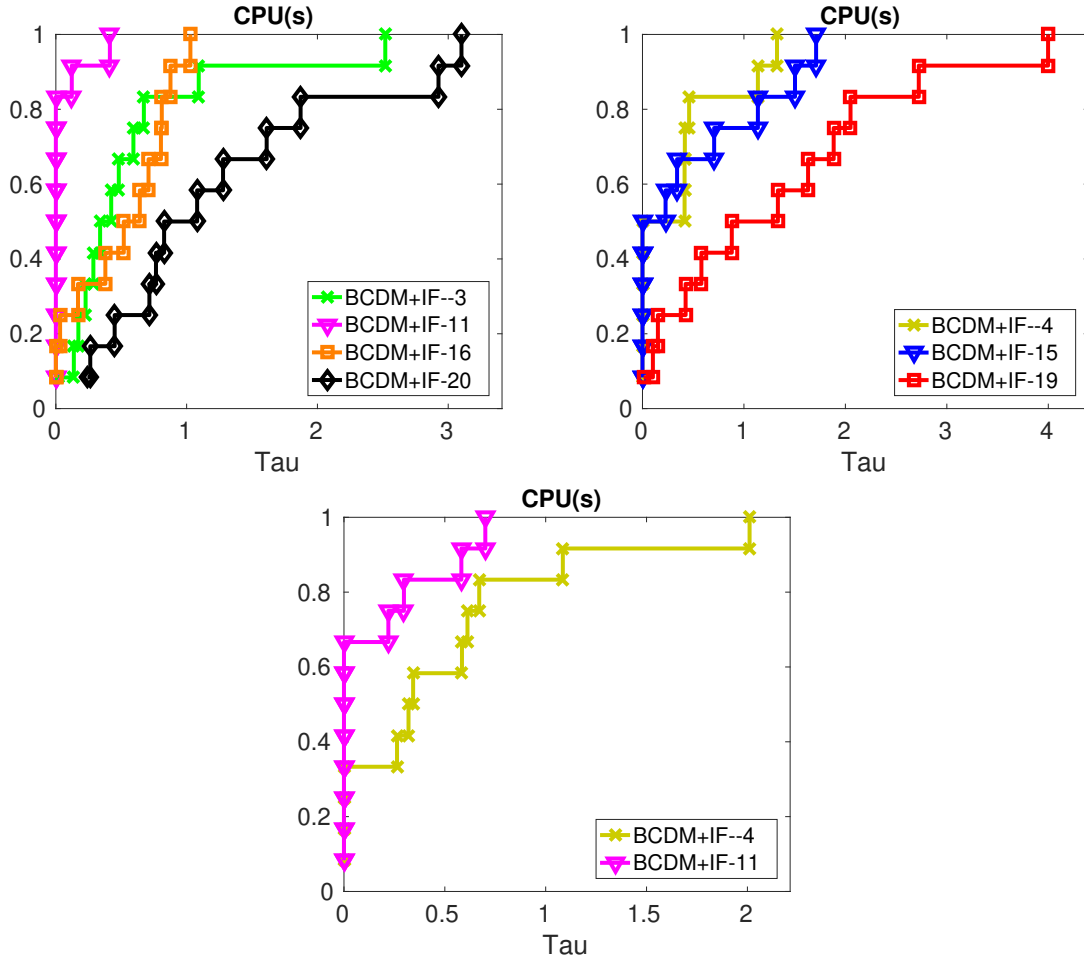
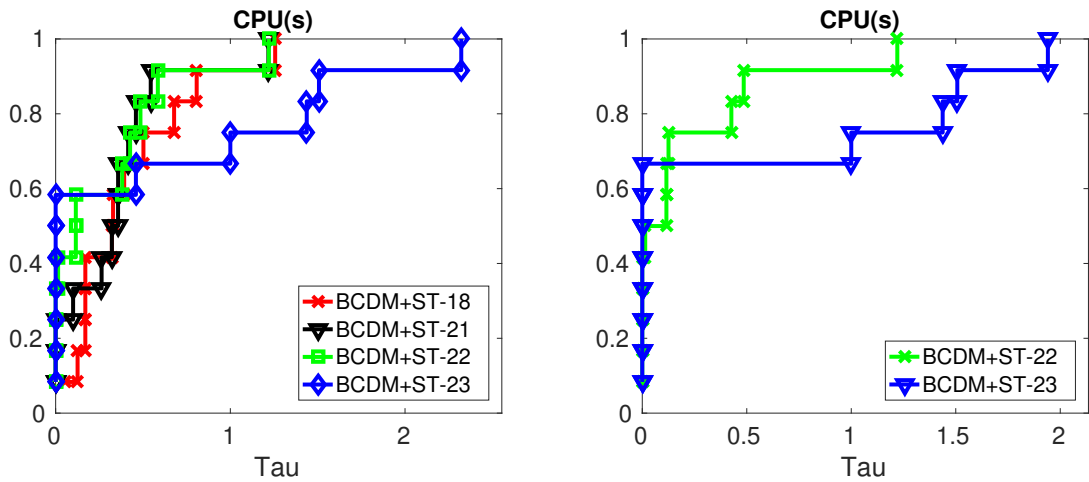
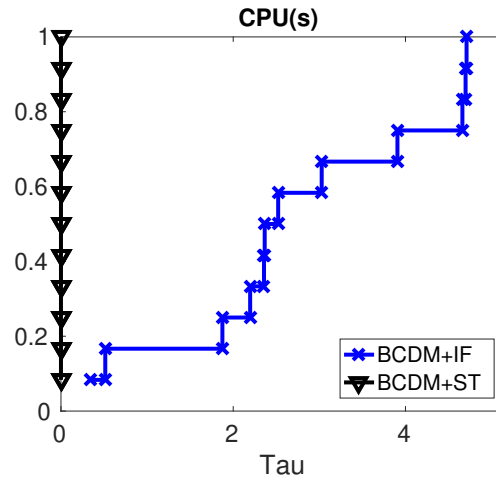
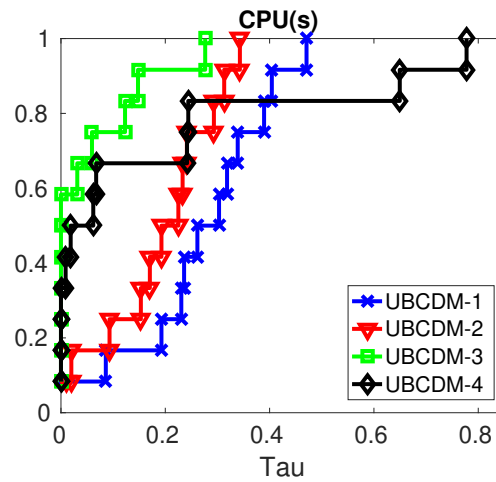
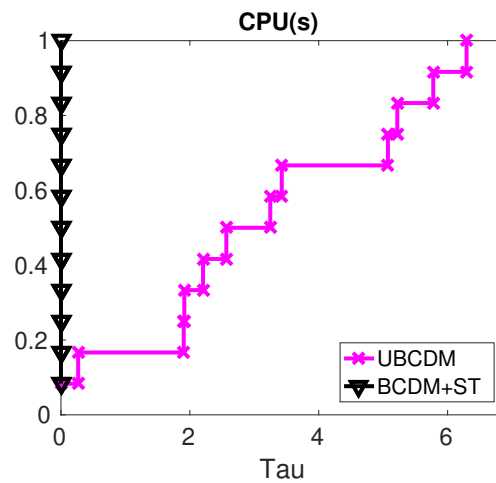


Figura 2 – *Performance profiles* mais significativos entre as 24 variantes do método BCDM+ST para os problemas da Tabela 1



Pela Figura 4, vemos que a melhor variante ajustada foi $\delta_F = [0.1n]$. Comparamos as melhores variantes ajustadas entre si, UBCDM com ActiveBCDM (Figura 5). Mais uma vez, o método que tenta identificar as restrições ativas é claramente o mais rápido.

Figura 3 – *Performance profile* entre BCDM+IF e BCDM+ST para os problemas da Tabela 1Figura 4 – *Performance profiles* mais significativos entre as 24 variantes do método UBCDM para os problemas da Tabela 1Figura 5 – *Performance profile* entre UBCDM e ActiveBCDM para os problemas da Tabela 1

Uma importante observação para a escolha de \mathcal{J} definida por (5.12) é que ela abre a possibilidade de aplicar as ideias de identificação para problemas irrestritos quando a função

$\psi(x)$ é a regularização ℓ_1 . De fato, o problema

$$\min_x f(x) + \lambda \|x\|_1 \quad (5.14)$$

pode ser reformulado como um problema com restrições

$$\begin{aligned} \min_{x_+, x_-} \quad & f(x_+ - x_-) + \lambda e^T (x_+ + x_-) \\ \text{s.a.} \quad & x_+ \geq 0 \\ & x_- \geq 0, \end{aligned} \quad (5.15)$$

onde $e = (1, \dots, 1)^T$. Em particular, cada solução \bar{x} do problema (5.14) pode ser reescrita em termos da solução (\bar{x}_+, \bar{x}_-) de (5.15), usando $\bar{x} = \bar{x}_+ - \bar{x}_-$.

Seja x^* uma solução de (5.14). Uma coordenada nula $x_i^* = 0$ está associada a duas restrições ativas $(x_+^*)_i = 0$ e $(x_-^*)_i = 0$ de (5.15). Então, podemos usar a estimativa das restrições ativas \mathcal{J} dada por (5.12) implicitamente para (5.15) tentando identificar tais coordenadas e acelerar a convergência, mesmo para problemas irrestritos como em (5.14). Nas próximas subseções, aplicaremos o método **ActiveBCDM** buscando explorar essas ideias.

5.3.2 LASSO

5.3.2.1 Conjunto de dados e critério de parada

Agora que escolhemos a melhor estratégia para definir o conjunto \mathcal{J} no algoritmo *Active BCDM* para resolver problemas do tipo (5.10), podemos voltar nossa atenção para o problema *LASSO* (5.2).

Com o objetivo de testar nossas ideias, selecionamos um conjunto de 49 problemas reais da literatura. Incluímos dois tipos de dados em nosso conjunto: o primeiro, para o qual a matriz A tem mais linhas do que colunas, com um forte apelo de regularização dos dados usando a técnica de quadrados mínimos, e o segundo, que contém uma coleção de problemas para os quais a matriz A tem mais colunas do que linhas, que se enquadram em problemas de seleção de variáveis. Os problemas estão listados nas Tabelas 3 e 4, junto com informações sobre a dimensão, fonte, valor de função alvo usado como critério de parada F_{LASSO}^* e a porcentagem de coordenadas nulas da solução $\text{nz}(x_{LAS}^*)$, encontrada pelo método usado para obter o critério de parada. A informação presente nas duas últimas colunas das tabelas referidas anteriormente será detalhada na Subseção 5.3.3. O valor de função alvo para os problemas *LASSO* foram calculados da mesma forma que na Subseção 5.3.1, isto é, usamos um limitante superior para o valor mínimo de função, para cada problema, que está correto até o terceiro dígito e tem diferença de uma unidade para o quarto dígito. Os problemas sofreram um pré-processamento simples, apenas excluindo as colunas nulas eventualmente presentes. Como na última subseção, usaremos o valor de $\lambda = 0.1 \|A^T b\|_\infty$, isto é, $\lambda = 0.1 \|\nabla f(0)\|_\infty$ para todos os testes. Esse valor de λ tem sido usado em vários artigos para realizarem seus experimentos numéricos [38, 46] e está relacionado com o valor de λ que garante a solução nula para o problema (5.14), como demonstrado em (5.11) para o caso particular $f(x) = 1/2 \|Ax - b\|_2^2$.

Tabela 3 – Problemas usados nos testes com matriz possuindo mais linhas do que colunas

Rótulo	Nome	(s, n)	F_{LASSO}^*	$\text{nz}(x_{LAS}^*)$	F_{LOG}^*	$\text{nz}(x_{LOG}^*)$
SL_1	[14] a1a.t	(30956, 123)	1.061×10^4	95.12%	1.605×10^4	95.12%
SL_2	[14] a2a.t	(30296, 123)	1.037×10^4	95.12%	1.569×10^4	95.12%
SL_3	[14] a4a.t	(27780, 123)	9.499×10^3	95.12%	1.439×10^4	95.12%
SL_4	[14] connect-4	(67557, 126)	2.537×10^4	91.26%	*****	*****
SL_5	[14] dna.scale	(2000, 180)	2.355×10^3	32.77%	*****	*****
SL_6	[14] mnist	(60000, 717)	3.241×10^5	95.53%	*****	*****
SL_7	[14] mushrooms	(8124, 112)	2.552×10^3	96,42%	3.741×10^3	98.21%
SL_8	[14] plishing	(11055, 68)	1.034×10^3	88.23%	4.265×10^3	86.76%
SL_9	[14] protein	(17766, 356)	7.108×10^3	77.80%	*****	*****
SL_{10}	[14] w2a	(3470, 293)	1.069×10^3	94.53%	1.551×10^3	95.22%
SL_{11}	[14] w4a.t	(42383, 300)	1.301×10^4	95.33%	1.880×10^4	95.33%
SL_{12}	[14] w5a.t	(39861, 300)	1.224×10^4	95.33%	1.769×10^4	95.33%
SL_{13}	[14] w6a.t	(32561, 300)	9.978×10^3	95.33%	1.442×10^4	95.33%
SL_{14}	[14] w8a.t	(14951, 300)	4.588×10^3	95.33%	6.634×10^3	95.33%
SL_{15}	[14] w5a	(2833, 299)	9.196×10^2	95.98%	1.342×10^3	95.98%
SL_{16}	[14] real-sim	(72309, 20958)	2.562×10^4	99.68%	3.641×10^4	99.75%
SL_{17}	[14] rcv1-test-bin	(677399, 42735)	2.428×10^5	99.83%	3.546×10^5	99.85%
SL_{18}	[14] rcv1-test-mult	(518571, 41400)	3.473×10^7	99.84%	3.505×10^5	99.78%
SL_{19}	[27] J-Lee	(181395, 105353)	1.271×10^2	99.88%	1.137×10^5	98.20%
SL_{20}	[14] webspam-unigram	(350000, 138)	1.413×10^5	92.02%	2.063×10^5	92.02%
SL_{21}	[16] Maragal-4	(1964, 1027)	1.865×10^1	99.22%	*****	*****
SL_{22}	[16] Maragal-5	(4654, 3296)	4.381×10^1	99.72%	3.077×10^3	95.35%
SL_{23}	[16] Maragal-6	(21255, 10144)	5.968×10^1	99.90%	1.425×10^4	97.39%
SL_{24}	[16] Maragal-7	(46845, 26525)	1.414×10^2	99.95%	3.181×10^4	99.18%

5.3.2.2 Otimização extra no subspaço das restrições inativas e seleção de parâmetros

Novamente precisamos ajustar os valores dos parâmetros δ_{DP} e δ_F que definem completamente o método **ActiveBCDM**. Fizemos isso da mesma forma que na Subseção 5.3.1, testando uma combinação de 24 valores diferentes em um subconjunto dos 49 problemas. Usamos os problemas de SL_1 a SL_9 e de SC_1 a SC_9 para ajustar os parâmetros.

Para definir os melhores parâmetros δ_{DP}, δ_F para **ActiveBCDM**, comparamos o tempo médio de *CPU* entre 10 rodadas no conjunto de problemas teste e analisamos seus respectivos *performance profiles*.

Para as variantes do método **ActiveBCDM**, o desempenho das variantes mais expressivas estão presentes na Figura 6. Podemos ver que os valores que produziram o melhor desempenho para **ActiveBCDM** foram $\delta_{DP} = 10^3$ e $\delta_F = [0.1n]$.

Em uma implementação típica de um método de descenso coordenado para o problema *LASSO*, o gradiente da parte suave da função objetivo não é calculado completamente cada vez que o passo do bloco de coordenadas é atualizado, as coordenadas do gradiente são calculadas somente quando necessário, veja por exemplo [36]. Contudo, quando implementamos **ActiveBCDM**, após δ_F iterações, precisamos calcular o gradiente da parte suave da função objetivo por inteiro para estimar o conjunto \mathcal{J} de possíveis restrições ativas. Como esse cálculo gera um grande esforço computacional, quando comparado a uma única iteração do nosso método de descenso coordenado, é natural para nós tentarmos explorar essa oportunidade para realizar um passo

Tabela 4 – Problemas usados nos testes com matriz possuindo mais colunas do que linhas

Rótulo	Nome	(s, n)	F_{LASSO}^*	$\text{nz}(x_{LAS}^*)$	F_{LOG}^*	$\text{nz}(x_{LOG}^*)$
SC_1	[9] peppers05-6-6	(32768, 65536)	7.277×10^4	92.53%	*****	*****
SC_2	[9] peppers05-12-12	(32768, 65536)	1.343×10^5	92.27%	*****	*****
SC_3	[9] peppers025-12-12	(16384, 65536)	1.184×10^5	88.83%	*****	*****
SC_4	[3] SparcoProblem401	(29166, 57344)	1.605×10^2	98.66%	*****	*****
SC_5	[3] SparcoProblem402	(29166, 57344)	1.605×10^2	98.66%	*****	*****
SC_6	[3] SparcoProblem603	(1024, 4096)	1.099×10^2	99.75%	*****	*****
SC_7	[9] finance1000	(30465, 216842)	1.846×10^5	99.99%	2.110×10^4	99.99%
SC_8	[28] dbworld-bodies	(64, 4702)	5.134×10^0	99.72%	2.104×10^1	99.44%
SC_9	[28] dexter-train	(300, 7751)	1.175×10^2	99.75%	1.683×10^2	99.78%
SC_{10}	[28] dexter-valid	(300, 7847)	1.147×10^2	99.65%	1.667×10^2	99.71%
SC_{11}	[28] dorothea-train	(800, 88119)	2.043×10^2	99.89%	3.238×10^2	99.92%
SC_{12}	[28] dorothea-valid	(350, 72113)	8.833×10^1	99.88%	1.410×10^2	99.91%
SC_{13}	[14] news20-binary	(19996, 1355191)	7.334×10^3	99.99%	1.078×10^4	99.99%
SC_{14}	[14] news20-scale	(15935, 60346)	6.586×10^5	99.98%	1.075×10^4	99.86%
SC_{15}	[14] news20-t-scale	(3993, 39128)	1.652×10^5	99.98%	2.699×10^3	99.90%
SC_{16}	[14] rcv1-train-bin	(20242, 44504)	7.064×10^3	99.82%	1.042×10^4	99.84%
SC_{17}	[14] rcv1-train-mult	(15564, 36842)	1.054×10^6	99.79%	1.022×10^4	99.35%
SC_{18}	[14] sector-scale	(6412, 49087)	6.009×10^6	99.90%	4.335×10^3	99.94%
SC_{19}	[14] sector-t-scale	(3207, 39234)	2.994×10^6	99.89%	2.182×10^3	99.89%
SC_{20}	[27] Blanc-Mel	(186414, 685568)	2.596×10^2	99.99%	1.189×10^5	99.84%
SC_{21}	[28] farm-ads-vect	(4143, 54877)	1.444×10^3	99.91%	2.129×10^3	99.92%
SC_{22}	[9] mug05-12-12	(12410, 24820)	5.363×10^4	93.73%	*****	*****
SC_{23}	[9] mug025-12-12	(6205, 24820)	4.982×10^4	92.19%	*****	*****
SC_{24}	[9] mug075-12-12	(13651, 24820)	5.837×10^4	94.40%	*****	*****
SC_{25}	[16] Maragal-8	(33212, 60845)	2.037×10^2	99.76%	2.180×10^4	99.65%

extra na tentativa de acelerar a convergência do método.

Para atingir esse objetivo, nos inspiramos no artigo [21], um trabalho com a meta de incorporar informação de segunda ordem no modelo que define o subproblema (2.6), resolvendo-o de maneira inexata. Depois de calcular o conjunto \mathcal{J} , associado às restrições ativas, e \mathcal{I} , associado às variáveis inativas, tentaremos realizar um passo ao longo da direção

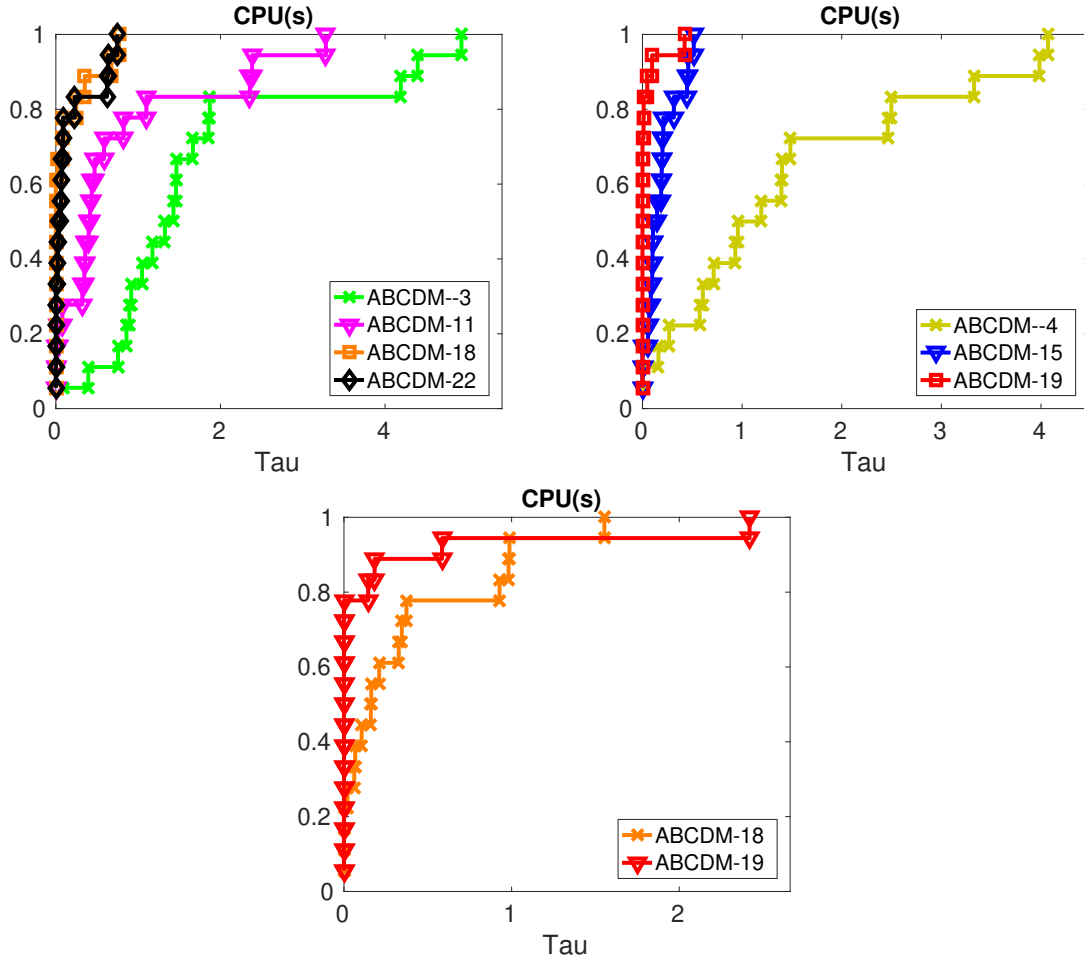
$$h_{\mathcal{I}}(x) = \arg \min_{x+h \in \mathcal{X}} \left\{ \nabla_{\mathcal{I}} f(x)^T h_{\mathcal{I}} + \frac{1}{2} h_{\mathcal{I}}^T D h_{\mathcal{I}} + \psi_{\mathcal{I}}(x_{\mathcal{I}} + h_{\mathcal{I}}) \right\}, \quad (5.16)$$

com $h = U_{\mathcal{I}} h_{\mathcal{I}}$. A matrix D carrega informação de segunda ordem da função suave f . Seguindo os resultados numéricos de [21], testamos duas escolhas para D , $D = \text{diag}(\nabla_{\mathcal{I}}^2 f(x))$ e $D = \nabla_{\mathcal{I}}^2 f(x) + \sigma I$, com $\sigma = 10^{-6}$, onde $\text{diag}(\cdot)$ é uma função que retorna a matriz diagonal cujos elementos são os que compõem a diagonal do argumento. A convergência do algoritmo foi preservada por forçar a monotonicidade da função objetivo ao longo da sequência de iterações gerada. Para garantir isso, nós empregamos um critério de decréscimo suficiente inspirado no algoritmo SpARSA [46]. Para decidir quando $h_{\mathcal{I}}(x)$ é aceito, realizamos o teste

$$F(x + U_{\mathcal{I}} h_{\mathcal{I}}) \leq F(x) - \frac{\gamma}{2} \|h_{\mathcal{I}}\|_2^2, \text{ com } \gamma = 10^{-6},$$

e aceitamos o passo se o teste é satisfeito. Caso contrário, o passo extra é ignorado. Chamaremos essa variação de ActiveBCDM+S0, em que S0 vem em alusão ao termo de segunda ordem, escrito em inglês, *Second Order*.

Figura 6 – *Performance profiles* mais significativos entre as 24 variantes do método ActiveBCDM para 18 problemas das Tabelas 3 e 4



Cada uma das escolhas das matrizes do modelo (5.16) levam a subproblemas muito distintos de serem resolvidos. A escolha $D = \text{diag}(\nabla_{\mathcal{I}}^2 f(x))$ gera um modelo separável para $\psi(x) = \lambda \|x\|_1$, e portanto, que pode ser resolvido com fórmula fechada; o método para esse caso será chamado **ActiveBCDM+S01**. Para o caso $D = \nabla_{\mathcal{I}}^2 f(x) + \sigma I$, temos um problema quadrático que não pode ser resolvido de maneira exata, e portanto, precisamos escolher uma metodologia para resolvê-lo.

Para esse fim, decidimos testar três estratégias distintas para obter, aproximadamente, o minimizador do subproblema (5.16). As três estratégias são métodos desenvolvidos para resolver problemas envolvendo a regularização da norma 1, que tentam explorar informação sobre a hessiana da parte suave do problema. O método **ActiveBCDM+S02** usa um método apresentado no artigo [1] e chamado **OWL-QN**, sigla do inglês *Ortant Wise Limited-memory Quasi-Newton*. Para as outras duas opções, **ActiveBCDM+S03** e **ActiveBCDM+S04**, usamos dois algoritmos **PSSas** e **PSSgb** introduzidos em [39], que usam informações de segunda ordem em diversos problemas, inclusive o problema de regressão logística com regularização ℓ_1 . Todos os 3 métodos envolvem a aplicação do método BFGS com memória limitada, desenvolvido especificamente para a regularização ℓ_1 . Utilizamos implementações dos 3 métodos em MATLAB feitas pelo professor Mark Schmidt [39] e

disponível no site ¹, com os nomes *L1General-OWL*, *L1General-PSSas* e *L1General-PSSgb*, implementações dos métodos *Orthant-wise learning*, *Projected scaled sub-gradient (active set variant)* e *Projected scaled sub-gradient (Gafni-Bertsekas variant)*, respectivamente.

Mostraremos um algoritmo (Algoritmo 3) que acrescenta, ao nosso Algoritmo 1, a minimização extra nas variáveis inativas, isto é, nas variáveis do conjunto \mathcal{I} .

Algoritmo 3: *Active Block Coordinate Descent Method plus Second Order Descent Step (Active BCDM+SO)*

Input: $x^0 \in \mathcal{X}$, $\delta_{DP}, \delta_F \in \mathbb{N}$, $B \in \mathbb{S}_{++}^n$, $\epsilon \in \mathbb{R}_+$, $\ell_{\max} \in \mathbb{N}_+$, $\sigma = \gamma = 10^{-6}$
Output: x^k

begin
 $\mathcal{I} \leftarrow \{1, \dots, m\};$
 $\mathcal{J} \leftarrow \emptyset;$
 $\ell \leftarrow 1;$
repeat
 Defina a distribuição de probabilidade;

$$\mathbb{P}(i) = \begin{cases} \frac{\delta_{DP}}{\delta_{DP}|\mathcal{I}| + |\mathcal{J}|}, & \text{se } i \in \mathcal{I}, \\ \frac{1}{\delta_{DP}|\mathcal{I}| + |\mathcal{J}|}, & \text{se } i \in \mathcal{J}. \end{cases}$$

for $k = (\ell - 1)\delta_F + 1$ **to** $\ell\delta_F$ **do**
 Escolha um bloco $i \in \{1, \dots, m\}$ com a distribuição de probabilidade anterior;
 Encontre $h_i(x^k)$ solução do problema (2.6);
 $x^{k+1} \leftarrow x^k + U_i h_i(x^k);$
 $v_{(i)} \leftarrow h_i(x^k);$
end
 Escolha o conjunto $\mathcal{J} \subset \{1, \dots, m\}$ de alguma maneira;
 $\mathcal{I} \leftarrow \{1, \dots, m\} - \mathcal{J};$
 $D = \text{diag}(\nabla_{\mathcal{I}}^2 f(x))$ ou $D = \nabla_{\mathcal{I}}^2 f(x) + \sigma I;$
 Resolva o problema (5.16) de maneira exata ou inexata;
 Calcule o decréscimo na função objetivo;
 $F(x^{\ell\delta_F} + U_{\mathcal{I}} h_{\mathcal{I}}(x^{\ell\delta_F})) \leq F(x^{\ell\delta_F}) - \frac{\gamma}{2} \|h_{\mathcal{I}}(x^{\ell\delta_F})\|_2^2;$
 if *decrécimo é satisfeito* **then**
 $x^{\ell\delta_F} \leftarrow x^{\ell\delta_F} + U_{\mathcal{I}} h_{\mathcal{I}}(x^{\ell\delta_F});$
 end
 $\ell \leftarrow \ell + 1;$
until $\|v\| \leq \epsilon$ *or* $\ell = \ell_{\max};$
end

Fizemos os testes para ajustar os parâmetros δ_{DP}, δ_F , para as quatro versões do método **ActiveBCDM+SO**(\cdot). Usamos novamente o tempo médio de *CPU* entre 10 rodadas para cada problema teste como medida de desempenho e analisamos seus respectivos *performance*

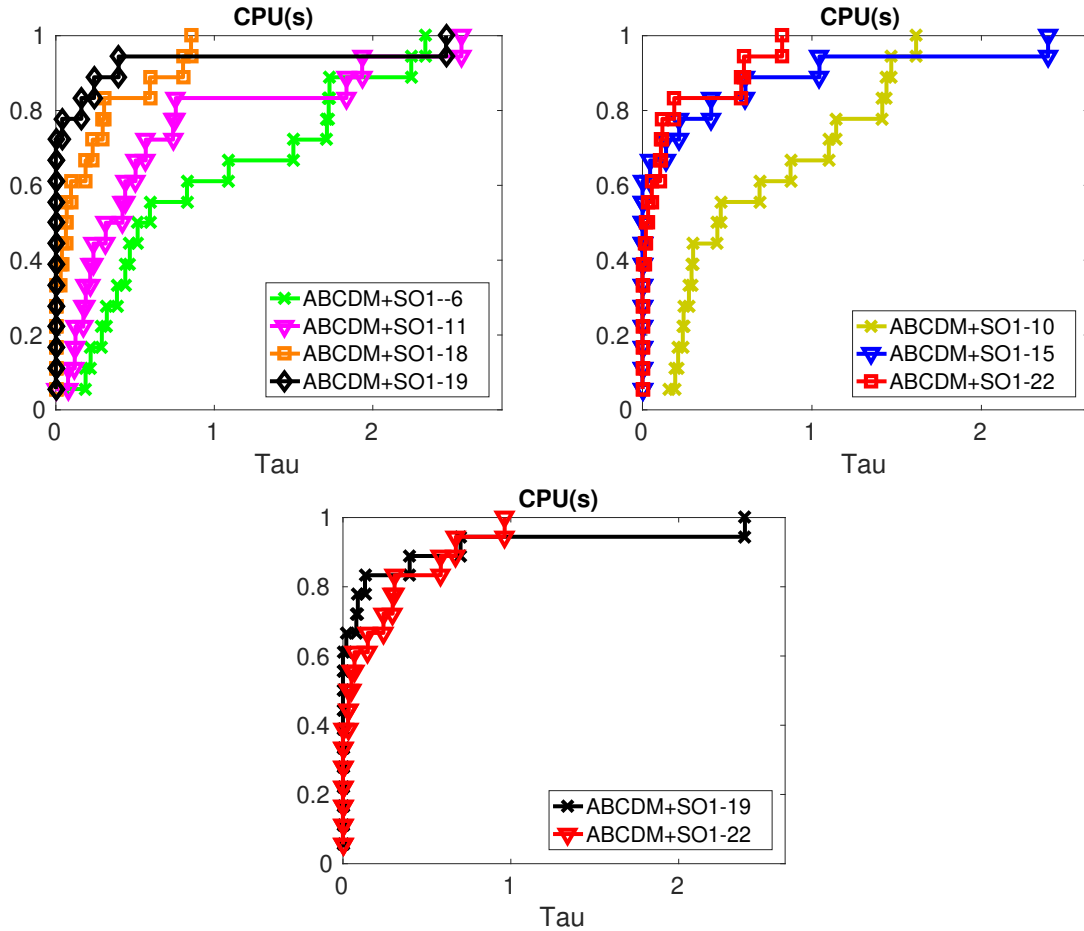
¹ <http://www.cs.ubc.ca/~schmidtm/Software/L1General.html>

profiles para concluir qual a melhor combinação de parâmetros. As Figuras 7, 8, 9 e 10 mostram os *performance profiles* mais promissores, para cada um dos métodos analisados.

Comparando as quatro figuras citadas anteriormente, podemos ver que as melhores combinações de parâmetros para cada um dos métodos são:

- ActiveBCDM+S01: $\delta_{DP} = 10^3$ e $\delta_F = [0.1n]$;
- ActiveBCDM+S02: $\delta_{DP} = 5$ e $\delta_F = [0.1n]$;
- ActiveBCDM+S03: $\delta_{DP} = 2$ e $\delta_F = [0.01n]$;
- ActiveBCDM+S04: $\delta_{DP} = 100$ e $\delta_F = [0.01n]$.

Figura 7 – *Performance profiles* mais significativos entre as 24 variantes do método ActiveBCDM+S01 para 18 problemas das Tabelas 3 e 4



Com as escolhas dos melhores parâmetros para os métodos ActiveBCDM+S0(·) nesse conjunto de 18 problemas, comparamos os 4 métodos entre si por meio de *performance profiles* usando o tempo médio de CPU, Figura 11. Vemos por meio dessa figura, entre os três métodos que usam informação de segunda ordem no passo extra (5.16), o método ActiveBCDM+S04 foi mais eficiente e resolveu os problemas mais rapidamente. Comparando os métodos ActiveBCDM+S04 e ActiveBCDM+S01, observamos que o método ActiveBCDM+S04 teve um desempenho superior,

Figura 8 – *Performance profiles* mais significativos entre as 24 variantes do método ActiveBCDM+SO2 para 18 problemas das Tabelas 3 e 4

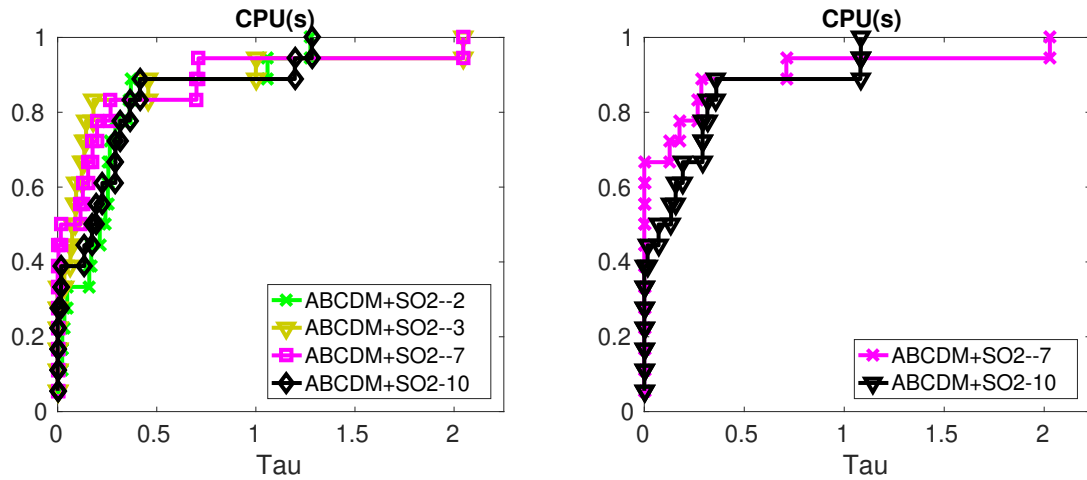
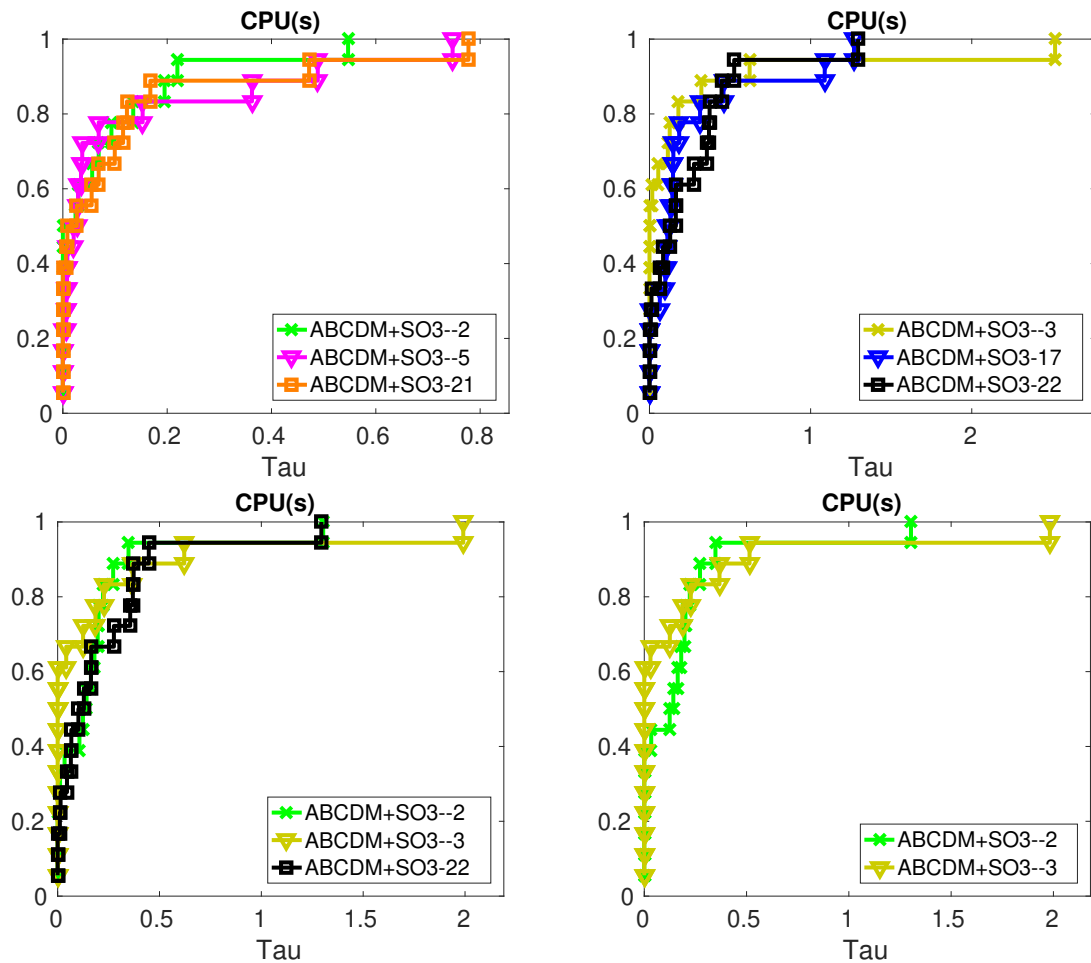


Figura 9 – *Performance profiles* mais significativos entre as 24 variantes do método ActiveBCDM+SO3 para 18 problemas das Tabelas 3 e 4



pois foi mais eficiente e resolveu os problemas em tempo semelhante ao método ActiveBCDM+SO1. Chamaremos na sequência da subseção, esse método de melhor desempenho simplesmente de ActiveBCDM+SO.

Figura 10 – *Performance profiles* mais significativos entre as 24 variantes do método ActiveBCDM+SO4 para 18 problemas das Tabelas 3 e 4

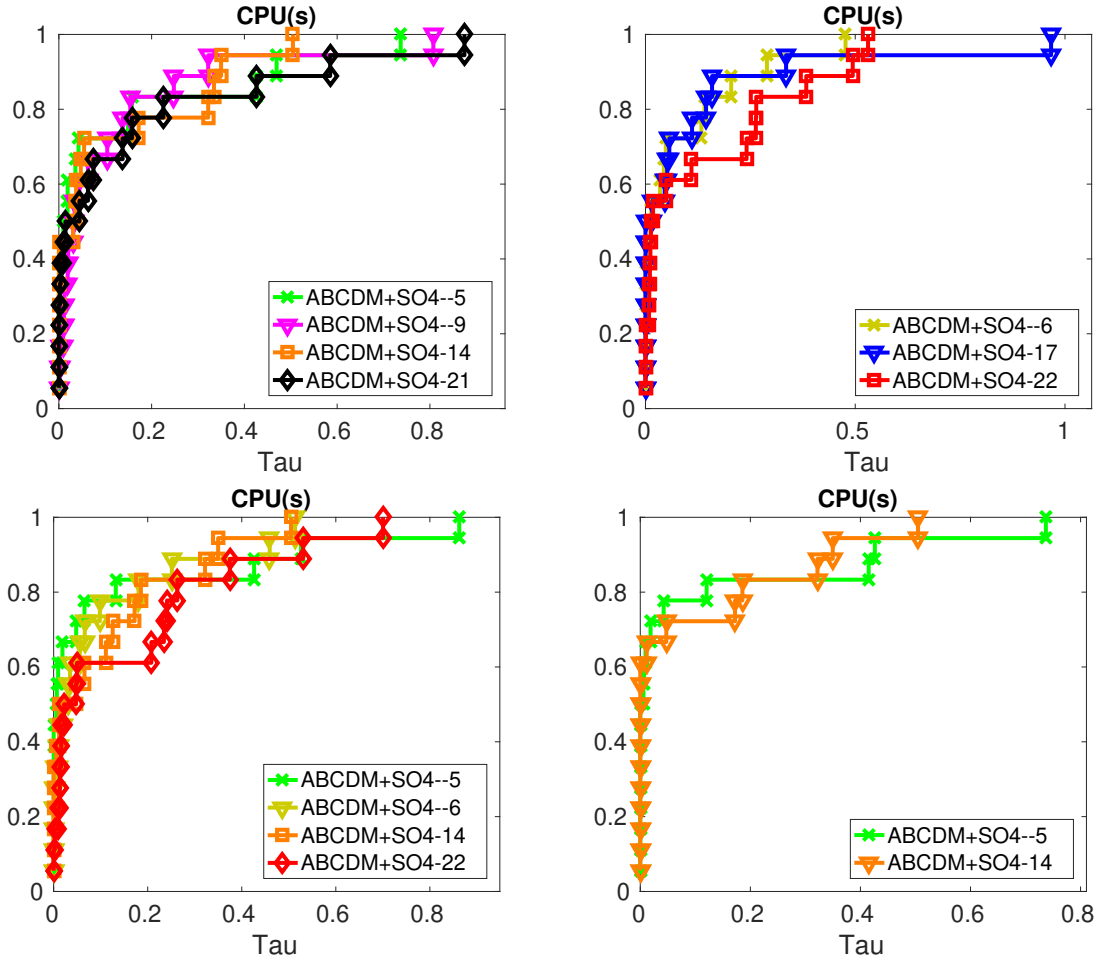
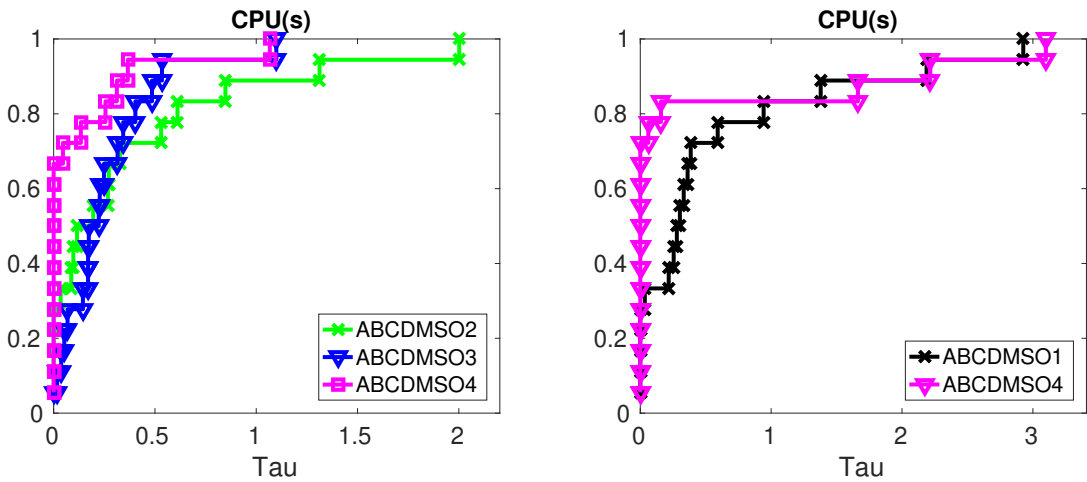


Figura 11 – *Performance profiles* entre as variantes do método ActiveBCDM+SO(\cdot) calibradas para 18 problemas das Tabelas 3 e 4



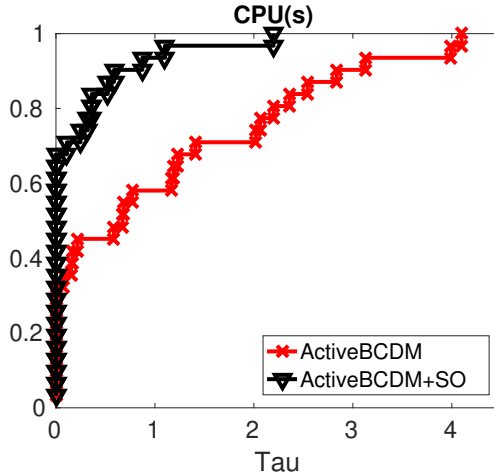
Na próxima subseção, comparamos as versões ajustadas dos algoritmos *Active BCDM* e *Active BCDM*+SO entre si e entre outros métodos bem conhecidos da literatura, para resolver o problema *LASSO*.

5.3.2.3 Comparação com outros métodos

Nesta subseção compararemos as versões calibradas dos algoritmos *Active BCDM* e *Active BCDM+SO* contra outros métodos da literatura, resolvendo 31 problemas obtidos depois de descartarmos os problemas SL_1 a SL_9 da Tabela 3 e SC_1 a SC_9 da Tabela 4, que foram considerados no estágio de ajuste dos parâmetros, por meio de *performance profiles* obtidos através do tempo médio de *CPU* calculado levando em consideração 10 rodadas.

Inicialmente, comparamos os métodos ajustados na subseção anterior, **ActiveBCDM** e **ActiveBCDM+SO**. O *performance profile* contendo esse resultado está presente na Figura 12. Podemos notar que o método **ActiveBCDM+SO** tem desempenho superior ao método **ActiveBCDM**, para esse conjunto de problemas, tanto com respeito à eficiência quanto com relação a resolver os problemas em menor tempo.

Figura 12 – *Performance profile* entre **ActiveBCDM** e **ActiveBCDM+SO** para 31 problemas das Tabelas 3 e 4



O primeiro método comparativo testado é conhecido como **SpaRSA** [46]. Ele é baseado na iteração

$$x^{k+1} = \operatorname{argmin}_z \nabla f(x^k)^T (z - x^k) + \frac{\alpha_k}{2} \|z - x^k\|_2^2 + \lambda \|z\|_1,$$

com tamanho de passo α_k selecionado usando uma regra não-monotóna. Para detalhes, veja [46].

O segundo método é chamado **FAST-BCD2-E**, e foi proposto em [38]. Ele é um método de descenso coordenado que obtém direções de descida com a mesma regra do algoritmo *Active BCDM*. Ele também se apoia em uma estratégia de identificação das coordenadas nulas de um minimizador do problema, porém usando uma abordagem diferente da nossa. Além disso, eles usam essa identificação de maneira distinta do nosso texto. Após esse passo de identificação das coordenadas nulas, eles fixam um subconjunto das coordenadas não nulas que mais violam as condições de otimalidade para serem atualizadas nas próximas iterações do método de maneira cíclica. Para mais detalhes, veja [38].

O terceiro método é conhecido como **OWL-QN** [1]. Ele usa um método BFGS de memória limitada aplicado ao problema (5.14), usando o subgradiente de norma mínima da função objetivo F no lugar do gradiente do modelo, e minimizando o modelo sobre um ortante específico, para o

qual é garantida a diferenciabilidade da função F . Usamos um código implementado por Mark Schmidt e descrito em [39].

Também testamos outros dois métodos, PSSas e PSSgb. Ambos são implementações de estratégias de BFGS com memória limitada e foram desenvolvidos por Mark Schmidt em sua tese de doutorado [39]. Foram escolhidos porque apresentaram bons resultados numéricos no contexto de regressão logística com regularização ℓ_1 .

Para os testes, usamos as implementações originais dos métodos disponibilizadas pelos autores (em MATLAB) e usamos os parâmetros padrões de cada um dos métodos. Para todos os testes, comparamos o tempo total de *CPU* gasto por cada um dos métodos para encontrar um ponto cujo valor de função objetivo é menor ou igual a F^* , todos eles começando do vetor nulo. Como alguns métodos são não determinísticos, como *Active BCDM*, consideramos o tempo médio de 10 simulações para construir os *performance profiles* e proceder com a análise.

A Figura 13 apresenta o *performance profile* comparando todos os métodos baseados nas atualizações BFGS contra o método *ActiveBCDM+SO*. A figura à esquerda compara os métodos entre si em resolver nosso conjunto de problemas teste e a figura à direita compara o método baseado na heurística BFGS de melhor desempenho, chamado PSSgb, contra *ActiveBCDM+SO*. O método *ActiveBCDM+SO* mostra uma grande vantagem em termos de eficiência, porém perde em termos do tempo gasto para resolver todos os problemas, quando comparado ao método PSSgb.

Por meio da Figura 14, comparamos o método *ActiveBCDM+SO* com FAST-BCD2-E, (figura à esquerda) e com SpaRSA (figura à direita). Os *performance profiles* indicam que *ActiveBCDM+SO* tem um desempenho superior a ambos os métodos para esse conjunto de problemas teste.

Figura 13 – *Performance profiles* entre os métodos quase-Newton (esquerda) e entre o melhor método quase-Newton e *ActiveBCDM+SO* (direita) para 31 problemas das Tabelas 3 e 4.

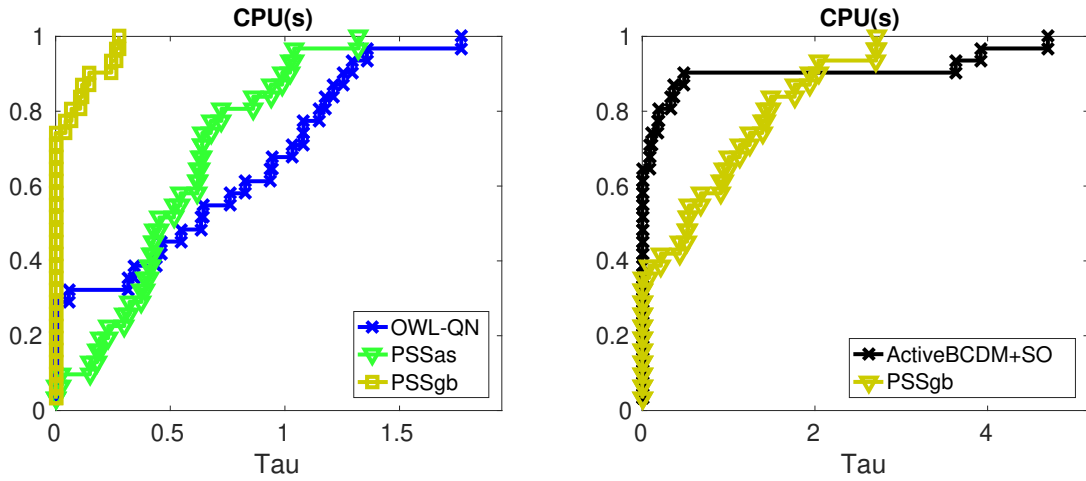
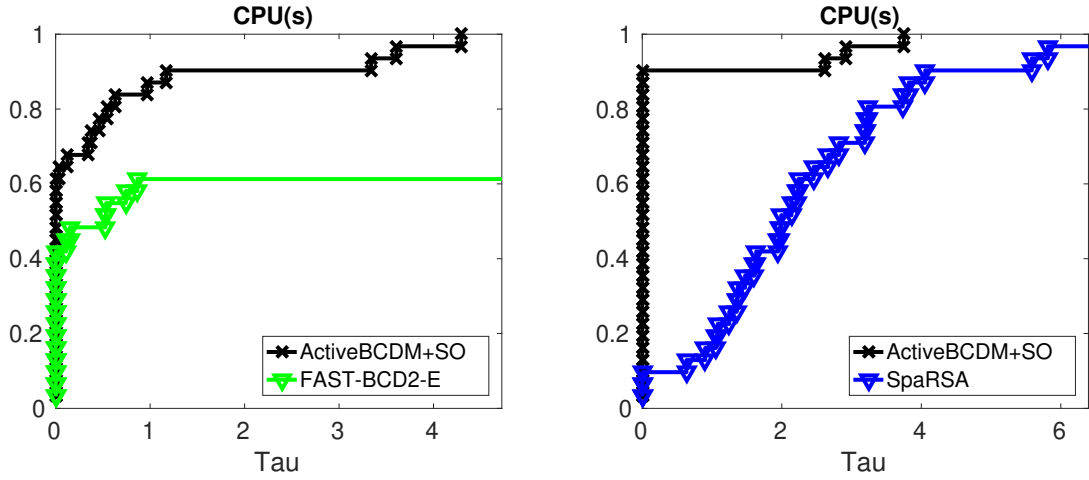


Figura 14 – *Performance profiles* entre ActiveBCDM+SO e FAST-BCD2-E (esquerda) e entre ActiveBCDM+SO e SpaRSA (direita) para 31 problemas das Tabelas 3 e 4.



5.3.3 Regressão logística com regularização ℓ_1

5.3.3.1 Conjunto de dados e critério de parada

Nessa subseção, descreveremos os resultados envolvendo os testes das variantes do algoritmo *Active BCDM* para resolver o problema de regressão logística com regularização ℓ_1 (5.4). Consideramos o parâmetro $\lambda = 0.1\|\nabla f(x^0)\|$, similar à escolha usada no problema *LASSO* para construir soluções esparsas. Todos os métodos analisados usam como ponto inicial o vetor $x^0 = 0$, como no problema anterior.

O problema de regressão logística com regularização ℓ_1 é geralmente aplicado em problemas de classificação binária, como explicado na Seção 5.2. Porém, somente os problemas 1 – 3, 8, 10 – 17, 19, e 20, da Tabela 3 e os problemas 8 – 13, 16, 20, e 21 da Tabela 4 têm essa característica, totalizando 23 problemas teste de classificação binária.

Para tentarmos usar a informação disponível da matriz A dos 26 problemas restantes, criamos um vetor binário b aleatoriamente usando o seguinte procedimento. Começamos gerando um vetor x aleatório com coordenadas em $\mathcal{N}(0, 1)$. Depois, anulamos aproximadamente 50% das entradas de x , novamente aleatoriamente, e calculamos o vetor b como o sinal do vetor Ax . Mais uma vez aleatoriamente, trocamos o sinal de aproximadamente 10% das entradas do vetor b para introduzir erros de classificação nos dados. Finalmente, checamos, resolvendo o problema com o método *PSSgb*, se a solução tem ao menos 90% das coordenadas nulas. Os problemas para os quais essa meta foi atingida foram SL_7 , SL_{18} , SL_{22} a SL_{24} , SC_7 , SC_{14} , SC_{15} , SC_{17} a SC_{19} , SC_{25} . Adicionamos tais problemas ao conjunto original de 23 problemas de classificação binária, descritos anteriormente, totalizando 35 problemas testes. A ênfase em problemas com soluções muito esparsas é justificada pelo fato que a estratégia de identificação possuirá mais efeito nesse contexto.

Como na Subseção 5.3.1, usamos como critério de parada o valor de função alvo que é um limitante superior para o valor de função mínimo, com erro de uma unidade no quarto dígito. Esses limitantes superiores são fornecidos na penúltima coluna das Tabelas 3 e 4. Na

última coluna, apresentamos a porcentagem de coordenadas nulas da solução encontrada pelo método para obter o valor de função alvo, representado por $\text{nz}(x_{LOG}^*)$. Convencionamos * * * * para indicar os problemas cujos dados são inapropriados para os testes, isto é, mesmo tentando criar um novo vetor b artificial para esses problemas, a solução do novo problema não tinha a esparsidade desejada.

5.3.3.2 Otimização extra no subspaço das restrições inativas e seleção de parâmetros

Nessa subseção, iremos ajustar os parâmetros δ_{DP} , δ_F e a escolha da matriz D como na Subseção 5.3.2.1 para os métodos **ActiveBCDM** e **ActiveBCDM+SO**. Usamos 12 problemas, dentre os 35 selecionados, nesse procedimento de ajuste, nomeados por SL_1 , SL_2 , SL_3 , SL_8 , SL_{10} , SL_{11} , SC_8 , SC_9 , SC_{10} , SC_{11} , SC_{12} , SC_{13} . Usamos para a calibragem somente os problemas que tinham originalmente o vetor b binário.

Por meio das Figuras 15, 16, 17, 18 e 19, vemos que as melhores combinações de parâmetros são:

- **ActiveBCDM**: $\delta_{DP} = 10^4$ e $\delta_F = [0.001n]$;
- **ActiveBCDM+SO1**: $\delta_{DP} = 10^4$ e $\delta_F = [0.001n]$;
- **ActiveBCDM+SO2**: $\delta_{DP} = 10^3$ e $\delta_F = [0.01n]$;
- **ActiveBCDM+SO3**: $\delta_{DP} = 10^3$ e $\delta_F = [0.01n]$;
- **ActiveBCDM+SO4**: $\delta_{DP} = 10^3$ e $\delta_F = [0.01n]$.

Escolhidas as melhores variantes para cada método **ActiveBCDM+SO**(\cdot), comparamos as melhores variantes entre si, os resultados estão presentes na Figura 20. Por meio da Figura 20 (esquerda), vemos que o desempenho do método **ActiveBCDM+SO4** é superior ao dos outros métodos que usam informação de segunda ordem. Quando comparado com o método **ActiveBCDM+SO1**, Figura 20 (direita), vemos que o método **ActiveBCDM+SO4** ganha em eficiência e resolve quase 80% dos problemas bem mais rápido que o método **ActiveBCDM+SO1**, porém perde no tempo gasto para resolver todos os problemas teste. Mesmo assim, pelo desempenho geral, escolhemos o método **ActiveBCDM+SO4** para representar o algoritmo *Active BCDM+SO*. Na sequência da subseção este será chamado apenas de **ActiveBCDM+SO**.

5.3.3.3 Comparação com outros métodos

Comparamos as versões calibradas dos algoritmos *Active BCDM* e *Active BCDM+SO* e outros cinco métodos da literatura no conjunto de 23 problemas restantes, após excluirmos os 12 problemas usados para o ajuste de parâmetros. Mais uma vez, os métodos usados nesse contexto são implementados (em MATLAB) por meio das versões originais dos autores e parâmetros padrões. A métrica para comparação utilizada foi o tempo de *CPU* médio obtido por meio de 10 simulações.

Figura 15 – *Performance profiles* mais significativos entre as 24 variantes do método ActiveBCDM para 12 problemas das Tabelas 3 e 4

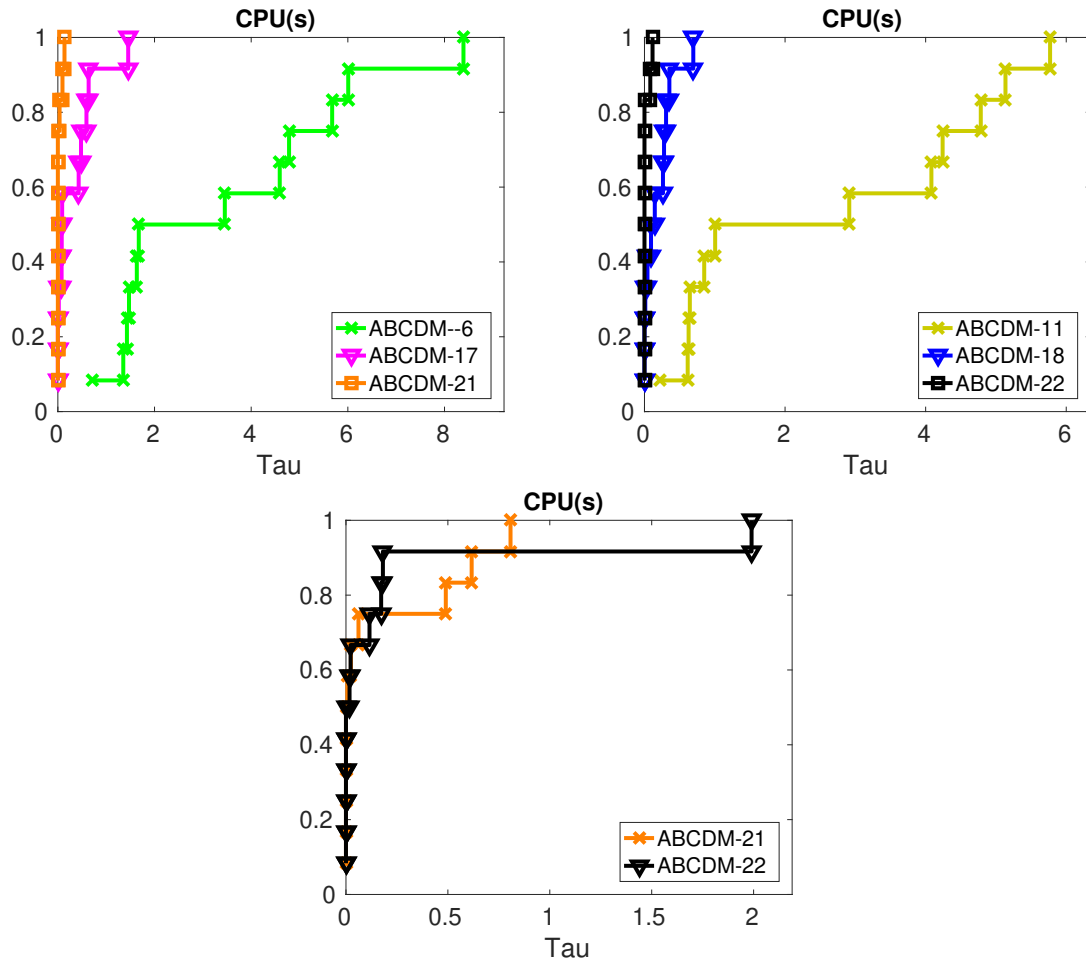
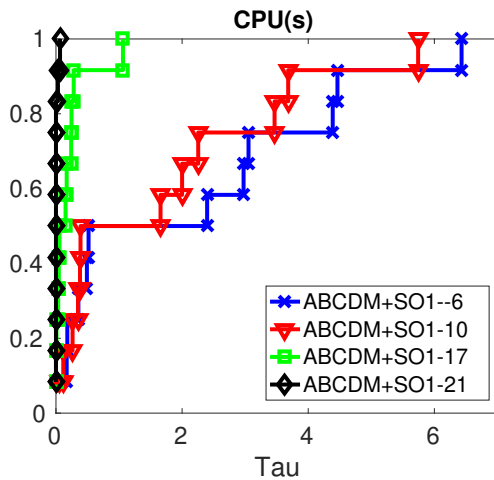
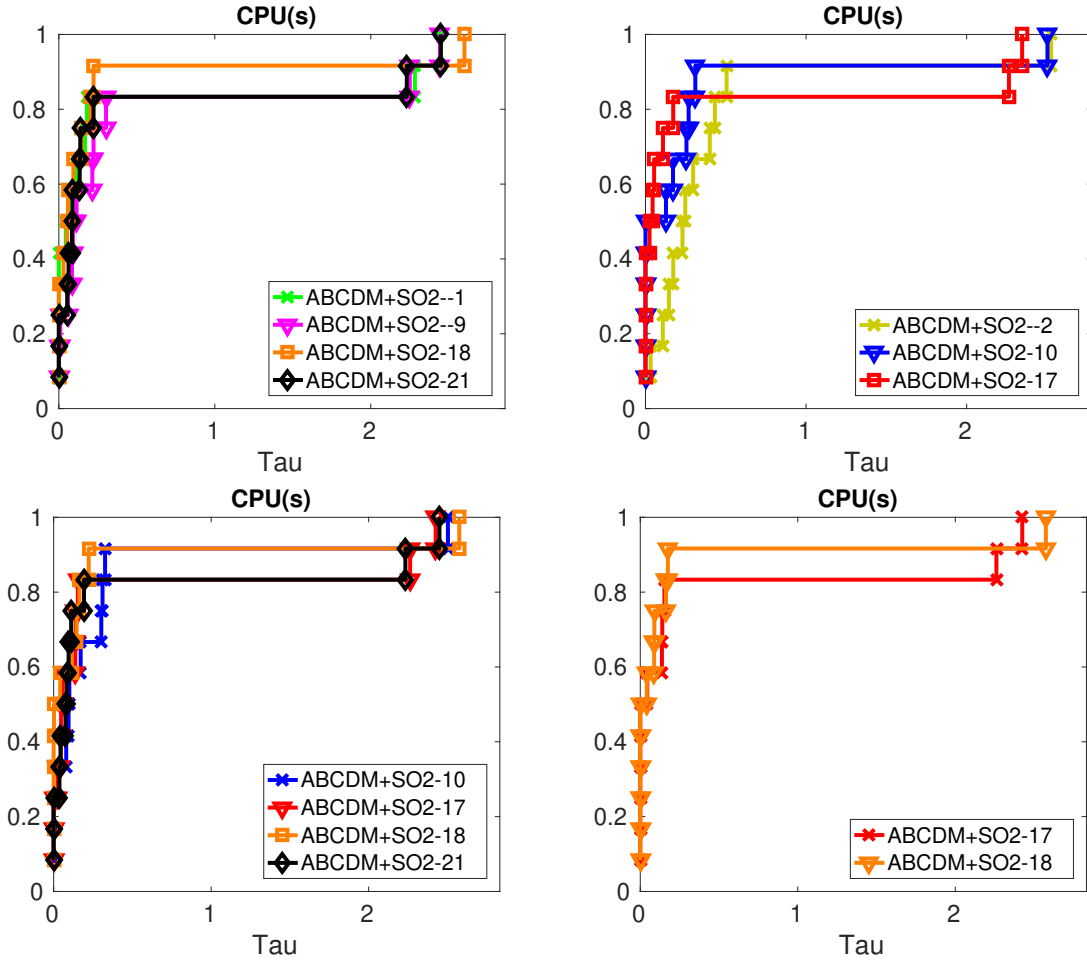


Figura 16 – *Performance profiles* mais significativos entre as 24 variantes do método ActiveBCDM+SO1 para 12 problemas das Tabelas 3 e 4



Comparamos os métodos ActiveBCDM e ActiveBCDM+SO (Figura 21). Podemos ver que, para o problema regressão logística com regularização ℓ_1 , o passo extra de segunda ordem se mostrou eficiente em resolver nosso conjunto de problemas teste.

Figura 17 – *Performance profiles* mais significativos entre as 24 variantes do método ActiveBCDM+SO2 para 12 problemas das Tabelas 3 e 4

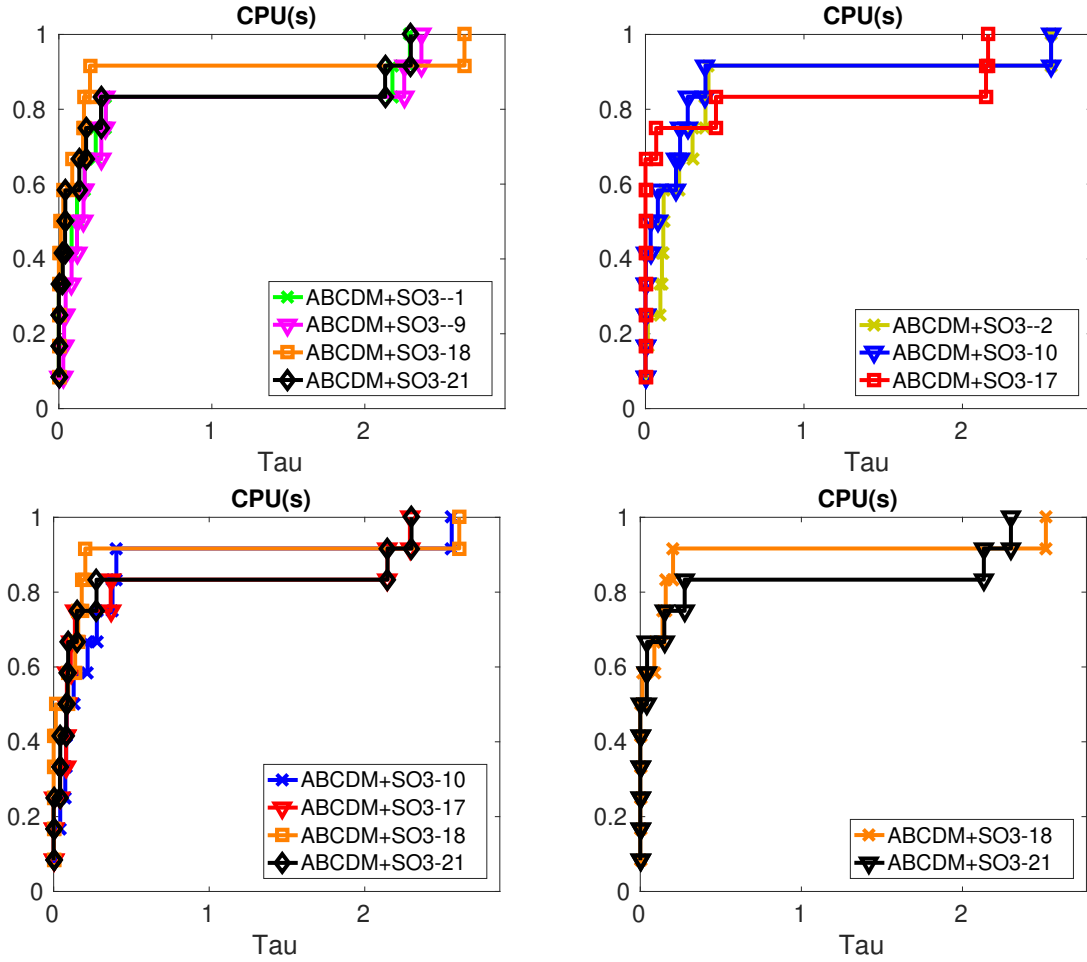


Na sequência, comparamos o método ActiveBCDM+SO com outros métodos da literatura. Começamos comparando com as três heurísticas baseadas no método BFGS de memória limitada que foram usadas na subseção anterior, nomeados como OWL-QN, PSSas, e PSSgb. A Figura 22 mostra, à esquerda, o *performance profile* entre esses três métodos resolvendo os 23 problemas teste mencionados anteriormente. À direita, segue o *performance profile* comparando o método tipo BFGS com melhor desempenho, PSSgb, com ActiveBCDM+SO. Verificamos que o método PSSgb tem desempenho superior ao método ActiveBCDM+SO nesse conjunto de testes analisado.

Finalmente, comparamos o método ActiveBCDM+SO com os métodos FCDv.1 e FCDv.2, introduzidos em [21]. Também métodos de descenso coordenado, FCDv.1 e FCDv.2 atualizam os blocos de maneira semelhante ao algoritmo *Active BCDM*, mas admitem que os subproblemas sejam resolvidos inexatamente, usando informação da diagonal da Hessiana (FCDv.1) e a Hessiana completa (FCDv.2) da parte suave da função objetivo f por blocos, com blocos de tamanho $[0.001n]$.

Analisamos o tempo médio do CPU entre os dois métodos abordados em [21] (Figura 23) à esquerda. Após, analisamos o tempo gasto pelo método de melhor desempenho,

Figura 18 – *Performance profiles* mais significativos entre as 24 variantes do método ActiveBCDM+SO3 para 12 problemas das Tabelas 3 e 4



FCDv.1, contra o nosso método de descenso coordenado ActiveBCDM+SO. Vemos que o método ActiveBCDM+SO claramente tem um desempenho muito superior aos métodos FCDv.1 e FCDv.2.

5.4 Testes em FORTRAN

Nosso objetivo nessa seção é estudar o comportamento dos métodos de descenso coordenado analisados nesse trabalho, na linguagem de programação FORTRAN. A principal justificativa para utilizar essa linguagem de programação, após apresentarmos um vasto conjunto de testes na linguagem de programação MATLAB, vem do fato de que nela podemos explorar as versões em paralelo dos métodos de descenso coordenado abordados no texto e testá-las em computadores com um número maior de núcleos de processamento, desde que em nosso contexto, os computadores com maior número de núcleos de processamento não possuem licenças disponíveis para uso do MATLAB.

Nossos experimentos numéricos foram realizados em um Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz. Usamos o compilador de FORTRAN da Intel (ifort) e as bibliotecas da Intel fornecidas pelo pacote Intel(R) Composer XE 2013. Como todas as rotinas envolvendo

Figura 19 – *Performance profiles* mais significativos entre as 24 variantes do método ActiveBCDM+SO4 para 12 problemas das Tabelas 3 e 4

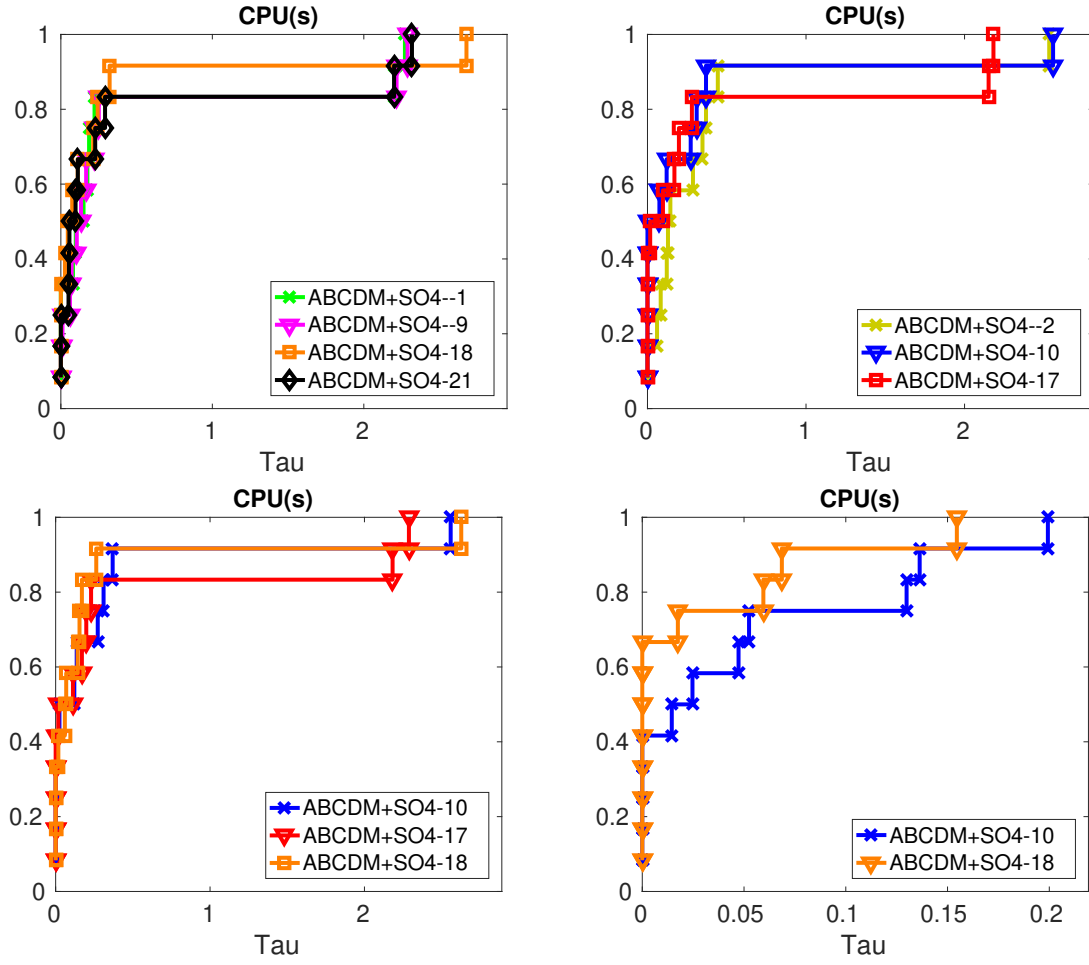
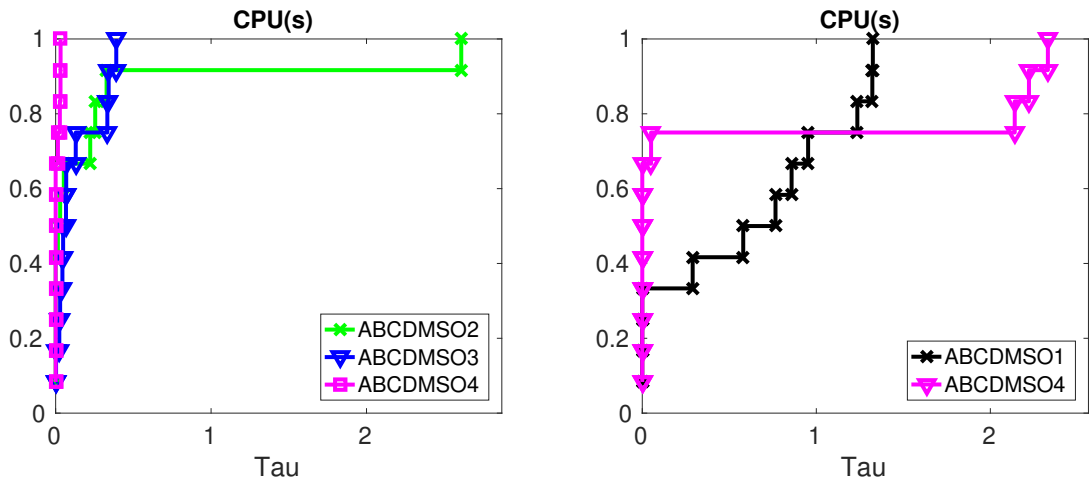


Figura 20 – *Performance profile* entre as variantes do método ActiveBCDM+SO(\cdot) calibradas para 12 problemas das Tabelas 3 e 4



operações entre matrizes e vetores utilizadas na nossa implementação, disponíveis nos pacotes da Intel(R), podem ser executadas em paralelo, exigimos que elas utilizassem, no máximo, o mesmo número de *threads* pré-estabelecido para que o restante do programa fosse executado.

Figura 21 – *Performance profile* entre ActiveBCDM e ActiveBCDM+SO para 23 problemas das Tabelas 3 e 4

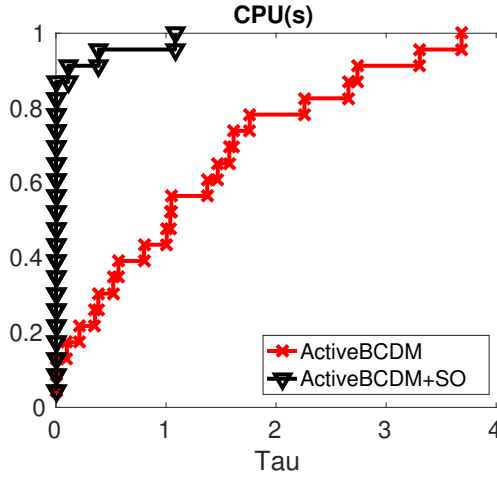
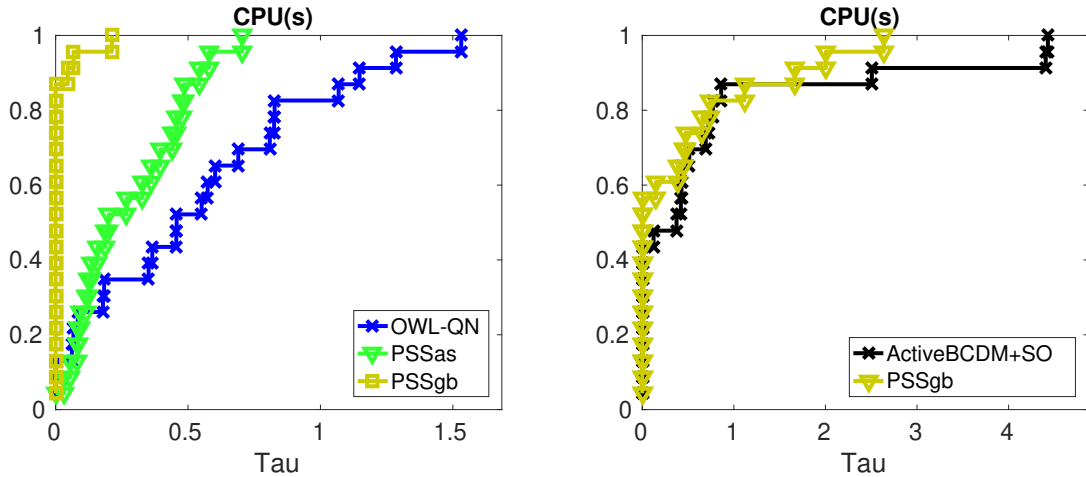


Figura 22 – *Performance profiles* entre os métodos quase-Newton (esquerda) e entre o método quase-Newton com melhor desempenho e ActiveBCDM+SO (direita) para 23 problemas das Tabelas 3 e 4



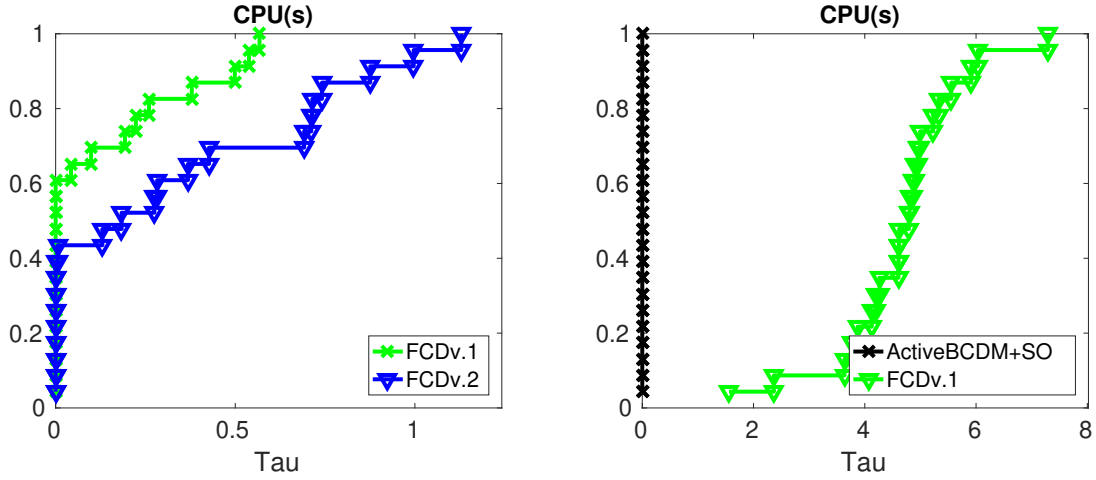
5.4.1 Escolha do conjunto \mathcal{J}

Antes de analisarmos os métodos em paralelo, repetiremos a abordagem feita na Seção 5.3 em MATLAB e estudaremos qual das duas estratégias para identificação das restrições ativas, BCDM+ST ou BCDM+IF, produz o melhor benefício computacional ao algoritmo *Active BCDM* implementado em FORTRAN.

Para fazer esses testes comparativos, usamos novamente o conjunto de 12 dados da Tabela 1 aplicados ao problema *LASSO* com variáveis não-negativas (5.10). Primeiramente, fizemos uma calibração dos parâmetros δ_{DP} e δ_F para os métodos baseados no algoritmo *Active BCDM*, BCDM+ST e BCDM+IF, e calibração do parâmetro δ_F para as variantes do método UBCDM, escolhendo o mesmo conjunto de parâmetros usados na Subseção 5.3.1.

Adotamos como critério de parada, para cada uma das variantes, o mesmo valor de função usado na Seção 5.3.1 e presente na Tabela 1, coluna F^* . Segundo esse critério de parada,

Figura 23 – *Performance profiles* entre os métodos FCDv.1 e FCDv.2 (esquerda) e entre o melhor dos métodos FCD e ActiveBCDM+SO (direita) para 23 problemas das Tabelas 3 e 4



como alguns problemas dessa classe são resolvidos em um tempo inferior ao que as rotinas de cálculo de tempo do relógio de parede conseguem captar, decidimos escolher como medida de tempo, para cada variante, o tempo total gasto para resolver cada um dos problemas 20 vezes. Apresentaremos algumas figuras contendo os *performance profiles* das variantes mais promissoras de cada um dos métodos analisados.

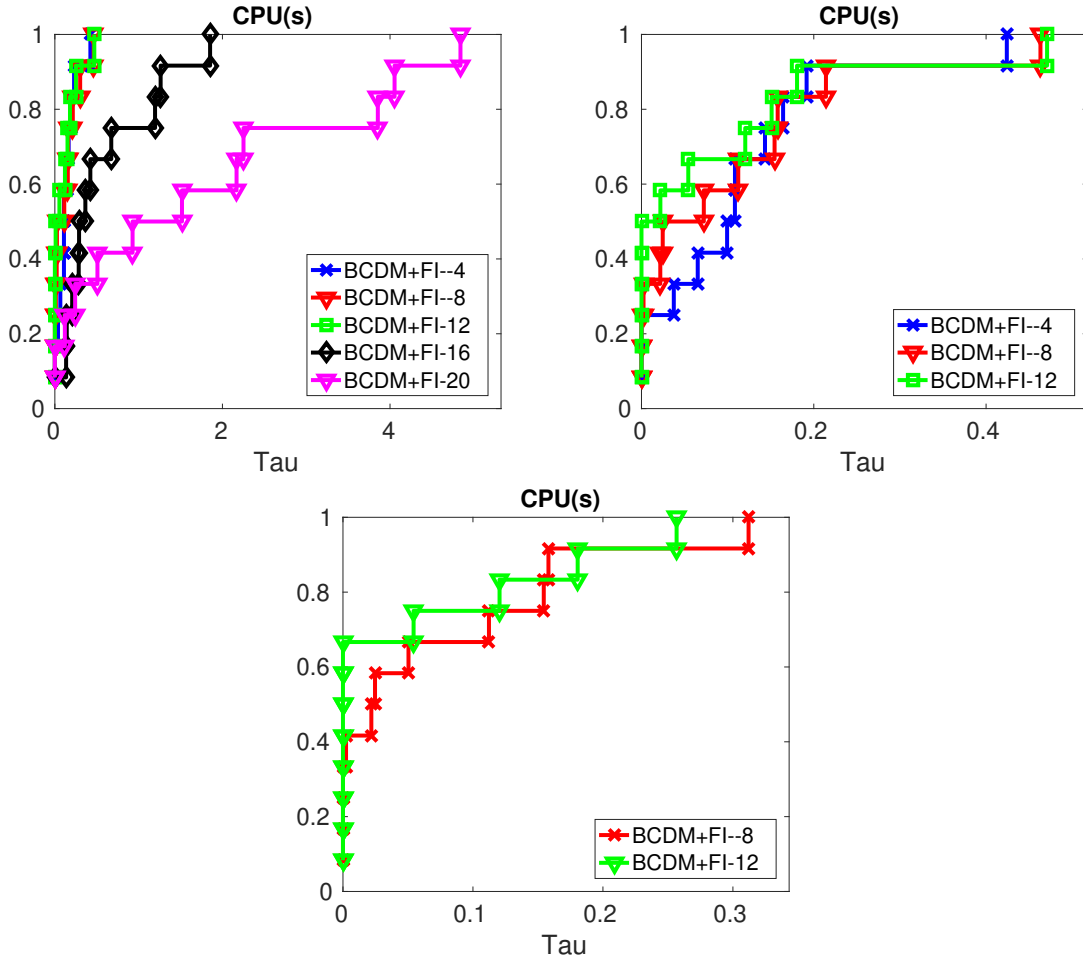
Vemos, por meio das Figuras 24, 25 e 26, as variantes que produziram os melhores resultados para os métodos BCDM+IF, BCDM+ST e UBCDM, que foram $\{\delta_{DP}, \delta_F\} = \{10, n\}$, $\{\delta_{DP}, \delta_F\} = \{100, [0.1n]\}$, $\delta_F = [0.1n]$, respectivamente.

Na sequência, comparamos as melhores variantes dos métodos BCDM+IF e BCDM+ST, usando novamente um *performance profile*, confrontando o tempo total para resolver cada problema 20 vezes (Figura 27). É evidente, por essa figura, que a melhor escolha para o conjunto \mathcal{J} é aquela fornecida pelo método BCDM+ST. De agora em diante, usaremos BCDM+ST como implementação padrão do Algoritmo 1 e a chamaremos simplesmente de **ActiveBCDM**.

Por fim, comparamos o tempo total gasto por 20 simulações da melhor variante do método UBCDM com **ActiveBCDM** no conjunto de 12 problemas. Os resultados foram graficamente descritos por meio do *performance profile* (Figura 28). Vemos que o método **ActiveBCDM** teve um desempenho superior ao método UBCDM em 90% dos problemas, mostrando um primeiro indício de como a identificação, promovida pelo método **ActiveBCDM**, pode acelerar os métodos de descenso coordenado em FORTRAN.

O método **ActiveBCDM** mostrou-se menos eficiente em FORTRAN do que sua versão em MATLAB. Não sabemos explicar essa diferença de resultados simplesmente pela mudança da linguagem de programação utilizada. Mostraremos, na sequência do texto, que o método **ActiveBCDM** tem um comportamento melhor do que o apresentado nesse primeiro experimento. Uma observação interessante é que fizemos os mesmos testes em FORTRAN em um computador diferente, processador Intel(R) Core(TM) i7-4610M CPU @ 3.00GHz e 8GB de memória RAM, e obtivemos *performance profiles* semelhantes aos da Figura 5, porém não sabemos o motivo dessa

Figura 24 – *Performance profiles* mais significativos entre as 24 variantes do método BCDM+IF para os dados da Tabela 1, em FORTRAN



divergência nos resultados para o processador Intel Xeon(R) CPU E5-2650 v2 @ 2.60GHz com 32GB de memória RAM.

5.4.2 Testes *Active PCDM*

Nesta seção, faremos alguns experimentos numéricos envolvendo métodos de descenso coordenado em paralelo, em especial, o método desenvolvido nesse texto, Algoritmo 2 (*Active PCDM*), visando responder duas perguntas:

- O paralelismo é capaz de gerar uma aceleração no método, comparado ao caso serial, como previsto pelo resultado de complexidade?
- Introduzir no método *Active PCDM* o conjunto \mathcal{J} , fixado na subseção anterior, produz alguma aceleração ao método quando comparado com a versão que faz escolhas uniformes dos blocos de coordenadas que são atualizados durante o método, correspondente à versão desenvolvida, e com resultados numéricos presentes em [37]?

Os métodos em paralelo podem ser divididos em duas classes: síncronos e assíncronos. Para o nosso contexto, os métodos síncronos são aqueles em que, a cada iteração, o cálculo dos

Figura 25 – *Performance profiles* mais significativos entre as 24 variantes do método BCDM+ST para os problemas da Tabela 1, em FORTRAN

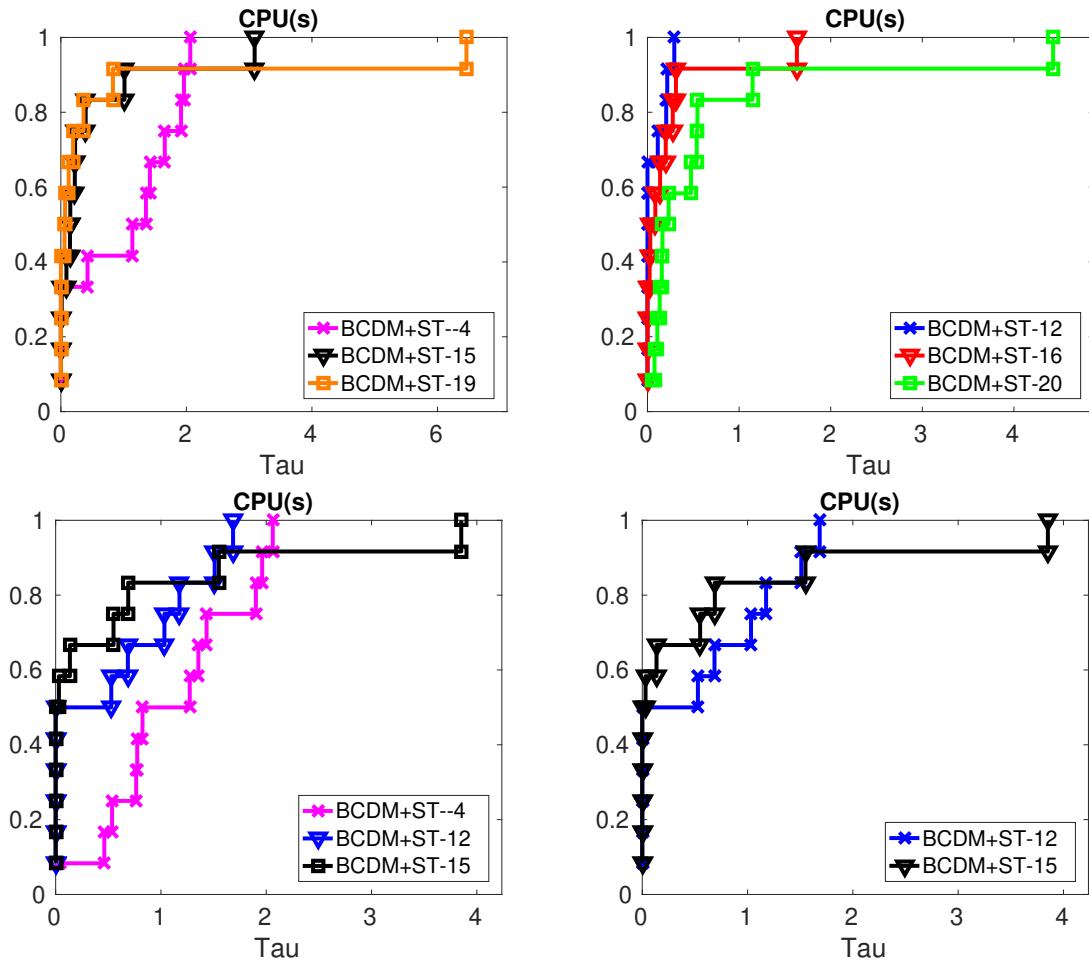
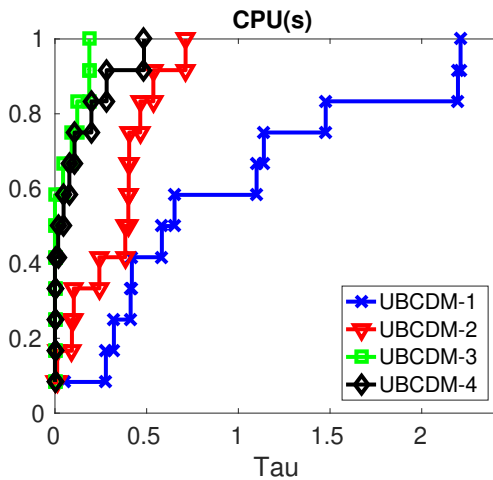


Figura 26 – *Performance profiles* mais significativos entre as 24 variantes do método UBCDM para os problemas da Tabela 1, em FORTRAN



blocos de direções de descida são divididos entre diferentes *threads* e, no final, toda a informação é sincronizada antes que seja feito o novo cálculo dos blocos de direções de descida em paralelo. No caso assíncrono, todos os blocos de coordenadas são atualizadas de maneira não sistemática,

Figura 27 – *Performance profile* entre BCDM+IF e BCDM+ST para os problemas da Tabela 1, em FORTRAN

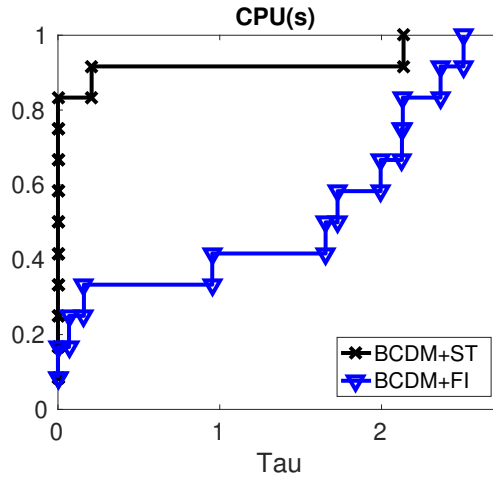
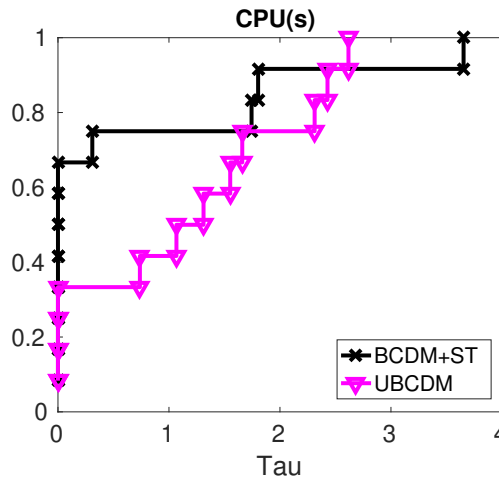


Figura 28 – *Performance profile* entre UBCDM e ActiveBCDM para os problemas da Tabela 1, em FORTRAN



quando uma *thread* está livre, ela escolhe um novo bloco de coordenadas e faz o cálculo da direção de descida com base na informação mais recente disponível na memória, sem a necessidade de esperar as outras *threads* fazerem seu trabalho.

Pela exposição desse texto, os resultados teóricos foram desenvolvidos para uma versão síncrona de método de descenso coordenado por blocos. No entanto, como descrito no *survey* [45] e constatado nos primeiros experimentos numéricos, quando tentamos aplicar nosso método nesse contexto, a versão síncrona não é capaz de apresentar alguma aceleração no método quando comparado ao caso serial. Em vista disso, faremos a mesma escolha dos autores em [37], apesar da teoria desenvolvida ser toda voltada para um método síncrono, construiremos e analisaremos uma versão assíncrona dos métodos de descenso coordenado.

Essa escolha melhorou o comportamento do método com respeito ao tempo, porém exigiu que fizéssemos uma modificação no nosso código para garantir a qualidade nas soluções produzidas pelo método em paralelo. Quando nosso método é executado de maneira assíncrona, o gradiente da parte suave da função objetivo precisa ser compartilhado entre todas as *threads*

simultaneamente. Esse vetor sofre um efeito conhecido em programação paralela como *race condition*, no qual uma variável pode ser modificada por mais de uma *thread* ao mesmo tempo, o que causa no programa um resultado imprevisível, desde que as *threads* estão competindo para efetuar as modificações.

Para diminuir esse efeito e garantir a qualidade da solução da nossa implementação dos métodos em paralelo, acrescentamos, após cada ciclo de iterações do método, o re-cálculo do gradiente da parte suave da função objetivo, baseado no ponto corrente. Com isto, conseguimos que esse vetor esteja sempre correto no início de cada ciclo, pois ele é fundamental para o nosso método, tanto por usá-lo para o cálculo das direções de descida, quanto para estabelecer o conjunto \mathcal{J} .

Esta escolha é justificada pela implementação do método PCDM1, descrita no artigo [37], que toma essa mesma decisão para o controle da qualidade da solução produzida pelo método. Nesse contexto, adicionamos este cálculo extra na versão serial do nosso método, porque não existem duas ou mais *threads* competindo pela modificação do valor na memória nesse caso, para garantir a consistência entre os programas serial e paralelo.

Nessa seção faremos todos os testes usando o problema *LASSO*, Seção 5.1, com escolha $\lambda = 0.1 \|A^T b\|_\infty$. Diferentemente da análise feita na seção de testes em MATLAB, deixaremos como trabalho futuro analisar o problema de regressão logística com regularização ℓ_1 , Seção 5.2.

Para fazer os testes comparativos com outros métodos desenvolvidos na literatura, precisamos calibrar os parâmetros δ_{DP}, δ_F do nosso método *Active PCDM*. Nesse momento tomaremos duas decisões que consideramos serem coerentes para a etapa de calibração.

Primeiramente, calibraremos apenas o valor de δ_{DP} e deixaremos o valor de $\delta_F = n$. Essa escolha é justificada, pois na etapa de testes comparativos com outros métodos da literatura, usaremos para comparação um método apresentado no artigo [37] e este tem o ciclo de iterações de tamanho n entre o passo de correção do gradiente, como discutido anteriormente. Caso calibrássemos o valor de δ_F e obtivéssemos um valor menor do que n , nosso método corrigiria mais vezes o vetor gradiente da parte suave da função, possivelmente contaminado com erros, favorecendo o valor de função e penalizando o tempo gasto, o que causaria uma difícil comparação entre os métodos, tanto do ponto de vista do tempo quanto no que se refere ao valor de função.

Segundo, faremos a calibração apenas para a versão serial do método *Active PCDM*, o qual chamaremos de **ActivePCDM-1TH**. Justificamos tal escolha pois a calibração do valor δ_{DP} envolve a capacidade de identificação das restrições ativas do método e consideramos que esse valor dificilmente mudará pelo uso de um número maior de *threads*. Além disso, caso fizéssemos a etapa de calibração com um conjunto de valores de *threads* e os resultados da calibração divergissem para algum desses valores, não faria sentido comparar o *speedup*² do método com um número variado de *threads*, pois estaríamos comparando programas diferentes.

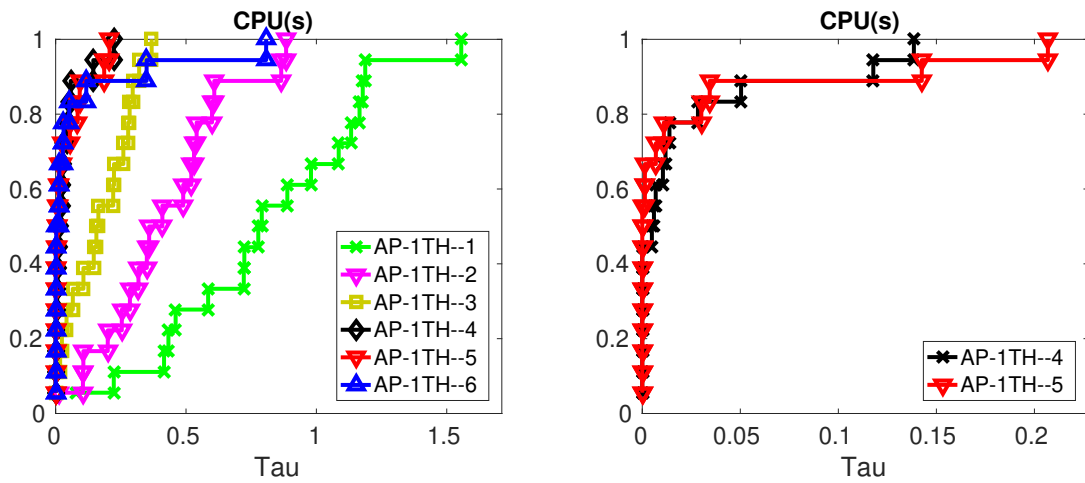
Testaremos 6 variantes do método **ActivePCDM-1TH**, cada uma delas associada ao

² *Speedup*: número que mede o desempenho relativo entre dois sistemas de processamento, ao resolver um mesmo problema.

conjunto de valores $\delta_{DP} = \{2, 5, 10, 10^2, 10^3, 10^4\}$, a função objetivo *LASSO* (Seção 5.1), usando os 18 problemas *SL1 – SL9* e *SC1 – SC9* descritos nas Tabelas 3 e 4. Como medida de tempo, usamos novamente o tempo total de parede para o método resolver cada problema 20 vezes, com critério de parada como sendo o valor de função F_{LASSO}^* , presente nas Tabelas 3 e 4.

Apresentamos *performance profiles*, Figura 29, comparando os tempos entre as 6 variantes do método **ActivePCDM-1TH**. Vemos que a variante com melhor desempenho foi aquela que resolveu mais de 90% dos problemas no menor tempo, isto é, $\delta_{DP} = 10^3$. Usaremos esse valor de δ_{DP} para os testes envolvendo o método **ActivePCDM** na sequência do texto.

Figura 29 – *Performance profiles* mais significativos entre as 6 variantes do método **ActivePCDM-1TH** para 18 problemas das Tabelas 3 e 4, em FORTRAN



5.4.2.1 Testando a qualidade da nossa implementação em paralelo

Como esse foi nosso primeiro contato com a programação em paralelo, decidimos testar a qualidade da nossa implementação dos métodos **ActivePCDM** e da versão em paralelo do método **UBCDM**, testado na Subseção 5.4.1, que chamaremos no decorrer do texto de **PUBCDM**. Escolhemos implementar esse último método, pois confrontaremos com uma implementação em C++ do mesmo método, com resultados numéricos apresentados no artigo [37], chamado nesse artigo de **PCDM1**, e disponibilizado para *download* ³.

Para realizar os testes com o método **PCDM1**, fizemos uma modificação na implementação original para facilitar a comparação dos tempos entre nossa implementação em FORTRAN e o método **PCDM1**, implementado em C++. Para citá-la, acrescentamos o cálculo do tempo envolvendo o cálculo extra do gradiente da parte suave da função objetivo e do valor de função corrente, no final de cada ciclo do método. Ele foi retirado do cálculo do tempo do relógio de parede, na versão original do código dos autores.

Visando tornar mais justa a comparação do tempo gasto pelos métodos, não acrescentaremos, nas nossas implementações em FORTRAN, o tempo gasto para calcular o valor de

³ http://www.maths.ed.ac.uk/~prichtar/i_software.html

ω e das constantes de Lipschitz por blocos, devido ao método PCDM1 não levar em consideração esses cálculos na medição do tempo gasto.

O primeiro conjunto de dados que usaremos para comparar o desempenho dos métodos são problemas gerados aleatoriamente, similares aos usados em uma parte dos experimentos envolvendo o método PCDM1 [37, Exemplo 8.1], cujo gerador foi obtido junto à implementação do método PCDM1. Com este gerador, podemos controlar a dimensão da matriz do problema e uma quantidade fixa de elementos não nulos para todas as colunas da matriz gerada. Por um lado, isso introduz uma grande artificialidade no problema, pois poucos problemas extraídos de situações reais teriam essa característica. Por outro lado, isso garante um controle no número de elementos não nulos das linhas da matriz, fundamental para um bom desempenho dos métodos de descenso coordenado em paralelo, já que o *speedup* depende do número de elementos não nulos da linha com mais elementos não nulos, a constante ω da Subseção 2.2.

Construímos 3 problemas aleatórios com dimensões próximas dos valores usados em [37, Exemplo 8.1] e descritos na Tabela 5. Nessa tabela apresentamos algumas informações relevantes sobre o problema: dimensão da matriz dos dados, número de elementos não nulos em cada uma das colunas da matriz, valor de função da solução do problema arredondada pela quarta casa, uma unidade para cima, e porcentagem de elementos nulos da solução.

Usaremos para os testes dessa seção, além dos 3 problemas sintéticos, 4 problemas reais presentes nas Tabelas 3 e 4. Escolheremos 2 de cada tabela, excluindo-se os 18 problemas usados para calibragem dos parâmetros, com o critério que esses 2 problemas sejam aqueles com maior e menor valores de ω proporcionalmente, em cada uma das tabelas. Tais problemas são: *SL17*, *SL20* e *SC20*, *SC25*.

Tabela 5 – Problemas gerados aleatoriamente

Rótulo	(s, n)	# $\neq 0$ por coluna	F_{LASSO}^*	$\text{nz}(x_{LAS}^*)$
<i>AL1</i>	$(1 \times 10^7, 2 \times 10^7)$	30	2.804×10^{13}	99.9985%
<i>AL2</i>	$(2 \times 10^7, 1 \times 10^7)$	30	6.658×10^{14}	99.9999%
<i>AL3</i>	$(2 \times 10^7, 1 \times 10^7)$	20	1.065×10^{14}	99.9998%

Para os experimentos numéricos, rodamos cada um dos 3 métodos, PCDM1, PUBCDM e ActivePCDM, a quantidade fixa de $100n$ iterações e variamos o número de *threads* dentro do conjunto de valores $\{1, 2, 4, 8\}$. Nas tabelas com os resultados dos experimentos, colocamos o tempo gasto por cada método, separando-os pelo número de *threads* e descrevemos o valor da aceleração do programa com mais de uma *thread*, quando comparado ao programa serial; essa informação aparecerá nas tabelas como *speedup*.

Tabela 6 – Desempenho dos métodos em paralelo para o problema *AL1*, com percentual $\omega/n = 0.0005\%$, para um número fixo de iterações, com 1, 2, 4, e 8 *threads*

Método	1T tempo	2T tempo	4T tempo	8T tempo	<i>speedup</i> 2T	<i>speedup</i> 4T	<i>speedup</i> 8T
PCDM1	262.10	151.47	77.14	40.86	1.73	3.39	6.35
PUBCDM	177.37	99.77	55.13	41.24	1.77	3.21	4.30
ActivePCDM	228.28	129.76	70.27	48.87	1.75	3.24	4.67

Tabela 7 – Desempenho dos métodos em paralelo para o problema *AL2*, com percentual $\omega/n = 0.00038\%$, para um número fixo de iterações, com 1, 2, 4, e 8 *threads*

Método	1T tempo	2T tempo	4T tempo	8T tempo	<i>speedup</i> 2T	<i>speedup</i> 4T	<i>speedup</i> 8T
PCDM1	114.43	77.42	39.85	22.28	1.47	2.87	5.13
PUBCDM	108.31	56.97	33.36	25.71	1.90	3.24	4.21
ActivePCDM	138.48	73.26	42.52	29.63	1.89	3.25	4.67

Tabela 8 – Desempenho dos métodos em paralelo para o problema *AL3*, com percentual $\omega/n = 0.0003\%$, para um número fixo de iterações, com 1, 2, 4, e 8 *threads*

Método	1T tempo	2T tempo	4T tempo	8T tempo	<i>speedup</i> 2T	<i>speedup</i> 4T	<i>speedup</i> 8T
PCDM1	90.07	63.42	32.95	18.51	1.42	2.73	4.86
PUBCDM	93.01	49.86	30.80	20.08	1.86	3.01	4.63
ActivePCDM	119.77	64.55	36.18	23.33	1.85	3.31	5.13

Tabela 9 – Desempenho dos métodos em paralelo para o problema *SL17*, com percentual $\omega/n = 2.86\%$, para um número fixo de iterações, com 1, 2, 4, e 8 *threads*

Método	1T tempo	2T tempo	4T tempo	8T tempo	<i>speedup</i> 2T	<i>speedup</i> 4T	<i>speedup</i> 8T
PCDM1	82.42	104.64	59.90	33.89	0.78	1.37	2.43
PUBCDM	46.14	30.74	17.65	14.77	1.50	2.61	3.12
ActivePCDM	791.27	498.42	267.82	196.23	1.58	2.95	4.03

Tabela 10 – Desempenho dos métodos em paralelo para o problema *SL20*, com percentual $\omega/n = 92.02\%$, para um número fixo de iterações, com 1, 2, 4, e 8 *threads*

Método	1T tempo	2T tempo	4T tempo	8T tempo	<i>speedup</i> 2T	<i>speedup</i> 4T	<i>speedup</i> 8T
PCDM1	39.77	47.07	17.08	10.34	0.84	2.32	3.84
PUBCDM	14.42	10.81	8.58	7.67	1.33	1.68	1.88
ActivePCDM	27.96	22.94	15.26	11.16	1.21	1.83	2.50

Tabela 11 – Desempenho dos métodos em paralelo para o problema *SC20*, com percentual $\omega/n = 0.18\%$, para um número fixo de iterações, com 1, 2, 4, e 8 *threads*

Método	1T tempo	2T tempo	4T tempo	8T tempo	<i>speedup</i> 2T	<i>speedup</i> 4T	<i>speedup</i> 8T
PCDM1	23.74	18.35	9.51	5.54	1.29	2.49	4.28
PUBCDM	24.33	14.00	7.44	4.69	1.73	3.27	5.18
ActivePCDM	33.58	17.48	10.65	6.54	1.92	3.15	5.13

Tabela 12 – Desempenho dos métodos em paralelo para o problema *SC25*, com percentual $\omega/n = 10.58\%$, para um número fixo de iterações, com 1, 2, 4, e 8 *threads*

Método	1T tempo	2T tempo	4T tempo	8T tempo	<i>speedup</i> 2T	<i>speedup</i> 4T	<i>speedup</i> 8T
PCDM1	2.86	3.22	1.97	1.24	0.88	1.45	2.30
PUBCDM	2.02	1.12	0.87	0.72	1.80	2.32	2.80
ActivePCDM	7.66	4.13	2.87	2.06	1.85	2.66	3.71

Os problemas *AL1–AL3*, *SL17*, *SC20* e *SC25* foram resolvidos até a exaustão por todos os métodos analisados e encontraram um minimizador local do problema. O método **ActivePCDM** não conseguiu encontrar o minimizador do problema *SL20* para 2, 4 e 8 *threads*, cuja precisão da solução encontrada com respeito ao valor de função foi de 15, 9 e 5 casas decimais, respectivamente. Os métodos **PCDM1** e **PUBCDM** não conseguiram resolver o problema *SL20* até a

exaustão. O método PCDM1 conseguiu resolver o problema *SL20* com uma precisão média de 7 casas decimais, com respeito ao valor de função, enquanto o método PUBCDM resolveu o mesmo problema com uma precisão média de 8 casas decimais.

Pelas Tabelas 6 a 12, vemos que em 4 dos 7 problemas, o método PCDM1 de [37] escala bem o tempo gasto pelo método com mais de uma *thread*, comparado ao tempo do método serial. Observamos um ótimo escalamento do tempo desse método quando passa de 4 para 8 *threads*, porém em 3 problemas seu desempenho para 2 *threads* foi pior do que o tempo para 1 *thread*, especialmente para *SL17*, um problema que seria, a princípio, favorável para a aceleração do método.

Nota-se que nos problemas reais nossas implementações em paralelo, PUBCDM e ActivePCDM, tiveram um desempenho, com respeito ao tempo, superior ao método PCDM1. Outro aspecto interessante foi que o valor proporcional de ω , comparado à dimensão das colunas das matrizes do problema, causou mais impacto no desempenho das nossas implementações em paralelo do que na implementação em C++.

As tabelas anteriores mostraram como a estrutura da matriz de dados pode trazer grande impacto na capacidade de escalamento do tempo entre os métodos serial e paralelo. Por exemplo, a estrutura dos problemas artificiais *AL1–AL3*, com controle na quantidade de elementos não nulos nas colunas da matriz, faz com que as colunas da matriz tenham valores bem distribuídos e garante um bom desempenho dos métodos em paralelo. Apesar desses problemas artificiais representarem uma situação ideal para a aplicação dos métodos de descenso coordenado em paralelo, eles tornam-se pouco representativos, pois sua estrutura peculiar é dificilmente encontrada em problemas extraídos de situações realísticas. Por isso, na próxima subseção de testes, trabalharemos somente com problemas cujos dados são reais.

5.4.2.2 Testes de desempenho

Nessa subseção, faremos testes envolvendo os três métodos usados na seção anterior, visando responder às duas perguntas feitas na Subseção 5.4.2. Para isso, usaremos como critério de parada o valor de função descrito nas Tabelas 3 e 4 por F_{LASSO}^* . Para cada problema, calcularemos o tempo do relógio de parede que cada método gastou para resolvê-lo 20 vezes, segundo o critério de parada. Pela aleatoriedade dos métodos, devemos resolver um mesmo problemas várias vezes para garantir uma relevância estatística dos resultados.

Para responder à primeira pergunta dessa subseção, analisaremos o desempenho dos métodos em paralelo, com respeito ao tempo gasto para encontrar o critério de parada, variando o número de *threads* no conjunto $\{1, 2, 4, 8\}$ nos mesmos 4 problemas reais usados na subseção anterior.

Apresentaremos quatro tabelas, na sequência do texto, contendo a aceleração real (\mathcal{AR}) produzida pelos resultados numéricos com respeito ao número de iterações e a aceleração do método com relação ao tempo, descrita pelas tabelas como *speedup*. Na base das tabelas, acrescentamos a aceleração esperada teoricamente, com respeito ao número de iterações, descritas para os métodos uniformes (3.29) e para o nosso método *Active PCDM* (3.30), como uma tripla

de valores, representando a aceleração esperada para 2, 4 e 8 *threads*, respectivamente. Como a expressão (3.30), depende do tamanho do conjunto \mathcal{I} e este é variável durante o método, usaremos como estimativa para ele, o valor de $|\mathcal{I}|$ que produz a melhor aceleração possível, isto é, escolheremos $|\mathcal{I}|$ como o tamanho das restrições inativas da face ótima, este valor está disponível na coluna $nz(x_{LAS}^*)$ das Tabelas 3 e 4.

Tabela 13 – Resultados da aceleração real *versus* aceleração teórica e aceleração com respeito ao tempo para o problema *SL17*.

Método	\mathcal{AR} 2T	\mathcal{AR} 4T	\mathcal{AR} 8T	<i>speedup</i> 2T	<i>speedup</i> 4T	<i>speedup</i> 8T
PCDM1	0.96	0.93	0.84	0.76	1.28	2.09
PUBCDM	0.99	0.95	0.86	1.65	1.94	3.27
ActivePCDM	1.07	1.00	1.01	0.95	1.47	2.26

$$\mathcal{AE}_{PCDM1} = (1.94, 3.68, 6.66), \mathcal{AE}_{APCDM} = (1.22, 1.37, 1.46)$$

Tabela 14 – Resultados da aceleração real *versus* aceleração teórica e aceleração com respeito ao tempo para o problema *SL20*.

Método	\mathcal{AR} 2T	\mathcal{AR} 4T	\mathcal{AR} 8T	<i>speedup</i> 2T	<i>speedup</i> 4T	<i>speedup</i> 8T
PCDM1	0.63	0.37	0.21	0.54	0.86	0.77
PUBCDM	0.50	0.26	0.13	0.69	0.45	0.26
ActivePCDM	0.60	0.31	0.15	0.77	0.53	0.39

$$\mathcal{AE}_{PCDM1} = (1.04, 1.06, 1.07), \mathcal{AE}_{APCDM} = (1.00, 1.00, 1.00)$$

Tabela 15 – Resultados da aceleração real *versus* aceleração teórica e aceleração com respeito ao tempo para o problema *SC20*.

Método	\mathcal{AR} 2T	\mathcal{AR} 4T	\mathcal{AR} 8T	<i>speedup</i> 2T	<i>speedup</i> 4T	<i>speedup</i> 8T
PCDM1	0.95	1.10	1.04	1.84	4.21	6.84
PUBCDM	1.04	1.03	1.01	1.89	2.67	2.92
ActivePCDM	0.97	0.97	0.86	1.80	2.78	4.00

$$\mathcal{AE}_{PCDM1} = (1.99, 3.97, 7.89), \mathcal{AE}_{APCDM} = (1.83, 3.13, 4.86)$$

Tabela 16 – Resultados da aceleração real *versus* aceleração teórica e aceleração com respeito ao tempo para o problema *SC25*.

Método	\mathcal{AR} 2T	\mathcal{AR} 4T	\mathcal{AR} 8T	<i>speedup</i> 2T	<i>speedup</i> 4T	<i>speedup</i> 8T
PCDM1	0.99	0.99	0.98	0.90	1.44	2.27
PUBCDM	0.90	0.77	0.59	1.54	1.57	1.78
ActivePCDM	1.00	1.00	1.00	1.23	1.07	2.48

$$\mathcal{AE}_{PCDM1} = (1.80, 3.03, 4.59), \mathcal{AE}_{APCDM} = (1.15, 1.24, 1.29)$$

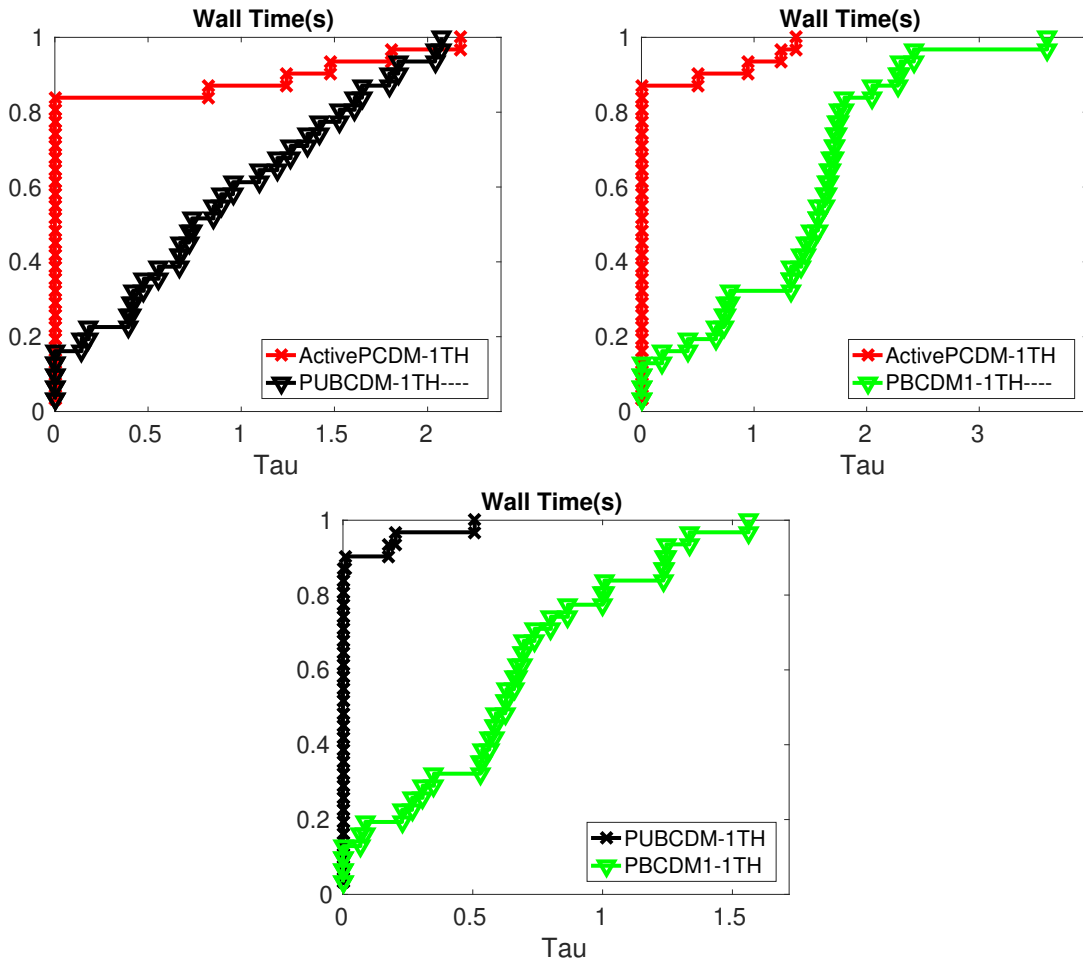
Por meio das Tabelas 13 a 16, vemos que a diferença entre a aceleração prevista teoricamente e a aceleração real, com respeito ao número de iterações, é bem diferente para todos os métodos. Esse é um resultado esperado, pois as implementações em paralelo diferem substancialmente da maneira como o método foi teoricamente previsto. Por exemplo, teoricamente ele era síncrono e na prática, torna-se assíncrono, dentre outras mudanças descritas no texto. Porém, um fato interessante é que, mesmo com o pobre escalamento, com respeito ao número

de iterações, vemos que existe uma grande aceleração dos métodos com respeito ao tempo, especialmente quando usam-se 8 *threads*. Essa aceleração deixou de acontecer em um problema, *SL20*, cujo valor de ω o classifica como severamente não aconselhável para se aplicar os métodos em paralelo.

Para o problema *SC20*, no entanto, observamos uma ótima aceleração da implementação em C++ feita pelos autores em [37], com respeito ao tempo. Nesse problema, a aceleração ficou muito próxima aos valores previstos quando comparados com a aceleração relacionada ao número de iterações \mathcal{AE}_{PCDM1} .

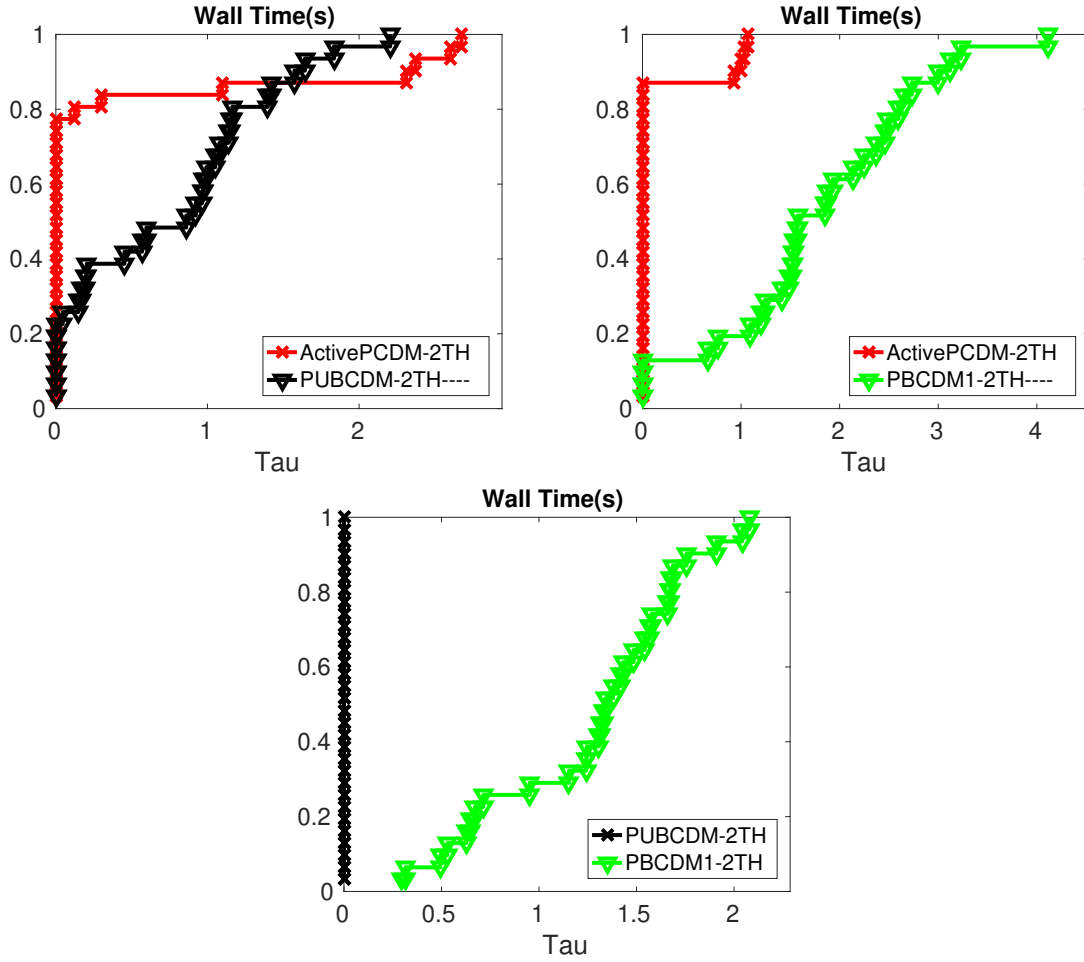
Para lidar com a segunda pergunta dessa subseção, faremos *performance profiles* envolvendo os três métodos escolhidos para análise nesse texto, PCDM1, PUBCDM e ActivePCDM, comparando o tempo gasto para resolver cada um dos 31 problemas, 10 – 24 da Tabela 3 e 10 – 25 da Tabela 4, no total de 20 vezes. Para nossa análise, variaremos também o número de *threads* usadas para execução dos métodos no conjunto de valores $\{1, 2, 4, 8\}$.

Figura 30 – *Performance profiles* entre ActivePCDM-1TH, PUBCDM-1TH e PBCDM1-1TH dois a dois para 31 problemas das Tabelas 3 e 4, em FORTRAN



Por meio das Figuras 30 a 33, podemos ver que, para os 4 valores de *threads* escolhidos, o método ActivePCDM é superior aos outros métodos em paralelo para resolver o conjunto de problemas, com o critério de parada escolhido. Notamos que o método PBCDM1 escala melhor o

Figura 31 – *Performance profiles* entre ActivePCDM-2TH, PUBCDM-2TH e PBCDM1-2TH dois a dois para 31 problemas das Tabelas 3 e 4, em FORTRAN



tempo para 4 e 8 *threads* do que nossas implementações em paralelo, fazendo com que a diferença de tempo entre os métodos em FORTRAN diminua para o método implementado em C++, para esses valores de *threads*.

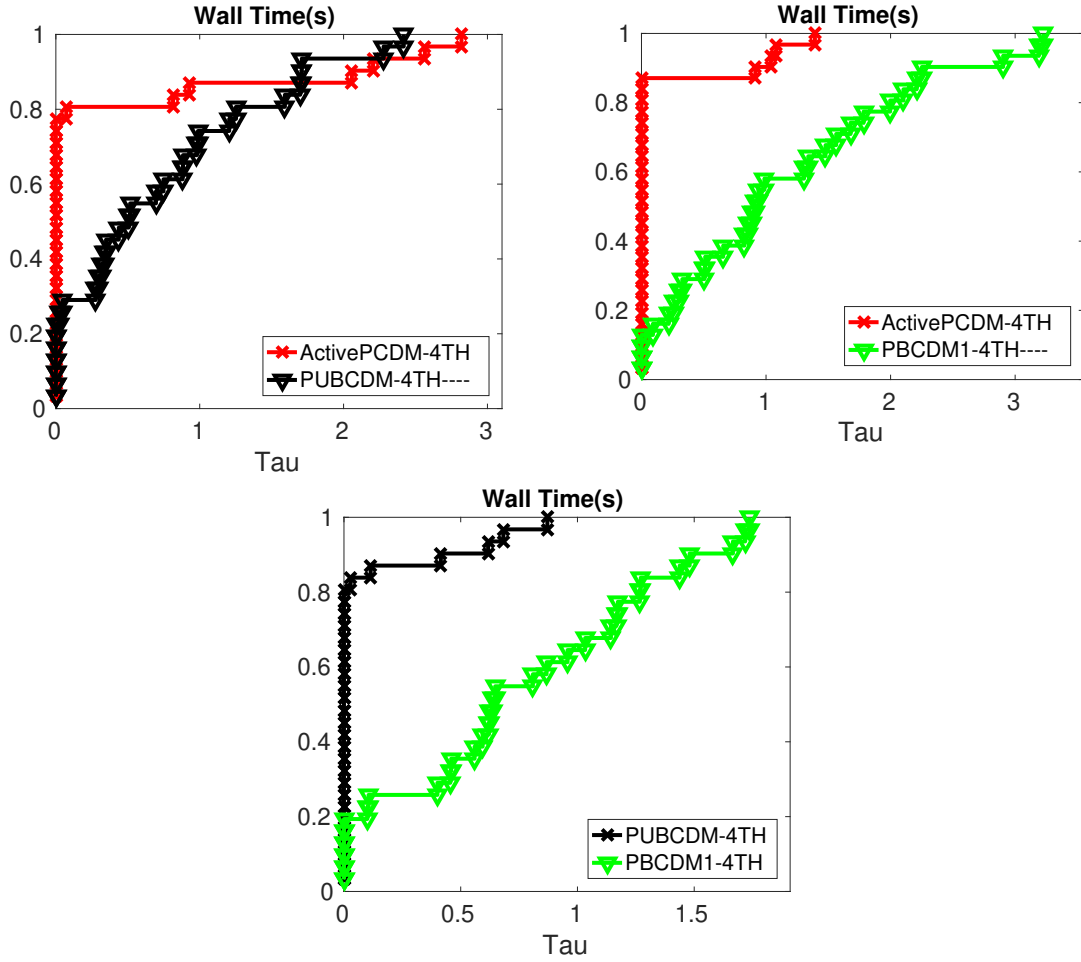
Com isso, vemos que o acréscimo da estratégia de identificação das restrições ativas no método de descenso coordenado por blocos também produz aceleração quando comparado ao método que escolhe as coordenadas de maneira uniforme, pelo menos para o problema *LASSO* com o valor de λ especificado no texto.

Para concluir, traremos mais duas figuras contendo *performance profiles* dos métodos ActivePCDM e PUBCDM, envolvendo o tempo gasto para resolver um conjunto de problemas num total de 20 vezes.

Dividiremos os testes em duas figuras, pois dependendo da estrutura da matriz, número maior de linhas (Tabela 3) ou número maior de colunas (Tabela 4), o comportamento dos métodos em paralelo muda drasticamente, principalmente, pelo fato que nossos problemas são em geral retangulares, acarretando que aqueles com mais linhas tem, na média, ω proporcionalmente maior do que os com mais colunas.

Pela Figura 34, vemos que para o método ActivePCDM, a versão serial é a de melhor

Figura 32 – *Performance profiles* entre ActivePCDM-4TH, PUBCDM-4TH e PBCDM1-4TH dois a dois para 31 problemas das Tabelas 3 e 4, em FORTRAN



desempenho nos problemas da Tabela 3. Para o método PUBCDM ficaríamos entre o método que usa 1 ou 2 *threads* como os de melhores desempenhos. Todavia, levando em conta que acrescentamos o cálculo extra para a correção do gradiente no método serial, novamente, escolheríamos como melhor método, o serial. Logo, nesse caso, o paralelismo não produz uma aceleração no tempo, no aspecto qualitativo das soluções.

Para o conjunto de problemas da Tabela 4, temos um quadro diferente, como representado na Figura 35. Nesse caso, notamos que o método que usa mais *threads* foi aquele que produziu melhores resultados com respeito ao tempo.

Nosso intuito, como trabalho futuro, é verificar se os fenômenos descritos nessa subseção, também ocorrem para o problema de regressão logística com regularização ℓ_1 .

Figura 33 – *Performance profiles* entre ActivePCDM-8TH, PUBCDM-8TH e PBCDM1-8TH dois a dois para 31 problemas das Tabelas 3 e 4, em FORTRAN

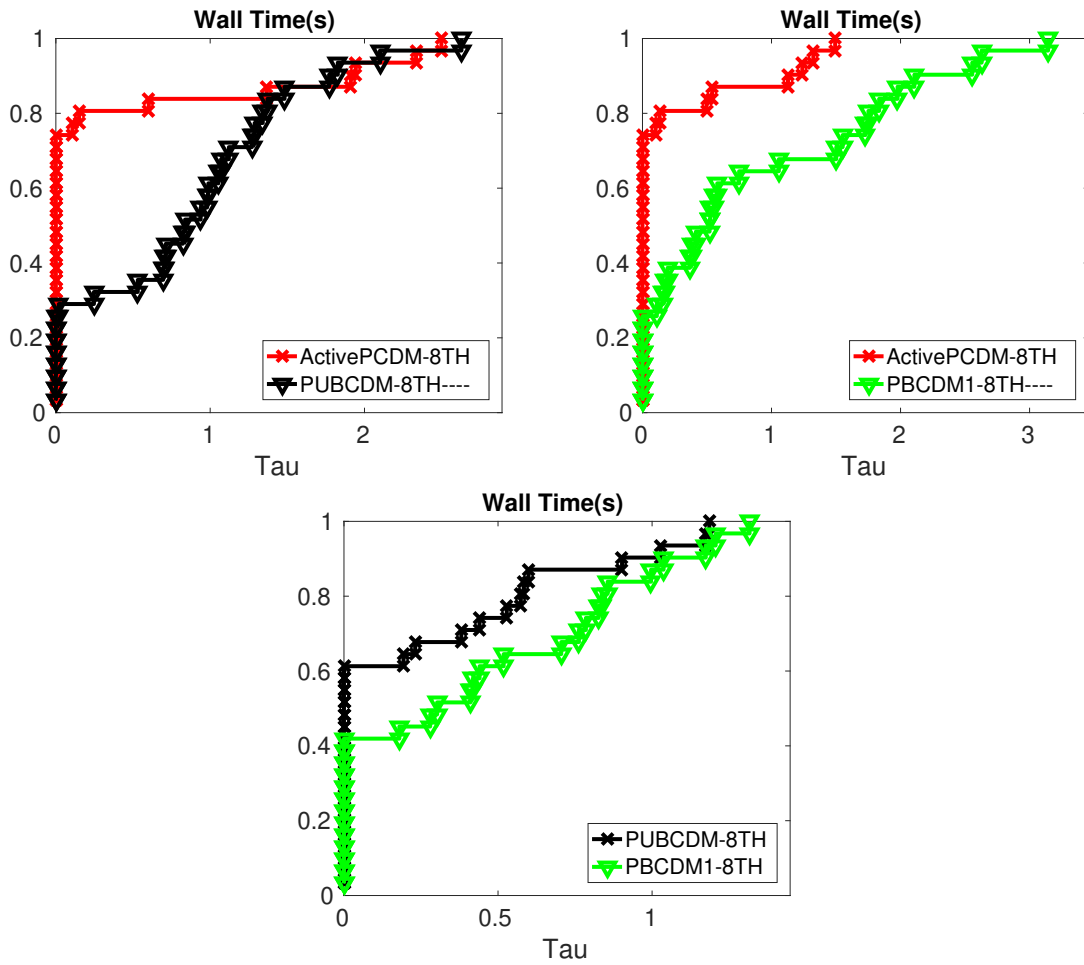


Figura 34 – *Performance profiles* do método ActivePCDM (figura à esquerda) e o método PUBCDM (figura à direita), variando o número de *threads* para 15 problemas da Tabela 3, em FORTRAN

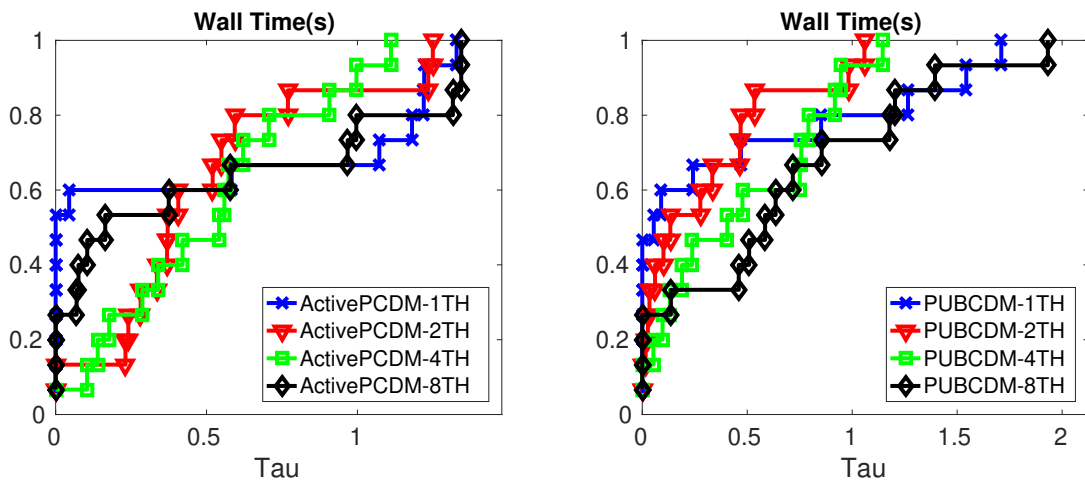
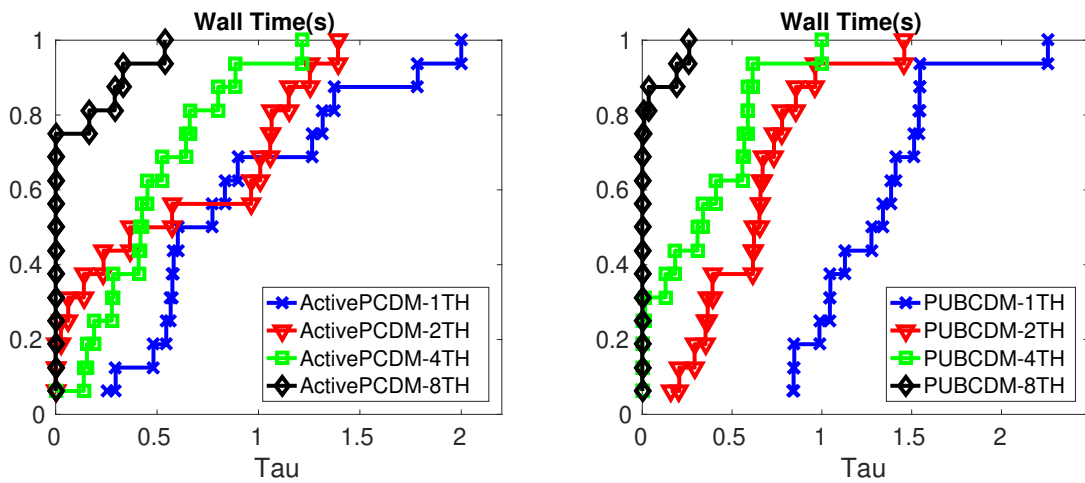


Figura 35 – *Performance profiles* do método ActivePCDM (figura à esquerda) e o método PUBCDM (figura à direita), variando o número de *threads* para 16 problemas da Tabela 4, em FORTRAN



6 Considerações Finais

Propusemos estratégias de identificação para acelerar o método de descenso coordenado por blocos estudado. Mostramos, sob certas hipóteses, que o método tem a propriedade de identificação das restrições ativas, generalizando o conceito de degeneração de um ponto estacionário, definido para funções suaves em [12], e adicionando uma hipótese sobre o comportamento de uma sequência de vetores pertencentes a subdiferencial da parte não suave da função objetivo.

Usando o conceito de função identificadora [19] e tomando como base resultados do artigo [44], criamos um exemplo de função identificadora para nosso problema de interesse, compatível com a simplicidade numérica exigida pelo método e pela proposta de aplicação a problemas de porte enorme. Uma desvantagem dessa função é a exigência de uma hipótese restritiva sobre a parte suave da função objetivo, que deve ser fortemente convexa numa vizinhança do ponto estacionário para garantia teórica do seu funcionamento.

Tendo em vista as estratégias de identificação, desenvolvemos uma versão melhorada do algoritmo BCDM, chamado de *Active BCDM*, controlada por dois parâmetros: δ_{DP} , fator de preferência dos blocos pertencentes ao conjunto de variáveis classificadas como inativas \mathcal{I} e δ_F , frequência de iterações entre a reclassificação dos blocos do conjunto \mathcal{I} . Provamos resultados de convergência global para as versões serial e paralela desse algoritmo. Até aonde vai nosso conhecimento, trouxemos uma novidade para esse tipo de método, pois nossas propostas consideram que a probabilidade na escolha dos blocos pode ser alterada a cada ciclo de iterações do algoritmo.

Analizamos duas estratégias de identificação para o algoritmo *Active BCDM* em 12 testes envolvendo o problema *LASSO* com variáveis não negativas (5.13) para definir a melhor escolha para ser usada pelo algoritmo. Vimos que a estratégia que melhor capturou as restrições ativas, classificou como ativas o conjunto de blocos de coordenadas tal que

$$\mathcal{J} \equiv \{i \mid x_i = 0 \text{ e } h_i = 0\}.$$

Esse conjunto combina duas informações relevantes, $x_i = 0$, que compreende os blocos de coordenadas classificadas como ativas por qualquer função identificadora, e $h_i = 0$, que contempla a estacionariedade local do bloco de coordenadas. Esta última, permite que classifiquemos como ativo um bloco de coordenadas mesmo sem ter sido visitado pelo método em ciclos de iterações anteriores, propriedade relevante de ser incorporada em métodos de descenso coordenado.

Nossa escolha feita sobre a identificação do algoritmo *Active BCDM*, proporcionou a sua aplicação a problemas irrestritos com regularização ℓ_1 . Essa classe de problemas possui formulação equivalente com restrições de caixa, cuja classificação das restrições ativas remete à localização das componentes esparsas de uma solução do problema.

Apresentamos uma seção de testes com o intuito de definir boas escolhas para os parâmetros (δ_F, δ_{DP}) e comparar nosso algoritmo com outros métodos da literatura. Para isso,

construímos um conjunto de 49 testes determinísticos extraídos de 7 fontes: [3, 7, 9, 14, 16, 27, 28]

Para colocar nosso algoritmo em perspectiva, confrontamos sua versão serial e uma segunda variação dele, baseada em ideias propostas no artigo [21] e chamado *Active BCDM+SO*, com outros 7 algoritmos desenvolvidos por outros autores: OWL-QN [1]; PSSas e PSSgb [39]; FAST-BCD2-E [38]; SpARSA [46]; FCDv.1 e FCDv.2 [21], todos eles implementados em MATLAB. Dentre tais métodos, os 5 primeiros foram aplicados ao problema *LASSO* e os 3 primeiros unidos aos 2 últimos para o problema de regressão logística com regularização ℓ_1 .

No contexto de programação em paralelo, comparamos, para o problema *LASSO*, nosso algoritmo de descenso coordenado com outros dois métodos de descenso coordenado que escolhem os blocos de maneira uniforme: UBCDM, PCDM1 [37], sendo o último método implementado em C++ e os outros 2 em FORTRAN.

Nos testes em MATLAB, notou-se que o desempenho do nosso algoritmo é competitivo com os melhores algoritmos analisados para resolver cada problema. Vale notar que testamos métodos que apresentaram resultados numéricos expressivos em seus artigos e alguns que usam informação de segunda ordem. Nos testes em paralelo, vimos que nosso algoritmo tem desempenho superior aos métodos de descenso coordenado que não usam informação sobre as restrições ativas e mostramos como a estrutura dos dados é determinante para a escolha entre usar, ou não, o paralelismo para resolução dos problemas.

Como trabalho futuro, pretendemos investigar os métodos em paralelo para o problema de regressão logística com regularização ℓ_1 . Outra proposta envolve a extensão dos resultados desse trabalho em um contexto mais amplo, por exemplo, para funções objetivo cuja parte suave não possui constantes de Lipschitz por blocos globais.

Referências

- [1] G. Andrew and J. Gao. Scalable training of l_1 -regularized log-linear models. *International Conference on Machine Learning*, 2007.
- [2] A. Beck and L. Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060, 2013.
- [3] E. van den Berg, M. P. Friedlander, G. Hennenfent, F. Herrmann, R. Saab, and Ö. Yılmaz. Sparco: A testing framework for sparse reconstruction. Technical Report TR-2007-20, Dept. Computer Science, University of British Columbia, Vancouver, October 2007.
- [4] D. P. Bertsekas, A. Nedi, A. E Ozdaglar, et al. *Convex analysis and optimization*. Athena Scientific, Nashua, NH, USA, 2003.
- [5] D.P. Bertsekas and J.N. Tsitsiklis. *Introduction to Probability*. Athena Scientific books. Athena Scientific, 2002.
- [6] E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, pages 1196–1211, 2000.
- [7] R. F. Boisvert, R. Pozo, K. Remington, R. F. Barrett, and J. J. Dongarra. *Matrix Market: a web resource for test matrix collections*, pages 125–137. Springer US, Boston, MA, 1997.
- [8] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [9] J. K. Bradley, A. Kyrola, D. Bickson, and C. Guestrin. Parallel coordinate descent for l_1 -regularized loss minimization. In ICML2011, editor, *Proceedings of the 28th International Conference on Machine Learning*, pages 1–8, Bellevue, Washington, USA, 2011. The International Machine Learning Society.
- [10] J. V. Burke and J. J. Moré. On the identification of active constraints. *SIAM J. Numer. Anal.*, 25(5):1197–1211, 1988.
- [11] J. V. Burke and J. J Moré. Exposing constraints. *SIAM Journal on Optimization*, 4(3):573–595, 1994.
- [12] P. H. Calamai and J. J. Moré. Projected gradient methods for linearly constrained problems. *Mathematical Programming*, 39(1):93–116, 1987.
- [13] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.
- [14] C. Chang and C. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

- [15] A. Daneshmand, F. Facchinei, V. Kungurtsev, and G. Scutari. Hybrid random/deterministic parallel algorithms for nonconvex big data optimization. *CoRR*, abs/1407.4504, 2014.
- [16] T. A. Davis and Y. Hu. The university of florida sparse matrix collection. *ACM Trans. Math. Softw.*, 38(1):1:1–1:25, December 2011.
- [17] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.
- [18] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.
- [19] F. Facchinei, A. Fischer, and C. Kanzow. On the accurate identification of active constraints. *SIAM Journal on Optimization*, 9(1):14–32, 1998.
- [20] S. D. Flåm. On finite convergence and constraint identification of subgradient projection methods. *Mathematical Programming*, 57(1):427–437, 1992.
- [21] K. Fountoulakis and R. Tappenden. A flexible coordinate descent method. <https://arxiv.org/abs/1507.03713>, 2017. Online; accessed 01-September-2017.
- [22] A. Friedlander, J. M. Martínez, and S. A. Santos. A new trust region algorithm for bound constrained minimization. *Applied Mathematics and Optimization*, 30(3):235–266, 1994.
- [23] W. L. Hare and A. S. Lewis. Identifying active manifolds. *Algorithmic Operations Research*, 2(2):75, 2007.
- [24] D. Kim, S. Sra, and I. S. Dhillon. A non-monotonic method for large-scale non-negative least squares. *Optimization Methods Software*, 28(5):1012–1039, October 2013.
- [25] J. Kim, N. Ramakrishnan, M. Marwah, A. Shah, and H. Park. Regularization paths for sparse nonnegative least squares problems with applications to life cycle assessment tree discovery. In *2013 IEEE 13th International Conference on Data Mining*, pages 360–369, 2013.
- [26] K. Koh, S. Kim, and S. Boyd. An interior-point method for large-scale l1-regularized logistic regression. *J. Mach. Learn. Res.*, 8:1519–1555, December 2007.
- [27] P. Komarek. Paul komarek’s webpage. <http://komarix.org/ac/ds/>. Accessed: 2017-01-29.
- [28] M. Lichman. UCI machine learning repository. <http://archive.ics.uci.edu/ml>, 2013.
- [29] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. The MIT Press, Cambridge, USA, 2012.
- [30] A. Muniategui, R. N.-Cadenas, Mi. Vázquez, X. L. Aranguren, X. Agirre, A. Luttun, F. Prosper, A. Pascual-Montano, and A. Rubio. Quantification of miRNA-mRNA interactions. *PLoS ONE*, 7(2):1–10, 2012.

- [31] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.
- [32] Y. E. Nesterov. *Introductory lectures on convex optimization: A basic course*. Springer Publishing Company, Incorporated, 2014.
- [33] A. Y. Ng. Feature selection, l_1 vs. l_2 regularization and rotational invariance. In *Proceedings of the 21st International Conference on Machine Learning*, pages 78–86, 2004.
- [34] N. Parikh and S. P. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2014.
- [35] P. Richtárik and M. Takáč. *Efficient Serial and Parallel Coordinate Descent Methods for Huge-Scale Truss Topology Design*, pages 27–32. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [36] P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1):1–38, 2014.
- [37] P. Richtárik and M. Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1):433–484, 2016.
- [38] M. De Santis, S. Lucidi, and F. Rinaldi. A fast active set block coordinate descent algorithm for ℓ_1 -regularized least squares. *SIAM Journal on Optimization*, 26(1):781–809, 2016.
- [39] M. Schmidt. *Graphical Model Structure Learning with L1-Regularization*. PhD thesis, University of British Columbia, 2010.
- [40] M. Slawski. Problem-specific analysis of non-negative least squares solvers with a focus on instances with sparse solutions (working paper). <https://sites.google.com/site/slawskimartin/publications>, 2013. accessed 01-September-2017.
- [41] R. Tappenden, P. Richtárik, and J. Gondzio. Inexact coordinate descent: Complexity and preconditioning. *Journal of Optimization Theory and Applications*, 170(1):144–176, 2016.
- [42] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996’, publisher=’JSTOR.
- [43] P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1):387–423, 2009.
- [44] Z. Wen, W. Yin, H. Zhang, and D. Goldfarb. On the convergence of an active-set method for l_1 minimization. *Optimization Methods and Software*, 27(6):1127–1146, 2012.
- [45] S. J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.
- [46] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, 2009.

-
- [47] L. Wu and Y. Yang. Nonnegative elastic net and application in index tracking. *Applied Mathematics and Computation*, 227:541 – 552, 2014.
 - [48] L. Wu, Y. Yang, and H. Liu. Nonnegative-lasso and application in index tracking. *Computational Statistics & Data Analysis*, 70:116 – 126, 2014.
 - [49] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.
 - [50] E. H. Zarantonello. Projections on convex sets in Hilbert space and spectral theory. In Eduardo H. Zarantonello, editor, *Contributions to Nonlinear Functional Analysis*, pages 237–424. Academic Press, 1971.

Apêndices

APÊNDICE A – Resultados sobre Probabilidade

Nesse Apêndice, descreveremos alguns resultados sobre Probabilidade que serão utilizados durante o texto, elaborado com base em [5].

Definição A.1. *Seja X uma variável aleatória discreta que assume valores no conjunto discreto $\{x_1, x_2, \dots, x_n\}$, com probabilidades dadas por $\mathbb{P}(X = x_i)$, $i \in \{1, \dots, n\}$, em que $\mathbb{P}(X = x_i)$ descreve a probabilidade da variável X ser igual a x_i . Definiremos o valor esperado de X , denotado por $\mathbb{E}[X]$, como sendo*

$$\mathbb{E}[X] = \sum_{i=1}^n x_i \mathbb{P}(X = x_i).$$

Definição A.2. *Sejam X, Y variáveis aleatórias discretas que assumem valores no conjunto discreto $\{x_1, x_2, \dots, x_n\}$, com probabilidades dadas por $\mathbb{P}(X = x_i), \mathbb{P}(Y = x_i)$, $i \in \{1, \dots, n\}$. Definiremos*

$$\mathbb{P}[X = x_i, Y = x_j] = \mathbb{P}[X = x_i \text{ e } Y = x_j], \quad \forall i, j \in \{1, \dots, n\},$$

$$\mathbb{P}(X = x_i) = \sum_{j=1}^n \mathbb{P}[X = x_i, Y = x_j]$$

e

$$\mathbb{P}(Y = x_i) = \sum_{j=1}^n \mathbb{P}[X = x_j, Y = x_i].$$

Definição A.3. *Seja X uma variável aleatória discreta que assume valores no conjunto discreto $\mathcal{A} = \{x_1, x_2, \dots, x_n\}$ com probabilidades dadas por $\mathbb{P}(X = x_i)$, $i \in \{1, \dots, n\}$. Considere uma função $\phi : \mathcal{A} \subset \mathcal{D}_1 \rightarrow \mathbb{R}$ qualquer. Definiremos a variável aleatória $Y = \phi(X)$ que assume valores no conjunto $\{\phi(x_1), \phi(x_2), \dots, \phi(x_n)\}$ da forma*

$$\mathbb{P}(Y = y_i) = \sum_{\{j \mid \phi(x_j) = y_i\}} \mathbb{P}[X = x_j],$$

onde $\{y_1, y_2, \dots, y_m\} \subset \{\phi(x_1), \phi(x_2), \dots, \phi(x_n)\}$.

Definição A.4. *Sejam X, Y variáveis aleatórias discretas que assumem valores no conjunto discreto $\mathcal{A} = \{x_1, x_2, \dots, x_n\}$ com probabilidades dadas por $\mathbb{P}(X = x_i), \mathbb{P}(Y = x_i)$, $i \in \{1, \dots, n\}$. Considere uma função $\Phi : \mathcal{A} \times \mathcal{A} \subset \mathcal{D}_1 \times \mathcal{D}_2 \rightarrow \mathbb{R}$ qualquer. Definiremos a variável aleatória $Z = \Phi(X, Y)$ que assume valores no conjunto*

$$\mathcal{D}_3 = \{\Phi(x_1, x_1), \Phi(x_1, x_2), \dots, \Phi(x_1, x_n), \dots, \Phi(x_n, x_1), \dots, \Phi(x_n, x_n)\}$$

da forma

$$\mathbb{P}(Z = z_i) = \sum_{\{(i,j) \mid \Phi(x_i, x_j) = z_i\}} \mathbb{P}[X = x_i, Y = x_j],$$

onde $\{z_1, z_2, \dots, z_{n^2}\} \subset \mathcal{D}_3$.

A seguir, traremos alguns resultados bem conhecidos sobre o valor esperado.

Lema A.1. *Sejam X, W variáveis aleatórias que assumem valores no conjunto discreto $\mathcal{A} = \{x_1, x_2, \dots, x_n\}$, $\phi : \mathcal{A} \subset \mathcal{D}_1 \rightarrow \mathbb{R}$, a variável aleatória $Y = \phi(X)$, $\Phi : \mathcal{A} \times \mathcal{A} \subset \mathcal{D}_1 \times \mathcal{D}_2 \rightarrow \mathbb{R}$, a variável aleatória $Z = \Phi(X, W)$ satisfazendo as Definições A.3 e A.4 e $a, b, c \in \mathbb{R}$ constantes. Então:*

- (i) $\mathbb{E}[\phi(X)] = \sum_{i=1}^n \phi(x_i) \mathbb{P}(X = x_i);$
- (ii) $\mathbb{E}[\Phi(X, W)] = \sum_{i=1}^n \sum_{j=1}^n \Phi(x_i, x_j) \mathbb{P}(X = x_i, W = x_j);$
- (iii) $\mathbb{E}[aX + bW + c] = a\mathbb{E}[X] + b\mathbb{E}[W] + c;$
- (iv) Se $X \geq 0$, então $\mathbb{E}[X] \geq 0;$
- (v) Se $X \geq W$, então $\mathbb{E}[X] \geq \mathbb{E}[W].$

Prova: (i) Usando as Definições A.1 e A.3 e a notação $Y = \phi(X)$, vemos que

$$\begin{aligned}
 \mathbb{E}[\phi(X)] &= \mathbb{E}[Y] \\
 &\stackrel{\text{Def. A.1}}{=} \sum_{j=1}^m y_j \mathbb{P}(Y = y_j) \\
 &\stackrel{\text{Def. A.3}}{=} \sum_{j=1}^m y_j \sum_{\{i \mid \phi(x_i) = y_j\}} \mathbb{P}(X = x_i) \\
 &= \sum_{j=1}^m \sum_{\{i \mid \phi(x_i) = y_j\}} y_j \mathbb{P}(X = x_i) \\
 &= \sum_{j=1}^m \sum_{\{i \mid \phi(x_i) = y_j\}} \phi(x_i) \mathbb{P}(X = x_i) \\
 &= \sum_{i=1}^n \phi(x_i) \mathbb{P}(X = x_i).
 \end{aligned}$$

(ii) Dada a função $\Phi(x, w)$, obtemos que

$$\begin{aligned}
 \mathbb{E}[\Phi(X, W)] &= \mathbb{E}[Z] \\
 &\stackrel{\text{Def. A.1}}{=} \sum_{k=1}^s z_k \mathbb{P}(Z = z_k) \\
 &\stackrel{\text{Def. A.4}}{=} \sum_{k=1}^s \sum_{\{(i,j) \mid \Phi(x_i, x_j) = z_k\}} z_k \mathbb{P}(X = x_i, W = x_j) \\
 &= \sum_{k=1}^s \sum_{\{(i,j) \mid \Phi(x_i, x_j) = z_k\}} \Phi(x_i, x_j) \mathbb{P}(X = x_i, W = x_j) \\
 &= \sum_{i=1}^n \sum_{j=1}^n \Phi(x_i, x_j) \mathbb{P}(X = x_i, W = x_j).
 \end{aligned}$$

(iii) Considerando $\Phi(x, w) = ax + bw + c$ temos que

$$\begin{aligned}
 \mathbb{E}[aX + bW + c] &= \mathbb{E}[\Phi(X, W)] \\
 &\stackrel{\text{Lem. A.1(ii)}}{=} \sum_{i=1}^n \sum_{j=1}^n \Phi(x_i, x_j) \mathbb{P}(X = x_i, W = x_j). \\
 &= a \sum_{i=1}^n \sum_{j=1}^n x_i \mathbb{P}(X = x_i, W = x_j) + b \sum_{i=1}^n \sum_{j=1}^n x_j \mathbb{P}(X = x_i, W = x_j) + \\
 &\quad + c \sum_{i=1}^n \sum_{j=1}^n \mathbb{P}(X = x_i, W = x_j) \\
 &= a \sum_{i=1}^n x_i \sum_{j=1}^n \mathbb{P}(X = x_i, W = x_j) + b \sum_{j=1}^n x_j \sum_{i=1}^n \mathbb{P}(X = x_i, W = x_j) + \\
 &\quad + c \\
 &\stackrel{\text{Def. A.2}}{=} a \sum_{i=1}^n x_i \mathbb{P}(X = x_i) + b \sum_{j=1}^n x_j \mathbb{P}(W = x_j) + c \\
 &\stackrel{\text{Def. A.1}}{=} a\mathbb{E}[X] + b\mathbb{E}[W] + c
 \end{aligned}$$

(iv) Esse item é óbvio pelo fato que, se $X \geq 0$, então $x_i \geq 0$, para todo $x_i \in X$, e usando a Definição A.1.

(v) O último item é consequência imediata dos itens (ii) e (iii). Desde que, se $X \geq W$, isso é equivalente a dizer que a variável aleatória $Z = \Phi(X, W) = X - W$ satisfaz $\Phi(x_i, x_j) \geq 0$, para todo $x_i \in X$ e $x_j \in W$. Portanto, pelos itens (ii) e (iii) e pela Definição A.1 vemos que

$$0 \leq \mathbb{E}[\Phi(X, W)] \stackrel{\text{Lema A.1(ii),(iii)}}{=} \mathbb{E}[X] - \mathbb{E}[W].$$

□

Utilizando os resultados anteriores, trabalharemos com uma variável aleatória, $\hat{\mathcal{S}}$, que assume valores no conjunto $[p]$, como definido na Lista de Símbolos. Quando trabalhamos com esse tipo de variável aleatória, podemos descrevê-la por meio de duas distribuições de probabilidades equivalentes.

Na primeira, a distribuição de probabilidade do conjunto $\hat{\mathcal{S}}$ é igual para cada um dos conjuntos pertencentes a $[p]$. Essa distribuição será denotada pela expressão $\mathbb{P}(\hat{\mathcal{S}} = \mathcal{S})$. A segunda é uma distribuição de probabilidade que atribui valores com respeito aos índices $i \in \{1, \dots, p\}$. Essa situação será descrita por $\mathbb{P}(i \in \hat{\mathcal{S}})$.

Apresentaremos na sequência, um lema que usará a definição do valor esperado de uma variável aleatória discreta dado pela Definição A.1 quando analisamos uma soma de valores reais sobre uma variável aleatória que representa um conjunto de índices. Esse resultado foi retirado de [37, Lema 3].

Lema A.2. *Seja $\emptyset \neq J \subset [p]$ um conjunto fixado e seja $\hat{\mathcal{S}}$ uma variável aleatória discreta assumindo valores em $[p]$. Se θ_i, θ_{ij} , com $i, j \in \{1, \dots, p\}$ são constantes reais, então:*

$$(i) \mathbb{E} \left[\sum_{i \in J \cap \hat{\mathcal{S}}} \theta_i \right] = \sum_{i \in J} \theta_i \mathbb{P}(i \in \hat{\mathcal{S}});$$

$$(ii) \mathbb{E} \left[\sum_{i \in J \cap \hat{\mathcal{S}}} \theta_i \mid |J \cap \hat{\mathcal{S}}| = k \right] = \sum_{i \in J} \theta_i \mathbb{P}(i \in \hat{\mathcal{S}} \mid |J \cap \hat{\mathcal{S}}| = k);$$

(iii)

$$\mathbb{E} \left[\sum_{i \in J \cap \hat{\mathcal{S}}} \sum_{j \in J \cap \hat{\mathcal{S}}} \theta_{ij} \right] = \sum_{i \in J} \sum_{j \in J} \theta_{ij} \mathbb{P}(i \in \hat{\mathcal{S}}, j \in \hat{\mathcal{S}}),$$

em que vamos definir $\mathbb{P}(i \in \hat{\mathcal{S}}, j \in \hat{\mathcal{S}}) = \sum_{\mathcal{S}: \{i, j\} \subset \mathcal{S}} \mathbb{P}(\hat{\mathcal{S}} = \mathcal{S})$.

Prova: (i) Pela Definição A.1, sabemos que

$$\begin{aligned} \mathbb{E} \left[\sum_{i \in J \cap \hat{\mathcal{S}}} \theta_i \right] &= \sum_{\mathcal{S} \subset [n]} \left(\sum_{i \in J \cap \mathcal{S}} \theta_i \right) \mathbb{P}(\hat{\mathcal{S}} = \mathcal{S}) \\ &= \sum_{i \in J \cap \mathcal{S}} \left(\sum_{\mathcal{S} \subset [n]} \theta_i \mathbb{P}(\hat{\mathcal{S}} = \mathcal{S}) \right) \\ &= \sum_{i \in J} \left(\sum_{\mathcal{S}: i \in \mathcal{S}} \theta_i \mathbb{P}(\hat{\mathcal{S}} = \mathcal{S}) \right) \\ &= \sum_{i \in J} \theta_i \left(\sum_{\mathcal{S}: i \in \mathcal{S}} \mathbb{P}(\hat{\mathcal{S}} = \mathcal{S}) \right) \\ &= \sum_{i \in J} \theta_i \mathbb{P}(i \in \hat{\mathcal{S}}) \end{aligned}$$

(ii) Usando argumentos similares aos do item (i), temos que

$$\begin{aligned} \mathbb{E} \left[\sum_{i \in J \cap \hat{\mathcal{S}}} \theta_i \mid |J \cap \hat{\mathcal{S}}| = k \right] &= \sum_{\mathcal{S} \subset [n]} \left(\sum_{i \in J \cap \mathcal{S}} \theta_i \right) \mathbb{P}(\hat{\mathcal{S}} = \mathcal{S} \mid |J \cap \hat{\mathcal{S}}| = k) \\ &= \sum_{i \in J \cap \mathcal{S}} \theta_i \mathbb{P}(i \in \hat{\mathcal{S}} \mid |J \cap \hat{\mathcal{S}}| = k) \end{aligned}$$

(iii) Usando argumentos similares aos do item (i), vemos que

$$\begin{aligned}
 \mathbb{E} \left[\sum_{i \in J \cap \hat{\mathcal{S}}} \sum_{j \in J \cap \hat{\mathcal{S}}} \theta_{ij} \right] &= \sum_{\mathcal{S} \subset [n]} \left(\sum_{i \in J \cap \mathcal{S}} \sum_{j \in J \cap \mathcal{S}} \theta_{ij} \right) \mathbb{P}(\hat{\mathcal{S}} = \mathcal{S}) \\
 &= \sum_{i \in J \cap \mathcal{S}} \sum_{j \in J \cap \mathcal{S}} \left(\sum_{\mathcal{S} \subset [n]} \theta_{ij} \mathbb{P}(\hat{\mathcal{S}} = \mathcal{S}) \right) \\
 &= \sum_{i \in J} \sum_{j \in J} \left(\sum_{\mathcal{S}: \{i,j\} \subset \mathcal{S}} \theta_{ij} \mathbb{P}(\hat{\mathcal{S}} = \mathcal{S}) \right) \\
 &= \sum_{i \in J} \sum_{j \in J} \theta_{ij} \left(\sum_{\mathcal{S}: \{i,j\} \subset \mathcal{S}} \mathbb{P}(\hat{\mathcal{S}} = \mathcal{S}) \right) \\
 &= \sum_{i \in J} \sum_{j \in J} \theta_{ij} \mathbb{P}(i \in \hat{\mathcal{S}}, j \in \hat{\mathcal{S}})
 \end{aligned}$$

□

Apresentaremos outro resultado, que pode ser facilmente mostrado usando o Lema A.2 e representa parcialmente o Teorema 4 de [37].

Proposição A.1. *Seja $\emptyset \neq J \subset [p]$ um conjunto fixado e seja $\hat{\mathcal{S}}$ uma variável aleatória discreta assumindo valores em $[p]$.*

$$\begin{aligned}
 (i) \quad \mathbb{E} \left[|J \cap \hat{\mathcal{S}}| \right] &= \sum_{i \in J} \mathbb{P}(i \in \hat{\mathcal{S}}); \\
 (ii) \quad \mathbb{E} \left[|J \cap \hat{\mathcal{S}}|^2 \right] &= \sum_{i \in J} \sum_{j \in J} \mathbb{P}(i \in \hat{\mathcal{S}}, j \in \hat{\mathcal{S}}).
 \end{aligned}$$

Prova: (i) Esse item, segue do fato que

$$|J \cap \hat{\mathcal{S}}| = \sum_{i \in J \cap \hat{\mathcal{S}}} 1. \quad (\text{A.1})$$

Usando a expressão A.1, vemos que

$$\begin{aligned}
 \mathbb{E} \left[|J \cap \hat{\mathcal{S}}| \right] &= \mathbb{E} \left[\sum_{i \in J \cap \hat{\mathcal{S}}} 1 \right] \\
 &\stackrel{\text{Lema A.2(i)}}{=} \sum_{i \in J} \mathbb{P}(i \in \hat{\mathcal{S}})
 \end{aligned}$$

(ii) A prova, vem do fato que

$$|J \cap \hat{\mathcal{S}}|^2 = \sum_{i \in J \cap \hat{\mathcal{S}}} \sum_{j \in J \cap \hat{\mathcal{S}}} 1. \quad (\text{A.2})$$

Usando a expressão A.2, vemos que

$$\begin{aligned} \mathbb{E} \left[|J \cap \hat{\mathcal{S}}|^2 \right] &= \mathbb{E} \left[\sum_{i \in J \cap \hat{\mathcal{S}}} \sum_{j \in J \cap \hat{\mathcal{S}}} 1 \right] \\ &\stackrel{\text{Lema A.2(ii)}}{=} \sum_{i \in J} \sum_{j \in J} \mathbb{P}(i \in \hat{\mathcal{S}}, j \in \hat{\mathcal{S}}) \end{aligned}$$

□