
Método variacional com atualização múltipla de ganhos para controle de sistemas lineares com parâmetros sujeitos a saltos Markovianos não observados

Larissa Tebaldi de Oliveira

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 08/08/2014

Assinatura: _____

**Método variacional com atualização múltipla de ganhos
para controle de sistemas lineares com parâmetros
sujeitos a saltos Markovianos não observados**

Larissa Tebaldi de Oliveira

***Orientador:* Prof. Dr. Eduardo Fontoura Costa**

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

**USP – São Carlos
Agosto de 2014**

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

O48m Oliveira, Larissa Tebaldi de
Método variacional com atualização múltipla de ganhos para controle de sistemas lineares com parâmetros sujeitos a saltos Markovianos não observados / Larissa Tebaldi de Oliveira; orientador Eduardo Fontoura Costa. -- São Carlos, 2014.
112 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2014.

1. Sistemas lineares com saltos Markovianos. 2. Controle ótimo. 3. Problema biquadrático. I. Costa, Eduardo Fontoura, orient. II. Título.

À minha família.

Agradecimentos

Agradeço ao Prof. Dr. Eduardo Fontoura Costa pela oportunidade dada e por toda paciência, atenção, tempo e esforço empenhado durante todo o período deste trabalho.

Aos meus pais Maria Carolina e Cacildo, que sempre me incentivaram a estudar, por todo carinho e suporte dado ao longo da minha vida.

Ao Renato por tudo que passamos juntos nos últimos anos, pela sua amizade, seu carinho, sua força, sua calma e, principalmente, sua paciência.

À todos os colegas, professores e funcionários do Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES, à Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq pelos apoios financeiros e pelas bolsas de mestrado concedidas sob os processos DS-7899600/M e 2012/14085-5.

Neste trabalho foi estudado um problema de controle de sistemas lineares com saltos Markovianos sem observação da variável de salto, que pode ser escrito como um problema de otimização de considerável complexidade. As contribuições para a área estão divididas em três aspectos. Um dos avanços foi a elaboração de um contraexemplo para a conjectura de que há somente um mínimo local isolado para o problema. Além disso, foi estudado o problema de otimização *intermediário*, que consiste em fixar todas as variáveis do problema exceto duas matrizes de ganhos, e os resultados indicam que, com uma pequena alteração na formulação, este é um problema biquadrático. Por fim, novos algoritmos foram elaborados a partir de um método disponível na literatura, chamado de método Variacional, adaptando-o para atualizar os ganhos aos pares, levando a problemas intermediários biquadráticos. Três métodos foram implementados para a resolução destes problemas: dois métodos clássicos de descida, Newton e Gradiente, e uma adaptação do próprio método Variacional. Para a análise dos resultados foram utilizados exemplos gerados aleatoriamente a partir do *Gerador de SLSM*, que pode ser encontrado na literatura, e o método Variacional como “referência” para comparação com os métodos propostos.

Palavras-Chave: Sistemas lineares com saltos Markovianos, Controle ótimo, Problema biquadrático, Unicidade da solução.

Abstract

This work addresses a control problem arising in linear systems with Markov jumps without observation of the jump variable and advances in three different aspects. First, it is presented a counterexample to the conjecture that states about the uniqueness of local minimum. Second, the *intermediary* optimization problem, which sets all the variables of the problem except two arrays of gains, was studied and the results suggested that a slight modification in the formulation makes the intermediary problem a biquadratic one. Finally, new algorithms were developed based on a method available in the literature, which is frequently referred to as the Variational method, adapting it to update the gains in pairs, leading to biquadratic intermediary problems. Three methods were implemented to solve these intermediary problems: two classical descent methods, Newton and Gradient, and an adaptation of the Variational method. To evaluate the performance of the proposed methods, randomly generated examples were used and the Variational method was set as “reference” for comparing the results.

Keywords: Markovian jumps linear systems, Optimal control, Biquadratic problems, Uniqueness of the solution.

Lista de Siglas

ER	Exemplos Resolvidos
ENR	Exemplos não Resolvidos
ENRMI	Exemplos não Resolvidos dentro do Máximo de Iterações
MV	Método Variacional
MVDG	Método Variacional Duplo com Gradiente
MVDN	Método Variacional Duplo com Newton
MVM	Método Variacional Modificado
PB_k	Problema Biquadrático k
PQ_k	Problema Quadrático k
SLSM	Sistemas Lineares com Saltos Markovianos

Lista de Figuras

3.1	Curvas de nível próximas ao ganho ótimo $g_1^*(27)$	35
3.2	Curvas de nível próximas ao ganho ótimo $g_2^*(27)$	35
5.1	Comportamento da atualização dos ganhos quando utilizado o método Variacional Modificado (setas contínua), presente na maioria dos exemplos do Capítulo 6, e comportamento desejado (seta tracejada).	46
6.1	Classificação dos resultados obtidos pelos métodos Variacional Clássico e Variacional Modificado, para o conjunto A.	58
6.2	Custo do MV vs custo do MVM, para o conjunto A.	58
6.3	Tempo de CPU do MV vs tempo de CPU do MVM, para o conjunto A.	59
6.4	Número de iterações do MV vs número de iterações do MVM, para o conjunto A.	60
6.5	Custo do MV vs custo do MVM, para o conjunto B.	61
6.6	Tempo de CPU do MV vs tempo de CPU do MVM, para o conjunto B.	62
6.7	Número de iterações do MV vs número de iterações do MVM, para o conjunto B.	62
6.8	Classificação dos resultados obtidos pelos métodos Variacional Clássico e Variacional Duplo com Gradiente, para o conjunto A.	63
6.9	Custo do MV vs custo do MVDG, para o conjunto A.	64
6.10	Tempo de CPU do MV vs tempo de CPU do MVDG, para o conjunto A.	65
6.11	Número de iterações do MV vs número de iterações do MVDG, para o conjunto A.	65
6.12	Categorização das iterações do Variacional Duplo com Gradiente, para o conjunto A.	66
6.13	Classificação dos resultados obtidos pelo método Variacional Duplo com Gradiente, para Conjunto B.	67
6.14	Custo do MV vs custo do MVDG, para o conjunto B.	67
6.15	Tempo de CPU do MV vs tempo de CPU do MVDG, para o conjunto B.	68
6.16	Número de iterações do MV vs número de iterações do MVDG, para o conjunto B.	69
6.17	Classificação dos resultados obtidos pelos métodos Variacional Clássico e Variacional Duplo com Newton.	70
6.18	Custo do MV vs custo do MVDN, para o conjunto A.	70
6.19	Tempo de CPU do MV vs tempo de CPU do MVDN, para o conjunto A.	71
6.20	Número de iterações do MV vs número de iterações do MVDN, para o conjunto A.	72
6.21	Classificação dos resultados obtidos pelo método Variacional Duplo com Newton, para Conjunto B.	72
6.22	Custo do MV vs custo do MVDN, para o conjunto B.	73

6.23	Tempo de CPU do MV vs tempo de CPU do MVDN, para o conjunto B.	73
6.24	Número de iterações do MV vs número de iterações do MVDN, para o conjunto B.	74
B.1	Classificação dos resultados obtidos pelos métodos Variacional Clássico e Variacional Modificado, para o conjunto A.	95
B.2	Custo do MV vs custo do MVM, para o conjunto A.	96
B.3	Tempo de CPU do MV vs tempo de CPU do MVM, para o conjunto A.	96
B.4	Número de iterações do MV vs número de iterações do MVM, para o conjunto A.	97
B.5	Custo do MV vs custo do MVM, para o conjunto B.	98
B.6	Tempo de CPU do MV vs tempo de CPU do MVM, para o conjunto B.	98
B.7	Número de iterações do MV vs número de iterações do MVM, para o conjunto B.	99
B.8	Classificação dos resultados obtidos pelos métodos Variacional Clássico e Variacional Duplo com Gradiente, para o conjunto A.	99
B.9	Custo do MV vs custo do MVDG, para o conjunto A.	100
B.10	Tempo de CPU do MV vs tempo de CPU do MVDG, para o conjunto A.	100
B.11	Número de iterações do MV vs número de iterações do MVDG, para o conjunto A.	101
B.12	Classificação dos resultados obtidos pelo método Variacional Duplo com Gradiente, para Conjunto B.	101
B.13	Custo do MV vs custo do MVDG, para o conjunto B.	102
B.14	Tempo de CPU do MV vs tempo de CPU do MVDG, para o conjunto B.	102
B.15	Número de iterações do MV vs número de iterações do MVDG, para o conjunto B.	103
B.16	Classificação dos resultados obtidos pelos métodos Variacional Clássico e Variacional Duplo com Newton.	103
B.17	Custo do MV vs custo do MVDN, para o conjunto A.	104
B.18	Tempo de CPU do MV vs tempo de CPU do MVDN, para o conjunto A.	104
B.19	Número de iterações do MV vs número de iterações do MVDN, para o conjunto A.	105
B.20	Classificação dos resultados obtidos pelo método Variacional Duplo com Newton, para Conjunto B.	105
B.21	Custo do MV vs custo do MVDN, para o conjunto B.	106
B.22	Tempo de CPU do MV vs tempo de CPU do MVDN, para o conjunto B.	106
B.23	Número de iterações do MV vs número de iterações do MVDN, para o conjunto B.	107

Lista de Algoritmos

2.1	Método Variacional	29
2.2	Método de Gradiente	30
2.3	Método de Newton	31
4.1	Passos para Encontrar a Função Biquadrática	41
5.1	Método Variacional Duplo	45
5.2	Método Variacional Modificado	46
5.3	Método Variacional Duplo com Gradiente	53
5.4	Método Variacional Duplo com Newton	54

Sumário

1	Introdução	19
1.1	Organização da Dissertação	21
2	Definições e Resultados Preliminares	23
2.1	Notações	23
2.2	Sistemas Lineares com Saltos Markovianos	24
2.3	Formulação do Problema de Otimização	26
2.4	Breve Descrição do Método Variacional	28
2.5	Método de Gradiente	30
2.6	Método de Newton	31
2.7	Problema Biquadrático	32
3	Características do Problema	33
4	Adaptação do PB_k ao Problema Biquadrático	37
5	Método Variacional com Atualização de Dois Ganhos	43
5.1	Adaptação do Problema para o Cálculo de dois Ganhos Consecutivos	43
5.2	Métodos Computacionais	45
5.2.1	Variacional Modificado	45
5.2.2	Métodos de Descida	47
5.2.2.1	Cálculo das Derivadas de Primeira e Segunda Ordem	47
5.2.2.2	Variacional Duplo com Gradiente	53
5.2.2.3	Variacional Duplo com Newton	54
6	Resultados Computacionais	55
6.1	Variacional Clássico vs Variacional Modificado	57
6.2	Variacional Clássico vs Variacional Duplo com Gradiente	63
6.3	Variacional Clássico vs Variacional Duplo com Newton	69
6.4	Considerações Finais deste Capítulo	75
7	Conclusões	77
	Referências Bibliográficas	79

A	Demonstração do Teorema 4.1	83
A.1	Resultados Preliminares	83
A.2	Demonstração do Teorema 4.1	83
B	Resultados Computacionais Adicionais	95
B.1	Variacional Clássico vs Variacional Modificado	95
B.2	Variacional Clássico vs Variacional Duplo com Gradiente	99
B.3	Variacional Clássico vs Variacional Duplo com Newton	103
C	Código-fonte do Algoritmo 4.1	109
C.1	Códigos-fonte em <i>MATLAB</i> [®]	109

Introdução

Os sistemas lineares estudados nesta pesquisa são chamados de Sistemas Lineares com Saltos Markovianos (SLSM). Uma das principais aplicações desse tipo de sistema é a modelagem de problemas onde há alterações abruptas de comportamento, como sistemas suscetíveis a falhas e perturbações ambientais repentinas. Porém, problemas estocásticos não são os únicos que podem ser modelados por SLSM. Devido à sua capacidade de generalizar e recuperar propriedades de sistemas lineares clássicos, os SLSM podem ser empregados também em problemas com tratamento determinístico. Entre algumas das aplicações de SLSM temos os sistemas robóticos em SARIDIS (1983), os receptores térmicos solares em SWORDER e ROGERS (1983), os sistemas aeronáuticos em STEVENS e LEWIS (1991), os modelos macroeconômicos em DO VAL e BAŞAR (1999) e os portfólios de investimentos em OLIVEIRA (2011). Outros exemplos são citados em SIQUEIRA e TERRA (2004), LAM et al. (2007) e RAOUF e BOUKAS (2007). Esta área tem sido muito investigada nas últimas duas décadas, e o leitor poderá encontrar centenas de referências nos principais livros em SLSM: COSTA et al. (2005), DRAGAN et al. (2009), COSTA et al. (2013).

Os SLSM podem ser divididos em três casos segundo a observação do estado, que para estes sistemas é formado pelo estado da cadeia de Markov, θ , e pelo estado dos subsistemas, x : SLSM com observação completa, com observação parcial, e sem observação. No primeiro caso, as variáveis são conhecidas (medidas com grande precisão) em todos os instantes de tempo, no segundo apenas uma parte destas variáveis são conhecidas, e no terceiro caso não são conhecidos nenhum de seus valores.

Considerando que em muitos casos é difícil conseguir medidas exatas das variáveis de estado, faz-se importante os estudos associados ao problema teórico e à implementação computacional de métodos para os casos com observação parcial/sem observação da cadeia de Markov.

Neste trabalho é considerado o caso sem observação da variável de salto, ou seja, um SLSM com observação parcial. O problema de otimização consiste em minimizar uma função objetivo quadrática, aqui normalmente chamada de *custo*, e com *horizonte* finito N . A variável de otimização consiste em uma sequência de *matrizes de ganho*, denotada por $\mathbf{g} = \{g(0), g(1), \dots, g(N - 1)\}$. Conforme explicado mais detalhadamente na Seção 3, trata-se de um problema de alta dimensionalidade (N matrizes) e não convexo, sendo portanto de difícil tratamento e solução numérica.

Há alguns resultados importantes sobre este problema, sendo provavelmente o mais importante deles o método Variacional proposto e analisado em termos de convergência em DO VAL e BAŞAR (1999). Neste texto, este método também é chamado de *método Variacional Clássico*, para contrapor com os métodos propostos. Vale comentar que há uma demonstração de que este método gera soluções cujos custos diminuem ao longo das iterações, mas não se sabe se a sequência de ganhos converge. Além disso, pode haver convergência para um mínimo local do problema.

As contribuições para a área estão divididas em três aspectos. Um dos avanços foi a elaboração de um contraexemplo para a conjectura de que há somente um mínimo local isolado para o problema. Esse exemplo foi o único obtido em uma base de 1.000 exemplos gerados aleatoriamente, sugerindo que não é algo simples de ser encontrado.

O segundo aspecto foi o estudo do problema de otimização *intermediário*, denominado PB_k , que consiste em fixar todos os ganhos exceto dois deles, $g(k)$ e $g(k + 1)$, e os resultados indicam que, com uma pequena modificação nestas variáveis, este é um problema biquadrático. A demonstração feita considera o caso em que a ação de controle é escalar, porém um algoritmo foi desenvolvido e testado de forma que, considerando 1.000 exemplos gerados aleatoriamente, foi possível obter uma função objetivo biquadrática em todos os sistemas. Este é um forte indício de que o problema é biquadrático.

Por fim, o método Variacional Clássico foi adaptado para atualizar dois ganhos consecutivos. Vale ressaltar que isto pode ser facilmente estendido para o caso de múltiplos ganhos, veja Observação 5.1. A motivação para esse estudo surge do fato de existirem trabalhos que utilizam métodos onde apenas um ganho é atualizado a cada iteração, é o caso do método Variacional Clássico, e métodos que atualizam todos os ganhos simultaneamente, não havendo nenhum caso intermediário.

Em termos gerais, as conclusões obtidas sobre os métodos implementados são: O método Variacional Clássico alcança os melhores resultados, em termos de custo e tempo de CPU, para a grande maioria dos exemplos, porém há uma parcela significativa de exemplos em que a estratégia de atualização de dois ganhos apresenta menor número de iterações, justificando a busca por

métodos mais eficiente para o problema PB_k . Por fim, os métodos de descida utilizados com intuito de diminuir o tempo computacional não apresentaram bons resultados.

Esta pesquisa resultou em dois trabalhos: um para II Congresso de Matemática Aplicada e Computacional - CMAC-SE, apresentado em seção oral, OLIVEIRA e COSTA (2013); e outro para o XXXV Congresso Nacional de Matemática Aplicada e Computacional - CNMAC, OLIVEIRA e COSTA (2014).

1.1 Organização da Dissertação

O restante deste trabalho está estruturado da seguinte maneira: no *Capítulo 2* são apresentadas as notações e os conceitos preliminares, importantes para o entendimento do trabalho, como os resultados sobre SLSM, a definição do problema de otimização e os métodos estudados. No *Capítulo 3* são apresentadas algumas características da função custo de N estágios e o primeiro resultado deste trabalho: a unicidade da solução ótima. Os resultados acerca da adaptação desta função, com atualização de dois ganhos, como uma função biquadrática e a demonstração para o caso em que a ação de controle é escalar são apresentados no *Capítulo 4*. No *Capítulo 5* é apresentada a formulação matemática para atualização de dois ganhos consecutivos e um breve resumo sobre os métodos desenvolvidos para sua resolução. As análises feitas a partir dos resultados obtidos por estes métodos, quando aplicados a dois conjuntos de exemplos gerados aleatoriamente pelo *Gerador de SLSM* de BORTOLIN (2012), são apresentadas no *Capítulo 6*. Por fim, no *Capítulo 7* são apresentadas as conclusões e as propostas para trabalhos futuros.

Definições e Resultados Preliminares

Neste capítulo são apresentadas algumas definições e resultados preliminares necessários para o desenvolvimento da pesquisa, organizados da seguinte forma: na Seção 2.1 são apresentadas as *notações e operadores* utilizados ao longo do texto e na Seção 2.2 as definições e resultados relacionados a Sistemas Lineares com Saltos Markovianos (SLSM). A formulação matemática do problema de otimização é feita na Seção 2.3 e na Seção 2.4 é feita uma breve descrição do método Variacional de VARGAS (2004), que será utilizado para comparar os métodos implementados neste trabalho. Nas Seções 2.5 e 2.6 é feita uma breve descrição dos métodos clássicos de descida, Gradiente e Newton, respectivamente, e por fim, a formulação do problema biquadrático é apresentada na Seção 2.7.

2.1 Notações

Sejam T e N números inteiros e positivos e $\mathcal{T} = \{1, 2, \dots, T\}$ e $\mathcal{K} = \{0, 1, \dots, N - 1\}$. O espaço linear normado formado por todas as matrizes reais com dimensão $n \times m$ ($n \times n$) é representado por $\mathcal{M}^{n,m}$ (\mathcal{M}^n). Define-se $\mathbb{M}^{n,m} = \{U = (U_1, U_2, \dots, U_T) : U_i \in \mathcal{M}^{n,m}, i \in \mathcal{T}\}$ como sendo o espaço linear de todas as T -sequências de matrizes e $\mathcal{S}^n = \{U \in \mathcal{M}^n : U = U'\}$ o subespaço linear normalizado de \mathcal{M}^n de matrizes simétricas, em que U' é o transposto da matriz U . Denota-se $\mathbb{S}^n = \{U = (U_1, U_2, \dots, U_T) : U_i \in \mathcal{S}^n, i \in \mathcal{T}\}$ como sendo o espaço linear de todas as T -sequências de matrizes simétricas. Seja $\mathcal{S}^{n+} = \{U \in \mathcal{S}^n : U \succ 0\}$ e $\mathcal{S}^{n0} = \{U \in \mathcal{S}^n : U \succeq 0\}$ o cone aberto (fechado) de matrizes definidas (semi-definidas) positiva de \mathcal{S}^n , além disso, denota-se o valor esperado por $E\{\cdot\}$, a medida de Dirac de um conjunto S como 1_S , o operador traço por $\text{Tr}\{\cdot\}$ e $\|U\|_2^2 = \sum_{i=1}^T \text{Tr}\{U_i' U_i\}$, para $U \in \mathbb{M}^{n,m}$. O espaço $\mathbb{M}^{n,m}$ equipado

com o produto interno (2.1) forma um espaço de Hilbert.

$$\langle U, V \rangle = \sum_{i=1}^T \text{Tr}\{U_i' V_i\}. \quad (2.1)$$

Sejam $A \in \mathcal{M}^{n,m}$ e $B \in \mathcal{M}^{r,s}$, o produto de Kronecker usual das matrizes A e B , denotado por $A \otimes B$, é dado pela matriz $nr \times ms$ a seguir

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1m}B \\ \vdots & \ddots & \vdots \\ a_{n1}B & \cdots & a_{nm}B \end{bmatrix}, \quad (2.2)$$

e a vetorização da matriz A , denotada por $\text{vec}(A)$, é dada pelo vetor $nm \times 1$ (2.3) de todos os elementos da matriz A empilhados coluna por coluna.

$$\text{vec}(A) = \begin{bmatrix} a_{11} & \cdots & a_{n1} & a_{12} & \cdots & a_{n2} & \cdots & a_{1m} & \cdots & a_{nm} \end{bmatrix}'. \quad (2.3)$$

2.2 Sistemas Lineares com Saltos Markovianos

Em termos simples, um Sistema Linear com Saltos Markovianos (SLSM) é um sistema linear cujos parâmetros variam no tempo e estas variações seguem um padrão que não é determinístico e nem totalmente aleatório. Em geral, há um certo número de (matrizes de) parâmetros, por exemplo A_1 , A_2 e A_3 , e em cada instante de tempo apenas um destes parâmetros está ativo. A mudança entre eles obedece a um processo estocástico, denotado nesse trabalho por $\theta(k)$, uma variável aleatória que toma valores em $\mathcal{T} = \{1, 2, \dots, T\}$. Por exemplo, no instante k , se $\theta(k) = 2$ então a matriz de parâmetros $A_{\theta(k)}$ assume o valor A_2 .

Um processo estocástico $\theta(k)$ pode ter características bastante complexas, por este motivo neste trabalho será considerado um processo mais simples, satisfazendo a propriedade de Markov, levando a um problema mais tratável.

Definição 2.1 *O processo estocástico $\theta(k)$ é uma cadeia de Markov quando*

$$\Pr[\theta(k+1) = j | \theta(0) = i_0, \theta(1) = i_1, \dots, \theta(k) = i] = \Pr[\theta(k+1) = j | \theta(k) = i]. \quad (2.4)$$

Note que a informação sobre $\theta(k)$ é um fator importante para o problema em questão. Existem três casos acerca da observação desta variável: ela pode ser conhecida em todos os instantes k do sistema, ou seja, o sistema possui observação completa das variáveis de estado; pode ser apenas parcialmente conhecida; ou não ser conhecida em nenhum instante.

Para o caso de um sistema que possui observação completa das variáveis de estado, a solução do problema, definido na próxima seção, é facilmente obtida através da resolução de uma Equação Algébrica de Riccati Acoplada (EARA), porém para os casos de observação parcial

e não observação das variáveis de estado, não há um método geral que obtenha de forma satisfatória a solução ótima do sistema. Considerando que em muitos casos é difícil conseguir medidas exatas de $\theta(k)$, faz-se importante o estudo associado a implementação computacional de métodos com observação parcial da cadeia de Markov e a comparação dos mesmos e de suas soluções encontradas.

Neste trabalho é considerado o caso em que $\theta(k)$ não é observado e possui probabilidade de transição do estado i para o estado j dada pela matriz $\mathbb{P} = [p_{ij}] \in \mathcal{M}^T$, em que T é a quantidade de estados da cadeia e p_{ij} é dado por:

$$p_{ij}(k) = \Pr[\theta(k+1) = j | \theta(k) = i]. \quad (2.5)$$

A probabilidade inicial da variável aleatória θ estar no estado $i \in \mathcal{T}$ é dada por $\pi_i(0) = \Pr[\theta(0) = i]$ e o vetor $\pi(0) = [\pi_1(0), \dots, \pi_T(0)]$ é chamado de vetor de distribuição de probabilidade inicial. O vetor de distribuição de probabilidade após k -passos, $\pi(k) = [\pi_1(k), \dots, \pi_T(k)]$, onde $\pi_i(k) = \Pr\{\theta(k) = i\}$ e $i \in \mathcal{T}$, pode ser determinado pelo vetor de condições iniciais $\pi(0)$ e pela k -ésima potência da matriz \mathbb{P} , através da equação de Chapman-Kolmogorov: $\pi(k) = \pi(0)\mathbb{P}^k$.

Assumindo conhecidos os seguintes conjuntos de matrizes $A = \{A_1, A_2, \dots, A_T\} \in \mathbb{M}^n$, $B = \{B_1, B_2, \dots, B_T\} \in \mathbb{M}^{n,m}$, $C = \{C_1, C_2, \dots, C_T\} \in \mathbb{S}^{n_0}$, $D = \{D_1, D_2, \dots, D_T\} \in \mathbb{S}^{m+}$ e $G = \{G_1, G_2, \dots, G_T\} \in \mathbb{M}^{n,p}$, o SLSM possui a seguinte formulação:

$$\Phi_M : \begin{cases} x(k+1) = A_{\theta(k)}x(k) + B_{\theta(k)}u(k) + G_{\theta(k)}w(k), \\ y(k) = x'(k)C_{\theta(k)}x(k) + u'(k)D_{\theta(k)}u(k), \end{cases} \quad (2.6)$$

em que o par (x, θ) determina o *estado do sistema*, $y(k)$ é o chamado *custo por estágio*, $u(k)$ é o *controle*, uma variável que pode ser projetada a fim de minimizar um critério associado ao custo por estágio, e $w(k)$ é uma variável aleatória, independente, com média zero e covariância fixada. Neste trabalho x é uma variável conhecida em todos os instantes de tempo. Além disso, $A_{\theta(k)} = A_i \in A$ quando $\theta(k) = i$ e similarmente para as demais matrizes em (2.6).

Definição 2.2 Considere a matriz $\mathbb{P} = [p_{ij}] \in \mathcal{M}^T$, a sequência de matrizes $G \in \mathbb{M}^{n,p}$ e a distribuição $\pi(k)$ da cadeia de Markov. Para quaisquer $V \in \mathbb{M}^n$ e $U \in \mathbb{S}^{n_0}$, definem-se os operadores lineares $\mathcal{E}, \mathcal{L}_V, \tau_V : \mathbb{S}^{n_0} \rightarrow \mathbb{S}^{n_0}$ e a sequência de matrizes $\Sigma(k) \in \mathbb{S}^{n_0}$ como:

$$\mathcal{E}_i(U) = \sum_{j=1}^T p_{ij} U_j, \quad (2.7)$$

$$\mathcal{L}_{V,i}(U) = V_i' \mathcal{E}_i(U) V_i, \quad (2.8)$$

$$\tau_{V,i}(U) = \sum_{j=1}^T p_{ji} V_j U_j V_j', \quad (2.9)$$

$$\Sigma_i(k) = \sum_{j=1}^T p_{ji} \pi_j(k) G_j G_j', \quad (2.10)$$

sendo que $\tau_V^0(U) = U$ e $\tau_V^{(t)}(U) = \tau_V(\tau_V^{(t-1)}(U))$ para $t \geq 1$.

Definição 2.3 O segundo momento de $x(k)$, condicionado ao estado Markoviano $\theta(k)$ é definido por um conjunto de matrizes $X(k) = \{X_1(k), X_2(k), \dots, X_T(k)\} \in \mathbb{S}^{n_0}$, em que

$$X_i(k) = E\{x(k)x'(k)1_{\theta(k)=i}\}, \quad i \in \mathcal{T}, k \in \mathcal{K}. \quad (2.11)$$

Lema 2.1 (VARGAS (2004)) Seja $U = \{U_1(k), U_2(k), \dots, U_T(k)\} \in \mathbb{S}^{n_0}$, temos que:

$$E\{x'(k)U_{\theta(k)}x(k)\} = \sum_{i=1}^T \text{Tr}\{U_i X_i(k)\} = \langle X(k), U \rangle, \quad \forall k \in \mathcal{K}. \quad (2.12)$$

2.3 Formulação do Problema de Otimização

Sendo $E\{y(k)\}$ o valor esperado do custo por estágio, a função objetivo do problema de otimização será

$$\sum_{k=0}^{N-1} E\{y(k)\}, \quad (2.13)$$

em que $y(k)$ é dado pela Equação (2.6).

Lembrando que neste trabalho x é observado e θ não é observado, ou seja, trata-se de um sistema com observação parcial do estado, seria natural tentar minimizar a função objetivo com base nestas informações, de maneira que o controle resultaria na forma $u(k) = f(x(0), x(1), \dots, x(k))$. É possível mostrar que a solução ótima também pode ser escrita na forma $u(k) = f(x(k), \psi_k)$, sendo ψ_k a distribuição condicionada de θ a todo histórico de observações de x . Este é um problema de controle extremamente complicado. Para que se tenha uma ideia da complexidade, note que é possível escolher $u(k)$ que favoreça obter informação sobre $\theta(k)$ através de $x(k+1)$, assim o controle tem influência direta no sistema e na função objetivo, mas também nas distribuições ψ_k , e estas influências são normalmente conflitantes, o que é comumente chamado de *controle dual*.

Levando em conta a dificuldade explicada acima, consideramos uma restrição simplificadora de que o controle é uma função linear do estado,

$$u(k) = g(k)x(k), \quad k \geq 0, \quad (2.14)$$

sendo a matriz $g(k) \in \mathcal{M}^{m,n}$ denominada de *ganho*, para $k \in \mathcal{K}$, de modo que agora o problema de otimização pode ser reescrito como

$$\min_{g(0), \dots, g(N-1)} \sum_{k=0}^{N-1} E\{x'(k)[C_{\theta(k)} + g'(k)D_{\theta(k)}g(k)]x(k)\}. \quad (2.15)$$

Além disso, pelo mesmo motivo $g(k)$ deve ser escolhido baseando-se apenas nos parâmetros conhecidos, e não na sequência de observações de $x(0), \dots, x(k)$, caso contrário recairíamos no problema de controle dual.

Seja \mathcal{G} o conjunto formado por todas as possíveis seqüências de matrizes de ganho $\mathbf{g} = \{g(0), g(1), \dots, g(N-1)\}$. Seja também $g'(k)Dg(k) = \{(g'(k)D_1g(k)), \dots, (g'(k)D_Tg(k))\}$ e $A + Bg(k) = \{(A_1 + B_1g(k)), \dots, (A_T + B_Tg(k))\}$. A Proposição 2.1, apresenta uma expressão determinística que facilita o cálculo de $X(k)$.

Proposição 2.1 (COSTA et al. (2005)) Considere $V(k) \in \mathbb{S}^{n_0}$ definido de acordo com

$$\begin{cases} V(k+1) = \tau_{A+Bg(k)}(V(k)) + \Sigma(k), \\ V_i(0) = V \in \mathcal{S}^{n_0}. \end{cases}$$

Vale que $V(k) = X(k)$, sempre que $V_i(0) = X_i(0) = \pi_i(0)x(0)x'(0)$, $i \in \mathcal{T}$ e $k \in \mathcal{K}$.

Utilizando o Lema 2.1, em que $U_{\theta(k)} = C_{\theta(k)} + g'(k)D_{\theta(k)}g(k)$, temos

$$E\{y(k)\} = \langle X(k), C + g'(k)Dg(k) \rangle, \quad \forall k \in \mathcal{K}, \quad (2.16)$$

e portanto o custo de N estágios para o sistema Φ terá a seguinte forma:

$$\mathcal{J}_N = \sum_{k=0}^{N-1} E\{y(k)\} = \sum_{k=0}^{N-1} \langle X(k), C + g'(k)Dg(k) \rangle. \quad (2.17)$$

A partir de todos esses resultados é possível formular o problema de otimização, que consiste em determinar uma seqüência de ganhos $\mathbf{g} \in \mathcal{G}$ que minimize a Equação (2.17). Formalmente isto é formulado da seguinte maneira:

$$\begin{aligned} & \min_{\mathbf{g} \in \mathcal{G}} \mathcal{J}_N \\ & \text{s.a. : } \begin{cases} X_i(k+1) = \tau_{A+Bg(k),i}(X(k)) + \Sigma_i(k), \\ X_i(0) = \pi_i(0)x(0)x'(0). \end{cases} \end{aligned} \quad (2.18)$$

Neste trabalho será considerado o caso em que $T \geq 2$, uma vez que com $T = 1$ o problema se reduz ao caso sem saltos nos parâmetros, que já é bem conhecido e sua solução pode ser facilmente obtida via equações de Riccati. Este problema encontra-se em aberto, pois ainda não há um algoritmo que determine os ganhos ótimos \mathbf{g}^* de forma que o custo de N estágios seja garantidamente um mínimo global.

No entanto, há um algoritmo denominado de método Variacional, proposto por DO VAL e BAŞAR (1999) e também estudado por VARGAS (2004), para o problema do custo de N estágios. Por se tratar de um método baseado em uma condição necessária de otimalidade, ele encontra apenas um ponto estacionário de \mathcal{J}_N . Este método é explicado mais detalhadamente a seguir, uma vez que ele é usado para comparação de resultados ao longo desse texto.

2.4 Breve Descrição do Método Variacional

Em DO VAL e BAŞAR (1999) é proposto um método Variacional para tratar de problemas de controle de SLSM sem ruído aditivo, cuja extensão para problemas com ruído aditivo está em VARGAS (2004). Para facilitar a referência, o método Variacional será descrito a seguir.

Definição 2.4 *Sejam os operadores $\mathcal{E}, \mathcal{L} : \mathbb{S}^{n_0} \rightarrow \mathbb{S}^{n_0}$ definidos pelas Equações (2.7) e (2.8), respectivamente. Definem-se $L(k) \in \mathbb{S}^{r_0}$ e $\omega(k) \in \mathbb{M}^1$ como:*

$$L_i(k) = C_i + g'(k)D_i g(k) + \mathcal{L}_{A+Bg(k)}(L(k+1)), \quad (2.19)$$

$$\omega_i(k) = \mathcal{E}(\omega(k+1)) + \text{Tr}\{\mathcal{E}(L(k+1))G_i G_i'\}, \quad (2.20)$$

sendo $L_i(N) = 0$ e $\omega_i(N) = 0$, $i \in \mathcal{T}$.

Proposição 2.2 (VARGAS et al. (2004)) *Para cada $k_0 \in \mathcal{K}$, vale a seguinte identidade:*

$$\sum_{k=k_0}^{N-1} E\{y(k)\} = \langle L(k_0), X(k_0) \rangle + \pi(k_0)' \omega(k_0). \quad (2.21)$$

O custo de N estágios apresentando em (2.17) pode ser escrito como

$$\begin{aligned} \mathcal{J}_N = & \sum_{\ell=0}^{k-1} \langle X(\ell), C + g'(\ell)Dg(\ell) \rangle + \langle X(k), C + g'(k)Dg(k) \rangle \\ & + \sum_{\ell=k+1}^{N-1} \langle X(\ell), C + g'(\ell)Dg(\ell) \rangle. \end{aligned} \quad (2.22)$$

Utilizando (2.21), temos que a Equação (2.22) pode ser reescrita da seguinte forma:

$$\begin{aligned} \mathcal{J}_N = & \sum_{\ell=0}^{k-1} \langle X(\ell), C + g'(\ell)Dg(\ell) \rangle + \langle X(k), C + g'(k)Dg(k) \rangle \\ & + \langle L(k+1), X(k+1) \rangle + V(k+1), \end{aligned} \quad (2.23)$$

em que $L(k+1)$ e $V(k+1) = \pi(k+1)' \omega(k+1)$ são conhecidos e $X(k+1) = \tau_{A+Bg(k)}(X(k)) + \Sigma(k)$.

Considerando os ganhos de $g(0)$ até $g(k-1)$ e de $g(k+1)$ até $g(N-1)$ conhecidos na Equação (2.23), resta apenas uma partição (k) da sequência de ganhos a ser atualizada. A fim de facilitar futuras referências o problema de atualizar a partição k da sequência de ganhos será denominado de *Problema Quadrático k* , ou PQ_k , uma vez que o problema é quadrático na incógnita $g(k)$. A Equação (2.24) apresenta a formulação deste problema.

$$PQ_k : \min_{g(k)} \mathcal{J}_N \quad (2.24)$$

sendo

$$\mathcal{J}_N = \langle X(k), C + g'(k)Dg(k) \rangle + \langle L(k+1), \tau_{A+Bg(k)}(X(k)) + \Sigma(k) \rangle + \tilde{V}(k+1),$$

em que as matrizes $X(k)$, $L(k+1)$ e o escalar $\tilde{V}(k+1)$ são conhecidos.

Derivando a equação do PQ_k em relação a incógnita $g(k)$, o teorema da otimalidade para o método Variacional é estabelecido conforme a seguir.

Teorema 2.1 (VARGAS et al. (2004)) *Suponha que exista uma sequência de ganhos \mathbf{g} que realiza o mínimo global do problema (2.18), então \mathbf{g} satisfaz, para cada $k \in \mathcal{K}$, a seguinte equação:*

$$\sum_{i=1}^T [(D_i + B_i' \mathcal{E}_i(L(k+1))B_i) g(k) + B_i' \mathcal{E}_i(L(k+1))A_i] X_i(k) = 0. \quad (2.25)$$

A partir desses resultados obtemos o Algoritmo 2.1 para o método Variacional que minimiza localmente o *custo de N estágios*, satisfazendo a condição necessária de otimalidade apresentada em (2.25). O Exemplo 2.1, adaptado de BORTOLIN (2012), ilustra a utilização deste método.

Algoritmo 2.1 Método Variacional

Passo 0: Inicie o processo com $\eta = 0$ e escolha uma sequência de ganhos iniciais $\mathbf{g}^\eta = \{g^\eta(0), \dots, g^\eta(N-1)\}$.

Passo 1: $\forall k \in \mathcal{K}$ encontre $X^\eta(k) \in \mathcal{S}^n$ tal que

$$\begin{cases} X^\eta(k+1) = \tau_{A+Bg^\eta(k)}(X^\eta(k)) + \Sigma_i(k), \\ X_i^\eta(0) = \pi_i(0)x(0)x'(0), \end{cases} \quad (2.26)$$

onde τ e Σ são dados pelas Equações (2.9) e (2.10), respectivamente.

Passo 2: Faça $\eta = \eta + 1$. Para cada $k = N-1, N-2, \dots, 0$, determine $g^\eta(k)$ que minimize PQ_k através da Equação (2.25) com $L_i(k)$ dado pela Definição 2.4.

Critério de Parada: $|\mathcal{J}_N(\mathbf{g}^\eta) - \mathcal{J}_N(\mathbf{g}^{\eta-1})| / \mathcal{J}_N(\mathbf{g}^\eta) < \epsilon$, para ϵ dado. Se o critério não for satisfeito volte ao passo 2.

Exemplo 2.1 *Considere os seguintes dados:*

$$A_1 = \begin{bmatrix} 0,7633 & -0,0215 & 0,1252 \\ -0,6456 & 0,2100 & -0,0592 \\ -0,4566 & -0,2645 & 0,5546 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -6,7366 & 7,5238 & -1,0969 \\ -5,2434 & 5,7396 & -0,6748 \\ 2,5139 & -3,4616 & 1,0579 \end{bmatrix},$$

$$C_1 = \begin{bmatrix} 1,2178 & 0,4220 & -0,3873 \\ 0,4220 & 0,1463 & -0,1342 \\ -0,3873 & -0,1342 & 0,1231 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 18,5930 & -21,9334 & 6,0864 \\ -21,9334 & 26,1372 & -7,3977 \\ 6,0864 & -7,3977 & 2,1727 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 0,4313 \\ 0,0739 \\ 3,9401 \end{bmatrix}, \quad B_2 = \begin{bmatrix} -0,9231 \\ -0,7560 \\ 0,3353 \end{bmatrix}, \quad G_1 = \begin{bmatrix} -0,2889 \\ 1,9192 \\ 4,6867 \end{bmatrix}, \quad G_2 = \begin{bmatrix} -0,0773 \\ -0,0126 \\ 0,4450 \end{bmatrix},$$

$$P = \begin{bmatrix} 0,7514 & 0,2486 \\ 0,0178 & 0,9822 \end{bmatrix}, \quad \pi(0) = [0,3263 \quad 0,6737], \quad x(0) = \begin{bmatrix} -1,1634 \\ 1,1837 \\ -0,0154 \end{bmatrix},$$

$D_1 = 0,15$, $D_2 = 0,5861$, $\epsilon = 10^{-4}$, $N = 100$, $T = 2$ e $g^0(k) = \begin{bmatrix} 0 & 0 \end{bmatrix}$, para todo $k \in \mathcal{K}$.

O Algoritmo 2.1 foi implementado utilizando o software MATLAB[®] e realizou $\eta = 5$ iterações até localizar uma solução, cujo custo de N estágios, \mathcal{J}_N , foi aproximadamente igual a 546,2521.

A sequência de ganhos \mathbf{g}^* pode ser encontrada no Apêndice D de BORTOLIN (2012).

2.5 Método de Gradiente

Seja uma função contínua $f(z)$ que possui derivadas parciais contínuas de primeira ordem. O vetor gradiente $\nabla f(z)$ aponta para a direção de maior crescimento no valor da função $f(z)$, logo se estamos buscando o valor mínimo desta função, é natural que caminhemos na direção contrária ao gradiente.

O método de Gradiente é um método iterativo de busca linear que usa, na iteração η , a direção de descida $d_\eta = -\nabla f(z)$. A sequência de pontos é gerada por

$$z^{\eta+1} = z^\eta + \alpha_\eta \cdot d_\eta, \quad (2.27)$$

em que a distância $\alpha_\eta \geq 0$, a se mover ao longo da direção d_η , pode ser encontrada resolvendo-se o problema unidimensional na variável α que minimiza $f(z^\eta + \alpha \cdot d_\eta)$. O Algoritmo 2.2 apresenta os passos do método de Gradiente.

Algoritmo 2.2 Método de Gradiente

Passo 0: Inicie o processo com $\eta = 0$ e escolha um ponto inicial z_0 .

Passo 1: Calcule $\nabla f(z)$.

Passo 2: Encontre α_η resolvendo o problema unidimensional $\min_\alpha f(z^\eta + \alpha \cdot d_\eta)$.

Passo 3: Atualize $z^{\eta+1} = z^\eta + \alpha_\eta \cdot d_\eta$, em que $d_\eta = -\nabla f(z)$.

Critério de Parada: Se $\|z^{\eta+1} - z^\eta\| / \|z^{\eta+1}\| < \epsilon$, para ϵ dado, a solução ótima foi encontrada, PARE. Se não, faça $\eta = \eta + 1$ e volte ao passo 1.

O método de Gradiente pode ser diretamente empregado em (2.18) na atualização de todos os ganhos, ou seja, fazendo $z^\eta = (g^\eta(0), g^\eta(1), \dots, g^\eta(N-1))$. Contudo essa abordagem não será explorada neste texto, uma vez que já foi estudada em BORTOLIN (2012).

2.6 Método de Newton

O método de Newton estende a ideia do método de Gradiente e consiste em aproximar repetidamente a função $f(z)$ por uma função quadrática $Q(z)$ em torno de um ponto dado z^η até que um critério de parada previamente definido seja satisfeito.

Considere a expansão por série de Taylor até a segunda ordem da função $f(z)$ nas vizinhanças do ponto z^η

$$f(z) \simeq Q(z) = f(z) + \nabla f(z^\eta)(z - z^\eta) + \frac{1}{2}(z - z^\eta)'H(z^\eta)(z - z^\eta), \quad (2.28)$$

em que $\nabla f(z^\eta)$ é o vetor gradiente e $H(z^\eta)$ é a matriz hessiana da função $f(z)$ no ponto z^η .

Sendo a derivada de $Q(z)$ com relação a variável z^η igual a zero, temos

$$\nabla f(z^\eta) + H(z^\eta)(z - z^\eta) = 0. \quad (2.29)$$

Assim, temos que a direção de busca do método de Newton, na iteração η , é dada pelo sistema linear

$$H(z^\eta) \cdot d_\eta = -\nabla f(z^\eta), \quad (2.30)$$

e a sequência de pontos obtida durante as iterações do método por

$$z^{\eta+1} = z^\eta + d_\eta. \quad (2.31)$$

Seja z^* um mínimo local e assumindo $\nabla f(z^*) = 0$, $H(z^*)$ definida positiva e $f(z)$ uma função cujas derivadas parciais de primeira e segunda ordem são contínuas, temos que $H(z^\eta)$ é definida positiva nos pontos próximos a z^* , assim o ponto sucessor $z^{\eta+1}$ é bem definido. O Algoritmo 2.3 apresenta os passos do método de Newton.

Algoritmo 2.3 Método de Newton

Passo 0: Inicie o processo com $\eta = 0$ e escolha um ponto inicial z_0 .

Passo 1 Calcule $\nabla f(z^\eta)$ e $H(z^\eta)$.

Passo 2: Resolva o sistema linear: $H(z^\eta) \cdot d_\eta = -\nabla f(z^\eta)$.

Passo 3: Atualize $z^{\eta+1} = z^\eta + d_\eta$.

Critério de Parada: Se $\|z^{\eta+1} - z^\eta\|/\|z^{\eta+1}\| < \varepsilon$, para ε dado, a solução ótima foi encontrada, PARE. Se não, faça $\eta = \eta + 1$ e volte ao passo 1.

Da mesma forma que o método de Gradiente, o método de Newton também pode ser diretamente empregado em (2.18) para atualização simultânea de todos os ganhos. Os resultados desta adaptação podem ser encontrados em BORTOLIN (2012).

2.7 Problema Biquadrático

Recentemente, a otimização polinomial, em especial os problemas polinomiais de grau maior que dois, tem despertado o interesse para pesquisa. Sendo um caso especial, o problema biquadrático pode ser aplicado em diferentes áreas como física quântica, comunicação sem fio e processamento de imagem e sinal conforme citado em ZHANG et al. (2011).

A Equação (2.32) apresenta a formulação deste problema.

$$\min_{v,z} Q(v, z) = \sum_{i,k=1}^n \sum_{j,l=1}^m a_{ijkl} v_i z_j v_k z_l, \quad (2.32)$$

em que é assumido, sem perda de generalidade, que os coeficientes a_{ijkl} satisfazem a propriedade simétrica $a_{ijkl} = a_{kjil} = a_{ilkj}$, para $i, k = 1, \dots, n$ e $j, l = 1, \dots, m$. Claramente temos que se v ou z forem fixos, a Equação (2.32) torna-se um problema de otimização quadrático.

Na literatura, este problema é encontrado sempre sujeito a um conjunto de restrições, por exemplo, em LING et al. (2009) o problema de otimização biquadrático está sujeito a $\|v\| = 1$ e $\|z\| = 1$; e em YANG e Q. (2012), ZHANG et al. (2011) e LING et al. (2012) está sujeito a restrições quadráticas.

Características do Problema

O problema de otimização introduzido na Equação (2.18) é definido como um problema de otimização contínua, não linear e irrestrita, pois é possível reescrevê-lo da seguinte forma reduzida

$$\begin{aligned}
 \min_{g \in \mathcal{G}} \quad & \langle (X(0)), C + g'(0)Dg(0) \rangle + \sum_{k=1}^{N-1} \langle \tau_{A+Bg(k-1)} \tau_{A+Bg(k-2)} \dots \tau_{A+Bg(0)} (X(0)) \\
 & + \tau_{A+Bg(k-1)} \dots \tau_{A+Bg(1)} (\Sigma(0)) + \tau_{A+Bg(k-1)} \dots \tau_{A+Bg(2)} (\Sigma(1)) + \dots \\
 & + \tau_{A+Bg(k-1)} (\Sigma(k-2)) + \Sigma(k-1), C + g'(k)Dg(k) \rangle,
 \end{aligned} \tag{3.1}$$

em que a condição inicial é dada por $X_i(0) = \pi_i(0)x(0)x'(0)$. Apesar do instante k assumir apenas valores inteiros, caracterizando assim um problema de tempo discreto, o problema de otimização é classificado como contínuo, pois as incógnitas $g(k) \in \mathcal{M}^{m,n}$, $k \in \mathcal{K}$ podem assumir quaisquer valores reais. Vale ressaltar que a função objetivo de (3.1) não é convexa, veja VARGAS et al. (2013).

Um dos métodos disponíveis na literatura, e que detem os melhores resultados para esta classe de problemas, é o método Variacional. Proposto por DO VAL e BAŞAR (1999) e posteriormente estudado por diversos autores, este método baseia-se em uma condição necessária, conforme apresentado em VARGAS et al. (2004), do tipo “gradiente nulo”, dada pela Equação (2.25).

Ao longo dos anos surgiram alguns trabalhos que conjecturaram sobre a suficiência desta condição, por exemplo VARGAS et al. (2005), sem chegar contudo a uma conclusão definitiva. Foi observado recentemente, em VARGAS et al. (2013), que a questão da unicidade do mínimo da função objetivo continua em aberto.

Durante o levantamento dos conjuntos de exemplos utilizados nos testes computacionais, foi encontrado um exemplo, cujos dados são apresentados no Exemplo 3.1, que a partir de ganhos iniciais distintos obtém mínimos também distintos. Estes resultados mostram que o problema de custo quadrático de N estágios para SLSM pode apresentar mínimos locais múltiplos e distintos. Em particular, conclui-se que a condição de otimalidade dada pela Equação (2.25) não é suficiente e que o método Variacional pode convergir para diferentes mínimos locais.

Exemplo 3.1 *Considere os parâmetros:*

$$\begin{aligned}
A_1 &= \begin{bmatrix} 0.3182 & -0.4145 \\ 0.5992 & 1.3150 \end{bmatrix}, A_2 = \begin{bmatrix} -1.6236 & -0.1252 \\ 1.6573 & -2.5347 \end{bmatrix}, A_3 = \begin{bmatrix} -1.5097 & 2.7404 \\ -3.8474 & 2.7029 \end{bmatrix}, A_4 = \begin{bmatrix} -24.5608 & 35.9306 \\ -12.6778 & 18.0155 \end{bmatrix}, \\
C_1 &= \begin{bmatrix} 2.2161 & -2.5044 \\ -2.5044 & 2.8302 \end{bmatrix}, C_2 = \begin{bmatrix} 0.0037 & 0.0808 \\ 0.0808 & 1.7709 \end{bmatrix}, C_3 = \begin{bmatrix} 0.4955 & -0.4143 \\ -0.4143 & 0.3464 \end{bmatrix}, C_4 = \begin{bmatrix} 9.2357 & -12.4190 \\ -12.4190 & 16.6996 \end{bmatrix}, \\
B_1 &= \begin{bmatrix} -0.7975 \\ -0.1901 \end{bmatrix}, B_2 = \begin{bmatrix} -0.1653 \\ 0.7806 \end{bmatrix}, B_3 = \begin{bmatrix} 0.3732 \\ -0.2935 \end{bmatrix}, B_4 = \begin{bmatrix} -0.7853 \\ -0.4005 \end{bmatrix}, G_1 = \begin{bmatrix} 2.4078 \\ 0.7081 \end{bmatrix}, G_2 = \begin{bmatrix} 0.7858 \\ -0.8289 \end{bmatrix}, \\
G_3 &= \begin{bmatrix} 0.5203 \\ 0.3073 \end{bmatrix}, G_4 = \begin{bmatrix} -0.5039 \\ -0.2670 \end{bmatrix}, \pi(0) = [0.2440 \quad 0.1691 \quad 0.5412 \quad 0.0457], x(0) = \begin{bmatrix} -0.5212 \\ 0.0605 \end{bmatrix}, D_1 = \\
0.3613, D_2 &= 0.4700, D_3 = 0.0588, D_4 = 0.3300, T = 4, N = 27 \text{ e } P = [p_{ij}], \text{ onde } p_{11} = p_{22} = p_{33} = 1, p_{42} = \\
p_{43} &= 0.2840, p_{44} = 0.4321 \text{ e os demais nulos.}
\end{aligned}$$

Para obter os resultados apresentados a seguir o Algoritmo 2.1 foi implementado através do software *MATLAB*[®] na versão 8.1. O método foi inicializado com duas sequências de ganhos iniciais distintas $\mathbf{g}_1^0(k) = [0 \quad 0]$ e $\mathbf{g}_2^0(k) = [-2.1563 \quad 1.8474]$, para $k = \{0, \dots, N - 1\}$, e tolerância $\epsilon = 10^{-16}$.

A partir da sequência de ganhos iniciais \mathbf{g}_1^0 obtemos a sequência de ganhos ótimos \mathbf{g}_1^* cujo custo ótimo é $\mathcal{J}_{N_1}^* \approx 2.72989 \times 10^{22}$. Já com a sequência de ganhos \mathbf{g}_2^0 obtemos a sequência de ganhos ótimos \mathbf{g}_2^* com custo ótimo inferior $\mathcal{J}_{N_2}^* \approx 1.36997 \times 10^{21}$.

Foi verificado que os custos são mais altos na vizinhança de \mathbf{g}_1^* através de diversos gráficos de curvas de nível. Para isto foram tomados diferentes pares de ganhos, por exemplo os valores da posição $[1, 1]$ de $g_1^*(k)$ com os valores da posição $[1, 2]$ de $g_1^*(k + 1)$, porém em todos os casos as curvas obtidas foram semelhantes àquela mostrada na Figura 3.1. O mesmo vale para a vizinhança de \mathbf{g}_2^* , representada pela Figura 3.2. Em ambas figuras o ganho ótimo é destacado.

Concluimos formalizando o seguinte resultado.

Teorema 3.1 *O problema de otimização (2.18) possui múltiplos minimizadores locais.*

Outra característica importante do problema de otimização, dado pela Equação (2.18), que vale a pena ser ressaltada, pois afeta diretamente o custo computacional necessário para sua resolução, é a alta dimensionalidade, uma vez que suas variáveis de decisão estão diretamente ligadas ao horizonte do problema. Por exemplo, seja um sistema com horizonte $N = 100$, isto implica que será necessário determinar um total de 100 matrizes de ganho $g(k), k \in \mathcal{K} = \{0, 1, \dots, N - 1\}$.

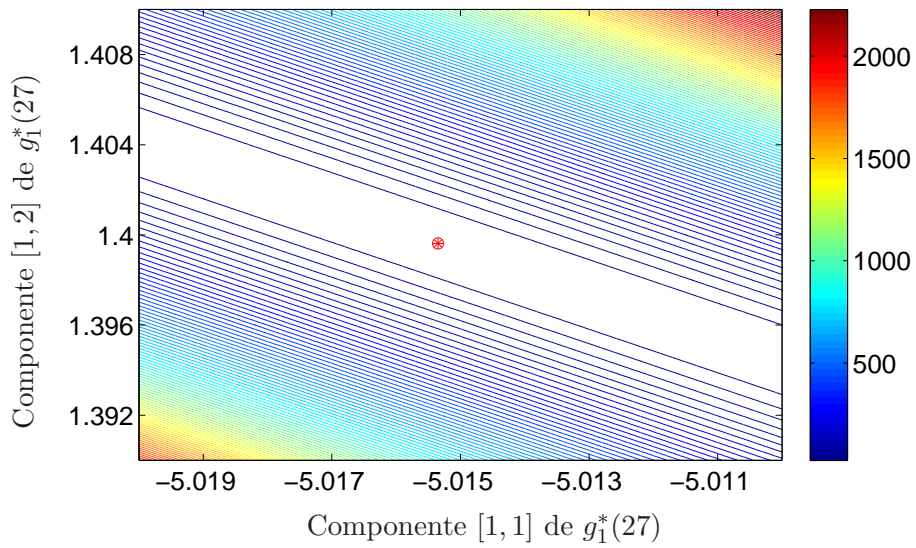


Figura 3.1: Curvas de nível próximas ao ganho ótimo $g_1^*(27)$.

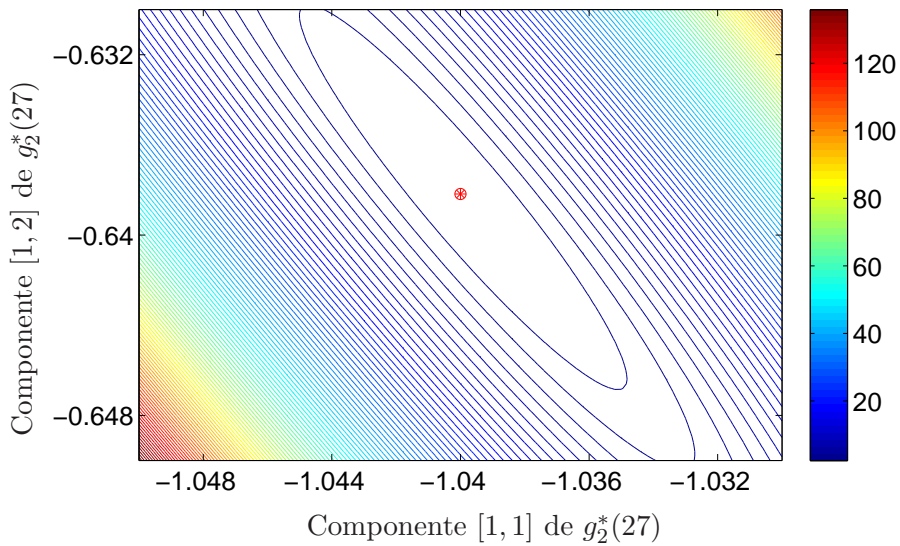


Figura 3.2: Curvas de nível próximas ao ganho ótimo $g_2^*(27)$.

Além da alta dimensionalidade, a complexidade do problema é outro aspecto relevante e está diretamente relacionado a “quantidade” de ganhos que são atualizados por vez. Considere inicialmente o caso de atualizar apenas um ganho, ou seja, os ganhos de $g(0)$ até $g(k-1)$ e de $g(k+1)$ até $g(N-1)$ são conhecidos, assim o somatório em (2.23) e as variáveis $L(k+1)$ e $V(k+1)$ estão disponíveis, restando como variável apenas termos *quadráticos* de $g(k)$ (veja PQ_k em Equação (2.24)). Este é um problema simples cuja solução é encontrada apenas derivando a equação do PQ_k e igualando o resultado a zero.

Agora, considere conhecidos os ganhos de $g(0)$ até $g(k - 1)$ e de $g(k + 2)$ até $g(N - 1)$, ou seja, o caso em que a atualização é feita em dois ganhos, $g(k)$ e $g(k + 1)$, simultaneamente. Não há resultados disponíveis na literatura que caracterizam este problema, assim parte dessa lacuna é preenchida no capítulo a seguir.

Adaptação do PB_k ao Problema Biquadrático

Seja a seguinte forma expandida em dois ganhos da Equação (2.23):

$$\begin{aligned} \mathcal{J}_N = & \sum_{\ell=0}^{k-1} \langle X(\ell), C + g'(\ell)Dg(\ell) \rangle + \langle X(k), C + g'(k)Dg(k) \rangle \\ & + \langle X(k+1), C + g'(k+1)Dg(k+1) \rangle \\ & + \langle L(k+2), X(k+2) \rangle + V(k+2). \end{aligned} \quad (4.1)$$

Considerando a matriz $L(k+2)$, o escalar $V(k+2)$ e os ganhos de $g(0)$ até $g(k-1)$ e de $g(k+2)$ até $g(N-1)$ conhecidos na Equação (4.1), obtemos o problema de atualizar duas “partições” da sequência de ganhos. Conforme será visto a seguir, há fortes indícios que este problema é biquadrático, por isso, a fim de facilitar futuras referências, ele será denominado de PB_k . A Equação (4.2) apresenta sua formulação.

$$PB_k : \quad \min_{g(k), g(k+1)} \mathcal{J}_N \quad (4.2)$$

sendo

$$\begin{aligned} \mathcal{J}_N = & \langle X(k), C + g'(k)Dg(k) \rangle + \langle X(k+1), C + g'(k+1)Dg(k+1) \rangle \\ & + \langle L(k+2), X(k+2) \rangle + \tilde{V}(k+2), \end{aligned}$$

com $X(k+1) = \tau_{A+Bg(k)}(X(k)) + \Sigma(k)$, $X(k+2) = \tau_{A+Bg(k+1)}\tau_{A+Bg(k)}(X(k)) + \tau_{A+Bg(k+1)}(\Sigma(k)) + \Sigma(k+1)$; e $X(k)$, $L(k+2)$ e $\tilde{V}(k+2)$ conhecidos. Note que se uma das variáveis for fixada tanto na Equação (2.32) quanto na Equação (4.1), ambas passarão a ter uma forma quadrática em relação àquela variável não fixada.

Conjectura 4.1 A Equação (4.1) é biquadrática, podendo ser reescrita na forma dada pela Equação (2.32).

Primeiramente, a caracterização do problema será feita a partir de um sistema escalar bem simples, cujos dados são apresentados no Exemplo 4.1.

Exemplo 4.1 Considere o SLSM com os seguintes parâmetros $A_1 = 0.3$, $A_2 = 0.1$, $B_1 = -1$, $B_2 = 1$, $G_1 = G_2 = 0$, $C_1 = C_2 = 0.4$, $D_1 = D_2 = 1$, $N = 2$, $x(0) = 2$, $\pi(0) = [0.25 \quad 0.75]$ e $\mathbb{P} = [p_{ij}]$, com $i, j = 1, 2$ onde $p_{11} = 0.6$, $p_{12} = 0.4$, $p_{21} = 0.2$ e $p_{22} = 0.8$. Considere também $g(0) = p$ e $g(1) = q$, apenas por simplicidade. Depois de algumas manipulações algébricas em (2.18) a função objetivo fica:

$$5p^2q^2 - 0.12p^2q + 5.642p^2 - 0.2pq^2 + 0.088pq - 0.0148p + 0.17q^2 - 0.0124q + 1.65074. \quad (4.3)$$

Note que não é possível reescrever a Equação (4.3) com $v = g(0) = p$ e $z = g(1) = q$ de forma que a mesma fique parecida com $\sum_{i,k=1}^n \sum_{j,l=1}^m a_{ijkl} v_i z_j v_k z_l$, uma vez que há termos como p^2q , pq e p . Uma forma de resolver esta questão foi considerar $v = [g(0) \quad 1]$ e $z = [g(1) \quad 1]$, onde $m = n = 2$, obtendo a seguinte forma expandida de (2.32)

$$\begin{aligned} \sum_{i,j=1}^2 \sum_{k,l=1}^2 a_{ijkl} v_i v_j z_k z_l = & a_{1111} p^2 q^2 + a_{2111} p q^2 + a_{1211} p^2 q + a_{1121} p q^2 \\ & + a_{1112} p^2 q + a_{2211} p q + a_{2121} q^2 + a_{2112} p q + a_{1221} p q \\ & + a_{1212} p^2 + a_{1122} p q + a_{2221} q + a_{2122} q + a_{2212} p + a_{1222} p + a_{2222}. \end{aligned}$$

Assim, é possível reescrever todos os termos da Equação (4.3) e ainda garantir que $a_{ijkl} = a_{kjil} = a_{ilkj}$, para $i, k \in \{1, \dots, n\}$ e $j, l \in \{1, \dots, m\}$.

O exemplo acima deixa claro que é preciso adicionar uma posição com valor constante, que por simplicidade será considerado o valor 1, ao vetor de ganho, contudo algumas adaptações dimensionais serão necessárias às matrizes de coeficientes. Neste texto, será realizada somente a demonstração para o caso em que o controle $u(k)$ é escalar, ou seja, o ganho é vetorial, $g(k) \in \mathcal{M}^{1,n}$.

Lema 4.1 Sejam $\tilde{A} = \{\tilde{A}_1, \tilde{A}_2, \dots, \tilde{A}_T\} \in \mathbb{M}^{n+1}$, $\tilde{B} = \{\tilde{B}_1, \tilde{B}_2, \dots, \tilde{B}_T\} \in \mathbb{M}^{n+1,1}$, $\tilde{C} = \{\tilde{C}_1, \tilde{C}_2, \dots, \tilde{C}_T\} \in \mathbb{S}^{(n+1)0}$, $\tilde{D} = \{\tilde{D}_1, \tilde{D}_2, \dots, \tilde{D}_T\} \in \mathbb{S}^{1+}$, $\tilde{G} = \{\tilde{G}_1, \tilde{G}_2, \dots, \tilde{G}_T\} \in \mathbb{M}^{n+1,p}$ e $\tilde{g}(k) \in \mathcal{M}^{1,n+1}$ dados, por exemplo, da seguinte forma:

$$\tilde{A}_{\theta(k)} = \begin{bmatrix} * & & & & & \\ & A_{\theta(k)} & & & & \\ & \vdots & & & & \\ & * & & & & \\ 0 & \dots & 0 & & & * \end{bmatrix}, \tilde{B}_{\theta(k)} = \begin{bmatrix} B_{\theta(k)} \\ 0 \end{bmatrix}, \tilde{C}_{\theta(k)} = \begin{bmatrix} C_{\theta(k)} & & & * \\ & \vdots & & \\ & * & & \\ * & \dots & * & * \end{bmatrix}, \tilde{G}_{\theta(k)} = \begin{bmatrix} G_{\theta(k)} \\ 0 \quad \dots \quad 0 \end{bmatrix},$$

$\tilde{D}_{\theta(k)} = D_{\theta(k)}$ e $\tilde{g}(k) = [g(k) \quad 1]$, $\forall k \in \mathcal{K}$, em que $*$ pode assumir qualquer valor. Seja o SLSM

$$\tilde{\Phi} : \begin{cases} x(k+1) = (\tilde{A}_{\theta(k)} + \tilde{B}_{\theta(k)}\tilde{g}(k))x(k) + \tilde{G}_{\theta(k)}w(k), \\ y(k) = x'(k)(\tilde{C}_{\theta(k)} + \tilde{g}'(k)\tilde{D}_{\theta(k)}\tilde{g}(k))x(k), \end{cases}$$

então vale que $\tilde{\mathcal{J}}_N = \mathcal{J}_N$ com condição inicial $\pi_i(0)$ e $\tilde{x}(0) = \begin{bmatrix} x(0) \\ 0 \end{bmatrix}$ em (2.18).

Demonstração: É obtida por inspeção direta do funcional de custo. Temos que $\tilde{x}(k) = \begin{bmatrix} x(k) \\ 0 \end{bmatrix}$, para $k \geq 0$, assim

$$\begin{aligned} \tilde{\mathcal{J}}_N &= \sum_{k=0}^{N-1} E\{\tilde{x}'(k)[\tilde{C}_{\theta(k)} + \tilde{g}'(k)\tilde{D}_{\theta(k)}\tilde{g}(k)]\tilde{x}(k)\} \\ &= \sum_{k=0}^{N-1} E\{x'(k)[C_{\theta(k)} + g'(k)D_{\theta(k)}g(k)]x(k)\} = \mathcal{J}_N. \end{aligned} \quad (4.4)$$

■

Teorema 4.1 Seja $\tilde{g}(k) = [g(k) \quad 1]$, $\forall k \in \mathcal{K}$. O custo de N estágios ($\tilde{\mathcal{J}}_N$), expandido conforme o Lema 4.1, pode ser escrito como:

$$\tilde{\mathcal{J}}_N = \sum_{i=1}^T \sum_{j=1}^T p_{ij}(\mu \cdot \text{vec}[\text{vec}(\tilde{g}'(k)\tilde{g}(k))\text{vec}'(\tilde{g}'(k+1)\tilde{g}(k+1))]) + \hat{V}(k+2),$$

em que

$$\mu = \sum_{h=1}^{23} \mu_h \quad e \quad \hat{V}(k+2) = \tilde{V}(k+2) + \sum_{m=1}^3 \xi_m$$

com

$$\begin{aligned} \mu_1 &= \text{vec}'[P \text{vec}(\tilde{D}_i \tilde{X}_i(k)) \text{vec}'(a'a)], \\ \mu_2 &= \text{vec}'[\text{vec}(\text{vec}(\tilde{X}_i(k) \tilde{A}'_i \tilde{C}'_j \tilde{B}_i) a) \text{vec}'(a'a)], \\ \mu_3 &= \text{vec}'[\text{vec}(a' \text{vec}(\tilde{B}'_i \tilde{C}'_j \tilde{A}_i \tilde{X}_i(k))) \text{vec}'(a'a)], \\ \mu_4 &= \text{vec}'[P \text{vec}(\tilde{B}'_i \tilde{C}'_j \tilde{B}_i \tilde{X}_i(k)) \text{vec}'(a'a)], \\ \mu_5 &= \text{vec}'[\text{vec}(a'a) \text{vec}'(\tilde{D}_j \tilde{A}_i \tilde{X}_i(k) \tilde{A}'_i) P], \\ \mu_6 &= \text{vec}'[(\tilde{X}_i(k) \tilde{A}'_i \otimes a' \tilde{B}'_i \tilde{D}_j) P], \\ \mu_7 &= \text{vec}'[(a' \tilde{B}'_i \otimes \tilde{X}_i(k) \tilde{A}'_i \tilde{D}_j) P], \\ \mu_8 &= \text{vec}'[\text{vec}'((\tilde{B}'_i \otimes \tilde{B}'_i \tilde{D}_j) P) \otimes \text{vec}'(\tilde{X}_i(k)) P], \\ \mu_9 &= \text{vec}'[\text{vec}(\text{vec}(\tilde{X}_i(k) \tilde{A}'_i \tilde{A}'_j \tilde{\mathcal{E}}_j(\tilde{L}) \tilde{A}_j \tilde{B}_i) a) \text{vec}'(a'a)], \end{aligned}$$

$$\begin{aligned}
\mu_{10} &= \text{vec}'[\text{vec}(a' \text{vec}'(\tilde{B}'_i \tilde{A}'_j \tilde{\mathcal{E}}_j(\tilde{L}) \tilde{A}_j \tilde{A}_i \tilde{X}_i(k))) \text{vec}'(a'a)], \\
\mu_{11} &= \text{vec}'[\text{vec}(a'a) \text{vec}'(a' \text{vec}'(\tilde{B}'_j \tilde{\mathcal{E}}_j(\tilde{L}) \tilde{A}_j \tilde{A}_i \tilde{X}_i(k) \tilde{A}_i))], \\
\mu_{12} &= \text{vec}'[\text{vec}(a'a) \text{vec}'(\text{vec}(\tilde{A}_i \tilde{X}_i(k) \tilde{A}'_i \tilde{A}'_j \tilde{\mathcal{E}}_j(\tilde{L}) \tilde{B}_j) a)], \\
\mu_{13} &= \text{vec}'[P \text{vec}(\tilde{B}'_i \tilde{A}'_j \tilde{\mathcal{E}}_j(\tilde{L}) \tilde{A}_j \tilde{B}_i \tilde{X}_i(k)) \text{vec}'(a'a)], \\
\mu_{14} &= \text{vec}'[\text{vec}(a'a) \text{vec}'(\tilde{B}'_j \tilde{\mathcal{E}}_j(\tilde{L}) \tilde{B}_j \tilde{A}_i \tilde{X}_i(k) \tilde{A}'_i) P], \\
\mu_{15} &= \text{vec}'[\tilde{X}_i(k) \tilde{A}'_i \otimes a \tilde{B}'_i \tilde{A}'_j \tilde{\mathcal{E}}_j(\tilde{L}) \tilde{B}_j a], \\
\mu_{16} &= \text{vec}'[\tilde{X}_i(k) \tilde{A}'_i \tilde{A}'_j \tilde{\mathcal{E}}_j(\tilde{L}) \tilde{B}_j a \otimes a' \tilde{B}'_i], \\
\mu_{17} &= \text{vec}'[a' \tilde{B}'_i \otimes \tilde{X}_i(k) \tilde{A}'_i \tilde{A}'_j \tilde{\mathcal{E}}_j(\tilde{L}) \tilde{B}_j a], \\
\mu_{18} &= \text{vec}'[a' \tilde{B}'_i \tilde{A}'_j \tilde{\mathcal{E}}_j(\tilde{L}) \tilde{B}_j a \otimes \tilde{X}_i(k) \tilde{A}'_i], \\
\mu_{19} &= \text{vec}'[(\tilde{X}_i(k) \tilde{A}'_i \otimes a' \tilde{B}'_i \tilde{B}'_j \tilde{\mathcal{E}}_j(\tilde{L}) \tilde{B}_j) P], \\
\mu_{20} &= \text{vec}'[(a' \tilde{B}'_i \otimes \tilde{X}_i(k) \tilde{A}'_i \tilde{B}'_j \tilde{\mathcal{E}}_j(\tilde{L}) \tilde{B}_j) P], \\
\mu_{21} &= \text{vec}'[\text{vec}'(\tilde{B}'_i \otimes \tilde{B}'_i \tilde{A}'_j \tilde{\mathcal{E}}_j(\tilde{L}) \tilde{B}_j a) \otimes \text{vec}'(\tilde{X}_i(k)) P], \\
\mu_{22} &= \text{vec}'[\text{vec}(\tilde{B}'_i \tilde{A}'_j \tilde{\mathcal{E}}_j(\tilde{L}) \tilde{B}_j a' \otimes \tilde{B}'_i) \otimes \text{vec}'(\tilde{X}_i(k)) P], \\
\mu_{23} &= \text{vec}'[\text{vec}'((\tilde{B}'_i \otimes \tilde{B}'_i \tilde{B}'_j \tilde{\mathcal{E}}_j(\tilde{L}) \tilde{B}_j) P) \otimes \text{vec}'(\tilde{X}_i(k)) P], \\
\xi_1 &= \text{Tr}\{\tilde{X}_i(k) \tilde{C}_i\}, \\
\xi_2 &= \text{Tr}\{\tilde{X}_i(k) \tilde{A}'_i \tilde{C}_j \tilde{A}_i\} \quad e \\
\xi_3 &= \text{Tr}\{\tilde{X}(k) \tilde{A}' \tilde{A}' \tilde{\mathcal{E}}(\tilde{L}) \tilde{A} \tilde{A}\},
\end{aligned}$$

com $\tilde{L} = \tilde{L}(k+2)$ conhecido, P tal que $\text{vec}(b'b) = P \text{vec}(b' \otimes b')$ e $a = [0 \dots 0 \quad 1] \in \mathcal{M}^{1, n+1}$.

Demonstração: A prova está disponível no Apêndice A.

Observação 4.1 *A implementação computacional do método biquadrático não foi viável devido à pouca informação disponível acerca do algoritmo e pelas incompatibilidades entre às restrições dos problemas. De fato, a restrição gerada pela necessidade da última posição dos ganhos possuir um valor constante não se encaixa em nenhum modelo disponível na literatura e as restrições nos modelos disponíveis também não encaixam no problema PB_k . Apesar de um método biquadrático não ter sido implementado, temos uma boa estimativa (através do método Variacional Modificado) do número de iterações que seriam necessárias para resolver o problema em (2.18) usando o método Variacional Duplo com Biquadrático.*

O cálculo analítico dos coeficientes da equação biquadrática é computacionalmente custoso, por isso foi desenvolvido o Algoritmo 4.1, que apresenta as etapas para encontrar a representação biquadrática da função PB_k de maneira numérica, para qualquer sistema. Resumidamente a estratégia é montar um sistema linear $Ax = b$ onde as incógnitas x são os coeficiente biquadráticos a_{ijkl} a serem encontrados, a matriz dos coeficientes do sistema A é obtida através de ganhos aleatórios e o vetor b é formado pelos custos obtidos para cada ganho. Vale ressaltar que para a execução do Algoritmo 4.1 não é necessário nenhuma adaptação dimensional, conforme apresentada no Exemplo 4.1, às matrizes de coeficientes.

O Algoritmo 4.1, cujo código fonte está disponível no Apêndice C, foi implementado utilizando o software *MATLAB*[®] na versão 8.1. Os testes foram realizados em um computador com processador Intel Core i7 com velocidade de 3.4GHz, memória RAM de 64GB e sistema operacional *Windows*[®]7 em uma plataforma de 64 bits. Um total de 1.000 SLSM, gerados aleatoriamente de acordo com o *Gerador de SLSM* apresentado em BORTOLIN (2012) e SILVA (2012), foi utilizado e em todos os casos o Algoritmo 4.1 se mostrou robusto e as soluções encontradas descreveram um problema biquadrático.

Algoritmo 4.1 Passos para Encontrar a Função Biquadrática

Dados iniciais: Dados A, B, C, D e G as matrizes dos coeficientes do sistema (2.6); s, r o tamanho da matriz de ganho, com a sequência de ganhos $\mathbf{g} \in \mathcal{G}$ e de $X \in \mathbb{S}^{r_0}$ conhecidas; \mathbb{P} a matriz de probabilidade; e π o vetor de distribuição de probabilidade.

Passo 1: Para $m = 1, 2, \dots, (r \cdot s + 1)^4$;

Passo 1.1: Redefina $g(k)$ e $g(k + 1)$ randomicamente e atualize $X(k + 1)$ e $X(k + 2)$ tal que $X_i(k + 1) = \tau_{A+Bg(k)}(X_i(k)) + \Sigma_i(k)$ e $X_i(k + 2) = \tau_{A+Bg(k+1)}\tau_{A+Bg(k)}(X_i(k)) + \tau_{A+Bg(k+1)}(\Sigma_i(k)) + \Sigma_i(k + 1)$, onde τ e Σ são dados pelas Equações (2.9) e (2.10), respectivamente.

Passo 1.2: Monte um vetor de incógnitas c_f empilhando as variáveis a_{ijkl} , por exemplo, da seguinte forma: seja $\text{pos} = (i - 1)(r \cdot s + 1)^3 + (j - 1)(r \cdot s + 1)^2 + (k - 1)(r \cdot s + 1) + (l - 1)(r \cdot s + 1)^0 + 1$ a regra para determinar a posição em que a variável a_{ijkl} será alocada, para $i, j, k, l \in \{1, \dots, r \cdot s + 1\}$, logo $c_f(\text{pos}) = a_{ijkl}$.

Passo 1.3: Seja $v = [\text{vec}(g(k)) \quad 1]$ e $z = [\text{vec}(g(k + 1)) \quad 1]$, monte a m -ésima linha da matriz H , por exemplo, da seguinte forma: $H(m, \text{pos}) = v(i) \cdot z(j) \cdot v(k) \cdot z(l)$, para $i, j, k, l \in \{1, \dots, r \cdot s + 1\}$ e a m -ésima linha do vetor coluna $b(m, 1) = \mathcal{J}_N$.

Passo 2: Resolva $H \cdot c_f = b$.

Método Variacional com Atualização de Dois Ganhos

Conforme visto na Seção 2.4, o método Variacional proposto por DO VAL e BAŞAR (1999) calcula, a cada iteração η e para cada instante de tempo k , o ganho $g^\eta(k)$, $k \in \mathcal{K}$ que minimiza a função objetivo (2.24). Os conceitos necessários para adaptar este método para atualizar dois ganhos consecutivos são apresentados na Seção 5.1. Em seguida, na Seção 5.2 são descritos os algoritmos implementados: um algoritmo diretamente adaptado do método Variacional e dois métodos de descida clássicos: o método de Gradiente e o método de Newton.

5.1 Adaptação do Problema para o Cálculo de dois Ganhos Consecutivos

Nesta seção é apresentada a formulação matemática da adaptação do problema PQ_k para atualização simultânea de dois ganhos consecutivos, ou seja, PQ_k deixa de ter apenas uma partição de ganho, $g(k)$, e passa a ter duas, $g(k)$ e $g(k+1)$, levando a um problema aqui chamado de PB_k . A motivação para esta adaptação surge do fato de não existirem, na literatura, métodos que explorem situações intermediárias referente a quantidade de ganhos atualizados, ou seja, há apenas trabalhos que utilizam um dos dois tipos de métodos: o Variacional Clássico, onde somente um ganho é atualizado a cada iteração; métodos que atualizam todos os ganhos simultaneamente como em BORTOLIN (2012), SILVA (2012) e mais recentemente em VARGAS et al. (2014).

Como já apresentado na Capítulo 4, o problema PB_k pode ser escrito a partir da seguinte forma expandida da Equação (2.23):

$$\begin{aligned} \mathcal{J}_N = & \sum_{\ell=0}^{k-1} \langle X(\ell), C + g'(\ell)Dg(\ell) \rangle + \langle X(k), C + g'(k)Dg(k) \rangle \\ & + \langle X(k+1), C + g'(k+1)Dg(k+1) \rangle + \langle L(k+2), X(k+2) \rangle + V(k+2). \end{aligned} \quad (5.1)$$

Considerando os ganhos $g(0), \dots, g(k-1), g(k+2), \dots, g(N-1)$, as matrizes $X(k)$, $L(k+2)$ e o escalar $V(k+2)$ conhecidos na Equação (5.1), a formulação do PB_k é dada como:

$$PB_k : \quad \min_{g(k), g(k+1)} \mathcal{J}_N \quad (5.2)$$

sendo

$$\begin{aligned} \mathcal{J}_N = & \langle X(k), C + g'(k)Dg(k) \rangle + \langle X(k+1), C + g'(k+1)Dg(k+1) \rangle + \langle L(k+2), X(k+2) \rangle \\ & + \tilde{V}(k+2), \end{aligned}$$

com $X(k+2) = \tau_{A+Bg(k+1)}\tau_{A+Bg(k)}(X(k)) + \tau_{A+Bg(k+1)}(\Sigma(k)) + \Sigma(k+1)$ e $X(k+1) = \tau_{A+Bg(k)}(X(k)) + \Sigma(k)$.

Observação 5.1 *Note que esta estratégia pode ser facilmente estendida para o caso onde a atualização é feita em m ganhos consecutivos. Formalmente isso consiste em calcular $g(k), g(k+1), \dots, g(k+m-1)$ de modo que minimizem a seguinte forma expandida da Equação (2.24):*

$$\begin{aligned} \mathcal{J}_N = & \sum_{\ell=0}^{k-1} \langle X(\ell), C + g'(\ell)Dg(\ell) \rangle + \langle X(k), C + g'(k)Dg(k) \rangle \\ & + \langle X(k+1), C + g'(k+1)Dg(k+1) \rangle + \dots \\ & + \langle X(k+m-1), C + g'(k+m-1)Dg(k+m-1) \rangle \\ & + \langle L(k+m), X(k+m) \rangle + V(k+m). \end{aligned} \quad (5.3)$$

Nesta equação são conhecidos os ganhos $g(0), \dots, g(k-1), g(k+m), \dots, g(N-1)$, a matriz $L(k+m)$ e o escalar $V(k+m)$.

Por brevidade, o método Variacional para a atualização de dois ganhos consecutivos será denominado de método Variacional Duplo. O Algoritmo 5.1 apresenta os passos deste método e é fortemente baseado no Algoritmo 2.1, com as principais modificações nos passos 2 e 4, a fim de se adequar ao problema PB_k , e no passo 3, onde a atualização dos ganhos é feita de acordo com o algoritmo escolhido.

Algoritmo 5.1 Método Variacional Duplo

Passo 0: Caso N seja ímpar, acrescente um estágio terminal trivial em Φ_M (2.6) de forma que o novo N resulte par. Inicie o processo com $\eta = 0$ e escolha uma sequência de ganhos inicial $g^\eta = \{g^\eta(0), \dots, g^\eta(N-1)\}$.

Passo 1: Para todo $k \in \mathcal{K}$ encontre $X^\eta(k)$ tal que

$$\begin{cases} X_i^\eta(k+1) = \tau_{A+Bg^\eta(k)}(X_i^\eta(k)) + \Sigma_i(k), \\ X_i^\eta(0) = \pi_i(0)x(0)x'(0), \end{cases}$$

onde τ e Σ são dados pelas Equações (2.9) e (2.10), respectivamente.

Passo 2: Faça $\eta = \eta + 1$ e $k = N - 1$.

Passo 3.0: Faça $\zeta = 0$ e $g^\zeta(k) = g^\eta(k)$ e $g^\zeta(k+1) = g^\eta(k+1)$.

Passo 3.1: Calcule $g^{\zeta+1}(k)$ e $g^{\zeta+1}(k+1)$ do problema PB_k baseando-se em $g^\zeta(k)$ e $g^\zeta(k+1)$, respectivamente, utilizando um dos algoritmos deste capítulo.

Passo 3.2: Avalie o **Crítério de Parada:** Se $\|g^{\zeta+1}(k) - g^\zeta(k)\|/\|g^{\zeta+1}(k)\| < \varepsilon_1$ e $\|g^{\zeta+1}(k+1) - g^\zeta(k+1)\|/\|g^{\zeta+1}(k+1)\| < \varepsilon_1$ para ε_1 dado, o critério foi satisfeito então PARE. Senão, faça $\zeta = \zeta + 1$ e volte ao passo 3.1.

Passo 4: Faça $g^\eta(k) = g^\zeta(k)$ e $g^\eta(k+1) = g^\zeta(k+1)$ e atualize $X^\eta(k+1)$ e $X^\eta(k+2)$ como no passo 1. Faça $k = k - 2$ e volte ao passo 3.0 se $k \geq 0$.

Crítério de Parada Variacional: Se $\|\mathcal{J}_N(g^\eta) - \mathcal{J}_N(g^{\eta-1})\|/\|\mathcal{J}_N(g^\eta)\| < \varepsilon$ para ε dado, o critério foi satisfeito então PARE. Senão, volte ao passo 2.

5.2 Métodos Computacionais

Esta seção é destinada a apresentar os métodos implementados para resolver o problema PB_k , no passo 3 do Algoritmo 5.1: Variacional Modificado, Gradiente e Newton.

As próximas subseções estão divididas da seguinte maneira: na Subseção 5.2.1 é apresentada uma modificação do método Variacional para atualização de dois ganhos e na Subseção 5.2.2 os resultados acerca dos métodos de descida: os cálculos das derivadas de primeira e segunda ordem do problema PB_k em 5.2.2.1, e em seguida os métodos de Gradiente, em 5.2.2.2, e de Newton, em 5.2.2.3, ambos adaptados ao método Variacional Duplo.

5.2.1 Variacional Modificado

Além dos métodos de descida, a serem vistos ainda neste capítulo, foi implementada uma modificação no próprio método Variacional Clássico, onde são atualizados dois ganhos consecutivos, satisfazendo a condição de otimalidade dada pela Equação (2.25).

Formalmente isso é atualizar $g(k+1)$ satisfazendo a condição necessária de otimalidade dada pela Equação (2.25) e, em seguida, atualizar $g(k)$ da mesma maneira, de acordo com a

estratégia do método Variacional. Ao invés de continuar atualizando toda a sequência de ganhos restantes, o método volta a atualizar $g(k+1)$ e $g(k)$ até que os erros satisfaçam um critério de parada, dado pelo passo 3.2 do Algoritmo 5.1. Apenas quando esse critério for satisfeito o método continuará atualizando o restante dos ganhos, passando então para $g(k-1)$ e $g(k-2)$ e procedendo analogamente ao exposto acima.

Note que a atualização dos ganhos $g(k)$ e $g(k+1)$ é feita através do método Variacional Clássico, logo este algoritmo será denominado de método Variacional Duplo com Variacional, ou simplesmente método Variacional Modificado. O Algoritmo 5.2 apresenta a modificação necessária ao Algoritmo 5.1 para o método Variacional Modificado.

Algoritmo 5.2 Método Variacional Modificado

Passo 3.1: (método Variacional Clássico) Determine $g^{\zeta+1}(k)$ que minimize PQ_k através da Equação (2.25) com $L_i(k)$ dado pela Definição 2.4. Em seguida determine $g^{\zeta+1}(k+1)$ da mesma maneira.

O comportamento dos ganhos obtidos pelo Algoritmo 5.2, na maioria dos exemplos do Capítulo 6, quando analisados ao longo das iterações ζ apresentou um formato “caracol”, conforme pode ser visto pelas setas contínuas no exemplo da Figura 5.1. Note que é interessante buscar métodos que encontrem a sequência de ganhos ótimos de maneira mais direta e eficiente, que permitam “caminhar na diagonal” (seta tracejada). É com esse intuito que os métodos de descida foram propostos.

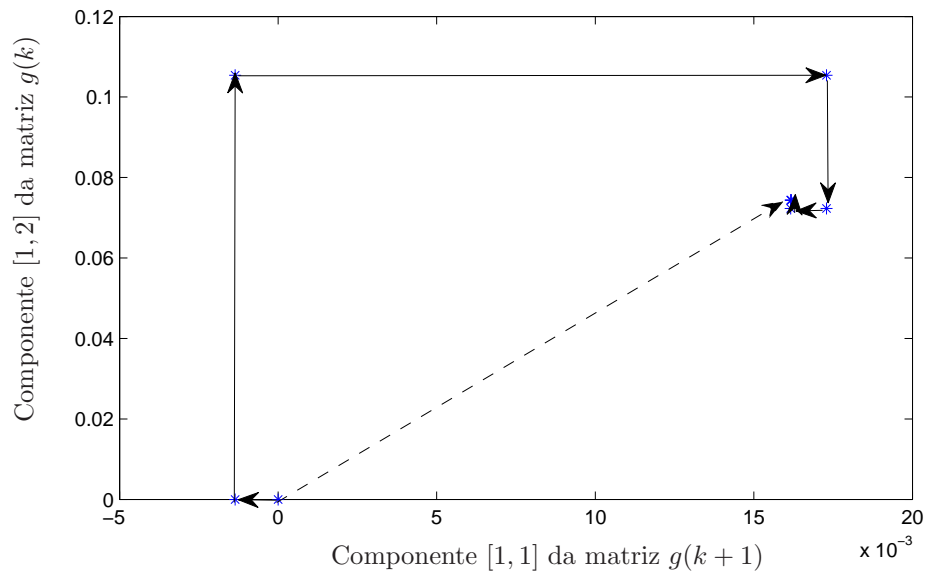


Figura 5.1: Comportamento da atualização dos ganhos quando utilizado o método Variacional Modificado (setas contínuas), presente na maioria dos exemplos do Capítulo 6, e comportamento desejado (seta tracejada).

5.2.2 Métodos de Descida

Os métodos de Newton e Gradiente foram escolhidos por serem métodos clássicos para otimização de funções objetivo não lineares e também por existir na literatura autores que obtiveram bons resultados em problemas gerais utilizando estes métodos. Podemos citar BORTOLIN (2012), onde os métodos de Newton e Gradiente são utilizados para encontrar todos os ganhos do problema de otimização (2.18).

Os métodos de descida tratados neste capítulo buscam minimizar PB_k dado pela Equação (5.2) a partir de um ponto inicial dado $g^0(k)$, $k \in \mathcal{K}$, uma direção de busca d_ζ e um tamanho de passo $\alpha_\zeta > 0$ ao longo das iterações ζ . A sequência de pontos factíveis é obtida atualizando $g^{\zeta+1}(k) = g^\zeta(k) + \alpha_\zeta d_\zeta$. Este procedimento é repetido até que um critério de parada previamente definido seja satisfeito.

Para o cálculo da direção de busca d_ζ do método de Gradiente será utilizada a primeira derivada da Equação (5.2) e o passo α_ζ a ser dado nesta direção será determinado pelo mínimo da função $\mathcal{J}_N(\alpha) = \mathcal{J}_N(\mathbf{g}^\zeta + \alpha d_\zeta)$ na variável α . Para a direção do método de Newton serão utilizadas as derivadas de primeira e segunda ordem da Equação (5.2) e o tamanho do passo α_ζ será considerado igual a 1 em todas as iterações do algoritmo. Por questão de simplicidade, o índice ζ deste passo será suprimido.

5.2.2.1 Cálculo das Derivadas de Primeira e Segunda Ordem

Para a implementação computacional dos métodos de descida, detalhados a seguir, são necessários os cálculos das derivadas de primeira e segunda ordem da função (5.2). Para isso serão imprescindíveis algumas regras para derivação de matrizes.

Para a derivada parcial do traço de uma matriz com respeito a uma matriz real são necessárias as seguintes regras, que podem ser encontradas em JOHO (2000), FACKLER (2005) e LEE (2009).

$$\begin{aligned} \frac{\partial \text{Tr}\{MP\}}{\partial P} &= M', & \frac{\partial \text{Tr}\{MPN\}}{\partial P} &= M'N', & \frac{\partial \text{Tr}\{MP'N\}}{\partial P} &= NM, \\ \frac{\partial \text{Tr}\{MP'NP\}}{\partial P} &= N'PM' + NPM & \text{e} & & \frac{\partial \text{Tr}\{MP'NPO\}}{\partial P} &= N'PM'O' + NPCM. \end{aligned} \quad (5.4)$$

Teorema 5.1 *Considere PB_k dado pela Equação (5.2). A derivada de primeira ordem*

$$\nabla \mathcal{J}_N(g(k), g(k+1)) = \left[\frac{\partial \mathcal{J}_N}{\partial g(k)}, \frac{\partial \mathcal{J}_N}{\partial g(k+1)} \right] \quad (5.5)$$

pode ser calculada por:

$$\begin{aligned}
\frac{\partial \mathcal{J}_N}{\partial g(k)} &= \sum_{i=1}^T \sum_{j=1}^T p_{ij} (B'_i \mathcal{E}_i(C) A_i + B'_i A'_j \mathcal{E}_j(L) A_j A_i + D_i g(k) + B'_i \mathcal{E}_i(C) B_i g(k) \\
&\quad + B'_i g'(k+1) D_j g(k+1) A_i + B'_i g'(k+1) B'_j \mathcal{E}_j(L) A_j A_i \\
&\quad + B'_i A'_j \mathcal{E}_j(L) B_j g(k+1) A_i + B'_i g'(k+1) B'_j \mathcal{E}_j(L) B_j g(k+1) A_i \\
&\quad + B'_i g'(k+1) D_j g(k+1) B_i g(k) + B'_i A'_j \mathcal{E}_j(L) A_j B_i g(k) \\
&\quad + B'_i A'_j \mathcal{E}_j(L) B_j g(k+1) B_i g(k) + B'_i g'(k+1) B'_j \mathcal{E}_j(L) A_j B_i g(k) \\
&\quad + B'_i g'(k+1) B'_j \mathcal{E}_j(L) B_j g(k+1) B_i g(k)) X_i(k),
\end{aligned} \tag{5.6}$$

$$\begin{aligned}
\frac{\partial \mathcal{J}_N}{\partial g(k+1)} &= \sum_{i=1}^T \sum_{j=1}^T p_{ij} (D_j g(k+1) A_i X_i(k) A'_i + B'_j \mathcal{E}_j(L) A_j A_i X_i(k) A'_i \\
&\quad + D_j g(k+1) B_i g(k) X_i(k) A'_i + B'_j \mathcal{E}_j(L) B_j g(k+1) A_i X_i(k) A'_i \\
&\quad + B'_j \mathcal{E}_j(L) A_j B_i g(k) X_i(k) A'_i + B'_j \mathcal{E}_j(L) B_j g(k+1) B_i g(k) X_i(k) g'(k) B'_i \\
&\quad + B'_j \mathcal{E}_j(L) B_j g(k+1) B_i g(k) X_i(k) A'_i + B'_j \mathcal{E}_j(L) A_j A_i X_i(k) g'(k) B'_i \\
&\quad + B'_j \mathcal{E}_j(L) A_j B_i g(k) X_i(k) g'(k) B'_i + B'_j \mathcal{E}_j(L) B_j g(k+1) A_i X_i(k) g'(k) B'_i \\
&\quad + D_j g(k+1) B_i g(k) X_i(k) g'(k) B'_i + D_j g(k+1) A_i X_i(k) g'(k) B'_i),
\end{aligned} \tag{5.7}$$

onde $L = L(k+2)$ é conhecido.

Demonstração: Pela Proposição 2.2 temos que minimizar \mathcal{J}_N é equivalente a minimizar $\langle X(k), L(k) \rangle$, portanto nossa prova se resumirá em desenvolver $\langle X(k), L(k) \rangle$ a fim de obtermos apenas termos conhecidos e as incógnitas do problema, $g(k)$ e $g(k+1)$. Para isso serão utilizados os operadores \mathcal{E} e \mathcal{L} dados pelas Equações (2.7) e (2.8), respectivamente, e a matriz L dada por (2.19).

$$\begin{aligned}
\langle X(k), L(k) \rangle &= \sum_{i=1}^T \text{Tr}\{X_i(k) L_i(k)\} = \sum_{i=1}^T \text{Tr}\{X_i(k) [C_i + g'(k) D_i g(k) + \mathcal{L}_i(L(k+1))]\} \\
&= \sum_{i=1}^T \text{Tr}\{X_i(k) [C_i + g'(k) D_i g(k) + (A_i + B_i g(k))' \mathcal{E}_i(L(k+1)) (A_i + B_i g(k))]\} \\
&= \sum_{i=1}^T \text{Tr}\{X_i(k) [C_i + g'(k) D_i g(k) + (A_i + B_i g(k))' \mathcal{E}_i(C + g'(k+1) D g(k+1) \\
&\quad + \mathcal{L}(L(k+2))) (A_i + B_i g(k))]\} \\
&= \sum_{i=1}^T \text{Tr}\{X_i(k) [C_i + g'(k) D_i g(k) + (A_i + B_i g(k))' \mathcal{E}_i[C + g'(k+1) D g(k+1) \\
&\quad + (A + B g(k+1))' \mathcal{E}(L(k+2)) (A + B g(k+1))] (A_i + B_i g(k))]\}.
\end{aligned} \tag{5.8}$$

Note que há uma variável de interesse dentro do operador \mathcal{E}_i , logo é necessário abrir esse termo seguindo a Equação (2.7). Por fim, depois de realizadas todas as distributivas existentes na Equação (5.8), obtemos a seguinte forma equivalente:

$$\begin{aligned}
& \sum_{i=1}^T \sum_{j=1}^T p_{ij} \text{Tr}\{X_i(k)[C_i + g'(k)D_i g(k) + A'_i \mathcal{E}_i(C)A_i + A'_i \mathcal{E}_i(C)B_i g(k) + g'(k)B'_i \mathcal{E}_i(C)A_i \\
& + g'(k)B'_i \mathcal{E}_i(C)B_i g(k) + A'_i g'(k+1)D_j g(k+1)A_i + A'_i g'(k+1)D_j g(k+1)B_i g(k) \\
& + A'_i A'_j \mathcal{E}_j(L)A_j A_i + A'_i A'_j \mathcal{E}_j(L)B_j g(k+1)A_i + A'_i g'(k+1)B'_j \mathcal{E}_j(L)A_j A_i \\
& + A'_i g'(k+1)B'_j \mathcal{E}_j(L)B_j g(k+1)A_i + A'_i A'_j \mathcal{E}_j(L)A_j B_i g(k) + A'_i A'_j \mathcal{E}_j(L)B_j g(k+1)B_i g(k) \\
& + A'_i g'(k+1)B'_j \mathcal{E}_j(L)A_j B_i g(k) + A'_i g'(k+1)B'_j \mathcal{E}_j(L)B_j g(k+1)B_i g(k) \\
& + g'(k)B'_i g'(k+1)D_j g(k+1)A_i + g'(k)B'_i g'(k+1)D_j g(k+1)B_i g(k) \\
& + g'(k)B'_i A'_j \mathcal{E}_j(L)A_j B_i g(k) + g'(k)B'_i A'_j \mathcal{E}_j(L)B_j g(k+1)B_i g(k) \\
& + g'(k)B'_i g'(k+1)B'_j \mathcal{E}_j(L)A_j B_i g(k) + g'(k)B'_i g'(k+1)B'_j \mathcal{E}_j(L)B_j g(k+1)B_i g(k) \\
& + g'(k)B'_i A'_j \mathcal{E}_j(L)A_j A_i + g'(k)B'_i A'_j \mathcal{E}_j(L)B_j g(k+1)A_i \\
& + g'(k)B'_i g'(k+1)B'_j \mathcal{E}_j(L)A_j A_i + g'(k)B'_i g'(k+1)B'_j \mathcal{E}_j(L)B_j g(k+1)A_i]\},
\end{aligned} \tag{5.9}$$

onde $L = L(k+2)$ é conhecido. Derivando cada parcela da Equação (5.9) em relação a $g(k)$, em que as demais variáveis são consideradas conhecidas, utilizando as regras descritas nas Equações (5.4), temos:

$$\begin{aligned}
& \frac{\partial \text{Tr}\{X_i(k)C_i\}}{\partial g(k)} = 0, \\
& \frac{\partial \text{Tr}\{X_i(k)g'(m)D_i g(k)\}}{\partial g(k)} = D'_i g(k)X'_i(k) + D_i g(k)X_i(k) = 2(D_i g(k)X_i(k)), \\
& \frac{\partial \text{Tr}\{X_i(k)A'_i \mathcal{E}_i(C)A_i\}}{\partial g(k)} = 0, \\
& \frac{\partial \text{Tr}\{X_i(k)A'_i \mathcal{E}_i(C)B_i g(k)\}}{\partial g(k)} = B'_i \mathcal{E}'_i(C)A_i X'_i(k) = B'_i \mathcal{E}_i(C)A_i X_i(k), \\
& \frac{\partial \text{Tr}\{X_i(k)g'(k)B'_i \mathcal{E}_i(C)A_i\}}{\partial g(k)} = B'_i \mathcal{E}_i(C)A_i X_i(k), \\
& \frac{\partial \text{Tr}\{X_i(k)g'(k)B'_i \mathcal{E}_i(C)B_i g(k)\}}{\partial g(k)} = B'_i \mathcal{E}'_i(C)B_i g(k)X'_i(k) + B'_i \mathcal{E}_i(C)B_i g(k)X_i(k) \\
& = 2(B'_i \mathcal{E}_i(C)B_i g(k)X_i(k)), \\
& \frac{\partial \text{Tr}\{X_i(k)A'_i g'(k+1)D_j g(k+1)A_i\}}{\partial g(k)} = 0, \\
& \frac{\partial \text{Tr}\{X_i(k)A'_i g'(k+1)D_j g(k+1)B_i g(k)\}}{\partial g(k)} = B'_i g'(k+1)D'_j g(k+1)A_i \\
& = B'_i g'(k+1)D_j g(k+1)A_i,
\end{aligned}$$

⋮

Continuando desta forma chegamos a:

$$\begin{aligned} \frac{\partial \mathcal{J}_N}{\partial g(k)} = & \sum_{i=1}^T \sum_{j=1}^T p_{ij} [B'_i \mathcal{E}_i(C) A_i + B'_i A'_j \mathcal{E}_j(L) A_j A_i + D_i g(k) + B'_i \mathcal{E}_i(C) B_i g(k) \\ & + B'_i g'(k+1) D_j g(k+1) A_i + B'_i g'(k+1) B'_j \mathcal{E}_j(L) A_j A_i + B'_i A'_j \mathcal{E}_j(L) B_j g(k+1) A_i \\ & + B'_i g'(k+1) B'_j \mathcal{E}_j(L) B_j g(k+1) A_i + B'_i g'(k+1) D_j g(k+1) B_i g(k) \\ & + B'_i A'_j \mathcal{E}_j(L) A_j B_i g(k) + B'_i A'_j \mathcal{E}_j(L) B_j g(k+1) B_i g(k) \\ & + B'_i g'(k+1) B'_j \mathcal{E}_j(L) A_j B_i g(k) + B'_i g'(k+1) B'_j \mathcal{E}_j(L) B_j g(k+1) B_i g(k)] X(k). \end{aligned}$$

Analogamente, derivando a Equação (5.9) em relação a $g(k+1)$, obtemos:

$$\begin{aligned} \frac{\partial \mathcal{J}_N}{\partial g(k+1)} = & \sum_{i=1}^T \sum_{j=1}^T p_{ij} (D_j g(k+1) A_i X_i(k) A'_i + D_j g(k+1) B_i g(k) X_i(k) A'_i \\ & + B'_j \mathcal{E}_j(L) A_j A_i X_i(k) A'_i + B'_j \mathcal{E}_j(L) B_j g(k+1) A_i X_i(k) A'_i \\ & + B'_j \mathcal{E}_j(L) A_j B_i g(k) X_i(k) A'_i + B'_j \mathcal{E}_j(L) B_j g(k+1) B_i g(k) X_i(k) A'_i \\ & + D_j g(k+1) A_i X_i(k) g'(k) B'_i + B'_j \mathcal{E}_j(L) A_j A_i X_i(k) g'(k) B'_i \\ & + B'_j \mathcal{E}_j(L) B_j g(k+1) A_i X_i(k) g'(k) B'_i + D_j g(k+1) B_i g(k) X_i(k) g'(k) B'_i \\ & + B'_j \mathcal{E}_j(L) A_j B_i g(k) X_i(k) g'(k) B'_i + B'_j \mathcal{E}_j(L) B_j g(k+1) B_i g(k) X_i(k) g'(k) B'_i). \end{aligned}$$

■

Para a demonstração da matriz Hessiana serão necessárias algumas regras de derivação de matriz em relação a matriz. Algumas destas regras não foram encontradas na literatura, por isso foi necessário estender regras existentes a fim de adequá-las as nossas necessidades. Vale ressaltar que tais regras talvez já existam, porém não foram encontradas, provavelmente por questões de notação, mesmo assim serão destacadas a seguir, mas suas demonstrações serão omitidas.

Lema 5.1

$$\begin{aligned} \frac{\partial MP'N}{\partial P} &= (N' \otimes M)T, & \frac{\partial MP'NPO}{\partial P} &= (O'P'N' \otimes M)T + (O' \otimes MP'N) \quad e \\ \frac{\partial MPNP'O}{\partial P} &= (O'PN' \otimes M) + (O' \otimes MPN)T, \end{aligned} \quad (5.10)$$

onde M, P, N, O e T são matrizes, e $Tvec(P) = vec(P')$.

Além destas, será usada também $\frac{\partial MPN}{\partial P} = (N' \otimes M)$, que pode ser encontrada em FACKLER (2005).

Teorema 5.2 Considere PB_k dado pela Equação (5.2). A derivada de segunda ordem

$$H_{\mathcal{J}_N}(g(k), g(k+1)) = \begin{bmatrix} \frac{\partial^2 \mathcal{J}_N}{\partial^2 g(k)} & \frac{\partial^2 \mathcal{J}_N}{\partial g(k+1)g(k)} \\ \frac{\partial^2 \mathcal{J}_N}{\partial g(k)g(k+1)} & \frac{\partial^2 \mathcal{J}_N}{\partial^2 g(k+1)} \end{bmatrix} \quad (5.11)$$

pode ser calculada por:

$$\begin{aligned} \frac{\partial^2 \mathcal{J}_N}{\partial^2 g(k)} &= \sum_{i=1}^T \sum_{j=1}^T p_{ij} (X_i \otimes D_i + X_i \otimes B'_i \mathcal{E}_i(C) B_i + X_i \otimes B'_i g'(k+1) D_j g(k+1) B_i \\ &+ X_i \otimes B'_i A'_j \mathcal{E}_j(L) A_j B_i + X_i \otimes B'_i A'_j \mathcal{E}_j(L) B_j g(k+1) B_i \\ &+ X_i \otimes B'_i g'(k+1) B'_j \mathcal{E}_j(L) A_j B_i + X_i \otimes B'_i g'(k+1) B'_j \mathcal{E}_j(L) B_j g(k+1) B_i), \end{aligned} \quad (5.12)$$

$$\begin{aligned} \frac{\partial^2 \mathcal{J}_N}{\partial g(k)g(k+1)} &= \sum_{i=1}^T \sum_{j=1}^T p_{ij} ((X_i A'_i g'(k+1) D_j \otimes B'_i) T + X_i A'_i \otimes B'_i g'(k+1) D_j \\ &+ (X_i A'_i A'_j \mathcal{E}_j(L) B_j \otimes B'_i) T + (X_i A'_i g'(k+1) B'_j \mathcal{E}_j(L) B_j \otimes B'_i) T \\ &+ X_i A'_i \otimes B'_i A'_j \mathcal{E}_j(L) B_j + X_i A'_i \otimes B'_i g'(k+1) B'_j \mathcal{E}_j(L) B_j \\ &+ (X_i g'(k) B'_i g'(k+1) D_j \otimes B'_i) T + X_i g'(k) B'_i \otimes B'_i g'(k+1) D_j \\ &+ X_i g'(k) B'_i \otimes B'_i A'_j \mathcal{E}_j(L) B_j + (X_i g'(k) B'_i A'_j \mathcal{E}_j(L) B_j \otimes B'_i) T \\ &+ (X_i g'(k) B'_i g'(k+1) B'_j \mathcal{E}_j(L) B_j \otimes B'_i) T \\ &+ X_i g'(k) B'_i \otimes B'_i g'(k+1) B'_j \mathcal{E}_j(L) B_j), \end{aligned} \quad (5.13)$$

$$\begin{aligned} \frac{\partial^2 \mathcal{J}_N}{\partial g(k+1)g(k)} &= \sum_{i=1}^T \sum_{j=1}^T p_{ij} (A_i X_i \otimes D_j g(k+1) B_i + A_i X_i \otimes B'_j \mathcal{E}_j(L) A_j B_i \\ &+ (B_i \otimes B'_j \mathcal{E}_j(L) A_j A_i X_i) T + A_i X_i \otimes B'_j \mathcal{E}_j(L) B_j g(k+1) B_i \\ &+ (B_i \otimes D_j g(k+1) A_i X_i) T + (B_i \otimes B'_j \mathcal{E}_j(L) B_j g(k+1) A_i X_i) T \\ &+ (B_i \otimes D_j g(k+1) B_i g(k) X_i) T + (B_i \otimes B'_j \mathcal{E}_j(L) A_j B_i g(k) X_i) T \\ &+ B_i g(k) X_i \otimes B'_j \mathcal{E}_j(L) A_j B_i + B_i g(k) X_i \otimes B'_j \mathcal{E}_j(L) B_j g(k+1) B_i \\ &+ (B_i \otimes B'_j \mathcal{E}_j(L) B_j g(k+1) B_i g(k) X_i) T), \end{aligned} \quad (5.14)$$

$$\begin{aligned} \frac{\partial^2 \mathcal{J}_N}{\partial^2 g(k+1)} &= \sum_{i=1}^T \sum_{j=1}^T p_{ij} (A_i X_i A_i \otimes D_j + A_i X_i g'(k) B'_i \otimes D_j + A_i X_i A'_i \otimes B'_j \mathcal{E}_j(L) B_j \\ &+ B_i g(k) X_i B'_i \otimes D_j + A_i X_i g'(k) B'_i \otimes B'_j \mathcal{E}_j(L) B_j + B_i g(k) X_i A'_i \otimes D_j \\ &+ B_i g(k) X_i A'_i \otimes B'_j \mathcal{E}_j(L) B_j + B_i g(k) X_i g'(k) B'_i \otimes B'_j \mathcal{E}_j(L) B_j), \end{aligned} \quad (5.15)$$

sendo T tal que $T \text{vec}(g) = \text{vec}(g')$, e $L = L(k+2)$ conhecido.

Demonstração: Para essa demonstração é necessário apenas a aplicação das regras de derivação de matrizes em relação a uma matriz descritas em (5.10) e $\frac{\partial MPN}{\partial P} = (N' \otimes M)$. A cada derivada parcial basta considerar um dos ganhos conhecidos, de forma que temos a derivada de uma matriz em relação a uma matriz. A seguir será detalhado $\frac{\partial^2 \mathcal{J}_N}{\partial^2 g(k)}$, porém os demais termos da matriz Hessiana são obtidos com raciocínio análogo. Derivando a Equação (5.6) em relação a $g(k)$, em que as demais variáveis são consideradas conhecidas, utilizando as regras já descritas obtemos:

$$\begin{aligned} \frac{\partial B'_i \mathcal{E}_i(C) A_i X_i(k)}{\partial g(k)} &= 0, \\ \frac{\partial B'_i A'_j \mathcal{E}_j(L) A_j A_i X_i(k)}{\partial g(k)} &= 0, \\ \frac{\partial D_i g_i(k) X_i(k)}{\partial g(k)} &= X'_i \otimes D_i = X_i \otimes D_i, \\ \frac{\partial B'_i \mathcal{E}_i(C) B_i g(k) X_i(k)}{\partial g(k)} &= X_i \otimes B'_i \mathcal{E}_i(C) B_i, \\ \frac{\partial B'_i g'(k+1) D_j g(k+1) A_i X_i(k)}{\partial g(k)} &= 0, \\ \frac{\partial B'_i g'(k+1) B'_j \mathcal{E}_j(L) A_j A_i X_i(k)}{\partial g(k)} &= 0, \\ \frac{\partial B'_i A'_j \mathcal{E}_j(L) B_j g(k+1) A_i X_i(k)}{\partial g(k)} &= 0, \\ \frac{\partial B'_i g'(k+1) B'_j \mathcal{E}_j(L) B_j g(k+1) A_i X_i(k)}{\partial g(k)} &= 0, \\ \frac{\partial B'_i g'(k+1) D_j g(k+1) B_i g(k) X_i(k)}{\partial g(k)} &= X_i \otimes B'_i g'(k+1) D_j g(k+1) B_i, \\ \frac{\partial B'_i A'_j \mathcal{E}_j(L) A_j B_i g(k) X_i(k)}{\partial g(k)} &= X_i \otimes B'_i A'_j \mathcal{E}_j(L) A_j B_i, \\ \frac{\partial B'_i A'_j \mathcal{E}_j(L) B_j g(k+1) B_i g(k) X_i(k)}{\partial g(k)} &= X_i \otimes B'_i A'_j \mathcal{E}_j(L) B_j g(k+1) B_i, \\ \frac{\partial B'_i g'(k+1) B'_j \mathcal{E}_j(L) B_j g(k+1) B_i g(k) X_i(k)}{\partial g(k)} &= X_i \otimes B'_i g'(k+1) B'_j \mathcal{E}_j(L) B_j g(k+1) B_i, \\ \frac{\partial B'_i g'(k+1) B'_j \mathcal{E}_j(L) A_j B_i g(k) X_i(k)}{\partial g(k)} &= X_i \otimes B'_i g'(k+1) B'_j \mathcal{E}_j(L) A_j B_i. \end{aligned}$$

Assim,

$$\begin{aligned} \frac{\partial^2 \mathcal{J}_N}{\partial^2 g(k)} &= \sum_{i=1}^T \sum_{j=1}^T p_{ij} (X_i \otimes D_i + X_i \otimes B'_i \mathcal{E}_i(C) B_i + X_i \otimes B'_i g'(k+1) D_j g(k+1) B_i \\ &\quad + X_i \otimes B'_i A'_j \mathcal{E}_j(L) A_j B_i + X_i \otimes B'_i A'_j \mathcal{E}_j(L) B_j g(k+1) B_i \\ &\quad + X_i \otimes B'_i g'(k+1) B'_j \mathcal{E}_j(L) A_j B_i + X_i \otimes B'_i g'(k+1) B'_j \mathcal{E}_j(L) B_j g(k+1) B_i). \end{aligned}$$

■

Com estes resultados foi possível implementar computacionalmente os principais métodos de descida, Newton e Gradiente, adaptados ao método Variacional Duplo, como pode ser visto nos Algoritmos 5.3 e 5.4 adiante.

5.2.2.2 Variacional Duplo com Gradiente

O método de Gradiente adaptado ao método Variacional Duplo é um método iterativo de busca linear para os ganhos $g(k)$ e $g(k+1)$ do problema PB_k (passo 3.1 do Algoritmo 5.1), que usa como direção de descida

$$d_\zeta(g^\zeta(k), g^\zeta(k+1)) = -\nabla \mathcal{J}_N(g^\zeta(k), g^\zeta(k+1)),$$

sendo $\nabla \mathcal{J}_N(g^\zeta(k), g^\zeta(k+1))$ o vetor gradiente da função \mathcal{J}_N do PB_k , dado pela Equação (5.2), aplicado a $(g^\zeta(k), g^\zeta(k+1))$ na iteração ζ . A sequência de ganhos é atualizada por

$$(g^{\zeta+1}(k), g^{\zeta+1}(k+1)) = (g^\zeta(k), g^\zeta(k+1)) + \alpha_\zeta d_\zeta(g^\zeta(k), g^\zeta(k+1)), \quad (5.16)$$

em que o passo $\alpha_\zeta \geq 0$ é encontrado minimizando $\mathcal{J}_N[(g^\zeta(k), g^\zeta(k+1)) + \alpha d_\zeta(g^\zeta(k), g^\zeta(k+1))]$, um problema unidimensional na variável α .

O Algoritmo 5.3 apresenta a modificação necessária ao Algoritmo 5.1 para adaptação do método de Gradiente ao método Variacional Duplo.

Algoritmo 5.3 Método Variacional Duplo com Gradiente

Passo 3.1: (método de Gradiente) Calcule $\nabla \mathcal{J}_N(g^\zeta(k), g^\zeta(k+1))$ através de (5.5).

Passo 3.1.1: Faça $d_\zeta(g^\zeta(k), g^\zeta(k+1)) = -\nabla \mathcal{J}_N(g^\zeta(k), g^\zeta(k+1))$.

Passo 3.1.2: Calcule o tamanho do passo α_ζ de modo que $\min_\alpha \mathcal{J}_N$.

Passo 3.1.3: Faça $(g^{\zeta+1}(k), g^{\zeta+1}(k+1)) = (g^\zeta(k), g^\zeta(k+1)) + \alpha_\zeta d_\zeta(g^\zeta(k), g^\zeta(k+1))$.

5.2.2.3 Variacional Duplo com Newton

O método de Newton para o problema PB_k , dado pela Equação (5.2), consiste em aproximar a função $\mathcal{J}_N(g)$ por uma função quadrática $Q(g)$, em torno de um ponto dado $g = (g^\zeta(k), g^\zeta(k+1))$, até que um critério de parada previamente definido seja satisfeito.

A direção de busca do método de Newton, a cada iteração ζ , é dada pelo sistema linear $H(g^\zeta) \cdot d_\zeta = -\nabla \mathcal{J}_N(g^\zeta)$ e a sequência de pontos é obtida por $g^{\zeta+1} = g^\zeta + d_\zeta$.

Seja g^* um mínimo local, assumindo que $\nabla \mathcal{J}_N(g^*) = 0$, $H(g^*)$ é definida positiva e \mathcal{J}_N possui derivadas parciais de primeira e segunda ordem contínuas, temos que $H(g^\zeta)$ é definida positiva nos pontos próximos a g^* e, assim, o ponto sucessor $g^{\zeta+1}$ é bem definido.

O Algoritmo 5.4 apresenta a modificação necessária ao Algoritmo 5.1 para a adaptação do método de Newton ao método Variacional Duplo e que resulta numa sequência de ganhos \mathbf{g}^* que minimizam localmente o custo \mathcal{J}_N .

Algoritmo 5.4 Método Variacional Duplo com Newton

Passo 3.1: (método de Newton) Calcule $\nabla \mathcal{J}_N(g^\zeta(k), g^\zeta(k+1))$ e $H_{\mathcal{J}_N}(g^\zeta(k), g^\zeta(k+1))$ via (5.5) e (5.11), respectivamente.

Passo 3.1.1: Resolva o sistema linear:

$$H_{\mathcal{J}_N}(g^\zeta(k), g^\zeta(k+1)) \cdot d_\zeta(g^\zeta(k), g^\zeta(k+1)) = -\nabla \mathcal{J}_N(g^\zeta(k), g^\zeta(k+1)).$$

Passo 3.1.2: Determine o novo ganho $(g^{\zeta+1}(k), g^{\zeta+1}(k+1)) = (g^\zeta(k), g^\zeta(k+1)) + d_\zeta(g^\zeta(k), g^\zeta(k+1))$.

Resultados Computacionais

Este capítulo é destinado a apresentar as avaliações e comparações de desempenho entre os métodos implementados e aplicados ao problema de otimização dado pela Equação (2.18). Para facilitar a leitura e interpretação dos resultados obtidos, como um todo, o capítulo é iniciado antecipando as principais conclusões.

- O método Variacional Clássico alcança os melhores resultados em termos de custo e de tempo de CPU na grande maioria dos exemplos;
- O método Variacional Modificado apresenta menor número de iterações em uma parcela significativa de exemplos, contudo gasta mais tempo, justificando a busca por métodos mais eficiente para o problema PB_k ;
- Em termos gerais, os métodos de descida, utilizados neste trabalho com intuito de diminuir o tempo computacional, não apresentaram bons resultados. Há uma grande quantidade de exemplos cujos resultados (tempo, número de iteração e custo) obtidos através do Variacional Modificado são melhores do que aqueles obtidos pelo Variacional Duplo com Newton ou Gradiente.

Todos algoritmos citados nas seções anteriores desta dissertação foram implementados utilizando o software *MATLAB*[®] na versão 8.1 e os testes foram realizados em um computador com processador Intel Core i7 com velocidade de 3.4GHz, memória RAM de 64GB e sistema operacional *Windows*[®]7 em uma plataforma de 64 bits.

Foram utilizados dois conjuntos de exemplos durante os testes, ambos gerados aleatoriamente com o *Gerador de SLSM* apresentado em BORTOLIN (2012) e SILVA (2012). Este gerador

possui um parâmetro α que varia entre 1 e 10, relacionado com o “grau de MS-estabilizabilidade” do SLSM, que pode ser ajustado de forma que valores baixos de alfa, como $\alpha = 1$, significam sistemas *bastante MS-estabilizáveis* (geralmente levando a custos mais baixos) e valores altos de alfa, como $\alpha = 10$, significam sistemas *pouco MS-estabilizáveis*.

O **conjunto A** possui 1.000 exemplos igualmente distribuídos dentro dos níveis α , ou seja, 100 exemplos para cada α do gerador. Analisando o comportamento dos números de iterações obtidos com este conjunto em diferentes intervalos de custos, foi possível notar que a estratégia de atualização de dois ganhos não é eficiente para exemplos com custos “médios” (aqui considerados entre 10^{50} e 10^{100}) e “altos” (entre 10^{100} e 10^{150}). A fim de explorar melhor o cenário em que a atualização dupla dos ganhos se mostra competitiva, um segundo conjunto de exemplos, chamado de **conjunto B**, foi definido. Para este conjunto foram gerados, da mesma forma, 10.000 exemplos igualmente distribuídos dentro dos níveis α e selecionados, aleatoriamente, 1.000 exemplos com custo igual ou inferior a 10^{50} (custo “baixo”). Vale ressaltar que neste conjunto os exemplos não estão mais igualmente distribuídos entre os vários níveis de *MS-estabilidade* (veja Tabela 6.1).

α	Nº de Exemplos no Conj. A	Nº de Exemplos no Conj. B
1	100	61
2	100	105
3	100	117
4	100	135
5	100	126
6	100	131
7	100	123
8	100	90
9	100	65
10	100	47

Tabela 6.1: Caracterização dos conjuntos de exemplos de SLSM.

Observação 6.1 Além dos conjuntos A e B , formados por sistemas com cadeias aperiódicas, veja ÇINLAR (1975), foi considerado um terceiro conjunto, C , composto por sistemas com cadeias periódicas de período 2, contudo os resultados foram muito similares àqueles obtidos pelos conjuntos A e B , por isso serão omitidos.

O critério de parada é composto por 3 decisões. Primeiro, se $\|X(k)\| > 10^{150}$ para algum valor de $k \in \mathcal{K}$ e/ou $\|\nabla \mathcal{J}_N\| > 10^{150}$ o algoritmo é interrompido. Segundo, o número máximo de iterações ficará definido em 500. Terceiro, a fim de analisar o comportamento das soluções em diferentes precisões foram considerados $\varepsilon = 10^{-8}$ e $\varepsilon = 10^{-12}$ para Variacional Duplo, ou seja para o critério de parada no final do horizonte; e $\varepsilon_1 = 10^{-4}$ e $\varepsilon_1 = 10^{-8}$ para o critério de parada do subproblema PB_k , que é responsável pela atualização do par de ganhos $g(k)$ e $g(k+1)$. Fazendo a combinação destas precisões, temos 4 cenários.

Como os resultados obtidos para os quatro cenários acima foram semelhantes, nesta dissertação serão apresentados apenas os resultados de dois deles: $\varepsilon = 10^{-8}$ e $\varepsilon_1 = 10^{-4}$ durante este capítulo; e $\varepsilon = 10^{-12}$ e $\varepsilon_1 = 10^{-4}$ no Apêndice B.

Observação 6.2 *Para os resultados apresentados neste capítulo os Algoritmos 2.1 e 5.1 foram iniciados com uma sequência de ganhos nulos. Porém, também foi testado o caso em que a sequência inicial é factível, ou seja, foi utilizada como solução inicial para o método Variacional Duplo a sequência dos ganhos obtidos a partir da primeira iteração do método Variacional Clássico. Contudo nenhuma melhora nos resultados foi constatada.*

São considerados como **exemplos resolvidos** por um determinado método aqueles que atendem os critérios de parada dentro do número máximo de iterações, visto que o critério de parada não garante que a solução alcançada é ótima. Assim, ao dizer que um método resolve um problema não estamos afirmando que o controle obtido tem custo próximo do valor ótimo. Por esse motivo, também é analisada a relação entre os custos obtidos, o número de iterações e os tempos gastos por cada um dos métodos implementados.

Vale ressaltar que a análise do número de iterações η está incluída essencialmente por duas razões: em primeiro lugar porque os métodos não melhoraram os custos e tempos computacionais, porém alguns exemplos apresentaram um número menor de iterações, caracterizando um cenário positivo para busca de métodos mais eficiente para o problema PB_k . Em segundo lugar, porque o número de iterações do método Variacional Modificado serve como boa estimativa para o número de iterações do método Variacional com Biquadrático, vide a Observação 4.1.

Para as análises apresentadas os conjuntos de exemplos serão subdivididos em 3 tipos: os exemplos resolvidos (ER), já mencionados; os exemplos não resolvidos dentro do número máximo de iterações permitidas (ENRMI), ou seja, que não atenderam os critérios de parada, mas que encontraram uma solução factível; e os exemplos não resolvidos/interrompidos (ENR), pois $\|X(k)\| > 10^{150}$ para algum valor de $k \in \mathcal{K}$ e/ou $\|\nabla \mathcal{J}_N\| > 10^{150}$.

O método Variacional Clássico é adotado como “referência” para comparação com os métodos propostos, ou seja, cada um dos métodos desenvolvidos é comparado com o Variacional Clássico, separadamente, nas três seções seguintes.

6.1 Variacional Clássico vs Variacional Modificado

Nesta seção é apresentada a comparação entre o método Variacional Clássico (MV) descrito no Capítulo 2 (Seção 2.4) e o método Variacional Modificado (MVM) descrito no Capítulo 5 (Seção 5.2.1).

Resultados com o Conjunto A

Primeiramente são analisados os percentuais de ER, ENR e ENRMI obtidos com o uso do método Variacional Clássico e do método Variacional Modificado. Conforme apresentado na

Figura 6.1, esses percentuais mostraram-se bem semelhantes. No total, temos que 667 exemplos foram resolvidos por ambos os métodos.

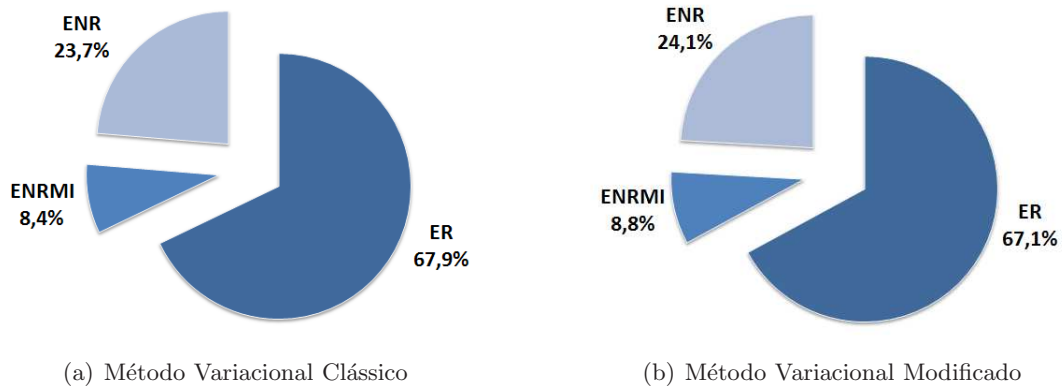


Figura 6.1: Classificação dos resultados obtidos pelos métodos Variacional Clássico e Variacional Modificado, para o conjunto A.

A Figura 6.2 apresenta o comportamento dos custos de N estágios obtidos pelo método Variacional Clássico, \mathcal{J}_{MV} , em relação aos custos obtidos pelo método Variacional Modificado, \mathcal{J}_{MVM} . A partir deste gráfico é possível concluir que somente a estratégia de atualizar dois ganhos consecutivos não é suficiente para obter custos menores para estes exemplos.

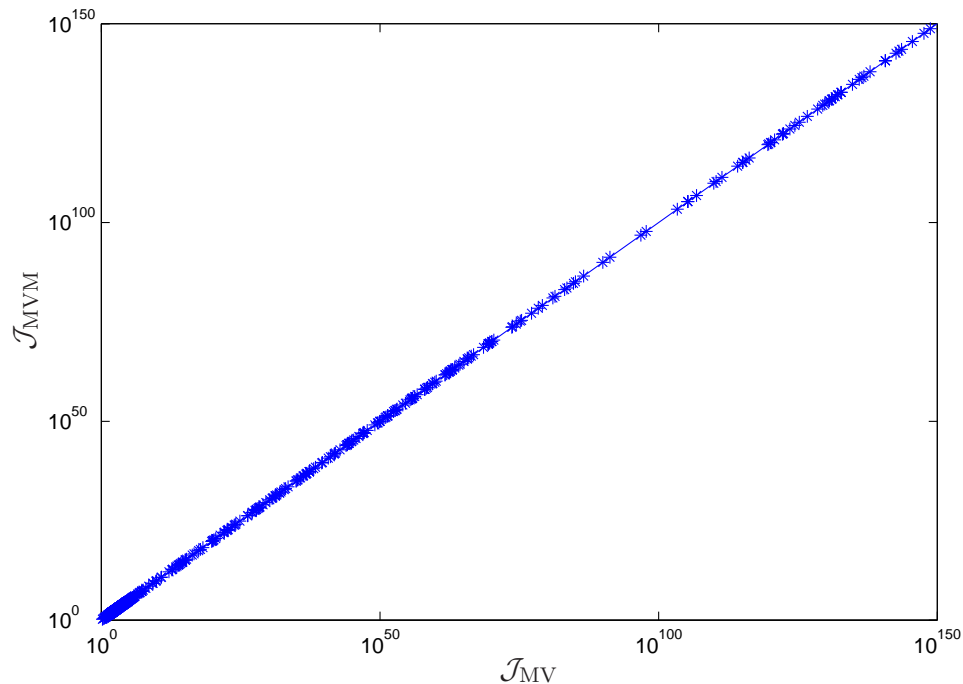


Figura 6.2: Custo do MV vs custo do MVM, para o conjunto A.

Na Figura 6.3 está apresentado o comportamento entre os tempos de CPU obtidos pelos métodos Variacional Clássico, T_{MV} , e Variacional Modificado, T_{MVM} .

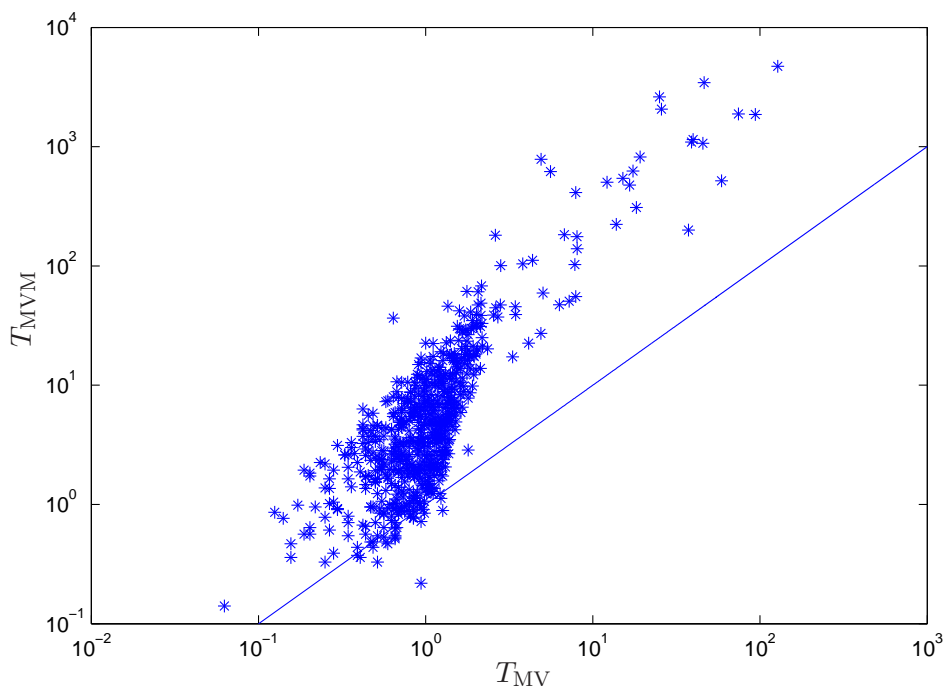


Figura 6.3: Tempo de CPU do MV vs tempo de CPU do MVM, para o conjunto A.

Lembre-se que o método Variacional Clássico, usado como referência para as comparações, possui apenas um contador de iteração, η (ao final do horizonte), e resolve, a cada ganho, apenas um sistema linear, ou seja “ $\zeta = 1$ ”. Assim, é justificável que:

1. o tempo gasto pelo método Variacional Modificado, conforme a Figura 6.3, seja maior, pois houve um aumento no número de sistemas lineares resolvidos, $\zeta \geq 1$, uma vez que os ganhos em atualização só deixam de ser atualizados após um critério de parada ser satisfeito;
2. a comparação acerca do número de iterações gastas seja feita na variável η .

Assim, na Figura 6.4 está apresentado o comportamento das iterações gastas pelo método Variacional Clássico, η_{MV} , em relação as iterações gastas pelo método Variacional Modificado, η_{MVM} . Note que não é possível afirmar qual dos métodos teve o melhor desempenho em relação aos resultados obtidos, logo um teste de hipótese, que pode ser encontrado em TRIOLA (1999), é utilizado para testar uma suposição a ser formulada sobre a razão entre o número de iterações obtido por cada método.

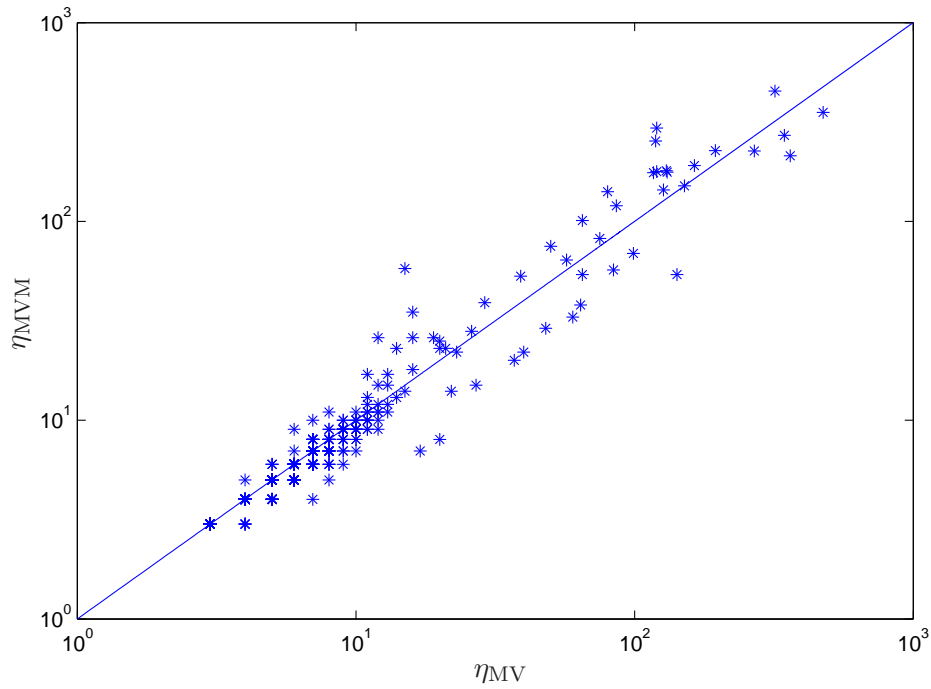


Figura 6.4: Número de iterações do MV vs número de iterações do MVM, para o conjunto A.

Teste de Hipótese 6.1 *Deseja-se testar se a média μ , razão entre o número de iterações η_{MV} do método Variacional Clássico e o número de iterações η_{MVM} do método Variacional Modificado, é igual a 1, contra a alternativa de ser maior que 1. Inferimos através da amostra que o desvio padrão do número de iterações é dado por $\sigma = 0,1796$. As duas hipóteses sobre a média da amostra são denotadas por H_0 e H_1 , respectivamente. Assim:*

$$\begin{aligned} H_0 : \mu &= 1 & (\eta_{MV} \leq \eta_{MVM}) \\ H_1 : \mu &> 1 & (\eta_{MV} > \eta_{MVM}). \end{aligned}$$

O erro ao rejeitar a hipótese H_0 quando esta é verdadeira é dado por α e será considerado 5%, ou seja, desejamos saber se é possível obter tal média $\bar{\mu}$ sendo a H_0 verdadeira, dentro de um porcentagem de 5% de erro. O escore z da média amostral é dado por:

$$z = \frac{\bar{\mu} - \mu}{S} = \frac{1,0340 - 1}{0,007} \approx 4,8571,$$

onde $S = \frac{\sigma}{\sqrt{Q}}$ e Q é o tamanho da amostra. O valor p (p), que diz quão provável é obter uma média amostral tão extrema quanto a nossa dado que H_0 é verdadeira, para este z é menor que 0,0001. Sabe-se que $p < \alpha$ implica em rejeitar a Hipótese Nula e concluir que o MVM alcança menores η em relação ao MV. A probabilidade de estarmos enganados nesta conclusão é de $\alpha = 5\%$.

Fica claro que há uma parcela de exemplos que utilizam um número menor de iterações η com a estratégia de atualização de dois ganhos.

Resultados com o Conjunto B

Como já mencionado, este conjunto é formado por 1.000 exemplos gerados aleatoriamente e com custo, obtido pelo método Variacional Clássico, igual ou inferior a 10^{50} , ou seja, 100% dos exemplos deste conjunto foram resolvidos pelo método Variacional Clássico. O método Variacional Modificado também resolveu todos os exemplos deste conjunto.

Na Figura 6.5 está apresentado o comportamento dos custos de N estágios obtidos pelo método Variacional Clássico, \mathcal{J}_{MV} , em relação aos custos obtidos pelo método Variacional Modificado, \mathcal{J}_{MVM} . Note que o comportamento dos custo deste conjunto é igual ao conjunto A, ou seja, ainda não há mudanças no valor dos custos obtidos pelo Variacional Modificado em relação ao método Variacional Clássico.

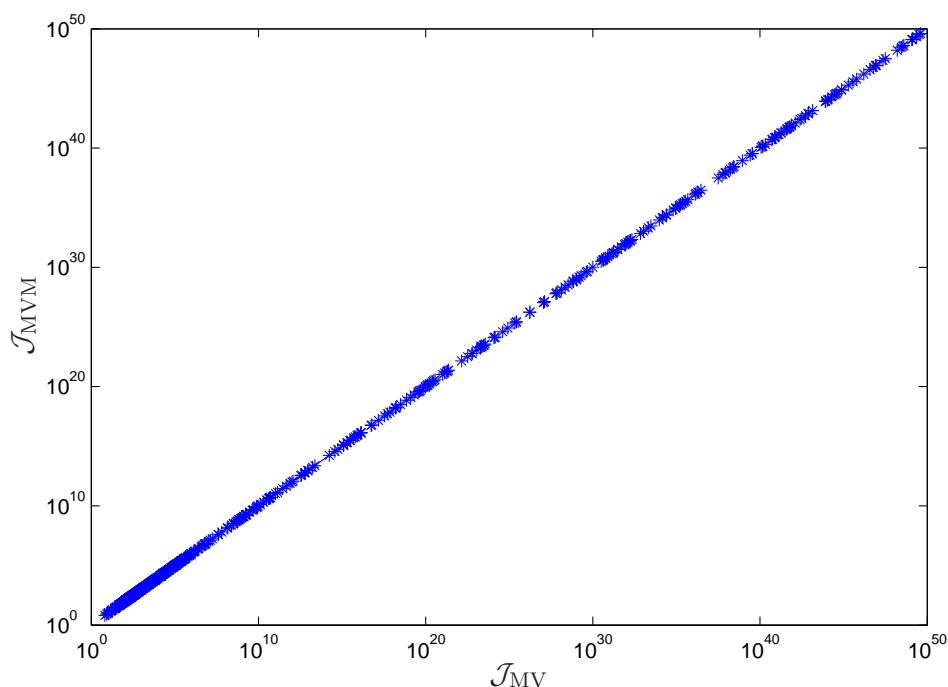


Figura 6.5: Custo do MV vs custo do MVM, para o conjunto B.

Assim como no conjunto A, os tempos de CPU gastos pelo método Variacional Modificado, T_{MVM} , foram superiores aos tempos de CPU gastos pelo método Variacional Clássico, T_{MV} , como pode ser visto na Figura 6.6.

Por fim, na Figura 6.7 está apresentado o comportamento das iterações η gastas pelo método Variacional Clássico, η_{MV} , em relação as iterações gastas pelo método Variacional Modificado, η_{MVM} .

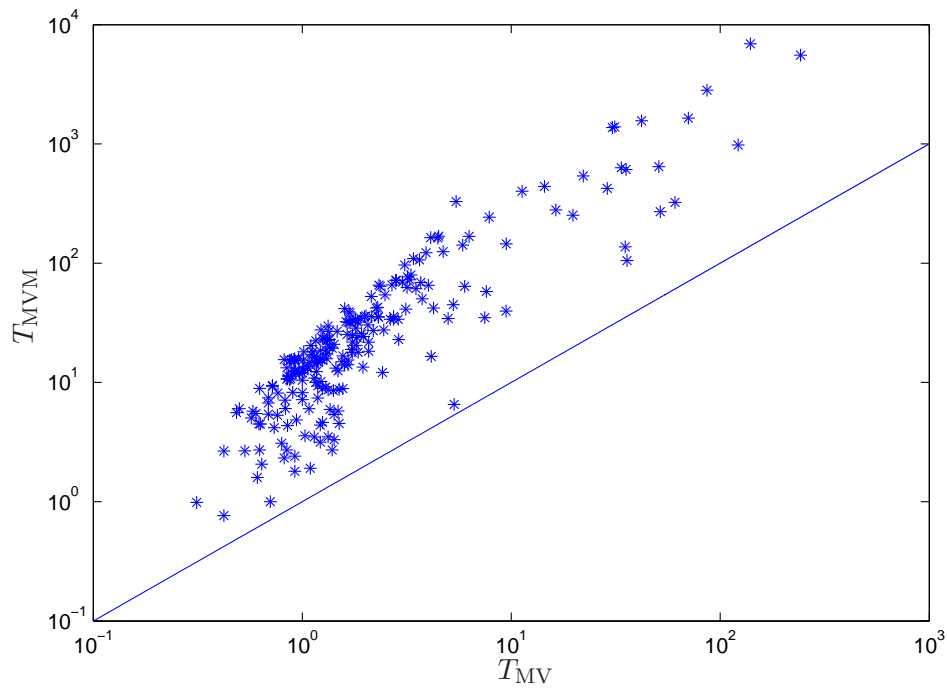


Figura 6.6: Tempo de CPU do MV vs tempo de CPU do MVM, para o conjunto B.

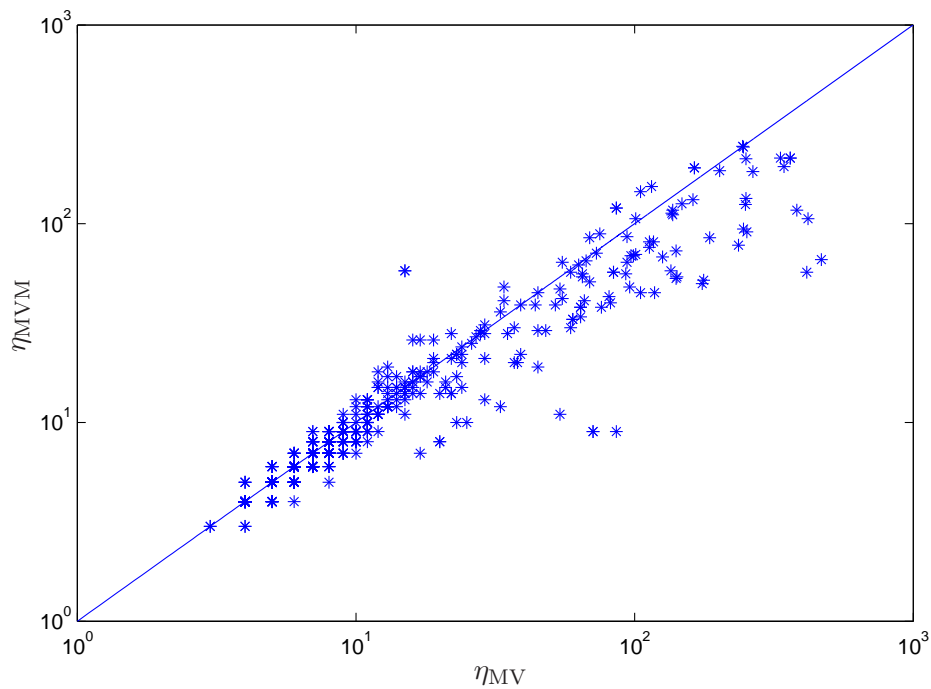


Figura 6.7: Número de iterações do MV vs número de iterações do MVM, para o conjunto B.

Novamente não é possível afirmar qual dos métodos teve o melhor desempenho em relação aos resultados obtidos, portanto um teste de hipótese é utilizado para testar uma suposição a ser formulada sobre a razão entre o número de iterações obtidos por cada métodos.

Teste de Hipótese 6.2 *Deseja-se testar se a média μ , razão entre o número de iterações η_{MV} do método Variacional Clássico e o número de iterações η_{MVM} do método Variacional Modificado, é igual a 1, contra a alternativa de ser maior que 1. Inferimos através da amostra que o desvio padrão do número de iterações é dado por $\sigma = 0,6933$. As duas hipóteses sobre a média da amostra são denotadas por H_0 e H_1 , respectivamente. Assim:*

$$\begin{aligned} H_0 : \mu &= 1 & (\eta_{MV} \leq \eta_{MVM}) \\ H_1 : \mu &> 1 & (\eta_{MV} > \eta_{MVM}). \end{aligned}$$

O erro ao rejeitar a hipótese H_0 quando esta é verdadeira é dado por α e será considerado 5% e o escore z da média amostral é dado por:

$$z = \frac{\bar{\mu} - \mu}{S} = \frac{1,1742 - 1}{0,0262} \approx 5,7535,$$

onde $S = \frac{\sigma}{\sqrt{Q}}$ e Q é o tamanho da amostra. O valor p (p) para este z é menor que 0,0001. Sabe-se que $p < \alpha$ implica em rejeitar a Hipótese Nula e concluir que o MVM alcança menores η em relação ao MV. A probabilidade de estarmos enganados nesta conclusão é de $\alpha = 5\%$.

6.2 Variacional Clássico vs Variacional Duplo com Gradiente

Nesta seção é apresentada a comparação entre o método Variacional Clássico (MV), descrito na Seção 2.4, e o método Variacional Duplo com Gradiente (MVDG), descrito na Seção 5.2.2.2.

Resultados com o Conjunto A

Os percentuais de ER, ENR e ENRMI obtidos com o uso do método Variacional Clássico e do método Variacional Duplo com Gradiente são apresentados na Figura 6.8. O total de exemplos resolvidos por ambos os métodos foi de 672.

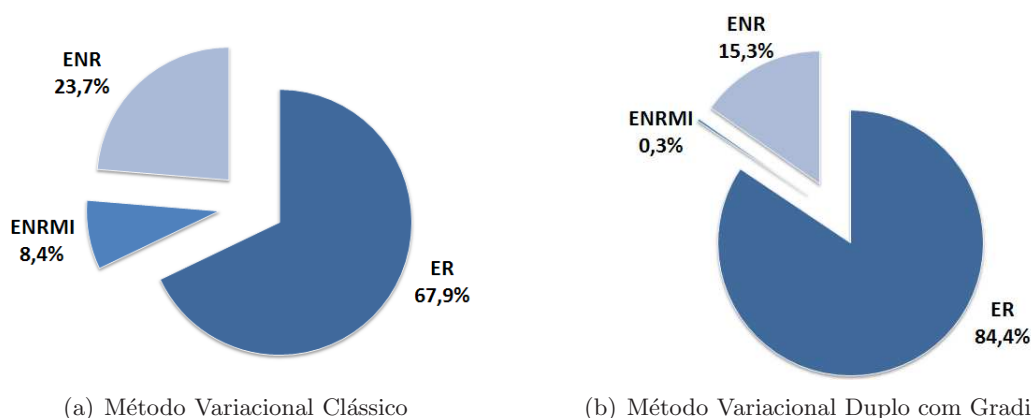


Figura 6.8: Classificação dos resultados obtidos pelos métodos Variacional Clássico e Variacional Duplo com Gradiente, para o conjunto A.

Na Figura 6.9 é apresentado o comportamento dos custos de N estágios obtidos pelo método Variacional Clássico, \mathcal{J}_{MV} , em relação aos custos obtidos pelo método Variacional Duplo com Gradiente, \mathcal{J}_{MVDG} . Utilizando o método de Gradiente para determinar a solução do problema PB_k foi possível obter um comportamento diferente, porém pior, daquele apresentado pelo método Variacional Modificado, ou seja, o método de Gradiente não foi capaz de encontrar custos melhores, no mínimo eles foram iguais.

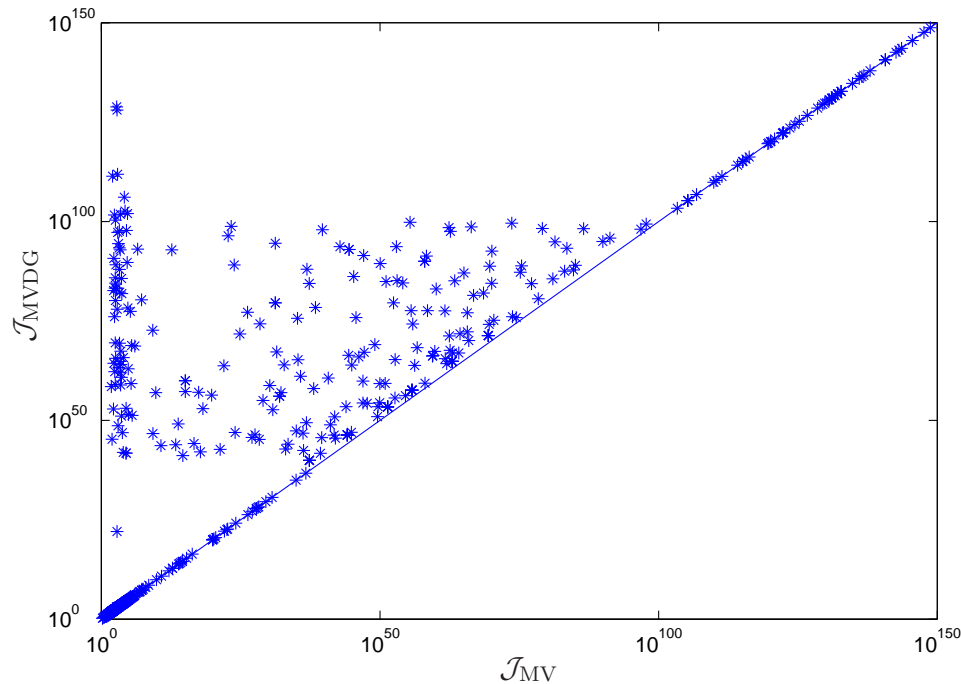


Figura 6.9: Custo do MV vs custo do MVDG, para o conjunto A.

O tempo de CPU do método Variacional Duplo com Gradiente continua sendo maior que o tempo do método Variacional Clássico, como pode ser visto na Figura 6.10, que apresenta o comportamento entre os tempos de CPU obtidos pelos métodos Variacional Clássico, T_{MV} , e Variacional Duplo com Gradiente, T_{MVDG} .

Na Figura 6.11 é apresentado o comportamento das iterações η gastas pelo método Variacional Clássico, η_{MV} , em relação as iterações gastas pelo método Variacional Duplo com Gradiente, η_{MVDG} .

Uma pergunta natural que surge a partir desta figura é: por que há um número significativo de exemplos que quando resolvidos pelo método Variacional Duplo com Gradiente apresentam um número de iterações η menor? A resposta pode ser vista na Figura 6.12 que mostra que a grande maioria dos exemplos que apresenta o número de iterações η menor apresentou custo \mathcal{J}_{MVDG} maior. É possível que esse também seja o motivo para o maior número de ER na Figura 6.8, ou seja, teríamos exemplos resolvidos apenas pelo MVDG, mas com custos maiores.

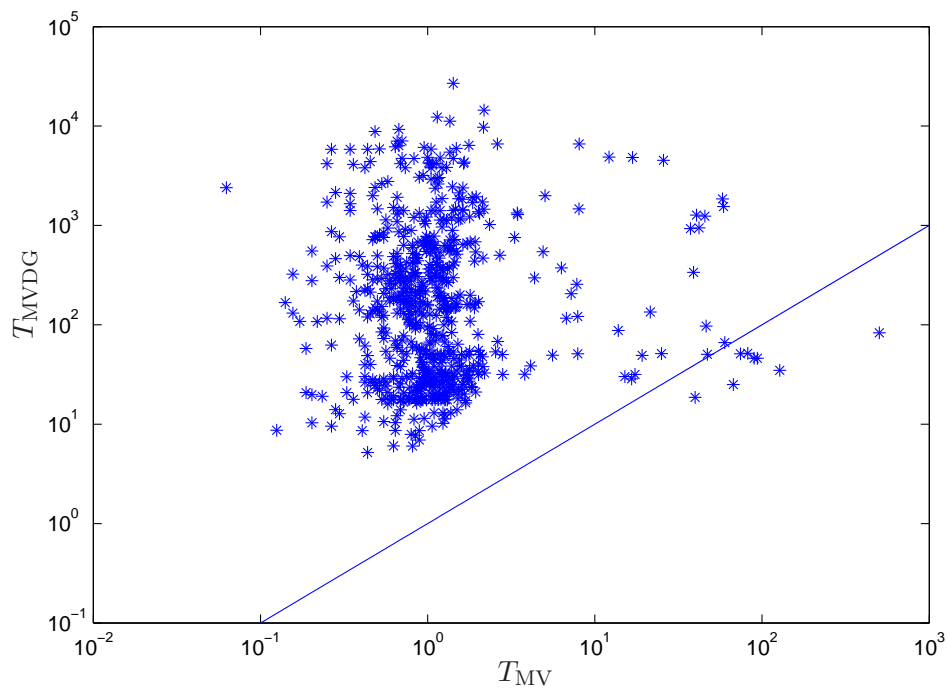


Figura 6.10: Tempo de CPU do MV vs tempo de CPU do MVDG, para o conjunto A.

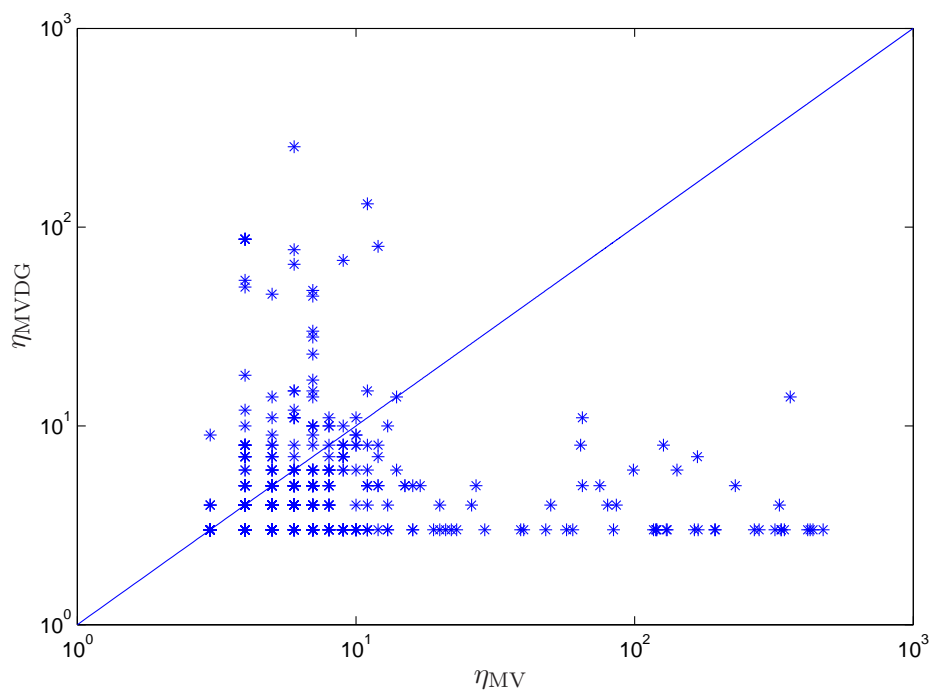


Figura 6.11: Número de iterações do MV vs número de iterações do MVDG, para o conjunto A.

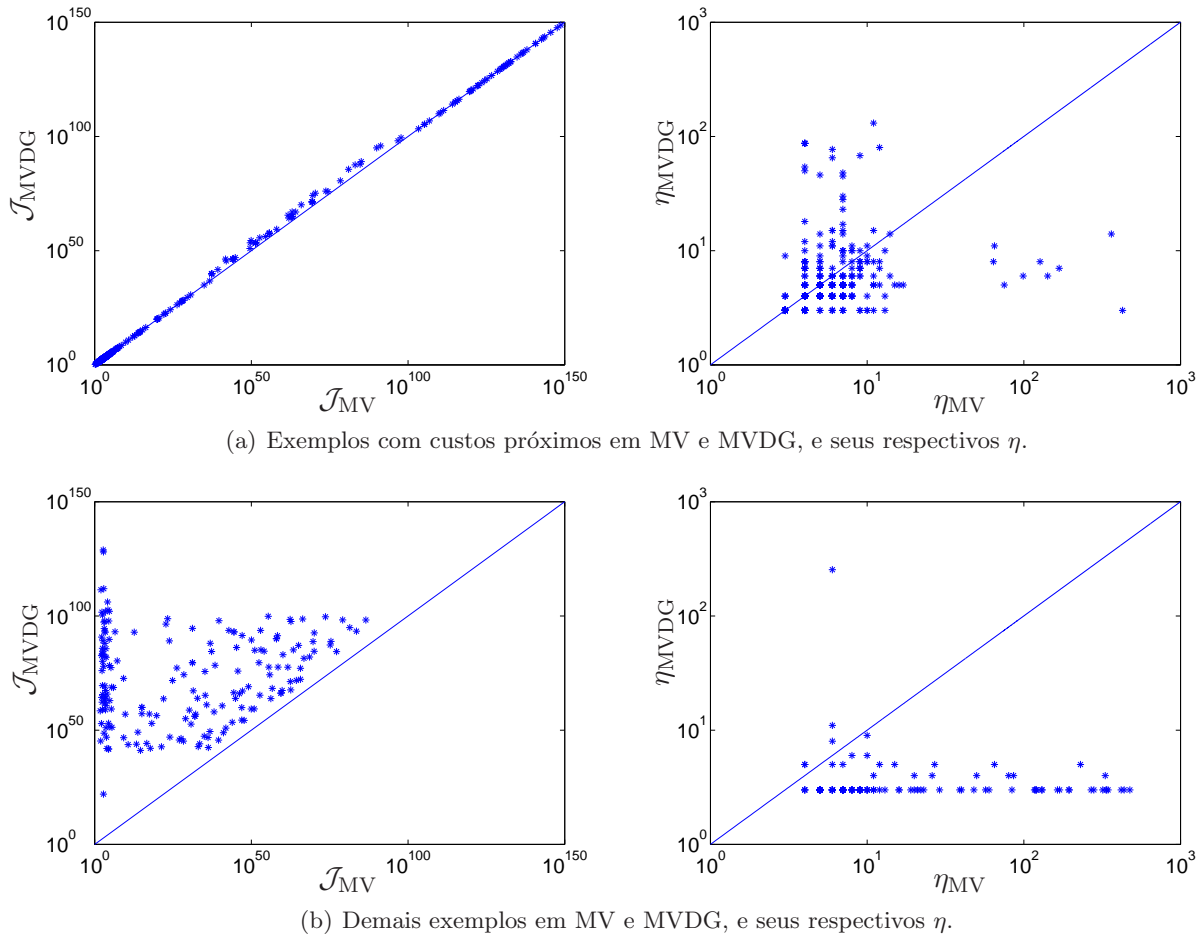


Figura 6.12: Categorização das iterações do Variacional Duplo com Gradiente, para o conjunto A.

Contudo, note que mesmo com Figura 6.12(a) não é possível afirmar qual dos métodos teve o melhor desempenho em relação ao número de iterações, por isso um teste de hipótese é utilizado.

Teste de Hipótese 6.3 *Deseja-se testar se a média μ , razão entre o número de iterações η_{MV} do método Variacional Clássico e o número de iterações η_{MVDG} do método Variacional Duplo com Gradiente, é igual a 1, contra a alternativa de ser maior que 1. Inferimos através da amostra que o desvio padrão do número de iterações é dado por $\sigma = 6,7609$. As duas hipóteses sobre a média da amostra são denotadas por H_0 e H_1 , respectivamente. Assim:*

$$H_0 : \mu = 1 \quad (\eta_{MV} \leq \eta_{MVDG})$$

$$H_1 : \mu > 1 \quad (\eta_{MV} > \eta_{MVDG}).$$

O erro ao rejeitar a hipótese H_0 quando esta é verdadeira é dado por α e será considerado 5% e o escore z da média amostral é dado por:

$$z = \frac{\bar{\mu} - \mu}{S} = \frac{1,7397 - 1}{0,3061} \approx 2,4165,$$

onde $S = \frac{\sigma}{\sqrt{Q}}$ e Q é o tamanho da amostra, que neste caso será $Q = 488$ (apenas os exemplos com custo próximo), vide Figura 6.12(a). O valor p (p) para este z é 0,0078. Sabe-se que $p < \alpha$ ($0,78\% < 5\%$) implica em rejeitar a Hipótese Nula e concluir que o MVDG alcança menores η em relação ao MV. A probabilidade de estarmos enganados nesta conclusão é de $\alpha = 5\%$.

Resultados com o Conjunto B

Como já mencionado, 100% dos exemplos deste conjunto foram resolvidos pelo método Variacional Clássico. Para o método Variacional Duplo com Gradiente, os resultados encontram-se na Figura 6.13. No total, temos que 687 exemplos foram resolvidos por ambos os métodos.

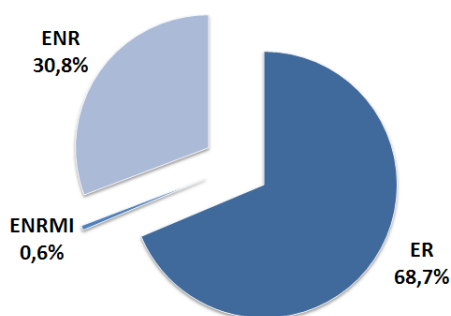


Figura 6.13: Classificação dos resultados obtidos pelo método Variacional Duplo com Gradiente, para Conjunto B.

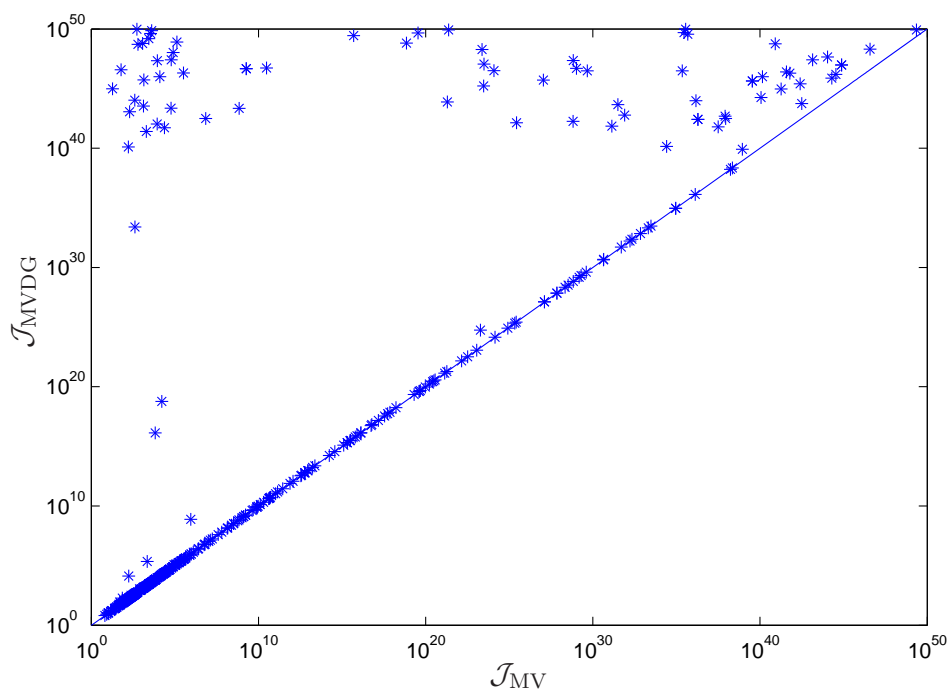


Figura 6.14: Custo do MV vs custo do MVDG, para o conjunto B.

O método Variacional Duplo com Gradiente novamente não conseguiu obter custos melhores quando comparado com o método Variacional Clássico, como pode ser visto na Figura 6.14. O mesmo vale para o tempo, conforme visto na Figura 6.15, que apresenta o comportamento entre os tempos de CPU obtidos pelos métodos Variacional Clássico, T_{MV} , e Variacional Duplo com Gradiente, T_{MVDG} .

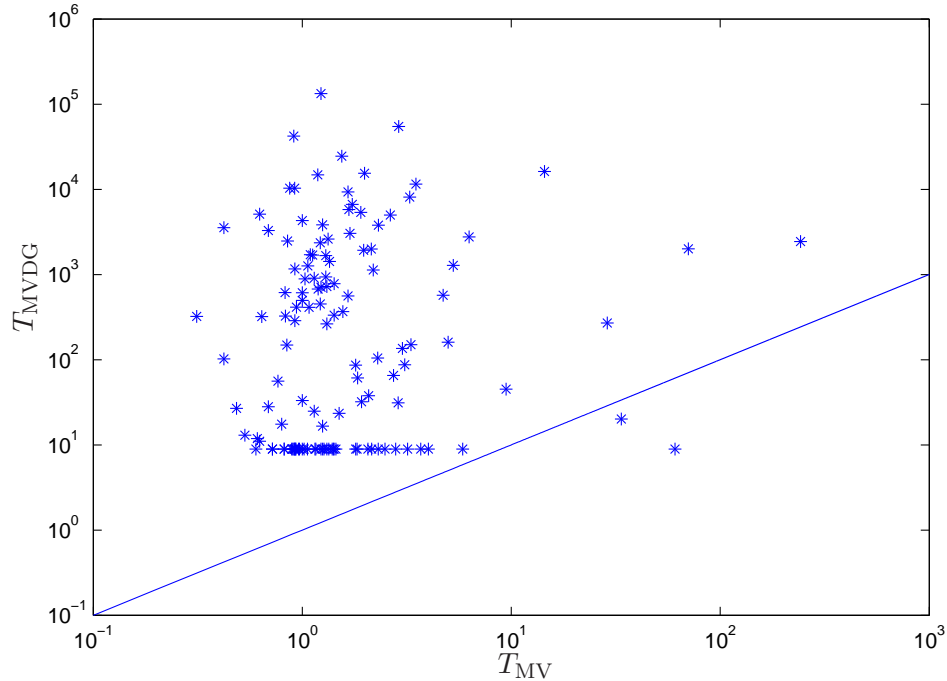


Figura 6.15: Tempo de CPU do MV vs tempo de CPU do MVDG, para o conjunto B.

Por fim, na Figura 6.16 é apresentado o comportamento das iterações η utilizadas pelo método Variacional Clássico, η_{MV} , em relação as iterações do método Variacional Duplo com Gradiente, η_{MVDG} . Analogamente aos casos anteriores, não é possível afirmar qual método teve melhor resultado em termos do número de iterações, de forma que, novamente, um teste de hipóteses é utilizado.

Teste de Hipótese 6.4 *Deseja-se testar se a média μ , razão entre o número de iterações η_{MV} do método Variacional Clássico e o número de iterações η_{MVDG} do método Variacional Duplo com Gradiente, é igual a 1, contra a alternativa de ser maior que 1. Inferimos através da amostra que o desvio padrão do número de iterações é dado por $\sigma = 5,5686$. As duas hipóteses sobre a média da amostra são denotadas por H_0 e H_1 , respectivamente. Assim:*

$$\begin{aligned} H_0 : \mu &= 1 & (\eta_{MV} \leq \eta_{MVDG}) \\ H_1 : \mu &> 1 & (\eta_{MV} > \eta_{MVDG}). \end{aligned}$$

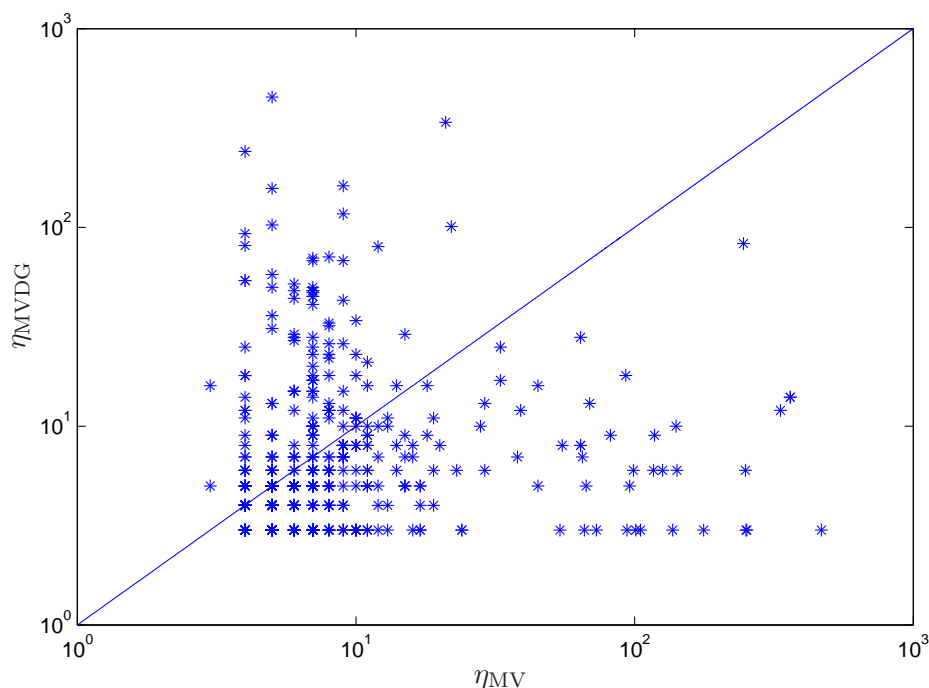


Figura 6.16: Número de iterações do MV vs número de iterações do MVDG, para o conjunto B.

O erro ao rejeitar a hipótese H_0 quando esta é verdadeira é dado por α e será considerado 5% e o escore z da média amostral é dado por:

$$z = \frac{\bar{\mu} - \mu}{S} = \frac{2,1206 - 1}{0,2720} \approx 4,1198,$$

onde $S = \frac{\sigma}{\sqrt{Q}}$ e Q é o tamanho da amostra. Neste teste, apenas os exemplos satisfazendo $\mathcal{J}_{MVDG}/\mathcal{J}_{MV} < 10^5$ (custos próximos na escala da Figura 6.14) são considerados, totalizando $Q = 599$. O valor p (p) para este z é menor que 0,0001. Sabe-se que $p < \alpha$ implica em rejeitar a Hipótese Nula e concluir que o MVDG alcança menores η em relação ao MV. A probabilidade de estarmos enganados nesta conclusão é de $\alpha = 5\%$.

6.3 Variacional Clássico vs Variacional Duplo com Newton

Nesta seção é apresentada a comparação entre o método Variacional Clássico (MV), descrito na Seção 2.4, e o método Variacional Duplo com Newton (MVDN), descrito na Seção 5.2.2.3.

Resultados com o Conjunto A

Os percentuais de ER, ENR e ENRMI obtidos com o uso do método Variacional Clássico e do método Variacional Duplo com Newton são apresentados na Figura 6.17. Nota-se uma

diminuição relativamente pequena, mas significativa, em ER. No total, 543 exemplos foram resolvidos por ambos os métodos.

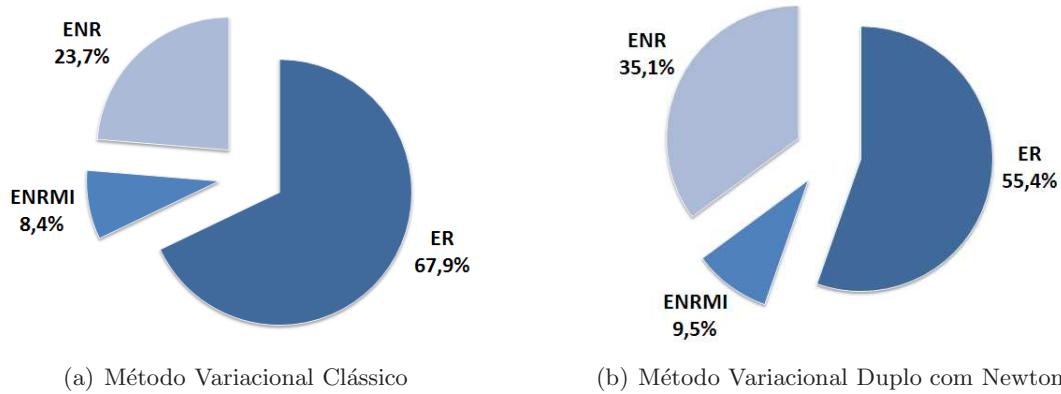


Figura 6.17: Classificação dos resultados obtidos pelos métodos Variacional Clássico e Variacional Duplo com Newton.

Na Figura 6.18 é apresentado o comportamento dos custos de N estágios obtidos pelo método Variacional Clássico, \mathcal{J}_{MV} , em relação aos custos obtidos pelo método Variacional Duplo com Newton, \mathcal{J}_{MVDN} . Como se vê, os custos obtidos são mais altos (ainda que relativamente próximos aos do MV) em todos os exemplos. Isso representa uma melhora em relação ao MVDG.

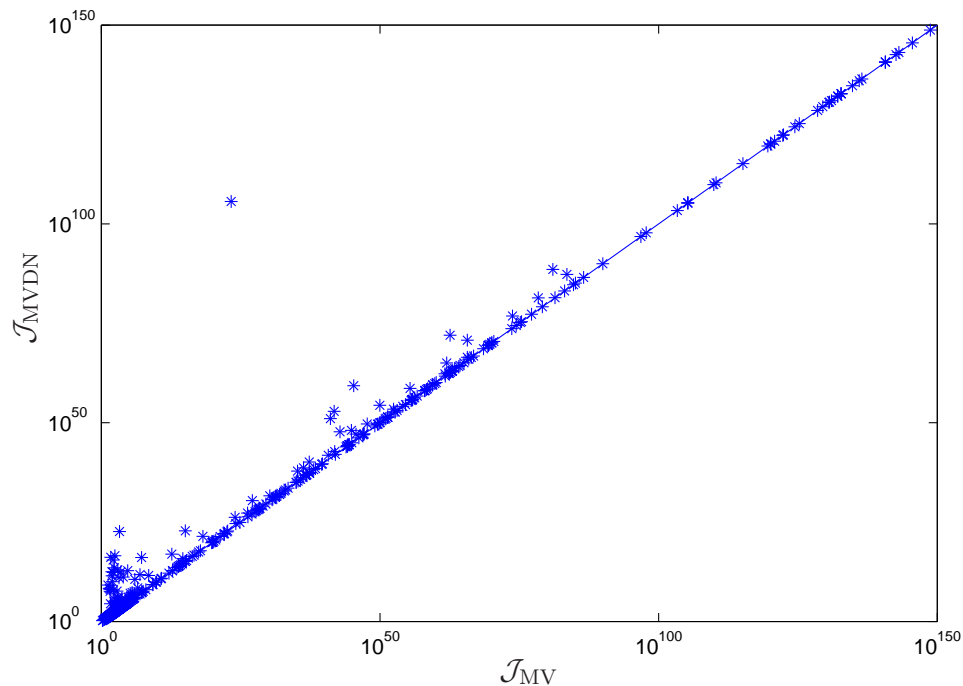


Figura 6.18: Custo do MV vs custo do MVDN, para o conjunto A.

Em termos do tempo de CPU, o resultado do MVDN foi ruim, como mostrado na Figura 6.19 que apresenta o comportamento entre os tempos de CPU obtidos pelos métodos Variacional Clássico, T_{MV} , e Variacional Duplo com Newton, T_{MVDN} .

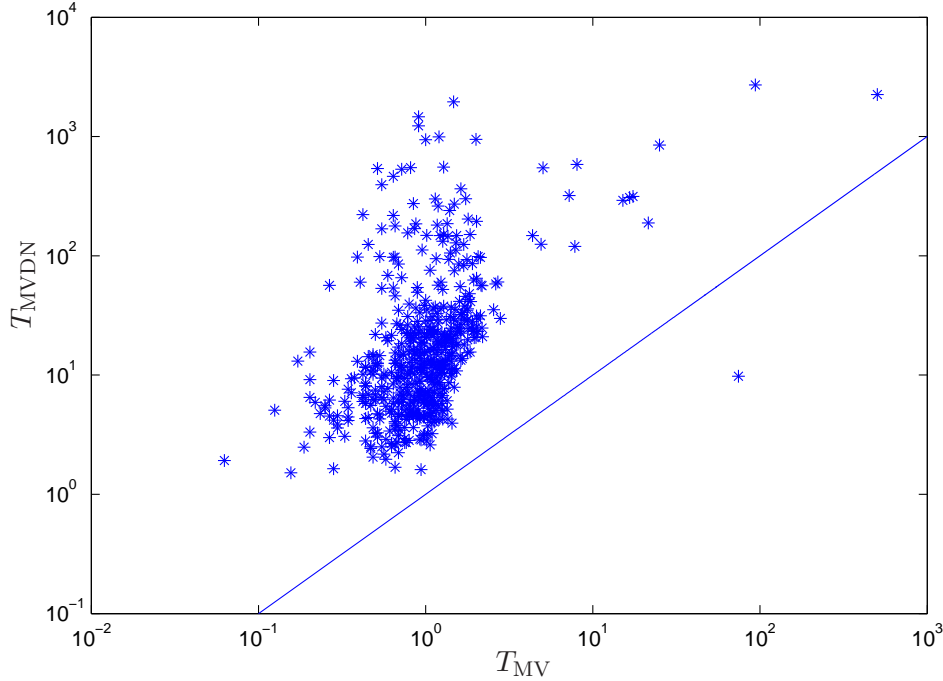


Figura 6.19: Tempo de CPU do MV vs tempo de CPU do MVDN, para o conjunto A.

Na Figura 6.20 é apresentado o comportamento das iterações η gastas pelo método Variacional Clássico, η_{MV} , em relação as iterações gastas pelo método Variacional Duplo com Newton, η_{MVDN} . Analogamente às seções anteriores, um teste de hipótese é utilizado para analisar o número de iterações.

Teste de Hipótese 6.5 *Deseja-se testar se a média μ , razão entre o número de iterações η_{MV} do método Variacional Clássico e o número de iterações η_{MVDN} do método Variacional Duplo com Newton, é igual a 1, contra a alternativa de ser maior que 1. Inferimos através da amostra que o desvio padrão do número de iterações é dado por $\sigma = 3,8408$. As duas hipóteses sobre a média da amostra são denotadas por H_0 e H_1 , respectivamente. Assim:*

$$\begin{aligned} H_0 : \mu &= 1 & (\eta_{MV} \leq \eta_{MVDN}) \\ H_1 : \mu &> 1 & (\eta_{MV} > \eta_{MVDN}). \end{aligned}$$

O erro ao rejeitar a hipótese H_0 quando esta é verdadeira é dado por α e será considerado 5% e o escore z da média amostral é dado por:

$$z = \frac{\bar{\mu} - \mu}{S} = \frac{1,0364 - 1}{0,1648} \approx 0,2209,$$

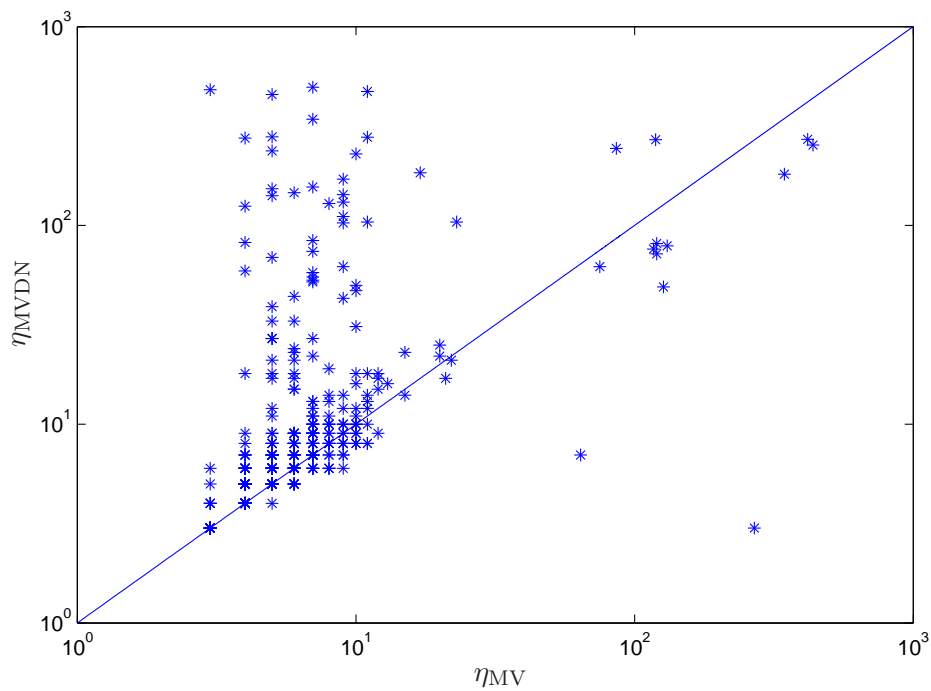


Figura 6.20: Número de iterações do MV vs número de iterações do MVDN, para o conjunto A.

onde $S = \frac{\sigma}{\sqrt{Q}}$ e Q é o tamanho da amostra. O valor p (p) z é 0,4129. Sabe-se que $p > \alpha$ (41,29% > 5%) implica em aceitar a Hipótese Nula e concluir que o MV alcança menores η em relação ao MVDN.

Resultados com o Conjunto B

Como já mencionado, todos os exemplos deste conjunto foram resolvidos pelo método Variacional Clássico e os resultados para o método Variacional Duplo com Newton estão ilustrados na Figura 6.21. No total, 695 exemplos foram resolvidos por ambos os métodos.

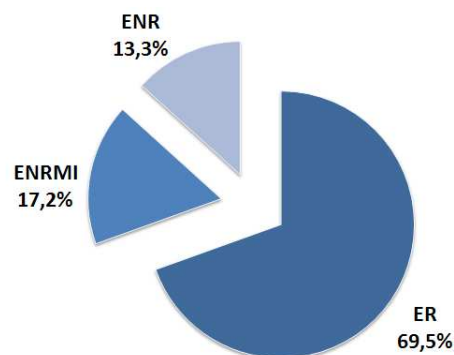


Figura 6.21: Classificação dos resultados obtidos pelo método Variacional Duplo com Newton, para Conjunto B.

Na Figura 6.22 é apresentada a relação entre os custos obtidos pelo método Variacional Clássico, \mathcal{J}_{MV} , e os custos obtidos pelo método Variacional Duplo com Newton, \mathcal{J}_{MVDN} .

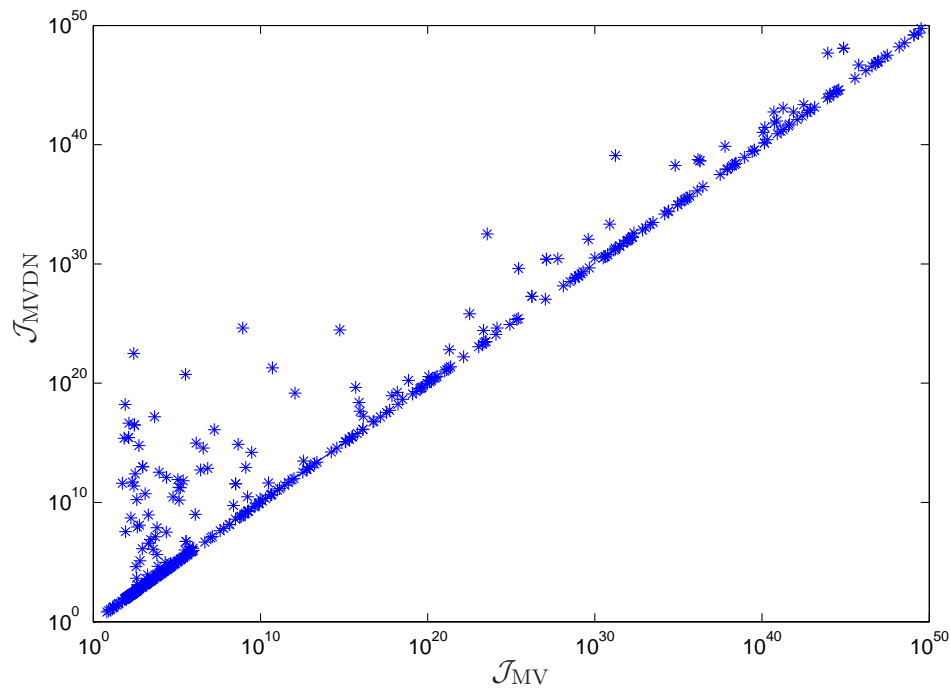


Figura 6.22: Custo do MV vs custo do MVDN, para o conjunto B.

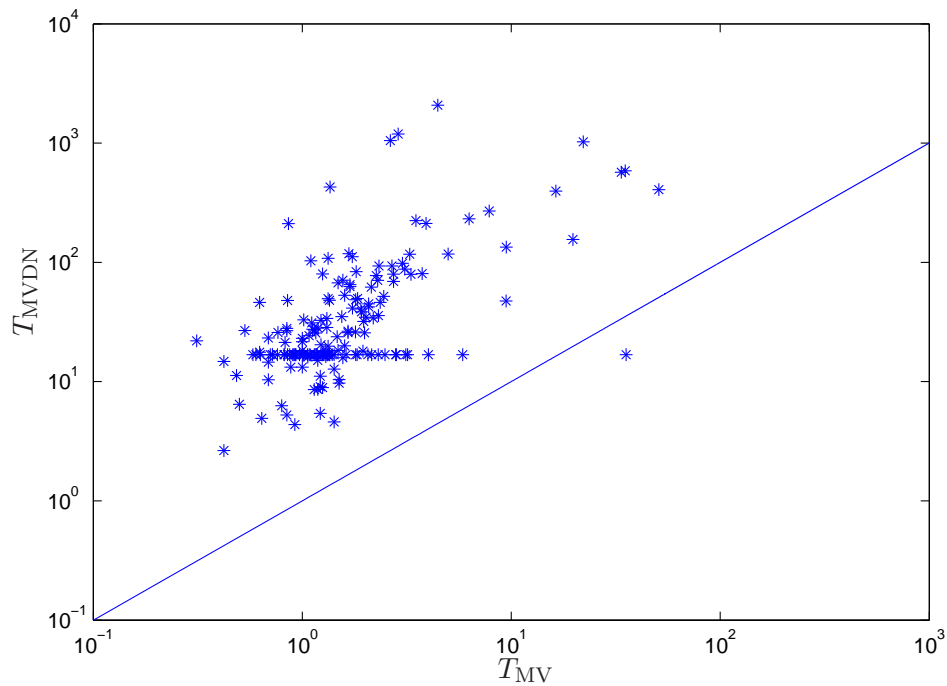


Figura 6.23: Tempo de CPU do MV vs tempo de CPU do MVDN, para o conjunto B.

Na Figura 6.23 é apresentado o comportamento entre os tempos de CPU obtidos pelos métodos Variacional Clássico, T_{MV} , e Variacional Duplo com Newton, T_{MVDN} . O tempo gasto pelo MVDN é novamente superior ao tempo gasto pelo MV. Por fim, na Figura 6.24 é apresentado o comportamento das iterações η gastas pelo método Variacional Clássico, η_{MV} , em relação as iterações η gastas pelo método Variacional Duplo com Newton, η_{MVDN} .

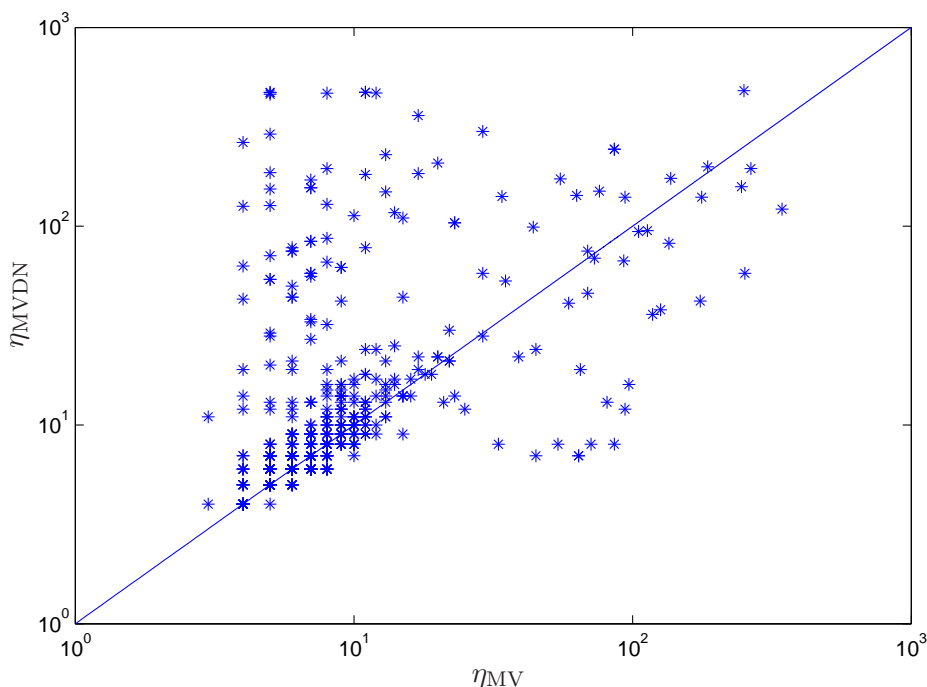


Figura 6.24: Número de iterações do MV vs número de iterações do MVDN, para o conjunto B.

Novamente, um teste de hipóteses é utilizado para analisar o resultado quanto ao número de iterações.

Teste de Hipótese 6.6 *Deseja-se testar se a média μ , razão entre o número de iterações η_{MV} do método Variacional Clássico e o número de iterações η_{MVDN} do método Variacional Duplo com Newton, é igual a 1, contra a alternativa de ser maior que 1. Inferimos através da amostra que o desvio padrão do número de iterações é dado por $\sigma = 1,0927$. As duas hipóteses sobre a média da amostra são denotadas por H_0 e H_1 , respectivamente. Assim:*

$$H_0 : \mu = 1 \quad (\eta_{MV} \leq \eta_{MVDN})$$

$$H_1 : \mu > 1 \quad (\eta_{MV} > \eta_{MVDN}).$$

O erro ao rejeitar a hipótese H_0 quando esta é verdadeira é dado por α e será considerado 5% e o escore z da média amostral é dado por:

$$z = \frac{\bar{\mu} - \mu}{S} = \frac{0,9979 - 1}{0,0496} \approx -0,0423,$$

onde $S = \frac{\sigma}{\sqrt{Q}}$ e Q é o tamanho da amostra. O valor p (p) para este z é 0,5160. Sabe-se que $p > \alpha$ (51,6% > 5%) implica em aceitar a Hipótese Nula e concluir que o MV alcança menores η em relação ao MVDN.

6.4 Considerações Finais deste Capítulo

Neste capítulo foram apresentados os resultados obtidos a partir dos experimentos computacionais, comparando os métodos desenvolvidos, descritos no Capítulo 5, com um método clássico encontrado na literatura: o método Variacional.

Analisando os dados obtidos pelo MV e o MVM, Seção 6.1, concluímos que o MV detem os melhores resultados em termos de custo e de tempo de CPU na grande maioria dos exemplos, em ambos os conjuntos e para quaisquer precisões. Contudo, o MVM apresenta em uma parcela significativa de exemplos um menor número de iterações η , motivando a busca por métodos mais eficientes para o problema PB_k .

O tempo a mais utilizado pelo MVM é justificável pelo aumento no número de sistemas lineares resolvidos, $\zeta \geq 1$, uma vez que os ganhos $g(k)$ e $g(k+1)$ só deixam de ser atualizados após um critério de parada ser satisfeito.

Para os dados obtidos pelo MV e o MVDG, Seção 6.2, pode-se concluir que o MVDG resolveu um percentual maior de exemplos do conjunto A quando comparado com o MV, porém os custos obtidos foram maiores. Possivelmente isto está relacionado aos exemplos que apresentaram o número de iterações η menor e custos maiores, conforme pode ser visto na Figura 6.12.

Em relação ao tempo consumido, tem-se que o MVDG é mais lento que o MV. A justificativa para este caso são os cálculos das derivadas do custo, além do aumento no número de atualizações realizadas nos ganhos.

Por fim, com os resultados obtidos pelo MV e o MVDN, Seção 6.3, tem-se que o MVDN resolveu um percentual menor de exemplos tanto no conjunto A quanto no conjunto B. Novamente os custos obtidos não foram melhores que aqueles obtidos pelo MV, porém foram melhores que os obtidos pelo MVDG.

Assim como no caso do MVDG, o MVDN é mais lento que o MV e a justificativa está novamente nos cálculos das derivadas (agora de primeira e segunda ordem) do custo, além do aumento no número de atualizações realizadas nos ganhos.

Por fim, vale ressaltar que além da parcela de exemplos que apresentaram menores números de iterações foi possível obter uma frequência pequena de exemplos em que o Variacional Duplo detêm melhores custos, porém em todos os casos estes são exemplos ENRMI, ou seja, não atenderam aos critérios de parada dentro de um número máximo de iterações. O limitante máximo para o número de iterações foi aumentado para 1.000, mas mesmo assim nenhum exemplo atendeu aos critérios de parada dentro deste novo limitante, por isso estes exemplos não foram considerados nesta análise.

Conclusões

As principais contribuições deste trabalho foram: apresentar um contraexemplo para a questão da suficiência da condição de otimalidade, dada pela Equação (2.25), a caracterização do problema PB_k com um problema biquadrático e a análise dos resultados obtidos pela implementação computacional de métodos para a resolução do problema PB_k . Cada um destes pontos é comentado a seguir.

Durante o levantamento do conjunto de exemplos utilizados nos testes computacionais, foi encontrado um exemplo em que é possível verificar que o problema de custo quadrático de N estágios para SLSM não observados/parcialmente observados pode apresentar mínimos locais múltiplos e distintos, ou seja, neste exemplo obtemos mínimos distintos a partir de ganhos iniciais também distintos.

Além disso, ao propor uma adaptação do método Variacional para atualizar dois ganhos consecutivos, obtemos a formalização do problema intermediário PB_k , até então não estudado pela literatura. A partir de um exemplo simples foi possível notar que, desde que feita uma pequena adaptação em suas variáveis, o problema PB_k é biquadrático. A demonstração feita considera o caso em que a ação de controle é escalar, porém testes computacionais confirmaram este resultado.

Por fim, para a resolução do problema PB_k foram utilizados três métodos: Variacional Clássico, que junto com o método Variacional Duplo é denominado de Variacional Modificado (Seção 5.2.1), Gradiente (Seção 5.2.2.2) e Newton (Seção 5.2.2.3). Para comparar o desempenho destes métodos dois conjuntos de exemplos, ambos gerados aleatoriamente com o *Gerador de SLSM* apresentado em BORTOLIN (2012) e SILVA (2012), foram utilizados. O método Variacional Clássico foi tomado como referência para a comparação entre os métodos.

Em termos gerais, os métodos propostos não apresentaram bons resultados (custo e tempo). Mesmo assim, os resultados obtidos sobre o número de iterações do método Variacional Modificado justificam a busca por melhores métodos para o problema PB_k . Provavelmente um método específico para um problema biquadrático seja a maneira mais eficiente de obter melhores desempenhos (tempo e iterações) e, possivelmente, melhores custos para o problema de otimização.

Como trabalhos futuros pretendemos primeiramente demonstrar formalmente a caracterização do caso geral do PB_k como um problema biquadrático. Além disso, consideramos interessante adaptar o PB_k a algum algoritmo biquadrático existente (ou adaptar este algoritmo ao problema) de modo que seja possível implementar o método biquadrático. Outra linha de pesquisa interessante seria buscar alternativas não tão rígidas para a quantidade de ganhos atualizados, por exemplo, atualizar dois ganhos não consecutivos, com ordem aleatória, ou ainda atualizar diferentes quantidades de ganhos ao longo das iterações.

Referências Bibliográficas

- BORTOLIN, D. C. (2012). Métodos numéricos para o controle linear quadrático com saltos e observação parcial de estado. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, SP.
- COSTA, O. L. V., FRAGOSO, M. D., e MARQUES, R. P. (2005). *Discrete-Time Markovian Jump Linear Systems*. Springer.
- COSTA, O. L. V., FRAGOSO, M. D., e TODOROV, M. G. (2013). *Continuous-Time Markov Jump Linear Systems*. Springer.
- DO VAL, J. e BAŞAR, T. (1999). Receding horizon control of jump linear systems and a macroeconomic policy problem. *Journal of Economic Dynamics & Control*, 23(8):1099 – 1131.
- DRAGAN, V., MOROZAN, T., e M., S. A. (2009). *Mathematical methods in robust control of discrete-time linear stochastic systems*. Springer.
- FACKLER, P. L. (2005). Notes on matrix calculus: North Carolina State University, USA. <http://www4.ncsu.edu/~pfackler/MatCalc.pdf>.
- ÇINLAR, E. (1975). *Introduction to stochastic processes*. Prentice Hall, New York.
- JOHO, M. (2000). *A systematic approach to adaptive algorithms for multichannel system identification, inverse modeling, and blind identification*. Tese de Doutorado, Swiss Federal Institute of Technology.
- LAM, J., SHU, Z., XU, S., e BOUKAS, E. K. (2007). Robust H_∞ control of descriptor discrete-time markovian jump systems. *International Journal of Control*, 80(3):374 – 385.
- LEE, J. (2009). IV - Differentiability and convexity, notas de aula para a disciplina EE 581: Optimal control, Department of Electrical Engineering, Pennsylvania State University, University

- Park, USA. http://www.personal.psu.edu/jxl77/courses/ee581_sp09/ee581_slides4.pdf.
- LING, C., NIE, J., QI, L., e YE, Y. (2009). Bi-quadratic optimization over unit spheres and semidefinite programming relaxations. *SIAM Journal on Optimization*, 20(3):1286 – 1310.
- LING, C., ZHANG, X., e QI, L. (2012). Semidefinite relaxation approximation for multivariate bi-quadratic optimization with quadratic constraints. *Numerical Linear Algebra with Applications*, 19(1):113 – 131.
- OLIVEIRA, A. (2011). *Controle ótimo de sistemas lineares com saltos Markovianos e ruídos multiplicativos sob critério de média variância ao longo do tempo*. Tese de Doutorado, Escola Politécnica da Universidade de São Paulo, Departamento de Engenharia de Telecomunicações e Controle.
- OLIVEIRA, L. T. e COSTA, E. F. (2013). Derivadas de primeira e segunda ordem do custo de t estágios para atualização simultânea de dois ganhos no controle de sistemas com saltos markovianos. *Congresso de Matemática Aplicada e Computacional*, pages 337 – 343.
- OLIVEIRA, L. T. e COSTA, E. F. (2014). Sobre a unicidade da solução no controle de sistemas lineares com saltos markovianos não observados, (ACEITO). *Congresso Nacional de Matemática Aplicada e Computacional*.
- RAOUF, J. e BOUKAS, E. (2007). Stabilization of discontinuous singular systems with markovian switching and saturating inputs. *American Control Conference, 2007*, pages 2442 – 2447.
- SARIDIS, G. N. (1983). Intelligent robotic control. *IEEE Transactions on Automatic Control*, 28(5):547 – 557.
- SILVA, C. A. (2012). *Algoritmos para o custo médio a longo prazo de sistemas com saltos markovianos parcialmente observados*. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, SP.
- SIQUEIRA, A. A. G. e TERRA, M. H. (2004). Nonlinear and markovian H_∞ -controls of underactuated manipulators. *IEEE Transactions on Control System Technology*, 12(6):811 – 826.
- STEVENS, B. L. e LEWIS, F. L. (1991). *Aircraft Modeling, Dynamics and Control*. Wiley, New York.
- SWORDER, D. D. e ROGERS, R. O. (1983). An LQ-solution to a control problem associated with a solar thermal central receiver. *IEEE Transactions on Automatic Control*, 28(10):971 – 978.

- TRIOLA, M. F. (1999). *Introdução à estatística*. Livros Técnicos e Científicos Editora, Rio de Janeiro.
- VARGAS, A. N. (2004). Controle por horizonte retrocedente de sistemas lineares com saltos markovianos e ruído aditivo. Dissertação de Mestrado, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, SP.
- VARGAS, A. N., BORTOLIN, D. C., COSTA, E. F., e DO VAL, J. B. R. (2014). Gradient-based optimization techniques for the design of static controllers for Markov jump linear systems with unobservable modes. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*.
- VARGAS, A. N., COSTA, E. F., e DO VAL, J. B. R. (2013). On the control of Markov jump linear systems with no mode observation: application to a DC motor device. *International Journal of Robust and Nonlinear Control*, 23(10):1136 – 1150.
- VARGAS, A. N., DO VAL, J. B. R., e COSTA, E. F. (2004). Receding horizon control of Markov jump linear systems subject to noise and unobserved state chain. In *43rd IEEE Conference on Decision and Control*, volume 4, pages 4381 – 4386.
- VARGAS, A. N., DO VAL, J. B. R., e COSTA, E. F. (2005). Optimality condition for the receding horizon control of Markov jump linear systems with non-observed chain and linear feedback controls. In *44th IEEE Conference on Decision and Control and 2005 European Control Conference*, pages 7308 – 7313.
- YANG, Y. e Q., Y. (2012). On solving biquadratic optimization via semidefinite relaxation. *Computational Optimization and Applications*, 53(3):845 – 867.
- ZHANG, X., LING, C., e QI, L. (2011). Semidefinite relaxation bounds for bi-quadratic optimization problems with quadratic constraints. *Journal of Global Optimization*, 49(2):293 – 311.

Demonstração do Teorema 4.1

A.1 Resultados Preliminares

Teorema A.1 Para $x, y \in \mathbb{R}^n$ vale:

$$\text{vec}(xy') = y' \otimes x'.$$

Teorema A.2 Para $x, y, w, z \in \mathbb{R}^n$ vale:

$$x'(y \otimes y)w'z = \text{vec}'(z' \otimes x'P)\text{vec}(\text{vec}(yy')w'),$$

com P tal que $\text{vec}(y'y) = P\text{vec}(y' \otimes y')$.

A.2 Demonstração do Teorema 4.1

Esta prova trata sempre do sistema expandido (4.1), contudo, por simplicidade, a notação utilizada será do sistema não expandido (2.6), ou seja, denotaremos $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}, \tilde{G}, \tilde{L}$ e \tilde{g} por A, B, C, D, G, L e g , respectivamente. Reescrevendo a Equação (4.1), para o sistema expandido $\tilde{\Phi}$, temos que:

$$\begin{aligned} \mathcal{J}_N = & \langle X(k), C + g'(k)Dg(k) \rangle + \langle X(k+1), C + g'(k+1)Dg(k+1) \rangle \\ & + \langle L(k+2), X(k+2) \rangle + \tilde{V}(k+2), \end{aligned}$$

lembrando que $\tilde{V}(k+2) = V(k+2) + \sum_{\ell=0}^{k-1} \langle X(\ell), C + g'(\ell)Dg(\ell) \rangle$. Usando a Proposição 2.1 e reescrevendo em termos do operador \mathcal{L} auto-adjunto de τ , obtemos

$$\begin{aligned} \mathcal{J}_N = & \langle X(k), C + g'(k)Dg(k) \rangle + \langle X(k), \mathcal{L}_{g(k)}(C + g'(k+1)Dg(k+1)) \rangle \\ & + \langle \mathcal{L}_{g(k)}(\mathcal{L}_{g(k+1)}(L(k+2))), X(k) \rangle + \tilde{V}(k+2). \end{aligned} \quad (\text{A.1})$$

Na sequência são analisadas cada uma das parcelas de (A.1). Na primeira parcela, notando que

$$\begin{aligned} \text{Tr}\{X_i(k)g'(k)D_i g(k)\} &= \text{vec}'(X_i'(k))\text{vec}(g'(k)D_i g(k)) \\ &= \text{vec}'(X_i(k))[g'(k) \otimes g'(k)]\text{vec}(D_i) \\ &= [\text{vec}'(D_i) \otimes \text{vec}'(X_i(k))]\text{vec}(g'(k)g(k)) \\ &= \text{vec}'[P\text{vec}(D_i X_i(k))\text{vec}'(a'a)]\text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(a'a)] \\ &= \text{vec}'[P\text{vec}(D_i X_i(k))\text{vec}'(a'a)]\text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))] \end{aligned}$$

e denotando

$$\begin{aligned} \mu_1 &= \text{vec}'[P\text{vec}(D_i X_i(k))\text{vec}'(a'a)] \quad \text{e} \\ \xi_1 &= \text{Tr}\{X_i(k)C_i\}, \end{aligned}$$

temos

$$\begin{aligned} \langle X(k), C + g'(k)Dg(k) \rangle &= \sum_{i=1}^T \text{Tr}\{X_i(k)[C_i + g'(k)D_i g(k)]\} \\ &= \sum_{i=1}^T (\mu_1 \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))]) + \xi_1. \end{aligned} \quad (\text{A.2})$$

Expandindo a segunda parcela de (A.1)

$$\begin{aligned} & \langle X_i(k), \mathcal{L}_{g(k)}(C + g'(k+1)Dg(k+1)) \rangle \\ &= \sum_{i=1}^T \text{Tr}\{X_i'(k)[(A_i + B_i g(k))' \mathcal{E}_i(C + g'(k+1)Dg(k+1))(A_i + B_i g(k))]\} \\ &= \sum_{i=1}^T \sum_{j=1}^T p_{ij} \text{Tr}\{X_i(k)[(A_i + B_i g(k))'(C_j + g'(k+1)D_j g(k+1))(A_i + B_i g(k))]\}, \end{aligned}$$

e após algumas manipulações algébricas, temos

$$\begin{aligned} & \sum_{i=1}^T \sum_{j=1}^T p_{ij} \text{Tr}\{X_i(k)A_i' C_j A_i + X_i(k)g'(k)B_i' C_j A_i + X_i(k)A_i' C_j B_i g(k) \\ & + X_i(k)g'(k)B_i' C_j B_i g(k) + X_i(k)A_i' g'(k+1)D_j g(k+1)A_i \\ & + X_i(k)A_i' g'(k+1)D_j g(k+1)B_i g(k) + X_i'(k)g'(k)B_i' g'(k+1)D_j g(k+1)A_i \\ & + X_i(k)g'(k)B_i' g'(k+1)D_j g(k+1)B_i g(k)\}. \end{aligned} \quad (\text{A.3})$$

A seguir, cada termo de (A.3) é tratado separadamente. Temos que

$$\begin{aligned}
\text{Tr}\{X_i(k)g'(k)B'_iC_jA_i\} &= \text{Tr}\{B'_iC_jA_iX_i(k)g'(k)\} = \text{vec}'(X_i(k)A'_iC_jB_i)\text{vec}(g'(k)) \\
&= \text{vec}'[\text{vec}(\text{vec}(X_i(k)A'_iC_jB_i)a)\text{vec}'(a'a)]\text{vec}[\text{vec}(g'(k)a)\text{vec}'(a'a)] \\
&= \mu_2 \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.4}$$

onde $\mu_2 = \text{vec}'[\text{vec}(\text{vec}(X_i(k)A'_iC_jB_i)a)\text{vec}'(a'a)]$. Analogamente,

$$\begin{aligned}
\text{Tr}\{X_i(k)A'_iC_jB_i g(k)\} &= \text{vec}'(B'_iC_jA_iX_i(k))\text{vec}(g(k)) \\
&= \text{vec}'[\text{vec}(a'\text{vec}(B'_iC_jA_iX_i(k)))\text{vec}'(a'a)]\text{vec}[\text{vec}(a'g(k))\text{vec}'(a'a)] \\
&= \mu_3 \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.5}$$

de forma que $\mu_3 = \text{vec}'[\text{vec}(a'\text{vec}(B'_iC_jA_iX_i(k)))\text{vec}'(a'a)]$, e assim sucessivamente,

$$\begin{aligned}
\text{Tr}\{X_i(k)g'(k)B'_iC_jB_i g(k)\} &= \text{vec}'(X'_i(k))\text{vec}(g'(k)B'_iC_jB_i g(k)) \\
&= \text{vec}'(X_i(k))[g'(k) \otimes g'(k)]\text{vec}(B'_iC_jB_i) \\
&= [\text{vec}'(B'_iC_jB_i) \otimes \text{vec}'(X_i(k))]\text{vec}(g'(k) \otimes g'(k)) \\
&= \text{vec}'[P\text{vec}(B'_iC_jB_iX_i(k))\text{vec}'(a'a)]\text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(a'a)] \\
&= \mu_4 \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.6}$$

onde $\mu_4 = \text{vec}'[P\text{vec}(B'_iC_jB_iX_i(k))\text{vec}'(a'a)]$,

$$\begin{aligned}
\text{Tr}\{X_i(k)A'_i g'(k+1)D_j g(k+1)A_i\} &= \text{Tr}\{A_iX_i(k)A'_i g'(k+1)D_j g(k+1)\} \\
&= \text{vec}'(A_iX'_i(k)A'_i)\text{vec}(g(k+1)D_j g(k+1)) \\
&= \text{vec}'(A_iX_i(k)A'_i)[g'(k+1) \otimes g'(k+1)]\text{vec}(D_j) \\
&= [\text{vec}'(D_j) \otimes \text{vec}'(A_iX_i(k)A'_i)]\text{vec}(g'(k+1) \otimes g(k+1)) \\
&= \text{vec}'[\text{vec}(a'a)\text{vec}'(D_jA_iX_i(k)A'_i)P]\text{vec}[\text{vec}(a'a)\text{vec}(g'(k+1)g(k+1))] \\
&= \mu_5 \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.7}$$

em que $\mu_5 = \text{vec}'[\text{vec}(a'a)\text{vec}'(D_jA_iX_i(k)A'_i)P]$,

$$\begin{aligned}
\text{Tr}\{X_i(k)g'(k)B'_i g'(k+1)D_j g(k+1)A_i\} &= \text{Tr}\{A_i X_i(k)g'(k)B'_i g'(k+1)D_j g(k+1)\} \\
&= \text{vec}'(B_i g(k)X'_i(k)A'_i) \text{vec}(g'(k+1)D_j g(k+1)) \\
&= \text{vec}'(B_i g(k)X_i(k)A'_i)[g'(k+1) \otimes g'(k+1)] \text{vec}(D_j) \\
&= [\text{vec}'(D_j) \otimes \text{vec}'(B_i g(k)X_i(k)A'_i)] \text{vec}(g'(k+1) \otimes g'(k+1)) \\
&= \text{vec}'(D_j B_i g(k)X_i(k)A'_i) \text{vec}(g'(k+1) \otimes g'(k+1)) \\
&= \text{vec}'(D_j B_i g(k)X_i(k)A'_i) P \text{vec}(g'(k+1)g(k+1)) \\
&= \text{vec}'(D_j B_i a a' g(k)X_i(k)A'_i) P \text{vec}(g'(k+1)g(k+1)) \\
&= \text{vec}'(a' g(k))[X_i(k)A'_i \otimes a' B'_i D_j] P \text{vec}(g'(k+1)g(k+1)) \\
&= [\text{vec}'(g'(k+1)g(k+1)) \otimes \text{vec}'(a' g(k))] \text{vec}[(X_i(k)A'_i \otimes a' B'_i D_j)P] \\
&= \text{vec}'[(X_i(k)A'_i \otimes a' B'_i D_j)P] \text{vec}[\text{vec}(a' g(k)) \text{vec}'(g'(k+1)g(k+1))] \\
&= \mu_6 \cdot \text{vec}[\text{vec}(g'(k)g(k)) \text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.8}$$

onde $\mu_6 = \text{vec}'[(X_i(k)A'_i \otimes a' B'_i D_j)P]$,

$$\begin{aligned}
\text{Tr}\{X_i(k)A'_i g'(k+1)D_j g(k+1)B_i g(k)\} &= \text{Tr}\{B_i g(k)X_i(k)A'_i g'(k+1)D_j g(k+1)\} \\
&= \text{vec}'(A_i X_i(k)g'(k)B'_i) \text{vec}(g'(k+1)D_j g(k+1)) \\
&= \text{vec}'(A_i X_i g'(k)B'_i)[g'(k+1) \otimes g'(k+1)] \text{vec}(D_j) \\
&= [\text{vec}'(D_j) \otimes \text{vec}'(A_i X_i(k)g'(k)B'_i)] \text{vec}(g'(k+1)g(k+1)) \\
&= \text{vec}'(D_j A_i X_i(k)g'(k)B'_i) P \text{vec}(g'(k+1)g(k+1)) \\
&= \text{vec}'(D_j A_i X_i(k)g'(k) a a' B'_i) P \text{vec}(g'(k+1)g(k+1)) \\
&= \text{vec}'(g'(k)a)[a' B'_i \otimes X_i(k)A'_i D_j] P \text{vec}(g'(k+1)g(k+1)) \\
&= [\text{vec}'(g'(k+1)g(k+1)) \otimes \text{vec}'(g'(k)a)] \text{vec}[(a' B'_i \otimes X_i(k)A'_i D_j)P] \\
&= \text{vec}'[(a' B'_i \otimes X_i(k)A'_i D_j)P] \text{vec}(\text{vec}(g'(k)a) \text{vec}'(g'(k+1)g(k+1))) \\
&= \mu_7 \cdot \text{vec}[\text{vec}(g'(k)g(k)) \text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.9}$$

em que $\mu_7 = \text{vec}'[(a' B'_i \otimes X_i(k)A'_i D_j)P]$ e por fim,

$$\begin{aligned}
& \text{Tr}\{X_i(k)g'(k)B'_i g'(k+1)D_j g(k+1)B_i g(k)\} \\
&= \text{Tr}\{B_i g(k)X_i(k)g'(k)B'_i g'(k+1)D_j g(k+1)\} \\
&= \text{vec}'(B_i g(k)X_i(k)g'(k)B'_i) \text{vec}(g'(k+1)D_j g(k+1)) \\
&= \text{vec}'(B_i g(k)X_i(k)g'(k)B'_i)[g'(k+1) \otimes g'(k+1(k))]\text{vec}(D_j) \\
&= [\text{vec}'(D_j) \otimes \text{vec}'(B_i g(k)X_i(k)g'(k)B'_i)] \text{vec}(g'(k+1) \otimes g'(k+1)) \\
&= \text{vec}'(D_j B_i g(k)X_i(k)g'(k)B'_i) P \text{vec}(g'(k+1)g(k+1)) \\
&= \text{vec}'(g(k)X_i(k)g'(k))[B'_i \otimes B'_i D_j] P \text{vec}(g'(k+1)g(k+1)) \\
&= [\text{vec}'(g'(k+1)g(k+1)) \otimes \text{vec}'(g(k)X_i(k)g'(k))]\text{vec}([B'_i \otimes B'_i D_j]P) \\
&= \text{vec}'(g(k)X_i(k)g'(k)) \text{vec}'(g'(k+1)g(k+1)) \text{vec}([B'_i \otimes B'_i D_j]P) \\
&= \text{vec}'(X_i(k))[g(k) \otimes g'(k)] \text{vec}'(g'(k+1)g(k+1)) \text{vec}([B'_i \otimes B'_i D_j]P) \\
&= [\text{vec}'([B'_i \otimes B'_i D_j]P) \otimes \text{vec}'(X_i(k))P] \text{vec}[\text{vec}(g'(k)g(k)) \text{vec}'(g'(k+1)g(k+1(k)))] \\
&= \mu_8 \cdot \text{vec}[\text{vec}(g'(k)g(k)) \text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.10}$$

com $\mu_8 = \text{vec}'[\text{vec}'((B'_i \otimes B'_i D_j)P) \otimes \text{vec}'(X_i(k))P]$. Substituindo os termos de (A.4) até (A.10) em (A.3):

$$\begin{aligned}
& \langle X_i(k), \mathcal{L}_{g(k)}(C + g'(k+1)Dg(k+1)) \rangle \\
&= \sum_{i=1}^T \sum_{j=1}^T p_{ij} \left(\sum_{h=2}^8 \mu_h \cdot \text{vec}[\text{vec}(g'(k)g(k)) \text{vec}'(g'(k+1)g(k+1))] \right) + \xi_2,
\end{aligned} \tag{A.11}$$

em que $\xi_2 = \text{Tr}\{X_i(k)A'_i C_j A_i\}$. Expandindo a terceira parcela de (A.1)

$$\begin{aligned}
& \langle X(k), \mathcal{L}_{g(k)}(\mathcal{L}_{g(k+1)}(L)) \rangle = \langle X(k), \sum_{j=1}^T p_{ij} (A + Bg(k))' \mathcal{L}_{g(k+1)}(L) (A + Bg(k)) \rangle \\
&= \langle X(k), g'(k)B' \mathcal{L}_{g(k+1)}(L)A + g'(k)B' \mathcal{L}_{g(k+1)}(L)Bg(k) + A' \mathcal{L}_{g(k+1)}(L)A + A' \mathcal{L}_{g(k+1)}(L)Bg(k) \rangle \\
&= \langle X(k), g'(k)B_i (A_j + B_j g(k+1))' \mathcal{E}_j(L) (A_j + B_j g(k+1)) A_i \\
&\quad + g'(k)B'_i (A_j + B_j g(k+1))' \mathcal{E}_j(L) (A_j + B_j g(k+1)) B_j g(k) \\
&\quad + A'_i (A_j + B_j g(k+1))' \mathcal{E}_j(L) (A_j + B_j g(k+1)) A_i \\
&\quad + A'_i (A_j + B_j g(k+1))' \mathcal{E}_j(L) (A_j + B_j g(k+1)) B_i g(k) \rangle,
\end{aligned}$$

com $L = L(k+2)$ conhecido, e realizando algumas manipulações algébricas, tem-se

$$\begin{aligned}
& \langle X(k), g'(k)B'A'\mathcal{E}(L)AA + g'(k)B'A'\mathcal{E}(L)Bg(k+1)A + g'(k)B'g'(k+1)B'\mathcal{E}(L)AA \\
& + g'(k)B'g(k+1)B'\mathcal{E}(L)Bg(k+1)A + g'(k)B'A'\mathcal{E}(L)ABg(k) \\
& + g'(k)B'A'\mathcal{E}(L)Bg(k+1)Bg(k) + g'(k)B'g'(k+1)B'\mathcal{E}(L)ABg(k) + A'A'\mathcal{E}(L)AA \\
& + A'A'\mathcal{E}(L)Bg(k+1)A + A'g'(k+1)B'\mathcal{E}(L)AA + A'g'(k+1)B'\mathcal{E}(L)Bg(k+1)A \\
& + A'A'\mathcal{E}(L)ABg(k) + A'A'\mathcal{E}(L)Bg(k+1)Bg(k) + A'g'(k+1)B'\mathcal{E}(L)ABg(k) \\
& + g'(k)B'g'(k+1)B'\mathcal{E}(L)Bg(k+1)Bg(k) + A'g'(k+1)B'\mathcal{E}(L)Bg(k+1)Bg(k) \rangle.
\end{aligned} \tag{A.12}$$

Analogamente ao que foi feito com (A.3), cada termo de (A.12) é tratado separadamente a seguir. Temos que

$$\begin{aligned}
& \text{Tr}\{X_i(k)g'(k)B'_iA'_j\mathcal{E}_j(L)A_jA_i\} = \text{Tr}\{B'_iA'_j\mathcal{E}_j(L)A_jA_iX_i(k)g'(k)\} \\
& = \text{vec}'(X_i(k)A'_iA'_j\mathcal{E}_j(L)A_jB_i)\text{vec}(g'(k)) \\
& = \text{vec}'[\text{vec}(\text{vec}(X_i(k)A'_iA'_j\mathcal{E}_j(L)A_jB_i)a)\text{vec}'(a'a)]\text{vec}[\text{vec}(g'(k)a)\text{vec}'(a'a)] \\
& = \mu_9 \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.13}$$

em que $\mu_9 = \text{vec}'[\text{vec}(\text{vec}(X_i(k)A'_iA'_j\mathcal{E}_j(L)A_jB_i)a)\text{vec}'(a'a)]$,

$$\begin{aligned}
& \text{Tr}\{X_i(k)A'_iA'_j\mathcal{E}_j(L)A_jB_i g(k)\} = \text{vec}'(B'_iA'_j\mathcal{E}_j(L)A_jA_iX_i(k))\text{vec}(g(k)) \\
& = \text{vec}'[\text{vec}(a'\text{vec}'(B'_iA'_j\mathcal{E}_j(L)A_jA_iX_i(k)))\text{vec}'(a'a)]\text{vec}[\text{vec}(a'g(k))\text{vec}'(a'a)] \\
& = \mu_{10} \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.14}$$

onde $\mu_{10} = \text{vec}'[\text{vec}(a'\text{vec}'(B'_iA'_j\mathcal{E}_j(L)A_jA_iX_i(k)))\text{vec}'(a'a)]$,

$$\begin{aligned}
& \text{Tr}\{X_i(k)A'_iA'_j\mathcal{E}_j(L)B_jg(k+1)A_i\} = \text{Tr}\{A_iX_i(k)A'_iA'_j\mathcal{E}_j(L)B_jg(k+1)\} \\
& = \text{vec}'(B'_j\mathcal{E}_j(L)A_jA_iX_i(k)A_i)\text{vec}(g(k+1)) \\
& = \text{vec}'[\text{vec}(a'a)\text{vec}'(a'\text{vec}'(B'_j\mathcal{E}_j(L)A_jA_iX_i(k)A_i))]\text{vec}[\text{vec}(a'a)\text{vec}'(a'g(k+1)a)] \\
& = \mu_{11} \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.15}$$

onde $\mu_{11} = \text{vec}'[\text{vec}(a'a)\text{vec}'(a'\text{vec}'(B'_j\mathcal{E}_j(L)A_jA_iX_i(k)A_i))]$,

$$\begin{aligned}
& \text{Tr}\{X_i(k)A'_i g'(k+1)B'_j\mathcal{E}_j(L)A_jA_i\} = \text{Tr}\{B'_j\mathcal{E}_j(L)A_jA_iX_i(k)A'_i g'(k+1)\} \\
& = \text{vec}'(A_iX_i(k)A'_iA'_j\mathcal{E}_j(L)B_j)\text{vec}(g'(k+1)) \\
& = \text{vec}'[\text{vec}(a'a)\text{vec}'(\text{vec}(A_iX_i(k)A'_iA'_j\mathcal{E}_j(L)B_j)a)]\text{vec}[\text{vec}(a'a)\text{vec}'(g'(k+1)a)] \\
& = \mu_{12} \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.16}$$

com $\mu_{12} = \text{vec}'[\text{vec}(a'a)\text{vec}'(\text{vec}(A_iX_i(k)A'_iA'_j\mathcal{E}_j(L)B_j)a)]$,

$$\begin{aligned}
\text{Tr}\{X_i(k)g'(k)B'_iA_j\mathcal{E}_j(L)A_jB_ig(k)\} &= \text{vec}'(X_i(k))\text{vec}(g'(k)B'_iA_j\mathcal{E}_j(L)A_jB_ig(k)) \\
&= \text{vec}'(X_i(k))[g'(k) \otimes g'(k)]\text{vec}(B'_iA_j\mathcal{E}_j(L)A_jB_i) \\
&= [\text{vec}'(B'_iA_j\mathcal{E}_j(L)A_jB_i) \otimes \text{vec}'(X_i(k))]\text{vec}(g'(k) \otimes g'(k)) \\
&= \text{vec}'[P\text{vec}(B'_iA_j\mathcal{E}_j(L)A_jB_iX_i(k))\text{vec}'(a'a)]\text{vec}[\text{vec}(g(k)g(k))\text{vec}(a'a)] \\
&= \mu_{13} \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.17}$$

em que $\mu_{13} = \text{vec}'[P\text{vec}(B'_iA_j\mathcal{E}_j(L)A_jB_iX_i(k))\text{vec}'(a'a)]$,

$$\begin{aligned}
\text{Tr}\{X_i(k)A'_i g'(k+1)B'_j\mathcal{E}_j(L)B_jg(k+1)A_i\} &= \text{Tr}\{A_iX_i(k)A'_i g'(k+1)B'_j\mathcal{E}_j(L)B_jg(k+1)\} \\
&= \text{vec}'(A_iX_i(k)A'_i)\text{vec}(g'(k+1)B'_j\mathcal{E}_j(L)B_jg(k+1)) \\
&= \text{vec}'(A_iX_i(k)A'_i)[g'(k+1) \otimes g'(k+1)]\text{vec}(B'_j\mathcal{E}_j(L)B_j) \\
&= [\text{vec}'(B'_j\mathcal{E}_j(L)B_j) \otimes \text{vec}'(A_iX_i(k)A'_i)]\text{vec}(g'(k+1) \otimes g'(k+1)) \\
&= \text{vec}'[\text{vec}(a'a)\text{vec}'(B'_j\mathcal{E}_j(L)B_jA_iX_i(k)A'_i)P]\text{vec}[\text{vec}(a'a)\text{vec}'(g'(k+1)g(k+1))] \\
&= \mu_{14} \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.18}$$

onde $\mu_{14} = \text{vec}'[\text{vec}(a'a)\text{vec}'(B'_j\mathcal{E}_j(L)B_jA_iX_i(k)A'_i)P]$,

$$\begin{aligned}
\text{Tr}\{X_i(k)g'(k)B'_iA'_j\mathcal{E}_j(L)B_jg(k+1)A_i\} &= \text{vec}'(A_jB_ig(k)X_i(k))\text{vec}(\mathcal{E}_j(L)B_jg(k+1)A_i) \\
&= \text{vec}'(A_jB_iaa'g(k)X_i(k))\text{vec}(\mathcal{E}_j(L)B_jaa'g(k+1)A_i) \\
&= \text{vec}'(a'g(k))[X_i(k) \otimes a'B'_iA'_j][A'_i \otimes \mathcal{E}_j(L)B_ja]\text{vec}(a'g(k+1)) \\
&= \text{vec}'(a'g(k))[X_i(k)A'_i \otimes a'B'_iA'_j\mathcal{E}_j(L)B_ja]\text{vec}(a'g(k+1)) \\
&= [\text{vec}'(a'g(k+1)) \otimes \text{vec}'(a'g(k))]\text{vec}(X_i(k)A'_i \otimes a'B'_iA'_j\mathcal{E}_j(L)B_ja) \\
&= \text{vec}'(X_i(k)A'_i \otimes aB'A'E(L)Ba)\text{vec}[\text{vec}(a'g(k))\text{vec}'(a'g(k+1))] \\
&= \mu_{15} \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.19}$$

em que $\mu_{15} = \text{vec}'[X_i(k)A'_i \otimes aB'_iA'_j\mathcal{E}_j(L)B_ja]$,

$$\begin{aligned}
\text{Tr}\{X_i(k)g(k)B'_i g'(k+1)B'_j\mathcal{E}_j(L)A_jA_i\} &= \text{vec}'(B_ig(k)X_i(k))\text{vec}(Ig'(k+1)B'_j\mathcal{E}_j(L)A_jA_i) \\
&= \text{vec}'(B_iaa'g(k)X_i(k))\text{vec}(Ig'(k+1)aa'B'_j\mathcal{E}_j(L)A_jA_i) \\
&= \text{vec}'(a'g(k))[X_i(k) \otimes a'B_i][A'_iA'_j\mathcal{E}_j(L)B_ja \otimes I]\text{vec}(g'(k+1)a) \\
&= \text{vec}'(a'g(k))[X_i(k)A'_iA'_j\mathcal{E}_j(L)B_ja \otimes a'B_i]\text{vec}(g'(k+1)a) \\
&= [\text{vec}'(g'(k+1)a) \otimes \text{vec}'(a'g(k))]\text{vec}(X_i(k)A'_iA'_j\mathcal{E}_j(L)B_ja \otimes a'B_i) \\
&= \text{vec}'[X_i(k)A'_iA'_j\mathcal{E}_j(L)B_ja \otimes a'B_i]\text{vec}[\text{vec}(a'g(k))\text{vec}'(g'(k+1)a)] \\
&= \mu_{16} \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.20}$$

com $\mu_{16} = \text{vec}'[X_i(k)A'_iA'_j\mathcal{E}_j(L)B_ja \otimes a'B'_i]$,

$$\begin{aligned}
& \text{Tr}\{X_i(k)A'_iA'_j\mathcal{E}_j(L)B_jg(k+1)B_ig(k)\} \\
&= \text{Tr}\{B_ig(k)X_i(k)A'_iA'_j\mathcal{E}_j(L)B_jg(k+1)I\} \\
&= \text{vec}'(X_i(k)g'(k)B'_i)\text{vec}(A'_iA'_j\mathcal{E}_j(L)B_jg(k+1)I) \\
&= \text{vec}'(X_i(k)g'(k)aa'B'_i)\text{vec}(A'_iA'_j\mathcal{E}_j(L)B_jaa'g(k+1)I) \\
&= \text{vec}'(g'(k)a)[a'B'_i \otimes X_i(k)][I \otimes A'_iA'_j\mathcal{E}_j(L)B_ja]\text{vec}(ag(k+1)) \\
&= \text{vec}'(g'(k)a)[a'B'_i \otimes X_i(k)A'_iA'_j\mathcal{E}_j(L)B_ja]\text{vec}(ag(k+1)) \\
&= [\text{vec}(ag(k+1)) \otimes \text{vec}'(g'(k)a)]\text{vec}(a'B'_i \otimes X_i(k)A'_iA'_j\mathcal{E}_j(L)B_ja) \\
&= \text{vec}'[a'B'_i \otimes X_i(k)A'_iA'_j\mathcal{E}_j(L)B_ja]\text{vec}[\text{vec}(g'(k)a)\text{vec}'(ag(k+1))] \\
&= \mu_{17} \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.21}$$

em que $\mu_{17} = \text{vec}'[a'B'_i \otimes X_i(k)A'_iA'_j\mathcal{E}_j(L)B_ja]$,

$$\begin{aligned}
& \text{Tr}\{X_i(k)A'_ig'(k+1)B'_j\mathcal{E}_j(L)A_jB_ig(k)\} \\
&= \text{Tr}\{B_ig(k)X_i(k)A'_ig'(k+1)B'_j\mathcal{E}_j(L)A_j\} \\
&= \text{vec}'(X_i(k)g'(k)B'_i)\text{vec}(A'_ig'(k+1)B'_j\mathcal{E}_j(L)A_j) \\
&= \text{vec}'(X_i(k)g'(k)aa'B'_i)\text{vec}(A'_ig'(k+1)aa'B'_j\mathcal{E}_j(L)A_j) \\
&= \text{vec}'(g'(k)a)[a'B'_i \otimes X_i(k)][A_j\mathcal{E}_j(L)Ba \otimes A'_i]\text{vec}(g'(k+1)a) \\
&= \text{vec}'(g'(k)a)[a'B'_iA_j\mathcal{E}_j(L)Ba \otimes X_i(k)A'_i]\text{vec}(g'(k+1)a) \\
&= [\text{vec}'(g'(k+1)a) \otimes \text{vec}'(g'(k)a)]\text{vec}(a'B'_iA_j\mathcal{E}_j(L)Ba \otimes X_i(k)A'_i) \\
&= \text{vec}'[a'B'_iA'_j\mathcal{E}_j(L)B_ja \otimes X_i(k)A'_i]\text{vec}[\text{vec}(g'(k)a)\text{vec}'(g'(k+1)a)] \\
&= \mu_{18} \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.22}$$

onde $\mu_{18} = \text{vec}'[a'B'_iA'_j\mathcal{E}_j(L)B_ja \otimes X_i(k)A'_i]$,

$$\begin{aligned}
& \text{Tr}\{X_i(k)g'(k)B'_ig'(k+1)B'_j\mathcal{E}_j(L)B_jg(k+1)A_i\} \\
&= \text{Tr}\{A_iX_i(k)g'(k)B'_ig'(k+1)B'_j\mathcal{E}_j(L)B_jg(k+1)\} \\
&= \text{vec}'(B_ig(k)X_i(k)A'_i)\text{vec}(g'(k+1)B'_j\mathcal{E}_j(L)B_jg(k+1)) \\
&= \text{vec}'(B_ig(k)X_i(k)A'_i)[g'(k+1) \otimes g'(k+1)]\text{vec}(B'_j\mathcal{E}_j(L)B_j) \\
&= [\text{vec}'(B'_j\mathcal{E}_j(L)B_j) \otimes \text{vec}'(B_ig(k)X_i(k)A'_i)]\text{vec}(g'(k+1) \otimes g'(k+1)) \\
&= [\text{vec}'(B'_j\mathcal{E}_j(L)B_jB_ig(k)X_i(k)A'_i)]P\text{vec}(g'(k+1)g(k+1)) \\
&= [\text{vec}'(B'_j\mathcal{E}_j(L)B_jB_iaa'g(k)X_i(k)A'_i)]P\text{vec}(g'(k+1)g(k+1)) \\
&= \text{vec}'(a'g(k))[X_i(k)A'_i \otimes a'B_iB'_j\mathcal{E}_j(L)B_j]P\text{vec}(g'(k+1)g(k+1)) \\
&= [\text{vec}'(g'(k+1)g(k+1)) \otimes \text{vec}'(a'g(k))]\text{vec}((X_i(k)A'_i \otimes a'B_iB'_j\mathcal{E}_j(L)B_j)P) \\
&= \text{vec}'[(X_i(k)A'_i \otimes a'B_iB'_j\mathcal{E}_j(L)B_j)P]\text{vec}[\text{vec}(a'g(k))\text{vec}'(g'(k+1)g(k+1))] \\
&= \mu_{19} \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.23}$$

com $\mu_{19} = \text{vec}'[(X_i(k)A'_i \otimes a'B'_iB'_j\mathcal{E}_j(L)B_j)P]$,

$$\begin{aligned}
& \text{Tr}\{X_i(k)A'_i g'(k+1)B'_j\mathcal{E}_j(L)B_j g(k+1)B_i g(k)\} \\
&= \text{Tr}\{B_i g(k)X_i(k)A'_i g'(k+1)B'_j\mathcal{E}_j(L)B_j g(k+1)\} \\
&= \text{vec}'(A_i X_i(k)g'(k)B_i) \text{vec}(g'(k+1)B'_j\mathcal{E}_j(L)B_j g(k+1)) \\
&= \text{vec}'(A_i X_i(k)g'(k)B_i)[g'(k+1) \otimes g'(k+1)] \text{vec}(B'_j\mathcal{E}_j(L)B_j) \\
&= [\text{vec}'(B'_j\mathcal{E}_j(L)B_j) \otimes \text{vec}'(A_i X_i(k)g'(k)B_i)] \text{vec}(g'(k+1) \otimes g'(k+1)) \\
&= \text{vec}'(B'_j\mathcal{E}_j(L)B_j A_i X_i(k)g'(k)B_i) P \text{vec}(g'(k+1)g(k+1)) \\
&= \text{vec}'(B'_j\mathcal{E}_j(L)B_j A_i X_i(k)g'(k)aa'B_i) P \text{vec}(g'(k+1)g(k+1)) \\
&= \text{vec}'(g'(k)a)[a'B_i \otimes X_i(k)A'_i B'_j\mathcal{E}_j(L)B_j] P \text{vec}(g'(k+1)g(k+1)) \\
&= [\text{vec}(g'(k+1)g(k+1)) \otimes \text{vec}'(g'(k)a)] \text{vec}((a'B_i \otimes X_i(k)A'_i B'_j\mathcal{E}_j(L)B_j)P) \\
&= \text{vec}'[(a'B_i \otimes X_i(k)A'_i B'_j\mathcal{E}_j(L)B_j)P] \text{vec}[\text{vec}(g'(k)a) \text{vec}'(g'(k+1)g(k+1))] \\
&= \mu_{20} \cdot \text{vec}[\text{vec}(g'(k)g(k)) \text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.24}$$

com $\mu_{20} = \text{vec}'[(a'B_i \otimes X_i(k)A'_i B'_j\mathcal{E}_j(L)B_j)P]$,

$$\begin{aligned}
& \text{Tr}\{X_i(k)g'(k)B'_i A'_j\mathcal{E}_j(L)B_j g(k+1)B_i g(k)\} \\
&= \text{Tr}\{g(k)X_i(k)g'(k)B'_i A'_j\mathcal{E}_j(L)B_j g(k+1)B_i\} \\
&= \text{vec}'(g(k)X_i(k)g'(k)) \text{vec}(B'_i A'_j\mathcal{E}_j(L)B_j aa'g(k+1)B_i) \\
&= \text{vec}'(g(k)X_i(k)g'(k))[B'_i \otimes B'_i A'_j\mathcal{E}_j(L)B_j a] \text{vec}(aa'g(k+1)) \\
&= [\text{vec}'(aa'g(k+1)) \otimes \text{vec}'(g(k)X_i(k)g'(k))] \text{vec}(B'_i \otimes B'_i A'_j\mathcal{E}_j(L)B_j a) \\
&= [\text{vec}'(g(k)X_i(k)g'(k)) \text{vec}'(aa'g(k+1))] \text{vec}(B'_i \otimes B'_i A'_j\mathcal{E}_j(L)B_j a) \\
&= \text{vec}'(X_i(k))[g'(k) \otimes g'(k)] \text{vec}'(aa'g(k+1))] \text{vec}(B'_i \otimes B'_i A'_j\mathcal{E}_j(L)B_j a) \\
&= \text{vec}'[\text{vec}'(B'_i \otimes B'_i A'_j\mathcal{E}_j(L)B_j a) \otimes \text{vec}'(X_i(k))P] \text{vec}[\text{vec}(g'(k)g(k)) \text{vec}'(aa'g(k+1))] \\
&= \mu_{21} \cdot \text{vec}[\text{vec}(g'(k)g(k)) \text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.25}$$

onde $\mu_{21} = \text{vec}'[\text{vec}'(B'_i \otimes B'_i A'_j\mathcal{E}_j(L)B_j a) \otimes \text{vec}'(X_i(k))P]$,

$$\begin{aligned}
& \text{Tr}\{X_i(k)g'(k)B'_i g'(k+1)B'_j \mathcal{E}_j(L)A_j B_i g(k)\} \\
&= \text{Tr}\{g(k)X_i(k)g'(k)B'_i g'(k+1)B'_j \mathcal{E}_j(L)A_j B_i\} \\
&= \text{vec}'(g(k)X_i(k)g'(k))\text{vec}(B'_i g'(k+1)aa'B'_j \mathcal{E}_j(L)A_j B_i) \\
&= \text{vec}'(g(k)X_i(k)g'(k))[B'_i A'_j \mathcal{E}_j B_i a' \otimes B_i]\text{vec}(g'(k+1)a) \\
&= [\text{vec}'(g'(k+1)a) \otimes \text{vec}'(g(k)X_i(k)g'(k))]\text{vec}(B'_i A'_j \mathcal{E}_j B_i a' \otimes B_i) \\
&= \text{vec}'(g(k)X_i(k)g'(k))\text{vec}'(g'(k+1)a)\text{vec}(B'_i A'_j \mathcal{E}_j B_i a' \otimes B_i) \\
&= \text{vec}'(X_i(k))[g'(k) \otimes g'(k)]\text{vec}'(g'(k+1)a)\text{vec}(B'_i A'_j \mathcal{E}_j B_i a' \otimes B_i) \\
&= \text{vec}'[\text{vec}(B'_i A'_j \mathcal{E}_j(L)B_j a' \otimes B'_i) \otimes \text{vec}'(X_i(k))P]\text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)a)] \\
&= \mu_{22} \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.26}$$

em que $\mu_{22} = \text{vec}'[\text{vec}(B'_i A'_j \mathcal{E}_j(L)B_j a' \otimes B'_i) \otimes \text{vec}'(X_i(k))P]$, e por fim

$$\begin{aligned}
& \text{Tr}\{X_i(k)g'(k)B'_i g'(k+1)B'_j \mathcal{E}_j(L)B_j g(k+1)B_i g(k)\} \\
&= \text{Tr}\{B_i g(k)X_i(k)g'(k)B'_i g'(k+1)B'_j \mathcal{E}_j(L)B_j g(k+1)\} \\
&= \text{vec}'(B_i g(k)X_i(k)g'(k)B'_i)\text{vec}(g'(k+1)B'_j \mathcal{E}_j(L)B_j g(k+1)) \\
&= \text{vec}'(B_i g(k)X_i(k)g'(k)B'_i)[g'(k+1) \otimes g'(k+1)]\text{vec}(B'_j \mathcal{E}_j(L)B_j) \\
&= [\text{vec}'(B'_j \mathcal{E}_j(L)B_j) \otimes \text{vec}'(B_i g(k)X_i(k)g'(k)B'_i)]\text{vec}(g'(k+1) \otimes g'(k+1)) \\
&= \text{vec}'(B'_j \mathcal{E}_j(L)B_j B_i g(k)X_i(k)g'(k)B'_i)P\text{vec}(g'(k+1)g(k+1)) \\
&= \text{vec}'(g(k)X_i(k)g'(k))[B'_i \otimes B'_i B'_j \mathcal{E}_j(L)B_j]P\text{vec}(g'(k+1)g(k+1)) \\
&= [\text{vec}'(g'(k+1)g(k+1)) \otimes \text{vec}'(g(k)X_i(k)g'(k))]\text{vec}((B'_i \otimes B'_i B'_j \mathcal{E}_j(L)B_j)P) \\
&= \text{vec}'(g(k)X_i(k)g'(k))\text{vec}'(g'(k+1)g(k+1))\text{vec}((B'_i \otimes B'_i B'_j \mathcal{E}_j(L)B_j)P) \\
&= \text{vec}'(X_i(k))[g'(k) \otimes g'(k)]\text{vec}'(g'(k+1)g(k+1))\text{vec}((B'_i \otimes B'_i B'_j \mathcal{E}_j(L)B_j)P) \\
&= \text{vec}'[\text{vec}'((B'_i \otimes B'_i B'_j \mathcal{E}_j(L)B_j)P) \otimes \text{vec}'(X_i(k))P]\text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))] \\
&= \mu_{23} \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))],
\end{aligned} \tag{A.27}$$

onde $\mu_{23} = \text{vec}'[\text{vec}'((B'_i \otimes B'_i B'_j \mathcal{E}_j(L)B_j)P) \otimes \text{vec}'(X_i(k))P]$. Substituindo os termos de (A.13) até (A.27) em (A.12):

$$\begin{aligned}
& \langle \mathcal{L}_{g(k)}(\mathcal{L}_{g(k+1)}(L(k+2))), X(k) \rangle \\
&= \sum_{i=1}^T \sum_{j=1}^T p_{ij} \left(\sum_{h=9}^{23} \mu_h \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))] \right) + \xi_3,
\end{aligned} \tag{A.28}$$

em que $\xi_3 = \text{Tr}\{X(k)A'A'\mathcal{E}(L)AA\}$. Portanto, a Equação (A.1) pode ser reescrita pela soma das Equações (A.2), (A.11) e (A.28), onde a constante $\widehat{V}(k+2) = \widetilde{V}(k+2) + \sum_{m=1}^3 \xi_m$, ou seja,

para o sistema expandido $\tilde{\Phi}$, temos que:

$$\mathcal{J}_N = \sum_{i=1}^T \sum_{j=1}^T p_{ij} \left(\sum_{h=1}^{23} \mu_h \cdot \text{vec}[\text{vec}(g'(k)g(k))\text{vec}'(g'(k+1)g(k+1))] + \widehat{V}(k+2) \right),$$

com

$$\begin{aligned} \mu_1 &= \text{vec}'[P\text{vec}(D_i X_i(k))\text{vec}'(a'a)], \\ \mu_2 &= \text{vec}'[\text{vec}(\text{vec}(X_i(k)A'_i C_j B_i) a) \text{vec}'(a'a)], \\ \mu_3 &= \text{vec}'[\text{vec}(a' \text{vec}(B'_i C_j A_i X_i(k))) \text{vec}'(a'a)], \\ \mu_4 &= \text{vec}'[P\text{vec}(B'_i C_j B_i X_i(k)) \text{vec}'(a'a)], \\ \mu_5 &= \text{vec}'[\text{vec}(a'a) \text{vec}'(D_j A_i X_i(k) A'_i) P], \\ \mu_6 &= \text{vec}'[(X_i(k) A'_i \otimes a' B'_i D_j) P], \\ \mu_7 &= \text{vec}'[(a' B'_i \otimes X_i(k) A'_i D_j) P], \\ \mu_8 &= \text{vec}'[\text{vec}'((B'_i \otimes B'_i D_j) P) \otimes \text{vec}'(X_i(k)) P], \\ \mu_9 &= \text{vec}'[\text{vec}(\text{vec}(X_i(k) A'_i A'_j \mathcal{E}_j(L) A_j B_i) a) \text{vec}'(a'a)], \\ \mu_{10} &= \text{vec}'[\text{vec}(a' \text{vec}'(B'_i A'_j \mathcal{E}_j(L) A_j A_i X_i(k))) \text{vec}'(a'a)], \\ \mu_{11} &= \text{vec}'[\text{vec}(a'a) \text{vec}'(a' \text{vec}'(B'_j \mathcal{E}_j(L) A_j A_i X_i(k) A_i))], \\ \mu_{12} &= \text{vec}'[\text{vec}(a'a) \text{vec}'(\text{vec}(A_i X_i(k) A'_i A'_j \mathcal{E}_j(L) B_j) a)], \\ \mu_{13} &= \text{vec}'[P\text{vec}(B'_i A_j \mathcal{E}_j(L) A_j B_i X_i(k)) \text{vec}'(a'a)], \\ \mu_{14} &= \text{vec}'[\text{vec}(a'a) \text{vec}'(B'_j \mathcal{E}_j(L) B_j A_i X_i(k) A'_i) P], \\ \mu_{15} &= \text{vec}'[X_i(k) A'_i \otimes a B'_i A'_j \mathcal{E}_j(L) B_j a], \\ \mu_{16} &= \text{vec}'[X_i(k) A'_i A'_j \mathcal{E}_j(L) B_j a \otimes a' B'_i], \\ \mu_{17} &= \text{vec}'[a' B'_i \otimes X_i(k) A'_i A'_j \mathcal{E}_j(L) B_j a], \\ \mu_{18} &= \text{vec}'[a' B'_i A'_j \mathcal{E}_j(L) B_j a \otimes X_i(k) A_i], \\ \mu_{19} &= \text{vec}'[(X_i(k) A'_i \otimes a' B'_i B'_j \mathcal{E}_j(L) B_j) P], \\ \mu_{20} &= \text{vec}'[(a' B'_i \otimes X_i(k) A'_i B'_j \mathcal{E}_j(L) B_j) P], \\ \mu_{21} &= \text{vec}'[\text{vec}'(B'_i \otimes B'_i A'_j \mathcal{E}_j(L) B_j a) \otimes \text{vec}'(X_i(k)) P], \\ \mu_{22} &= \text{vec}'[\text{vec}(B'_i A'_j \mathcal{E}_j(L) B_j a' \otimes B'_i) \otimes \text{vec}'(X_i(k)) P], \\ \mu_{23} &= \text{vec}'[\text{vec}'((B'_i \otimes B'_i B'_j \mathcal{E}_j(L) B_j) P) \otimes \text{vec}'(X_i(k)) P], \\ \xi_1 &= \text{Tr}\{X_i(k) C_i\}, \\ \xi_2 &= \text{Tr}\{X_i(k) A'_i C_j A_i\} \quad \text{e} \\ \xi_3 &= \text{Tr}\{X(k) A' A' \mathcal{E}(L) A A\}, \end{aligned}$$

onde $L = L(k+2)$ é conhecido, P é tal que $\text{vec}(y'y) = P\text{vec}(y' \otimes y')$ e $a = [0 \dots 0 \quad 1] \in \mathcal{M}^{1, n+1}$.

Resultados Computacionais Adicionais

Neste apêndice são apresentados os resultados obtidos com as precisões $\varepsilon = 10^{-12}$ e $\varepsilon_1 = 10^{-4}$, que podem ser comparados aos resultados apresentados no Capítulo 6.

B.1 Variacional Clássico vs Variacional Modificado

Primeiramente são apresentados os percentuais de ER, ENR e ENRMI obtidos com o uso do método Variacional Clássico e do método Variacional Modificado, para o conjunto A.

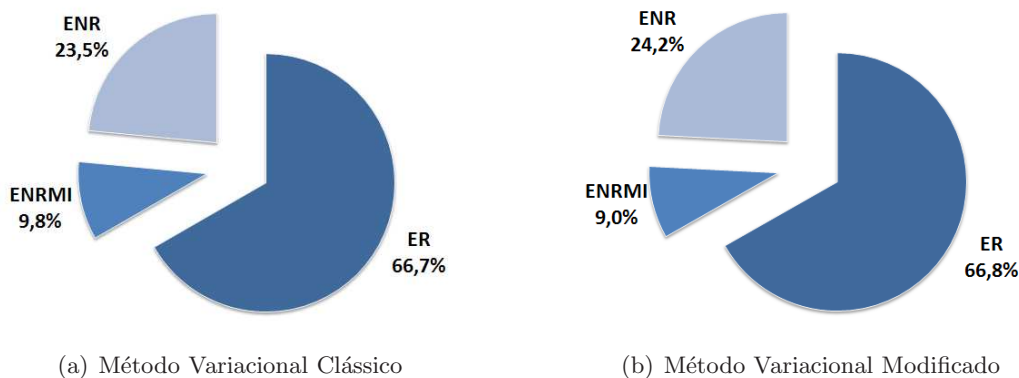


Figura B.1: Classificação dos resultados obtidos pelos métodos Variacional Clássico e Variacional Modificado, para o conjunto A.

As Figuras B.2 a B.4 apresentam o comportamento dos custos, tempo de CPU e número de iterações do método Variacional Modificado (MVM) quando comparado com o método Variacional Clássico (MV) para o conjunto A.

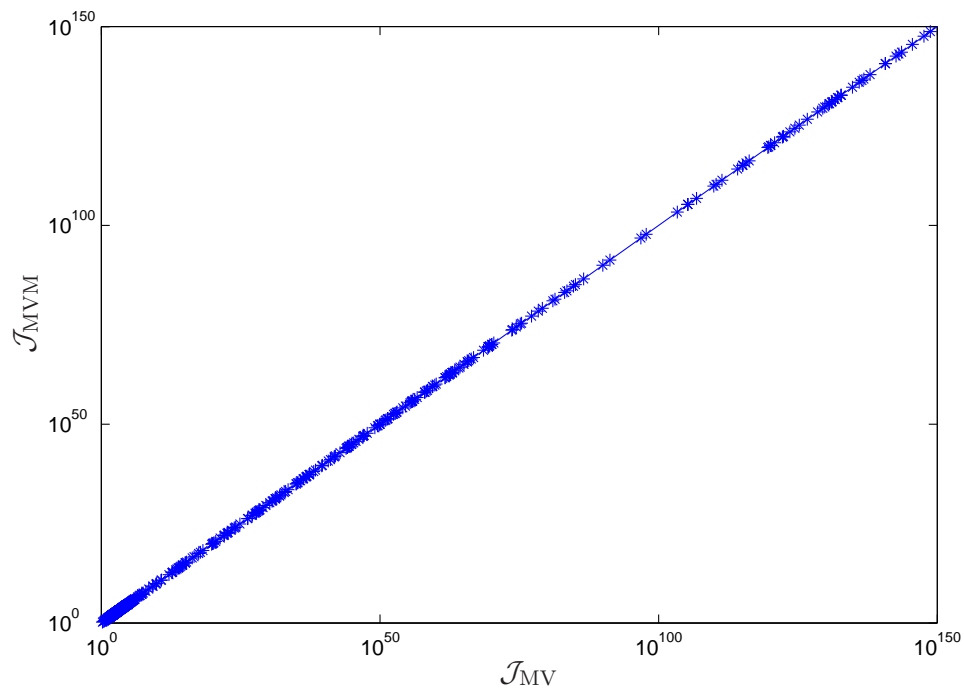


Figura B.2: Custo do MV vs custo do MVM, para o conjunto A.

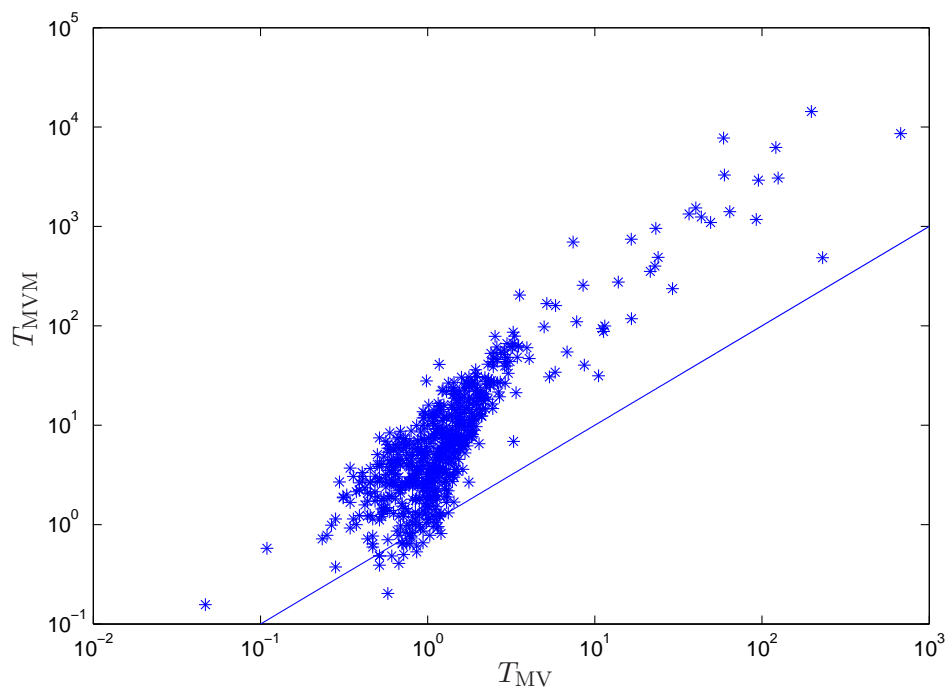


Figura B.3: Tempo de CPU do MV vs tempo de CPU do MVM, para o conjunto A.

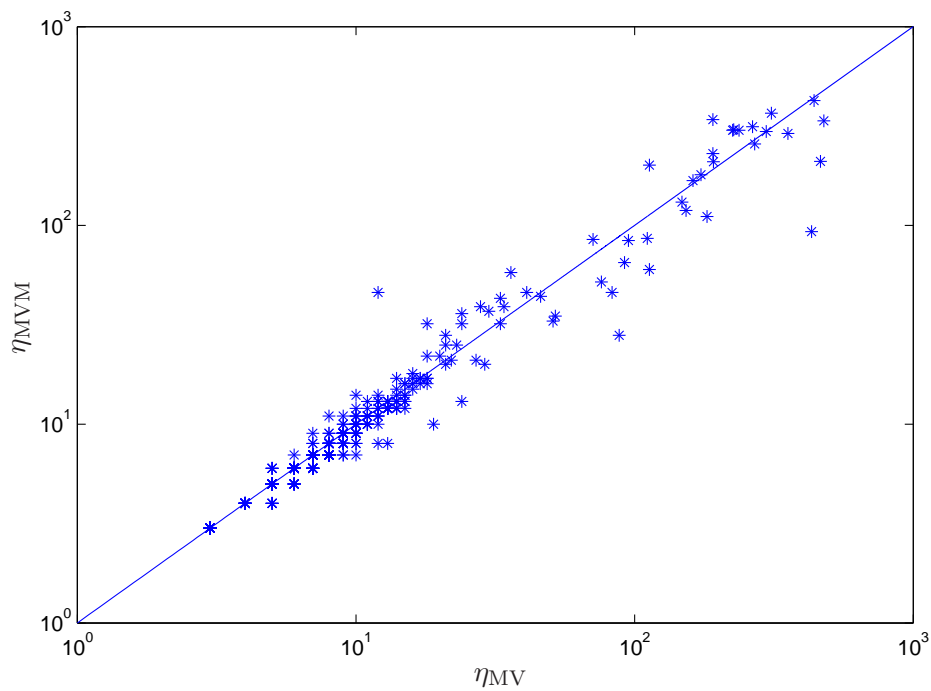


Figura B.4: Número de iterações do MV vs número de iterações do MVM, para o conjunto A.

Como na Seção 6.1, ambos os métodos resolveram 100% dos exemplos do conjunto B. As Figuras B.5 a B.7 apresentam o comportamento dos custos, tempo de CPU e número de iterações do método Variacional Modificado (MVM) quando comparado com o método Variacional Clássico (MV), para o conjunto B.

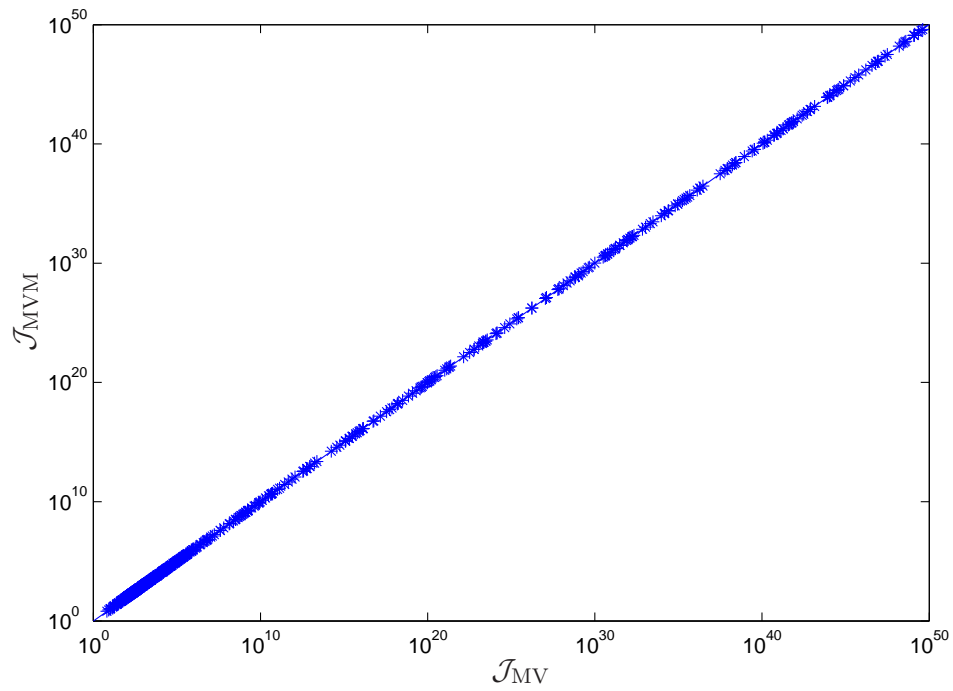


Figura B.5: Custo do MV vs custo do MVM, para o conjunto B.

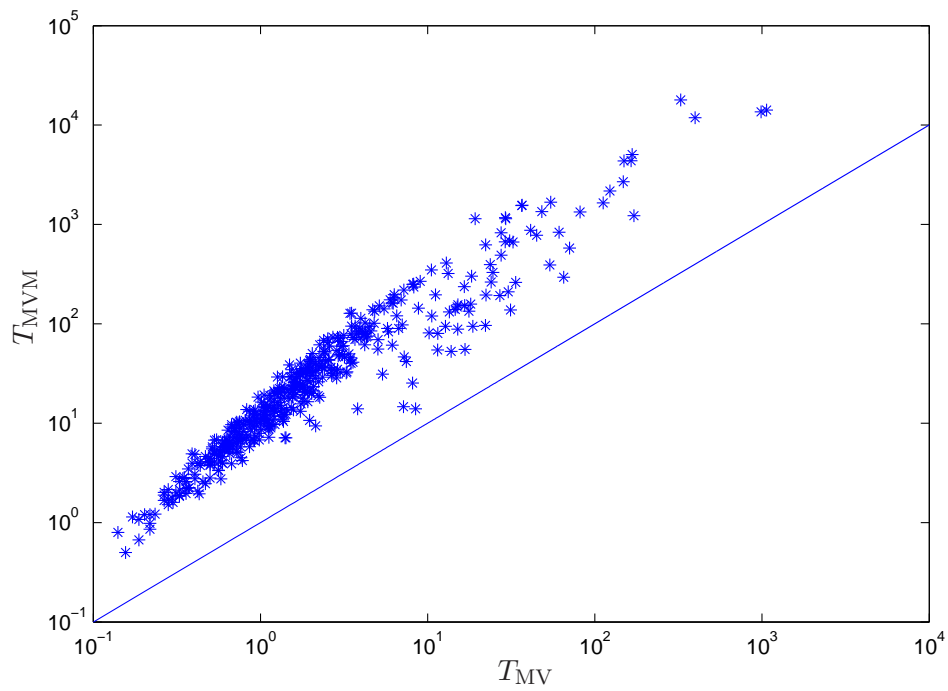


Figura B.6: Tempo de CPU do MV vs tempo de CPU do MVM, para o conjunto B.

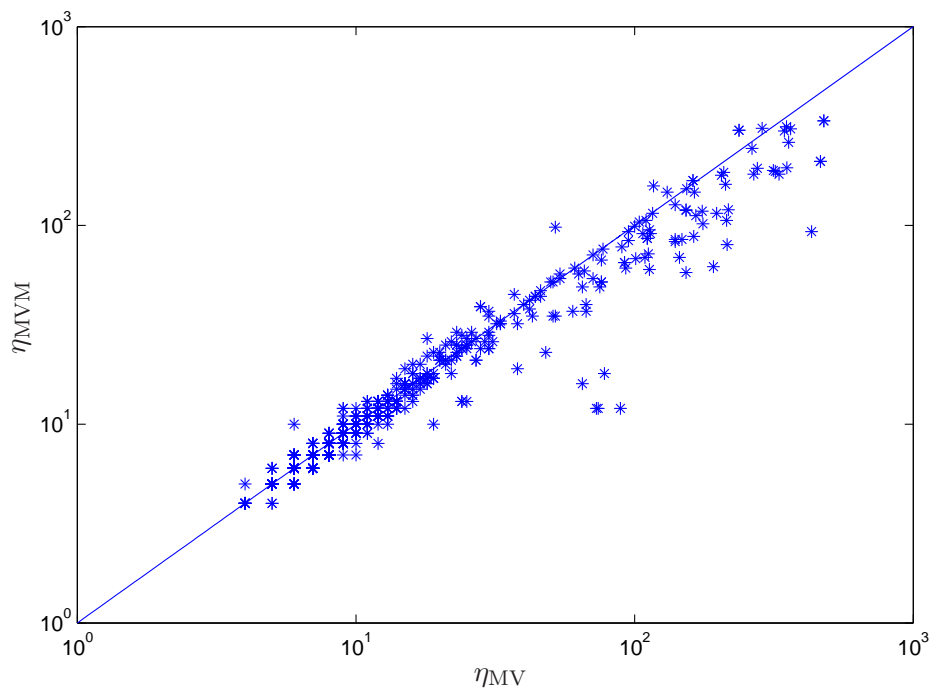


Figura B.7: Número de iterações do MV vs número de iterações do MVM, para o conjunto B.

B.2 Variacional Clássico vs Variacional Duplo com Gradiente

Os percentuais de ER, ENR e ENRMI obtidos com o uso do método Variacional Clássico e do método Variacional Duplo com Gradiente, para o conjunto A, são apresentados na Figura B.8.

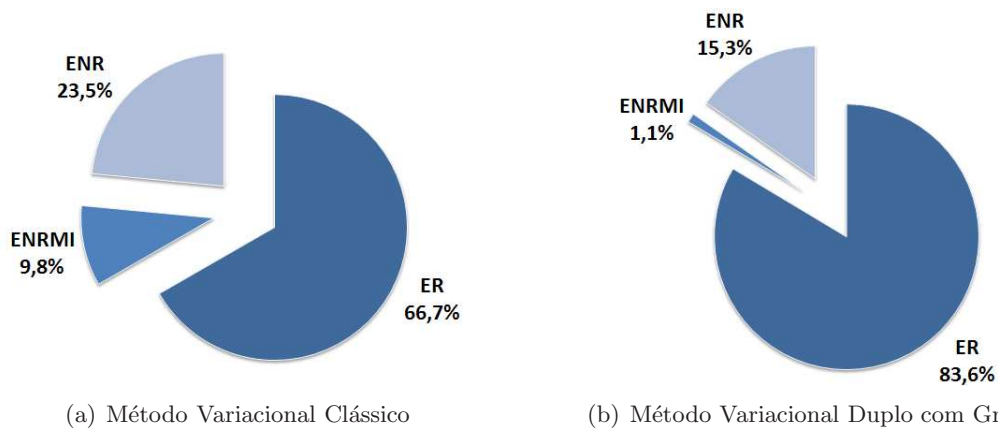


Figura B.8: Classificação dos resultados obtidos pelos métodos Variacional Clássico e Variacional Duplo com Gradiente, para o conjunto A.

As Figuras B.9 a B.11 apresentam o comportamento dos custos, tempo de CPU e número de iterações do método Variacional Duplo com Gradiente (MVDG) quando comparado com o método Variacional Clássico (MV) para o conjunto A.

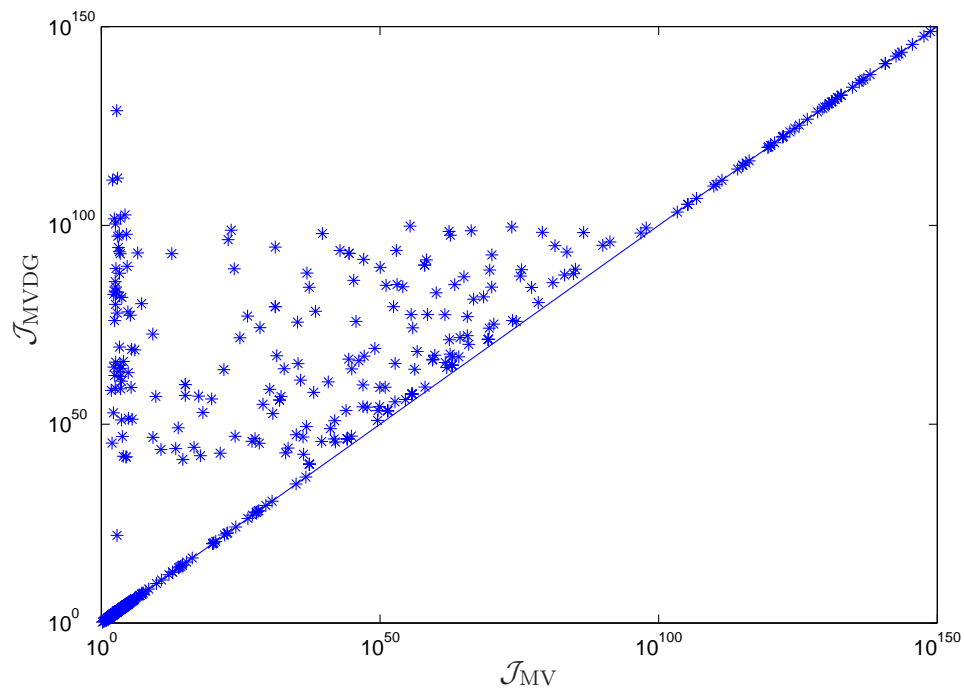


Figura B.9: Custo do MV vs custo do MVDG, para o conjunto A.

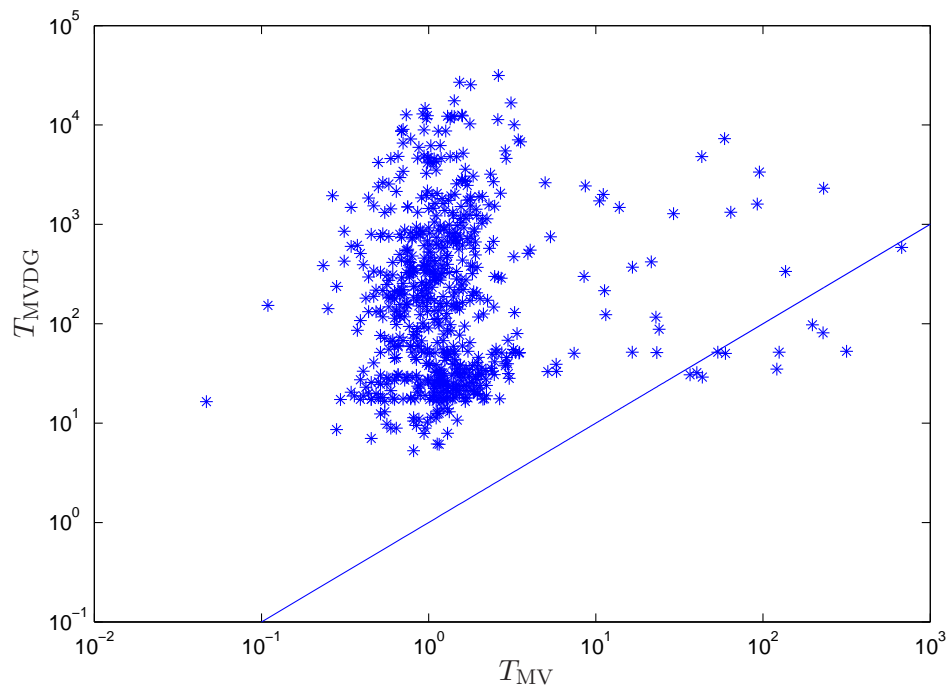


Figura B.10: Tempo de CPU do MV vs tempo de CPU do MVDG, para o conjunto A.

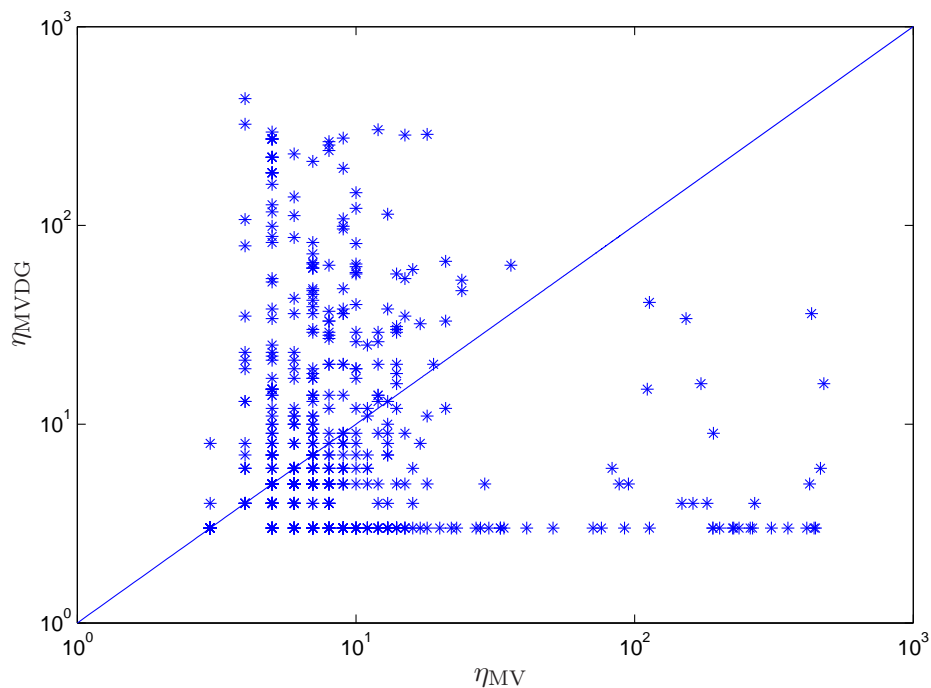


Figura B.11: Número de iterações do MV vs número de iterações do MVDG, para o conjunto A.

Como já mencionado, o método Variacional Clássico resolveu 100% dos exemplos do conjunto B. Os percentuais de ER, ENR e ENRMI obtidos com o uso do método Variacional Duplo com Gradiente para este conjunto são apresentados na Figura B.12.

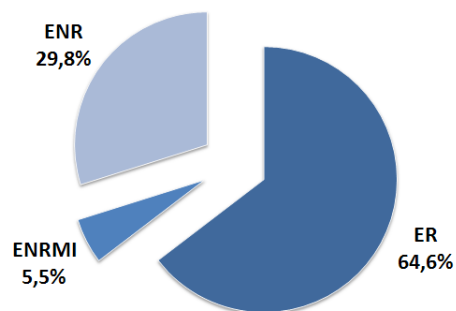


Figura B.12: Classificação dos resultados obtidos pelo método Variacional Duplo com Gradiente, para Conjunto B.

As Figuras B.13 a B.15 apresentam o comportamento dos custos, tempo de CPU e número de iterações do método Variacional Duplo com Gradiente (MVDG) quando comparado com o método Variacional Clássico (MV) para o conjunto B.

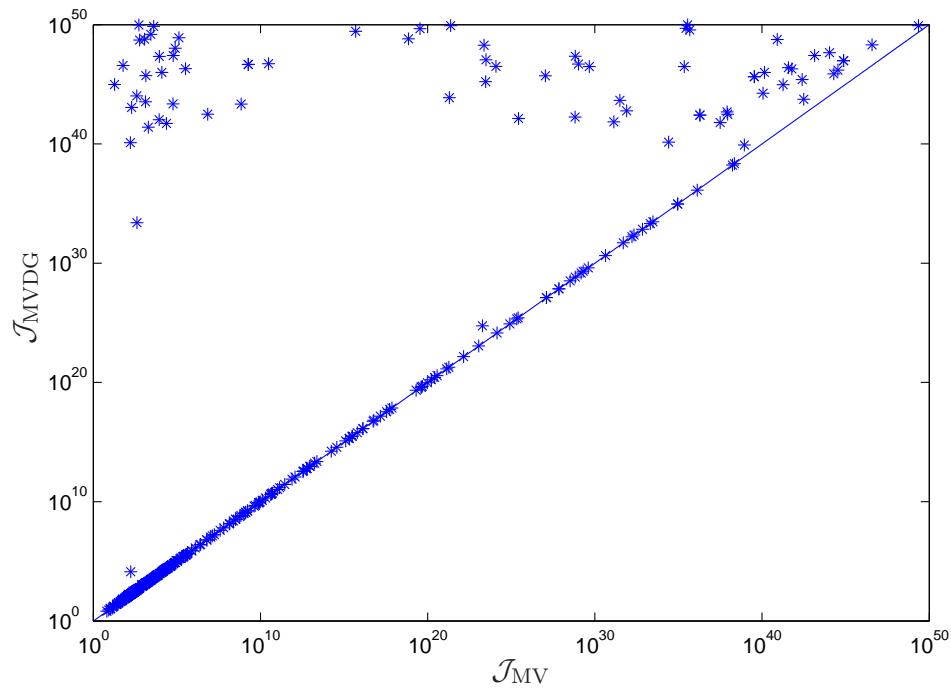


Figura B.13: Custo do MV vs custo do MVDG, para o conjunto B.

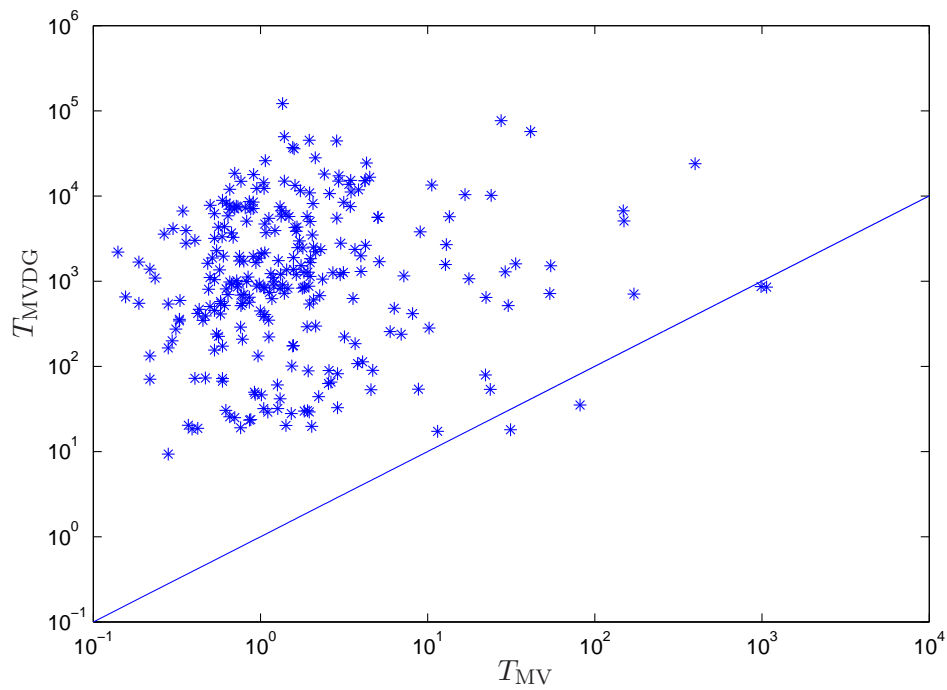


Figura B.14: Tempo de CPU do MV vs tempo de CPU do MVDG, para o conjunto B.

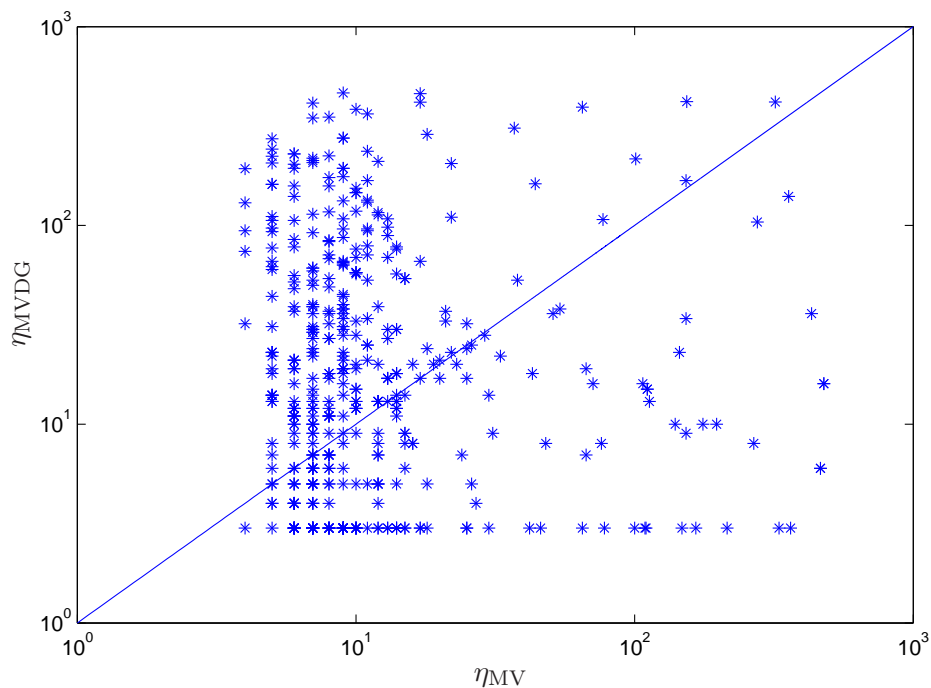


Figura B.15: Número de iterações do MV vs número de iterações do MVDG, para o conjunto B.

B.3 Variacional Clássico vs Variacional Duplo com Newton

Os percentuais de ER, ENR e ENRMI obtidos com o uso do método Variacional Clássico e do método Variacional Duplo com Newton, para o conjunto A, são apresentados na Figura B.16.

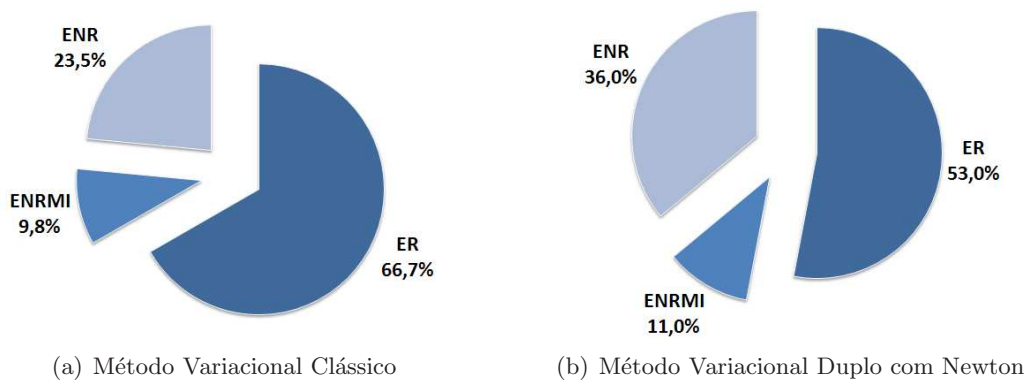


Figura B.16: Classificação dos resultados obtidos pelos métodos Variacional Clássico e Variacional Duplo com Newton.

As Figuras B.17 a B.19 apresentam o comportamento dos custos, tempo de CPU e número de iterações do método Variacional Duplo com Newton (MVDN) quando comparado com o método Variacional Clássico (MV) para o conjunto A.

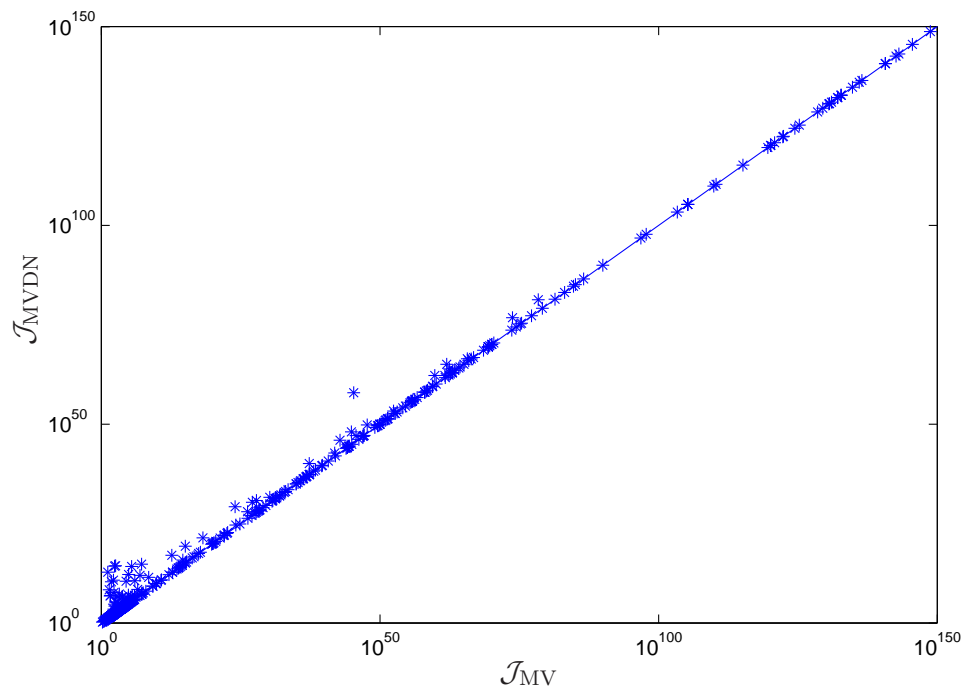


Figura B.17: Custo do MV vs custo do MVDN, para o conjunto A.

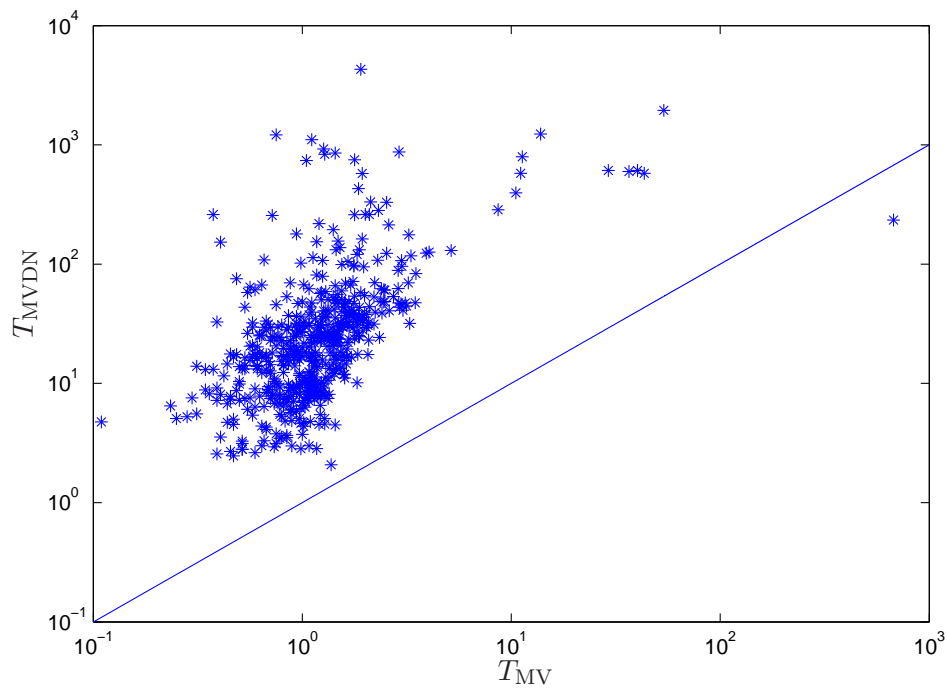


Figura B.18: Tempo de CPU do MV vs tempo de CPU do MVDN, para o conjunto A.

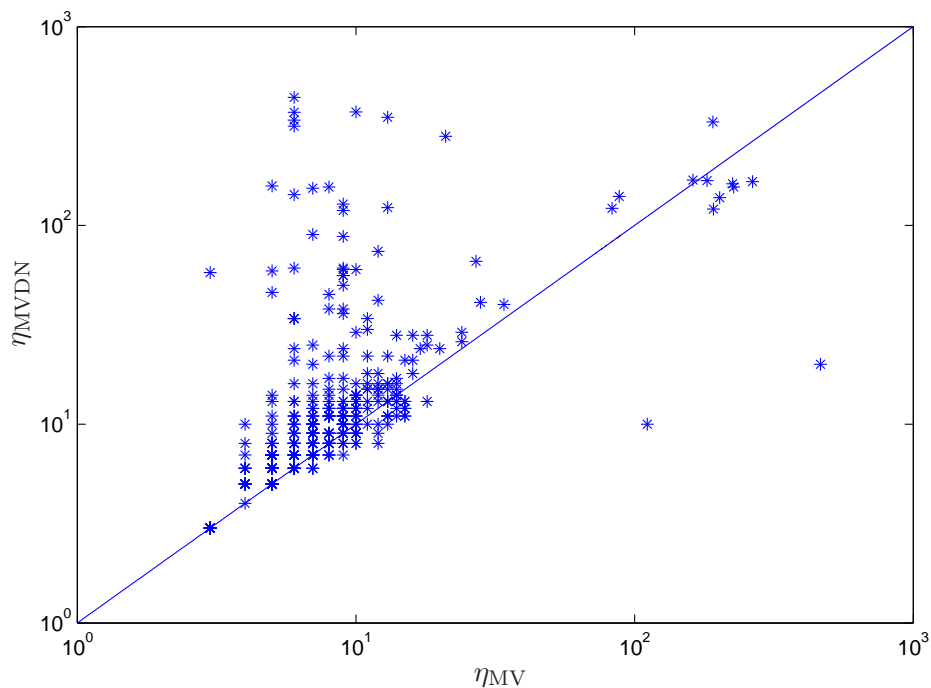


Figura B.19: Número de iterações do MV vs número de iterações do MVDN, para o conjunto A.

O método Variacional Clássico resolveu 100% dos exemplos do conjunto B. Os percentuais de ER, ENR e ENRMI obtidos com o uso do método Variacional Duplo com Newton, para o conjunto B, são apresentados na Figura B.20.

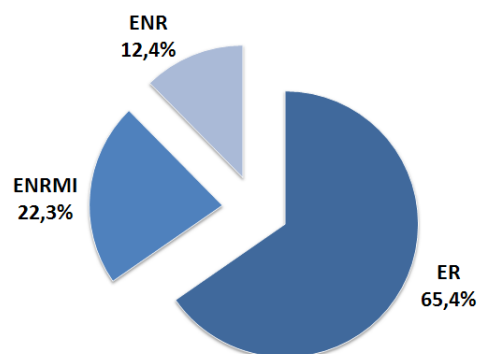


Figura B.20: Classificação dos resultados obtidos pelo método Variacional Duplo com Newton, para Conjunto B.

E as Figuras B.21 a B.23 apresentam o comportamento dos custos, tempo de CPU e número de iterações do método Variacional Duplo com Newton (MVDN) quando comparado com o método Variacional Clássico (MV) para o conjunto B.

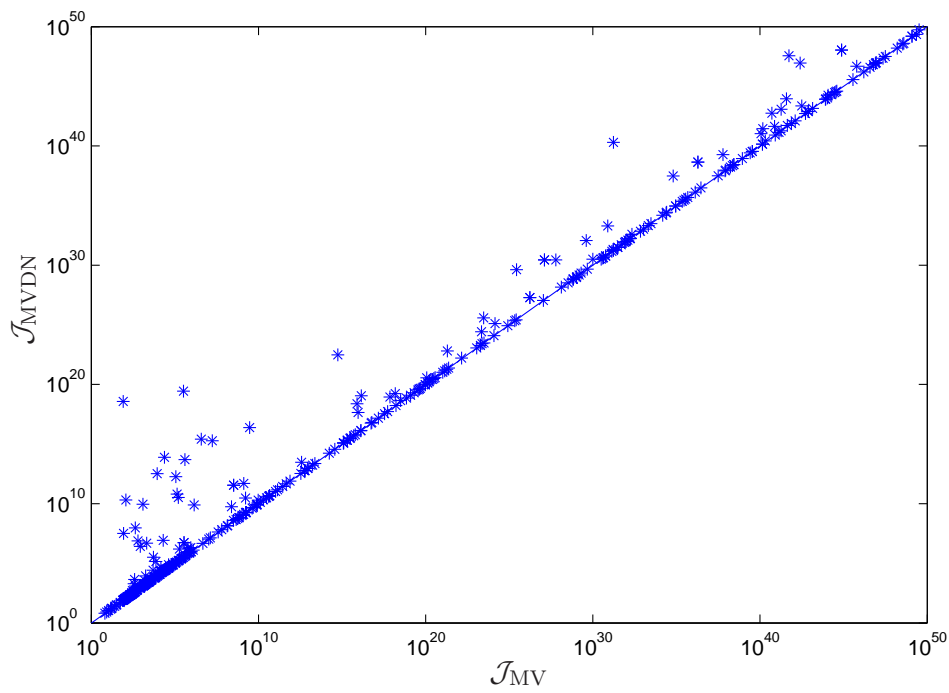


Figura B.21: Custo do MV vs custo do MVDN, para o conjunto B.

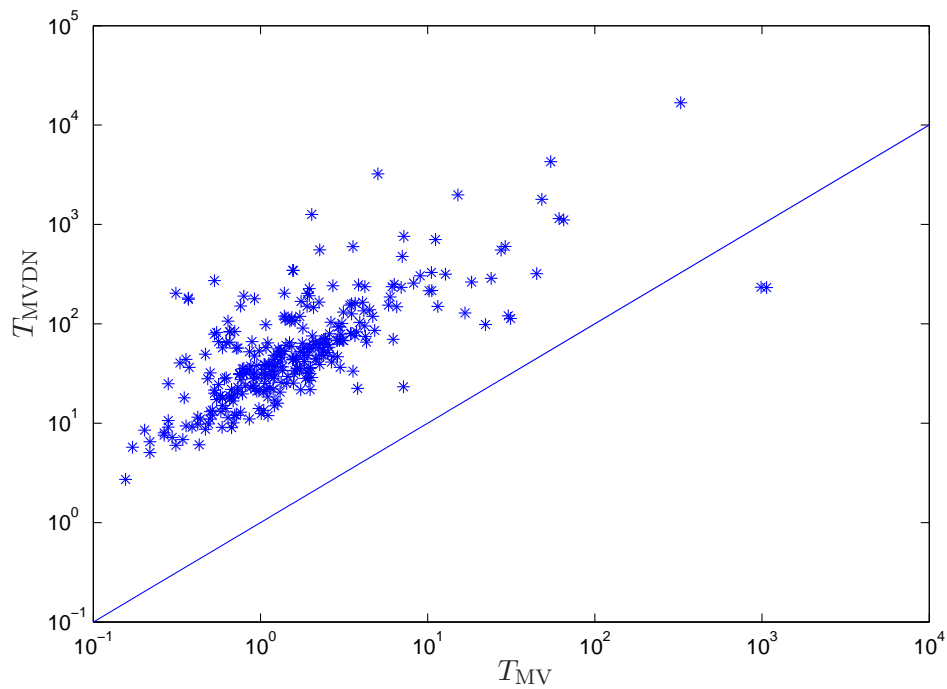


Figura B.22: Tempo de CPU do MV vs tempo de CPU do MVDN, para o conjunto B.

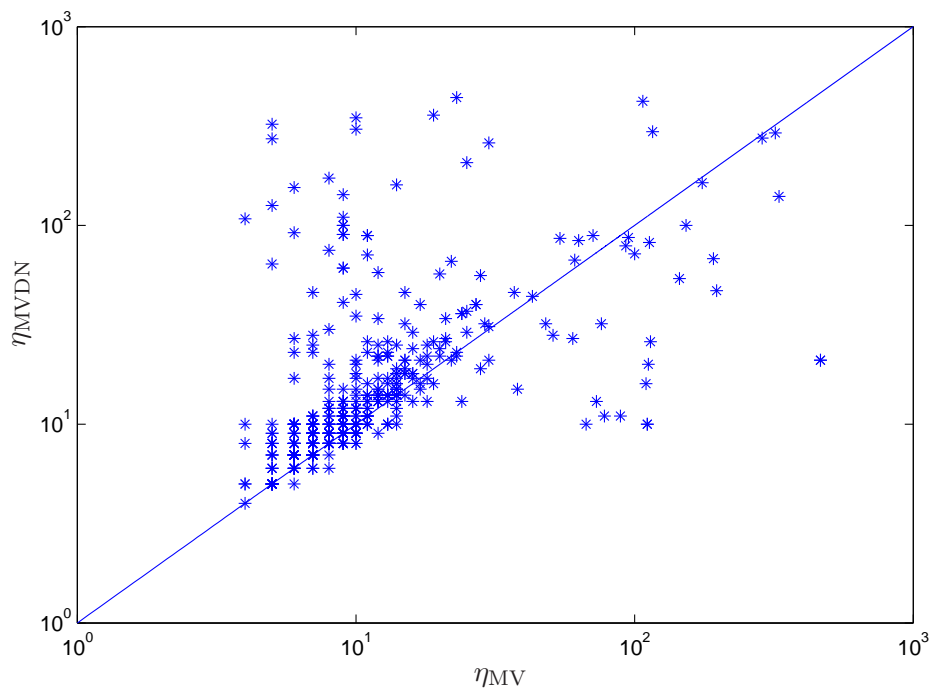


Figura B.23: Número de iterações do MV vs número de iterações do MVDN, para o conjunto B.

Código-fonte do Algoritmo 4.1

Nos códigos apresentados a seguir são usadas as seguintes notações: $\eta = \eta$, $\Sigma = \Sigma$, $\mathbb{P} = \mathbb{P}$ e $T_{\text{theta}} = T$, lembrando que $\mathcal{T} = \{1, 2, \dots, T\}$. Os operadores τ e \mathcal{E} , a matriz L e o custo \mathcal{J}_N são calculados pelas funções *operT*, *operE*, *operL* e *custo*, respectivamente. Além disso, s e r são tal que $g(k) \in \mathcal{M}^{s,r}$.

Na função *biquadratico* está adicionado um passo que incorpora ao sistema $H \cdot c_f = b$ as restrições de simetria $a_{ijkl} = a_{kjil} = a_{ilkj}$. Os resultados se mostraram indiferentes a presença destas restrições, uma vez que elas saem naturalmente dos cálculos. Elas podem ser retiradas do código simplesmente apagando todo trecho demarcado entre os comentários “Simetria”.

C.1 Códigos-fonte em MATLAB[®]

```
Function Tau = operT(X,g,k,eta)
global A B Prob r Ttheta
for i = 1:Ttheta
    soma = zeros(r);
    for j = 1:Ttheta
        MF = A(:,j) + B(:,j) × g(:,k,eta);
        soma = soma + Prob(j,i) × MF × X(:,k,j,eta) × MF';
    end
    Tau(:,k,i,eta) = soma;
end
```

Function El = operE(L,k,eta)

```

global Prob r Ttheta
for i = 1:Ttheta
    soma = zeros(r);
    for j = 1:Ttheta
        soma = soma + Prob(i,j) × L(:,j,k,eta);
    end
    El(:,i) = soma;
end

```

Function L = operL(El,g,k,eta)

```

global A B C D Ttheta
for i = 1:Ttheta
    MF = A(:,i) + B(:,i) × g(:,k,eta);
    L(:,i) = C(:,i) + g(:,k,eta)' × D(:,i) × g(:,k,eta) + MF' × El(:,i) × MF;
end

```

Function Jk = custo(g,X,k,eta,L)

```

global C D Ttheta
soma = 0;
for l = k:k+1
    soma1 = 0;
    for i = 1:Ttheta
        Aux = C(:,i) + g(:,l,eta)' × D(:,i) × g(:,l,eta);
        print = trace(Aux' × X(:,l,i,eta));
        soma1 = soma1 + print;
    end
    soma = soma + soma1;
end
soma1 = 0;
for i = 1:Ttheta
    print = trace(L(:,k+2,i,eta)' × X(:,k+2,i,eta));
    soma1 = soma1 + print;
end
Jk = soma + soma1;

```



```
%Simetria
```

```
cf = pinv(H)×b;
```