



UNIVERSIDADE FEDERAL DO MARANHÃO
UNIVERSIDADE FEDERAL DO PIAUÍ
Doutorado em Ciência da Computação Associação
UFMA/UFPI

Pedro Henrique Alves Ribeiro

**Reconstrução de imagem com nmBDCA para modelos
não convexos**

Orientador: Prof. Dr. João Carlos de Oliveira Souza

Teresina - PI
fevereiro, 2025

Pedro Henrique Alves Ribeiro

**Reconstrução de imagem com nmBDCA para modelos
não convexos**

TESE DE DOUTORADO

Tese apresentada ao Programa de Doutorado em Ciência da Computação, Associação UFMA/UFPI, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Orientador: Prof. Dr. João Carlos de Oliveira Souza

Teresina - PI
fevereiro, 2025

FICHA CATALOGRÁFICA
Universidade Federal do Piauí
Biblioteca Comunitária Jornalista Carlos Castello Branco
Serviço de Processos Técnicos

R484r Ribeiro, Pedro Henrique Alves.
Reconstrução de imagem com nmBDCA para modelos não
convexos / Pedro Henrique Alves Ribeiro. – 2025.
109 f. : il.

Tese (Doutorado) – Universidade Federal do Maranhão,
Universidade Federal do Piauí, Pós-Graduação em Ciência da
Computação, Teresina, 2025.

“Orientador: Prof. Dr. João Carlos de Oliveira Souza.”

1. Otimização não convexa. 2. Remoção de ruído. 3. Total
Variation. 4. DCA. 5. BDCA. 6. nmBDCA. I. Souza, João Carlos de
Oliveira. II. Título.

CDD 005.1

Bibliotecário: Géσιο dos Santos Barros - CRB3/1469

Pedro Henrique Alves Ribeiro

Reconstrução de imagem com nmBDCA para modelos não convexos

Tese apresentada ao Programa de Doutorado em Ciência da Computação, Associação UFMA/UFPI, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Aprovada pela banca examinadora em 26/02/2025

Prof. Dr. João Carlos de Oliveira Souza
Orientador
Universidade Federal do Piauí

Prof(a). Dra. Kelly Cristina Poldi
Universidade de Campinas - UNICAMP

Prof. Dr. Orizon Pereira Ferreira
Universidade Federal de Goiás - UFG

Prof. Dr. Paulo Sérgio Marques dos Santos
Universidade Federal do Delta do Parnaíba - UFDPAr

Prof. Dr. Pedro Jorge Sousa dos Santos
Universidade Federal do Delta do Parnaíba - UFDPAr

Prof. Dr. Ricardo de Andrade Lira Rabêlo
Universidade Federal do Piauí - UFPI

Certificamos que esta é a versão original e final da Tese de Doutorado que foi julgada adequada para obtenção do título de Doutor em Ciência da Computação.

Prof. Dr. João Carlos de Oliveira Souza
Orientador

Prof. Dr. André Castelo Branco Soares
Coordenador

Teresina - PI, 26 de fevereiro de 2025

A Deus, pela força e proteção. À minha família, razão de onde estou. À minha amada esposa, Amanda, pela parceria eterna em todas as nossas lidas.

Agradecimentos

Agradeço a Deus, por sempre me fortalecer na busca dos meus objetivos, e pelo amparo nas horas difíceis.

À minha família, alicerce de tudo que faço.

À minha esposa, parceira de todas as horas.

Ao meu Orientador, Prof. João Carlos, por dedicar tempo e zelo em sua orientação. Com certeza, um dos melhores orientadores que já tive.

Ao meu grande amigo Ronnyel, parceiro de todas as experiências de vida.

Aos meus colegas da UFMA e UFPI.

Aos professores membros da banca examinadora.

A todos os professores e funcionários do DCCMAPI, pelos ensinamentos e colaboração.

A todos que me estimularam ou me ajudaram de alguma forma.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

"...para a santificação dos homens e a glorificação de Deus."

(Sacrosanctum Concilium,6)

Resumo

A reconstrução de imagens é muito importante para atividades que dependem da análise ótica e comparativa de dados. Os sinais, obtidos através de sistemas de aquisição, podem ser corrompidos por vários fatores, como movimentação da câmera e ruídos, mas podem ser recuperados pela aplicação de técnicas que os modelam matematicamente. Com isso, ferramentas de otimização contínua vem se tornando populares nos últimos anos em problemas de imagem. Este trabalho busca reconstruir imagens ruidosas, utilizando o *non-monotone Boosted DC Algorithm* (nmBDCA), uma variante acelerada do *Difference of Convex Algorithm* (DCA), com uma versão não convexa do modelo de *Total Variation* (TV), para obter melhor desempenho computacional que o DCA, com qualidade superior. Os resultados são mostrados em três seções: a primeira com imagens em preto e branco, a segunda com imagens médicas de tomografia computadorizada (TC) e a terceira comparando o nmBDCA a uma versão do *Boosted Difference of Convex Algorithm* (BDCA) com suavização da primeira componente DC, reconstruindo imagens médicas de TC e imagens de ressonância magnética (RM), com diferentes tipos e níveis de ruído. Os resultados da primeira seção enfatizam que o nmBDCA realiza reconstruções com maior PSNR em todos os experimentos, menor tempo de CPU e SSIM maior ou igual ao DCA em 91,67% dos testes. O modelo de TV não convexo é mais robusto que o convexo na presença de mais ruído, apresentando SSIM e PSNR maiores em todos os experimentos realizados. Na segunda seção, o nmBDCA supera DCA em qualidade de reconstrução e tempo de CPU em todos os testes. O desempenho do modelo não convexo aplicado com nmBDCA supera o modelo convexo em SSIM e tempo de CPU em todos os experimentos, sendo superior em PSNR em 77,78% dos os testes. Na terceira seção de resultados é proposta uma variação do BDCA em que uma técnica de suavização é aplicada na primeira componente do modelo DC. Assim, a reconstrução de imagem é similar ao nmBDCA em termos de qualidade (SSIM e PSNR), perdendo apenas em tempo de CPU. Isso mostra o potencial da versão proposta para análise de problemas nesta área de estudo.

Palavras-chave: Otimização não convexa, Remoção de ruído, *Total Variation*, DCA, BDCA, nmBDCA.

Abstract

Image reconstruction is important for activities that rely on optical and comparative data analysis. Signals obtained through acquisition systems can be inconsistent due to several factors, such as camera movement and noise, but they are by applying techniques that model them mathematically. With this, continuous optimization tools have become popular in recent years in image problems. This work seeks to reconstruct noisy images, using non-monotone Boosted DC Algorithm (nmBDCA), an accelerated variant of Difference of Convex Algorithm (DCA), with a non-convex version of the Total Variation (TV) model, to obtain better computational performance than DCA, with similar quality or superior. The results are shown in three sections: the first with black and white images, the second with medical computed tomography (CT) images and the third comparing nmBDCA to a version of *Boosted Difference of Convex Algorithm* (BDCA) with smoothing of the first DC component, reconstructing medical CT images and magnetic resonance images (MRI), with different types and levels of noise. The results of the first section emphasize that nmBDCA performs reconstructions with greater PSNR in all experiments, lower CPU time and SSIM greater than or equal to DCA in 91.67% of tests. The non-convex TV model is more robust than the convex one in the presence of more noise, presenting higher SSIM and PSNR in all experiments performed. In the second section, nmBDCA outperforms DCA in survivability and CPU time in all tests. The performance of the non-convex model applied with nmBDCA outperforms the convex model in SSIM and CPU time in all experiments, being superior in PSNR in 77.78% of those tested. In the third section of results, a variation of BDCA is proposed in which a smoothing technique is applied to the first component of the DC model. Thus, image reconstruction is similar to nmBDCA in terms of quality (SSIM and PSNR), second only in CPU time. This shows the potential of the proposal for analyzing problems in this area of study.

Keywords: Non-convex optimization, Denoising, Total Variation, DCA, BDCA, nmBDCA.

Lista de ilustrações

Figura 1 – Modelo visual da representação esparsa de um sinal.	28
Figura 2 – Representação de um dicionário.	28
Figura 3 – Pixels de uma imagem.	28
Figura 4 – Fluxograma de execução das implementações.	33
Figura 5 – Imagens preto e branco com ruído.	53
Figura 6 – Imagens médicas com ruído.	54
Figura 7 – Varredura do FISTA com variância 0,01.	59
Figura 8 – Varredura do FISTA com variância 0,05.	60
Figura 9 – Varredura do FISTA com variância 0,10.	61
Figura 10 – Varredura do FISTA com variância 0,15.	62
Figura 11 – Reconstrução com FISTA.	63
Figura 12 – Varredura do DCA e do nmBDCA com variância 0,01.	65
Figura 13 – Varredura do DCA e do nmBDCA com variância 0,05.	66
Figura 14 – Varredura do DCA e do nmBDCA com variância 0,10.	67
Figura 15 – Varredura do DCA e do nmBDCA com variância 0,15.	68
Figura 16 – Reconstrução com DCA.	70
Figura 17 – Reconstrução com nmBDCA.	70
Figura 18 – Comparação de resultados para variância 0,01.	72
Figura 19 – Comparação de resultados para variância 0,05.	72
Figura 20 – Comparação de resultados para variância 0,10.	73
Figura 21 – Comparação de resultados para variância 0,15.	73
Figura 22 – Boxplots com variância 0,005.	76
Figura 23 – Boxplots com variância 0,0075.	77
Figura 24 – Boxplots com variância 0,01.	78
Figura 25 – Reconstruções de imagens de TC com variância 0,005.	80
Figura 26 – Reconstruções de imagens de TC com variância 0,0075.	81
Figura 27 – Reconstruções de imagens de TC com variância 0,01.	81
Figura 28 – Reconstruções da imagens de RM com ruído Gaussiano de variância 0,025	84
Figura 29 – Reconstruções das imagens de TC corrompidas por ruído Riciano de variância 0,20.	87

Lista de tabelas

Tabela 1 – Comparação de trabalhos.	21
Tabela 2 – Imagens usadas na Seção 5.3.	48
Tabela 3 – Imagens usadas na Seção 5.4 e 5.5.	48
Tabela 4 – Funções de penalização $\phi_a : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ côncavas, diferenciáveis e não decrescente, parametrizadas por $a > 0$	51
Tabela 5 – Métricas selecionadas para análise das imagens.	56
Tabela 6 – Parâmetros de inicialização do FISTA.	57
Tabela 7 – Resultados de reconstrução com FISTA.	58
Tabela 8 – Parâmetros de inicialização para DCA e nmBDCA.	64
Tabela 9 – Resultados de reconstrução com DCA e nmBDCA.	69
Tabela 10 – Métricas de avaliação para reconstruções com imagens preto e branco (A tolerância do FISTA é 5×10^{-10} . A tolerância das técnicas DC é 5×10^{-4}).	74
Tabela 11 – Métricas de avaliação para reconstruções com imagem de TC (A tolerância do FISTA é 5×10^{-10} . A tolerância das técnicas DC é 5×10^{-4}).	79
Tabela 12 – Métricas de avaliação para RM com ruído Gaussiano	84
Tabela 13 – Métricas de avaliação para os experimentos com ruído Riciano	86

Lista de abreviaturas e siglas

TV	<i>Total Variation</i>
ROF	<i>Rudin Osher Fatemi</i>
FISTA	<i>Fast Iterative Shrinkage Thresholding Algorithm</i>
DC	<i>Difference of convex functions</i>
DCA	<i>Difference of Convex Algorithm</i>
BDCA	<i>Boosted DC Algorithm</i>
nmBDCA	<i>Non-monotone Boosted DC Algorithm</i>
SSIM	<i>Structural Similarity Index Measure</i>
PSNR	<i>Peak Signal-to-Noise Ratio</i>

Sumário

1	INTRODUÇÃO	13
1.1	Definição do Problema	13
1.1.1	Visão geral da proposta	15
1.1.2	Objetivos	15
1.1.3	Justificativa	16
1.1.4	Publicações	17
1.2	Estrutura desta Tese	17
2	TRABALHOS RELACIONADOS	18
3	FUNDAMENTAÇÃO TEÓRICA	22
3.1	Conceitos básicos de otimização	22
3.2	Diferença de funções convexas (DC)	25
3.3	Modelos de representação de imagem	27
3.4	Considerações finais	31
4	ALGORITMOS	33
4.1	<i>Fast Iterative Shrinkage Thresholding Algorithm (FISTA)</i>	33
4.2	O algoritmo DCA	41
4.3	<i>Non-monotone boosted DC Algorithm (nmBDCA)</i>	43
4.4	Considerações finais	47
5	RECONSTRUÇÃO DE IMAGENS	48
5.1	Remoção de ruído em problemas não convexos	49
5.1.1	<i>Denoising</i> com modelo TV_ϕ	50
5.1.2	Algoritmos DCA-FISTA e nmBDCA-FISTA com TV_ϕ	51
5.2	Inserção de ruído	53
5.3	Reconstrução de imagens preto e branco	57
5.3.1	Reconstrução com FISTA	57
5.3.2	Reconstrução com DCA e nmBDCA	63
5.3.3	Comparação entre os modelos convexo e não convexo	71
5.4	Reconstrução de imagens médicas	74
5.5	Comparação entre nmBDCA e BDCA suavizado	82
5.5.1	Experimentos com ruído Gaussiano	82
5.5.2	Experimentos com ruído Riciano	84
5.6	Comentários finais dos experimentos	87

6	CONCLUSÕES E CONSIDERAÇÕES FINAIS	90
7	APÊNDICES	92
7.1	Demonstração do Lema 4.1	92
7.2	Demonstração da Proposição 4.1	92
7.3	Demonstração da Proposição 4.2	93
7.4	Demonstração do Teorema 4.3	93
7.5	Demonstração da Proposição 4.3	94
7.6	Demonstração do Lema 4.2	96
7.7	Demonstração da Proposição 4.4	96
7.8	Demonstração do Corolário 4.1	96
7.9	Demonstração da Proposição 4.5	96
7.10	Demonstração do Teorema 4.4	97
7.11	Demonstração do Teorema 4.5	98
7.12	Demonstração da Proposição 4.6	98
7.13	Demonstração do Teorema 4.6	99
7.14	Demonstração do Teorema 4.7	99
7.15	Demonstração da Proposição 5.1	100
7.16	Demonstração da Proposição 5.2	100
	REFERÊNCIAS	102

1 Introdução

Os problemas que envolvem a reconstrução de imagens, além de poder envolver grande quantidade de variáveis, necessitam da manutenção de recursos importantes, como bordas e a textura; veja [Ma, Yu e Zeng \(2013\)](#), [Jidesh \(2014\)](#), [Fang et al. \(2016\)](#). Assim, uma propriedade importante dos modelos para processamento em imagem é a preservação de descontinuidades em suas soluções, com o intuito de manter a identificação precisa das bordas da imagem. Em consonância, outro objetivo dos modelos é a robustez, definida como a capacidade do modelo se manter estável na presença de ruídos e *outliers* ([BEN-TAL; NEMIROVSKI, 1998](#); [BEN-TAL; GHAOUI; NEMIROVSKI, 2009](#)).

Abordagens de sucesso para esses problemas minimizam funções com base nas normas euclidianas. Nesse caso, os algoritmos de otimização aplicados exploram a estrutura de problemas em grande escala (grande quantidade de amostras) de modo a alcançar melhores soluções.

1.1 Definição do Problema

A representação esparsa de sinais consiste na modelagem pela combinação linear de partes constituintes de um banco de dados, denominado dicionário ([CHEN; DONOHO, 1994](#)). Ela corresponde à amostragem e compressão de sinais, e supera as limitações do teorema de amostragem de Nyquist-Shannon, que afirma que a frequência de amostragem deve ser pelo menos duas vezes maior como a frequência mais alta do sinal, e, ainda, permite robustez na presença de ruído ([CANDÈS; ROMBERG; TAO, 2006](#); [AHARON; ELAD; BRUCKSTEIN, 2006](#); [DONOHO, 2006](#)). Assim, a representação esparsa permite a amostragem de grandes quantidades de dados com economia de memória ([ELAD; FIGUEIREDO; MA, 2010](#)).

Técnicas de processamento de sinal geralmente necessitam de representações significativas, que capturem as características úteis do sinal, ou seja, devem-se destacar as características inerentes ao sinal ([RUBINSTEIN; BRUCKSTEIN; ELAD, 2010](#)). A remoção de ruídos (*Denoising*) necessita da separação eficiente entre sinal e o ruído. Com a compressão, a representação deve capturar uma grande parte do sinal com apenas alguns coeficientes. Para essas aplicações, as representações esparsas utilizam restrições aplicadas ao vetor de coeficientes, que correspondem às normas. As normas utilizadas com frequência são l_0 ou l_1 , e influenciam diretamente a esparsidade e dificuldade do problema a ser estudado ([STARCK; MURTAGH; FADILI, 2010](#)).

A representação esparsa é muito utilizada em processamento de sinais com

o objetivo de filtrar ruídos em sinais diversos (ANDRADE, 2020). A restauração de imagens desempenha um papel importante em muitas aplicações, e tem o objetivo de recuperar com alta qualidade uma imagem degradada na presença de ruído (MA; YU; ZENG, 2013). Na literatura, alguns ruídos considerados são os ruídos de impulso (BECK; TBOULLE, 2009b), ruídos gaussianos (CHAN; DONG; HINTERMÜLLER, 2010; CAI; CHAN; NIKOLOVA, 2010), misturas de ruídos (DONG et al., 2012), etc.

De acordo com Jia e Zhao (2010), o modelo ROF, proposto por Rudin, Osher e Fatemi (1992), é considerado um dos mais usados para a remoção de ruídos em imagem. Esse modelo propõe a regularização para eliminação de ruídos, e consiste em medir o quanto os valores do sinal mudam entre si, buscando encontrar uma aproximação que obtenha a menor variação total (*Total Variation* - TV), cuja reconstrução esteja mais próxima ao sinal original (BAO et al., 2008). Assim, a eliminação de ruído por variação total consiste em minimizar uma função objetivo, a partir de um parâmetro de regularização (HÜTTER; RIGOLLET, 2016). O modelo proposto é adequado para problemas com ruído gaussiano de média zero, mas encontra dificuldade em problemas com outros tipos de ruído, como é o caso do ruído sal-pimenta (*salt-and-pepper*). Neste caso, pode-se aplicar uma variação do modelo ROF ao se utilizar um termo de ajuste com norma l_1 , modelo conhecido como TV- l_1 , que de acordo com Nikolova (2004) e Chan e Esedoglu (2005), possui propriedades interessantes como robustez ao ruído.

A representação esparsa consiste, então, de um problema de otimização. A solução para esses problemas é geralmente alcançada com o uso de algum método computacional. Para isso, existem técnicas de otimização, como redes neurais, métodos de busca linear, método dos mínimos quadrados, método do lagrangeano aumentado, otimização convexa, cada um adaptado a um tipo específico de problema (SILVA, 2017).

Introduzido por Tao e Souad (1986), o *Difference of Convex Algorithm* (DCA) é um algoritmo feito especificamente para diferença de funções convexas (funções DC), que é a base da programação DC. É aplicado em muitos campos, como aprendizado de máquina, otimização financeira, telecomunicações (AN; TAO, 2005; THI; DINH, 2018) e em processamento de imagens (THI; DINH, 2017; YU; ZHANG, 2018; LIANG; ZHANG; JACOBS, 2020; YOU; NIU, 2021).

Com a gama de estudos a partir dos trabalhos de Tao e Souad (1986), Artacho, Fleming e Vuong (2018) desenvolve um algoritmo para minimizar a diferença de funções convexas diferenciáveis (suaves), que acelera a convergência do algoritmo DCA, provando que o ponto calculado pelo DCA pode ser usado para definir uma direção de descida para a função objetivo avaliada neste ponto. O algoritmo é intitulado *Boosted Difference of Convex Function Algorithm* (BDCA) e se baseia em uma combinação de DCA com uma etapa de busca que usa essa direção de descida, de forma que a principal diferença em relação do DCA é o ponto de partida usado para a busca linear,

neste caso, o ponto gerado pelo DCA, em vez de usar a iteração anterior.

Ferreira, Santos e Souza (2024) desenvolvem o método *non-monotone Boosted DC Algorithm* (nmBDCA), substituindo a busca monótona do BDCA por uma busca não monótona, para controlar o crescimento da função objetivo por meio de um parâmetro, o que permite que o método seja implementado para funções DC cujas componentes são não diferenciáveis. O método proposto tem propriedades semelhantes de convergência assintótica e limites de complexidade de iteração, mas é melhor em desempenho computacional (tempo de CPU) que os algoritmos DC clássicos para resolução de problemas em programação DC.

1.1.1 Visão geral da proposta

A proposta deste trabalho é utilizar uma versão acelerada do DCA, denominada *non-monotone Boosted DC Algorithm* (nmBDCA), em conjunto com um modelo TV não convexo, para remoção de ruído em imagens e avaliação do modelo TV na presença de diferentes níveis e tipos de ruído. Isso permitirá a evolução do processo de reconstrução de imagens em termos computacionais, em robustez quanto a presença de ruídos, e quanto as reconstruções já realizadas com técnicas clássicas.

Para isso, o modelo e o método são testados em imagens corrompidas por ruídos Gaussiano e Riciano, com diferentes variâncias, e a qualidade das imagens obtidas é avaliada e comparada com base nas métricas de *Peak Signal-to-Noise Ratio* (PSNR) e *Structural Similarity Index Metrics* (SSIM). Assim, os resultados são analisados de duas formas. A primeira analisa o desempenho de reconstrução entre os modelos convexo e não convexo, e a segunda verifica o desempenho computacional entre as técnicas aplicadas.

1.1.2 Objetivos

Objetivo geral

Aplicar o *non-monotone Boosted DC Algorithm* (nmBDCA) com modelo TV não convexo para reconstruir imagens ruidosas. Com isso, obter reconstruções com qualidades superiores ao DCA. Além disso, propor uma versão do BDCA que suaviza a primeira componente DC, mantendo a qualidade das imagens reconstruídas, em comparação do nmBDCA.

Objetivos específicos

Como objetivos específicos, este trabalho busca:

- Aplicar a técnica FISTA com modelo TV convexo para reconstruir imagens;

- Aplicar as técnicas DCA e nmBDCA com modelo TV não convexos para reconstruir imagens;
- Implementar uma versão suavizada do BDCA com modelo TV não convexos para reconstruir imagens;
- Comparar a qualidade das reconstruções entre os modelos convexo e não convexo;
- Comparar a qualidade das reconstruções do nmBDCA e do DCA;
- Avaliar o desempenho computacional do nmBDCA para reconstrução de imagens, e comparar com o DCA;
- Comparar a qualidade das reconstruções e o desempenho computacional entre nmBDCA e uma versão suavizada do BDCA, em diferentes níveis e tipos de ruído;
- Analisar a influência do parâmetro de regularização dos modelos TV convexo e não convexo.

1.1.3 Justificativa

A natureza de cada problema permite especificar os algoritmos e definir as variáveis a se usar em cada método. Alguns trabalhos aplicam tipos de TV (anisotrópica e isotrópica) em conjunto com o DCA para eliminação de ruídos de imagem; veja [Yu e Zhang \(2018\)](#), [Brooks et al. \(2019\)](#). Outros propõem um modelo de regularização TV para a eliminação de ruído ou estudam o desfoque em imagens; veja [Thanh et al. \(2020\)](#), [Zhang, Zhu e Zhu \(2020\)](#). Mas, não realizam um estudo que contempla a inserção da abordagem DC, a comparação de técnicas aceleradas, e a comparação entre modelos distintos, para imagens com diferentes características, corrompidas por diferentes níveis e tipos de ruído.

Dessa forma, apesar do escopo geral dos trabalhos ser a reconstrução de imagens, nesta Tese, tem-se a aplicação do método acelerado *non-monotone Boosted DC Algorithm* (nmBDCA) para minimizar a diferença de duas funções convexas (funções DC), cujas componentes DC são possivelmente não suaves, para reconstruir imagens ruidosas, em diferentes níveis e tipos de ruído, com imagens de diferentes características com menos tempo e qualidade superior ao DCA. Espera-se alcançar isso porque o nmBDCA tem propriedades de convergência e complexidade de iteração semelhantes aos métodos de descida usuais para minimizar funções convexas, porém melhor desempenho computacional. Além disso, aplica-se uma versão suavizada do BDCA, buscando manter a qualidade das reconstruções obtidas, em comparação ao nmBDCA, sem a necessidade de aplicação da busca não monótona.

1.1.4 Publicações

Parte desta Tese foi publicada no trabalho [Ferreira et al. \(2024\)](#), intitulado **Image denoising with a non-monotone boosted DCA for non-convex models**, na revista ***Computers and Electrical Engineering***¹. Uma outra publicação no trabalho [Ribeiro e Souza \(2024\)](#), com o título ***Image denoising with non-convex models: a comparison between BDCA and nmBDCA***, foi realizada por IEEE, oriunda da participação na **37ª Conference on Graphics, Patterns and Images (SIBGRAPI 2024)**, que ocorreu nos dias 30/09 a 03/10 de 2024, em Manaus-AM².

1.2 Estrutura desta Tese

Esta Tese tem a seguinte estrutura: O Capítulo 2 aborda trabalhos relacionados ao tema, situando o leitor na discussão. O Capítulo 3 fundamenta as técnicas utilizadas, com conceitos e preceitos básicos que norteiam informações a posteriori. O Capítulo 4 apresenta as etapas que compõem a metodologia proposta. São descritas a sequência de implementação de técnicas e a base teórica para a implementação de cada uma. O Capítulo 5 trata dos experimentos, resultados e discussões, fornecendo informações de configuração das técnicas, resultados visuais e numéricos para cada reconstrução. O Capítulo 6 expõe as considerações finais e possíveis trabalhos futuros. Por fim, algumas propriedades de convergência dos métodos são demonstradas no Capítulo 7.

¹ <https://doi.org/10.1016/j.compeleceng.2024.109306>

² <https://doi.org/10.1109/SIBGRAPI62404.2024.10716308>

2 Trabalhos relacionados

A representação esparsa tem papel significativo para o processamento de sinais, por isso, tem sido amplamente utilizada em classificação, remoção de ruídos, reconhecimento facial, etc. Ela é uma técnica para representar sinais pela combinação linear de amostras, constituintes de um banco de amostras maior, e é desejável por permitir robustez na presença de ruído, e a inserção de conhecimento prévio dos dados em sua representação matemática (XU; YANG; JIANG, 2017).

Wang et al. (2017) propõem uma estrutura de representação esparsa baseada em minimização de normas l_p não convexa para eliminação de ruído de imagens. Para tornar a proposta tratável e robusta, o algoritmo de limiar suave generalizado é adotado para resolver o problema de remoção de ruído em imagem. Os resultados experimentais demonstram que a abordagem proposta não apenas supera muitos métodos de remoção de ruído, mas também resulta em velocidade competitiva à de tais métodos. Yang et al. (2019), por sua vez, criam um modelo de representação esparsa para remoção de ruído de imagens denominado modelo de Markov oculto com contorno adaptativo (HMM-PCNN). Eles realizam a decomposição da imagem ruidosa, criam um modelo estatístico da decomposição e aplicam probabilidade para transformar o problema em bayesiano. Os resultados obtidos revelam superioridade em relação aos métodos de Markov oculto e limiar de wavelet.

Agregado à representação esparsa de sinais estão as técnicas de remoção de ruído, como o modelo ROF. Campagna et al. (2017) modificam o modelo ROF a partir da inserção de um termo de penalidade baseado em derivadas de segunda ordem. Assim, a partir de experimentos, eles confirmam as vantagens do esquema proposto para melhorar o contraste e elevar a qualidade visual da imagem de ressonâncias magnéticas, em relação ao modelo ROF sem a penalidade. Zhang et al. (2018) estudam a precisão do fluxo óptico na navegação de micro veículos aéreos, em condições de baixa luminosidade, uma vez que esse sistema complementa as tarefas de reconhecimento e vigilância em baixa altitude e ambientes de voo. O estudo avalia a aplicação do modelo ROF em conjunto com a programação convexa para a remoção de ruído de imagens, em condições de iluminação inconstante, para calcular a posição e altitude dos micro-veículos aéreos. Os resultados mostram que a precisão do cálculo do fluxo óptico é melhorada pela decomposição e eliminação de ruído.

Percebe-se que o modelo ROF, geralmente, é auxiliado por melhorias a cada problema de imagem. Uma outra ferramenta eficaz para processamento de imagem e redução de ruído é o *Total Variation* (TV), conforme Thanh, Prasath et al. (2018),

que desenvolvem um método de estimação de parâmetros baseado em características do ruído sal e pimenta, eficaz para imagens sem contraste muito alto e com alto nível de ruído, e comparam com outros métodos de remoção desse ruído. Concluem que o método proposto pode funcionar bem em vários níveis de ruído, tornando-se mais eficaz em imagens com alto nível de ruído (de 50% e acima). Assim, em imagens com nível de ruído de 80%, o método pode ser competitivo para a remoção do ruído sal e pimenta.

Melhores desempenhos em modelagem, remoção de ruídos e reconstrução de imagens são obtidos ao se agregar o modelo TV com DCA, como exposto no trabalho de [Zhou \(2021\)](#). Ele apresenta um método de recuperação esparsa baseado na função de erro generalizado, a partir da introdução de uma função de penalidade que envolve a forma e os parâmetros de escala dos dados. A análise envolve o modelo TV e o DCA, e os experimentos demonstram a melhoria proporcionada pela abordagem na aplicação prática em sistemas de ressonância magnética para a reconstrução de imagens.

O modelo TV é uma regularização convexa e não preserva as características de uma imagem, apresentando limitações como o efeito escada, pixelização da imagem em regiões lisas e perda de regiões com textura fina por não considerar a existência de regiões particularmente ricas em detalhes. Assim, há uma clara necessidade de estudar melhorias nos métodos variacionais, como ocorre em [Fang e Xianghai \(2023\)](#) e [Shi, Li e Lou \(2023\)](#), que indica a inserção da regularização funcional não convexa como uma abordagem significativa a ser escolhida para obter imagens com mais qualidade. O objetivo da introdução da regularização não convexa é ajustar as distribuições de gradiente da imagem e melhorar a restauração de regiões muito detalhadas, como bordas e texturas. No entanto, pode tornar o problema estudado um desafio do ponto de vista computacional ou matemático, pois a técnica aplicada pode não retornar soluções adequadas devido à minimização não convexa envolvendo muitos mínimos locais.

[Bai \(2019\)](#) aplica o modelo TV com o DCA para restaurar imagens com ruído multiplicativo, contendo um termo de fidelidade não convexo e um termo variacional. Pelos experimentos realizados, obtém-se solução melhor, em comparação com o algoritmo de projeção de gradiente para resolver os modelos clássicos de remoção de ruído multiplicativo, demonstrando que o algoritmo proposto supera métodos tradicionais para remoção de ruído multiplicativo.

Utilizando também a otimização não convexa e a variação total não suave, [Wu et al. \(2022\)](#) aplicam o BDCA para recuperar imagens médicas. Eles demonstram que a sequência gerada pelo algoritmo proposto converge para um ponto estacionário com os valores da função objetivo decrescendo monotonicamente, e que a abordagem desenvolvida supera alguns métodos para recuperação em termos de qualidade e tempo.

[Zhang et al. \(2022\)](#), por sua vez, analisa o problema de restauração de imagem com ruído de impulso, e propõem um modelo variacional que integra o termo de ajuste de dados não convexo e a regularização de TV não convexa, o que permite detectar o ruído de impulso de maneira mais eficaz. Ainda, é desenvolvido um novo algoritmo de diferenças de funções convexas para resolução do modelo variacional. Os resultados experimentais mostram que o algoritmo proposto é eficiente e comparável com métodos atuais.

Uma aplicação do nmBDCA para remover o ruído Riciano, que muitas vezes corrompe imagens de ressonância magnética, foi realizada por [Sun, Xu e Feng \(2023\)](#). Foi proposto um modelo e aplicado o nmBDCA para resolver o problema não convexo de remoção simultânea de ruído e borrão. Os resultados numéricos mostram melhorias na qualidade da restauração das imagens estudadas, preservando detalhes importantes e reduzindo o efeito da escada.

No entanto, esta aplicação difere deste trabalho, uma vez que o nmBDCA é aplicado com um modelo não convexo baseado na variação total penalizada para reconstruir imagens corrompidas por ruído gaussiano, preservando as bordas e regiões constantes com mais qualidade e comparando os resultados com o modelo convexo. Para endossar os resultados obtidos, uma varredura no parâmetro de regularização do modelo foi realizada e o nível de ruído inserido em cada imagem alterado, permitindo mostrar a superioridade do modelo convexo em relação à qualidade das imagens obtidas, e ao desempenho computacional (tempo de CPU) superior do nmBDCA em relação ao DCA.

Conforme explicitado, a reconstrução adequada de imagens necessita da utilização de diversas técnicas, principalmente quando as imagens possuem ruído. Assim, a utilização de boas técnicas para modelagem e remoção de ruído em imagem não será suficiente, caso a reconstrução dessas imagens não possua complexidade e tempo adequados ao problema estudado.

Assumindo, então, as condições de complexidade obtidas por [Artacho, Fleming e Vuong \(2018\)](#), [Artacho e Vuong \(2020\)](#) e as de diferenciabilidade das componentes DC do trabalho de [Ferreira, Santos e Souza \(2024\)](#), este trabalho propõe aplicar o nmBDCA para reconstruir imagens, com tempo de CPU melhor que o DCA e qualidade de reconstrução superior, e avaliar a robustez do modelo não convexo para diferentes níveis de ruído gaussiano, comparando-a com o modelo convexo. A Tabela 1 apresenta uma comparação para situar as contribuições deste trabalho.

Tabela 1 – Comparação de trabalhos.

Trabalho	M. não convexo	Método DC	Acel.	Compon. não difer.	Var. do ruído (gaussiano/riciano)	Comp. BDCA e nmBDCA (gaussiano/riciano)	Análise do parâm. de regular.	Compara modelos
Zhou (2021)	Sim	Sim	Não	Não	Não/Não	Não/Não	Não	Não
Bai (2019)	Sim	Sim	Não	Não	Não/Não	Não/Não	Não	Não
Zhang et al. (2022)	Sim	Sim	Não	Não	Sim/Não	Não/Não	Não	Não
Wu et al. (2022)	Sim	Sim	Sim	Não	Não/Sim	Não/Não	Não	Não
Sun, Xu e Feng (2023)	Sim	Sim	Sim	Sim	Não/Sim	Não/Sim	Não	Não
Este trabalho	Sim	Sim	Sim	Sim	Sim/Sim	Sim/Sim	Sim	Sim

As principais contribuições desse trabalho são: Verificação da qualidade das reconstruções e do tempo de CPU do nmBDCA, em comparação ao DCA; Estudo da influência do nível de ruído em reconstruções com os modelos TV convexo e não convexo, comparando o desempenho desses modelos; Análise da reconstrução de imagens médicas com nmBDCA para modelos TV não convexos; Avaliação do desempenho de qualidade e tempo de CPU entre nmBDCA e uma versão suavizada do BDCA, em reconstruções com diferentes níveis de ruído Gaussiano e Riciano. Essas contribuições mostram o ganho na qualidade da reconstrução quando o modelo não convexo é aplicado e também enfatizam que o uso do nmBDCA traz melhorias no desempenho computacional.

3 Fundamentação teórica

Este capítulo apresenta definições e conceitos usados para o desenvolvimento deste estudo, bem como modelos representativos de imagem e problemas de otimização com funções convexas. As notações básicas apresentadas a seguir, acerca do espaço Euclidiano, podem ser encontradas em [Lima \(2015\)](#), [Beck \(2017\)](#), [Izmailov e Solodov \(2005\)](#), [Hiriart-Urruty e Lemarechal \(1993\)](#) e [Rockafellar e Wets \(2009\)](#).

3.1 Conceitos básicos de otimização

Dado um número natural n , o espaço Euclidiano n -dimensional, denotado por \mathbb{R}^n , é o produto cartesiano de n cópias de \mathbb{R} , isto é:

$$\mathbb{R}^n = \underbrace{\mathbb{R} \times \mathbb{R} \times \cdots \times \mathbb{R} \times \mathbb{R}}_{n \text{ vezes}}.$$

Um elemento x é um ponto de \mathbb{R}^n denotado por uma n -upla de números reais, isto é, $x = (x_1, x_2, \dots, x_n)$, em que, para cada $i \in \{1, \dots, n\}$, x_i é um número real, dito a i -ésima coordenada de x . Ainda, pode-se definir a norma Euclidiana (l_2), dado $x = (x_1, \dots, x_n)$, como:

$$\|x\| = \sqrt{\langle x, x \rangle} = \sqrt{x^\top x} = \sqrt{x_1^2 + \cdots + x_n^2},$$

a norma da soma (l_1) por:

$$\|x\|_1 = |x_1| + \cdots + |x_n|,$$

onde $|x| = \max\{x, -x\}$ é o módulo de x . De um modo geral, a norma- p é dada por:

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}, \text{ com } p \geq 1 \text{ um número real.}$$

A projeção de um ponto $x \in \mathbb{R}^n$ sobre um conjunto $C \subset \mathbb{R}^n$ é um ponto pertencente a C que está mais próximo de x . Em outras palavras, a projeção $P_C(x)$ de um ponto $x \in \mathbb{R}^n$ sobre um conjunto $C \subset \mathbb{R}^n$ é a solução do problema:

$$P_C(x) = \arg \min_{y \in C} \|x - y\|.$$

Quando o conjunto C é fechado e não vazio, a projeção está bem definida. Além disso, caso C seja convexo, a projeção é única ([IZMAILOV; SOLODOV, 2005](#)).

Considerando uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}$, diz-se que f é semicontínua inferiormente (resp. superiormente) em um ponto $x \in \mathbb{R}^n$, se para qualquer sequência $\{x^k\} \subset \mathbb{R}^n$

convergindo para x , tem-se:

$$f(x) \leq \liminf_{k \rightarrow +\infty} f(x^k) \quad (\text{resp. } \limsup_{k \rightarrow +\infty} f(x^k) \leq f(x)).$$

A função f é dita convexa, se para todo $x, y \in \mathbb{R}^n$ e $\lambda \in [0,1]$, tem-se:

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

Quando a desigualdade acima é estrita, dizemos que f é estritamente convexa. A função é chamada de fortemente convexa (com constante σ) caso exista $\sigma > 0$, tal que:

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) - \sigma \lambda(1 - \lambda)\|x - y\|^2.$$

Em particular, f é σ -fortemente convexa se e somente se $f - \frac{\sigma}{2}\|\cdot\|^2$ é convexa.

Outra definição importante em otimização é o de subdiferencial (conhecido como subdiferencial de Fenchel-Moreau) de uma função em um ponto denotado por $\partial f(x)$, é definido por:

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

Diz-se que f é uma função Lipschitz contínua com constante L , se existe $L > 0$ tal que para todo $x, y \in \mathbb{R}^n$, onde:

$$|f(x) - f(y)| \leq L\|x - y\|.$$

Quando a propriedade anterior é válida em uma vizinhança U de um ponto $\hat{x} \in \mathbb{R}^n$, diz-se que f é localmente Lipschitz em \hat{x} . Se f é localmente Lipschitz em todos os pontos, diz-se apenas que f é localmente Lipschitz.

Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função localmente Lipschitz em $x \in \mathbb{R}^n$ e $d \in \mathbb{R}^n$, com $d \neq 0$, uma direção. A derivada direcional de Clarke de f em x na direção de d , denotada por $f^\circ(x, d)$, é definida por:

$$f^\circ(x, d) := \limsup_{\substack{y \rightarrow x \\ t \rightarrow 0}} \frac{f(y + td) - f(y)}{t}$$

Assim, define-se o subdiferencial de Clarke de f em x , denotado por $\partial_C f(x)$, como:

$$\partial_C f(x) := \{w \in \mathbb{R}^n : \langle w, d \rangle \leq f^\circ(x, d), \quad \forall d \in \mathbb{R}^n\}.$$

A seguir, são apresentados alguns resultados clássicos de otimização, usados ao longo deste trabalho.

Proposição 3.1. *Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e $x \in \mathbb{R}^n$. Então o subdiferencial $\partial f(x)$ é fechado e convexo.*

Demonstração. Veja [Rockafellar e Wets \(2009, Teorema 8.6\)](#). □

Proposição 3.2. *Se uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ tem um ponto de mínimo local em x , então $0 \in \partial f(x)$. Se f é convexa, tal condição é necessária e suficiente para um ponto ser mínimo global. Ainda, se $f = f_1 + f_2$ com f_2 continuamente diferenciável, a condição $0 \in \partial f(x)$ tem a forma $-\nabla f_2(x) \in \partial f_1(x)$.*

Demonstração. Veja [Rockafellar e Wets \(2009, Teorema 10.1\)](#). □

Proposição 3.3. *Se f_1 é localmente Lipschitz em x , f_2 é semicontínua inferiormente, então:*

$$\partial(f_1 + f_2)(x) \subseteq \partial f_1(x) + \partial f_2(x).$$

Demonstração. Veja [Rockafellar e Wets \(2009, Página 431\)](#). □

Proposição 3.4. *Seja $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função convexa. Seja $(z^k)_{k \in \mathbb{N}} \subset \mathbb{R}^n$ tal que $\lim_{k \rightarrow \infty} z^k = z^*$. Se $(s^k)_{k \in \mathbb{N}}$ é uma sequência tal que $s^k \in \partial \phi(z^k)$, para todo $k \in \mathbb{N}$, então $(s^k)_{k \in \mathbb{N}}$ é limitado e seus pontos de acumulação pertencem a $\partial \phi(z^*)$.*

Demonstração. Veja [Hiriart-Urruty e Lemarechal \(1993, Proposição 6.2.1\)](#). □

Proposição 3.5. *Seja $\phi_1 : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função fortemente convexa com módulo $\sigma > 0$, e seja $\phi_2 : \mathbb{R}^n \rightarrow \mathbb{R}$ convexa. Então $\phi_1 + \phi_2$ é uma função fortemente convexa com módulo $\sigma > 0$.*

Demonstração. Veja [Beck \(2017, Lema 5.20\)](#). □

Proposição 3.6. *Seja $\phi_1 : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função convexa, e $\phi_2 : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função convexa e não decrescente. Então $\phi_2(\phi_1(x))$ é uma função convexa.*

Demonstração. Veja [Izmailov e Solodov \(2005, Proposição 3.4.3\)](#). □

Proposição 3.7. *Seja $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função. Então, ϕ é uma função convexa se, e somente se, $\partial f(x) \neq \emptyset$, para todo $x \in \mathbb{R}^n$. Além disso, as três afirmações a seguir são equivalentes:*

1. ϕ é uma função σ -fortemente convexa com $\sigma \geq 0$;
2. $\phi(y) \geq \phi(x) + \langle v, y - x \rangle + \frac{\sigma}{2} \|y - x\|^2$, para todo $x, y \in \mathbb{R}^n$ e todo $v \in \partial \phi(x)$;
3. $\langle w - v, x - y \rangle \geq \sigma \|y - x\|^2$, para todo $x, y \in \mathbb{R}^n$, todo $w \in \partial \phi(x)$ e todo $v \in \partial \phi(y)$.

Demonstração. Veja [Beck \(2017, Teorema 5.24\)](#). □

Proposição 3.8. *Seja $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função convexa. Então, ϕ tem um mínimo local/global x^* se, e somente se, $0 \in \partial f(x^*)$.*

Demonstração. Veja Beck (2017, Teorema 3.63). □

Definição 3.1. *Dada uma função $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$, o conjugado de ϕ é definido por $\phi^*(y) = \sup_{x \in \mathbb{R}^n} \{y^\top x - f(x)\}$.*

Proposição 3.9. *Seja $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função convexa fechada. Para qualquer $x, y \in \mathbb{R}^n$, as seguintes afirmações são equivalentes:*

(i) $y \in \partial\phi(x)$;

(ii) $x \in \partial\phi^*(y)$.

Demonstração. Veja Beck (2017, Teorema 4.20) □

Corolário 3.1. *Se $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função convexa e diferenciável, o gradiente $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ é contínuo.*

Demonstração. Veja Rockafellar e Wets (2009, Corolário 9.20). □

3.2 Diferença de funções convexas (DC)

As funções definidas pela diferença de duas funções convexas, funções DC, são uma classe especial de funções, não necessariamente convexas, onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função DC quando existem $g, h : \mathbb{R}^n \rightarrow \mathbb{R}$ funções convexas tal que $f(x) = g(x) - h(x)$, para todo $x \in \mathbb{R}^n$. As funções g e h são chamadas de componentes DC.

Algumas considerações acerca das funções DC são:

- Denota-se por $\Gamma_0(\mathbb{R}^n)$ o cone das funções semicontínua inferiormente e convexas definidas no \mathbb{R}^n e o espaço vetorial das funções DC por $DC(\mathbb{R}^n)$;
- O espaço das funções DC é o menor espaço vetorial que contém todas as funções convexas e contínuas definidas em um determinado conjunto; ver Bačák e Borwein (2011);
- $DC(\mathbb{R}^n)$ contém o espaço $C^{1,1}$ das funções diferenciáveis, cujo gradiente é localmente Lipschitz e os espaços das funções *lower* - C^2 , funções que são localmente supremas de uma família de funções duas vezes continuamente diferenciáveis; ver Ginchev e Gintcheva (2013, Teorema 2.4).

Por simplicidade, opta-se por apresentar os resultados no espaço \mathbb{R}^n . Entretanto, resultados envolvendo diferença de funções convexas (ou côncavas) já foram estudados em contextos mais gerais, como por exemplo espaços de Hilbert ou Banach, veja [Bačák e Borwein \(2011\)](#) e suas referências.

Define-se um problema de otimização DC por:

$$\min_{x \in \mathbb{R}^n} f(x) = g(x) - h(x),$$

onde $f \in DC(\mathbb{R})$, com $g, h \in \Gamma_0(\mathbb{R})$.

Seja $f \in DC(\mathbb{R}^n)$, então existem funções convexas $g, h : \mathbb{R}^n \rightarrow \mathbb{R}$ tal que $f(x) = g(x) - h(x)$. Assim, algumas propriedades de f são herdadas das funções g e h , como, por exemplo, que f é localmente Lipschitz em \mathbb{R}^n . Ainda, a derivada direcional de f , $d \mapsto f'(x, d)$, existe e $f'(x, d) = g'(x, d) - h'(x, d)$, para todo $x, d \in \mathbb{R}^n$. Além disso, tem-se que $\partial_C f(x) = \partial g(x) - \partial h(x)$, para todo x , onde $\partial_C f(x)$ denota o subdiferencial de Clarke; veja [Bačák e Borwein \(2011\)](#).

O subdiferencial $\partial_C f(x)$ não necessariamente se reduz a $\{\nabla f(x)\}$ quando f é diferenciável em $x \in \mathbb{R}^n$, diferentemente das funções convexas ([BAČÁK; BORWEIN, 2011](#), Exemplo 5.3). Para as funções DC, tem-se que $\partial_C f(x)$ se reduz quase sempre a $\{\nabla f(x)\}$, quando f é diferenciável; veja [Bačák e Borwein \(2011\)](#).

Outra propriedade válida para funções convexas, mas não para as funções DC, é que o limite de funções DC não necessariamente é uma função DC. Porém, sabe-se que toda função contínua definida em um conjunto convexo e compacto é limite uniforme de uma sequência de funções DC; veja [Bačák e Borwein \(2011, Proposição 2.2\)](#).

A seguir é apresentado uma importante propriedade de funções DC.

Teorema 3.1. *Toda função localmente DC em \mathbb{R}^n é globalmente DC em \mathbb{R}^n .*

Demonstração. Veja [Hartman \(1959\)](#). □

Como consequência do Teorema 3.1, tem-se que toda função da classe C^2 é uma função DC. Algumas propriedades de funções convexas também são válidas para funções DC, conforme explicitado a seguir.

Proposição 3.10. *Sejam $f_i \in DC(\mathbb{R}^n, i = 1, \dots, m)$. Tem-se que:*

1. $f(x) = \sum_{i=1}^m \lambda_i f_i(x)$, $\lambda_i \in \mathbb{R}$, é uma função DC;
2. $f(x) = \prod_{i=1}^m f_i(x)$ é uma função DC;
3. $f(x) = \max_{1 \leq i \leq m} f_i(x)$ é uma função DC;

4. $f(x) = \min_{1 \leq i \leq m} f_i(x)$ é uma função DC;

5. Se f é uma função DC e g uma função convexa, então $f \circ g$ é uma função DC.

Demonstração. Veja [Bačák e Borwein \(2011\)](#). □

3.3 Modelos de representação de imagem

Conforme já mencionado, a representação esparsa de sinais permite concentrar informações de um sinal em quantidades reduzidas de amostras, possibilitando que a reconstrução seja realizada com menor quantidade de amostras do sinal. Logo, em representação de sinais, esparsidade está associada a um sinal que possui grande número de valores nulos, ou seja, considerando um sinal $\alpha \in \mathbb{R}^n$, dado por $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$, diz-se que este sinal é esparsos se muitas de suas entradas são iguais a zero, isto é, o suporte $\Lambda(\alpha) = \{1 \leq i \leq n \mid \alpha_i \neq 0\}$ é de cardinalidade $k < n$, onde k é a quantidade de amostras não nulas em um sinal, denominado k -esparsos ([BATISTA, 2018](#)).

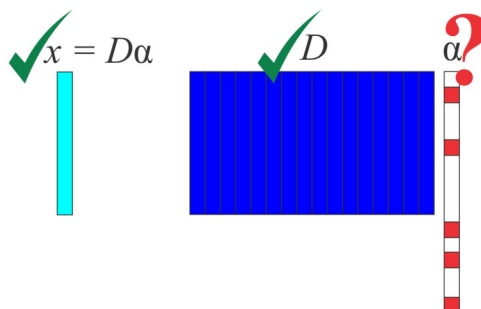
De forma mais geral, é possível representar esparsamente um sinal $x \in \mathbb{R}^m$ sobre um dicionário $D \in \mathbb{R}^{m \times n}$, correspondente a um conjunto de sinais onde cada elemento equivale a uma de n colunas referidas como átomos. O sinal x pode ser encontrado por uma combinação linear de colunas (átomos) do dicionário D , conforme a definição matemática a seguir, onde $\|\alpha\|_0 < k$, e $\|\cdot\|_0$ representa a norma de regularização l_0 que computa a quantidade de elementos não-nulos no vetor α ([RIBEIRO, 2017](#)).

$$\min_{\alpha} f(\alpha), \text{ tal que } D\alpha = x. \quad (3.1)$$

O conjunto de índices de k colunas selecionadas é chamado de suporte (Λ). Essa modelagem supõe que o sinal x reside em um subespaço de baixa dimensão, que é dado pelos k átomos de D que correspondem ao suporte de α . A Figura 1 demonstra visualmente essa representação, onde o dicionário D representa uma coleção de imagens (sinais), que podem ser combinadas para formar uma nova imagem, conforme Figura 2.

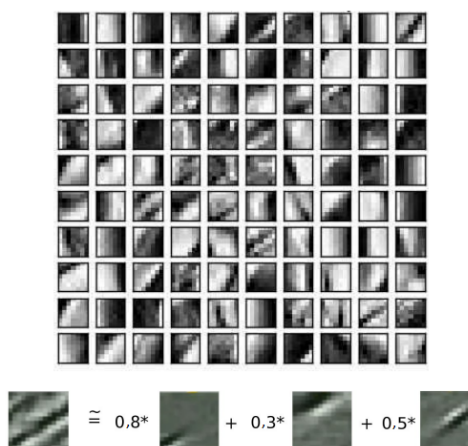
Uma imagem pode ser compreendida como uma matriz, onde cada valor da matriz corresponde a um pixel da imagem. Isso pode ser observado a seguir, onde a Figura 3.3 mostra uma parte da imagem, escolhida para evidenciar a matriz que a representa. Apesar de se terem valores dentro de uma escala (neste caso de 0 a 255, já que as imagens usadas são em escala de cinza), é interessante a normalização dos valores da matriz que representa a imagem. Assim, os valores, após a normalização, ficam entre 0 e 1.

Figura 1 – Modelo visual da representação esparsa de um sinal.



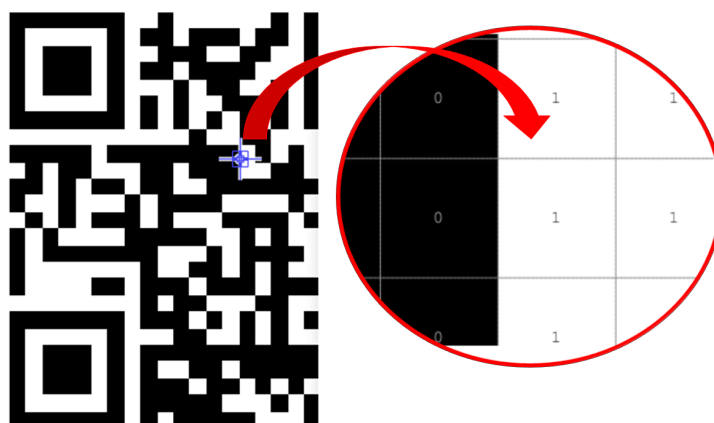
Fonte: Ribeiro (2017).

Figura 2 – Representação de um dicionário.



Fonte: Adaptado de Muniz (2014).

Figura 3 – Pixels de uma imagem.



De acordo com [Chambolle \(2004\)](#), modelo (3.1) é geralmente conhecido pelo nome de Busca de Base. A função $f(\alpha)$ é uma função de indução de esparsidade, e representa o modo como os dados devem ser tratados, de forma a considerar dados relevantes e desconsiderar os irrelevantes. Exemplos dessa função são as normas l_1 , l_2 e l_p .

Os problemas de imagem podem ser estudados a partir de sistemas lineares discretos ([ENGL; HANKE; NEUBAUER, 1996](#)), como mostrado a seguir, onde $A \in \mathbb{R}^{m \times n}$ e $b \in \mathbb{R}^m$ são conhecidos, ϵ é um vetor de ruído (ou perturbação) desconhecido, e x é o sinal/imagem “verdadeiro” e desconhecido a ser estimado ([BECK; TBOULLE, 2009b](#)).

$$Ax = b + \epsilon. \quad (3.2)$$

Alguns tipos de ruído são: Gaussiano, Sal-pimenta e Riciano (esta Tese considera os ruídos Gaussiano e Riciano). O ruído gaussiano é geralmente introduzido durante o processo de aquisição de dados de imagens e pode ser caracterizado pela adição de um valor de distribuição gaussiana de média zero em cada pixel da imagem; veja [Elaiyaraja, Kumaratharan e Rao \(2022\)](#). Em problemas com ruído Sal-pimenta, as imagens são afetadas por um ruído em que o pixel corrompido assume o valor de cinza máximo ou mínimo. Esse ruído aparece como pixels brancos e pretos na imagem corrompida; veja [Liang, Li e Zhao \(2021\)](#). A aquisição de imagens médicas pode ser influenciada pelo funcionamento de equipamentos eletromagnéticos. Neste caso, o ruído Riciano pode estar presente, e a modelagem desse problema é feita pela inserção de uma distribuição riciano; veja [Mohan, Krishnaveni e Guo \(2014\)](#).

A redução de ruído em imagens pode ser feita por métodos classificados como: métodos de domínio de transformação e métodos de domínio espacial. Estes visam remover o ruído pelo cálculo da correlação do valor de cinza de cada pixel, entre as imagens ruidosa e original ([LI et al., 2010](#)). Uma das classes de métodos espaciais são os métodos de redução de ruído variacional, que usam a imagem ruidosa b para obtenção da imagem reconstruída \hat{x} minimizando uma função energia E , como definido a seguir ([FAN et al., 2019](#)):

$$\hat{x} = \arg \min_x E(x).$$

Em problemas com a adição de ruído gaussiano, \hat{x} pode ser definido por:

$$\hat{x} = \arg \min_x \frac{1}{2} \|b - x\|_2^2 + \lambda R(x),$$

onde $\|b - x\|_2^2$ é um termo de fidelidade que denota a diferença entre as imagens originais e com ruído, $R(x)$ é um termo de regularização probabilístico bayesiano e λ é o parâmetro de regularização ([FAN et al., 2019](#)). O estudo das regularizações iniciou com [Tikhonov \(1963\)](#), que define um modelo, mostrado a seguir, considerado simples

e que minimiza a função energia $E(x)$ pela norma l_2 , onde $|\nabla x|$ denota a norma de ∇x em \mathbb{R}^2 , mas suaviza muitos detalhes da imagem (DOBSON; SANTOSA, 1996):

$$E(x) = \frac{1}{2} \|b - x\|_2^2 + \lambda \int |\nabla x|^2.$$

Buscando preservar os detalhes da imagem, foram realizados estudos sobre a difusão anisotrópica por Perona e Malik (1990) e Weickert et al. (1998), mas as bordas das imagens continuavam a estar borradas. Então, um modelo para o processamento de imagens, denominado *Total Variation* (TV), baseado no fato estatístico de que as imagens naturais são localmente suaves e que a intensidade do pixel varia gradualmente na maioria das regiões, foi introduzido por Rudin, Osher e Fatemi (1992). O modelo supõe que a degradação de uma imagem $U \in \mathbb{R}^{n_1 \times n_2}$, representada na forma vetorizada por $u \in \mathbb{R}^n$, onde $n = n_1 \cdot n_2$, pode ser modelada por um ruído aditivo (ϵ), conforme 3.3, onde ϵ é um ruído gaussiano aditivo de média zero e variância σ^2 .

$$u^\diamond = u + \epsilon. \quad (3.3)$$

A transformação de uma matriz em vetor (vetorização) pode ser realizada pela inserção consecutiva de colunas dessa matriz (comando `reshape` do MATLAB), o que significa transformá-la em um vetor linha. Como exemplo, pode-se observar as linhas de código a seguir, onde uma matriz X é transformada em um vetor coluna x :

```
input:
X=[1 2 3 4;5 6 7 8;9 10 11 12;13 14 15 16]
```

```
output:
X =
     1     2     3     4
     5     6     7     8
     9    10    11    12
    13    14    15    16
```

```
input:
x=reshape(X,4*4,1)
```

```
output:
x =[1 5 9 13 2 10 14 3 7 11 15 4 8 12 16]'
```

O TV é um processo/filtro para remoção de ruído (*Denoising*) que se baseia no princípio de que sinais com detalhes excessivos e possivelmente indesejados têm alta variação total. Com essa premissa, reduzir a variação total do sinal remove detalhes

indesejados e preserva detalhes importantes. Esse é o modelo proposto por [Rudin, Osher e Fatemi \(1992\)](#), denominado modelo ROF, que na forma discreta considera uma imagem vetorizada $u \in \mathbb{R}^n$ de dimensão n pixels, oriunda da matriz $U \in \mathbb{R}^{n_1 \times n_2}$, tal que $n = n_1 \cdot n_2$, uma matriz $u^\diamond \in \mathbb{R}^{m \times n}$ como a imagem ruidosa e a regularização TV discreta, correspondente à norma l_1 da norma p dos gradientes da imagem, dados da seguinte forma:

$$\min_u \lambda \|Du\|_{p,1} + \frac{1}{2} \|u - u^\diamond\|_2^2; \quad (3.4)$$

$$\|Du\|_{p,1} = \sum_{i=1, j=2}^{m,n} |(Du_{i,j})|_p = \sum_{i=1, j=2}^{m,n} ((Du_{i,j,1}^p) + (Du_{i,j,2}^p))^{1/p}. \quad (3.5)$$

O parâmetro de regularização λ , que deve ser positivo, desempenha função importante no processo de *Denoising*. Quando $\lambda = 0$, não há suavização e o resultado é o mesmo que minimizar a soma dos quadrados. Para $\lambda \rightarrow \infty$, no entanto, o termo de variação total se torna cada vez mais forte, forçando o resultado a ter variação total menor, à custa de ser menos parecido com o sinal ruidoso. Assim, a escolha do parâmetro de regularização é fundamental para alcançar a quantidade certa de remoção de ruído.

O termo $\|Du\|_{p,1}$ definido em (3.5) corresponde à equação genérica da variação total de um sinal. Atribuindo valores ao parâmetro p , denominamos para $p = 1$ a norma anisotrópica (denotado por TV_{Aniso}), e para $p = 2$, a norma isotrópica (denotado por TV_{Iso}), definidas a seguir ([LOU et al., 2015](#)).

$$TV_{Iso} = \|Du\|_{2,1} = \|\sqrt{|D_x u|^2 + |D_y u|^2}\|_1; \quad (3.6)$$

$$TV_{Aniso} = \|Du\|_1 = \|D_x u\|_1 + \|D_y u\|_1. \quad (3.7)$$

As variáveis D_x e D_y são os operadores de derivada parcial horizontal e vertical, respectivamente, sendo $D = [D_x; D_y]$ o operador gradiente (∇) na configuração discreta. O uso de uma das duas normas está atrelado às características do problema. A norma TV_{Iso} , por exemplo, é utilizada para problemas com aditivos (ruídos) gaussianos brancos, já que reduzem o ruído ao mesmo tempo que suavizam as bordas em maior ou menor grau. A norma TV_{Aniso} , por sua vez, é usada para o caso de remoção de discrepâncias e ruídos impulsivos (sal e pimenta), pois é robusta a tais valores.

3.4 Considerações finais

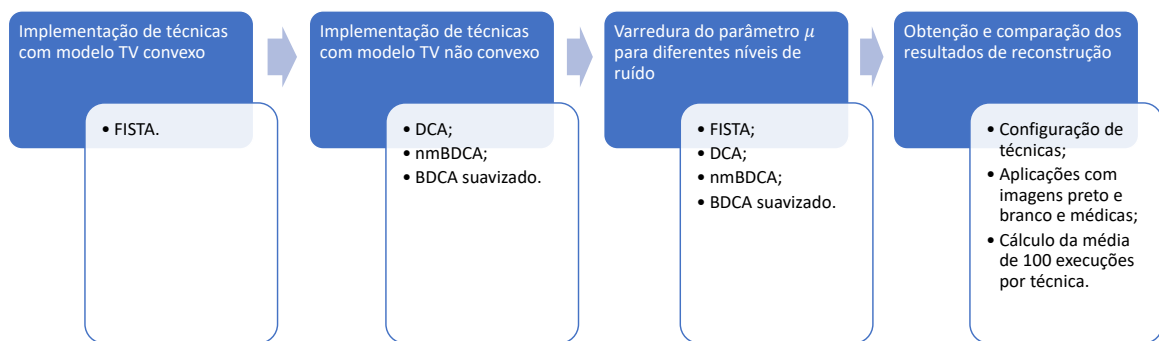
Esse capítulo apresentou conceitos e preceitos importantes para a compreensão das técnicas definidas e explicitadas no capítulo a seguir. A importância se dá não

somente pela compreensão isolada de cada técnica, mas para que fique clara como é feita a integração entre elas.

4 Algoritmos

Nesta seção estão descritas as técnicas implementadas em MATLAB. Conforme fluxograma da Figura 4, as implementações são usadas para reconstruir imagens, permitindo a avaliação das reconstruções e a comparação do desempenho das técnicas aplicadas.

Figura 4 – Fluxograma de execução das implementações.



4.1 *Fast Iterative Shrinkage Thresholding Algorithm (FISTA)*

A reconstrução de imagens, também chamada de remoção de ruído, utiliza com sucesso o modelo de variação total (RUDIN; OSHER; FATEMI, 1992). Nessa representação, conforme dito anteriormente, a imagem original $X \in \mathbb{R}^{n_1 \times n_2}$, com vetorização $x \in \mathbb{R}^n$, com $n = n_1 \cdot n_2$, é considerada uma matriz de *pixels*, e pode sofrer a influência de ruído ϵ (gaussiano, sal-pimenta, por exemplo) e borrão (operador A), passando a ser representada por b , conforme (3.2). Assim, o problema de representação de imagem consiste na determinação de x , e é um problema estudado extensivamente para recuperação de sinal/imagem na literatura através de métodos de regularização (FERNÁNDEZ-MARTÍNEZ et al., 2014; ENGL; HANKE; NEUBAUER, 1996; PEYRÉ; BOUGLEUX; COHEN, 2008).

Como visto anteriormente, uma abordagem eficaz para resolver o problema é baseada em *Total Variation (TV)*, que consiste na obtenção de x a partir da otimização de um modelo. Assim, de forma generalista, define-se A e b e determina-se x para o problema de minimização convexa dado por:

$$\min_x \{ \|Ax - b\|_2^2 + 2\lambda TV(x) \}, \quad (4.1)$$

onde $\lambda > 0$ e $TV(\cdot)$ é uma norma discreta de x , que pode ser escolhida conforme (3.6) e (3.7). Os problemas que desconsideram borrões de imagem consistem em tornar $A = \mathcal{I}$, sendo \mathcal{I} a matriz identidade.

Os métodos iterativos que são extensões do método do gradiente são usados para resolver muitos problemas que envolvem o processamento de sinal/imagem. Esses métodos são atraentes devido à sua simplicidade e eficácia para resolver problemas de grande escala, mas também são conhecidos por convergirem lentamente (BECK; TEBoulLE, 2009c).

Um exemplo de tal método, popular para resolver problemas de processamento de imagem, é o ISTA (*Iterative Shrinkage-Thresholding Algorithms*); veja Daubechies, Defrise e Mol (2004) e Figueiredo e Nowak (2003). O passo geral do ISTA é (BRUCK, 1977):

$$x_{k+1} = \tau_{\lambda t}(x_k - 2tA^\top(Ax_k - b)),$$

onde t é um passo apropriado, $A \in \mathbb{R}^{m \times n}$ é um operador borrão (operador convolução bidimensional), $b \in \mathbb{R}^m$ é uma imagem corrompida por ruído, $x \in \mathbb{R}^n$ é o sinal a ser estimado, que deve ter o mesmo tamanho de b , e $\tau_\alpha : \mathbb{R}^n \rightarrow \mathbb{R}^n$ é o operador definido por:

$$\tau_\alpha(x)_i = (|x_i| - \alpha)_+ \text{sgn}(x_i). \quad (4.2)$$

Considere um problema de minimização irrestrita de uma função continuamente diferenciável $f : \mathbb{R}^n \rightarrow \mathbb{R}$ como:

$$\min\{f(x) : x \in \mathbb{R}^n\}. \quad (4.3)$$

A resolução de (4.3) pode ser realizada pela aplicação do método do gradiente, que gera uma sequência x_k tal que:

$$x_0 \in \mathbb{R}^n, \quad x_k = x_{k-1} - t_k \nabla f(x_{k-1}), \quad (4.4)$$

onde $t_k > 0$ é o passo ajustável.

Sabe-se que (4.4) pode ser vista como uma regularização proximal e uma linearização de f em x_{k-1} (WALTER, 2014; POLYAK, 1987), e escrita de forma equivalente como:

$$x_k = \arg \min_x f(x_{k-1}) + \langle x - x_{k-1}, \nabla f(x_{k-1}) \rangle + \frac{1}{2t_k} \|x - x_{k-1}\|^2.$$

A ideia acima pode ser aplicada ao problema regularizado não diferenciável l_1 , dado por:

$$\min f(x) + \lambda \|x\|_1 : x \in \mathbb{R}^n. \quad (4.5)$$

Dessa forma, obtém-se um esquema iterativo da seguinte forma:

$$x_k = \arg \min_x f(x_{k-1}) + \langle x - x_{k-1}, \nabla f(x_{k-1}) \rangle + \frac{1}{2t_k} \|x - x_{k-1}\|^2 + \lambda \|x\|_1,$$

que pode ser reescrito, ao serem ignorados os termos constantes, por:

$$x_k = \arg \min_x \frac{1}{2t_k} \|x - (x_{k-1} - t_k \nabla f(x_{k-1}))\|^2 + \lambda \|x\|_1,$$

que é um caso especial do regime introduzido em [Passty \(1979\)](#), resolvido por (4.5).

Como a norma l_1 é separável, o cálculo de x_k se reduz a resolver um problema de minimização unidimensional para cada uma de suas componentes, que produz:

$$x_k = T_{\lambda t_k}(x_{k-1} - t_k \nabla f(x_{k-1})),$$

onde $T_\alpha : \mathbb{R}^n \rightarrow \mathbb{R}^n$ é o operador dado em (4.2). Uma condição típica para garantir a convergência de $\{x_k\}$ é exigir que $t_k \in (0, 1/\|A^\top A\|)$. Percebe-se, então, que a ideia do ISTA é construir uma regularização da função diferenciável linearizada em cada iteração. Tomando $f(x) = \|Ax - b\|^2$ em (4.5), o ISTA pode ser aplicado para resolver o problema de imagem com regularização l_1 , ou seja, resolver o seguinte problema:

$$\min\{F(x) = \|Ax - b\|^2 + \lambda \|x\|_1\}. \quad (4.6)$$

De modo geral, pode-se formular o seguinte problema a ser resolvido pelo ISTA:

$$\min\{F(x) = f(x) + g(x) : x \in \mathbb{R}^n\}, \quad (4.7)$$

supondo que:

- $g : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função convexa possivelmente não diferenciável;
- $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função convexa, continuamente diferenciável com gradiente Lipschitz contínuo, isto é:
 - * $\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|$, para todo x e $y \in \mathbb{R}^n$, onde $L > 0$ é a constante de Lipschitz do ∇f ;
- O problema (4.7) tem solução, ou seja, $X_* := \arg \min F \neq \emptyset$, e para $x^* \in X_*$, denotamos $F_* := F(x^*)$.

Um algoritmo para resolução do problema (4.7) pelo ISTA é apresentado no Algoritmo 1, onde as constantes `tol`, `max_iter` e `cont` são, respectivamente, tolerância de erro entre o valor calculado e o esperado, máxima iteração do método e contador de vezes que a tolerância foi alcançada, e p_L é um minimizador dado por:

$$p_L(y) := \arg \min\{Q_L(x,y) : x \in \mathbb{R}^n\},$$

onde:

$$Q_L(x,y) := f(y) + \langle x - y, \nabla f(y) \rangle + \frac{L}{2} \|x - y\|^2 + g(x).$$

Desconsiderando os termos constantes em y , algumas manipulações permitem definir p_L como:

$$p_L(y) = \arg \min_x \left\{ g(x) + \frac{L}{2} \left\| x - \left(y - \frac{1}{L} \nabla f(y) \right) \right\|^2 \right\}.$$

Dessa forma, com L sendo o tamanho de passo, o passo geral do ISTA corresponde a:

$$x_k = p_L(x_{k-1}).$$

Algorithm 1 ISTA com passo fixo

```

1: Considerando  $L$  a constante de Lipschitz do  $\nabla f$ ,  $x_0 \in \mathbb{R}^n$ ;
2: Definir  $\text{max\_iter}^a$ ,  $\text{tol}^a$ ,  $\text{cont}=0$ ,  $k=1$ ;
3: while  $k < \text{max\_iter}$  e  $\text{cont}^b < 5$  do
4:    $x_k = p_L(x_{k-1})$ ;
5:   if  $\frac{\|x_k - x_{k-1}\|}{\|x_k\|} < \text{tol}$  then
6:      $\text{cont} = \text{cont} + 1$ ;
7:   end if
8:    $x_{k-1} = x_k$ ;
9:    $k = k + 1$ ;
10: end while

```

^a As variáveis max_iter e tol são definidas *a priori*. Neste trabalho foram usados $\text{max_iter} = 200$ e $\text{tol} = 10^{-4}$;

^b A variável cont é definida no algoritmo FISTA e tem valor 5. Esse valor é definido pelos autores do algoritmo em Beck e Teboulle (2009b), que considera a convergência quando a solução está dentro do critério de tolerância por 5 iterações seguidas.

O algoritmo geral do ISTA (Algoritmo 1) é útil quando $p_L(\cdot)$ pode ser calculado analiticamente. Isso ocorre particularmente quando $g(\cdot)$ é separável, e nesse caso, o cálculo de p_L é a resolução de um problema de minimização unidimensional. Um exemplo é $g(\cdot)$ sendo a p -ésima potência de uma norma de x , com $p \geq 1$. Essa situação pode ser analisada com mais profundidade em alguns trabalhos, como Combettes e Wajs (2005), Moulin e Liu (1998).

Uma possível desvantagem do Algoritmo 1 é que a constante de Lipschitz L nem sempre é conhecida ou computável. Por exemplo, a constante de Lipschitz no problema de regularização (4.6) depende do autovalor máximo de $A^\top A$, onde para problemas de grande escala, nem sempre é facilmente calculável. Assim, tem-se a necessidade da análise do ISTA com uma regra de ajuste de passo, explicitado no Algoritmo 2.

A sequência $\{F(x_k)\}$ produzida pelo ISTA é não crescente. De fato, para cada $k \geq 1$:

$$F(x_k) \leq Q_{L_k}(x_k, x_{k-1}) \leq Q_{L_k}(x_{k-1}, x_{k-1}) = F(x_{k-1}),$$

onde L_k é escolhido pela regra de ajuste ou $L_k = L$ é a constante de Lipschitz do ∇f .

A análise de convergência do ISTA foi avaliada para os problemas (4.6) e (4.7), considerando condições que asseguram a convergência da sequência $\{x_k\}$, no trabalho de Beck e Teboulle (2009b), e constata que a complexidade de pior caso do ISTA é da ordem de $O(1/k)$, conforme teorema enunciado a seguir.

Algorithm 2 ISTA com passo ajustável

```

1: Considerando  $L_0 > 0, \eta > 1, x_0 \in \mathbb{R}^n$ ;
2: Definir  $\text{max\_iter}, \text{tol}, \text{cont} = 0, k = 1$ ;
3: while  $k < \text{max\_iter}$  e  $\text{cont} < 5$  do
4:    $\bar{L} = \eta^{ik} L_{k-1}$ ;
5:    $F(p_{\bar{L}}(x_{k-1})) \leq Q_{\bar{L}}(p_{\bar{L}}(x_{k-1}), x_{k-1})$ ;
6:    $L = \eta^{ik} L_{k-1}$ ;
7:    $x_k = p_L(x_{k-1})$ ;
8:   if  $\frac{\|x_k - x_{k-1}\|}{\|x_k\|} < \text{tol}$  then
9:      $\text{cont} = \text{cont} + 1$ ;
10:  end if
11:   $x_{k-1} = x_k$ ;
12:   $k = k + 1$ ;
13: end while

```

Teorema 4.1. *Seja $\{x_k\}$ a sequência gerada pelos Algoritmos 1 ou 2. Então para qualquer $k \geq 1$:*

$$F(x_k) - F(x^*) \leq \frac{\alpha L \|x_0 - x^*\|}{2k}, \quad \forall x^* \in X_*,$$

onde $\alpha = 1$ para passo fixo, ou $\alpha = \eta$ para passo ajustável.

Demonstração. Veja Beck e Teboulle (2009b, Página 192). □

Quando $g(x) = 0$, o modelo geral (4.7) consiste em minimizar uma função convexa, que reduz o ISTA ao método do gradiente. Neste cenário, diferenciável, existe um método de gradiente de primeira ordem “ótimo” (NESTEROV, 1983). O interessante é que esse método não requer mais do que uma avaliação de gradiente em cada iteração (o mesmo que o método do gradiente), mas apenas um ponto adicional a ser calculado. A extensão geral desse método para o problema (4.7), denominada FISTA (*Fast Iterative Shrinkage Thresholding Algorithm*), é explicitada pelo Algoritmo 3, cujo passo é fixo (BECK; TBOULLE, 2009b).

Algorithm 3 FISTA com passo fixo

```

1: Considerando  $L := L$  a constante de Lipschitz do  $\nabla f, y_1 = x_0 \in \mathbb{R}^n, t_1 = 1$ ;
2: Definir  $\text{max\_iter}, \text{tol}, \text{cont} = 0, k = 1$ ;
3: while  $k < \text{max\_iter}$  e  $\text{cont} < 5$  do
4:    $x_k = p_L(y_k)$ ;
5:    $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$ ;
6:    $y_{k+1} = x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1})$ ;
7:   if  $\frac{\|x_k - x_{k-1}\|}{\|x_k\|} < \text{tol}$  then
8:      $\text{cont} = \text{cont} + 1$ ;
9:   end if
10:   $k = k + 1$ ;
11: end while

```

Em razão da desvantagem do ISTA, já mencionada, o FISTA com passo ajustável é definido pelo Algoritmo 4. A principal diferença entre o FISTA e o ISTA é que o operador $p_L(\cdot)$ não é empregado no ponto anterior x_{k-1} , mas sim no ponto y_k que usa uma combinação linear dos dois pontos anteriores (x_{k-1}, x_{k-2}) . Como resultado, tem-se

a complexidade melhorada do FISTA, da ordem $O(1/k^2)$, conforme enuncia o teorema a seguir.

Teorema 4.2. *As sequências $\{x_k\}$, $\{y_k\}$ geradas pelo FISTA, satisfazem para cada $k \geq 1$:*

$$F(x_k) - F(x^*) \leq \frac{2\alpha L \|x_0 - x^*\|}{(k+1)^2}, \quad \forall x^* \in X_*,$$

onde $\alpha = 1$ para passo fixo, ou $\alpha = \eta$ para passo ajustável.

Demonstração. Veja Beck e Teboulle (2009b, Página 196). □

Algorithm 4 FISTA com passo ajustável

- 1: Considerando $L_0 > 0$, $\eta > 1$, $x_0 \in \mathbb{R}^n$;
- 2: Definir $y_1 = x_0$, $t_1 = 1$, max_iter , tol , $\text{cont} = 0$, $k = 1$;
- 3: **while** $k < \text{max_iter}$ e $\text{cont} < 5$ **do**
- 4: Calcular o menor índice não negativo i_k tal que com $\bar{L} = \eta^{i_k} L_{k-1}$, tem-se:

$$F(p_{\bar{L}}(y_k)) \leq Q_{\bar{L}}(p_{\bar{L}}(y_k), y_k);$$

- 5: $L_k = \eta^{i_k} L_{k-1}$;
 - 6: $x_k = p_{L_k}(y_k)$;
 - 7: $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$;
 - 8: $y_{k+1} = x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1})$;
 - 9: **if** $\frac{\|x_k - x_{k-1}\|}{\|x_k\|} < \text{tol}$ **then**
 - 10: $\text{cont} = \text{cont} + 1$;
 - 11: **end if**
 - 12: $k = k + 1$;
 - 13: $L_{k-1} = L_k$;
 - 14: **end while**
-

O problema (4.1) pode ser tratado pelo FISTA a partir da soma de funções, tal como:

$$\min\{F(x) = f(x) + g(x) : x \in \mathbb{R}^n\}, \quad (4.8)$$

onde:

- $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é continuamente diferenciável com gradiente contínuo Lipschitz $L(f)$, tal que: $\|\nabla f(x) - \nabla f(y)\| \leq L(f) \|x - y\|$, para todo x e $y \in \mathbb{R}^n$;
- $\|\cdot\|$ é a norma euclidiana e $L(f) > 0$ é a constante Lipschitz de ∇f ;
- $g : \mathbb{R}^n \rightarrow \mathbb{R}$, é uma função convexa adequada ao problema.

Assim, levando-se em consideração (4.1) e (4.8), pode-se definir um problema para remoção de ruído (*Denoising*) com o FISTA por:

$$\min_{x \in \mathcal{C}} \|x - b\|^2 + 2\lambda TV(x), \quad (4.9)$$

onde $TV(x)$ pode ser a norma isotrópica ou anisotrópica, $A = \mathcal{I}$, sendo \mathcal{I} a matriz identidade, e f e g são:

$$f(x) \equiv \|Ax - b\|^2, \quad g(x) \equiv 2\lambda TV(x) + \delta_C(x),$$

onde $x \in \mathbb{R}^n$ é uma imagem a ser restaurada, b é a imagem ruidosa, $\lambda > 0$ é um parâmetro de regularização; veja Beck e Teboulle (2009a, Seção II). Um caso particular é $C = B_{l,u}$, onde $B_{l,u}$ é o cubo n -dimensional dado por

$$B_{l,u} = \{x : l \leq x_{i,j} \leq u, \quad \forall i,j\}.$$

Isso é particularmente útil para modelar a situação em que os pixels têm limites inferiores e superiores, por exemplo, entre 0 e 255 ou 0 e 1.

Uma versão do FISTA (Algoritmo 5), que resolve o modelo (4.9), denominada *Fast Projected Gradient* (FGP), foi proposta por Beck e Teboulle (2009a) e consiste em uma versão melhorada dos Algoritmos 3 e 4. Essa técnica equivale à combinação do FISTA com o método de projeção de gradiente proposto por Nesterov (1983). De acordo com Beck e Teboulle (2009a), o FGP combina simplicidade e taxa de convergência global melhor do que os métodos baseados em projeções de gradiente atualmente conhecidos, podendo lidar com restrições vinculadas, e com as funções de TV isotrópicas e anisotrópicas.

Algorithm 5 FGP(B, λ)

```

1: Receber a imagem  $B \in \mathbb{R}^{n_1 \times n_2}$ ;
2: Definir  $\text{max\_iter}$ ,  $\text{tol}$ ,  $\text{cont}=0$ ,  $k = 1$ ;
3:  $(r_1, s_1) = (p_0, q_0) = (0_{(m-1) \times n}, 0_{m \times (n-1)})$ ,  $t_1 = 1$ ;
4: while  $k < \text{max\_iter}$  e  $\text{cont} < 5$  do
5:   Calcular  $(p_k, q_k) = PP \left[ (r_k, s_k) + \frac{1}{8\lambda} \mathcal{L}^T (P_C [B - \lambda \mathcal{L}(r_k, s_k)]) \right]$ ;
6:    $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$ ;
7:    $(r_{k+1}, s_{k+1}) = (p_k, q_k) + \left( \frac{t_k - 1}{t_{k+1}} \right) (p_k - p_{k-1}, q_k - q_{k-1})$ ;
8:   Calcular uma solução ótima  $x_k = P_C [B - \lambda \mathcal{L}(p_k, q_k)]$ ;
9:   if  $\frac{\|x_k - x_{k-1}\|}{\|x_k\|} < \text{tol}$  then
10:      $\text{cont} = \text{cont} + 1$ ;
11:   end if
12:    $k = k + 1$ ;
13: end while

```

onde:

- P é o conjunto de pares de matrizes (p, q) , tal que $p \in \mathbb{R}^{(m-1) \times n}$ e $q \in \mathbb{R}^{m \times (n-1)}$ que satisfaz:
 - * $p_{i,j}^2 + q_{i,j}^2 \leq 1, i = 1, \dots, m-1, j = 1, \dots, n-1$;
 - * $|p_{i,n}| \leq 1, i = 1, \dots, m-1$;

$$* |q_{m,j}| \leq 1, j = 1, \dots, n-1.$$

- P_P é uma projeção dada por $P_P(p,q) = (r,s)$, tal que:

$$* r_{i,j} = \begin{cases} \frac{p_{i,j}}{\max\left(1, \sqrt{p_{i,j}^2 + q_{i,j}^2}\right)}, & i = 1, \dots, m-1, j = 1, \dots, n-1 \\ \frac{p_{i,n}}{\max\left(1, |p_{i,n}|\right)}, & i = 1, \dots, m-1 \end{cases}$$

$$* s_{i,j} = \begin{cases} \frac{q_{i,j}}{\max\left(1, \sqrt{p_{i,j}^2 + q_{i,j}^2}\right)}, & i = 1, \dots, m-1, j = 1, \dots, n-1 \\ \frac{q_{m,j}}{\max\left(1, |q_{m,j}|\right)}, & j = 1, \dots, n-1 \end{cases}$$

- \mathcal{L} é uma operação linear $\mathcal{L} : \mathbb{R}^{(m-1) \times n} \times \mathbb{R}^{m \times (n-1)} \rightarrow \mathbb{R}^{m \times n}$ definida por:

$$* \text{Supondo } \mathcal{L}(p,q)_{i,j} = p_{i,j} + q_{i,j} - p_{i-1,j} - q_{i,j-1}, \text{ para } i = 1, \dots, m-1, j = 1, \dots, n-1, \text{ assumindo que } p_{0,j} = p_{m,j} = q_{i,0} = q_{i,n} = 0.$$

- O operador linear $\mathcal{L}^T : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{(m-1) \times n} \times \mathbb{R}^{m \times (n-1)}$ é dado por:

$$* \mathcal{L}^T(x) = (p,q), \text{ onde } p \in \mathbb{R}^{(m-1) \times n} \text{ e } q \in \mathbb{R}^{m \times (n-1)} \text{ são matrizes definidas por:}$$

$$\triangleright p_{i,j} = x_{i,j} - x_{i+1,j}, i = 1, \dots, m-1, j = 1, \dots, n;$$

$$\triangleright q_{i,j} = x_{i,j} - x_{i,j+1}, i = 1, \dots, m, j = 1, \dots, n-1.$$

- P_C é o operador projeção ortogonal no conjunto C . Por exemplo, para $C = B_{l,u}$, $P_{B_{l,u}}$ é:

$$* P_{B_{l,u}}(x)_{i,j} = \begin{cases} l, & \text{caso } x_{i,j} < l \\ x_{i,j}, & \text{caso } l \leq x_{i,j} \leq u \\ u, & \text{caso } x_{i,j} > u. \end{cases}$$

A aplicação do FISTA com FGP (FISTA-FGP) para os problemas de imagem neste trabalho é mostrado no Algoritmo 6.

Algorithm 6 FISTA-FGP(X, λ)

- 1: Carregar imagem $X \in \mathbb{R}^{n_1 \times n_2}$;
 - 2: Criar imagem $x \in \mathbb{R}^n$, onde $n = n_1 \cdot n_2$;
 - 3: Definir ruído ϵ e calcular $b = x + \epsilon$;
 - 4: Definir λ , tipo de TV;
 - 5: Criar $B \in \mathbb{R}^{n_1 \times n_2}$;
 - 6: Obter uma solução ótima $x_k = \text{FGP}(B, \lambda)$;
-

4.2 O algoritmo DCA

Os modelos de representação de imagem apresentados em (3.4) e (4.1) são problemas de otimização convexa, e portanto, podem ser representados de maneira que possam ser utilizados por DCA, conforme aplicado em Nguyen (2014). O DCA foi introduzido por Tao e Souad (1986), é a base da programação DC, e consiste em substituir a segunda componente convexa por uma aproximação linear. Assim, o DCA introduz o conceito de aproximar um problema não convexo, onde cada iteração do algoritmo requer a solução de um problema convexo.

Um problema geral DC possui a forma:

$$\min_{x \in \mathbb{R}^n} f(x) := g(x) - h(x), \quad (4.10)$$

onde g e h são funções convexas em \mathbb{R}^n . A função f é chamada de função DC, e $g - h$ uma decomposição DC de f .

Definição 4.1. Um ponto $x^* \in \mathbb{R}^n$ é ponto crítico de $\phi(x) = g(x) - h(x)$, como em (4.10), se:

$$\partial g(x^*) \cap \partial h(x^*) \neq \emptyset.$$

O DCA consiste em construir duas sequências $\{x^k\}$ e $\{y^k\}$, tais que $y^k \in \partial h(x^k)$ e $x^{k+1} \in \partial g^*(y^k)$. Pela Proposição 3.9, tem-se que cada iteração do DCA aproxima a parte côncava $-h$ e minimiza a função convexa resultante. O algoritmo geral do DCA é apresentado a seguir.

Algorithm 7 DCA

1: Definir $x^0 \in \mathbb{R}^n$, $k = 0$, tol , $\text{stop_rule} = +\infty$;

2: **while** $\text{stop_rule} > \text{tol}$ **do**

3: Calcular $w^k \in \partial h(x^k)$;

4: Calcular

$$y^k \in \arg \min \{g(x) - \langle w^k, x - x^k \rangle : x \in \mathbb{R}^n\}; \quad (4.11)$$

5: $\text{stop_rule} = \|y^k - x^k\|$;

6: $x^{k+1} = y^k$;

7: $k = k + 1$.

8: **end while**

Observa-se, do Algoritmo 7, que a cada iteração k , o DCA obtém o próximo ponto de iteração x^{k+1} resolvendo o subproblema convexo $x^{k+1} \in \arg \min \{g(x) - \langle w^k, x - x^k \rangle\}$, para $x \in \mathbb{R}^n$. Esse subproblema necessita ser resolvido por uma técnica de minimização, que neste trabalho é representada pela técnica FISTA. Algumas características do DCA são:

- O DCA é um método de descida sem a necessidade de empregar qualquer tipo de busca (*line search*);

- Se o valor ótimo do problema é finito e as sequências $\{x^k\}$ e $\{y^k\}$ são limitadas, então todo ponto de acumulação x^* da sequência $\{x^k\}$ é um ponto crítico de f ;
- Para algumas situações específicas o DCA tem convergência linear e convergência finita para programas DC poliedrais.

A definição precisa do DCA é uma questão crucial e altamente dependente da natureza intrínseca da decomposição DC. De acordo com [Niu \(2022\)](#), uma análise sobre a convergência do DCA deve ser feita a partir de algumas considerações definidas a seguir, complementares entre si, onde as considerações iniciais não são suficientes para garantir a boa definição do DCA e, por isso, necessitam das complementares; ver [Niu \(2022, Páginas 4 e 5\)](#).

- Considerações iniciais:

(H1) $g, h : \mathbb{R}^n \rightarrow \mathbb{R}$ são ambas fortemente convexas com módulo $\rho > 0$;

(H2) $f^* := \inf_{x \in \mathbb{R}^n} \{f(x) = g(x) - h(x)\} > -\infty$.

(H3) o conjunto solução do problema (4.10) é não vazio.

A hipótese **(H1)** não é teoricamente restritiva. Sabe-se que dadas duas funções convexas, g e h , pode-se adicionar um termo fortemente convexo $\frac{\rho}{2}\|x\|^2$ para se obter uma nova decomposição, com componentes fortemente convexas com módulo $\rho > 0$. As hipóteses **(H2)** e **(H3)** são comuns no contexto da programação DC; veja [Artacho, Fleming e Vuong \(2018\)](#), [Artacho e Vuong \(2020\)](#), [Neto et al. \(2020\)](#), [Ferreira, Santos e Souza \(2024\)](#).

- Considerações complementares:

* todos os subproblemas de (4.11) têm soluções ótimas (sob a hipótese **(H1)** o subproblema tem solução única).

Admitindo-se, então, que a sequência $\{x_k\}$ gerada pelo DCA, partindo de um ponto inicial $x^0 \in \mathbb{R}^n$, é bem definida, tem-se os seguintes resultados de convergência. Esses resultados são demonstrados em [Tao e Souad \(1986\)](#). As demonstrações são apresentadas no Capítulo 7 (Apêndices), por questões de completude.

Lema 4.1. *Se $y^k = x^k$ para algum $k \in \mathbb{N}$, então x^k é ponto crítico de f .*

Proposição 4.1. *A sequência $\{x^k\}_{k \in \mathbb{N}}$ gerada pelo Algoritmo 7 satisfaz uma das seguintes afirmações:*

- (i) *O Algoritmo 7 termina em um ponto crítico;*

(ii) A sequência $\{f(x^k)\}_{k \in \mathbb{N}}$ é decrescente, isto é, $f(x^{k+1}) < f(x^k)$, para todo $k \in \mathbb{N}$.

Proposição 4.2. Se $\{x^k\}_{k \in \mathbb{N}}$ é uma sequência gerada pelo Algoritmo 7, então

$$\sum_{k=0}^{+\infty} \|x^k - x^{k+1}\|^2 < +\infty,$$

e $\|x^k - x^{k+1}\| \rightarrow 0$ quando $k \rightarrow +\infty$.

Teorema 4.3. Cada ponto de acumulação da sequência $\{x^k\}_{k \in \mathbb{N}}$ gerado pelo Algoritmo 7, se houver, é um ponto crítico.

Propriedades de convergência e fundamentos teóricos detalhados do DCA podem ser encontrados em [An e Tao \(2005\)](#) e [Tao e An \(1997\)](#). É crucial notar que o DCA é construído a partir de componentes DC, que desempenham papel crítico na determinação da velocidade de convergência, estabilidade, robustez e globalidade das soluções buscadas.

4.3 Non-monotone boosted DC Algorithm (nmBDCA)

Embora o DCA seja um método usado em muitas aplicações, [Artacho, Fleming e Vuong \(2018\)](#) propôs um algoritmo que acelera a convergência do DCA, o *Boosted DC algorithm* (BDCA). O método resolve o mesmo problema do DCA, mas realiza uma busca linear com a regra do tipo Armijo, evitando soluções locais ruins e, assim, fornecendo melhores soluções.

A ideia principal do BDCA é definir uma direção de descida usando o ponto calculado pelo DCA e uma busca linear para dar um passo maior que o DCA, obtendo mais diminuição no valor objetivo em cada iteração. O uso do BDCA requer que a primeira componente DC seja diferenciável, para garantir uma direção descendente do ponto calculado pelo DCA, caso contrário, a direção pode ser ascendente. O Algoritmo 8 apresenta o BDCA. Propriedades de convergência e os fundamentos teóricos do BDCA podem ser encontrados em [Artacho, Fleming e Vuong \(2018, Seção 3\)](#).

Observa-se que se $t_{k-1} = 0$, então as iterações do BDCA coincidem com as iterações do DCA. Neste caso particular, os resultados da convergência se aplicam ao DCA. Pela Proposição 4.1 que, definindo $x^{k+1} := y^k$, a imagem de f sofre uma diminuição, ou seja, $f(y^k) \leq f(x^k)$. Além disso, tomando x^{k+1} na direção de d^k a partir de y^k , alcança-se uma diminuição ainda maior. Este fato é a ideia principal por trás do BDCA e melhora o desempenho do DCA em muitas aplicações.

Adicionalmente às hipóteses feitas ao DCA, o BDCA considera a seguinte suposição.

Algorithm 8 BDCA

-
- 1: Definir $t_{-1} > 0$, $\rho > 0$, $\zeta \in (0,1)$, $k = 0$, tol , $\text{stop_rule} = +\infty$;
 - 2: Definir um ponto inicial $x^0 \in \mathbb{R}^n$;
 - 3: **while** $\text{stop_rule} > \text{tol}$ **do**
 - 4: Calcular $w^k \in \partial h(x^k)$ e obter:

$$y^k \in \arg \min \{g(x) - \langle w^k, x - x^k \rangle : x \in \mathbb{R}^n\}; \quad (4.12)$$

- 5: Calcular $d^k = y^k - x^k$;
- 6: **if** $d^k = 0$ **then**
- 7: Pare e retorne o ponto x^k ;
- 8: **else**
- 9: Calcular $t_k = \zeta^{j_k} t_{k-1}$, onde:

$$j_k = \min \{j \in \mathbb{N} : \phi(y^k + \zeta^j t_{k-1} d^k) \leq \phi(y^k) - \rho (\zeta^j t_{k-1})^2 \|d^k\|^2\};$$

- 10: **end if**
 - 11: Calcular $x^{k+1} = y^k + t_k d^k$;
 - 12: $\text{stop_rule} = \|x^{k+1} - x^k\|$;
 - 13: $k = k + 1$;
 - 14: **end while**
-

(H4) $g : \mathbb{R}^n \rightarrow \mathbb{R}$ é diferenciável.

A estrutura de busca do tamanho do passo considerada no BDCA requer a suposição **(H4)**; caso contrário, a direção obtida poderá ser ascendente (veja [Artacho e Vuong \(2020, Exemplo 3.4\)](#)).

Os resultados de convergência do Algoritmo 8 estão em [Artacho, Fleming e Vuong \(2018\)](#) e [Artacho e Vuong \(2020\)](#). As demonstrações são mostradas no Capítulo 7 (Apêndices), por questões de completude.

Proposição 4.3. *Para todo $k \in \mathbb{N}$, vale o seguinte:*

- (i) $f(y^k) \leq f(x^k) - \rho \|d^k\|^2$;
- (ii) $f'(y^k; d^k) \leq -\rho \|d^k\|^2$;
- (iii) *Para todo $k \in \mathbb{N}$, existe algum $\delta_k > 0$ tal que*

$$f(y^k + \lambda d^k) \leq f(y^k) - \alpha \lambda^2 \|d^k\|, \quad \forall \lambda \in [0, \delta_k],$$

e, portanto, a busca linear no passo 3 do BDCA termina finitamente.

Lema 4.2. *Se $y^k = x^k$ para algum $k \in \mathbb{N}$, então x^k é ponto crítico de f .*

Proposição 4.4. *A sequência $\{x^k\}_{k \in \mathbb{N}}$ gerada pelo Algoritmo 8 satisfaz uma das seguintes afirmações:*

- (i) *O Algoritmo 8 termina em um ponto crítico;*

(ii) A sequência $\{f(x^k)\}_{k \in \mathbb{N}}$ está diminuindo, ou seja, $f(x^{k+1}) \leq f(x^k)$, para todo $k \in \mathbb{N}$.

Corolário 4.1. Se $\{x^k\}_{k \in \mathbb{N}}$ é uma sequência gerada pelo Algoritmo 8, então a sequência $\{f(x^k)\}_{k \in \mathbb{N}}$ é convergente.

Proposição 4.5. Se $\{x^k\}_{k \in \mathbb{N}}$ é uma sequência gerada pelo Algoritmo 8, então as seguintes afirmações são válidas:

(i) $\sum_{k=0}^{+\infty} \|d^k\|^2 < +\infty$. Em particular, $\|y^k - x^k\| \rightarrow 0$ quando $k \rightarrow +\infty$.

(ii) Se $\{\lambda_k\}_{k \in \mathbb{N}}$ é limitado, então $\sum_{k=0}^{+\infty} \|x^{k+1} - x^k\|^2 < +\infty$. Em particular, $\|x^{k+1} - x^k\| \rightarrow 0$ quando $k \rightarrow +\infty$.

Teorema 4.4. Cada ponto de acumulação de $\{x^k\}_{k \in \mathbb{N}}$ gerado pelo Algoritmo 8, se houver, é um ponto crítico de f .

A necessidade de diferenciabilidade da primeira componente DC no BDCA é uma suposição restritiva, que limita a aplicação do BDCA a modelos cuja primeira componente é diferenciável, ou seja, o BDCA não pode ser aplicado a problemas não diferenciáveis e não convexos, cujas duas componentes DC sejam não diferenciáveis. Para superar essa desvantagem, uma versão não diferenciável do BDCA foi proposta por [Ferreira, Santos e Souza \(2024\)](#), chamado de *non-monotone boosted DC Algorithm* (nmBDCA), que usa uma busca linear não monótona no BDCA, para permitir um possível crescimento nos valores da função objetivo controlados por um parâmetro. O Algoritmo 9 apresenta o nmBDCA geral.

O nmBDCA, que corresponde a uma versão não monótona do BDCA para resolver (4.10) com ambos os componentes DC possivelmente não diferenciáveis, gera, então, uma sequência $(x^k)_{k \in \mathbb{N}}$, que pode ser aplicado tanto para casos com g é diferenciável ou não. Se g é não diferenciável, supõe-se que $\nu_k > 0$, para todo $k \in \mathbb{N}$ e, portanto, estende-se os resultados do BDCA para o contexto não diferenciável. Se g for diferenciável, supõe-se que $\nu_k \geq 0$, para todo $k \in \mathbb{N}$. Neste caso, se $\nu_k = 0$, para todo $k \in \mathbb{N}$, então a busca linear não monótona do nmBDCA coincide com a busca monótona do BDCA. Caso contrário, se $\nu_k > 0$, para todo $k \in \mathbb{N}$, então nmBDCA pode ser visto como uma versão inexata do BDCA. É importante ressaltar que o problema resolvido pelo nmBDCA (passo 4 do Algoritmo 9) sempre tem uma solução única y_k (supondo **H1**), que é caracterizada por:

$$w^k \in \partial g(y^k), \forall k \in \mathbb{N}. \quad (4.13)$$

Algorithm 9 nmBDCA

- 1: Definir $t_{-1} > 0$, $\rho > 0$ e $\zeta \in (0,1)$, $k = 0$, tol , $\text{stop_rule} = \infty$;
- 2: Escolher um ponto inicial $x^0 \in \mathbb{R}^n$;
- 3: **while** $\text{stop_rule} > \text{tol}$ **do**
- 4: Calcular $w^k \in \partial h(x^k)$ e obter uma solução y^k para o problema:

$$y^k = \min_{x \in \mathbb{R}^n} \psi_k(x) = g(x) - \langle w^k, x - x^k \rangle;$$

- 5: Calcular $d^k = y^k - x^k$;
- 6: **if** $d^k = 0$ **then**
- 7: Parar e retornar x^k
- 8: **else**
- 9: Tomar $\nu_k \in \mathbb{R}_+$ (especificado antes) e calcular $t_k := \zeta^{j_k} t_{k-1}$, onde:

$$j_k = \min \{j \in \mathbb{N} : \phi(y^k + \zeta^j t_{k-1} d^k) \leq \phi(y^k) - \rho (\zeta^j t_{k-1})^2 \|d^k\|^2 + \nu_k\};$$

- 10: **end if**
- 11: Calcular $x^{k+1} = y^k + t_k d^k$;
- 12: $\text{stop_rule} = \|x^{k+1} - x^k\|$;
- 13: $k = k + 1$;
- 14: **end while**

O passo 5 do Algoritmo 9 tem um termo adicional $(\nu_k)_{k \in \mathbb{N}}$ na busca de Armijo, correspondente a uma sequência de parâmetros escolhida conforme as estratégias a seguir; veja [Ferreira, Santos e Souza \(2024, Seção 3.2\)](#).

- (E1)** Dado $\delta_{\min} \in [0,1)$, a sequência $(\nu_k)_{k \in \mathbb{N}} \subset \mathbb{R}_+$ é definida como: $\nu_0 \geq 0$ e ν_{k+1} , para cada $\delta_{k+1} \in [\delta_{\min}, 1]$, que satisfaz a condição $0 \leq \nu_{k+1} \leq (1 - \delta_{k+1})(f(x^k) - f(x^{k+1}) + \nu_k)$, $\forall k \in \mathbb{N}$;
- (E2)** $(\nu_k)_{k \in \mathbb{N}} \subset \mathbb{R}_+$ é tal que $\sum_{k=0}^{+\infty} \nu_k < +\infty$;
- (E3)** $(\nu_k)_{k \in \mathbb{N}} \subset \mathbb{R}_+$ é tal que para cada $\delta > 0$, existe k_0 tal que $\nu_k \leq \delta \|d^k\|^2$, para todo $k \geq k_0$.

Neste trabalho é considerado a escolha de $(\nu_k)_{k \in \mathbb{N}}$ como em **(E3)**. Portanto, limita-se aos resultados de convergência para esse caso.

Os resultados de convergência do Algoritmo 9 são demonstrados em [Ferreira, Santos e Souza \(2024\)](#). As demonstrações são apresentadas no Capítulo 7 (Apêndices), por questões de completude.

Teorema 4.5. *Se $\lim_{k \rightarrow \infty} \|d^k\| = 0$, então cada ponto de acumulação de $(x^k)_{k \in \mathbb{N}}$, se houver, é um ponto crítico de f .*

Proposição 4.6. *Seja $\{x^k\}_{k \in \mathbb{N}}$ a sequência gerada pelo Algoritmo 9. Suponha que $d^k \neq 0$ e $\nu_k > 0$ para cada $k \in \mathbb{N}$. Então, valem as seguintes afirmações:*

(i) *É válido que $\bar{\delta}_k := \nu_k / (g(y^k + d^k) + g(x^k) - 2g(y^k)) > 0$, e*

$$f(y^k + \lambda d^k) \leq f(y^k) - \alpha \lambda^2 \|d^k\|^2 + \nu_k, \quad \forall \lambda \in (0, \bar{\delta}_k],$$

onde $\delta_k := \min \left\{ \bar{\delta}_k, 1, \frac{3\rho}{2\alpha} \right\}$. Consequentemente, a busca linear no passo 3 está bem definida.

(ii) $f(x^{k+1}) \leq f(x^k) - (\rho + \alpha\lambda_k^2)\|d^k\|^2 + v_k$, para todo $k \in \mathbb{N}$.

Teorema 4.6. Se a sequência $\{v_k\}_{k \in \mathbb{N}} \subset \mathbb{R}_{++}$ for escolhida de acordo com a estratégia **(E3)**, então cada ponto de acumulação de $\{x^k\}_{k \in \mathbb{N}}$, se houver, é um ponto crítico f .

Teorema 4.7. Supondo que a sequência $(v_k)_{k \in \mathbb{N}} \subset \mathbb{R}_{++}$ seja escolhida de acordo com a estratégia **(E3)**. Seja $0 < \xi < 1$ e $k_0 \in \mathbb{N}$ tais que $v_k \leq \xi\rho\|d^k\|^2$, para todos $k \geq k_0$. Então, para cada $N \in \mathbb{N}$ tal que $N > k_0$, tem-se

$$\min\{\|d^k\| : k = 0, 1, \dots, N-1\} \leq \frac{\sqrt{f(x^0) - f^* + \sum_{k=0}^{k_0-1} v_k}}{\sqrt{(1-\xi)\rho}} \frac{1}{\sqrt{N}}.$$

Consequentemente, para um dado $\varepsilon > 0$ e $k_0 \in \mathbb{N}$ tais que $v_k \leq \xi\rho\|d^k\|^2$ para todo $k \geq k_0$, se $N \geq \max\{k_0, (f(x^0) - f^* + \sum_{k=0}^{k_0-1} v_k)/(\rho(1-\xi)\varepsilon^2)\}$, então

$$\min\{\|d^k\| : k = 0, 1, \dots, N-1\} \leq \varepsilon.$$

4.4 Considerações finais

Esse capítulo descreveu os algoritmos implementados. A técnica FISTA é usada para resolver o subproblema das técnicas DC (DCA, BDCA, nmBDCA), e aplicada em conjunto essas técnicas para a reconstrução de imagens. A partir das descrições e definições apresentadas, os resultados explicitados no capítulo a seguir podem ser obtidos.

5 Reconstrução de imagens

Esta seção apresenta as definições sobre os modelos de reconstrução de imagens usados, bem como suas utilizações em conjunto com as técnicas DC, as reconstruções e os resultados dos experimentos realizados para as abordagens apresentadas no Capítulo 4. Todas as implementações usam a mesma máquina, com sistema operacional Windows 10 Home Single Language Versão 21H2, com a seguintes configuração: Intel(R) Core(TM) i7-8550U CPU @ 1.80 GHz, 1.99 GHz; memória RAM de 8 GB e SSD de 256 GB.

Neste trabalho serão realizadas três seções de experimentos numéricos, com dois tipos de imagem: imagens sintéticas (preto e branco) e imagens médicas reais (em escala de cinza). As imagens em preto e branco descritas na Tabela 2 serão aplicadas na Seção 5.3, e foram obtidas em <https://sipi.usc.edu/database/database.php> e Oliveira e Tcheou (2019). A Seção 5.4 usa imagens de tomografia computadorizada (TC) de COVID-19, especificadas na Tabela 3, retiradas de <https://www.eibir.org/covid-19-Imaging-datasets/>, com acesso aberto. A Seção 5.5 usa uma imagem de ressonância magnética (RM) de pacientes com Alzheimer, disponível em <https://www.kaggle.com/datasets/sachinkumar413/alzheimer-mri-dataset/data>). Todos os testes foram implementados no *software* MATLAB[®].

Tabela 2 – Imagens usadas na Seção 5.3.

Imagem	Descrição	Cor	Dimensão
1	Checkerboard	black and white	200 × 200
2	Text	black and white	256 × 256
3	Qr code	black and white	200 × 200

Tabela 3 – Imagens usadas na Seção 5.4 e 5.5.

Imagem	Descrição	Cor	Dimensão
4	TC1	grayscale	200 × 200
5	TC2	grayscale	200 × 200
6	TC3	grayscale	200 × 200
7	RM	grayscale	128 × 128

Apesar das representações de imagem contarem com variadas fontes de ruído, apenas os ruídos Gaussiano e Riciano foram considerados nas análises realizadas, pois são comuns em aplicações com processamento de imagem/sinais. Ainda, apenas a norma isotrópica (TV_{Iso}) foi implementada para as técnicas aplicadas. Cada técnica possui parâmetros específicos, evidenciados em cada seção subsequente. Não foram utilizadas imagens coloridas para as reconstruções.

5.1 Remoção de ruído em problemas não convexos

Como já mencionado, técnicas de reconstrução de imagem são ferramentas importantes em sistemas de visão computacional e muitas outras aplicações que requerem imagens nítidas obtidas de imagens ruidosas. As imagens, então, podem ser representadas e processadas por técnicas computacionais que necessitam de parâmetros e uma forma padrão de representação das imagens a serem reconstruídas.

O FISTA possui um modelo padrão para a remoção de ruídos em imagens, que conta com a variação total convexa $TV(x)$, a imagem com ruído $b \in \mathbb{R}^n$ (vetorização da imagem B em escala de cinza com dimensões $n_1 \times n_2$, onde $n = n_1 \cdot n_2$), a imagem $x \in \mathbb{R}^n$, que é a solução do problema, e o parâmetro de fidelidade/regularização $\mu > 0$, conforme explicitado a seguir.

$$\min_{x \in \mathbb{R}^n} \frac{\mu}{2} \|x - b\|^2 + TV(x). \quad (5.1)$$

A remoção de ruído de imagens pela aplicação do método DCA, representado pelo modelo (4.10), pode ser alcançada pela inserção de uma técnica específica para reconstrução de imagens dentro do algoritmo do DCA. Nesse caso, a técnica a ser inserida é o FISTA, e o ponto de inserção será pela transformação do modelo (4.10) em (5.1).

A utilização de problemas convexos para *Denoising* possibilita a reconstrução de imagens por vários métodos. Entretanto, conforme explicitado por Oliveira e Tcheou (2019), as regularizações não convexas têm vantagens sobre as convexas, como na preservação de arestas e bordas (LANZA; MORIGI; SGALLARI, 2016; NIKOLOVA; NG; TAM, 2010). Isso é realizado pela aplicação de uma função penalizadora não convexa (ϕ), que induz gradientes de imagem esparsos $(\nabla x)_i$. A inserção da penalização no modelo representativo de imagens é definida por:

$$\min_{x \in \mathbb{R}^n} \frac{\mu}{2} \|x - b\|^2 + TV_\phi(x), \text{ com } TV_\phi(x) := \sum_{i=1}^n \phi(\|(\nabla x)_i\|), \quad (5.2)$$

onde $\phi : \mathbb{R}_+ \rightarrow \mathbb{R}$, $\mu > 0$ é um parâmetro de fidelidade e $(\nabla x)_i$ representa aproximações de diferenças finitas de derivadas parciais horizontais e verticais de primeira ordem, conforme explicitado por:

$$(\nabla x)_i := \begin{pmatrix} X_{l+1,j} - X_{l,j} \\ X_{l,j+1} - X_{l,j} \end{pmatrix}, \text{ com a } i - \text{ésima coordenada } x \text{ de armazenamento de } X_{l,j},$$

que define $\|(\nabla x)_i\| = \sqrt{(X_{l+1,j} - X_{l,j})^2 + (X_{l,j+1} - X_{l,j})^2}$, sendo $X \in \mathbb{R}^{n_1 \times n_2}$ e $x \in \mathbb{R}^n$, com $n = n_1 \times n_2$.

O problema (5.2) é não convexo e não diferenciável, tratado por aproximações locais (fortemente) convexas em Lanza, Morigi e Sgallari (2016). Dependendo da

função penalizadora escolhida, o problema (5.2) pode ser um problema de otimização convexa não diferenciável.

O trabalho de [Oliveira e Tcheou \(2019\)](#) mostra que, para uma ampla classe de funções penalizadoras ϕ , o problema (5.2) é de fato um problema de programação com decomposições DC. Para isso, assume que $\phi : \mathbb{R}_+ \rightarrow \mathbb{R}$ é côncava e não decrescente, e prova que, sob essas suposições, a função composta $TV_\phi(x) = \sum_{i=1}^n \phi(\|(\nabla x)_i\|)$ pode ser escrita como uma diferença de duas funções convexas, como $\tau TV(x) - TV_\phi(x)$, para todo $\tau \geq \phi'_+(0) \geq 0$.

A prova dessa afirmação leva em consideração o fato de que para uma função convexa $c : \mathbb{R}^n \rightarrow \mathbb{R}_+$, se $\phi : \mathbb{R}_+ \rightarrow \mathbb{R}$ é uma função côncava e não decrescente tal que $\phi'_+(0) < \infty$, então $\tau c(x) - \phi(c(x))$ é convexa para todo $\tau \geq \phi'_+(0)$; veja [Oliveira e Tcheou \(2019, Lema 3\)](#).

5.1.1 Denoising com modelo TV_ϕ

A inserção de uma função penalizadora $\phi : \mathbb{R}_+ \rightarrow \mathbb{R}$ em conjunto com o DCA torna o modelo de imagem uma representação dada por:

$$\min_{x \in \mathbb{R}^n} \frac{\mu}{2} \|x - b\|^2 + TV_\phi(x) = g(x) - h(x), \quad (5.3)$$

onde μ é o parâmetro de regularização do modelo de imagem, b é a imagem com ruído e g e h são funções convexas, isto é, (5.3) é um problema DC, como mostra a Proposição 5.1 a seguir; veja [Oliveira e Tcheou \(2019, Proposição 2\)](#).

Proposição 5.1. *Seja $\mu > 0$ e $b \in \mathbb{R}^n$ fixos, e $\rho > 0$ um parâmetro de fortemente convexa. Assuma que $\phi : \mathbb{R}_+ \rightarrow \mathbb{R}$ é uma função côncava e não decrescente. Então, $f(x) = \frac{\mu}{2} \|x - b\|^2 + TV_\phi(x)$ é uma função DC com componentes DC fortemente convexas*

$$g(x) = \frac{\mu + \rho}{2} \|x - b\|^2 + \tau TV(x) \quad \text{e} \quad h(x) = \frac{\rho}{2} \|x - b\|^2 + \tau TV(x) - TV_\phi(x), \quad (5.4)$$

para todo $\tau \geq \phi'_+(0)$.

Observação 5.1. *Seja f uma função DC com componentes g e h , isto é, $f(x) = g(x) - h(x)$ com g e h funções convexas. Sem perda de generalidade, pode-se assumir que as componentes DC são funções fortemente convexas, caso contrário, para determinadas funções fortemente convexas $\alpha(x)$, pode-se obter uma decomposição fortemente DC da seguinte forma: $f(x) = \tilde{g}(x) - \tilde{h}(x)$, onde $\tilde{g}(x) = g(x) + \alpha(x)$ e $\tilde{h}(x) = h(x) + \alpha(x)$ funções fortemente convexas.*

A utilização do FISTA em conjunto com o DCA requer a resolução de problemas como:

$$\min_{x \in \mathbb{R}^n} \|x - s^k\|^2 + 2\lambda TV(x), \quad (5.5)$$

com parâmetro λ e imagem corrompida s^k . O resultado seguinte garante que é possível aplicar FISTA resolvendo os subproblemas do nmBDCA.

Proposição 5.2. *Sejam g e h componentes DC (5.4) do modelo não convexo (5.3). Então, o subproblema (4.12) pode ser escrito como (5.5).*

Observação 5.2. *Os últimos resultados mostram que, na formulação (5.5), o FISTA recebe a imagem s^k , corrompida por $\frac{w^k}{\mu+\rho}$, e o parâmetro λ para resolver o problema de remoção de ruído em uma imagem. Em outras palavras, mostra que o FISTA pode resolver o subproblema (4.12).*

5.1.2 Algoritmos DCA-FISTA e nmBDCA-FISTA com TV_ϕ

Utilizando-se as Equações (5.4) e (5.5), pode-se implementar os algoritmos do DCA-FISTA e nmBDCA-FISTA com penalização (TV_ϕ), que deve ser inserida na função de variação total, compondo a função $h(x)$. Alguns exemplos de funções de penalização côncavas, diferenciáveis e não decrescentes são apresentados na Tabela 4 e obedecem os critérios da Seção 5.1.1 para $\tau \geq 1$. As descrições da Seção 5.1.1 e a Tabela 4 permitem definir os Algoritmos 10 e 11, que correspondem ao DCA e nmBDCA com variação total penalizada (TV_ϕ).

Tabela 4 – Funções de penalização $\phi_a : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ côncavas, diferenciáveis e não decrescente, parametrizadas por $a > 0$.

	ϕ_{log}	ϕ_{atan}	ϕ_{exp}
$\phi_a(r)$	$\frac{\log(1+ar)}{a}$	$\frac{\arctan\left(\frac{1+ar}{\sqrt{3}}\right) - \frac{\pi}{6}}{a\sqrt{3}/2}$	$\frac{1-e^{-ar}}{a}$
$\phi'_a(r)$	$\frac{1}{1+ar}$	$\frac{1}{1+ar+a^2r^2}$	$\frac{1}{e^{ar}}$

Algorithm 10 DCA-FISTA com TV_ϕ

```

1: Carregar imagem  $X \in \mathbb{R}^{n_1 \times n_2}$ ;
2: Obter imagem  $x \in \mathbb{R}^n$ , onde  $n = n_1 \cdot n_2$ ;
3: Definir os parâmetros  $\mu, \rho, \max\_iter, k = 0$ ;
4: Definir ruído  $\epsilon$ ;
5: Criar imagem  $b = x + \epsilon$ ;
6: Definir a função de penalização  $\phi$ ;
7: Calcular  $\phi'_+$  (usar a Tabela 4);
8:  $\lambda = \frac{\phi'_+(0)}{\mu + \rho}$ ;
9:  $x_k = b$ ;
10: while  $k < \max\_iter$  do
11:    $s^k = b + \frac{w^k}{\mu+\rho}$ ,  $w^k \in \partial h(x_k)$ ;
12:    $x_{k+1} = \text{FISTA-FGP}(s^k, \lambda)$ ;
13:    $\text{stop\_rule} = \frac{\|x_{k+1} - x_k\|}{\|x_{k+1}\|}$ ;
14:    $x_k = x_{k+1}$ ;
15:   if  $\text{stop\_rule} < \text{tol}$  then
16:     break;
17:   end if
18:    $k = k + 1$ ;
19: end while

```

Algorithm 11 nmBDCA com TV_ϕ

```

1: Carregar imagem  $X \in \mathbb{R}^{n_1 \times n_2}$ ;
2: Calcular  $x \in \mathbb{R}^n$ , com  $n = n_1 \cdot n_2$ ;
3: Definir de parâmetros  $\tau, t_0, \nu, \zeta, \mu, \rho, \max\_iter, k = 0$ ;
4: Inserir ruído  $\epsilon$ ;
5: Criar imagem  $b = x + \epsilon$ ;
6: Definir a função de penalização  $\phi$ ;
7: Calcular  $\phi'_+$ ;
8:  $\lambda = \frac{\phi'_+(0)}{\mu + \rho}$ ;
9:  $x_k = b$ ;
10: while  $k < \max\_iter$  do
11:    $s^k = b + \frac{w^k}{\mu + \rho}, w^k \in \partial h(x_k)$ ;
12:    $y_k = \text{FISTA-FGP}(s^k, \lambda)$ ;
13:    $d_k = y_k - x_k$ ;
14:   while  $f(y_k + t_k d_k) - f(y_k) + \tau(t_k \|d_k\|)^2 - \frac{\nu \|d_k\|^2}{k} > 0$  do
15:      $t_k = \zeta t_k$ ;
16:   end while
17:    $x_{k+1} = y_k + t_k d_k$ ;
18:    $\text{stop\_rule} = \frac{\|x_{k+1} - x_k\|}{\|x_{k+1}\|}$ ;
19:    $x_k = x_{k+1}$ ;
20:   if  $\text{stop\_rule} < \text{tol}$  then
21:     break;
22:   end if
23:    $k = k + 1$ ;
24: end while

```

O Algoritmo 11, que consiste no nmBDCA para remoção de ruído, obtém e vetoriza a imagem a ser analisada. Em seguida, os parâmetros de configuração, que podem variar de acordo com o problema, são definidos. A função desses parâmetros é definir: o parâmetro base (ν) para gerar uma sequência do nmBDCA, o parâmetro de fidelidade (μ) do modelo de imagem, o parâmetro de função fortemente convexa (ρ), as iterações máximas da reconstrução (\max_iter) e a variância do ruído gaussiano (ϵ) inserida na imagem (b).

Assim como no Algoritmo 10, o nmBDCA realiza o cálculo do parâmetro λ , cálculo das funções g e h , da imagem s^k , de uma solução do subproblema com o FISTA. Entretanto, após a obtenção de y_k (linha 11 do Algoritmo 11), uma busca do tipo Armijo é realizada, e pode ser descrita, de acordo com Fukushima e Mine (1981), como: definir a direção de busca em um ponto atual resolvendo o subproblema; determinar o tamanho do passo t_k pela regra do tipo Armijo (ARMIGO, 1966; POLAK, 1971); atualizar a solução (x_{k+1}). A regra Armijo contém uma sequência dada por $\frac{\nu \|d_k\|^2}{k}$, cujas estratégias de escolha são apresentadas na Seção 3.2 de Ferreira, Santos e Souza (2024).

O ajuste dos parâmetros específicos do nmBDCA (ζ, t_0, τ e ν), utilizados na busca Armijo, influencia diretamente o tempo de CPU do nmBDCA, pois afeta a busca realizada. A escolha de t_k (t_0 para a primeira iteração) determina a regularização no ponto inicial e, assim, pode influenciar o tempo para a execução da busca. O parâmetro τ pode ser entendido como uma compensação de t_k e reajusta sua influência na busca.

A sequência gerada pelo nmBDCA é baseada em ν e permite controlar, quando ocorre, o crescimento da função objetivo do problema. Ela pode ser ajustada para considerar os parâmetros τ e t_k com maior ênfase ou não.

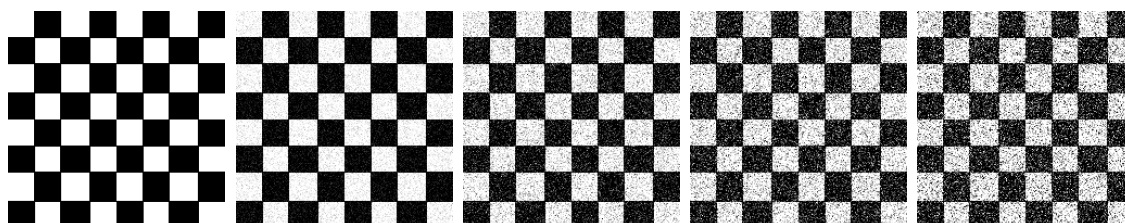
O nmBDCA também possui o parâmetro ζ , que pode afetar consideravelmente o desempenho do método, pois determina em cada iteração o passo a ser aplicado na busca do ponto de maior diminuição do função objetivo. Assim, valores inadequados de ζ causam perda de informação na busca. A particularidade dos parâmetros nmBDCA pode ser considerada um desafio para aplicação em outros problemas de imagem, pois requerem configuração em cada um deles. Por outro lado, reforça a necessidade de pesquisas que busquem automatizar ou viabilizar essa configuração.

5.2 Inserção de ruído

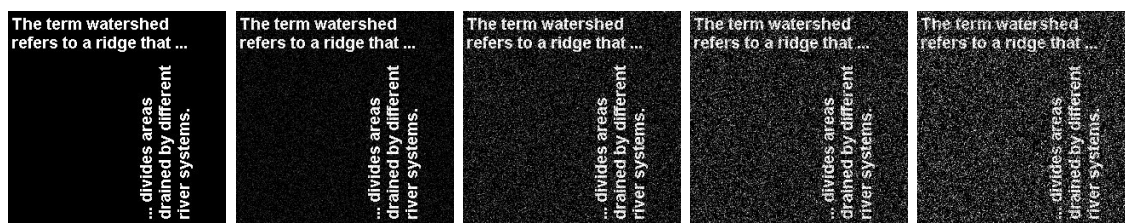
A inserção de ruído em imagens busca representar as distorções que podem ocorrer na aquisição de uma imagem. Neste caso, a adição de ruído evidencia o objetivo deste trabalho, que é a obtenção da imagem original através de técnicas computacionais, quando a imagem é degradada por ruído. A influência do ruído Gaussiano é ilustrada nas Figuras 5 e 6, com diferentes variâncias (para imagens preto e branco - 0,01, 0,05, 0,10, e 0,15; para médicas - 0,005, 0,0075 e 0,01).

Figura 5 – Imagens preto e branco com ruído.

(a) Checkerboard. (b) Variância 0,01. (c) Variância 0,05. (d) Variância 0,10. (e) Variância 0,15.



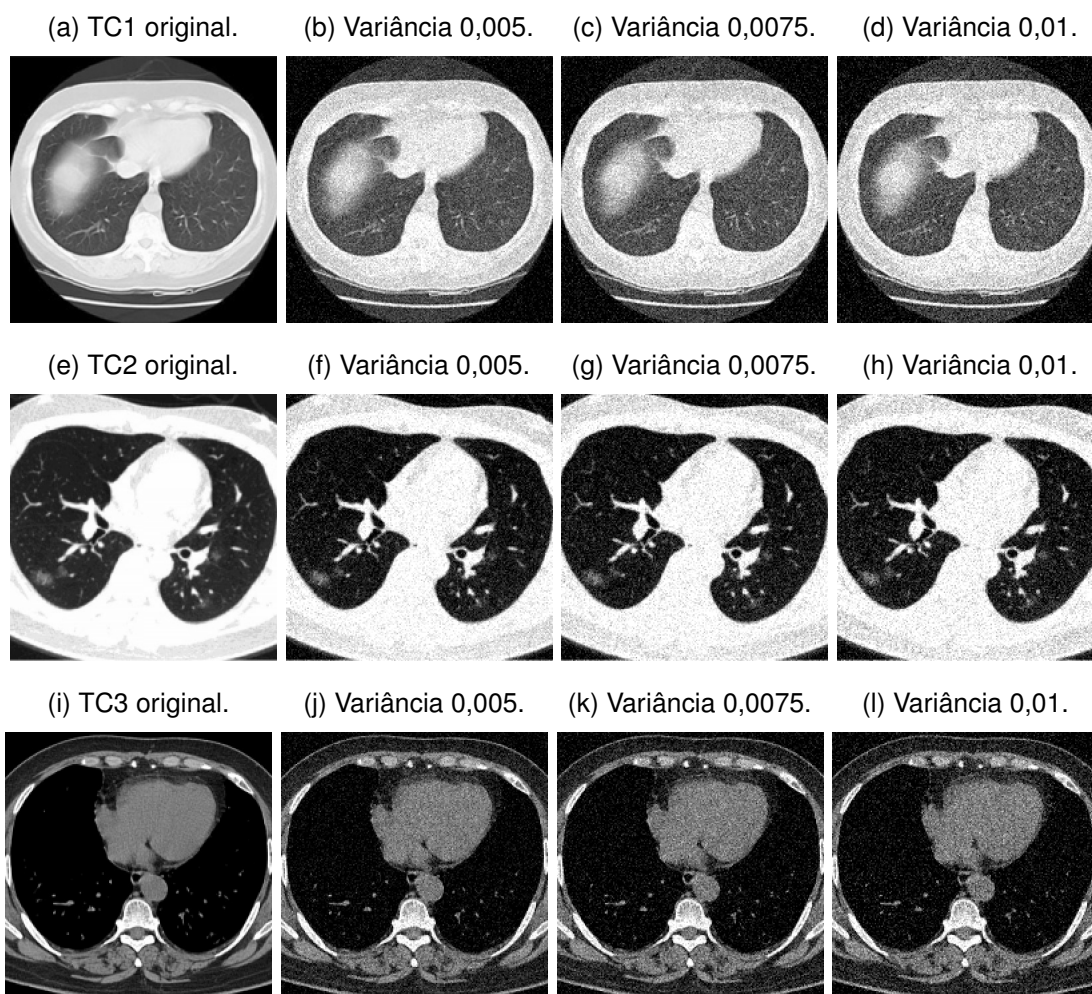
(f) Text. (g) Variância 0,01. (h) Variância 0,05. (i) Variância 0,10. (j) Variância 0,15.



(k) Qr code. (l) Variância 0,01. (m) Variância 0,05. (n) Variância 0,10. (o) Variância 0,15.



Figura 6 – Imagens médicas com ruído.



Além da análise visual, a qualidade das reconstruções pode ser avaliada numericamente por meio do cálculo de algumas métricas, como: *Peak Signal-to-Noise Ratio* (PSNR) e a *Structural Similarity Index Metric* (SSIM); a fim de verificar a qualidade de imagem em termos de contraste, resolução, ruído, iluminação, etc. Essas métricas são amplamente utilizadas em processamentos de imagens digitais, como compressão de imagem (MIN et al., 2017), restauração de imagem (AKBAR et al., 2020), redução de ruído de imagem (SHUKLA; PANDEY; REDDY, 2020) e vários outros (HORÉ; ZIOU, 2010).

O PSNR é uma métrica que define a relação entre a máxima energia de um sinal e o ruído que afeta sua representação fidedigna, resultante do cálculo do logaritmo do *Mean Squared Error* (MSE) da imagem, que tradicionalmente usa o método de soma como seu principal componente (SETIADI, 2021), resultando em uma grandeza em decibel (dB). A formulação do PSNR usa a soma do quadrado do valor do pixel da imagem original menos o pixel de imagem a se avaliar. Assim, de maneira geral,

tem-se:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\max^2}{\text{MSE}} \right),$$

onde \max é o nível máximo de uma imagem em escala de cinza de 8 bits ($\max = 255$).

O SSIM é outra ferramenta quantitativa usada para medir a qualidade de imagens, definido pela primeira vez em Wang et al. (2004), construído com base em três fatores principais: luminância, contraste e estrutura. Esses fatores substituem o método de somatório usado como base para o cálculo do PSNR (HORÉ; ZIOU, 2013). Matematicamente, o SSIM entre as imagens original (im) e reconstruída (im') é:

$$\text{SSIM}(im, im') = l(im, im') \cdot c(im, im') \cdot s(im, im'),$$

onde $l(im, im')$, $c(im, im')$ e $s(im, im')$, e suas componentes, são:

$$l(im, im') = \frac{2\mu_i\mu_{i'} + C_1}{\mu_i^2 + \mu_{i'}^2 + C_1}; \quad (5.6)$$

$$c(im, im') = \frac{2\sigma_i\sigma_{i'} + C_2}{\sigma_i^2 + \sigma_{i'}^2 + C_2}; \quad (5.7)$$

$$s(im, im') = \frac{\sigma_{ii'} + C_3}{\sigma_i\sigma_{i'} + C_3}; \quad (5.8)$$

$$\mu_i = \frac{\sum_{x=1}^M \sum_{y=1}^N \sum_{z=1}^O im_{xyz}}{MNO};$$

$$\sigma_i^2 = \frac{\sum_{x=1}^M \sum_{y=1}^N \sum_{z=1}^O (im_{xyz} - \mu_i)^2}{MNO};$$

$$\sigma_{ii'} = \frac{\sum_{x=1}^M \sum_{y=1}^N \sum_{z=1}^O (im_{xyz} - \mu_i)(im'_{xyz} - \mu_{i'})}{MNO};$$

$$\mu_{i'} = \frac{\sum_{x=1}^M \sum_{y=1}^N \sum_{z=1}^O im'_{xyz}}{MNO};$$

$$\sigma_{i'}^2 = \frac{\sum_{x=1}^M \sum_{y=1}^N \sum_{z=1}^O (im'_{xyz} - \mu_{i'})^2}{MNO};$$

$$\sigma_{i'i} = \frac{\sum_{x=1}^M \sum_{y=1}^N \sum_{z=1}^O (im'_{xyz} - \mu_{i'})(im_{xyz} - \mu_i)}{MNO},$$

onde M e N são as dimensões da imagem, e O se refere às camadas (cores) da imagem.

A Equação (5.6) compara a luminância (μ) entre as imagens e possui valor máximo de 1, obtido quando as imagens são iguais ($\mu_i = \mu_{i'}$). A Equação (5.7), por sua vez, compara o contraste (σ) das imagens, também possuindo valor máximo de 1, quando os contrastes, calculados com base no desvio padrão, são iguais. A função que compara as estruturas das imagens, com base no coeficiente de correlação, é (5.8), e possui valor máximo de 1 quando $\sigma_{ii'} = \sigma_i\sigma_{i'}$.

O SSIM máximo ($SSIM = 1$) acontecerá quando as três métricas forem iguais a 1. As constantes C_1, C_2, C_3 são usadas para evitar os denominadores zero (WANG et al., 2004). Alguns valores para essas constantes são: $C_1 = (0,01 \times 255)^2$, $C_2 = (0,03 \times 255)^2$ e $C_3 = \frac{C_2}{2}$. A Tabela 5 apresenta as descrições das métricas de avaliação selecionadas.

Tabela 5 – Métricas selecionadas para análise das imagens.

Métrica	Descrição
PSNR	Relação sinal-ruído de pico. Quanto maior, melhor a qualidade da imagem.
SSIM	Índice de Similaridade Estrutural. Quanto mais próximo de 1, melhor a qualidade da imagem.
Iteradas	Valor da iteração em que o método parou (critério de parada satisfeito).
$\frac{\ x_{k+1} - x_k\ }{\ x_{k+1}\ }$	Critério de parada definido para cada técnica computacional.
Tempo (s)	Tempo, em segundos, que a técnica leva para reconstruir a imagem.

5.3 Reconstrução de imagens preto e branco

Nesta seção o FISTA, com modelo convexo, DCA e nmBDCA, com modelo não convexo, são usados na reconstrução das imagens da Tabela 2. É realizada a análise do parâmetro do modelo de regularização μ . Ainda, são analisadas as qualidades de cada reconstrução e o tempo de CPU para cada técnica.

5.3.1 Reconstrução com FISTA

Esta seção apresenta os resultados obtidos pela aplicação do FISTA, que utiliza o modelo TV convexo para a reconstrução de imagens. A técnica busca, mediante aplicação de diferentes níveis de ruído, obter imagens próximas da imagem original. O intuito dessa análise é comparar o resultados das reconstruções visualmente e numericamente.

O FISTA necessita de algumas inicializações, apresentadas na Tabela 6. O parâmetro μ influencia significativamente o desempenho (métricas de avaliação) das técnicas aplicadas para reconstruir as imagens. Então, antes da obtenção dos resultados, foi realizada a varredura do μ , para cada nível de ruído, em cada imagem. Isso possibilita estimar um μ que permita melhores resultados de reconstrução (foi utilizado o μ cujo SSIM é maior, buscando destacar mais os detalhes da imagem). A varredura foi feita em uma faixa de valores (0,01 a 0,99; 1 a 30) suficiente para evidenciar os valores máximos de PSNR e SSIM, onde para cada μ foi gerado uma nova imagem com ruído (semente aleatória de ruído).

Tabela 6 – Parâmetros de inicialização do FISTA.

Parâmetro	Descrição
ϵ	Ruído inserido na Imagem original.
μ	Parâmetro de regularização do modelo de imagem.
max_iter	Número máximo de iterações por reconstrução.

As varreduras para o FISTA podem ser observadas nas Figuras 7 a 10. É evidente que as métricas PSNR e SSIM, à medida que a variância do ruído inserido aumenta, alcançam menores valores, corroborando a importância da avaliação do nível de ruído nas reconstruções. Nota-se, que a faixa de valores para os melhores PSNR e SSIM é pequena, o que significa que o μ possui muita influência sobre o desempenho das reconstruções. Quanto menor o valor de μ , mais lenta fica a técnica e mais ruído é removido da imagem. No entanto, para um μ muito pequeno, pode haver distorção dos *pixels* da imagem.

Os valores de μ que alcançam os maiores SSIM, em cada variância aplicada, e os resultados de cada reconstrução são mostrados na Tabela 7, que permite verificar o desempenho de reconstrução (métricas PSNR e SSIM), e o tempo para a obtenção

da imagem (tempo de CPU em segundos (s)). Nota-se que as reconstruções foram satisfatórias, com SSIM acima de 70% na maioria dos testes. Os resultados são obtidos pela média de 100 execuções da mesma técnica, para cada cenário de variância e μ configurado. As análises devem ser feitas entre mesmas imagens com variâncias diferentes, já que cada imagem tem suas características.

Nota-se que, à medida que as variâncias aumentam, as métricas PSNR e SSIM diminuem e o tempo de reconstrução se torna mais distinto entre as imagens, o que ressalta a influência dos níveis de ruído inserido. Todas as reconstruções com FISTA alcançaram as iterações máximas (30000 iterações), devido ao critério de parada de 5×10^{-10} definido pelos autores, visando garantir que o FISTA retorne a melhor imagem possível para o teste realizado. Então, os resultados apresentados são os melhores possíveis com o FISTA em cada teste.

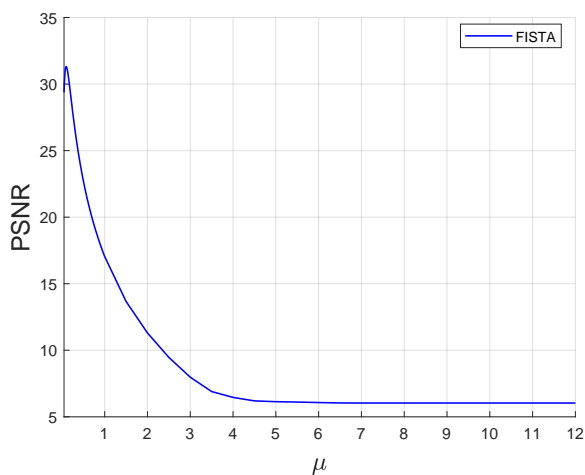
Tabela 7 – Resultados de reconstrução com FISTA.

<i>Variância 0,01</i>						
Imagem	Técnica	μ	PSNR	SSIM	Iter.	Tempo (s)
Checkerboard	FISTA	0,09	31,5074	0,9024	30000	64,9197
Text	FISTA	0,12	27,3387	0,9242	30000	150,5507
Qr code	FISTA	0,09	28,6996	0,9400	30000	62,7289
<i>Variância 0,05</i>						
Imagem	Técnica	μ	PSNR	SSIM	Iter.	Tempo (s)
Checkerboard	FISTA	0,2	24,9515	0,7951	30000	75,7891
Text	FISTA	0,27	20,9555	0,8226	30000	113,9681
Qr code	FISTA	0,17	22,1838	0,8599	30000	63,8970
<i>Variância 0,10</i>						
Imagem	Técnica	μ	PSNR	SSIM	Iter.	Tempo (s)
Checkerboard	FISTA	0,31	22,3435	0,7338	30000	90,8217
Text	FISTA	0,37	18,6978	0,7381	30000	125,5682
Qr code	FISTA	0,28	19,4782	0,8097	30000	64,5098
<i>Variância 0,15</i>						
Imagem	Técnica	μ	PSNR	SSIM	Iter.	Tempo (s)
Checkerboard	FISTA	0,37	20,8012	0,6979	30000	91,0962
Text	FISTA	0,52	16,7676	0,6794	30000	108,8225
Qr code	FISTA	0,34	17,9732	0,7700	30000	76,8756

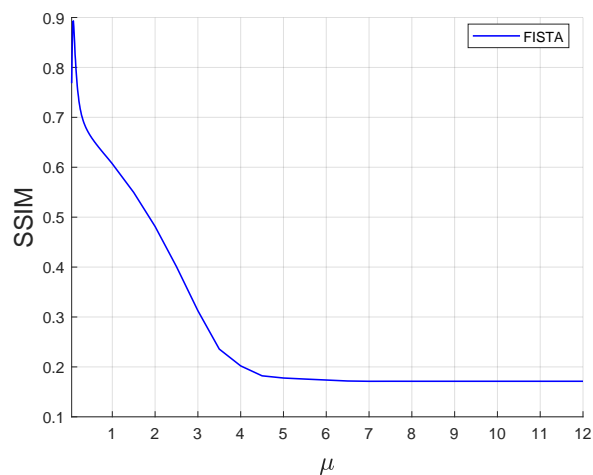
A análise visual das reconstruções pode ser realizada na Figura 11, onde a imagem original (à esquerda) e as imagens reconstruídas (à direita) são comparadas. É possível observar que o modelo TV convexo remove o ruído das imagens, preservando detalhes importantes em cada teste. Entretanto, há imagens que apresentam maior quantidade de *pixels* em cinza, à medida que a variância aumenta, significando que o modelo não preserva bem algumas de suas regiões. Assim, o modelo carece de melhorias para preservar mais os detalhes em cada imagem.

Figura 7 – Varredura do FISTA com variância 0,01.

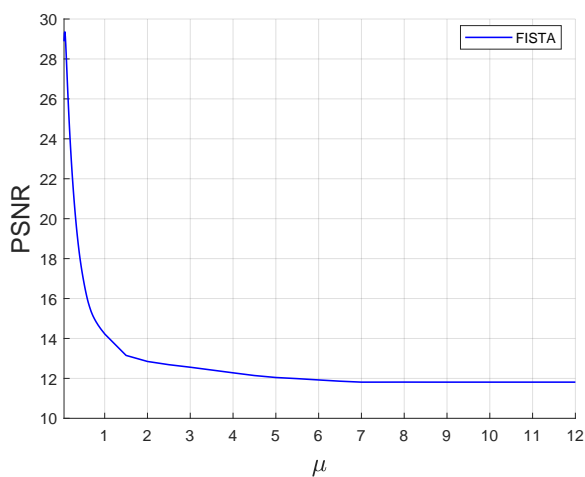
(a) Checkerboard PSNR.



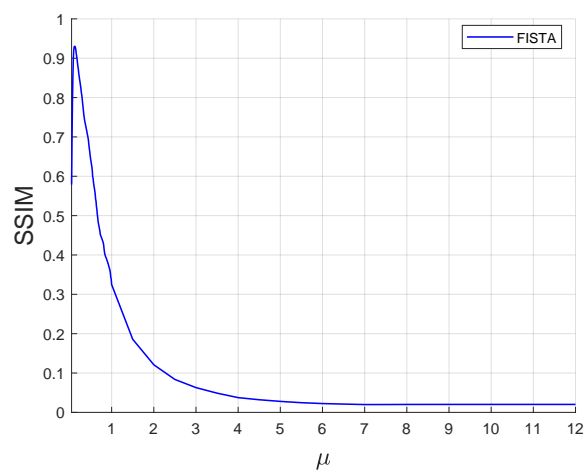
(b) Checkerboard SSIM.



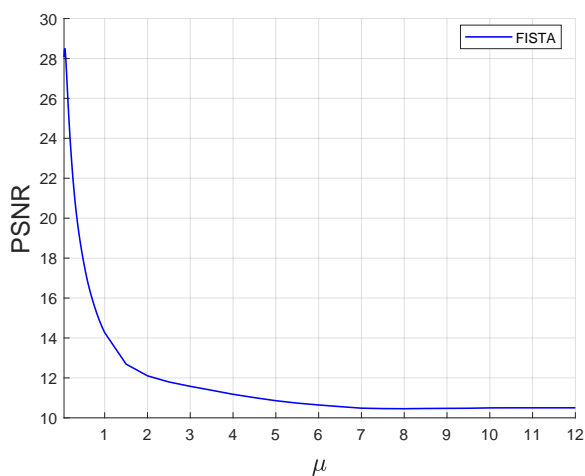
(c) Text PSNR.



(d) Text SSIM.



(e) Qr code PSNR.



(f) Qr code SSIM.

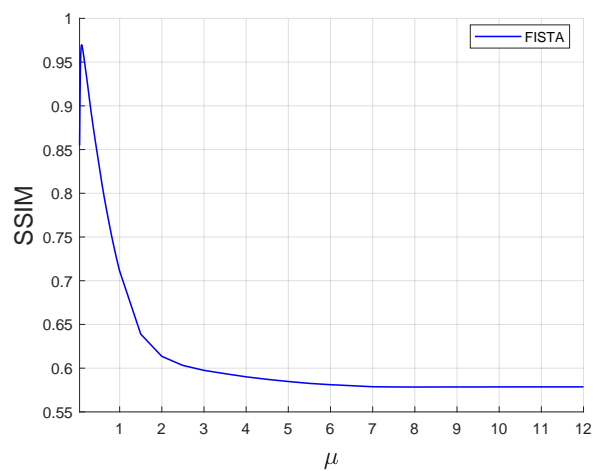
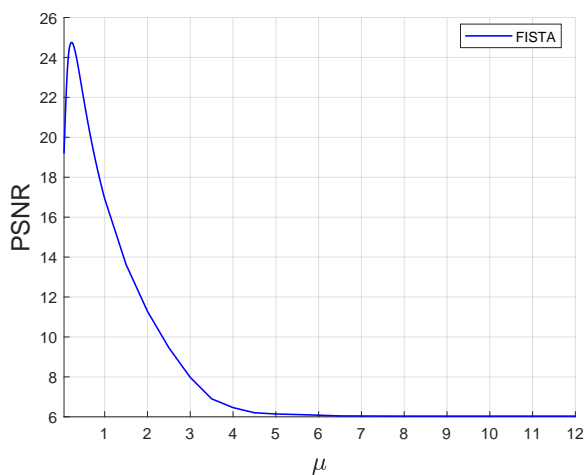
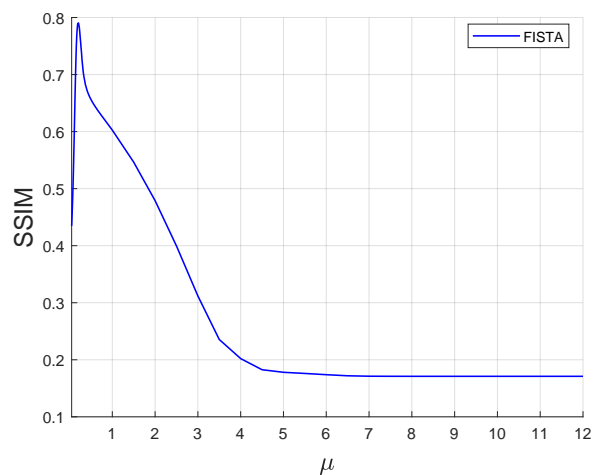


Figura 8 – Varredura do FISTA com variância 0,05.

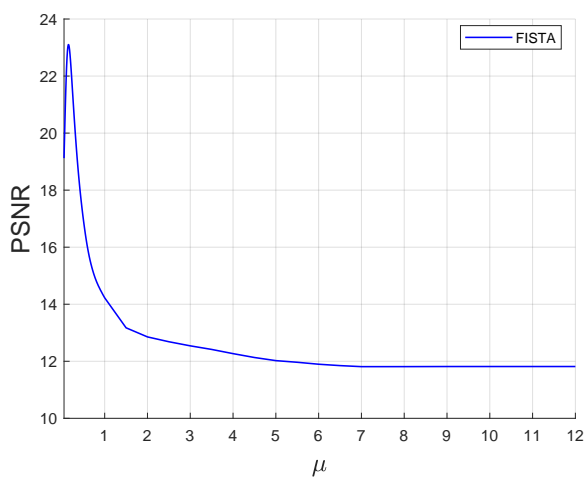
(a) Checkerboard PSNR.



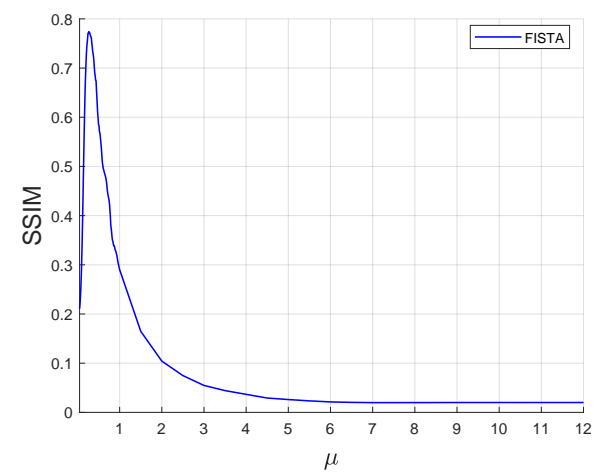
(b) Checkerboard SSIM.



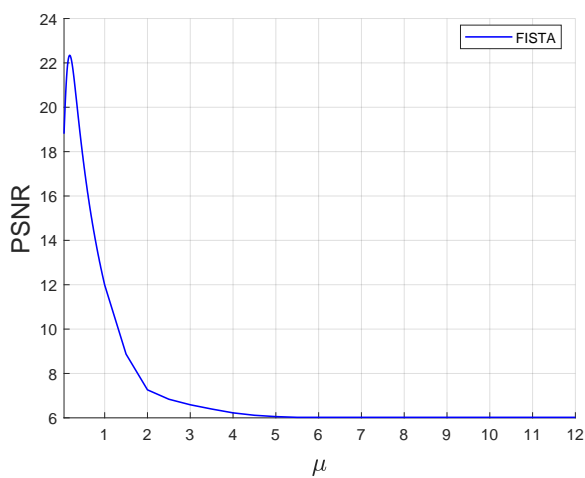
(c) Text PSNR.



(d) Text SSIM.



(e) Qr code PSNR.



(f) Qr code SSIM.

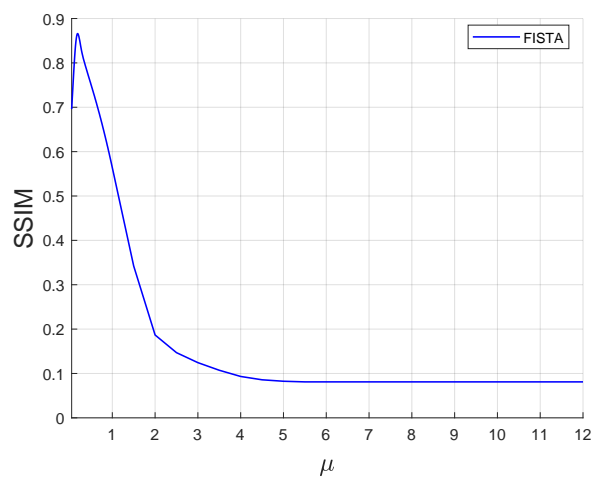
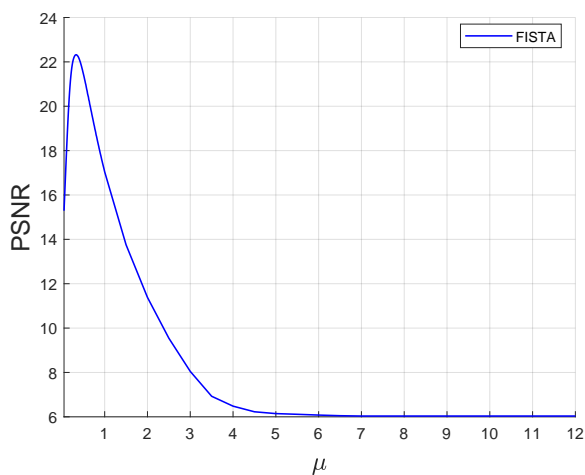
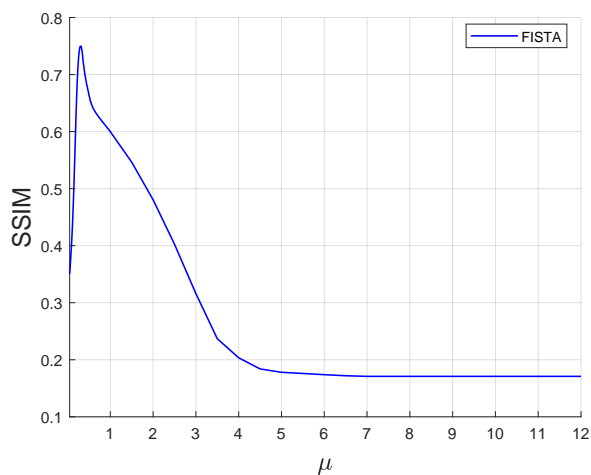


Figura 9 – Varredura do FISTA com variância 0,10.

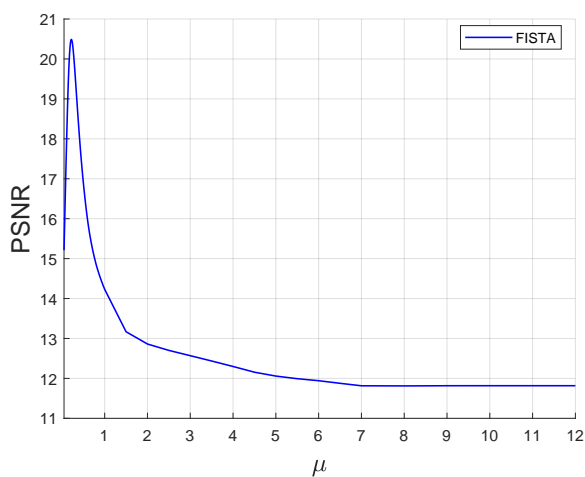
(a) Checkerboard PSNR.



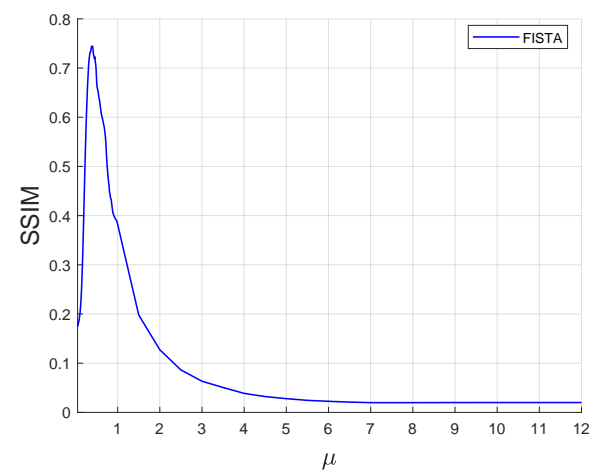
(b) Checkerboard SSIM.



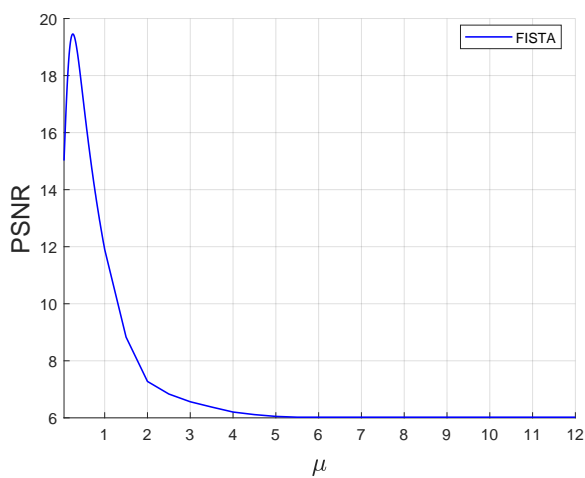
(c) Text PSNR.



(d) Text SSIM.



(e) Qr code PSNR.



(f) Qr code SSIM.

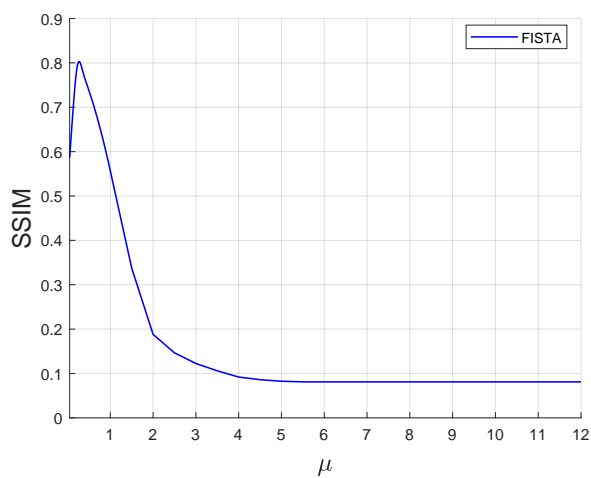
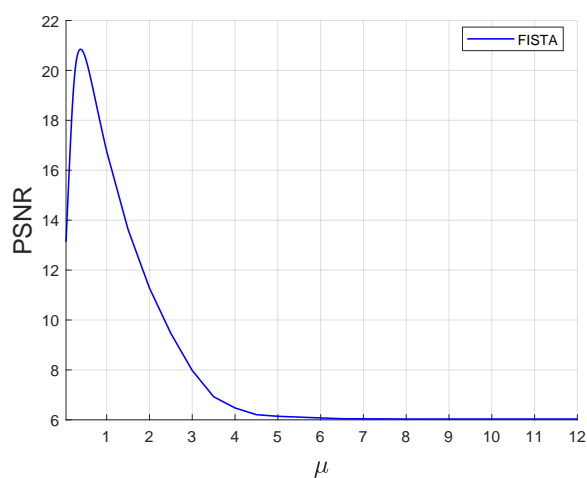
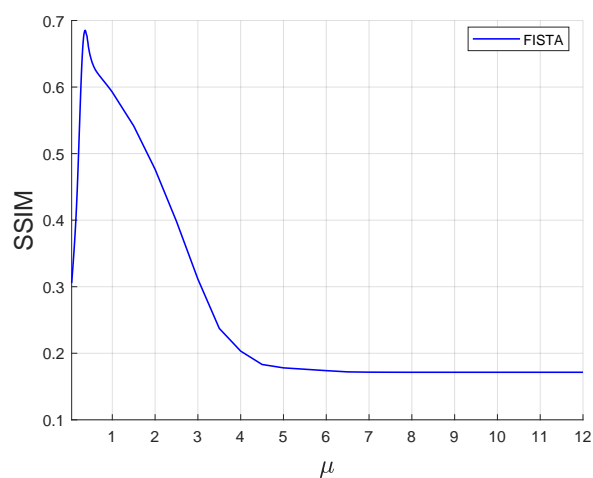


Figura 10 – Varredura do FISTA com variância 0,15.

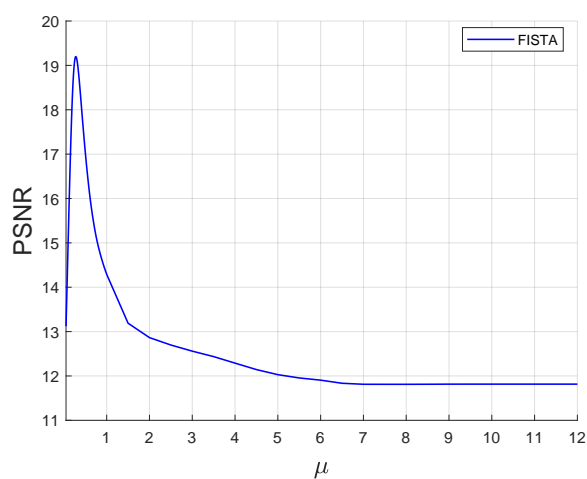
(a) Checkerboard PSNR.



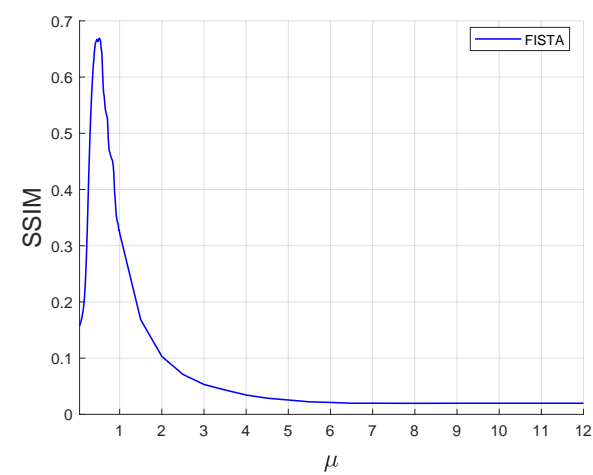
(b) Checkerboard SSIM.



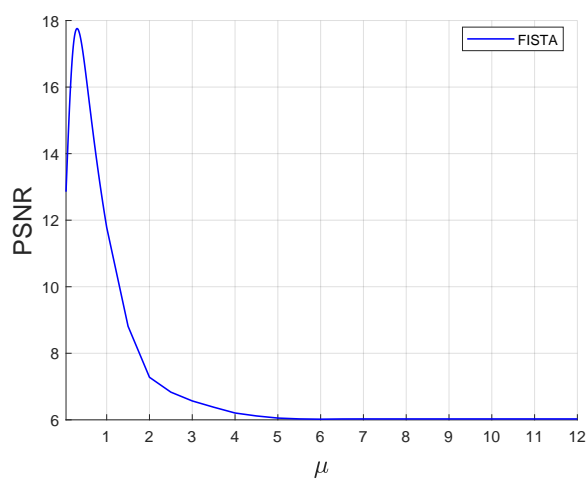
(c) Text PSNR.



(d) Text SSIM.



(e) Qr code PSNR.



(f) Qr code SSIM.

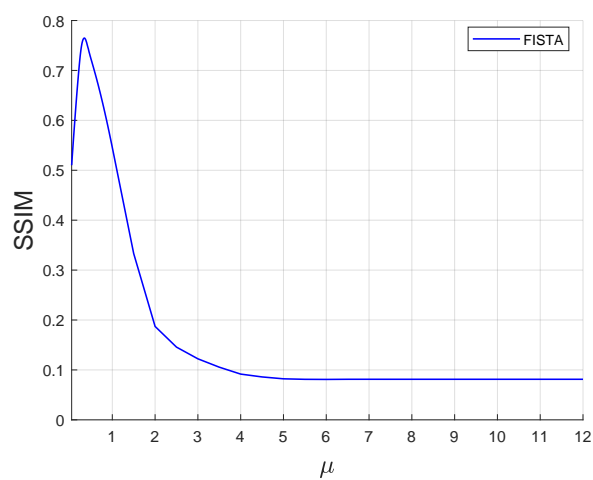
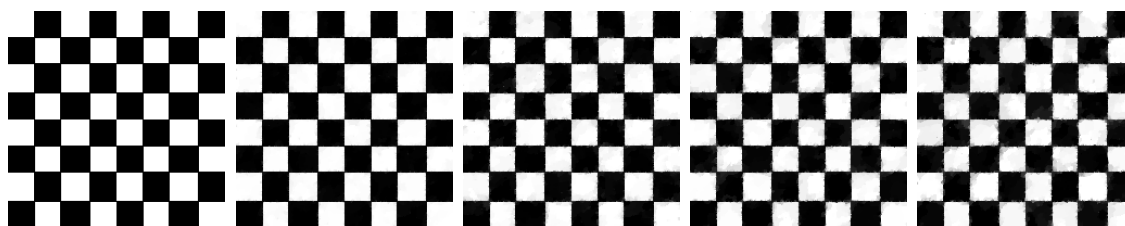
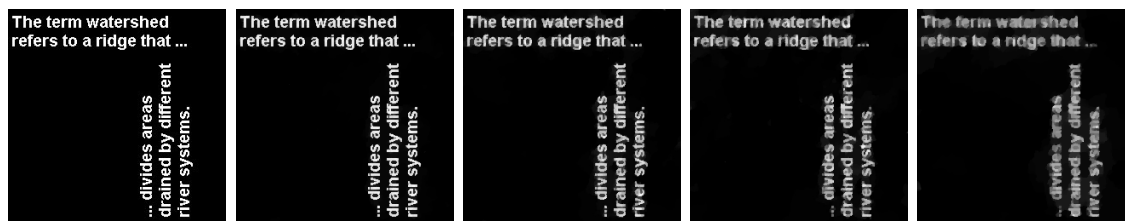


Figura 11 – Reconstrução com FISTA.

(a) Checkerboard. (b) Variância 0,01. (c) Variância 0,05. (d) Variância 0,10. (e) Variância 0,15.



(f) Text. (g) Variância 0,01. (h) Variância 0,05. (i) Variância 0,10. (j) Variância 0,15.



(k) Qr code. (l) Variância 0,01. (m) Variância 0,05. (n) Variância 0,10. (o) Variância 0,15.



5.3.2 Reconstrução com DCA e nmBDCA

Como explicado na Seção 5.1, a maior preservação de detalhes em imagens pode ser alcançada pelo uso de regularizações não convexas, a partir da alteração do modelo de imagem. Essa alteração requer que o modelo usado neste trabalho seja resolvido pelos métodos DCA e nmBDCA, em conjunto com o FISTA (o FISTA resolve o subproblema criado pelas técnicas DC). Assim, esta seção apresenta essas reconstruções a partir da interação entre as técnicas. Semelhante ao FISTA, as técnicas DC precisam de parâmetros de inicialização, evidenciados na Tabela 8.

Nota-se que as técnicas DC, principalmente o nmBDCA, têm mais parâmetros para configurar. Isso ocorre porque o modelo TV não convexo tem a função de penalização, e o nmBDCA possui os parâmetros da busca Armijo. Há parâmetros variáveis que devem ser configurados, levando em consideração as características do problema.

Tabela 8 – Parâmetros de inicialização para DCA e nmBDCA.

Parâmetro	DCA	nmBDCA	Valor	Descrição
ϵ	Sim	Sim	variável	Ruído inserido na imagem original.
μ	Sim	Sim	variável	Parâmetro de regularização do modelo de imagem.
ρ	Sim	Sim	1	Parâmetro de fortemente convexa.
ν	Não	Sim	variável	Parâmetro que gera uma sequência na regra Armijo.
ζ	Não	Sim	variável	Passo de decrescimento de t_0 .
t_0	Não	Sim	variável	Passo inicial da busca Armijo.
τ	Não	Sim	variável	Peso aplicado à regra Armijo.
max_iter	Sim	Sim	variável	Número de iterações por reconstrução.
ϕ	Sim	Sim	arco tangente	Função de penalização.

Assim como na seção anterior, o parâmetro μ das técnicas DC foram analisados antes da obtenção dos resultados de reconstrução, como mostram as Figuras 12 a 15. Diferente das varreduras com o FISTA, o μ das técnicas DC possui faixas de valores maiores (Figuras 12 e 13, por exemplo). Isso significa que a variação desse parâmetro influencia menos as reconstruções e, ainda, que existem resultados próximos em métricas, com μ diferentes, cujos tempos para reconstruir sejam distintos, afinal, quanto menor o μ , maior a chance de se gastar mais tempo com as reconstruções.

De maneira geral, em termos do PSNR e do SSIM, os desempenhos do DCA e nmBDCA são parecidos. Porém, como o nmBDCA tem uma busca, o custo computacional é diferente. Isso pode ser verificado na Tabela 9, obtida pela média das 100 execuções de cada técnica, que traz as configurações do DCA e nmBDCA e os resultados das reconstruções. O DCA e nmBDCA foram configurados com os seguintes critérios de parada: 200 iterações máximas e tolerância de parada de 5×10^{-4} , com função de penalização $\phi = \phi_{Atan}$ e parâmetro de fortemente convexa $\rho = 1$. Esses critérios foram definidos para permitir a funcionalidade das técnicas aplicadas, mas devem ser cuidadosamente estudados para outros problemas.

Observa-se que o nmBDCA gasta menos tempo que o DCA na maioria das reconstruções, além de possuir PSNR e SSIM similares, corroborando as varreduras realizadas. Quando o DCA ganha em tempo, possui desempenho inferior ao nmBDCA. Nota-se, também, que a mudança da variância em cada imagem pode ocasionar a alteração do parâmetro ν , tornando-o importante para alcançar bons resultados.

Figura 12 – Varredura do DCA e do nmBDCA com variância 0,01.

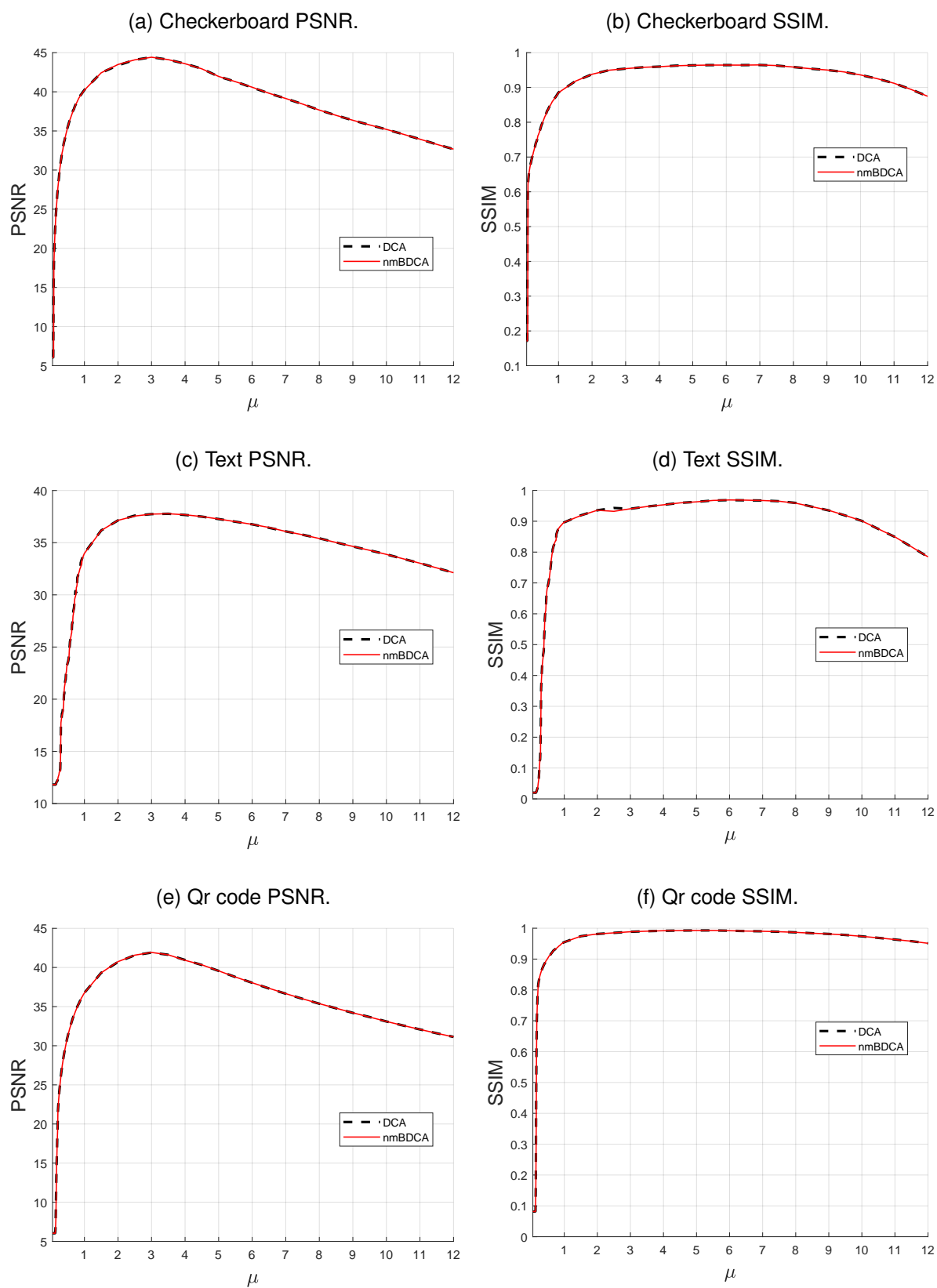
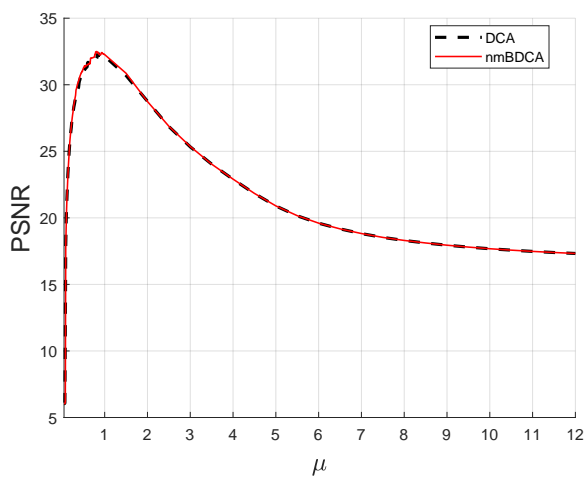
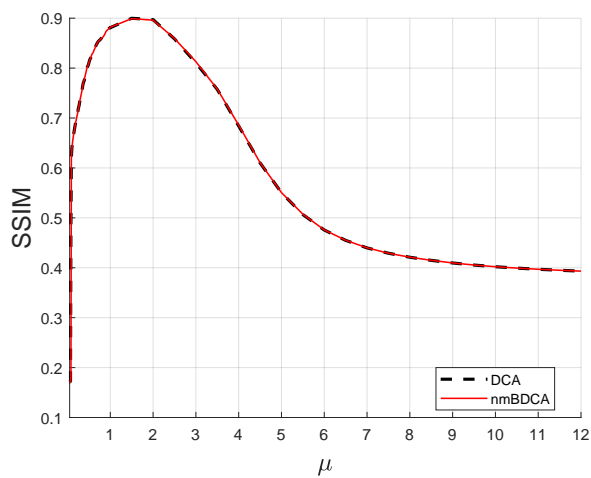


Figura 13 – Varredura do DCA e do nmBDCA com variância 0,05.

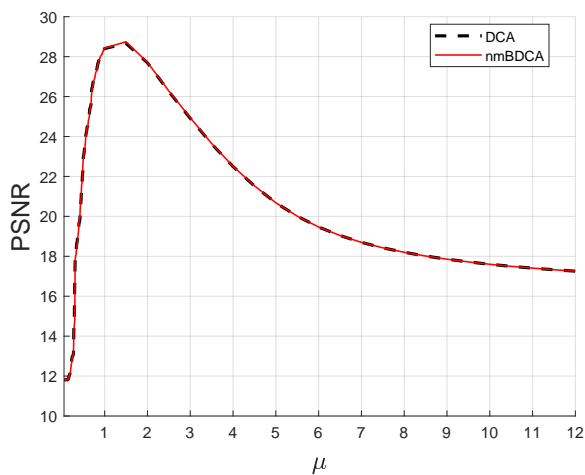
(a) Checkerboard PSNR.



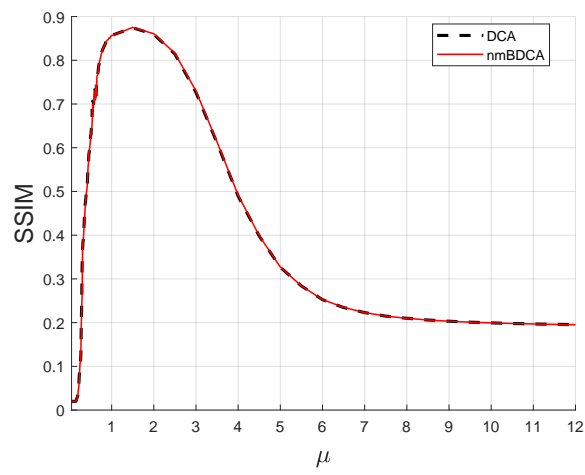
(b) Checkerboard SSIM.



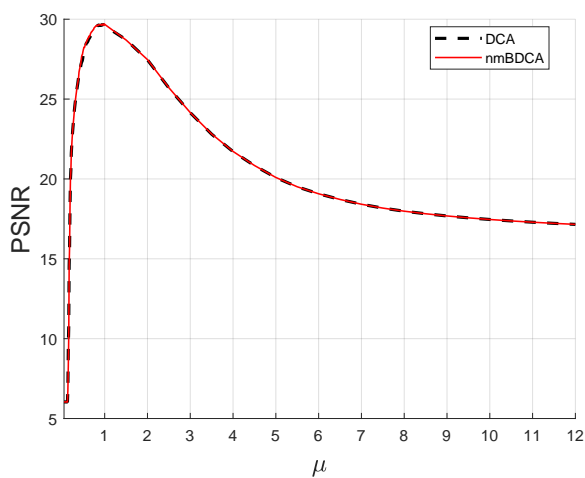
(c) Text PSNR.



(d) Text SSIM.



(e) Qr code PSNR.



(f) Qr code SSIM.

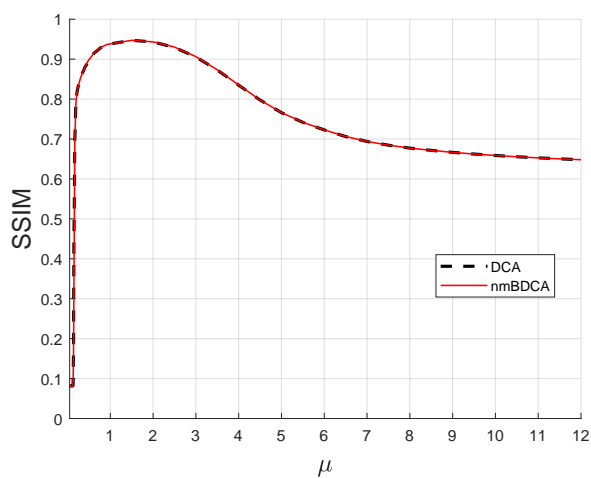
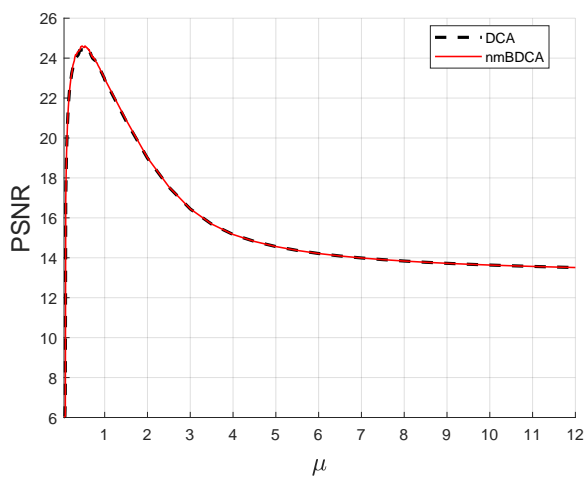
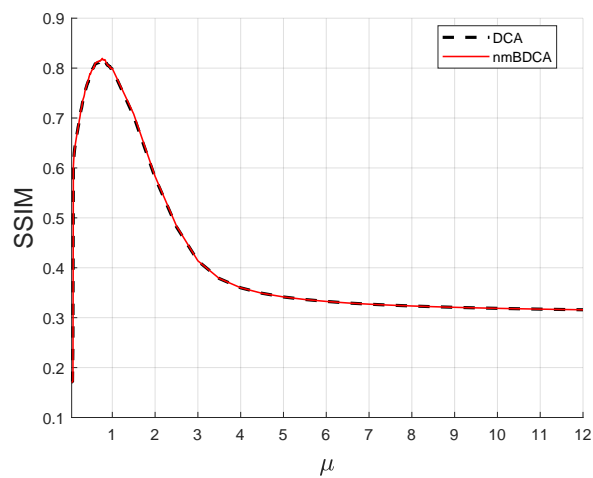


Figura 14 – Varredura do DCA e do nmBDCA com variância 0,10.

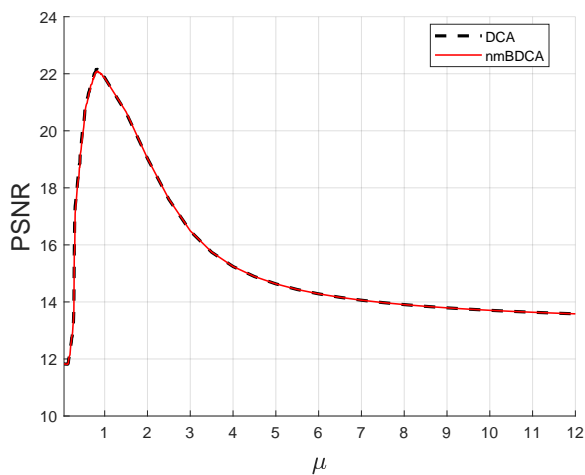
(a) Checkerboard PSNR.



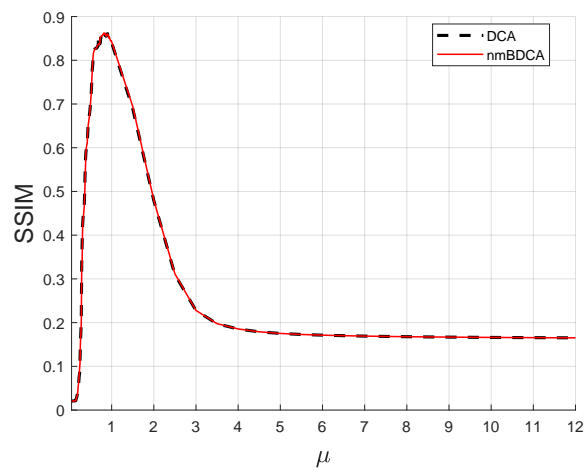
(b) Checkerboard SSIM.



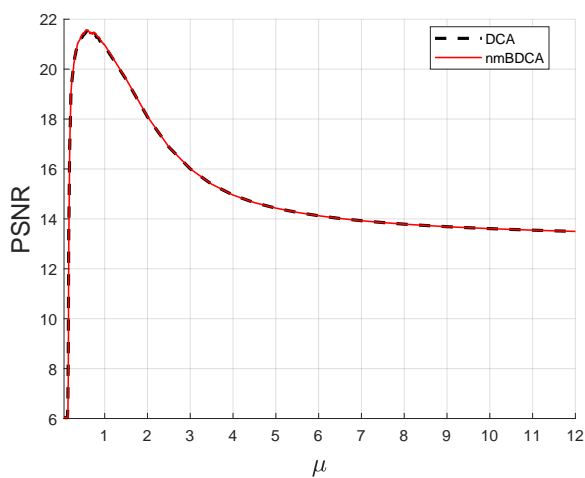
(c) Text PSNR.



(d) Text SSIM.



(e) Qr code PSNR.



(f) Qr code SSIM.

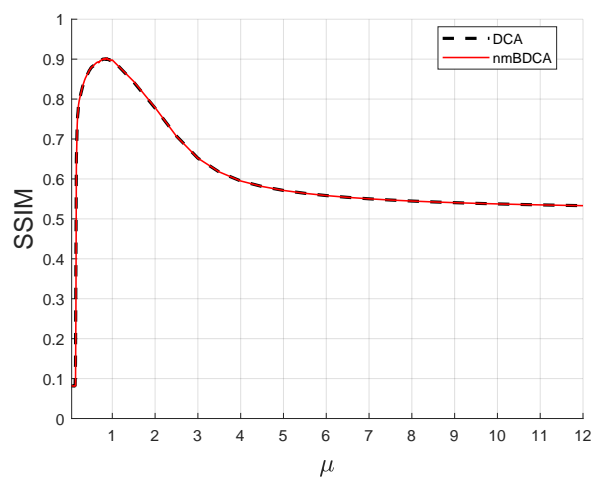
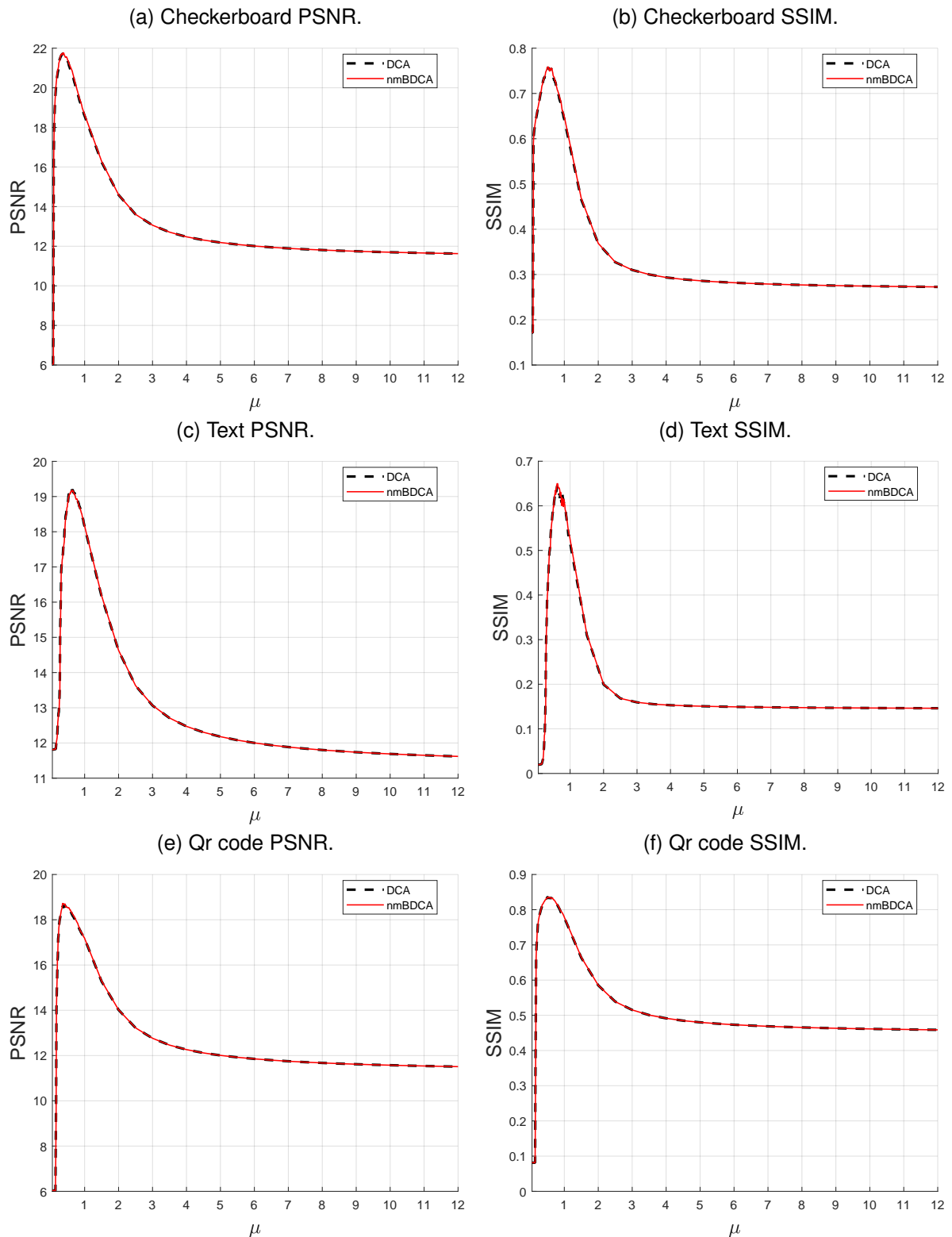


Figura 15 – Varredura do DCA e do nmBDCA com variância 0,15.



É perceptível, ainda, que à medida que as variâncias do ruído inserido aumentam, o μ do maior SSIM diminui e reforça a causa de aumento do tempo nas reconstruções. Os resultados são considerados satisfatórios, com SSIM acima de 76% em todos os testes realizados. É importante ressaltar que os parâmetros configurados podem alterar para outras imagens, sendo necessário analisar cada problema para obtenção de resultados aceitáveis.

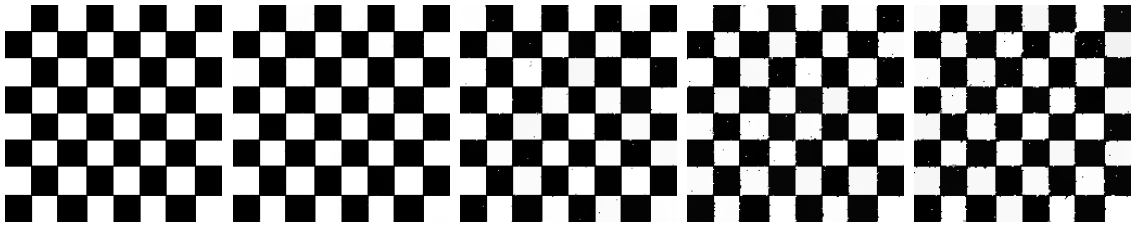
Tabela 9 – Resultados de reconstrução com DCA e nmBDCA.

Variância 0,01							
Imagem	Técnica	μ	ν	PSNR	SSIM	Iter.	Tempo (s)
Checkerboard	DCA	5	-	42,4985	0,9740	10,27	3,0467
	nmBDCA	5	50	42,5624	0,9740	7,13	2,6121
Text	DCA	6	-	36,8888	0,9647	17,95	21,9967
	nmBDCA	6	50	36,8933	0,9647	13,80	19,6283
Qr code	DCA	4	-	41,3309	0,9886	11,42	4,0008
	nmBDCA	4	50	41,3721	0,9887	8,31	3,4766
Variância 0,05							
Imagem	Técnica	μ	ν	PSNR	SSIM	Iter.	Tempo (s)
Checkerboard	DCA	1,5	-	31,0011	0,9099	36,04	10,5928
	nmBDCA	1,5	100	31,0869	0,9108	26,60	9,5465
Text	DCA	1,5	-	28,4618	0,9114	65,67	97,1271
	nmBDCA	1,5	60	28,4640	0,9116	56,09	96,3565
Qr code	DCA	1,5	-	28,6500	0,9578	43,99	28,9894
	nmBDCA	1,5	500	28,6972	0,9581	29,10	22,9369
Variância 0,10							
Imagem	Técnica	μ	ν	PSNR	SSIM	Iter.	Tempo (s)
Checkerboard	DCA	0,76	-	24,2283	0,8340	55,34	17,2762
	nmBDCA	0,76	100	24,3256	0,8355	41,22	15,3733
Text	DCA	1	-	22,0319	0,7995	107,92	161,9973
	nmBDCA	0,82	45	22,1315	0,8121	100,67	178,3707
Qr code	DCA	1	-	21,3256	0,8940	70,28	39,0311
	nmBDCA	1	500	21,3839	0,8959	45,79	29,1205
Variância 0,15							
Imagem	Técnica	μ	ν	PSNR	SSIM	Iter.	Tempo (s)
Checkerboard	DCA	0,46	-	21,0772	0,7674	67,69	21,8354
	nmBDCA	0,45	500	21,1852	0,7687	43,88	16,9160
Text	DCA	0,62	-	18,9701	0,6932	136,51	186,7794
	nmBDCA	0,64	45	18,9767	0,6926	125,27	184,9044
Qr code	DCA	0,52	-	18,6159	0,8333	106,49	52,4293
	nmBDCA	0,52	300	18,6609	0,8345	81,62	48,2520

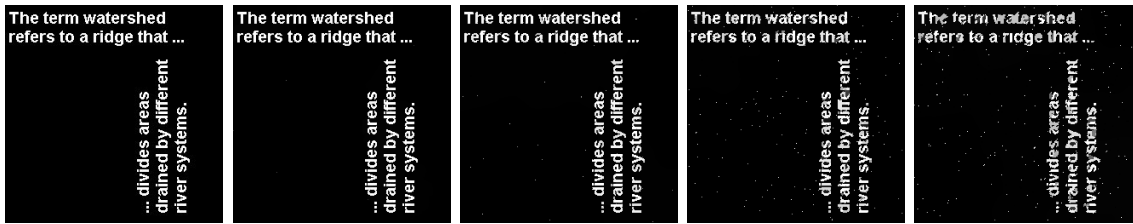
A implementação do DCA e do nmBDCA para a reconstrução das imagens evidencia o potencial que as técnicas tem para este fim, já que foram propostas para funções. Assim, as reconstruções obtidas com o DCA e nmBDCA podem ser avaliadas visualmente nas Figuras 16 e 17, onde se verifica imagens obtidas com qualidade, com menos *pixels* em cinza em relação às imagens reconstruídas com o FISTA, significando melhor preservação de bordas entre as superfícies da imagem. Também é nítida a influência do aumento da variância, pois as imagens reconstruídas apresentam pontos degradados pelo ruído, mas em geral, há grande semelhança entre as imagens reconstruídas e originais. Ratificando o exposto pela Tabela 9, as reconstruções com DCA e nmBDCA são similares, e o nmBDCA tem menor tempo de CPU que o DCA na maioria dos testes realizados.

Figura 16 – Reconstrução com DCA.

(a) Checkerboard. (b) Variância 0,01. (c) Variância 0,05. (d) Variância 0,10. (e) Variância 0,15.



(f) Text. (g) Variância 0,01. (h) Variância 0,05. (i) Variância 0,10. (j) Variância 0,15.

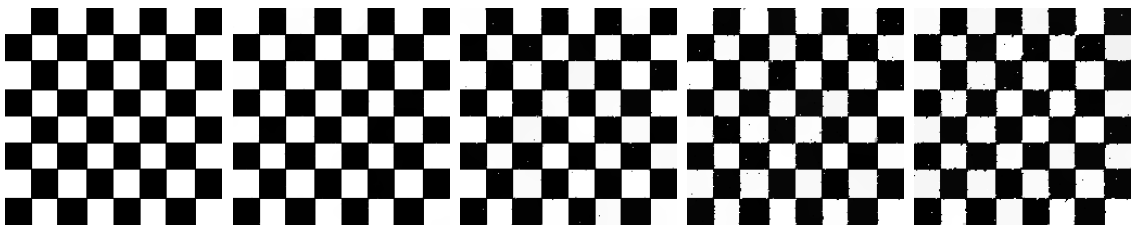


(k) Qr code. (l) Variância 0,01. (m) Variância 0,05. (n) Variância 0,10. (o) Variância 0,15.

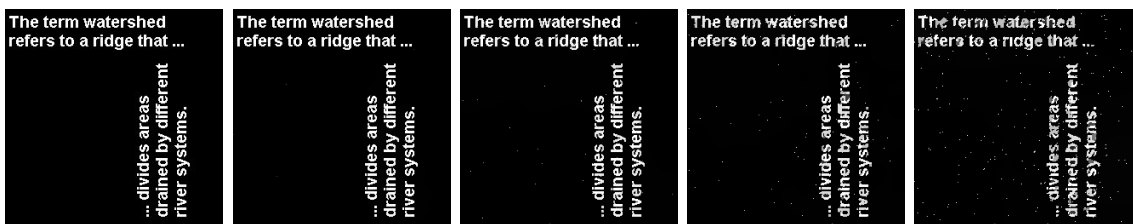


Figura 17 – Reconstrução com nmBDCA.

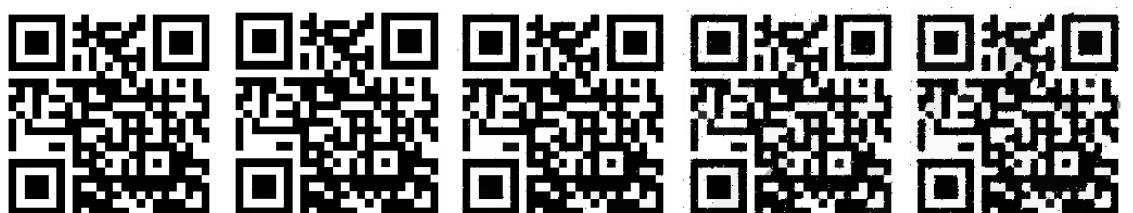
(a) Checkerboard. (b) Variância 0,01. (c) Variância 0,05. (d) Variância 0,10. (e) Variância 0,15.



(f) Text. (g) Variância 0,01. (h) Variância 0,05. (i) Variância 0,10. (j) Variância 0,15.



(k) Qr code. (l) Variância 0,01. (m) Variância 0,05. (n) Variância 0,10. (o) Variância 0,15.



5.3.3 Comparação entre os modelos convexo e não convexo

Esta seção busca ressaltar o desempenho dos testes realizados, comparando os resultados visuais e métricos obtidos com as Figuras 18 a 21 e na Tabela 10. Com variâncias 0,01 e 0,05, percebe-se que o nmBDCA tem desempenho similar ao DCA, mas superior ao FISTA, cujas reconstruções contém *pixels* em cinza principalmente nas regiões de separação entre as superfícies preto e branco. Porém, em geral, as reconstruções são satisfatórias.

As Figuras 20 e 21 tem as maiores variâncias do ruído inserido e ilustram a degradação das imagens e o quanto as reconstruções, principalmente com as técnicas DC, alcançam bons resultados, recuperando a maior parte dos detalhes nas imagens. Nessas figuras, o desempenho qualitativo das técnicas DC é superior ao FISTA, pois as imagens reconstruídas possuem menos *pixels* em cinza e maior nitidez.

A Tabela 10 compara as métricas calculadas, e evidencia que o nmBDCA gasta menos tempo na maioria das reconstruções. Ela ressalta que o SSIM e PSNR do nmBDCA são superiores aos do FISTA em todos os testes, o que reforça a robustez do modelo não convexo ao ruído, em relação ao modelo convexo, especialmente em problemas de reconstrução de imagens onde a qualidade da imagem estudada é de grande importância, e indica que o modelo não convexo é mais adequado que convexo. Há casos (16,67% dos testes) em que o FISTA ganha em tempo (testes com variância 0,10 e 0,15 para a imagem "Text"), mas apresenta desempenho qualitativo inferior ao das técnicas DC.

Nos testes realizados, o nmBDCA supera o DCA em PSNR e iterações em todos os testes, e em SSIM e tempo de CPU em 91,67% dos casos, corroborando a proposta do método nmBDCA de acelerar a convergência mantendo a qualidade da imagem obtida. Em experimentos onde o DCA possui SSIM superior ao nmBDCA (teste com variância de 0,15 para a imagem "Text"), o desempenho do nmBDCA ainda é considerado satisfatório por apresentar PSNR superior e menor tempo de CPU, indicando melhor remoção de ruído, com menos CPU tempo, mas com maior degradação dos *pixels* da imagem original.

O aumento da variância, na maioria dos testes, provoca diminuição do PSNR e SSIM e aumento do tempo de reconstrução. Isso é esperado, pois significa que mais ruído é inserido nas imagens, tornando o problema mais difícil de ser resolvido. Em experimentos onde o tempo do nmBDCA é maior que o do DCA (8,33% dos testes), o SSIM e PSNR do nmBDCA são superiores e, assim, o nmBDCA alcança resultados satisfatórios em termos de qualidade da imagem obtido. Fica claro, então, que apesar dos desempenhos de qualidade e tempo semelhantes, o desempenho do nmBDCA é superior ao DCA.

Figura 18 – Comparação de resultados para variância 0,01.

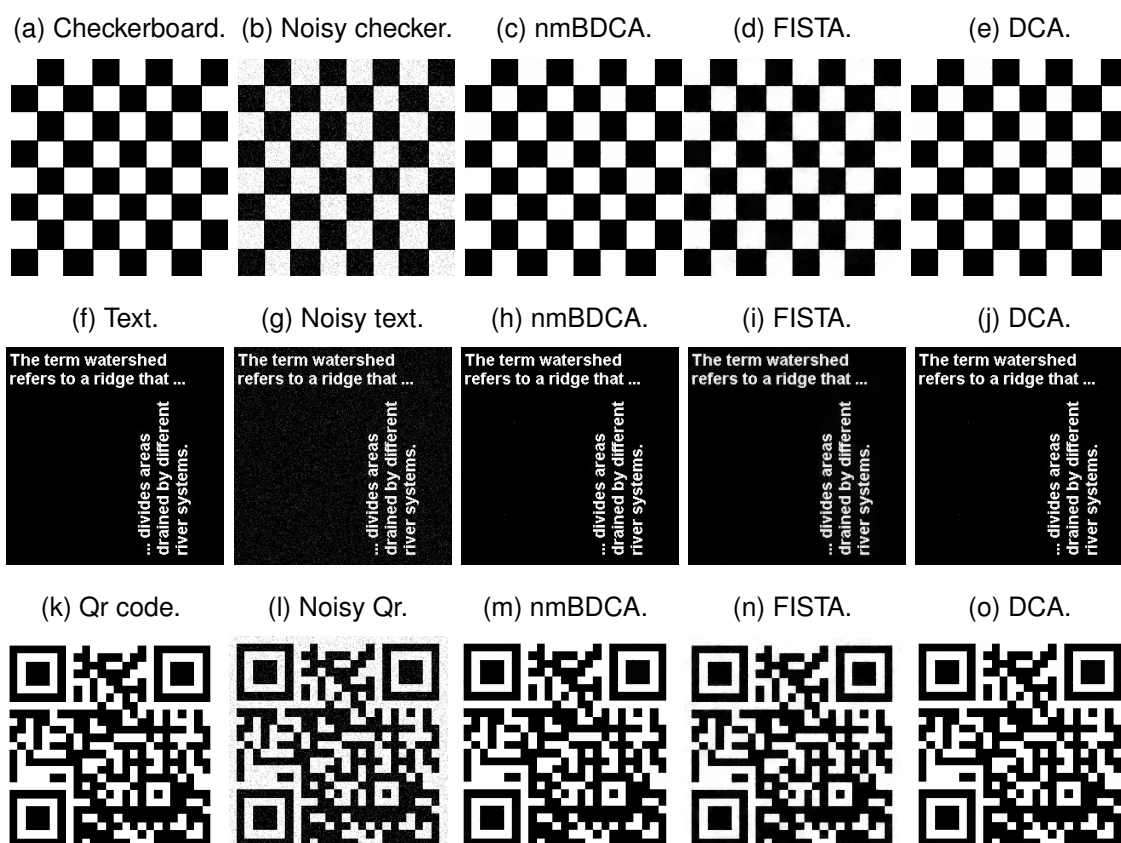


Figura 19 – Comparação de resultados para variância 0,05.

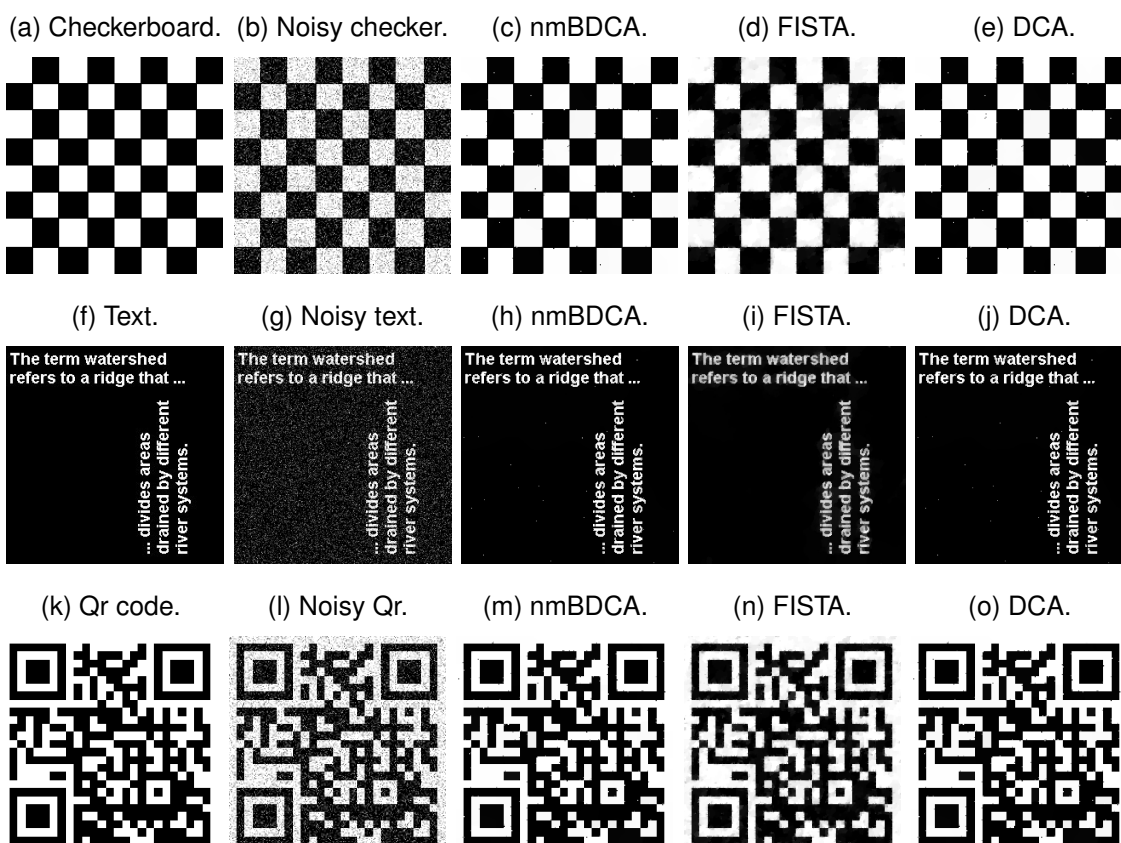


Figura 20 – Comparação de resultados para variância 0,10.

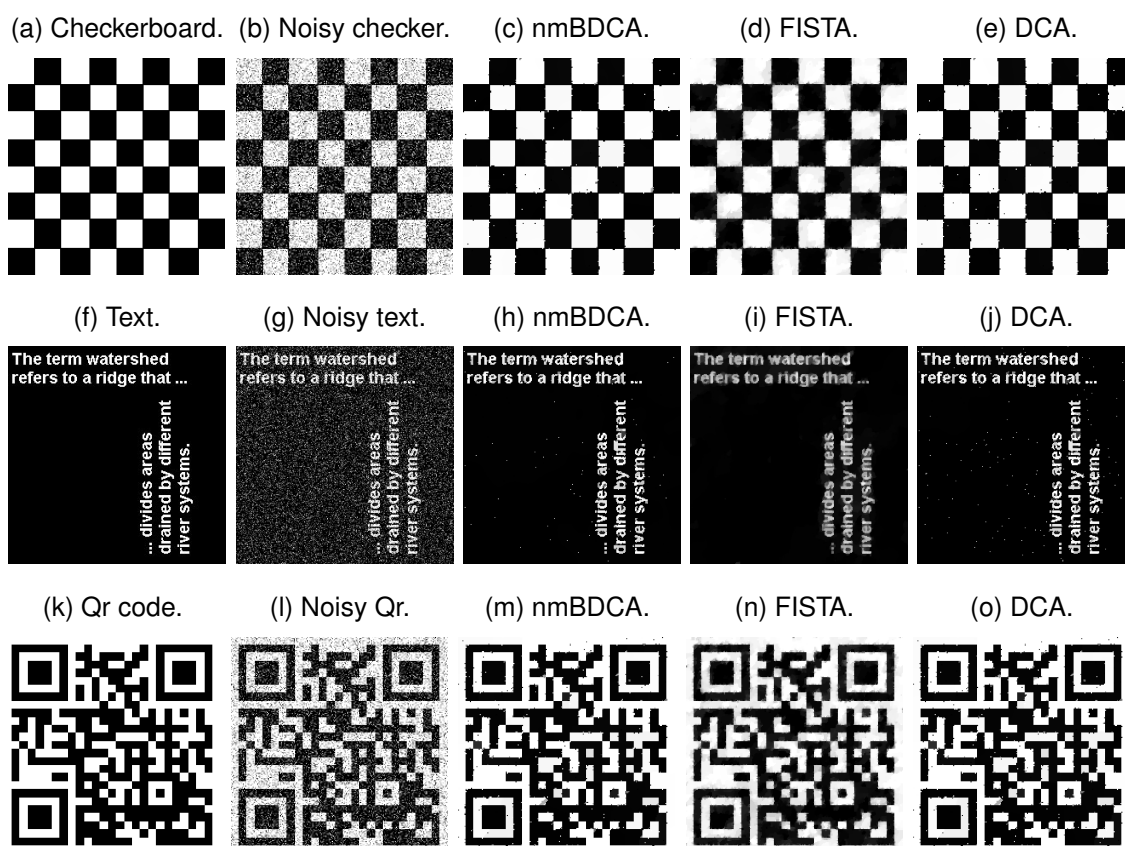


Figura 21 – Comparação de resultados para variância 0,15.

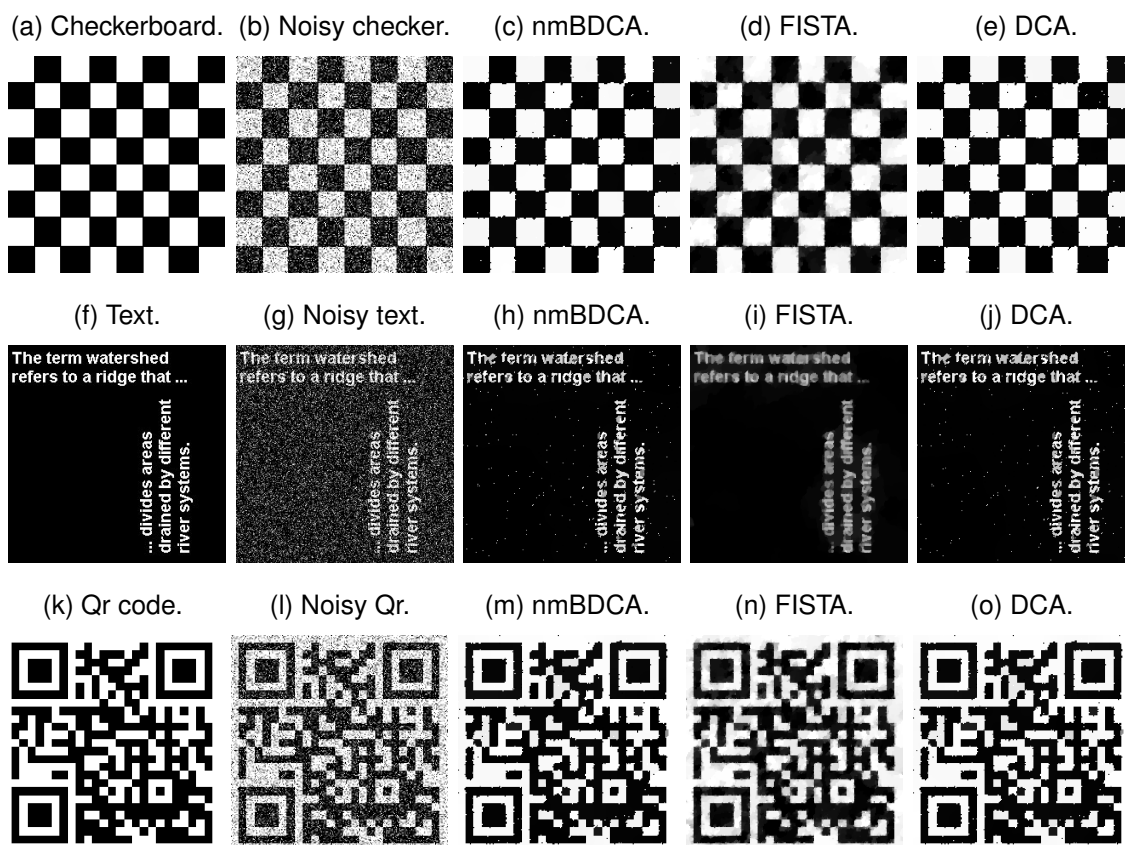


Tabela 10 – Métricas de avaliação para reconstruções com imagens preto e branco (A tolerância do FISTA é 5×10^{-10} . A tolerância das técnicas DC é 5×10^{-4}).

Variância 0,01										
Imag.	Téc.	μ	ζ	t_0	τ	ν	PSNR	SSIM	Iter.	Tempo (s)
Checkerboard	FISTA	0,09	-	-	-	-	31,5074	0,9024	30000	64,9197
	DCA	5	-	-	-	-	42,4985	0,9740	10,27	3,0467
	nmBDCA	5	0,05	0,9	0,15	50	42,5624	0,9740	7,13	2,6121
Text	FISTA	0,12	-	-	-	-	27,3387	0,9242	30000	150,5507
	DCA	6	-	-	-	-	36,8888	0,9647	17,95	21,9967
	nmBDCA	6	0,05	0,9	0,15	50	36,8933	0,9647	13,80	19,6283
Qr code	FISTA	0,09	-	-	-	-	28,6996	0,9400	30000	62,7289
	DCA	4	-	-	-	-	41,3309	0,9886	11,42	4,0008
	nmBDCA	4	0,05	0,9	0,15	50	41,3721	0,9887	8,31	3,4766
Variância 0,05										
Imag.	Téc.	μ	ζ	t_0	τ	ν	PSNR	SSIM	Iter.	Tempo (s)
Checkerboard	FISTA	0,2	-	-	-	-	24,9515	0,7951	30000	75,7891
	DCA	1,5	-	-	-	-	31,0011	0,9099	36,04	10,5928
	nmBDCA	1,5	0,05	0,9	0,15	100	31,0869	0,9108	26,60	9,5465
Text	FISTA	0,27	-	-	-	-	20,9555	0,8226	30000	113,9681
	DCA	1,5	-	-	-	-	28,4618	0,9114	65,67	97,1271
	nmBDCA	1,5	0,6	0,3	0,15	60	28,4640	0,9116	56,09	96,3565
Qr code	FISTA	0,17	-	-	-	-	22,1838	0,8599	30000	63,8970
	DCA	1,5	-	-	-	-	28,6500	0,9578	43,99	28,9894
	nmBDCA	1,5	0,05	0,9	0,15	500	28,6972	0,9581	29,10	22,9369
Variância 0,10										
Imag.	Téc.	μ	ζ	t_0	τ	ν	PSNR	SSIM	Iter.	Tempo (s)
Checkerboard	FISTA	0,31	-	-	-	-	22,3435	0,7338	30000	90,8217
	DCA	0,76	-	-	-	-	24,2283	0,8340	55,34	17,2762
	nmBDCA	0,76	0,05	0,9	0,15	100	24,3256	0,8355	41,22	15,3733
Text	FISTA	0,37	-	-	-	-	18,6978	0,7381	30000	125,5682
	DCA	1	-	-	-	-	22,0319	0,7995	107,92	161,9973
	nmBDCA	0,82	0,6	0,3	0,15	45	22,1315	0,8121	100,67	178,3707
Qr code	FISTA	0,28	-	-	-	-	19,4782	0,8097	30000	64,5098
	DCA	1	-	-	-	-	21,3256	0,8940	70,28	39,0311
	nmBDCA	1	0,05	0,9	0,15	500	21,3839	0,8959	45,79	29,1205
Variância 0,15										
Imag.	Téc.	μ	ζ	t_0	τ	ν	PSNR	SSIM	Iter.	Tempo (s)
Checkerboard	FISTA	0,37	-	-	-	-	20,8012	0,6979	30000	91,0962
	DCA	0,46	-	-	-	-	21,0772	0,7674	67,69	21,8354
	nmBDCA	0,45	0,05	0,9	0,15	500	21,1852	0,7687	43,88	16,9160
Text	FISTA	0,52	-	-	-	-	16,7676	0,6794	30000	108,8225
	DCA	0,62	-	-	-	-	18,9701	0,6932	136,51	186,7794
	nmBDCA	0,64	0,15	0,4	0,7	45	18,9767	0,6926	125,27	184,9044
Qr code	FISTA	0,34	-	-	-	-	17,9732	0,7700	30000	76,8756
	DCA	0,52	-	-	-	-	18,6159	0,8333	106,49	52,4293
	nmBDCA	0,52	0,05	0,9	0,15	300	18,6609	0,8345	81,62	48,2520

5.4 Reconstrução de imagens médicas

Semelhante à Seção 5.3, nesta seção são reconstruídas as imagens 4, 5 e 6 da Tabela 3, corrompidas por ruído Gaussiano com variâncias de 0,005, 0,0075 e 0,01. Os critérios de parada para configurar as técnicas nesta seção são os mesmos da seção 5.3. Os resultados foram obtidos calculando a média de cada métrica em 100 execuções com sementes aleatórias.

Apesar da realização das varreduras do parâmetro μ , aqui são usados gráficos de *boxplots* do PSNR e SSIM, obtidos para diferentes μ em cada modelo. Este gráfico apresenta os valores máximo, mínimo, mediano e discrepantes (*outliers*) e permite identificar a maior concentração de valores calculados, possibilitando avaliar o

desempenho dos modelos. Como na seção 5.3, os *boxplots* foram construídos a partir de 300 amostras coletadas, obtidas pela variação dos valores de μ de 0,01 a 0,99 e de 1 a 30. A utilização de *boxplots* destaca o desempenho qualitativo das técnicas a partir da faixa de bons valores de SSIM e PSNR (mediana e intervalo interquartil).

A análise entre os modelos convexo e não convexo pode ser feita comparando os *boxplots* SSIM e PSNR nas Figuras 22, 23 e 24, que mostram a mediana, o máximo, mínimo e a faixa de valores μ obtidos (tamanho do *boxplot*) para cada técnica, em cada variância de ruído testada. É possível notar que, para as três imagens de Tomografia computadorizada (TC), independente da variância, a presença massiva de valores mais baixos neste modelo torna a mediana do *boxplot* menor e a distância interquartilica maior que o do método DC, embora alguns valores máximos de PSNR e SSIM sejam maiores para o modelo convexo. Isso indica que é estatisticamente mais fácil calibrar o parâmetro μ do modelo não convexo para obter bom desempenho e, portanto, destaca a superioridade qualitativa (em PSNR e SSIM) do modelo não convexo nas reconstruções realizadas.

A avaliação usando os *boxplots* considera a concentração dos dados (localização do *boxplot*), e deixa claro que as medianas das técnicas DCA e nmBDCA, que usam o modelo não convexo, são maiores. Isso ocorre porque cada modelo possui uma faixa de bons valores μ . Neste caso, o modelo convexo possui uma faixa de bons valores menor que o não convexo; portanto, a determinação do μ , além de ser mais difícil, tem maior influência no desempenho do modelo. Verifica-se também que grande parte dos *outliers* (valores geralmente eliminados no pré-processamento dos dados) estão mais presentes nas métricas do modelo convexo. Isto destaca ainda mais o desempenho superior do modelo não convexo.

As Figuras 22, 23 e 24 também mostram que o desempenho qualitativo entre DCA e nmBDCA é semelhante. Isso ocorre porque nmBDCA aplica uma busca linear para superar o DCA em tempo de CPU, e ela geralmente não afeta as métricas qualitativas de PSNR e SSIM. Assim, os *boxplots* do DCA e nmBDCA mostram que as técnicas possuem uma faixa semelhante de bons valores de μ e que o valor escolhido para as reconstruções tem a mesma influência. O tempo de CPU entre DCA e nmBDCA pode ser avaliado na Tabela 11, que mostra as métricas calculadas para os experimentos com imagens de TC, obtidas por uma média de 100 execuções de cada técnica em cada variância aplicada.

A proposta do nmBDCA de acelerar a convergência do DCA é alcançada, pois o tempo de CPU do nmBDCA é inferior ao do DCA em todos os testes realizados. Esta tabela também mostra que, mesmo com *boxplots* semelhantes, o nmBDCA supera o DCA em termos de qualidade das reconstruções obtidas em todos os testes realizados, pois possui maior SSIM e PSNR. Nesta tabela, o cálculo das métricas SSIM e tempo

Figura 22 – Boxplots com variância 0,005.

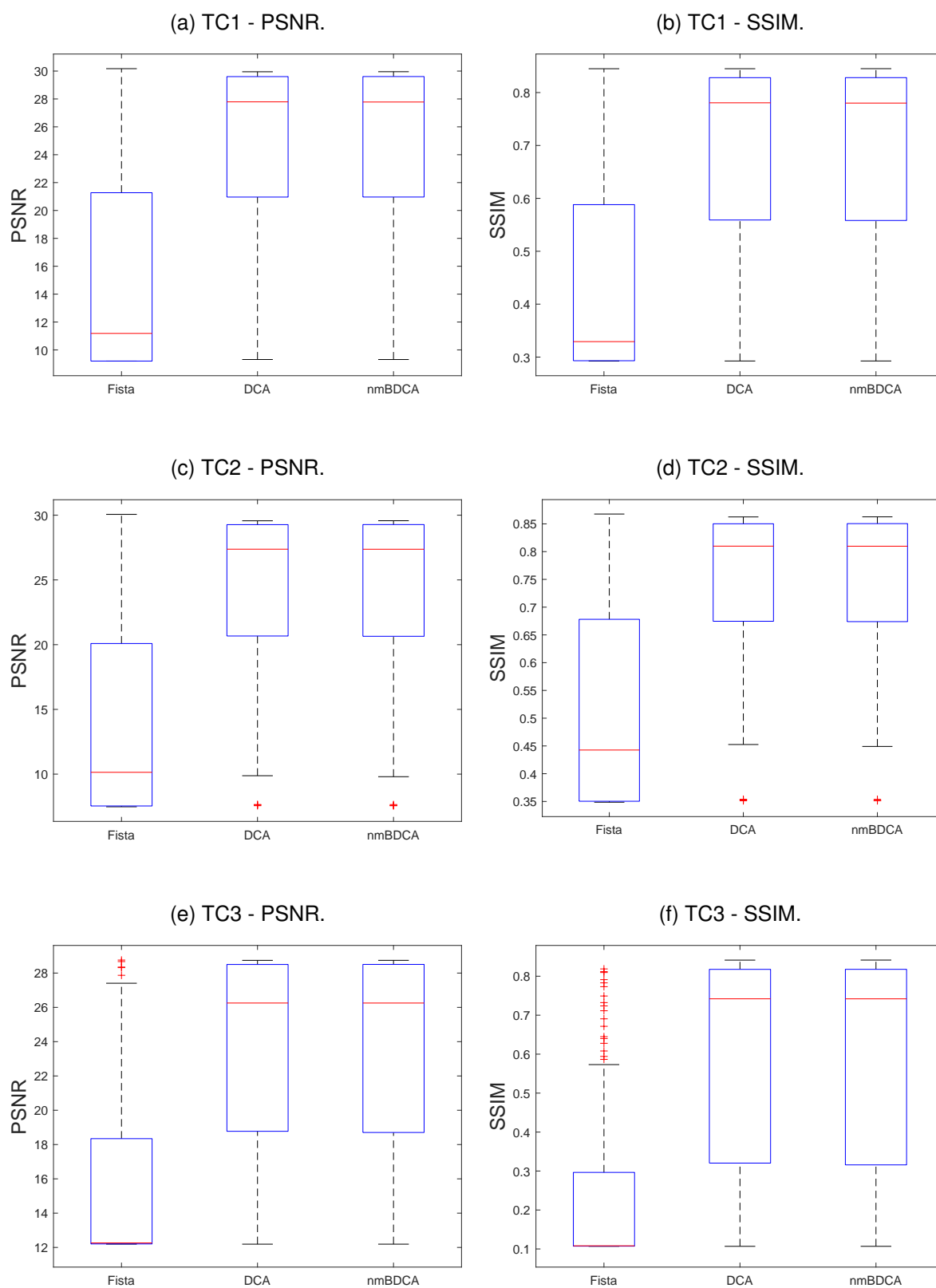


Figura 23 – Boxplots com variância 0,0075.

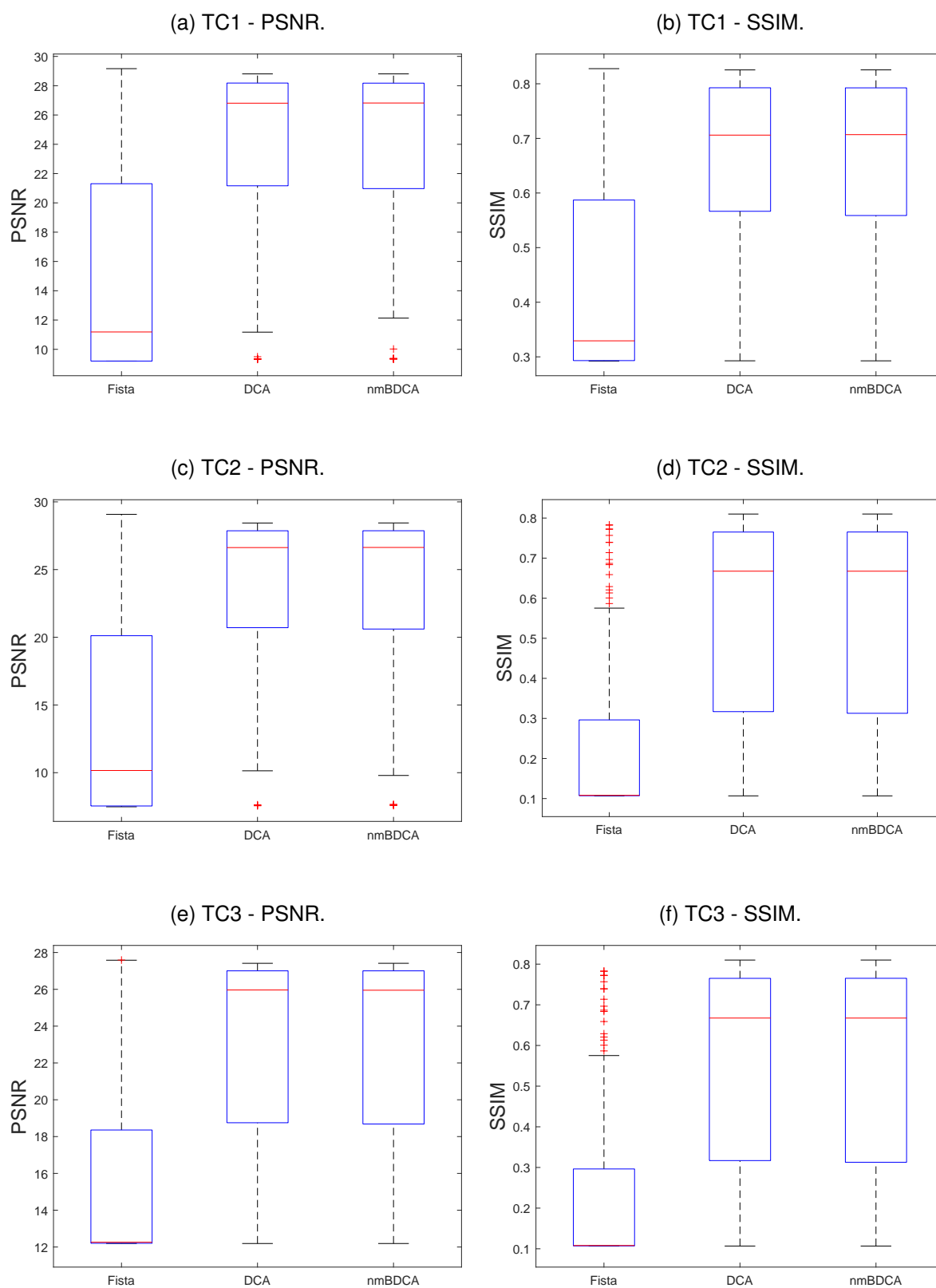
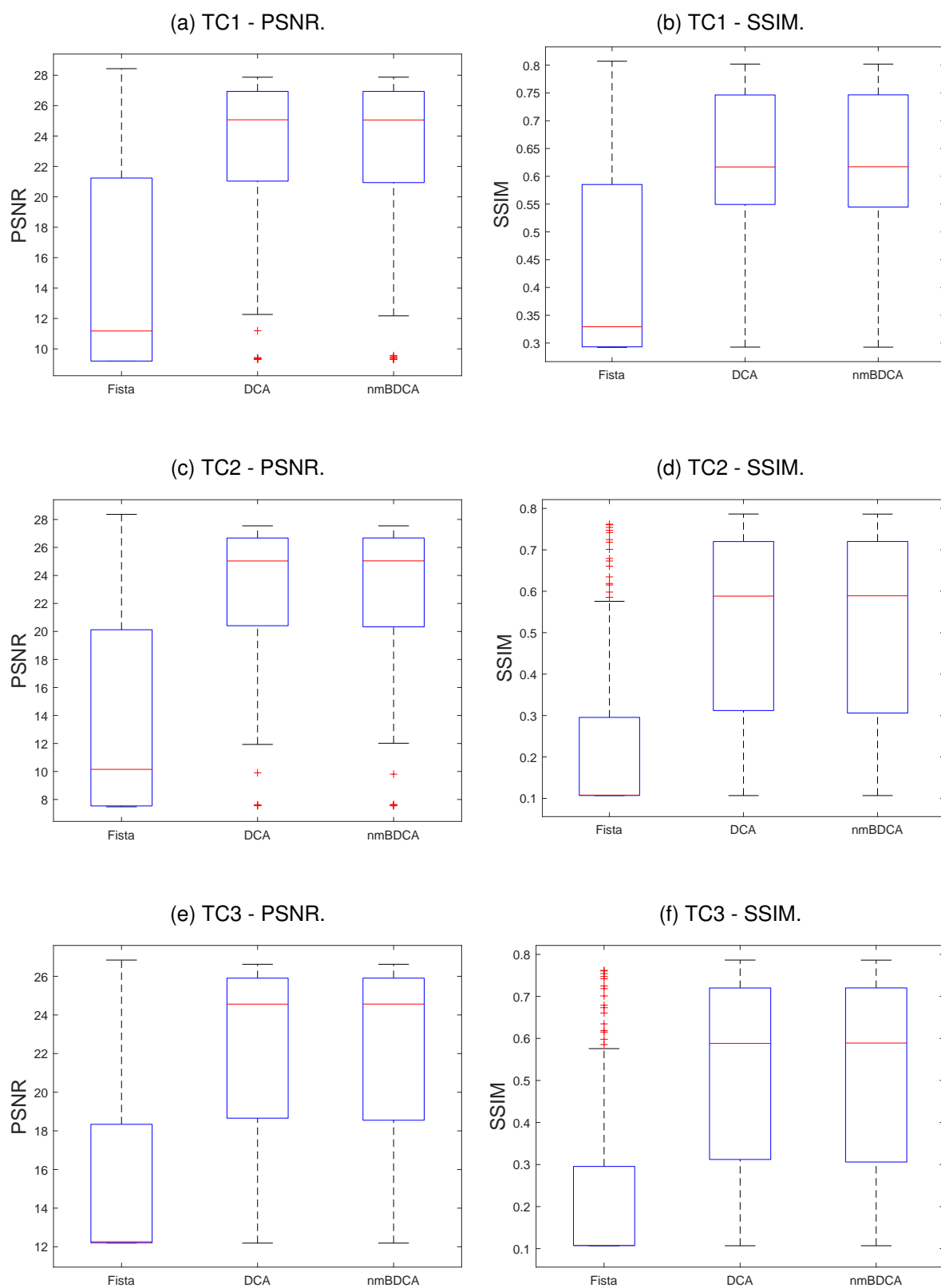


Figura 24 – Boxplots com variância 0,01.



de CPU também indica que o modelo não convexo supera o modelo convexo em todos os experimentos, sendo superior no PSNR em 77,78% dos testes. Isso indica que o modelo não convexo é mais adequado que o modelo convexo para a reconstrução de imagens médicas de TC e, em aplicações que dependem de tempo de CPU, mostra que o nmBDCA é mais adequado que o DCA.

A Tabela 11 também mostra os parâmetros utilizados por cada técnica. É notável que o parâmetro μ nos modelos convexo ($\mu < 1$) e não convexo ($\mu > 1$) varia significativamente em magnitude. Isso ressalta a importância de investigar esse parâmetro em reconstruções. A configuração da busca do nmBDCA foi feita com os parâmetros fixos $\zeta = 0,05$, $t_0 = 0,9$ e $\tau = 0,15$. O parâmetro ν foi alterado em alguns testes para garantir o menor tempo de CPU para nmBDCA (isso pode ser notado tanto nas iterações quanto no tempo de CPU) e demonstra, como na Seção 5.3, que cada problema requer o estudo dos parâmetros aplicados e quão trabalhoso isso pode ser. Nota-se também que o tempo de CPU e as iterações alcançadas podem não aumentar, mesmo com o aumento do ruído introduzido, devido à influência dos parâmetros das técnicas utilizadas, neste caso, o ν do mBDCA.

Tabela 11 – Métricas de avaliação para reconstruções com imagem de TC (A tolerância do FISTA é 5×10^{-10} . A tolerância das técnicas DC é 5×10^{-4}).

Variância 0,005										
Imagem	Técnica	μ	ζ	t_0	τ	ν	PSNR	SSIM	Iter.	Tempo (s)
TC1	FISTA	0,04	-	-	-	-	29,8854	0,8053	30000	73,5305
	DCA	10,9	-	-	-	-	29,5835	0,8369	10,23	3,0384
	nmBDCA	10,9	0,05	0,9	0,15	50	28,5832	0,8370	7,95	2,8957
TC2	FISTA	0,08	-	-	-	-	29,4364	0,8519	30000	74,3184
	DCA	15	-	-	-	-	29,6296	0,8634	8,95	1,9991
	nmBDCA	14,9	0,05	0,9	0,15	70	29,6352	0,8637	6,39	1,8723
TC3	FISTA	0,12	-	-	-	-	28,3119	0,8184	30000	73,2467
	DCA	8,4	-	-	-	-	28,5640	0,8389	13,39	4,4226
	nmBDCA	8,4	0,05	0,9	0,15	70	28,5859	0,8394	9,55	3,7460
Variância 0,0075										
Imagem	Técnica	μ	ζ	t_0	τ	ν	PSNR	SSIM	Iter.	Tempo (s)
TC1	FISTA	0,09	-	-	-	-	28,8052	0,8204	30000	75,3287
	DCA	10,6	-	-	-	-	28,7645	0,8217	13,42	3,8388
	nmBDCA	10,7	0,05	0,9	0,15	70	28,7727	0,8219	9,78	3,4021
TC2	FISTA	0,1	-	-	-	-	28,4091	0,8345	30000	71,1701
	DCA	11,1	-	-	-	-	28,3790	0,8418	12,23	3,1304
	nmBDCA	11	0,05	0,9	0,15	70	28,3818	0,8420	8,58	2,7792
TC3	FISTA	0,07	-	-	-	-	27,4055	0,7867	30000	72,2797
	DCA	9,9	-	-	-	-	27,3435	0,8097	20,01	7,5018
	nmBDCA	11,1	0,05	0,9	0,15	70	27,4569	0,8113	14,69	6,0179
Variância 0,01										
Imagem	Técnica	μ	ζ	t_0	τ	ν	PSNR	SSIM	Iter.	Tempo (s)
TC1	FISTA	0,12	-	-	-	-	27,8735	0,7954	30000	72,5502
	DCA	8,9	-	-	-	-	27,9039	0,8006	17,32	5,2131
	nmBDCA	8,8	0,05	0,9	0,15	80	27,9080	0,8012	13,53	5,0327
TC2	FISTA	0,12	-	-	-	-	27,6242	0,8207	30000	71,2268
	DCA	8,4	-	-	-	-	27,4315	0,8237	15,63	4,6286
	nmBDCA	8,4	0,05	0,9	0,15	70	27,4362	0,8239	11,99	4,3880
TC3	FISTA	0,09	-	-	-	-	26,4558	0,7636	30000	70,9048
	DCA	9	-	-	-	-	26,5457	0,7895	25,48	9,5697
	nmBDCA	9	0,05	0,9	0,15	80	26,5471	0,7895	20,71	9,2433

Assim como na seção 5.3, as imagens de TC reconstruídas são mostradas lado a lado nas Figuras 25, 26 e 27. Apesar da inserção do ruído Gaussiano, as técnicas aplicadas reconstróem as imagens de forma satisfatória, preservando a maior parte dos detalhes das imagens originais. Porém, mesmo com reconstruções visualmente semelhantes, é possível notar que o DCA e nmBDCA, que utilizam o modelo de TV não convexo, recuperam imagens com menos distorção. Isso fica evidente na comparação de texturas entre as imagens originais e recuperadas, onde é notável a presença de superfícies mais homogêneas nas imagens obtidas pelo FISTA, como nas Figuras 25i, 26d e 27n e menor remoção de ruído na Figura 25d.

Figura 25 – Reconstruções de imagens de TC com variância 0,005.

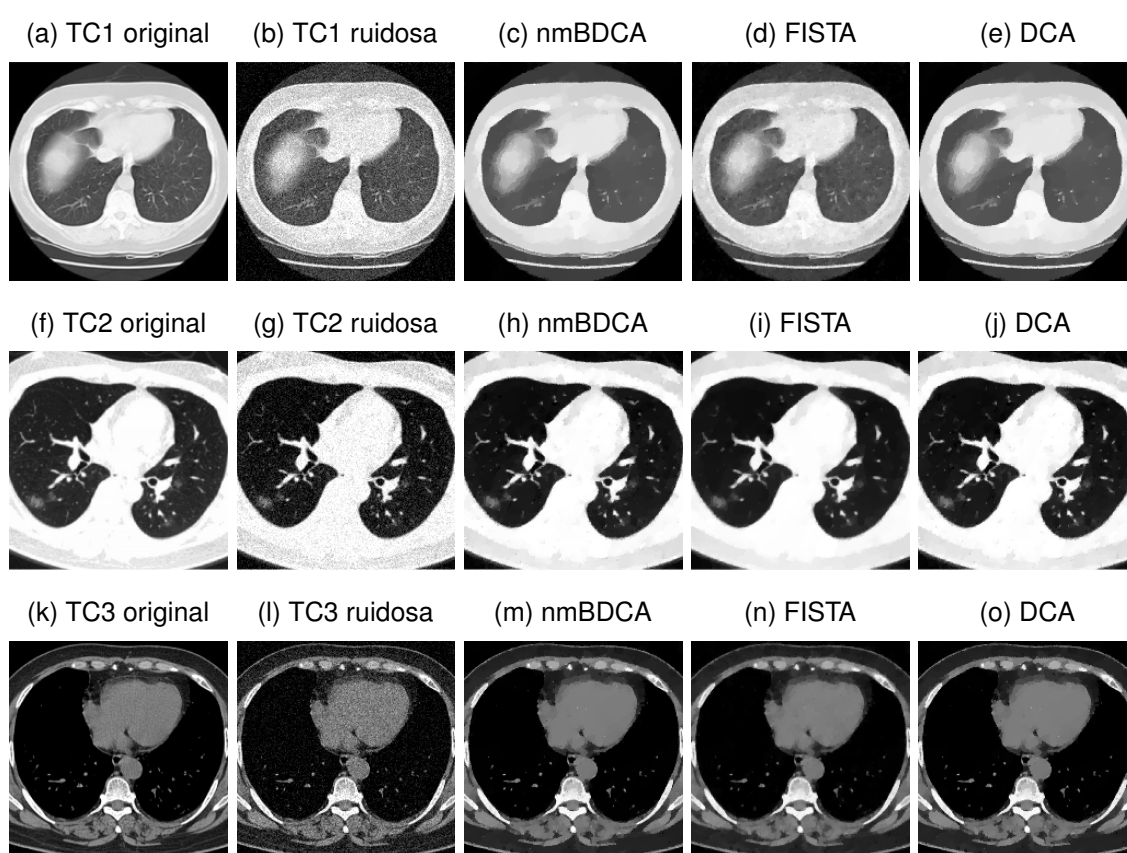


Figura 26 – Reconstruções de imagens de TC com variância 0,0075.

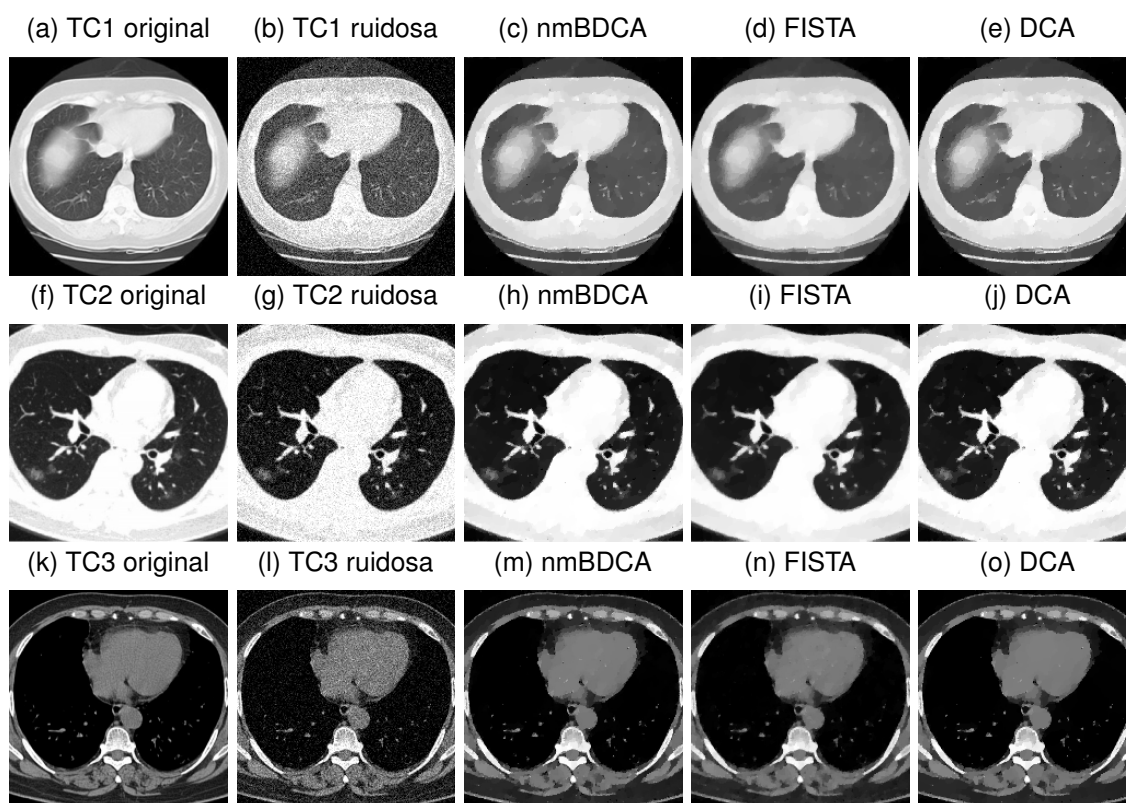
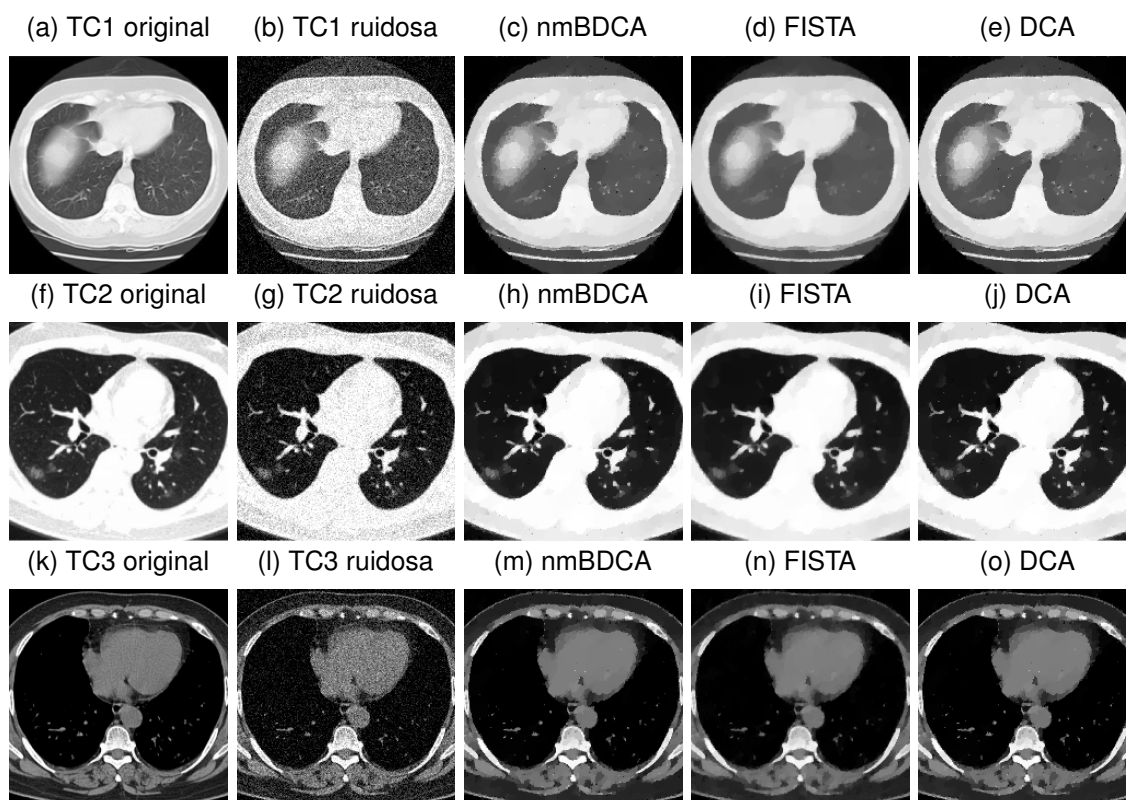


Figura 27 – Reconstruções de imagens de TC com variância 0,01.



5.5 Comparação entre nmBDCA e BDCA suavizado

Esta seção considera especificamente os métodos nmBDCA e uma suavização do BDCA para a reconstrução de imagens corrompidas por diferentes ruídos em diferentes níveis (ruídos Gaussiano e Riciano, muito presentes em aplicações de processamento de imagens e sinais). Assim, os experimentos são divididos em duas seções. A Seção 5.5.1 usa uma imagem real de ressonância magnética (RM) de tamanho 128×128 , disponível em <https://www.kaggle.com/datasets/sachinkumar413/alzheimer-mri-dataset/data>), corrompida por ruído Gaussiano. Por sua vez, a Seção 5.5.2 usa as imagens TC1 e TC2 descritas na Tabela 3, corrompidas por ruído Riciano.

Como já mencionado, para que o BDCA seja aplicável para problemas não convexos, a primeira componente ($g(x)$) do problema (4.10) deve ser diferenciável. Isso pode ser feito pela inserção de uma função suavização na componente $g(x)$. Neste trabalho, a suavização de $g(x)$ é feita com o uso da função Huber, definida por g_α , tal como:

$$g_\alpha(z) = \begin{cases} \frac{1}{2\alpha} \|z\|^2, & \|z\| \leq \alpha, \\ \|z\| - \frac{\alpha}{2}, & \|z\| > \alpha, \end{cases}$$

com $\alpha > 0$ e $z \in \mathbb{R}^n$ (nos testes realizados com o BDCA suavizado, $\alpha = 10^{-7}$). Mais informações sobre a função Huber podem ser encontradas em Huber (1992). A utilização da função Huber garante que o BDCA possa resolver o modelo não convexo. Porém, o uso da função de suavização na primeira componente DC torna o BDCA deste trabalho uma variação da versão original proposta por Artacho e Vuong (2020). Isso permite o desenvolvimento de outros estudos futuros.

Com a inserção da função de suavização na primeira componente, o algoritmo original do BDCA deve ser ajustado, de forma a inserir a função de suavização no problema a ser resolvido. Isso pode ser observado no Algoritmo 12, que corresponde à versão suavizada do BDCA. Nota-se que o problema (5.9) a ser resolvido é similar a (4.12) do Algoritmo 8, porém a primeira componente é $g_\alpha(x)$.

5.5.1 Experimentos com ruído Gaussiano

Esta seção mostra a comparação entre o nmBDCA e a versão suavizada do BDCA na reconstrução de uma imagem de ressonância magnética, quando corrompida por ruído Gaussiano com variâncias 0,005, 0,01 e 0,025. O parâmetro de regularização (μ) do modelo de imagem foi escolhido por varredura, como explicado nas seções anteriores, e permite estimar o (μ) que obtém o melhor SSIM para cada técnica.

As técnicas usam modelos não convexos para reconstrução de imagem e possuem as mesmas regras de parada usadas nos testes das Seções 5.3.2 e 5.4:

Algorithm 12 BDCA suavizado

-
- 1: Definir $t_{-1} > 0$, $\tau > 0$, $\zeta \in (0,1)$, $k = 0$, tol , $\text{stop_rule} = +\infty$;
 - 2: Selecionar um ponto inicial $x^0 \in \mathbb{R}^n$;
 - 3: **while** $\text{stop_rule} > \text{tol}$ **do**
 - 4: Definir $w^k \in \partial h(x^k)$ e calcular:

$$y^k \in \arg \min \{g_\alpha(x) - \langle w^k, x - x^k \rangle : x \in \mathbb{R}^n\}; \quad (5.9)$$

- 5: Calcular $d^k := y^k - x^k$. Se $\|d^k\| = 0$, pare e retorne x^k . Se não, calcular $t_k := \zeta^{j_k} t_{k-1}$, onde

$$j_k := \min \{j \in \mathbb{N} : \phi(y^k + \zeta^j t_{k-1} d^k) \leq \phi(y^k) - \tau (\zeta^j t_{k-1})^2 \|d^k\|^2\};$$

- 6: Calcular $x^{k+1} := y^k + t_k d^k$;
 - 7: $\text{stop_rule} = \|x^{k+1} - x^k\|$;
 - 8: $k = k + 1$.
 - 9: **end while**
-

máximo de 200 iterações e tolerância de parada 5×10^{-4} , com função de penalidade $\phi = \phi_{Atan}$, parâmetro de função fortemente convexa $\rho = 1$. O modelo e as funções g e h são os mesmos definidos na Seção 5.1.1, modelo (5.3) e funções (5.4). Porém, a inserção da suavização na primeira componente DC torna $TV(x) = TV_\alpha(x)$. As técnicas BDCA suavizado e nmBDCA também requerem a configuração de parâmetros específicos ($\zeta = 0,05$, $t_0 = 0,9$, $\tau = 0,15$) aplicados na busca Armijo. No nmBDCA, um parâmetro adicional é considerado $\nu_k = \frac{\nu \|d^k\|^2}{k}$, e corresponde à sequência gerada pelo nmBDCA a ser inserida na busca realizada. É importante destacar que os parâmetros de busca tanto para o nmBDCA quanto para o BDCA suavizado são os mesmos, para definir o mesmo cenário de busca entre as técnicas, e corroborar ainda mais os resultados alcançados.

A Tabela 12 mostra os resultados de reconstrução, obtidos pela média de cada métrica em 100 execuções com sementes aleatórias e cada parâmetro utilizado na configuração dos métodos nmBDCA e BDCA suavizado (o μ que fornece o melhor SSIM e os parâmetros específicos para cada técnica). Os dados da Tabela 12 mostram que as métricas PSNR e SSIM possuem valores próximos, mas os tempos de CPU entre BDCA suavizado e nmBDCA são diferentes, sendo muito maiores para BDCA suavizado.

Isso demonstra que o nmBDCA é melhor que o BDCA suavizado em aplicações onde o custo computacional é importante (destacam-se aplicações com recuperação de imagens em tempo real, onde os dados adquiridos são processados à medida que são coletados). Por outro lado, os resultados também indicam que os métodos utilizados podem ser aplicados a problemas que envolvem o processamento de imagens com diferentes características de cor e textura. Isto é importante porque mostra que eles são versáteis e gerais e podem ser usados para outros problemas de minimização. Em particular, a reconstrução de imagens de ressonância magnética de qualidade indica que o BDCA suavizado e o nmBDCA são promissores para aplicações médicas

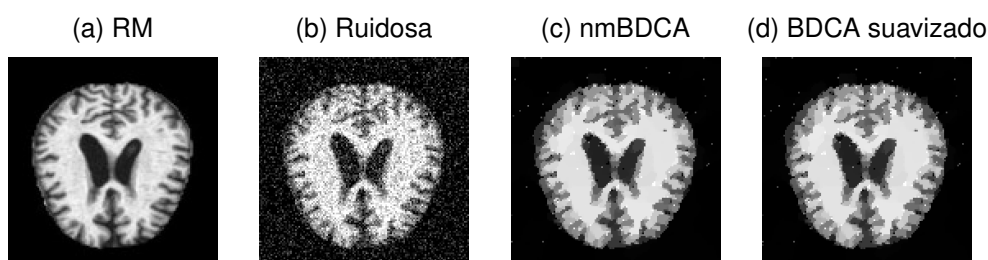
Tabela 12 – Métricas de avaliação para RM com ruído Gaussiano

Variância 0,005						
Imagem	Técnica	μ	ν	PSNR	SSIM	Tempo (s)
RM	BDCA suavizado	12,2	-	28,3481	0,8999	28,3
	nmBDCA	12,2	100	28,3495	0,9000	1,7
Variância 0,01						
Imagem	Técnica	μ	ν	PSNR	SSIM	Tempo (s)
RM	BDCA suavizado	8	-	28,1440	0,8525	54,2
	nmBDCA	8	100	28,1458	0,8527	4,0
Variância 0,025						
Imagem	Técnica	μ	ν	PSNR	SSIM	Tempo (s)
RM	BDCA suavizado	5	-	23,1785	0,7438	99,1
	nmBDCA	5	100	23,1863	0,7448	9,9

semelhantes.

A Figura 28 mostra, para ruído Gaussiano com variância 0,025, a comparação entre a imagem original de RM e as imagens reconstruídas. As reconstruções nmBDCA e BDCA suavizado são satisfatórias e de qualidade semelhante.

Figura 28 – Reconstruções das imagens de RM com ruído Gaussiano de variância 0,025



5.5.2 Experimentos com ruído Riciano

Semelhante à Seção 5.5.1, esta seção usa as imagens TC1 e TC2 descritas na Tabela 3, porém corrompidas por ruído Riciano com variâncias 0,10, 0,15 e 0,20. São usados os parâmetros $\zeta = 0,05$, $t_0 = 0,9$, $\tau = 0,15$ nas buscas do BDCA suavizado e nmBDCA. No nmBDCA, o mesmo parâmetro $\nu_k = \frac{\nu \|d^k\|^2}{k}$ é usado.

Muitos problemas de imagem envolvem o processamento de imagens médicas, que consistem em dados complexos que representam magnitude, fase e frequência. Eles são obtidos através da transformada inversa de Fourier e servem para representar as características de um paciente, por exemplo. Normalmente, esses dados podem ser corrompidos por um tipo de ruído conhecido como ruído Riciano, que depende dos sinais elétricos gerados durante o processo de aquisição.

Neste contexto, uma prática comum é transformar imagens de valores complexos em imagens com magnitude e fase. Nesta transformação, os dados de magnitude no domínio espacial são modelados como distribuição Riciano. O ruído é denominado

ruído Riciano e corresponde ao erro entre as intensidades da imagem subjacente e os dados de medição (MOHAN; KRISHNAVENI; GUO, 2014).

A distribuição de probabilidade Riciano de uma imagem corrompida f é dada por:

$$P(f|u) = \frac{f}{\sigma^2} \exp\left(-\frac{u^2 + f^2}{2\sigma^2}\right) I_0\left(\frac{fu}{\sigma^2}\right), \quad (5.10)$$

onde I_0 é o primeiro tipo de função de Bessel modificada com ordem zero (BOWMAN, 2012), σ^2 denota a variância do ruído gaussiano nas imagens reais e imaginárias (assumidas iguais) e $f = \sqrt{(u + \eta_1)^2 + \eta_2^2}$, onde u é a verdadeira amplitude da imagem, e η_1, η_2 representa o ruído gaussiano branco independente com o mesmo desvio padrão σ . Sugerimos a leitura de Sijbers et al. (1998), Chen et al. (2018), Gudbjartsson e Patz (1995) e Macovski (1996) para que se obtenham mais informações sobre a modelagem do ruído Riciano.

Considerando (5.10), Wu et al. (2022) propõe um modelo MAP-Riciano não suave e não convexo para remoção de ruído de imagens dado por:

$$\inf_{u \in BV(\Omega)} \left\{ E(u) = \frac{1}{2\sigma^2} \int_{\Omega} u^2 dx - \int_{\Omega} \log\left(I_0\left(\frac{fu}{\sigma^2}\right)\right) dx + \gamma \int_{\Omega} |\nabla u| \right\}, \quad (5.11)$$

onde γ é uma constante de regularização não negativa, D é um operador gradiente, $\int_{\Omega} |\nabla u|$ é a variação total TV e $BV(\Omega)$ é o subespaço de funções $u \in L^1(\Omega)$ tal que a seguinte quantidade é finita:

$$J(u) = \sup \left\{ \int_{\Omega} u(x) \operatorname{div} \varphi(x) dx \mid \varphi \in C_0^\infty(\Omega, \mathbb{R}^2), \|\varphi\|_{L^\infty(\Omega, \mathbb{R}^2)} \leq 1 \right\}.$$

Considerando (5.11), o modelo a ser minimizado por BDCA suavizado e nmBDCA é dado por:

$$\inf_{u \in \mathbb{R}^n, u \geq 0} \left\{ F(u) = \frac{1}{2\sigma^2} \|u\|^2 - \langle \log\left(I_0\left(\frac{fu}{\sigma^2}\right)\right), \mathbf{1} \rangle + \gamma TV(u) \right\}, \quad (5.12)$$

onde o vetor $\mathbf{1} = (1, 1, \dots, 1)^\top \in \mathbb{R}^n$, $TV(u) = TV_\alpha(u) = \sum_{i=1}^n h_\alpha((\nabla_h u)_i, (\nabla_v u)_i)$ é uma aproximação suave de TV isotrópica para BDCA suavizado, com h_α definido por (5.5); veja Wu et al. (2022, Seção 3) para mais detalhes sobre TV_α . A partir de (5.12), as funções $g(u)$ e $h(u)$ do problema DC a ser resolvido por BDCA suavizado e nmBDCA podem ser definidas por:

$$\begin{aligned} g(u) &= \frac{1}{2\sigma^2} \|u\|^2 + \gamma TV(u) + \frac{\rho}{2} \|u\|^2 \\ h(u) &= \langle \log\left(I_0\left(\frac{fu}{\sigma^2}\right)\right), \mathbf{1} \rangle + \frac{\rho}{2} \|u\|^2, \end{aligned} \quad (5.13)$$

onde ρ é uma constante positiva tornando os componentes $g(u)$ e $h(u)$ fortemente convexos, e a suposição $\rho > \tau$, uma constante positiva. Assim como nos experimentos das seções anteriores, o FISTA na versão FGP é aplicado para resolver o subproblema

(5.9) e, como o modelo a ser resolvido é (5.12), com componentes DC dados em (5.13), a cada iteração o FGP deve resolver um problema no forma de (5.5), onde $s^k = \frac{w^k \sigma^2}{1+\rho}$ e $\lambda = \frac{\gamma \sigma^2}{1+\rho}$.

A configuração do parâmetro de regularização (γ) de (5.12) também foi definida pela varredura dos valores deste parâmetro, nas mesmas condições descritas na Seção 5.3. Em relação aos critérios de parada utilizados, apenas o parâmetro da função fortemente convexa foi alterado para $\rho = 10^{-6}$. Todos os resultados apresentados na Tabela 13 foram obtidos calculando a média de cada métrica em 100 execuções, com sementes aleatórias, da mesma forma que na Seção 5.5.1.

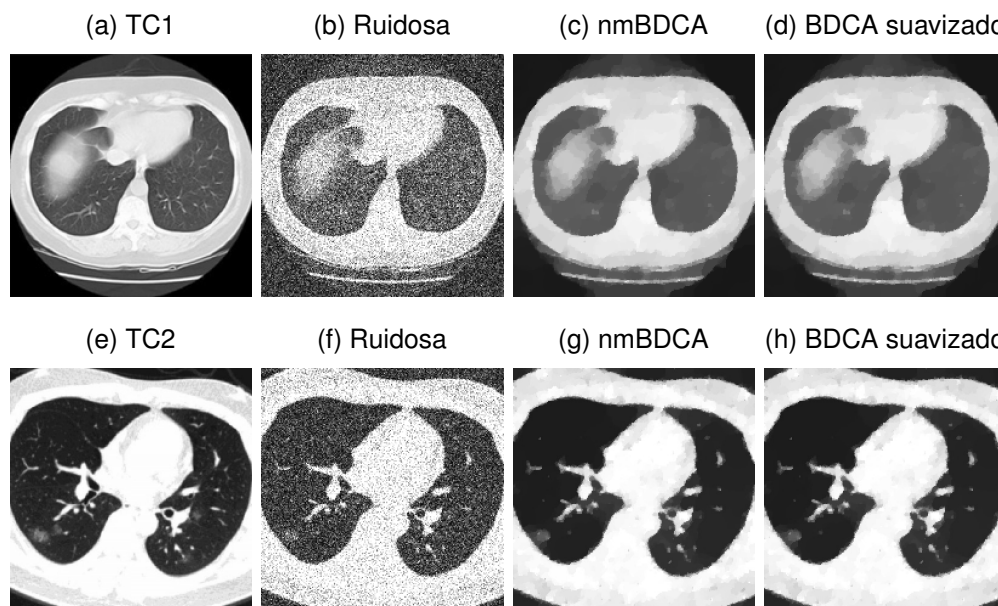
Tabela 13 – Métricas de avaliação para os experimentos com ruído Riciano

Variância 0,10						
Imagem	Técnica	γ	ν	PSNR	SSIM	Tempo (s)
CT1	BDCA suavizado	8,4	-	27,3707	0,6979	58,3
	nmBDCA	8,4	50	27,3714	0,6979	3,9
CT2	BDCA suavizado	7,5	-	27,7105	0,8256	32,5
	nmBDCA	7,5	50	27,7102	0,8256	2,3
Variância 0,15						
Imagem	Técnica	γ	ν	PSNR	SSIM	Tempo (s)
CT1	BDCA suavizado	5,9	-	25,0072	0,6461	71,0
	nmBDCA	5,9	50	25,0064	0,6461	4,9
CT2	BDCA suavizado	5,5	-	25,0808	0,7762	45,6
	nmBDCA	5,5	50	25,0815	0,7762	3,7
Variância 0,20						
Imagem	Técnica	γ	ν	PSNR	SSIM	Tempo (s)
CT1	BDCA suavizado	5	-	23,0764	0,6088	74,7
	nmBDCA	5	50	23,0765	0,6088	5,8
CT2	BDCA suavizado	4,3	-	23,0628	0,7358	52,3
	nmBDCA	4,3	50	23,0626	0,7358	4,2

A Tabela 13 contém os parâmetros de configuração (γ e ν) e as métricas de avaliação calculadas. Pode-se notar que em cada variância o parâmetro γ é ajustado, para que se obtenha o melhor SSIM possível. Porém, nesses experimentos, a manutenção do parâmetro $\nu = 50$ foi suficiente para permitir que o nmBDCA obtivesse menor tempo de CPU que o BDCA suavizado em todos os testes realizados. É possível observar que os desempenhos de qualidade de imagem (métricas PSNR e SSIM) são semelhantes, mas diferentes em relação ao tempo de CPU, como na Seção 5.5.1.

A comparação entre as imagens obtidas com nmBDCA e BDCA suavizado pode ser vista na Figura 29, e mostra, assim como na Seção 5.5.1, que as reconstruções são satisfatórias, e recuperam imagens originais de qualidade. Porém, mesmo que os desempenhos sejam qualitativamente semelhantes, a Tabela 13 comprova que o nmBDCA é superior ao BDCA suavizado em termos de tempo de CPU, obtendo imagens com menor custo computacional.

Figura 29 – Reconstruções das imagens de TC corrompidas por ruído Riciano de variância 0,20.



5.6 Comentários finais dos experimentos

Como explicado na Seção 4.3, o nmBDCA não precisa da diferenciabilidade do primeiro componente DC, e aplica uma busca não monótona controlada por uma sequência, parametrizada por ν , que calcula uma direção de descida maior que o DCA e permite obter a diminuição adicional no valor da função objetivo. Isso proporciona convergência em menor tempo de CPU e manutenção da qualidade de reconstrução das imagens analisadas, além de destacar a busca não monótona como ponto crucial do nmBDCA.

A busca realizada contém parâmetros $(t_0, \tau, \zeta$ e $\nu)$ que determinam o seu ponto inicial, ajustam sua intensidade, determinam a etapa de atualização e geram a sequência que controla o crescimento do valor da função objetivo (quando ocorre). Esses parâmetros podem ser considerados uma dificuldade na aplicação do método, pois sua configuração é necessária particular a cada problema. Porém, incentiva o surgimento de pesquisas, principalmente para a otimização.

O nmBDCA é aplicado com o modelo TV não convexo, que se diferencia do modelo convexo por apresentar uma função (ϕ) que penaliza a variação total e é aplicada para preservar mais detalhes da imagem, principalmente bordas e textura. Isto é importante, porque determinar um modelo que reconstrói melhores imagens permite seu uso em problemas onde a qualidade da imagem é essencial. Embora as funções ϕ sejam côncavas, o que tornaria o problema estudado não convexo e, portanto, de difícil solução, existem condições em que a aplicação de ϕ possibilita resolver o problema não convexo usando métodos DC. Desta forma, a abordagem de otimização DC pode

ser aplicada, e melhorias no tempo de CPU e na qualidade da reconstrução da imagem podem ser alcançadas.

A função ϕ usada neste trabalho é ϕ_{atan} , aplicada por Oliveira e Tcheou (2019). Porém, conforme descrito na Seção 5.1.2 e exemplificado pela Tabela 4, existem outras funções ϕ que satisfazem as condições de aplicação no modelo de TV não convexo, cuja inserção pode interferir no desempenho do modelo. Assim, trabalhos futuros envolvem o desenvolvimento de um conjunto de funções que satisfaçam as condições de aplicação no modelo não convexo, de forma a analisar a influência delas no desempenho do modelo e, também, estudar a possibilidade de tornar o método DC aplicado mais eficiente.

O parâmetro de configuração do modelo de imagem (μ) foi ajustado com base na varredura realizada para cada técnica. A Seção 5.3 apresenta os gráficos de varredura para cada imagem e variância aplicada. É possível notar que a faixa de valores bons de μ é diferente entre as técnicas FISTA (modelo convexo) e técnicas DC (modelo não convexo) e que a maior quantidade dos maiores valores de PSNR e SSIM está nas técnicas DC. Isso indica que o modelo não convexo é superior ao modelo convexo para a reconstrução de imagens nos experimentos realizados. As varreduras também mostram que o DCA e nmBDCA retornam PSNR e SSIM semelhantes para o mesmo μ . Porém, a Tabela 10 mostra que o nmBDCA, além de reconstruir imagens com menos tempo de CPU que o DCA, possui SSIM e PSNR superiores na maioria dos experimentos realizados.

A Seção 5.4 apresenta os resultados de experimentos com imagens médicas de TC do COVID-19. O parâmetro μ também é avaliado para configurar as técnicas, mas as varreduras são apresentadas com *boxplots* para destacar os valores calculados de PSNR e SSIM (*outliers*, máximo, mínimo e mediana). Isso destaca estatisticamente os valores de PSNR e SSIM obtidos pelos modelos convexos e não convexos, mostrando que o modelo não convexo contém a maior quantidade de μ que atinge os melhores PSNR e SSIM. O uso dos *boxplots* na Seção 5.4 mostra que o DCA e o nmBDCA possuem desempenhos semelhantes em termos de métricas de qualidade, mas são diferentes em termos de tempo de CPU para reconstruções. Isso é mostrado na Tabela 11, que apresenta os parâmetros utilizados e as medidas calculadas em cada variância para cada imagem.

A comparação específica entre nmBDCA e a versão suavizada do BDCA (este trabalho aplica a função Huber na primeira componente DC) é feita na Seção 5.5, que mostra a superioridade do nmBDCA principalmente quanto ao tempo de CPU. Percebe-se que as imagens reconstruídas nessa seção possuem boa qualidade, recuperando a maior parte dos detalhes das imagens originais. Assim, esses resultados mostram que o nmBDCA é indicado para aplicações onde o tempo de CPU é mais importante e que

as técnicas são promissoras para reconstrução de imagens médicas.

Este trabalho se limita ao estudo da remoção do ruído em imagens preto e branco e médicas de TC e RM. As seções de experimentos são divididas em testes que corrompem imagens com ruído Gaussiano e Riciano. Portanto, para outros ruídos, como ruído impulsivo (sal e pimenta), ruído de Poisson, o modelo de imagem deve contemplar ajustes no cálculo da variação total. Assim, este trabalho incentiva a análise futura de problemas com outros tipos de ruído e outros tipos de imagens.

6 Conclusões e considerações finais

Este trabalho propôs recuperar imagens corrompidas por ruído Gaussiano e Riciano, avaliando a influência do modelo de imagem aplicado. Para isso, utilizou o nmBDCA para reconstruir as imagens corrompidas, com maior qualidade e menor tempo de CPU que o DCA, e, ainda, comparou o desempenho dos modelos TV convexo e não convexo, com a inserção de diferentes níveis de ruído. Além disso, propôs uma versão suavizada do BDCA que recupera imagens com qualidade semelhante ao nmBDCA.

Na Seção 5.3, a análise do tempo de CPU e da qualidade das reconstruções do nmBDCA e a comparação com o DCA mostra que as imagens obtidas com nmBDCA são de melhor qualidade, porque apresentam melhor PSNR em todos os experimentos e melhores SSIM e tempo de CPU em 91,67 % dos testes realizados. Isto é um indício de que a aplicação do nmBDCA é mais adequada para problemas cujo desempenho computacional é crucial e que o nmBDCA com TV não convexa também é indicado para problemas onde a recuperação e tratamento da imagem é importante. Desta forma, os resultados alcançados são satisfatórios, e há uma grande contribuição para a área de otimização DC, que requer processamento de imagens.

Na mesma seção, a comparação entre os modelos não convexo e convexo, especificamente entre nmBDCA e FISTA, destaca que o modelo não convexo é mais robusto ao aumento do ruído inserido nas imagens, pois as métricas PSNR e SSIM são superiores em 100% dos testes realizados. Portanto, a comparação contribui com a determinação de modelos para problemas de reconstrução de imagens que envolvem diferentes níveis de ruído e sugere uma preferência pela escolha do modelo de TV não convexo.

A Seção 5.4 apresenta os *boxplots* semelhantes do desempenho qualitativo entre DCA e nmBDCA, mas a Tabela 11 indica a superioridade do nmBDCA quando a avaliação é feita sobre a média de execuções. Em relação ao tempo de CPU, o nmBDCA supera o DCA em todos os experimentos realizados com imagens de TC e, portanto, a proposta do nmBDCA de acelerar a convergência do DCA é alcançada. A comparação entre os modelos convexo e não convexo mostra que o modelo não convexo supera o modelo convexo apresentando maior SSIM e tempo de CPU em todos os testes e maior PSNR em 77,78% deles. Assim, indica que o modelo não convexo é mais adequado que o convexo para a reconstrução de imagens médicas de TC e que o nmBDCA é mais adequado para aplicações que dependem de tempo de CPU.

A comparação entre nmBDCA e o BDCA suavizado na reconstrução de imagens com diferentes níveis e tipos de ruído apresentada na Seção 5.5 mostra que o nmBDCA é superior, principalmente em tempo de CPU, e que ambas as técnicas podem ser aplicadas para a recuperação de imagens, mesmo com diferentes tipos de ruído. Em termos de qualidade, o desempenho das técnicas é similar (na maioria dos testes, o nmBDCA é superior), porém, para problemas cujo tempo de CPU é muito importante, o nmBDCA se destaca e é mais indicado.

Existem desafios na configuração dos parâmetros do modelo de imagem e na definição dos parâmetros de busca do nmBDCA e BDCA suavizado. O parâmetro de regularização μ tem influência determinante na qualidade das imagens obtidas e foi analisado antes das reconstruções, para que fosse determinado o melhor parâmetro para cada teste. Isso permitiu entender que cada modelo possui uma faixa distinta de valores μ que possibilita alcançar bons resultados.

Os parâmetros de busca do nmBDCA (ζ , t_0 , τ e ν) e do BDCA suavizado (ζ , t_0 , τ) estão diretamente ligados ao tempo de CPU que cada técnica leva para reconstruir as imagens. O número de parâmetros a serem ajustados é um desafio, pois cada teste pode exigir sua configuração individual. Além disso, também é importante analisar o valor inicial da etapa de redução da busca Armijo, pois isso afeta diretamente o tempo de CPU do nmBDCA.

Em pesquisas futuras, considera-se: analisar o impacto de outros tipos de ruído; usar conjuntos de dados de imagens com características diferentes (imagens coloridas, por exemplo); examinar o impacto das funções de penalidade no modelo não convexo e relatar a sua influência no desempenho do modelo.

7 Apêndices

7.1 Demonstração do Lema 4.1

De fato, a expressão (4.11) é equivalente a

$$w^k \in \partial g(y^k).$$

Suponha que y^k seja solução de (4.11). Então, como a função do subproblema é, em particular, convexa, tem-se que

$$0 \in \partial(g(\cdot) - \langle w^k, \cdot \rangle + \langle w^k, x^k \rangle)(y^k).$$

Pela Proposição 3.2 e Proposição 3.8, segue que $0 \in \partial g(y^k) - w^k$, que é, $w^k \in \partial g(y^k)$. Por outro lado, $w^k \in \partial h(x^k)$, devido ao Passo 2. Conseqüentemente, se $y^k = x^k$, segue que $\partial g(x^k) \cap \partial h(x^k) \neq \emptyset$. Logo, x^k é ponto crítico.

7.2 Demonstração da Proposição 4.1

De fato, conforme passo 3, o Algoritmo 7 termina quando $y^k = x^k$. Pelo Lema 4.1, x^k é um ponto crítico, e isso prova o item (i). Suponha que $x^{k+1} \neq x^k, \forall k \in \mathbb{N}$. Como $w^k \in \partial h(x^k) \cap \partial g(x^{k+1})$ para qualquer $k \in \mathbb{N}$, aplicando o item (iii) do Teorema 3.7 tem-se que:

$$g(x^k) - g(x^{k+1}) \geq \langle w^k, x^k - x^{k+1} \rangle + \frac{\rho}{2} \|x^k - x^{k+1}\|^2 \quad (7.1)$$

e

$$h(x^{k+1}) - h(x^k) \geq \langle w^k, x^{k+1} - x^k \rangle + \frac{\rho}{2} \|x^{k+1} - x^k\|^2. \quad (7.2)$$

Somando as desigualdades (7.1) e (7.2) termo a termo, tem-se

$$\begin{aligned} [g(x^k) - h(x^k)] - [g(x^{k+1}) - h(x^{k+1})] &\geq \langle w^k, x^k - x^{k+1} \rangle + \langle w^k, x^{k+1} - x^k \rangle + \rho \|x^k - x^{k+1}\|^2 \\ &= \langle w^k, x^k - x^{k+1} \rangle - \langle w^k, x^k - x^{k+1} \rangle + \rho \|x^k - x^{k+1}\|^2 \\ &= \rho \|x^k - x^{k+1}\|^2. \end{aligned}$$

Então

$$\rho \|x^k - x^{k+1}\|^2 \leq f(x^k) - f(x^{k+1}), \quad \forall k \in \mathbb{N}. \quad (7.3)$$

Em particular, $x^{k+1} \neq x^k$ implica $\|x^k - x^{k+1}\|^2 > 0$. Por isso,

$$0 < f(x^k) - f(x^{k+1}), \forall k \in \mathbb{N}.$$

Então, $f(x^{k+1}) < f(x^k)$ para qualquer $k \in \mathbb{N}$ e isso prova que $\{f(x^k)\}$ é decrescente.

7.3 Demonstração da Proposição 4.2

Considerando a soma parcial na desigualdade (7.3):

$$0 \leq \sum_{k=0}^N \rho \|x^k - x^{k+1}\|^2 \leq \sum_{k=0}^N [f(x^k) - f(x^{k+1})]$$

obtém-se

$$0 \leq \sum_{k=0}^N \rho \|x^k - x^{k+1}\|^2 \leq f(x^0) - f(x^{N+1}).$$

Pelas hipóteses **(H2)** e **(H3)**, $f^* \leq f(x^{N+1})$, para todo $N \in \mathbb{N}$, que implica que $-f(x^{N+1}) \leq -f^*$, para todo $N \in \mathbb{N}$. Portanto,

$$0 \leq \sum_{k=0}^N \rho \|x^k - x^{k+1}\|^2 \leq f(x^0) - f^*, \forall k \in \mathbb{N}. \quad (7.4)$$

Tomando o limite como $N \rightarrow +\infty$ em (7.4) tem-se

$$\sum_{k=0}^{+\infty} \|x^k - x^{k+1}\|^2 < +\infty.$$

Em particular, $\|x^k - x^{k+1}\| \rightarrow 0$ quando $k \rightarrow +\infty$.

7.4 Demonstração do Teorema 4.3

Seja x^* um ponto crítico de $\{x^k\}_{k \in \mathbb{N}}$. Então existe uma subsequência $\{x^{k_j}\}_{j \in \mathbb{N}}$, tal que $\lim_{j \rightarrow \infty} x^{k_j} = x^*$. Em particular, $\{x^{k_j}\}_{j \in \mathbb{N}}$ é limitado.

Afirma-se que $\lim_{j \rightarrow \infty} x^{k_j+1} = x^*$. Na verdade, pela desigualdade triangular, tem-se

$$\begin{aligned} \|x^{k_j+1} - x^*\| &= \|(x^{k_j+1} - x^{k_j}) + (x^{k_j} - x^*)\| \\ &\leq \|x^{k_j+1} - x^{k_j}\| + \|x^{k_j} - x^*\|. \end{aligned}$$

Então,

$$\|x^{k_j+1} - x^*\| \leq \|x^{k_j+1} - x^{k_j}\| + \|x^{k_j} - x^*\|, \forall j \in \mathbb{N}. \quad (7.5)$$

Tomando o limite com $j \rightarrow \infty$ em (7.5) e aplicando a Proposição 4.2, tem-se que $\lim_{j \rightarrow +\infty} \|x^{k_j+1} - x^*\| = 0$, por isso $\lim_{j \rightarrow +\infty} x^{k_j+1} = x^*$.

Pela limitação de $\{x^{k_j}\}_{j \in \mathbb{N}}$ e aplicando a Proposição 3.4, obtém-se a limitação de $\{w^{k_j}\}_{j \in \mathbb{N}}$, onde $w^{k_j} \in \partial h(x^{k_j})$, para todo $j \in \mathbb{N}$.

A condição de otimalidade de primeira ordem para (4.11) em $k = k_j$ implica

$$w^{k_j} \in \partial g(x^{k_j+1}).$$

Sem perda de generalidade, supõe-se $\{w^{k_j}\}_{j \in \mathbb{N}}$ convergente. Portanto,

$$w^{k_j} \in \partial g(x^{k_j+1}) \cap \partial h(x^{k_j}), \forall j \in \mathbb{N}.$$

Pela Proposição 3.4, tem-se que

$$\partial g(x^*) \cap \partial h(x^*) \neq \emptyset.$$

Portanto, x^* é um ponto crítico.

7.5 Demonstração da Proposição 4.3

(i) Como y^k é a única solução de (4.12), tem-se que $\nabla g(y^k) - w^k = 0$, ou seja, $\nabla g(y^k) = w^k$. Pela forte convexidade de g e h , como $w^k \in \partial h(x^k)$ tem-se

$$g(x^k) - g(y^k) \geq \langle w^k, x^k - y^k \rangle + \frac{\rho}{2} \|x^k - y^k\|^2$$

e

$$h(y^k) - h(x^k) \geq \langle w^k, y^k - x^k \rangle + \frac{\rho}{2} \|y^k - x^k\|^2.$$

Somando ambas as desigualdades acima termo a termo, tem-se

$$\begin{aligned} [g(x^k) - h(x^k)] - [g(y^k) - h(y^k)] &\geq \langle w^k, x^k - y^k \rangle + \langle w^k, y^k - x^k \rangle + \rho \|x^k - x^{k+1}\|^2 \\ &= \langle w^k, x^k - y^k \rangle - \langle w^k, x^k - y^k \rangle + \rho \|x^k - y^k\|^2 \\ &= \rho \|x^k - y^k\|^2. \end{aligned}$$

Sabe-se que $d^k = y^k - x^k$ e $f(x) = g(x) - h(x)$. Assim,

$$f(y^k) \leq f(x^k) - \rho \|d^k\|^2.$$

(ii) Pela definição de derivada direcional, tem-se

$$\begin{aligned} f'(y^k; d^k) &= \lim_{\lambda \downarrow 0} \frac{f(y^k + \lambda d^k) - f(y^k)}{\lambda} \\ &= \lim_{\lambda \downarrow 0} \frac{g(y^k + \lambda d^k) - h(y^k + \lambda d^k) - (g(y^k) - h(y^k))}{\lambda} \\ &= \lim_{\lambda \downarrow 0} \frac{g(y^k + \lambda d^k) - g(y^k)}{\lambda} - \lim_{\lambda \downarrow 0} \frac{h(y^k + \lambda d^k) - h(y^k)}{\lambda}. \end{aligned}$$

Como g é diferenciável, obtém-se

$$f'(y^k; d^k) = \langle \nabla g(y^k), d^k \rangle - \lim_{\lambda \downarrow 0} \frac{h(y^k + \lambda d^k) - h(y^k)}{\lambda}. \quad (7.6)$$

Por outro lado, pela convexidade de h podemos escolher $u \in \partial h(y^k)$, e então

$$h(y^k + \lambda d^k) - h(y^k) \geq \langle u, \lambda d^k \rangle,$$

que implica

$$\frac{h(y^k + \lambda d^k) - h(y^k)}{\lambda} \geq \langle u, d^k \rangle, \quad \forall \lambda > 0,$$

e portanto,

$$\lim_{\lambda \downarrow 0} \frac{h(y^k + \lambda d^k) - h(y^k)}{\lambda} \geq \langle u, d^k \rangle.$$

Aplicando este fato em (7.6), obtém-se

$$\begin{aligned} f'(y^k; d^k) &\leq \langle \nabla g(y^k), d^k \rangle - \langle u, d^k \rangle \\ &= \langle \nabla g(y^k) - u, d^k \rangle \\ &= \langle \nabla g(y^k) - u, y^k - x^k \rangle. \end{aligned}$$

Portanto

$$f'(y^k; d^k) \leq \langle \nabla g(y^k) - u, y^k - x^k \rangle. \quad (7.7)$$

Sabe-se que $\nabla g(y^k) = w^k \in \partial h(x^k)$. A função h é fortemente convexa com módulo ρ , então, pelo Teorema 3.7, ∂h é fortemente monótona com módulo ρ . Assim, como $u \in \partial h(y^k)$, afirma-se que

$$\langle \nabla g(y^k) - u, x^k - y^k \rangle \geq \rho \|x^k - y^k\|^2.$$

Portanto,

$$\langle \nabla g(y^k) - u, y^k - x^k \rangle \leq -\rho \|x^k - y^k\|^2.$$

A prova segue combinando a última desigualdade com (7.7).

(iii) Se $d^k = 0$ ou $\lambda = 0$, não há nada a provar. Caso contrário, tem-se

$$\lim_{\lambda \downarrow 0} \frac{f(y^k + \lambda d^k) - f(y^k)}{\lambda} = f'(y^k; d^k) \leq -\rho \|x^k - y^k\|^2 < -\frac{\rho}{2} \|d^k\|^2.$$

Portanto, existe algum $\bar{\delta}_k > 0$ tal que

$$\frac{f(y^k + \lambda d^k) - f(y^k)}{\lambda} < -\frac{\rho}{2} \|d^k\|^2, \quad \forall \lambda \in (0, \bar{\delta}_k],$$

isto é,

$$f(y^k + \lambda d^k) < f(y^k) - \frac{\rho\lambda}{2} \|d^k\|^2, \quad \forall \lambda \in (0, \bar{\delta}_k].$$

Escolha $\lambda \in (0, \bar{\delta}_k]$. Então

$$f(y^k) - \frac{\rho\lambda}{2} \|d^k\|^2 \leq f(y^k + \lambda d^k) - \alpha\lambda^2 \|d^k\|^2,$$

que equivale a

$$\lambda \leq \frac{\rho}{2\alpha}.$$

Portanto, definindo $\delta_k := \min \left\{ \bar{\delta}_k, \frac{\rho}{2\alpha} \right\}$, obtém-se

$$f(y^k + \lambda d^k) < f(y^k) - \alpha\lambda^2 \|d^k\|^2, \quad \forall \lambda \in (0, \delta_k].$$

7.6 Demonstração do Lema 4.2

Na verdade, a expressão em (4.12) é equivalente a

$$\nabla g(y^k) = w^k.$$

Por outro lado, $w^k \in \partial h(x^k)$, devido ao passo 2. Consequentemente, se $y^k = x^k$, segue-se que $\partial g(x^k) \cap \partial h(x^k) = \{\nabla g(x^k)\}$. Assim, x^k é um ponto crítico de f .

7.7 Demonstração da Proposição 4.4

Pelo passo 3, o Algoritmo 8 termina quando $y^k = x^k$. Pelo Lema 4.2, x^k é um ponto crítico, e isso prova o item (i). Combinando os itens (i) e (iii) da Proposição 4.3 com o Passo 4, obtém-se

$$\begin{aligned} f(y^k + \lambda_k d^k) &\leq f(y^k) - \alpha \lambda_k^2 \|d^k\| \\ &\leq f(x^k) - \rho \|d^k\|^2 - \alpha \lambda_k^2 \|d^k\| \\ &= f(x^k) - (\rho + \alpha \lambda_k^2) \|d^k\|^2, \quad \forall k \in \mathbb{N}, \end{aligned}$$

onde $\rho > 0$ é a constante de forte convexidade para as funções g e h . Assim, como $d^k \neq 0$ e $x^{k+1} = y^k + \lambda_k d^k$, tem-se

$$f(x^{k+1}) = f(y^k + \lambda_k d^k) < f(x^k), \quad \forall k \in \mathbb{N},$$

que completa a prova.

7.8 Demonstração do Corolário 4.1

Isso decorre imediatamente do fato de que f é limitado inferiormente, pela hipótese **(H3)**, e $\{f(x^k)\}_{k \in \mathbb{N}}$ é decrescente.

7.9 Demonstração da Proposição 4.5

(i) Pela Proposição 4.3 e passo 4 do Algoritmo 8, tem-se

$$f(x^{k+1}) \leq f(x^k) - (\alpha \lambda_k^2 + \rho) \|d^k\|^2, \quad \forall k \in \mathbb{N}.$$

Então,

$$(\alpha \lambda_k^2 + \rho) \|d^k\|^2 \leq f(x^k) - f(x^{k+1}), \quad \forall k \in \mathbb{N}.$$

Como $\alpha \lambda_k^2 \geq 0$, afirma-se que $\rho \|d^k\|^2 \leq (\alpha \lambda_k^2 + \rho) \|d^k\|^2$. Por isso,

$$\rho \|d^k\|^2 \leq f(x^k) - f(x^{k+1}), \quad \forall k \in \mathbb{N}. \quad (7.8)$$

Considerando a soma parcial na desigualdade (7.8):

$$0 \leq \sum_{k=0}^N \rho \|d^k\|^2 \leq \sum_{k=0}^N [f(x^k) - f(x^{k+1})]$$

obtem-se

$$0 \leq \sum_{k=0}^N \rho \|d^k\|^2 \leq f(x^0) - f(x^{N+1}).$$

Como $f^* = \inf_{x \in \mathbb{R}^n} f(x)$, pela hipótese **(H2)**, então $f^* \leq f(x^{N+1})$ implica $-f^* \geq -f(x^{N+1})$, para todo $N \in \mathbb{N}$. Portanto,

$$0 \leq \sum_{k=0}^N \rho \|d^k\|^2 \leq f(x^0) - f^*, \forall k \in \mathbb{N}. \quad (7.9)$$

Como $\rho > 0$ é constante, tomando o limite com $N \rightarrow +\infty$ em (7.9), tem-se

$$\sum_{k=0}^{+\infty} \|d^k\|^2 < +\infty.$$

(ii) Seja $\bar{\lambda} > 0$ tal que $0 \leq \lambda_k \leq \bar{\lambda}$, para todo $k \in \mathbb{N}$. Em primeiro lugar, nota-se que

$$\begin{aligned} \|x^{k+1} - x^k\|^2 &= \|y^k + \lambda_k d^k - x^k\|^2 \\ &= \|y^k + \lambda_k (y^k - x^k) - x^k\|^2 \\ &= \|(1 + \lambda_k) y^k - (1 + \lambda_k) x^k\|^2 \\ &= (1 + \lambda_k)^2 \|y^k - x^k\|^2 \\ &= (1 + \lambda_k)^2 \|d^k\|^2 \\ &\leq (1 + \bar{\lambda})^2 \|d^k\|^2, \quad \forall k \in \mathbb{N}. \end{aligned}$$

Por causa disso, e usando o item (i), afirma-se que

$$0 \leq \sum_{k=0}^{+\infty} \|x^{k+1} - x^k\|^2 \leq \sum_{k=0}^{+\infty} (1 + \bar{\lambda})^2 \|d^k\|^2 < +\infty.$$

Em particular, $\|x^k - x^{k+1}\| \rightarrow 0$ quando $k \rightarrow +\infty$.

7.10 Demonstração do Teorema 4.4

Como x^* é um ponto crítico de $\{x^k\}_{k \in \mathbb{N}}$, há uma subsequência $\{x^{k_j}\}_{j \in \mathbb{N}}$ tal que $\lim_{j \rightarrow +\infty} x^{k_j} = x^*$, ou seja, $\lim_{j \rightarrow +\infty} \|x^{k_j} - x^*\| = 0$. Pela última proposição, se $\{y^k\}_{k \in \mathbb{N}}$ é a seqüência gerada pelo passo 2 do Algoritmo 8, então $\lim_{k \rightarrow +\infty} \|y^k - x^k\| = 0$. Por isso,

$$\begin{aligned} \|y^{k_j} - x^*\| &= \|y^{k_j} - x^{k_j} + x^{k_j} - x^*\| \\ &\leq \|y^{k_j} - x^{k_j}\| + \|x^{k_j} - x^*\|, \quad \forall j \in \mathbb{N}, \end{aligned}$$

que implica que $\lim_{j \rightarrow +\infty} \|y^{k_j} - x^*\| = 0$, isto é, $\lim_{j \rightarrow +\infty} y^{k_j} = x^*$. O passo 2 fornece $\nabla g(y^{k_j}) \in \partial h(x^{k_j})$, para todo $j \in \mathbb{N}$. Pela Proposição 3.4 com o Corolário 3.1, segue-se que

$$\nabla g(x^*) \in \partial h(x^*).$$

Assim, x^* é um ponto crítico de f .

7.11 Demonstração do Teorema 4.5

Seja \bar{x} um ponto de acumulação de $(x^k)_{k \in \mathbb{N}}$, e $(x^{k_\ell})_{\ell \in \mathbb{N}}$ uma subsequência de $(x^k)_{k \in \mathbb{N}}$ tal que $\lim_{\ell \rightarrow \infty} x^{k_\ell} = \bar{x}$. Seja $(w^{k_\ell})_{\ell \in \mathbb{N}}$ e $(y^{k_\ell})_{\ell \in \mathbb{N}}$ as sequências correspondentes geradas pelo Algoritmo 9, isto é, $w^{k_\ell} \in \partial h(x^{k_\ell})$. De (4.13), tem-se que $w^{k_\ell} \in \partial g(y^{k_\ell})$. Como $\lim_{k \rightarrow \infty} \|d^k\| = 0$ e $\lim_{\ell \rightarrow \infty} x^{k_\ell} = \bar{x}$, obtém-se que $\lim_{\ell \rightarrow \infty} y^{k_\ell} = \bar{x}$. Considerando que $w^{k_\ell} \in \partial h(x^{k_\ell}) \cap \partial g(y^{k_\ell})$ e devido à convexidade de g e h , sem perda de generalidade, podemos aplicar a Proposição 3.4 para obter que $\lim_{\ell \rightarrow \infty} w^{k_\ell} = \bar{w} \in \partial g(\bar{x}) \cap \partial h(\bar{x})$, que conclui a prova.

7.12 Demonstração da Proposição 4.6

Antes de iniciar a prova, sabe-se que $d^k = y^k - x^k$. Para provar o item (ii), assume-se que $d^k \neq 0$ e é tomado $w^k \in \partial h(x^k)$. Como h é fortemente convexo com módulo $\sigma > 0$, segue do Teorema 3.7, item 2, que

$$h(y^k + \lambda d^k) \geq h(y^k) + \lambda \langle s, d^k \rangle + \frac{\sigma}{2} \lambda^2 \|d^k\|^2, \quad \forall s \in \partial h(y^k). \quad (7.10)$$

Além disso, levando em consideração que $w^k \in \partial h(x^k)$, pode-se aplicar o Teorema 3.7, item 3, para obter que $\langle s, d^k \rangle \geq \langle w^k, d^k \rangle + \sigma \|d^k\|^2$. Portanto, (7.10) se torna

$$h(y^k + \lambda d^k) \geq h(y^k) + \lambda \langle w^k, d^k \rangle + \sigma \lambda \|d^k\|^2 + \frac{\sigma}{2} \lambda^2 \|d^k\|^2. \quad (7.11)$$

Considerando que y^k é a solução para (9), tem-se que $g(y^k) - \langle w^k, d^k \rangle \leq g(x^k)$, que, combinado com (7.11), produz

$$-(h(y^k + \lambda d^k) - h(y^k)) \leq \lambda (g(x^k) - g(y^k)) - \sigma \lambda \|d^k\|^2 - \frac{\sigma}{2} \lambda^2 \|d^k\|^2. \quad (7.12)$$

Por outro lado, usando a forte convexidade de g com módulo $\sigma > 0$, tem-se

$$\begin{aligned} g(y^k + \lambda d^k) - g(y^k) &= g(\lambda(y^k + d^k) + (1 - \lambda)y^k) - g(y^k) \\ &\leq \lambda g(y^k + d^k) + (1 - \lambda)g(y^k) - \frac{\sigma}{2} \lambda(1 - \lambda) \|d^k\|^2 - g(y^k) \\ &= \lambda (g(y^k + d^k) - g(y^k)) - \frac{\sigma}{2} \lambda(1 - \lambda) \|d^k\|^2, \end{aligned} \quad (7.13)$$

para todo $\lambda \in (0,1]$. Combinando a definição de ϕ em (4.10), com (7.12) e (7.13), obtém-se

$$\begin{aligned} f(y^k + \lambda d^k) - f(y^k) &= g(y^k + \lambda d^k) - g(y^k) - (h(y^k + \lambda d^k) - h(y^k)) \\ &\leq -\frac{3\sigma}{2}\lambda\|d^k\|^2 + \lambda(g(y^k + d^k) + g(x^k) - 2g(y^k)). \end{aligned} \quad (7.14)$$

Além disso, segue do Teorema 3.7, item 2, que

$$g(y^k + d^k) \geq g(y^k) + \langle w, d^k \rangle + \frac{\sigma}{2}\|d^k\|^2, \quad g(x^k) \geq g(y^k) - \langle w, d^k \rangle + \frac{\sigma}{2}\|d^k\|^2,$$

para todo $w \in \partial g(y^k)$, que implica que $g(y^k + d^k) + g(x^k) - 2g(y^k) \geq \sigma\|d^k\|^2 > 0$. Assim, devido a $\nu_k > 0$, tem-se $0 < \hat{\delta}_k := \nu_k / (g(y^k + d^k) + g(x^k) - 2g(y^k))$, que prova a primeira da Proposição 4.6, item (i). Além disso, tem-se

$$0 < \lambda(g(y^k + d^k) + g(x^k) - 2g(y^k)) \leq \nu_k, \quad \lambda \in (0, \hat{\delta}_k],$$

com $\bar{\delta}_k := \min\{\hat{\delta}_k, 1, (3\sigma)/(2\rho)\}$. Portanto, a última desigualdade junto com (7.14) implica

$$f(y^k + \lambda d^k) - f(y^k) \leq -\rho\lambda^2\|d^k\|^2 + \nu_k, \quad \forall \lambda \in (0, \bar{\delta}_k],$$

que conclui a segunda afirmação do item (i) da Proposição 4.6. Finalmente, considerando que $\lim_{j \rightarrow \infty} \zeta^j \bar{\lambda}_k = 0$, segue da última desigualdade que a linha busca no Passo 3 está bem definida e a prova do item (i) está concluída. Para provar o item (ii), nota-se que a Proposição 4.6, item (i), implica que o Passo 4 está bem definido para $\nu_k > 0$. Assim, como $\lim_{j \rightarrow \infty} \zeta^j \bar{\lambda}_k = 0$, existe $j \in \mathbb{N}$ grande o suficiente, tal que $\lambda_k = \zeta^j \bar{\lambda}_k$ satisfaz $f(y^k + \lambda_k d^k) \leq f(y^k) - \rho\lambda_k^2\|d^k\|^2 + \nu_k$, que combinado com Ferreira, Santos e Souza (2024, Proposição 12, item (ii)) implica na Proposição 4.6, item (ii), e a prova da proposição está completa.

7.13 Demonstração do Teorema 4.6

Pela definição da estratégia (E3), existe $k_0 \in \mathbb{N}$ tal que $0 \leq \nu_k \leq \sigma\|d^k\|^2/2$, para todo $k \geq k_0$. Então, $\sigma\|d^k\|^2/2 \leq \sigma\|d^k\|^2 - \nu_k$, para todo $k \geq k_0$. Portanto, usando a Proposição 4.6, item (ii), tem-se $0 \leq \sigma\|d^k\|^2/2 \leq \phi(x^k) - \phi(x^{k+1})$, para todo $k \geq k_0$. Usando $f^* := \inf_{x \in \mathbb{R}^n} \{f(x) = g(x) - h(x)\} > -\infty$, conclui-se que $(\phi(x^k))_{k \geq k_0}$ é convergente. Além disso, segue-se que $\lim_{k \rightarrow \infty} \|d^k\| = 0$. Portanto, aplicando o Teorema 4.5, obtém-se o resultado desejado.

7.14 Demonstração do Teorema 4.7

Seja $\varsigma \in (0,1)$ e $k_0 \in \mathbb{N}$ tal que $\nu_k \leq \varsigma\sigma\|d^k\|^2$, para todo $k \geq k_0$. Segue da Proposição 4.6, item (ii), que $\sigma\|d^k\|^2 \leq f(x^k) - f(x^{k+1}) + \nu_k$, para todo $k = 0, 1, \dots, N-1$.

Somando na última desigualdade com $k = 0$ até $k = N - 1$, e assumindo $f^* := \inf_{x \in \mathbb{R}^n} \{f(x) = g(x) - h(x)\} > -\infty$, tem-se

$$\sigma \sum_{k=0}^{N-1} \|d^k\|^2 \leq f(x^0) - f^* + \sum_{k=0}^{k_0-1} \nu_k + \sum_{k=k_0}^{N-1} \nu_k.$$

Portanto, considerando que $\nu_k \leq \varsigma \sigma \|d^k\|^2$, para todo $k \geq k_0$, a última desigualdade se torna

$$\sum_{k=0}^{N-1} \sigma \|d^k\|^2 \leq f(x^0) - f^* + \sum_{k=0}^{k_0-1} \nu_k + \sum_{k=0}^{N-1} \varsigma \sigma \|d^k\|^2,$$

que equivale a $\sum_{k=0}^{N-1} (1-\varsigma) \sigma \|d^k\|^2 \leq f(x^0) - f^* + \sum_{k=0}^{k_0-1} \nu_k$. Assim, tem-se $N \min\{\|d^k\|^2 : k = 0, 1, \dots, N-1\} \leq (f(x^0) - f^* + \sum_{k=0}^{k_0-1} \nu_k) / ((1-\varsigma)\sigma)$, provando a primeira desigualdade. A outra desigualdade segue de modo análogo à primeira.

7.15 Demonstração da Proposição 5.1

Como ϕ é côncava, tem-se

$$\phi'_+(a) \geq \phi'_+(b), \text{ para todo } 0 \leq a \leq b. \quad (7.15)$$

Considerando $f(x) = g(x) - h(x)$, onde

$$g(x) = \frac{\mu + \rho}{2} \|x - b\|^2 + \tau TV(x) \text{ and } h(x) = \frac{\rho}{2} \|x - b\|^2 + \tau TV(x) - TV_\phi(x),$$

para um dado parâmetro $\rho > 0$ e $\tau \geq \phi'_+(0) \geq 0$. Tem-se que mostrar que g e h são funções fortemente convexas. De fato, g é fortemente convexa porque é a soma entre a função convexa $g_1(x) = \tau TV(x)$ e a função fortemente convexa $g_2(x) = \frac{\mu + \rho}{2} \|x - b\|^2$; veja a Proposição 3.5. Note que para mostrar a convexidade forte de h , usando argumento similar à função g , uma vez que a norma quadrada é fortemente convexa, basta provar que $\tau TV(x) - TV_\phi(x)$ é convexa. Segue-se da definição da função TV_ϕ em (5.2) que $\tau TV(x) - TV_\phi(x) = \psi(TV(x))$, onde $\psi(x) = \tau x - \phi(x)$. Como $\psi''(x) = -\phi''(x)$ e ϕ são côncavos, temos que $\psi(x)$ é convexo. Além disso, $\psi'_+(x) = \tau - \phi'_+(x) \geq \phi'_+(0) - \phi'_+(x) \geq 0$, para todo $x \geq 0$, onde a primeira desigualdade vem da suposição $\tau \geq \phi'_+(0)$ e a segunda desigualdade segue de (7.15). Assim, tem-se que $\psi(x)$ é não decrescente, em $x \geq 0$, e uma função convexa. Como $TV(x)$ é uma função convexa, conclui-se da Proposição 3.6 que $\tau TV(x) - TV_\phi(x) = \psi(TV(x))$ é uma função convexa e a prova está concluída.

7.16 Demonstração da Proposição 5.2

Como y^k é uma solução de (4.12), combinando as Proposições 5.1 e 3.8, tem-se que

$$0 \in \partial g(y^k) - w^k, \quad (7.16)$$

onde $w^k \in \partial h(x^k)$ com g e h como em (5.4). Usando $g(x) = \frac{\mu + \rho}{2} \|x - b\|^2 + \tau TV(x)$ em (7.16), obtém-se

$$0 \in (\mu + \rho)(y^k - b) + \tau \partial TV(y^k) - w^k,$$

que é equivalente para

$$0 \in (\mu + \rho) \left[y^k - \left(\frac{w^k}{\mu + \rho} + b \right) \right] + \tau \partial TV(y^k).$$

Mais uma vez, da Proposição 3.8, conclui-se que

$$y^k \in \arg \min \left\{ \frac{\mu + \rho}{2} \|x - s^k\|^2 + \tau TV(x) : x \in \mathbb{R}^n \right\},$$

onde $s^k = \frac{w^k}{\mu + \rho} + b$ e a prova está concluída.

Referências

- AHARON, M.; ELAD, M.; BRUCKSTEIN, A. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. **IEEE Transactions on signal processing**, IEEE, v. 54, n. 11, p. 4311–4322, 2006.
- AKBAR, J. M. et al. Joint method using akamatsu and discrete wavelet transform for image restoration. **Applied computing and informatics**, Emerald Publishing Limited, 2020.
- AN, L. T. H.; TAO, P. D. The dc (difference of convex functions) programming and dca revisited with dc models of real world nonconvex optimization problems. **Annals of operations research**, Springer, v. 133, n. 1, p. 23–46, 2005.
- ANDRADE, T. G. d. **Melhoramento de voz baseado em representações esparsas usando dicionários treinados**. Dissertação (Mestrado), 2020. Disponível em <<http://hdl.handle.net/11449/202193>>.
- ARMIJO, L. Minimization of functions having lipschitz continuous first partial derivatives. **Pacific Journal of mathematics**, Mathematical Sciences Publishers, v. 16, n. 1, p. 1–3, 1966.
- ARTACHO, F. J. A.; FLEMING, R. M.; VUONG, P. T. Accelerating the dc algorithm for smooth functions. **Mathematical Programming**, Springer, v. 169, n. 1, p. 95–118, 2018.
- ARTACHO, F. J. A.; VUONG, P. T. The boosted difference of convex functions algorithm for nonsmooth functions. **SIAM Journal on Optimization**, SIAM, v. 30, n. 1, p. 980–1006, 2020.
- BAČÁK, M.; BORWEIN, J. M. On difference convexity of locally lipschitz functions. **Optimization**, Taylor & Francis, v. 60, n. 8-9, p. 961–978, 2011.
- BAI, L. A new nonconvex approach for image restoration with gamma noise. **Computers & Mathematics with Applications**, Elsevier, v. 77, n. 10, p. 2627–2639, 2019.
- BAO, L.; LIU, W.; ZHU, Y.; PU, Z.; MAGNIN, I. E. Sparse representation based mri denoising with total variation. In: IEEE. **2008 9th International Conference on Signal Processing**. [S.l.], 2008. p. 2154–2157.
- BATISTA, P. V. do C. **Representação esparsa utilizando Wavelets e variação total aplicados ao processamento de sinais de descargas parciais**. [S.l.]: Universidade Federal de Minas Gerais, 2018.
- BECK, A. **First-order methods in optimization**. [S.l.]: SIAM, 2017.
- BECK, A.; TEOULLE, M. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. **IEEE transactions on image processing**, IEEE, v. 18, n. 11, p. 2419–2434, 2009.

- BECK, A.; TEBOULLE, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. **SIAM journal on imaging sciences**, SIAM, v. 2, n. 1, p. 183–202, 2009.
- BECK, A.; TEBOULLE, M. A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring. In: **2009 IEEE International Conference on Acoustics, Speech and Signal Processing**. [S.l.: s.n.], 2009. p. 693–696.
- BEN-TAL, A.; GHAOUI, L. E.; NEMIROVSKI, A. **Robust optimization**. [S.l.]: Princeton university press, 2009.
- BEN-TAL, A.; NEMIROVSKI, A. Robust convex optimization. **Mathematics of operations research**, INFORMS, v. 23, n. 4, p. 769–805, 1998.
- BOWMAN, F. **Introduction to Bessel functions**. [S.l.]: Courier Corporation, 2012.
- BROOKS, T.; MILDENHALL, B.; XUE, T.; CHEN, J.; SHARLET, D.; BARRON, J. T. Unprocessing images for learned raw denoising. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 11036–11045.
- BRUCK, R. E. On the weak convergence of an ergodic iteration for the solution of variational inequalities for monotone operators in hilbert space. **Journal of Mathematical Analysis and Applications**, Academic Press, v. 61, n. 1, p. 159–164, 1977.
- CAI, J.-F.; CHAN, R. H.; NIKOLOVA, M. Fast two-phase image deblurring under impulse noise. **Journal of Mathematical Imaging and Vision**, Springer, v. 36, n. 1, p. 46–53, 2010.
- CAMPAGNA, R.; CRISCI, S.; CUOMO, S.; MARCELLINO, L.; TORALDO, G. Modification of tv-rof denoising model based on split bregman iterations. **Applied Mathematics and Computation**, Elsevier, v. 315, p. 453–467, 2017.
- CANDÈS, E. J.; ROMBERG, J.; TAO, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. **IEEE Transactions on information theory**, IEEE, v. 52, n. 2, p. 489–509, 2006.
- CHAMBOLLE, A. An algorithm for total variation minimization and applications. **Journal of Mathematical imaging and vision**, Springer, v. 20, n. 1, p. 89–97, 2004.
- CHAN, R. H.; DONG, Y.; HINTERMÜLLER, M. An efficient two-phase l_1 -tv method for restoring blurred images with impulse noise. **IEEE Transactions on Image Processing**, IEEE, v. 19, n. 7, p. 1731–1739, 2010.
- CHAN, T. F.; ESEDOGLU, S. Aspects of total variation regularized l_1 function approximation. **SIAM Journal on Applied Mathematics**, SIAM, v. 65, n. 5, p. 1817–1837, 2005.
- CHEN, S.; DONOHO, D. Basis pursuit. In: IEEE. **Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers**. [S.l.], 1994. v. 1, p. 41–44.

- CHEN, W.; YOU, J.; CHEN, B.; PAN, B.; LI, L.; POMEROY, M.; LIANG, Z. A sparse representation and dictionary learning based algorithm for image restoration in the presence of rician noise. **Neurocomputing**, Elsevier, v. 286, p. 130–140, 2018.
- COMBETTES, P. L.; WAJS, V. R. Signal recovery by proximal forward-backward splitting. **Multiscale modeling & simulation**, SIAM, v. 4, n. 4, p. 1168–1200, 2005.
- DAUBECHIES, I.; DEFRISE, M.; MOL, C. D. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. **Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences**, Wiley Online Library, v. 57, n. 11, p. 1413–1457, 2004.
- DOBSON, D. C.; SANTOSA, F. Recovery of blocky images from noisy and blurred data. **SIAM Journal on Applied Mathematics**, SIAM, v. 56, n. 4, p. 1181–1198, 1996.
- DONG, B.; JI, H.; LI, J.; SHEN, Z.; XU, Y. Wavelet frame based blind image inpainting. **Applied and Computational Harmonic Analysis**, Elsevier, v. 32, n. 2, p. 268–279, 2012.
- DONOHO, D. L. Compressed sensing. **IEEE Transactions on information theory**, IEEE, v. 52, n. 4, p. 1289–1306, 2006.
- ELAD, M.; FIGUEIREDO, M. A.; MA, Y. On the role of sparse and redundant representations in image processing. **Proceedings of the IEEE**, IEEE, v. 98, n. 6, p. 972–982, 2010.
- ELAIYARAJA, G.; KUMARATHARAN, N.; RAO, T. C. S. Fast and efficient filter using wavelet threshold for removal of gaussian noise from mri/ct scanned medical images/color video sequence. **IETE Journal of research**, Taylor & Francis, v. 68, n. 1, p. 10–22, 2022.
- ENGL, H. W.; HANKE, M.; NEUBAUER, A. **Regularization of inverse problems**. [S.l.]: Springer Science & Business Media, 1996. v. 375.
- FAN, L.; ZHANG, F.; FAN, H.; ZHANG, C. Brief review of image denoising techniques. **Visual Computing for Industry, Biomedicine, and Art**, Springer, v. 2, n. 1, p. 1–12, 2019.
- FANG, J.; LIU, H.; LIU, H.; ZHANG, L.; LIU, J.; ZHANG, H.; LIU, C. Multiresolution convex variational model for multiphase image segmentation. **Computers & Electrical Engineering**, Elsevier, v. 54, p. 230–245, 2016.
- FANG, L.; XIANGHAI, W. Adaptive total-variation and nonconvex low-rank model for image denoising. **International Journal of Image and Graphics**, World Scientific, p. 2550016, 2023.
- FERNÁNDEZ-MARTÍNEZ, J. L.; PALLERO, J.; FERNÁNDEZ-MUÑIZ, Z.; PEDRUELO-GONZÁLEZ, L. M. The effect of noise and tikhonov's regularization in inverse problems. part i: The linear case. **Journal of Applied Geophysics**, Elsevier, v. 108, p. 176–185, 2014.
- FERREIRA, O. P.; RABELO, R. A. L.; RIBEIRO, P. H. A.; SANTOS, E. M.; SOUZA, J. C. O. Image denoising with a non-monotone boosted dca for non-convex models. **Computers and Electrical Engineering**, Elsevier, v. 117, p. 109306, 2024.

FERREIRA, O. P.; SANTOS, E. M.; SOUZA, J. C. O. A boosted dc algorithm for non-differentiable dc components with non-monotone line search. **Computational Optimization and Applications**, Springer, p. 1–36, 2024.

FIGUEIREDO, M. A.; NOWAK, R. D. An em algorithm for wavelet-based image restoration. **IEEE Transactions on Image Processing**, IEEE, v. 12, n. 8, p. 906–916, 2003.

FUKUSHIMA, M.; MINE, H. A generalized proximal point algorithm for certain non-convex minimization problems. **International Journal of Systems Science**, Taylor & Francis, v. 12, n. 8, p. 989–1000, 1981.

GINCHEV, I.; GINTCHEVA, D. Characterization and recognition of dc functions. **Journal of Global Optimization**, Springer, v. 57, n. 3, p. 633–647, 2013.

GUDBJARTSSON, H.; PATZ, S. The rician distribution of noisy mri data. **Magnetic resonance in medicine**, Wiley Online Library, v. 34, n. 6, p. 910–914, 1995.

HARTMAN, P. On functions representable as a difference of convex functions. **Pacific Journal of Mathematics**, Pacific Journal of Mathematics, A Non-profit Corporation, v. 9, n. 3, p. 707–713, 1959.

HIRIART-URRUTY, J.; LEMARECHAL, C. **Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods**. [S.l.]: Springer Berlin Heidelberg, 1993.

HORÉ, A.; ZIOU, D. Is there a relationship between peak-signal-to-noise ratio and structural similarity index measure? **IET Image Processing**, Wiley Online Library, v. 7, n. 1, p. 12–24, 2013.

HORÉ, A.; ZIOU, D. Image quality metrics: Psnr vs. ssim. In: IEEE. **2010 20th international conference on pattern recognition**. [S.l.], 2010. p. 2366–2369.

HUBER, P. J. Robust estimation of a location parameter. In: **Breakthroughs in statistics: Methodology and distribution**. [S.l.]: Springer, 1992. p. 492–518.

HÜTTER, J.-C.; RIGOLLET, P. Optimal rates for total variation denoising. In: PMLR. **Conference on Learning Theory**. [S.l.], 2016. p. 1115–1146.

IZMAILOV, A.; SOLODOV, M. **Otimização, volume 1: condições de otimalidade, elementos de análise convexa e de dualidade**. [S.l.]: Impa, 2005.

JIA, R.-Q.; ZHAO, H. A fast algorithm for the total variation model of image denoising. **Advances in Computational Mathematics**, Springer, v. 33, n. 2, p. 231–241, 2010.

JIDESH, P. A convex regularization model for image restoration. **Computers & Electrical Engineering**, Elsevier, v. 40, n. 8, p. 66–78, 2014.

LANZA, A.; MORIGI, S.; SGALLARI, F. Convex image denoising via non-convex regularization with parameter selection. **Journal of Mathematical Imaging and Vision**, Springer, v. 56, n. 2, p. 195–220, 2016.

LI, X.; HU, Y.; GAO, X.; TAO, D.; NING, B. A multi-frame image super-resolution method. **Signal Processing**, Elsevier, v. 90, n. 2, p. 405–414, 2010.

- LIANG, G.; ZHANG, Y.; JACOBS, N. Neural network calibration for medical imaging classification using dca regularization. **ICML UDL**, 2020.
- LIANG, H.; LI, N.; ZHAO, S. Salt and pepper noise removal method based on a detail-aware filter. **Symmetry**, MDPI, v. 13, n. 3, p. 515, 2021.
- LIMA, E. L. **Curso de análise vol. 2**. [S.l.]: Rio de Janeiro: IMPA, 2015.
- LOU, Y.; ZENG, T.; OSHER, S.; XIN, J. A weighted difference of anisotropic and isotropic total variation model for image processing. **SIAM Journal on Imaging Sciences**, SIAM, v. 8, n. 3, p. 1798–1823, 2015.
- MA, L.; YU, J.; ZENG, T. Sparse representation prior and total variation–based image deblurring under impulse noise. **SIAM Journal on Imaging Sciences**, SIAM, v. 6, n. 4, p. 2258–2284, 2013.
- MACOVSKI, A. Noise in mri. **Magnetic resonance in medicine**, Wiley Online Library, v. 36, n. 3, p. 494–497, 1996.
- MIN, X.; MA, K.; GU, K.; ZHAI, G.; WANG, Z.; LIN, W. Unified blind quality assessment of compressed natural, graphic, and screen content images. **IEEE Transactions on Image Processing**, IEEE, v. 26, n. 11, p. 5462–5474, 2017.
- MOHAN, J.; KRISHNAVENI, V.; GUO, Y. A survey on the magnetic resonance image denoising methods. **Biomedical signal processing and control**, Elsevier, v. 9, p. 56–69, 2014.
- MOULIN, P.; LIU, J. Analysis of multiresolution image denoising schemes using generalized-gaussian priors. In: IEEE. **Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis (Cat. No. 98TH8380)**. [S.l.], 1998. p. 633–636.
- MUNIZ, L. G. S. M. **Uma melhoria do algoritmo K-SVD com aplicações em reconhecimento facial**. [S.l.]: Universidade Federal do Ceará, 2014. Disponível em <<http://repositorio.ufc.br/handle/riufc/44703>>.
- NESTEROV, Y. E. A method for solving the convex programming problem with convergence rate $o\left(\frac{1}{k^2}\right)$. In: **Dokl. Akad. Nauk SSSR**, [S.l.: s.n.], 1983. v. 269, p. 543–547.
- NETO, J. X. C.; OLIVEIRA, P. R.; SOUBEYRAN, A.; SOUZA, J. C. O. A generalized proximal linearized algorithm for dc functions with application to the optimal size of the firm problem. **Annals of Operations Research**, Springer, v. 289, p. 313–339, 2020.
- NGUYEN, T. B. T. **DC programming and DCA in image processing: compressed sensing, segmentation and restoration**. Tese (Doutorado) — Université de Lorraine, 2014.
- NIKOLOVA, M. A variational approach to remove outliers and impulse noise. **Journal of Mathematical Imaging and Vision**, Springer, v. 20, n. 1, p. 99–120, 2004.
- NIKOLOVA, M.; NG, M. K.; TAM, C.-P. Fast nonconvex nonsmooth minimization methods for image restoration and reconstruction. **IEEE Transactions on Image Processing**, IEEE, v. 19, n. 12, p. 3073–3088, 2010.

- NIU, Y.-S. On the convergence analysis of dca. **arXiv preprint arXiv:2211.10942**, 2022.
- OLIVEIRA, W. D.; TCHEOU, M. P. An inertial algorithm for dc programming. **Set-Valued and Variational Analysis**, Springer, v. 27, n. 4, p. 895–919, 2019.
- PASSTY, G. B. Ergodic convergence to a zero of the sum of monotone operators in hilbert space. **Journal of Mathematical Analysis and Applications**, Elsevier, v. 72, n. 2, p. 383–390, 1979.
- PERONA, P.; MALIK, J. Scale-space and edge detection using anisotropic diffusion. **IEEE Transactions on pattern analysis and machine intelligence**, IEEE, v. 12, n. 7, p. 629–639, 1990.
- PEYRÉ, G.; BOUGLEUX, S.; COHEN, L. Non-local regularization of inverse problems. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2008. p. 57–68.
- POLAK, E. **Computational methods in optimization: a unified approach**. [S.l.]: Academic press, 1971. v. 77.
- POLYAK, B. T. Introduction to optimization. **Optimization Software, Inc, New York**, 1987.
- RIBEIRO, D. A. **Representação esparsa para preenchimento de buracos de expansão em sínteses de vistas baseada em profundidade**. [S.l.]: Universidade de Brasília, 2017.
- RIBEIRO, P. H. A.; SOUZA, J. C. O. Image denoising with non-convex models: A comparison between bdca and nmbdca. In: **2024 37th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. [S.l.: s.n.], 2024. p. 1–6.
- ROCKAFELLAR, R. T.; WETS, R. J.-B. **Variational analysis**. [S.l.]: Springer Science & Business Media, 2009. v. 317.
- RUBINSTEIN, R.; BRUCKSTEIN, A. M.; ELAD, M. Dictionaries for sparse representation modeling. **Proceedings of the IEEE**, IEEE, v. 98, n. 6, p. 1045–1057, 2010.
- RUDIN, L. I.; OSHER, S.; FATEMI, E. Nonlinear total variation based noise removal algorithms. **Physica D: nonlinear phenomena**, Elsevier, v. 60, n. 1-4, p. 259–268, 1992.
- SETIADI, D. R. I. M. Psnr vs ssim: imperceptibility quality assessment for image steganography. **Multimedia Tools and Applications**, Springer, v. 80, n. 6, p. 8423–8444, 2021.
- SHI, B.; LI, M.; LOU, Y. Adaptively weighted difference model of anisotropic and isotropic total variation for image denoising. **Journal of Nonlinear & Variational Analysis**, v. 7, n. 4, 2023.
- SHUKLA, A. K.; PANDEY, R. K.; REDDY, P. Generalized fractional derivative based adaptive algorithm for image denoising. **Multimedia Tools and Applications**, Springer, v. 79, n. 19, p. 14201–14224, 2020.
- SIJBERS, J.; DEKKER, A. J. den; DYCK, D. V.; RAMAN, E. Estimation of signal and noise from rician distributed data. In: CITESEER. **Proceedings of the international conference on signal processing and communications**. [S.l.], 1998. p. 140–142.

SILVA, F. T. de A. Processamento de sinais de descargas parciais utilizando dicionários sobre-completos e representações esparsas. Universidade Federal de Minas Gerais, 2017.

STARCK, J.-L.; MURTAGH, F.; FADILI, J. M. **Sparse image and signal processing: wavelets, curvelets, morphological diversity**. [S.l.]: Cambridge university press, 2010.

SUN, K.; XU, Y.; FENG, M. Non-monotone boosted dc and caputo fractional tailored finite point algorithm for rician denoising and deblurring. **Journal of Mathematical Imaging and Vision**, Springer, p. 1–18, 2023.

TAO, P. D.; AN, L. H. Convex analysis approach to dc programming: theory, algorithms and applications. **Acta mathematica vietnamica**, v. 22, n. 1, p. 289–355, 1997.

TAO, P. D.; SOUAD, E. B. Algorithms for solving a class of nonconvex optimization problems. methods of subgradients. In: **North-Holland Mathematics Studies**. [S.l.]: Elsevier, 1986. v. 129, p. 249–271.

THANH, D. N.; PRASATH, V. S. et al. Total variation l1 fidelity salt-and-pepper denoising with adaptive regularization parameter. In: IEEE. **2018 5th NAFOSTED Conference on Information and Computer Science (NICS)**. [S.l.], 2018. p. 400–405.

THANH, D. N. H.; HIEN, N. N.; PRASATH, S. et al. Adaptive total variation l1 regularization for salt and pepper image denoising. **Optik**, Elsevier, v. 208, p. 163677, 2020.

THI, H. A. L.; DINH, T. P. Difference of convex functions algorithms (dca) for image restoration via a markov random field model. **Optimization and Engineering**, Springer, v. 18, n. 4, p. 873–906, 2017.

THI, H. A. L.; DINH, T. P. Dc programming and dca: thirty years of developments. **Mathematical Programming**, Springer, v. 169, n. 1, p. 5–68, 2018.

TIKHONOV, A. N. On the solution of ill-posed problems and the method of regularization. In: RUSSIAN ACADEMY OF SCIENCES. **Doklady Akademii Nauk**. [S.l.], 1963. v. 151, n. 3, p. 501–504.

WALTER, É. Introduction to optimization. In: **Numerical Methods and Optimization**. [S.l.]: Springer, 2014. p. 167–175.

WANG, Q.; ZHANG, X.; WU, Y.; TANG, L.; ZHA, Z. Nonconvex weighted l_p minimization based group sparse representation framework for image denoising. **IEEE Signal Processing Letters**, IEEE, v. 24, n. 11, p. 1686–1690, 2017.

WANG, Z.; BOVIK, A. C.; SHEIKH, H. R.; SIMONCELLI, E. P. Image quality assessment: from error visibility to structural similarity. **IEEE transactions on image processing**, IEEE, v. 13, n. 4, p. 600–612, 2004.

WEICKERT, J. et al. **Anisotropic diffusion in image processing**. [S.l.]: Teubner Stuttgart, 1998. v. 1.

- WU, T.; GU, X.; LI, Z.; LI, Z.; NIU, J.; ZENG, T. Efficient boosted dc algorithm for nonconvex image restoration with rician noise. **SIAM Journal on Imaging Sciences**, SIAM, v. 15, n. 2, p. 424–454, 2022.
- XU, S.; YANG, X.; JIANG, S. A fast nonlocally centralized sparse representation algorithm for image denoising. **Signal Processing**, Elsevier, v. 131, p. 99–112, 2017.
- YANG, G.; LU, Z.; YANG, J.; WANG, Y. An adaptive contourlet hmm–pcnn model of sparse representation for image denoising. **IEEE Access**, IEEE, v. 7, p. 88243–88253, 2019.
- YOU, Y.; NIU, Y.-S. A refined inertial dca for dc programming. **arXiv preprint arXiv:2104.14750**, 2021.
- YU, Y.; ZHANG, J. Hybrid model and split bregman iteration algorithm for image denoising. In: ATLANTIS PRESS. **2nd International Forum on Management, Education and Information Technology Application (IFMEITA 2017)**. [S.l.], 2018. p. 361–368.
- ZHANG, B.; ZHU, G.; ZHU, Z. A tv-log nonconvex approach for image deblurring with impulsive noise. **Signal Processing**, Elsevier, v. 174, p. 107631, 2020.
- ZHANG, B.; ZHU, G.; ZHU, Z.; ZHANG, H.; ZHOU, Y.; KWONG, S. Impulse noise image restoration using nonconvex variational model and difference of convex functions algorithm. **IEEE Transactions on Cybernetics**, IEEE, 2022.
- ZHANG, L.; XIONG, Z.; LAI, J.; LIU, J. Research of optical flow aided mems navigation based on convex optimization and rof denoising. **Optik**, Elsevier, v. 158, p. 1575–1583, 2018.
- ZHOU, Z. Sparse recovery based on the generalized error function. **arXiv preprint arXiv:2105.13189**, 2021.