Universidade Estadual de Campinas
Instituto de Matemática, Estatística e
Computação Científica - IMECC
Departamento de Matemática

Discriminação de estados quânticos via programação semidefinida

Tatiane da Silva Evangelista
Doutorado em Matemática Aplicada

Orientador: **Prof. Dr. Carlile Campos Lavor**
Co-orientador: **Prof. Dr. Wilson Ricardo Matos Rabelo**

Abril - 2009
Campinas - SP

[1]Este trabalho contou com apoio financeiro da Fapesp - processo n^o 06/51214 – 7

Discriminação de estados quânticos via programação semidefinida

Este exemplar corresponde à redação final da tese devidamente corrigida e defendida por Tatiane da Silva Evangelista e aprovada pela comissão julgadora.

Campinas, 16 de abril de 2009

Prof. Dr. Carlile Campos Lavor

Orientador

Prof. Dr. Wilson R. M. Rabelo

Co-orientador

Banca Examinadora

1. Prof. Dr. Carlile Campos Lavor (orientador-UNICAMP)
2. Prof. Dr. Aurélio R. Leite Oliveira (UNICAMP)
3. Profa. Dra. Márcia Helena Costa Fampa (UFRJ)
4. Prof. Dr. Nelson Maculan Filho (UFRJ)
5. Prof. Dr. Renato Portugal (LNCC)

Tese apresentada ao Instituto de Matemática, Estatística e Computação Científica, UNICAMP, como requisito parcial para obtenção do título de **Doutora em Matemática Aplicada**.

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP
Bibliotecária: Maria Júlia Milani Rodrigues – CRB8a 2116

Evangelista, Tatiane da Silva
Ev14d Discriminação de estados quânticos via programação semidefinida / Tatiane da Silva Evangelista -- Campinas, [S.P. :s.n.], 2009.

Orientador : Carlile Campos Lavor ; Wilson R. M. Rabelo
Tese (doutorado) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Discriminação de estados quânticos. 2. Programação semidefinida. 3. Sistemas lineares. I. Lavor, Carlile Campos. II. Rabelo, Wilson Ricardo Matos. III. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. IV. Título.

Título em inglês: Semidefinite programming applied to quantum state discrimination.

Palavras-chave em inglês (Keywords): 1. Quantum state discrimination. 2. Semidefinite programming. 3. Linear systems.

Área de concentração: Otimização

Titulação: Doutora em Matemática Aplicada

Banca examinadora:

Prof. Dr. Carlile Campos Lavor (IMECC-UNICAMP)
Profr. Dr. Aurélio R. Leite Oliveira (IMECC-UNICAMP)
Profa. Dra. Márcia Helena Costa Fampa (UFRJ)
Prof. Dr. Nelson Maculan Filho (UFRJ)
Prof. Dr. Renato Portugal (LNCC)

Data da defesa: 16/04/2009

Programa de pós-graduação: Doutorado em Matemática Aplicada

Tese de Doutorado defendida em 16 de abril de 2009 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.

Prof. (a). Dr (a). CARLILE CAMPOS LAVOR

Prof. (a). Dr (a). AURÉLIO RIBEIRO LEITE DE OLIVEIRA

Prof. (a). Dr (a). NELSON MACULAN FILHO

Prof. (a). Dr (a). MÁRCIA HELENA COSTA FAMPA

Prof. (a) Dr. (a) RENATO PORTUGAL

"A alegria está no esforço e não
no resultado obtido. O esforço
total é a vitória total."

Mahatma Gandhi

Ao meu esposo
Goldwin,
dedico

Agradecimentos

Ao concluir este trabalho, agradeço:

Primeiramente a Deus por mais uma etapa concluída.

Ao Prof. Dr. Carlile Campos Lavor, pela amizade, oportunidade e honra de trabalharmos juntos. Sem a sua dedicação e competência, este trabalho não estaria concluído.

Aos membros da banca examinadora, pela disponibilidade, atenção despendida ao trabalho, correções, críticas e elogios. Em especial, ao Prof. Dr. Wilson R. M. Rabelo, que durante todo o desenvolvimento da tese sempre respondia às minhas dúvidas com bastante objetividade e clareza, a sua ajuda foi muito importante.

Aos professores do Imecc pelos conhecimentos passados durante o curso de doutorado. Em particular, ao Prof. Dr. Petrônio Pulino, pelos valiosos auxílios em programação em MatLab.

Ao Prof. Dr. Leo Liberti da École Polytechnique - França, pela oportunidade do estágio em seu grupo de pesquisa.

Ao meu esposo Goldwin, pela cumplicidade e amor incondicional. Por ter amor forte e tolerante para entender uma doutoranda em final de projeto; por participar junto nos meus sonhos; por ser meu companheiro "of Gold."

Aos meus pais Ademir e Marta, pelo amor, estímulo, carinho e compreensão, que através de um sorriso sincero, sempre me fizeram lembrar que os sonhos são possíveis.

Às minhas irmãs Katiane e Tays, pela inarrável oportunidade de ter convivido juntas durante a infância e junventude, compartilhando momentos alegres e sempre apoiando umas às outras nos momentos difícies.

Aos meus cunhados Guilherme e Paulo, que também acompanharam essa jornada com grande incentivo e carinho.

Ao meu novo amigo Wendel, que sempre à distância nunca mediu esforços para ajudar-me a entender o pacote CSDP, meu sincero agradecimento.

Aos meus colegas e amigos do grupo de pesquisa, Carina, Cristiano e Nolmar, pela interação, convívio e a inegável ajuda sempre que precisei.

Às amigas Marta e Sônia pela amigável convivência durante a minha estadia em Campinas.

À FAPESP que tornou possível esta pesquisa através do apoio financeiro e científico. Meu sincero agradecimento ao assessor deste projeto.

Finalmente, meu agradecimento a todos que direta ou indiretamente contribuíram para a realização deste trabalho: professores, funcionários, amigos e familiares.

Resumo

Neste trabalho, apresentamos um novo algoritmo para realizar a discriminação ótima de N estados quânticos puros não-ortogonais, que fornece o melhor conjunto de medidas POVM para o problema, através da extensão do espaço de Hilbert de N para $2N - 1$ dimensões. O algoritmo é baseado na programação semidefinida e na solução de sistemas lineares. O algoritmo foi implementado em Matlab e apresentou bons resultados computacionais.

Palavras-chave: discriminação quântica, programação semidefinida, sistemas lineares.

Abstract

In this work, we propose a new algorithm to perform the optimal discrimination of N non-orthogonal pure quantum states. This algorithm obtains the best set of POVM measurements for the problem, through the extension of the Hilbert space of N to $2N-1$ dimensions. The algorithm is based on semidefinite programming and on the solution of linear systems. The algorithm was implemented in Matlab and presented good computational results.

Keywords: quantum discrimination, semidefinite programming, linear systems.

CONTEÚDO

Introdução

Em meados da década de 80 do século XX, Benioff percebeu que as leis da física aparentemente não impunham qualquer barreira à redução do tamanho dos computadores [1]. Dessa forma, o único limite para tal miniaturização ocorreria quando os bits assumissem os mesmos princípios que regem a natureza do mundo microscópico, ou seja, os princípios quânticos.

Quase três décadas de intensa pesquisa já se passaram desde as especulações de Benioff; tempo suficiente para transformar a idéia em projetos de pesquisas milionários por todo o mundo, envolvendo cientistas cobiçosos por implementar os computadores quânticos ou para definitivamente contestá-los. Embora ainda não se tenha previsão de quando teremos resultados conclusivos, a teoria da informação quântica tem atraído investimentos financeiros e intelectuais de diversas áreas afins como matemática, fisíca e computação, devido às grandes vantagens sobre a computação clássica.

Um dos problemas importantes relacionados à teoria da informação quântica está vinculado a discriminação de estados quânticos puros não-ortogonais. Esta indistinguibilidade de estados quânticos não-ortogonais está no cerne da computação quântica e da informação quântica, onde o fato que estados quânticos possuem informações inacessíveis à medições desempenham papel central em algoritmos quânticos e criptografia quântica [4, 8, 11, 20].

Eldar [5] mostrou que a melhor medição para N estados não-ortogonais pode ser formulada com um problema de programação semidefinida [2, 21]. Baseando-se nesta formulação, Rabelo [17] propôs um algoritmo, chamado de Algoritmo Ótimo Discriminador (AOD), que implementa tal formulação.

O objetivo deste trabalho é propor um novo algoritmo baseado no AOD, com o intuito de simplificar os cálculos, melhorar os resultados computacionais e torná-lo mais compreensível pela comunidade matemática. O algoritmo foi implementado em MatLab e aplicado com sucesso em problemas com dimensões maiores que os da literatura.

O Capítulo 1 desta tese é introdutório, apresentando uma revisão dos conceitos básicos da mecânica quântica, necessários para a compreensão do algoritmo proposto.

Os Capítulos 2 e 3 descrevem a formulação do problema usando a programação semidefinida e o novo algoritmo aqui proposto, respectivamente.

Antes das conclusões, o Capítulo 4 apresenta os resultados computacionais.

CAPÍTULO 1

Conceitos básicos de Mecânica Quântica

A Mecânica Quântica é a mais completa e precisa descrição conhecida do mundo microscópico. Ela é também a base para a Computação e a Informação Quântica. Neste sentido, este capítulo apresenta os elementos da Mecânica Quântica necessários para esta tese.

1.1 O bit quântico

Assim como a Computação Clássica usa o chaveamento de sinais elétricos para codificar a informação na linguagem dos bits 0 e 1, a Computação Quântica, analogamente, codifica a informação usando sistema físico de dois níveis. O bit é, então, substituído pelo bit quântico, o q-bit, normalmente representados na base computacional $|0\rangle$ e $|1\rangle$, onde estes vetores são definidos por

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{e} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Em Mecânica Quântica, vetor é também chamado de estado. Usaremos os dois termos com o mesmo significado.

Sendo esses estados vetores do espaço de Hilbert, surge a primeira diferença fundamental entre os computadores clássicos e os computadores quânticos: enquanto que os computadores clássicos exclusivamente utilizam 0's ou 1's, os computadores quânticos admitem combinações lineares dos vetores $|0\rangle$ e $|1\rangle$ (superposição), ou seja,

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1.1}$$

onde α, β são coeficientes complexos (chamados de amplitudes) e satisfazem a condição de normalização dada por

$$|\alpha|^2 + |\beta|^2 = 1. \tag{1.2}$$

Dessa forma, a Computação Quântica não dispõe de dois estados apenas, mas de tantos quantos forem as combinações possíveis de α e β que satisfazem a Equação (1.2). Isso faz com que a quantidade de informação que pode ser armazenada no estado $|\varphi\rangle$ seja infinita. Entretanto, essa informação está no nível quântico. Para torná-la acessível no nível clássico, precisamos fazer uma medida.

A Mecânica Quântica diz que o processo de medida altera o estado de um q-bit $|\varphi\rangle$, fazendo-o assumir o estado $|0\rangle$ com probabilidade $|\alpha|^2$, ou o estado $|1\rangle$ com probabilidade $|\beta|^2$ (isto significa que os valores α e β não podem ser conhecidos através de uma medida). Resumindo: matematicamente, um q-bit é um vetor unitário num espaço de Hilbert.

A Tabela 1.1 fornece um resumo da notação-padrão utilizada em Mecânica Quântica para conceitos de Álgebra Linear. Este tipo de notação é chamado de *notação de Dirac*.

Notação	Descrição
z^*	Conjugado complexo de z.
$\|\varphi\rangle$	Vetor. Também chamado de ket.
$\langle\varphi\|$	Vetor dual de $\|\varphi\rangle$. Também chamado de bra.
$\langle\varphi\|\phi\rangle$	Produto escalar entre $\|\varphi\rangle$ e $\|\phi\rangle$.
$\|\varphi\rangle \otimes \|\phi\rangle$	Produto tensorial entre $\|\varphi\rangle$ e $\|\phi\rangle$.
A^*	Complexo conjugado da matriz A.
A^t	Matriz transposta de A.
$A^\dagger$	Matriz adjunta, ou seja, $A^\dagger = (A^t)^*$.

Tabela 1.1: Resumo da notação-padrão utilizada em Mecânica Quântica

1.2 A medição

Sabemos que para tornar a informação quântica acessível, precisamos fazer uma medição.

Por exemplo, considere o ensemble $\{(\mu_1, |Q_1\rangle), (\mu_2, |Q_2\rangle)\}$ num espaço de Hilbert de dimensão $N = 2$, onde

$$|Q_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle,$$
$$|Q_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle,$$

tal que $|\alpha_i|^2 + |\beta_i|^2 = 1$, para $i = 1, 2$.

Agora, imagine que este ensemble é colocado numa caixa hermeticamente fechada, para garantir que os estados estejam isolados do resto universo, ou seja, eles não interagem com o ambiente ou com qualquer outro ensemble que se encontre em suas vizinhanças.

Realizar uma medição significa, nesse caso, abrir a caixa e olhar para o ensemble. Segundo o postulado da medida (ver Seção 1.3), quando abrimos a caixa, o processo de medição na base computacional destrói a superposição, dando como resultado o estado $|0\rangle$ ou $|1\rangle$.

A pergunta é: quem está dentro da caixa, $|Q_1\rangle$ ou $|Q_2\rangle$? Ou seja, como fazer a distinção entre os estados? Se os estados são ortogonais não há problema, pois podemos construir uma base usando os próprios estados.

Para este tipo de problema, propomos o uso do Algoritmo Discriminador Quântico, que faz uma extensão do espaço de Hilbert inicial, transformando os estados quânticos não-ortogonais de entrada em uma configuração final discriminável.

1.3 Postulados da Mecânica Quântica

A Mecânica Quântica pode ser descrita como uma teoria matemática, governada por um conjunto de axiomas ou postulados, onde as implicações desses postulados descrevem o comportamento dos sistemas quânticos. Descreveremos agora, o operador densidade e os postulados da Mecânica Quântica baseados nesse operador.

1.3.1 Operador densidade

O operador densidade é usado para indicar que nosso conhecimento é incompleto devido às imperfeições na preparação dos estados quânticos, ou devido à nossa impossibilidade de se ter um conhecimento completo do estado quântico do sistema (por vezes, só temos acesso a uma parte do sistema total).

Quando conhecemos o vetor de estado do sistema, dizemos que o sistema é descrito por um estado puro. Por exemplo:

$$|\varphi\rangle = a|\varphi_1\rangle + b|\varphi_2\rangle, \tag{1.3}$$

onde $|\varphi_1\rangle, |\varphi_2\rangle \in \mathbb{C}^2$ e $a, b \in \mathbb{C}$.

Existem situações nas quais não se sabe em que estado o sistema se encontra, pois há apenas uma probabilidade do sistema ser encontrado no estado $|\varphi\rangle$. Mais precisamente, suponha que um sistema quântico esteja em algum estado $|\varphi_i\rangle$ de um conjunto de estados, com respectiva probabilidade μ_i. Assim, definimos:

- Um **ensemble de estados puros** como o conjunto:

$$\{(\mu_i, |\varphi_i\rangle) \quad : \quad \text{para} \quad i = 1, \ldots, N\} \quad \text{tal que} \quad \sum_i \mu_i = 1.$$

- O **operador densidade (ou matriz densidade)** do sistema como:

$$\rho = \sum_{i=1}^{N} \mu_i |\varphi_i\rangle\langle\varphi_i|.$$

1.3.2 Os postulados

A seguir, fornecemos os postulados da Mecânica Quântica utilizando o operador densidade.

O Postulado 1 estabelece a "arena de trabalho" da Mecânica Quântica, especificando como o estado de um sistema isolado deve ser descrito.

Postulado 1: *Associado a qualquer sistema físico (isolado), existe um espaço de Hilbert, conhecido como espaço de estados do sistema. Um estado (vetor) desse sistema é completamente descrito por um operador densidade ρ, tal que $tr(\rho) = 1$ e $\rho \geq 0$. Se o sistema está no estado ρ_i com probabilidade μ_i tal que $\sum_i \mu_i = 1$, para $i = 1, \ldots, N$, então o seu operador densidade será:*

$$\rho = \sum_{i=1}^{N} \mu_i \rho_i.$$

O Postulado 2 nos diz que a dinâmica de um sistema fechado é governada por transformações unitárias.

Postulado 2: *A evolução de um sistema quântico fechado é descrita por transformações unitárias, ou seja, o estado ρ_1 de um sistema em um instante t_1 está relacionado ao estado ρ_2 em um instante t_2 por um operador unitário U que depende somente de t_1 e t_2:*

$$\rho_2 = U \rho_1 U^{\dagger}.$$

Foi postulado que sistemas quânticos fechados evoluem de acordo com transformações unitárias. Para sistemas que não interagem com outros sistemas, isto está correto, mas existirão momentos em que os sistemas físicos externos deverão observá-los para verificar o que está acontecendo dentro deles. Nesse caso, haverá uma intervenção que acaba com o isolamento dos sistemas e que não é necessariamente descrita por uma transformação unitária.

Para explicar o que acontece nessas situações, introduz-se o Postulado 3, que fornece o modo como as medidas sobre sistemas quânticos devem ser descritas.

Postulado 3: *As medidas quânticas são descritas por operadores $\{M_j\}$ que satisfazem*

a seguinte equação de completitude:

$$\sum_j M_j^\dagger M_j = I.$$

Se o estado do sistema antes da medida for ρ, então a probabilidade de o resultado j ocorrer será dada por

$$p(j) = tr(M_j^\dagger M_j \rho)$$

e o estado do sistema após a medida será

$$\frac{M_j \rho M_j^\dagger}{tr(M_j^\dagger M_j \rho)}.$$

Para finalizar, fornecemos o Postulado 4 que nos diz como espaços de estados de sistemas quânticos diferentes devem ser combinados para formar sistemas compostos. Para apresentar o postulado, usaremos a definição abaixo de produto tensorial entre operadores densidade.

Definição 1.1. *Dadas as matrizes $A \in \mathbb{C}^{m\times n}$ e $B \in \mathbb{C}^{p\times q}$, o produto tensorial destas matrizes, denotado por $A \otimes B \in \mathbb{C}^{mp\times nq}$, é dado da seguinte forma:*

$$A \otimes B = \begin{bmatrix} A_{11}B & A_{12}B & \dots & A_{1n}B \\ A_{21}B & A_{22}B & \dots & A_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1}B & A_{m2}B & \dots & A_{mn}B \end{bmatrix},$$

onde A_{ij} representa o elemento da linha i e da coluna j da matriz A.

Postulado 4: *O estado composto por vários sistemas quânticos é o produto tensorial dos espaços de estados das suas componentes, isto é, se as componentes forem dadas por ρ_i para $i = 1, \dots, N$, então o sistema composto será $\rho_1 \otimes \dots \otimes \rho_N$.*

1.4 Medidas POVM

O Postulado 3 envolve dois elementos. Primeiro, fornece uma regra que descreve a estatística de medidas, ou seja, as diversas possibilidades dos diferentes resultados. Segundo, diz qual

o estado do sistema após a medida. Contudo, em algumas aplicações, a probabilidade dos diferentes resultados da medida é de pouco interesse, sendo mais importante o estado do sistema após a medida. Esse é, por exemplo, o caso em que a medida é feita somente uma vez e o experimento é concluído. Para tais casos, existe uma ferramenta matemática conhecida como *formalismo POVM* [14], que é especialmente bem adaptado para a análise de medidas.[1] Para isso, definimos operadores conhecidos como operadores de detecção Π_j.

Definição 1.2. *Os operadores de detecção Π_j são chamados elementos de POVM se satisfazem as seguintes condições:*

- *A probabilidade de se obter o resultado rotulado pelo índice j é dada por $p(j|\rho) = tr(\Pi_j \rho)$, onde ρ é o operador densidade do sistema.*
- *Todos os operadores de detecção são hermitianos e positivos semidefinidos, isto é, $\Pi_j = \Pi_j^\dagger$ e $\Pi_j \geq 0$, para todo j.*
- *Os operadores de detecção satisfazem a condição de completitude, ou seja, $\sum_j \Pi_j = I$.*

1.5 Estratégia UD

Usando o formalismo de medidas POVM, Ivanovic [10] definiu uma estratégia para o problema que consiste em distinguir estados quânticos não-ortogonais linearmente independentes sem ambiguidade, chamada de estratégia UD[2], descrita a seguir.

Consideremos que o sistema quântico é dado pelo ensemble $\{\mu_i, |Q_i\rangle\}$, para $i = 1, \ldots, N$, em um espaço de Hilbert de dimensão N. Para detectar os estados do sistema, a medição é construída formando $N + 1$ operadores de detecção $\{\Pi_i, 0 \leq i \leq N\}$ satisfazendo

$$\sum_{i=0}^{N} \Pi_i = I. \tag{1.4}$$

Estes operadores são construídos para que os estados sejam medidos corretamente ou a medição declara um resultado inconclusivo. Então, cada operador Π_i corresponde à de-

[1] a sigla POVM significa "positive operator-valued measure".

[2] a sigla UD significa "unambiguous discrimination".

tectação correta dos estados $|Q_i\rangle$, para $i = 1, \ldots, N$, e $\Pi_0 = I - \sum_{i=1}^{N} \Pi_i$ corresponde a um resultado inconclusivo. Os operadores de detecção devem obedecer:

$$\langle Q_i|\Pi_j|Q_i\rangle = p_i\delta_{ij},$$

para $i, j = 1, \ldots, N$, $0 \leq p_i \leq 1$, onde δ_{ij} é o delta de Kronecker.

Eldar [5] mostrou que a estratégia UD pode ser formulada como um problema de programação semidefinida [21, 2], onde os operadores de detecção são expressos da seguinte forma:

$$\Pi_i = p_i\mathcal{C}_i, \quad 1 \leq i \leq N, \tag{1.5}$$

onde $\mathcal{C}_i = |\tilde{Q}_i\rangle\langle\tilde{Q}_i|$ e os vetores $|\tilde{Q}_i\rangle$ não são normalizados e estão em um espaço de Hilbert de dimensão N. Além disso, representam os estados recíprocos associados aos estados $|Q_i\rangle$, para $i = 1, \ldots, N$, ou seja, $\langle\tilde{Q}_i|Q_j\rangle = \delta_{ij}$, para $1 \leq i, j \leq N$.

Dada a matriz Ψ, cujas colunas são os estados $|Q_i\rangle$, os estados $|\tilde{Q}_i\rangle$ são as colunas da matriz $\tilde{\Psi}$, dada por:

$$\tilde{\Psi} = \Psi(\Psi^t\Psi)^{-1}, \tag{1.6}$$

onde a matriz $\tilde{\Psi}$ é a pseudo-inversa de Moore-Penrose de Ψ [7].

Se os estados $|Q_i\rangle$ são preparados com probabilidades a priori μ_i, tais que $\sum_i \mu_i = 1$, então a probabilidade total de detecção correta do estado é:

$$P_D = \sum_{i=1}^{N} \mu_i\langle Q_i|\Pi_i|Q_i\rangle = \sum_{i=1}^{N} \mu_i p_i = \langle\mu|p\rangle, \tag{1.7}$$

onde $\mu = \sum_{i=1}^{N} \mu_i|i\rangle$ e $p = \sum_{i=1}^{N} p_i|i\rangle$.

Portanto, o problema consiste em escolher operadores de detecção $\Pi_i = p_i\mathcal{C}_i$, ou equivalentemente, encontrar os valores $p_i \geq 0$ para maximizar P_D, sujeito à condição da Equação (1.4), que pode ser expressa como

$$I - \sum_{i=1}^{N} p_i|\tilde{Q}_i\rangle\langle\tilde{Q}_i| \geq 0. \tag{1.8}$$

Em seguida, iremos abordar com detalhes o problema de otimização citado acima.

CAPÍTULO 2

Programação semidefinida e a discriminação de estados quânticos

Neste capítulo, veremos o problema geral de programação semidefinida, sua formulação para a discriminação de N estados quânticos não-ortogonais e os pacotes computacionais associados.

2.1 Programação semidefinida

Em geral, um problema de programação semidefinida (PSD) é um problema de minimização de uma função linear sujeito às restrições formadas por matrizes simétricas e positivas semidefinidas.

Um problema primal de programação semidefinida, na forma padrão, é definido por:

$$min \;\; f(x) = \langle c|x\rangle \tag{2.1}$$

$$\text{s.a} \;\; \left\{ \; F(x) = F_0 + \sum_{i=1}^{N} x_i F_i \geq 0, \right. \tag{2.2}$$

onde $|c\rangle, |x\rangle \in \mathbb{R}^N$, em que x_i denota a i-ésima componente de $|x\rangle$ e $F_0, \ldots, F_N \in \mathbb{R}^{m\times m}$ são matrizes simétricas. A expressão $F(x) \geq 0$ significa que $F(x)$ é positiva semidefinida, ou

seja,

$$\langle y|F(x)|y\rangle \geq 0, \quad \text{para todo} \quad y \in \mathbb{R}^N.$$

Definição 2.1. *Seja $|x\rangle$ uma solução primal do problema de PSD. Dizemos que:*

- *$|x\rangle$ é primal factível se $F(x) \geq 0$.*
- *$|x\rangle$ é estritamente primal factível se $F(x) > 0$. Neste caso, dizemos também que $|x\rangle$ é um ponto interior.*

Eldar [5] desenvolveu condições necessárias e suficientes para detectar os valores de p_i que maximizam P_D (1.7), sujeito à condição (1.8). Assim, formulou o problema dual associado ao problema primal (2.1) e (2.2):

$$max \;\; d(Z) = -tr(F_0 Z) \tag{2.3}$$

$$s.a \begin{cases} tr(F_i Z) = c_i, \quad 1 \leq i \leq N \\ \quad Z \geq 0, \end{cases} \tag{2.4}$$

onde $Z \in \mathbb{R}^{m\times m}$ é uma matriz simétrica positiva semidefinida.

Definição 2.2. *Seja Z uma solução dual do problema de PSD. Dizemos que:*

- *Z é dual factível se $Z \geq 0$.*
- *Z é estritamente dual factível se $Z > 0$. Neste caso, dizemos também que Z é um ponto interior.*

Uma característica dos problemas de programação semidefinida é que a variável dual Z assume a mesma forma que a matriz de restrição primal $F(x)$. Por exemplo, se a matriz $F(x)$ é diagonal em blocos, a variável dual Z também será diagonal em blocos [2, 21].

A função traço é linear, pois supondo $A = (a_{ij})$ e $B = (b_{ij})$, para $i, j = 1, \ldots, N$ e pela definição de traço, que é a soma dos elementos da diagonal principal da matriz, temos que

$$tr(\lambda A + B) = \sum_{i=1}^{N} (\lambda a_{ii} + b_{ii}) = \lambda \sum_{i=1}^{N} a_{ii} + \sum_{i=1}^{N} b_{ii} = \lambda tr(A) + tr(B).$$

Agora, considerando quaisquer pontos factíveis primal $|x\rangle$ e dual Z, temos que:

$$\begin{aligned} f(x) - d(Z) &= c^t x - (-tr(F_0 Z)) = c^t x + tr(F_0 Z) \\ &= \sum_{i=1}^{N} tr(F_i Z) x_i + tr(F_0 Z) \\ &= tr(F_0 Z + \sum_{i=1}^{N} x_i F_i Z) \\ &= tr\left((F_0 + \sum_{i=1}^{N} x_i F_i) Z \right) \\ &= tr(F(x)Z) \geq 0, \end{aligned} \tag{2.5}$$

onde usamos o fato de que $tr(AB) \geq 0$ se $A, B \geq 0$.

Assim, para todo $|x\rangle$ e Z factíveis, obtemos:

$$f(x) - d(Z) \geq 0 \Leftrightarrow d(Z) \leq f(x) \Leftrightarrow -tr(F_0 Z) \leq \langle c|x\rangle. \tag{2.6}$$

Logo, o valor ótimo dual, para qualquer ponto factível dual Z, coloca um limite inferior para o valor ótimo primal, para qualquer ponto factível primal $|x\rangle$.

Sejam p^* o valor ótimo do problema primal de PSD em (2.1) e (2.2), ou seja,

$$p^* = inf\{\langle c|x\rangle \quad : \quad F(x) \geq 0\},$$

e Z um ponto factível dual. Se a Equação (2.6) mantém a desigualdade para qualquer $|x\rangle$ factível, podemos concluir que

$$-tr(F_0 Z) \leq p^*.$$

Similarmente, os pontos factíveis primais produzem limites superiores para o problema dual,

$$d^* \leq \langle c|x\rangle,$$

onde d^* é o valor ótimo do problema dual de PSD (2.3) e (2.4), ou seja,

$$d^* = sup\{-tr(F_0 Z) \quad : \quad Z = Z^t \geq 0 \ \text{ e } \ tr(F_i Z) = c_i, \ \text{ para } \ i = 1, \ldots, N\}.$$

Disto tudo, obtemos que,

$$d^* \leq p^*.$$

O teorema abaixo [2] fornece condições para a igualdade da desigualdade acima.

Teorema 2.3. *Os valores primais e duais ótimos são iguais, isto é, $p^* = d^*$, se as seguintes afirmações forem satisfeitas:*

- *O problema primal (2.1) e (2.2) é estritamente factível, ou seja, existe $|x\rangle$ tal que $F(x) > 0$;*
- *O problema dual (2.3) e (2.4) é estritamente factível, ou seja, existe Z tal que $Z = Z^t > 0$ e $tr(F_i Z) = c_i$, para $i = 1, 2, \ldots, N$.*

Assumimos agora que existem $|x\rangle$ e Z estritamente factíveis. Assim, pelo Teorema 2.3, obtemos que

$$p^* = d^*,$$

e da Equação (2.5), temos que $tr(F(x)Z) = 0$. Se as matrizes simétricas $F(x)$ e Z são matrizes positivas semidefinidas [2], ou seja, $F(x) \geq 0$ e $Z \geq 0$, então

$$ZF(x) = 0. \tag{2.7}$$

Esta igualdade é chamada condição de complementaridade.

Dessa forma, a Equação (2.7), juntamente com as Equações (2.2) e (2.4), constituem um conjunto de condições necessárias e suficientes para $|x\rangle$ ser a solução ótima dos problemas (2.1) e (2.3), em que ambos os pontos primais e duais são estritamente factíveis, ou seja, dado $|x\rangle$ e Z factíveis estritamente, as condições de otimalidade do problema de PSD são:

- **Factibilidade primal:** $F(x) \geq 0$;
- **Factibilidade dual:** $Z \geq 0$ e $tr(F_i Z) = c_i$, para $1 \leq i \leq N$;
- **Complementaridade:** $ZF(x) = 0$.

2.2 Formulação PSD para a discriminação de estados quânticos

Agora, mostraremos a formulação de Eldar [5] para o problema UD, dado em (1.7) e (1.8), usando a PSD. Denotamos por $|p\rangle$ o vetor de componentes p_i e por $|\mu\rangle$ o vetor de componentes $-\mu_i$, para $1 \leq i \leq N$, com $\sum_{i=1}^{N} \mu_i = 1$. Então, o problema é

$$min \;\; f(p) = \langle \mu | p \rangle$$
$$\text{s.a} \;\; \left\{ \; I - \sum_{i=1}^{N} p_i |\tilde{Q}_i\rangle\langle\tilde{Q}_i| \geq 0,$$

onde $p_i \geq 0$. Para formular este problema como um problema de PSD, seja $\mathcal{C}_i = |\tilde{Q}_i\rangle\langle\tilde{Q}_i| \in \mathbb{R}^{N\times N}$ e F_i matrizes diagonais em blocos, para $i = 1, \ldots, N$, definidas como:

$$F_0 = \begin{pmatrix} I & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{pmatrix}, \quad F_1 = \begin{pmatrix} -\mathcal{C}_1 & & & \\ & 1 & & \\ & & 0 & \\ & & & \ddots \\ & & & & 0 \end{pmatrix}, \tag{2.8}$$

$$F_2 = \begin{pmatrix} -\mathcal{C}_2 & & & \\ & 0 & & \\ & & 1 & \\ & & & \ddots \\ & & & & 0 \end{pmatrix}, \quad \ldots, \quad F_N = \begin{pmatrix} -\mathcal{C}_N & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \\ & & & & 1 \end{pmatrix}.$$

Dessa forma,

$$\begin{aligned} F(p) &= F_0 + \sum_{i=1}^{N} p_i F_i \\ &= \begin{pmatrix} I - \sum_{i=1}^{N} p_i \mathcal{C}_i & & & \\ & p_1 & & \\ & & \ddots & \\ & & & p_N \end{pmatrix} \geq 0. \end{aligned}$$

Note que a restrição $F(p) \geq 0$ é equivalente a $I - \sum_{i=1}^{N} p_i |\tilde{Q}_i\rangle\langle\tilde{Q}_i| \geq 0$ e como $\mathcal{C}_i = |\tilde{Q}_i\rangle\langle\tilde{Q}_i|$, para $i = 1, \ldots, N$, são coordenadas da matriz F_i, que são simétricas e positivas definidas, segue que $\mathcal{C}_i$ também são simétricas e positivas definidas, para $i = 1, \ldots, N$. Assim, o

problema formulado por (1.7) e (1.8) pode ser escrito como o seguinte problema primal de PSD:

$$min_{p\in\mathbb{R}^N} \ \langle\mu|p\rangle \tag{2.9}$$

$$\text{s.a} \quad \begin{cases} I - \sum_{i=1}^{N} p_i\mathcal{C}_i \geq 0 \\ p_i \geq 0 \qquad\qquad (1 \leq i \leq N), \end{cases} \tag{2.10}$$

onde $|\mu\rangle$ é um vetor com componentes $-\mu_i$, sendo μ_i a probabilidade a priori de cada estado $|Q_i\rangle$ e $\mathcal{C}_i = |\tilde{Q}_i\rangle\langle\tilde{Q}_i| \in \mathbb{R}^{N\times N}$.

Como F_i são matrizes diagonais em bloco, podemos definir a variável dual Z na mesma estrutura, ou seja,

$$Z = \begin{pmatrix} X & & & \\ & z_1 & & \\ & & \ddots & \\ & & & z_N \end{pmatrix}, \tag{2.11}$$

onde $X \in \mathbb{R}^{N\times N}$ tem a mesma estrutura que $I - \sum_{i=1}^{N} p_i\mathcal{C}_i$ (neste caso, simétrica e positiva semidefinida) e $z_i \geq 0$, para $1 \leq i \leq N$.

Usando o fato de que a função traço é linear, obtemos:

$$\begin{aligned} tr(F_0Z) &= tr\left(\begin{pmatrix} I & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{pmatrix}\begin{pmatrix} X & & & \\ & z_1 & & \\ & & \ddots & \\ & & & z_N \end{pmatrix}\right) \\ &= tr\left(\begin{pmatrix} X & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{pmatrix}\right) \\ &= tr(X) \end{aligned}$$

e

$$tr(F_1Z) = -\mu_1 \Leftrightarrow tr\left(\left(\begin{pmatrix} -\mathcal{C}_1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 0 \end{pmatrix}\begin{pmatrix} X & & & \\ & z_1 & & \\ & & \ddots & \\ & & & z_m \end{pmatrix}\right)\right) = -\mu_1 \Leftrightarrow$$

$$tr\left(\left(\begin{pmatrix} -\mathcal{C}_1X & & & \\ & z_1 & & \\ & & \ddots & \\ & & & 0 \end{pmatrix}\right)\right) = -\mu_1 \Leftrightarrow tr(\mathcal{C}_1X) - z_1 = \mu_1.$$

Na forma geral:

$$tr(F_iZ) = -\mu_i \Leftrightarrow tr(\mathcal{C}_iX) - z_i = \mu_i, \quad \text{para} \quad i = 1, 2, \ldots, N.$$

Assim, podemos escrever o problema dual, associado ao problema (2.9) e (2.10), como:

$$max_{X\in\mathbb{R}^{N\times N}} \ (-tr(X)) \tag{2.12}$$

$$s.a \begin{cases} tr(\mathcal{C}_iX) - z_i = \mu_i \quad (1 \leq i \leq N) \\ X \geq 0 \\ z_i \geq 0 \qquad (1 \leq i \leq N). \end{cases}$$

Observe que,

$$ZF(p) = 0 \Leftrightarrow \begin{pmatrix} X & & & \\ & z_1 & & \\ & & \ddots & \\ & & & z_N \end{pmatrix}\begin{pmatrix} I - \sum_{i=1}^{N} p_i\mathcal{C}_i & & & \\ & p_1 & & \\ & & \ddots & \\ & & & p_N \end{pmatrix} = 0 \Leftrightarrow$$

$$\begin{pmatrix} X(I - \sum_{i=1}^{N} p_i\mathcal{C}_i) & & & \\ & z_1p_1 & & \\ & & \ddots & \\ & & & z_Np_N \end{pmatrix} = 0 \Leftrightarrow \begin{cases} X(I - \sum_{i=1}^{N} p_i\mathcal{C}_i) = 0 \\ z_ip_i = 0. \end{cases}$$

Logo,

$$ZF(p) = 0 \Leftrightarrow \begin{cases} X(I - \sum_{i=1}^{N} p_i \mathcal{C}_i) = 0 \\ z_i p_i = 0 \qquad (1 \leq i \leq N). \end{cases} \tag{2.13}$$

Portanto, $|p\rangle$ é uma solução ótima se, e somente se, as componentes p_i de $|p\rangle$ satisfazem as seguintes condições de otimalidade:

- **factibilidade primal:**

$$I - \sum_{i=1}^{N} p_i \mathcal{C}_i \geq 0 \ \text{ e } \ p_i \geq 0, \tag{2.14}$$

  para $i = 1, \ldots, N$. Note que,

$$\Pi_0 = I - \sum_{i=1}^{N} p_i \mathcal{C}_i \geq 0.$$

- **factibilidade dual:**

$$tr(\mathcal{C}_i X) - z_i = \mu_i, \ \ X \geq 0 \ \text{ e } \ z_i \geq 0, \tag{2.15}$$

  para $i = 1, \ldots, N$.

- **complementaridade:**

$$\begin{cases} X(I - \sum_{i=1}^{N} p_i \mathcal{C}_i) = 0 \\ z_i p_i = 0, \qquad \text{para} \ \ 1 \leq i \leq N. \end{cases} \tag{2.16}$$

  Note que,

$$X(I - \sum_{i=1}^{N} p_i \mathcal{C}_i) = 0 \Leftrightarrow X\Pi_0 = 0.$$

2.3 Programas computacionais

Em geral, não existe uma forma analítica para resolver o problema de maximização (1.7), sujeito a (1.8). Porém, como este problema é um problema de otimização convexa [15], existem vários métodos iterativos para resolvê-lo. Em particular, usamos o Matlab para achar a solução ótima $|p\rangle$ do problema:

$$min_{p\in\mathbb{R}^N} \ \langle\mu|p\rangle$$

$$s.a \begin{cases} I - \sum_{i=1}^{N} p_i \mathcal{C}_i \geq 0 \\ p_i \geq 0 \qquad \text{para} \quad 1 \leq i \leq N, \end{cases}$$

através dos pacotes LMI e o CSDP [9, 12, 16].

Estes pacotes garantem a convergência para o ótimo global em tempo polinomial.

O LMI (linear matrix inequality) [6] é um pacote que faz interface com os pacotes do Matlab: IQCβ [12] e Sedumi [16]. O algoritmo associado é uma versão do algoritmo projetor de Nesterov e Nemirovski [13]. Eldar [5] usou este pacote para um espaço de Hilbert tridimensional, trabalhando com as matrizes de restrição F_i em forma de diagonal em bloco, como dadas nas Equações (2.8). Ao invés disso, introduzimos este procedimento para um espaço de Hilbert de dimensão N e adotamos, na implementação do nosso algoritmo, as condições de restrição fornecidas pela Equação (2.10). Assim, as sequências de comando abaixo fornecem o código para achar a solução ótima de $|p\rangle$ via o LMI toolbox, denotado pela função *psd_lmi*:

```
%------- função : psd_lmi ------
%--- inicializando o LMI toolbox
  setlmis([])
%--- definindo o vetor |p> de dimensão N
  p = [1:N];
  for i = 1:N
     p(i)=lmivar(2,[1 1]);
%--- definindo as condições de restrição
     aux = C(:,:,i);
     lmiterm([1 1 1 p(i)],.5*1,aux,'s');
  end
     lmiterm([1 1 1 0],-eye(N));
```

```
  for i = 2:N+1
      lmiterm([-i 1 1 i-1],1,1);
  end
%--- solução interna do LMI
  lmisys=getlmis;
%--- transformando as probabilidades a priori em 'string'
  mu1=mat2dec3(lmisys,-mu);
%--- obtendo o valor ótimo de |p>
  [pstar,pot]=mincx(lmisys,mu1,[1e-12 0 0 0 0])
```

onde $pot = |pot\rangle$ e $p^* = \langle\mu_1|pot\rangle$ são, respectivamente, o valor ótimo e o valor global mínimo de $|p\rangle$.

O CSDP (Certified Software Development Professional) é um pacote apropriado para resolver problemas de programação semidefinida. O algoritmo associado é uma versão do algoritmo preditor corretor do método primal-dual de Helmberg et al. [9]. Este pacote foi escrito na linguagem em C, porém pode ser usado pelo Matlab através do toolbox yalmip. Rabelo [17] usou este procedimento para um espaço de Hilbert de dimensão N, e como Eldar, programou as matrizes de restrição F_i em forma de diagonal em bloco, como dadas nas Equações (2.8). Ao invés disso, trabalhamos com as condições de restrição fornecidas pela Equação (2.10), para um espaço de Hilbert também de dimensão N. Além disso, optamos por usar o comando "sdpsettings" na implementação, pois esta função imprime informações do andamento do algoritmo tais como: número de iterações, função objetivo primal, função objetivo dual, etc. Assim, nosso código é dado pelas seguintes sequências de comando, denotado pela função *psd_csdp*:

```
%------- função : psd_csdp ------
%--- definindo o vetor |p> de dimensão N
p=sdpvar(N,1);
%--- definindo as condições de restrição
I=eye(N);
R=I;
for i=1:N;
```

```
    R=R-p(i)*C(:,:,i);
end
F=set(R>=0);
for i=1:N
    F=F+set(p(i)>=0);
end
%--- minimizando <mu|p> sujeito as restrições
condition=solvesdp(F,-mu'*p,sdpsettings('solver','csdp'))
%--- obtendo o valor ótimo de |p>
pot=double(p)
```

Portanto, o processo para encontrar os valores p_i associados aos estados quânticos não-ortogonais e linearmente independentes $|Q_i\rangle$ com uma distribuição de probabilidade a priori μ_i, para $i = 1, \ldots, N$, resume-se em 3 passos:

- **Primeiro passo:** definir a matriz Ψ, tal que suas colunas são os estados de entrada, isto é, $\Psi = \left[\begin{array}{cccc} |Q_1\rangle & |Q_2\rangle & \ldots & |Q_N\rangle \end{array} \right]$. Em seguida, achar os estados recíprocos $|\tilde{Q}_i\rangle$ associados aos estados $|Q_i\rangle$, através da relação:

$$\tilde{\Psi} = \Psi(\Psi^t\Psi)^{-1} = \left[\begin{array}{cccc} |\tilde{Q}_1\rangle & |\tilde{Q}_2\rangle & \ldots & |\tilde{Q}_N\rangle \end{array} \right].$$

- **Segundo passo:** achar as matrizes $\mathcal{C}_i = |\tilde{Q}_i\rangle\langle\tilde{Q}_i|$, para $i = 1, \ldots, N$.

- **Terceiro passo:** aplicar um dos pacotes dados anteriormente, ou seja, LMI ou CSDP, para encontrar os valores p_i, onde o vetor das probabilidades a priori μ_i dos estados $|Q_i\rangle$ são definidos com entradas $-\mu_i$, para $i = 1, 2, \ldots, N$.

Para os dois primeiros passos, criamos também um programa no MatLab, com o objetivo de tornar os cálculos mais eficientes, da seguinte forma:

```
%--- function: matrizC
  function [C]= matrizC(Psi,N)
%--- pseudo inversa de Psi
  Psitil = Psi/(Psi'*Psi);
```

```
%--- vetores Qtil obtidos através das matrizes Psitil
  for i=1:N
      Qtili=Psitil(:,i);
  end
%---  matrizes Ci
  for i=1:N
      C(:,:,i) = Qtili*Qtili';
  end
```

É relevante observar que as matrizes $\mathcal{C}_i$, para $i = 1, \ldots, N$ definidas acima, foram implementadas de forma agrupada, ou seja, geramos uma matriz $\mathcal{C}$ de saída que contém todas estas matrizes, isto é, $\mathcal{C} = \begin{pmatrix} \mathcal{C}_1 & \mathcal{C}_2 & \ldots & \mathcal{C}_N \end{pmatrix}$, onde cada matriz $\mathcal{C}_i$ é simétrica positiva definida com dimensão N, para $i = 1, \ldots, N$.

2.4 Exemplo

Considere um ensemble constituído por três estados quânticos puros não-ortogonais com probabilidades $0,6$ e $0,2$ e $0,2$, respectivamente, ou seja,

$$\rho = 0,6|Q_1\rangle\langle Q_1| + 0,2|Q_2\rangle\langle Q_2| + 0,2|Q_3\rangle\langle Q_3|,$$

onde

$$|Q_1\rangle = \begin{pmatrix} \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{pmatrix}, \quad |Q_2\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}, \quad |Q_3\rangle = \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}.$$

Para encontrar os operadores de detecção ótima, temos que encontrar inicialmente os estados recíprocos $|\tilde{Q}_i\rangle$ associados aos estados $|Q_i\rangle$. Assim, seja Ψ a matriz cujas colunas são os estados $|Q_i\rangle$, para $i = 1, 2, 3$. Da Equação (1.6), temos

$$\tilde{\Psi} = \Psi(\Psi^t\Psi)^{-1} = \begin{pmatrix} 1,7321 & 0 & -1,4142 \\ -1,7321 & 1,4142 & 1,4142 \\ 1,7321 & -1,4142 & 0 \end{pmatrix},$$

onde os estados recíprocos $|\tilde{Q}_i\rangle$ são as colunas desta matriz. Em seguida, formamos as matrizes $\mathcal{C}_i = |\tilde{Q}_i\rangle\langle\tilde{Q}_i|$, para $i = 1, 2, 3$. Logo,

$$\mathcal{C}_1 = \begin{pmatrix} 3 & -3 & 3 \\ -3 & 3 & -3 \\ 3 & -3 & 3 \end{pmatrix}, \quad \mathcal{C}_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & -2 \\ 0 & -2 & 2 \end{pmatrix},$$

$$\mathcal{C}_3 = \begin{pmatrix} 2 & -2 & 0 \\ -2 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Agora, para encontrar o vetor ótimo $|p\rangle$, usamos os pacotes LMI ou CSDP, como apresentados anteriormente, e obtemos os seguinte resultados:

	$\|p\rangle$	iterações	tempo(s)	$p^* = \langle\mu\|p\rangle$
LMI	$\begin{pmatrix} 0,0572 \\ 0,0858 \\ 0,0858 \end{pmatrix}$	15	0,0310	-0,0686
CSDP	$\begin{pmatrix} 0,0572 \\ 0,0858 \\ 0,0858 \end{pmatrix}$	12	0,2500	-0,0686

Tabela 2.1: Resultados obtidos pelos pacotes LMI e CSDP.

Portanto, os operadores de detecção ótima $\Pi_i = p_i\mathcal{C}_i$, para $i = 1, 2, 3$, são

$$\Pi_1 = \begin{pmatrix} 0,1716 & -0,1716 & 0,1716 \\ -0,1716 & 0,1716 & -0,1716 \\ 0,1716 & -0,1716 & 0,1716 \end{pmatrix}, \quad \Pi_2 \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0,1716 & -0,1716 \\ 0 & -0,1716 & 0,1716 \end{pmatrix},$$

$$\Pi_3 = \begin{pmatrix} 0,1716 & -0,1716 & 0 \\ -0,1716 & 0,1716 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Agora, vamos provar que o vetor $|p\rangle$ dado na Tabela 2.1 é a solução ótima. Para esta finalidade, basta verificar as condições de otimalidade, ou seja, as Equações (2.14), (2.15) e (2.16).

- Pela Equação (2.14), temos que verificar: $\Pi_0 = I - \sum_{i=1}^{N} p_i \mathcal{C}_i \geq 0$ e $p_i \geq 0$.

  Como $|p\rangle = \begin{pmatrix} 0,0572 \\ 0,0858 \\ 0,0858 \end{pmatrix}$, segue que $p_i \geq 0$, para $i = 1, 2, 3$. O operador Π_0, chamado de operador inconclusivo, é dado por:

$$\Pi_0 = I - \sum_{i=1}^{3} p_i \mathcal{C}_i = \begin{pmatrix} 0,6569 & 0,3431 & -0,1716 \\ 0,3431 & 0,4853 & 0,3431 \\ -0,1716 & 0,3431 & 0,6569 \end{pmatrix}.$$

  Assim, para verificar que $\Pi_0 \geq 0$, ou seja, que Π_0 é uma matriz positiva semidefinida, basta provar que os seus autovalores são todos positivos. Desta forma, temos que os autovalores desta matriz são:

$$eig(\Pi_0) = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0,8284 \\ 0,9706 \end{pmatrix}.$$

  Logo, como todos os autovalores de Π_0 são não negativas, segue que Π_0 é uma matriz positiva semidefinida.

- Usando a decomposição de autovalores de Π_0, os autovetores associados aos autovalores de Π_0 são respectivamente,

$$\begin{pmatrix} |\tau_1\rangle & |\tau_2\rangle & |\tau_3\rangle \end{pmatrix} = \begin{pmatrix} 0,5 & -0,7071 & 0,5 \\ -0,7071 & 0 & -0,7071 \\ 0,5 & 0,7071 & 0,5 \end{pmatrix}.$$

  Logo, concluímos que o espaço nulo de Π_0 tem dimensão 1 e é gerado pelo seguinte vetor:

$$|\tau_1\rangle = \begin{pmatrix} 0,5 \\ -0,7071 \\ 0,5 \end{pmatrix}.$$

Portanto, para satisfazer as condições (2.15) e (2.16), devemos definir X como:

$$X = t|\tau_1\rangle\langle\tau_1| = \begin{pmatrix} 0,25t & -0,3536t & 0,25t \\ -0,3536t & 0,5t & -0,3536t \\ 0,25t & -0,3536t & 0,25t \end{pmatrix},$$

para algum $t \geq 0$. Como $p_1, p_2, p_3 > 0$, a relação $z_i p_i = 0$, para $i = 1, 2, 3$, implica que $z_1 = z_2 = z_3 = 0$. Desta forma, da igualdade $tr(\mathcal{C}_i X) = \mu_i + z_i$, para $i = 1, 2, 3$, temos que:

$$\begin{cases} tr(\mathcal{C}_1 X) = 8,7426t \\ tr(\mathcal{C}_1 X) = \mu_1 + z_1 = 0,6 \end{cases} \Rightarrow t = 0,0686$$

$$\begin{cases} tr(\mathcal{C}_2 X) = 2,9142t \\ tr(\mathcal{C}_2 X) = \mu_2 + z_2 = 0,2 \end{cases} \Rightarrow t = 0,0686$$

$$\begin{cases} tr(\mathcal{C}_3 X) = 2,9142t \\ tr(\mathcal{C}_3 X) = \mu_3 + z_3 = 0,2 \end{cases} \Rightarrow t = 0,0686.$$

Escolhendo $t = 0,0686$, temos que $X = \begin{pmatrix} 0,0171 & -0,0243 & 0,0171 \\ -0,0243 & 0,0343 & -0,0243 \\ 0,0171 & -0,0243 & 0,0171 \end{pmatrix}$. Como os autovalores de X são todos positivos, ou seja, $eig(X) = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0,0686 \end{pmatrix}$, segue que X é uma matriz positiva semidefinida. Logo, $X \geq 0$.

- Pela Equação (2.16), temos que

$$\begin{cases} X(I - \sum_{i=1}^{N} p_i \mathcal{C}_i) = X\Pi_0 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ z_i p_i = 0, \end{cases}$$

para $i = 1, 2, 3$.

Observe que $t = 0,0686 = -p^*$, onde p^* é o valor ótimo do problema PSD (2.9) e (2.10) dado por $p^* = \langle p|\mu\rangle$, com $|p\rangle = \begin{pmatrix} 0,0572 \\ 0,0858 \\ 0,0858 \end{pmatrix}$ e $|\mu\rangle = \begin{pmatrix} -0,6 \\ -0,2 \\ -0,2 \end{pmatrix}$.

CAPÍTULO 3

Algoritmo discriminador quântico

Neste capítulo, apresentamos um novo algoritmo para realizar a discriminação de N estados quânticos puros não-ortogonais. Este algoritmo fornece o melhor conjunto de medidas POVM para a discriminação dos estados quânticos. Através da extensão do espaço de Hilbert, obtém-se uma configuração final discriminável.

3.1 A motivação

Em [17, 18], Rabelo et al propõem um procedimento computacional chamado de Algoritmo Ótimo Discriminador (AOD), que implementa as medidas POVM para a estratégia da distinção de N estados puros não-ortogonais, usando a programação semidefinida e a minimização da norma. Baseado nas idéias do AOD, desenvolvemos um novo algoritmo, denominado Algoritmo Discriminador Quântico (ADQ).

A principal diferença entre o AOD e o ADQ é que usamos apenas conceitos de álgebra linear, substituindo o cálculo de raízes de um polinômio de grau 8. Além disso, desenvolvemos o algoritmo somente com uma condição para os estados de entrada a saber, a da conservação do produto escalar, pois a condição de normalização imposta no AOD tornou-se reduntante, uma vez que os estados de entrada satisfazem a condição de normalização (1.2). Não pudemos fazer comparações de tempo e número de iterações, pois não existem tais dados para o AOD.

Acreditamos também termos conseguido reduzir bastante os conceitos físicos envolvidos no AOD, deixando o ADQ mais compreensível para a comunidade de matemática e computação.

3.2 Passo a passo do ADQ

Assim como o AOD, o objetivo do ADQ é realizar a distinção de N estados quânticos puros não-ortogonais. Devido aos índices durante o processo da implementação do algoritmo, adotaremos a base ortonormal $\{|1\rangle, |2\rangle, \ldots, |N\rangle\}$, ao invés de $\{|0\rangle, |1\rangle, \ldots, |N-1\rangle\}$.

Agora, vamos explicar passo a passo cada etapa do ADQ.

3.2.1 Passo 1: forma escada

A primeira tarefa do ADQ é reescrever os N estados de entrada não-ortogonais $|Q_i\rangle$, em uma forma escada, numa base ortonormal $\{|i\rangle\}$, para $i = 1, \ldots, N$. Ou seja,

$$\begin{aligned}
|Q_1^{esc}\rangle &= |1\rangle, \\
|Q_2^{esc}\rangle &= c_{21}|1\rangle + c_{22}|2\rangle, \\
|Q_3^{esc}\rangle &= c_{31}|1\rangle + c_{32}|2\rangle + c_{33}|3\rangle, \\
|Q_4^{esc}\rangle &= c_{41}|1\rangle + c_{42}|2\rangle + c_{43}|3\rangle + c_{44}|4\rangle, \\
&\vdots \\
|Q_N^{esc}\rangle &= c_{N1}|1\rangle + c_{N2}|2\rangle + c_{N3}|3\rangle + \ldots + c_{NN}|N\rangle,
\end{aligned} \tag{3.1}$$

onde $\{|Q_i^{esc}\rangle\}$ pertencem ao espaço de Hilbert inicial de dimensão N e os coeficientes c_{ij} são obtidos usando a **conservação do produto escalar** sobre os estados de entrada, isto é, $\langle Q_i^{esc}|Q_j^{esc}\rangle = \langle Q_i|Q_j\rangle$, para todo $i, j = 1, \ldots, N$. Note que o fato dos estados de entrada serem normalizados garante a conservação da norma, isto é, $\langle Q_i^{esc}|Q_i^{esc}\rangle = \langle Q_i|Q_i\rangle = 1$, para $i = 1, \ldots, N$.

Definindo C como a matriz dos coeficientes c_{ij},

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ c_{21} & c_{22} & 0 & 0 & \dots & 0 \\ c_{31} & c_{32} & c_{33} & 0 & \dots & 0 \\ c_{41} & c_{42} & c_{43} & c_{44} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{N1} & c_{N2} & c_{N3} & c_{N4} & \dots & c_{NN} \end{pmatrix}, \tag{3.2}$$

obtemos a matriz Q^{esc} formada pelos estados de entrada na forma escada, dada por:

$$Q^{esc} = \begin{pmatrix} |Q_1^{esc}\rangle & |Q_2^{esc}\rangle & |Q_3^{esc}\rangle & \dots & |Q_N^{esc}\rangle \end{pmatrix} = \begin{pmatrix} 1 & c_{21} & c_{31} & \dots & c_{N1} \\ 0 & c_{22} & c_{32} & \dots & c_{N2} \\ 0 & 0 & c_{33} & \dots & c_{N3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & c_{NN} \end{pmatrix}.$$

Note que $Q^{esc} = C^t$.

Discutiremos agora como encontrar os coeficientes c_{ij}, mostrando com detalhes o cálculo para o caso $N = 4$, e apresentaremos em seguida a generalização para N estados quânticos.

Sejam $|Q_i\rangle$ estados puros não-ortogonais normalizados, definidos no espaço de Hilbert de dimensão 4, para $i = 1, 2, 3, 4$. A configuração desses estados na forma escada é dada por:

$$|Q_1^{esc}\rangle = |1\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |Q_2^{esc}\rangle = c_{21}|1\rangle + c_{22}|2\rangle = \begin{pmatrix} c_{21} \\ c_{22} \\ 0 \\ 0 \end{pmatrix},$$

$$|Q_3^{esc}\rangle = c_{31}|1\rangle + c_{32}|2\rangle + c_{33}|3\rangle = \begin{pmatrix} c_{31} \\ c_{32} \\ c_{33} \\ 0 \end{pmatrix}, \quad |Q_4^{esc}\rangle = c_{41}|1\rangle + c_{42}|2\rangle + c_{43}|3\rangle + c_{44}|4\rangle = \begin{pmatrix} c_{41} \\ c_{42} \\ c_{43} \\ c_{44} \end{pmatrix},$$

onde a matriz dos coeficientes é expressa como: $C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ c_{21} & c_{22} & 0 & 0 \\ c_{31} & c_{32} & c_{33} & 0 \\ c_{41} & c_{42} & c_{43} & c_{44} \end{pmatrix}$.

Como os estados preservam o produto escalar, podemos encontrar os elementos de C como descrito abaixo.

- Elementos da primeira coluna de C:

$$\begin{cases} \langle Q_1^{esc}|Q_1^{esc}\rangle = \langle Q_1|Q_1\rangle \Leftrightarrow 1 = 1 \\ \langle Q_1^{esc}|Q_2^{esc}\rangle = \langle Q_1|Q_2\rangle \Leftrightarrow c_{21} = \langle Q_1|Q_2\rangle \\ \langle Q_1^{esc}|Q_3^{esc}\rangle = \langle Q_1|Q_3\rangle \Leftrightarrow c_{31} = \langle Q_1|Q_3\rangle \\ \langle Q_1^{esc}|Q_4^{esc}\rangle = \langle Q_1|Q_4\rangle \Leftrightarrow c_{41} = \langle Q_1|Q_4\rangle. \end{cases}$$

- Elementos da segunda coluna de C:

$$\begin{cases} \langle Q_2^{esc}|Q_2^{esc}\rangle = \langle Q_2|Q_2\rangle \Leftrightarrow c_{21}^2 + c_{22}^2 = 1 \\ \langle Q_2^{esc}|Q_3^{esc}\rangle = \langle Q_2|Q_3\rangle \Leftrightarrow c_{21}c_{31} + c_{22}c_{32} = \langle Q_2|Q_3\rangle \\ \langle Q_2^{esc}|Q_4^{esc}\rangle = \langle Q_2|Q_4\rangle \Leftrightarrow c_{21}c_{41} + c_{22}c_{42} = \langle Q_2|Q_4\rangle, \end{cases}$$

  implicando que

$$c_{22} = \sqrt{1 - c_{21}^2} \quad \text{e} \quad c_{i2} = \frac{\langle Q_i|Q_2\rangle - c_{21}c_{i1}}{c_{22}}, \quad \text{para} \quad i = 3, 4.$$

- Elementos da terceira coluna de C:

$$\begin{cases} \langle Q_3^{esc}|Q_3^{esc}\rangle = \langle Q_3|Q_3\rangle \Leftrightarrow c_{31}^2 + c_{32}^2 + c_{33}^2 = 1 \\ \langle Q_3^{esc}|Q_4^{esc}\rangle = \langle Q_3|Q_4\rangle \Leftrightarrow c_{31}c_{41} + c_{32}c_{42} + c_{33}c_{43} = \langle Q_3|Q_4\rangle, \end{cases}$$

  resultando em

$$c_{33} = \sqrt{1 - c_{31}^2 - c_{32}^2} \quad \text{e} \quad c_{43} = \frac{\langle Q_4|Q_3\rangle - c_{31}c_{41} - c_{32}c_{42}}{c_{33}}.$$

- Quarta coluna de C:

$$\langle Q_4^{esc}|Q_4^{esc}\rangle = \langle Q_4|Q_4\rangle \Leftrightarrow c_{41}^2 + c_{42}^2 + c_{43}^2 + c_{44}^2 = 1,$$

  implicando que

$$c_{44} = \sqrt{1 - c_{41}^2 - c_{42}^2 - c_{43}^2}.$$

Estendendo para N estados quânticos, temos os seguintes resultados:

- Elementos da primeira coluna de C:

$$c_{i1} = \langle Q_i | Q_1 \rangle, \quad \text{para} \quad i = 2, \ldots, N. \tag{3.3}$$

- Elementos da diagonal de C:

$$c_{jj} = \sqrt{1 - \sum_{k=1}^{j-1} c_{jk}^2}, \quad \text{para} \quad j = 2, \ldots, N. \tag{3.4}$$

- Elementos abaixo da diagonal principal de C, com exceção dos elementos que se referem à primeira coluna:

$$c_{ij} = \frac{\langle Q_i | Q_j \rangle - \sum_{k=1}^{j-1} c_{ik} c_{jk}}{c_{jj}}, \quad \text{para} \quad i = 3, \ldots, N \quad \text{e} \quad j = 2, \ldots, N. \tag{3.5}$$

A implementação desta primeira rotina do ADQ, através do software MatLab, é dada abaixo:

```
%---  function: forma_escada
    function [Qesc]= forma_escada(Q,N)
%--- definindo o elemento c11 da matriz C
     C(1,1) = 1.0;
%--- coeficientes da 1° Coluna de C: Ci1, para i=2,...,N
    for i=2:N
        C(i,1) = dot(Q(:,1),Q(:,i));
    end
    for j = 2:N
    %--- coeficientes da diagonal de C
        soma = 0.0;
        for i = 1:(j-1)
            soma = soma + C(j,i)*C(j,i);
        end
```

```
        C(j,j) = sqrt( 1.0 - soma );
    %--- coeficientes abaixo da diagonal principal de C
        for i = (j+1):N
            soma=0.0;
            for k= 1 : (j-1)
                soma = soma + C(i,k)*C(j,k);
            end
        C(i,j)=(dot(Q(:,i),Q(:,j)) - soma)/C(j,j);
       end
    end
%---  obtendo a matriz Qesc
    Qesc=C';
```

Observação 3.2.1. *Como* $Q^{esc} = C^t$*, temos que:*

$$CC^t = Q^{esc^t}Q^{esc} = \langle Q_i^{esc}|Q_j^{esc}\rangle = \langle Q_i|Q_j\rangle = Q^tQ.$$

Como Q *é a matriz cujas colunas são os estados de entrada não-ortogonais e normalizados,* Q^tQ *é uma matriz simétrica e definida positiva. Logo,* Q^{esc} *é o fator de Cholesky da matriz* Q^tQ*, ou seja,* $Q^{esc} = chol(Q^tQ)$*.*

Observe o seguinte programa teste:

```
%---  dimensão do sistema
        N=3;
%--- vetores de entrada
        Q1=[1/sqrt(3);1/sqrt(3);1/sqrt(3)];
        Q2=[1/sqrt(2);1/sqrt(2);0];
        Q3=[0;1/sqrt(2);1/sqrt(2)];
        Q=[Q1 Q2 Q3];
%--- cálculo da matriz dos coeficentes pela função forma escada
        [Qesc]= forma_escada(Q,N);
%---  veja isso!
        S = Q'*Q;
```

```
        B = chol(S);
%--- comparando B com Qesc
        disp('     B='),    disp(B)
        disp('     Qesc='), disp(Qesc)
```

Executando o programa, obtemos:

```
     B=
    1.0000    0.8165    0.8165
         0    0.5774   -0.2887
         0         0    0.5000

     Qesc=
    1.0000    0.8165    0.8165
         0    0.5774   -0.2887
         0         0    0.5000
```

3.2.2 Passo 2: matrizes $\mathcal{C}_i$, para $i = 1, \ldots, N$

Para definir as matrizes $\mathcal{C}_i$, para $i = 1, \ldots, N$, precisamos primeiramente definir a matriz Ψ, cujas colunas são os estados de entrada, isto é, $\Psi = \left[\begin{array}{cccc} |Q_1\rangle & |Q_2\rangle & \ldots & |Q_N\rangle \end{array} \right]$. Em seguida, achar os estados recíprocos $|\tilde{Q}_i\rangle$ associados aos estados $|Q_i\rangle$, através da relação:

$$\tilde{\Psi} = \Psi(\Psi^t\Psi)^{-1} = \left[\begin{array}{cccc} |\tilde{Q}_1\rangle & |\tilde{Q}_2\rangle & \ldots & |\tilde{Q}_N\rangle \end{array} \right].$$

Logo as matrizes $\mathcal{C}_i$, para $i = 1, \ldots, N$, são definidas pela seguinte igualdade:

$$\mathcal{C}_i = |\tilde{Q}_i\rangle\langle\tilde{Q}_i|,$$

onde $|\tilde{Q}_i\rangle$ são as colunas de $\tilde{\Psi}$, para $i = 1, \ldots, N$.

Essa rotina foi implementada da seguinte forma:

```
%--- function: matrizC
  function [C]= matrizC(Psi,N)
```

```
%--- pseudo inversa de Psi
  Psitil = Psi/(Psi'*Psi);
%--- vetores Qtil obtidos através das matrizes Psitil
  for i=1:N
      Qtili=Psitil(:,i);
  end
%---  matrizes Ci
  for i=1:N
      C(:,:,i) = Qtili*Qtili';
  end
```

Note que, a matriz $\mathcal{C}$ de saída é dada em agrupamento, ou seja, $\mathcal{C} = \left[\begin{array}{cccc} \mathcal{C}_1 & \mathcal{C}_2 & \ldots & \mathcal{C}_N \end{array} \right]$, onde cada matriz $\mathcal{C}_i$ é simétrica positiva definida com dimensão N, para $i = 1, \ldots, N$.

3.2.3 Passo 3: programação semidefinida

Neste passo, o problema abaixo é resolvido usando a programação semidefinida, discutida no Capítulo 2.

$$min_{p \in \mathbb{R}^N} \ \langle \mu | p \rangle$$

$$s.a \begin{cases} I - \sum_{i=1}^{N} p_i \mathcal{C}_i \geq 0 \\ p_i \geq 0, \qquad \text{para} \quad 1 \leq i \leq N. \end{cases}$$

3.2.4 Passo 4: configuração final discriminável

Este passo finalizará o procedimento do ADQ, ou seja, estaremos aptos para realizar a distinção dos N estados quânticos não-ortogonais (lembramos que os estados de entrada não-ortogonais $|Q_i\rangle$, para $i = 1, \ldots, N$, obtiveram uma nova estrutura "na forma escada" de acordo com a Equação (3.2), dada no Passo 1). A partir de agora, nosso objetivo é obter uma nova configuração, a configuração final discriminável, fazendo com que as componentes de mesmos índices sejam fixas com a base ortonormal, ou seja, $\{|1\rangle, \ldots, |N\rangle\}$. Para isto,

utilizamos o problema dado em [3, 19] para a construção desta configuração, que se baseia na extensão do espaço de Hilbert inicial de dimensão N para $2N-1$.

Desta forma, a configuração final discriminável dos N estados quânticos não-ortogonais é dada por:

$$\begin{aligned}
|Q_{f1}\rangle &= g_{11}|1\rangle + g_{1,N+1}|N+1\rangle + \ldots + g_{1,2N-2}|2N-2\rangle + g_{1,2N-1}|2N-1\rangle \\
|Q_{f2}\rangle &= g_{22}|2\rangle + g_{2,N+1}|N+1\rangle + \ldots + g_{2,2N-2}|2N-2\rangle + g_{2,2N-1}|2N-1\rangle \\
|Q_{3f}\rangle &= g_{33}|3\rangle + g_{3,N+1}|N+1\rangle + \ldots + g_{3,2N-2}|2N-2\rangle \\
&\vdots \\
|Q_{fi}\rangle &= g_{ii}|i\rangle + g_{i,N+1}|N+1\rangle + \ldots + g_{i,2N+1-i}|2N+1-i\rangle \\
&\vdots \\
|Q_{fN}\rangle &= g_{NN}|N\rangle + g_{N,N+1}|N+1\rangle,
\end{aligned} \tag{3.6}$$

para $i = 3, \ldots, N$.

Nesta nova configuração, os estados possuem somente uma única componente na base ortonormal (espaço de Hilbert original), onde os seus índices são fixos com os da base associada, e as demais componentes da base estendida (espaço de Hilbert estendido), ou seja, $\{|N+1\rangle, \ldots, |2N-1\rangle\}$.

Agora, o objetivo torna-se encontrar os valores dos coeficientes associados às bases, isto é,

$$\begin{cases} g_{ii}, & \text{para} \quad i = 1, 2, \ldots, N & \text{(coeficientes originais)} \\ g_{ij}, & \text{para} \quad i = 1, \ldots, N, \quad j = N+1, \ldots, 2N-1 & \text{(coeficientes adicionais).} \end{cases}$$

Logo, definimos:

- Os g_{ii}, para $i = 1, \ldots, N$, como a raiz quadrada dos valores p_i, obtidos da solução do problema semidefinido sobre os estados de entrada, para $i = 1, \ldots, N$, ou seja,

$$g_{ii} = \sqrt{p_i}. \tag{3.7}$$

- Os outros coeficientes g_{ij}, para $i = 1, \ldots, N$ e $j = N+1, \ldots, 2N-1$, são definidos de modo a preservar o produto interno de colunas entre os estados na forma escada com os estados na configuração final discriminável.

Garantimos a otimalidade da configuração final discriminável pelo fato de usarmos os valores p_i, para $i = 1, \ldots, N$, na sua construção, onde p_i são as soluções do problema de programação semidefinida como estudado no Capítulo 2.

A seguir, descreveremos o método proposto para determinar as componentes da Equação (3.6). Inicialmente, definimos a matriz dos coeficientes G da configuração final discriminável da seguinte forma:

$$G = \left(\begin{array}{ccccc:cccccccc} g_{11} & 0 & 0 & \ldots & 0 & g_{1,N+1} & g_{1,N+2} & \cdots & g_{1,2N+1-i} & \cdots & g_{1,2N-2} & g_{1,2N-1} \\ 0 & g_{22} & 0 & \ldots & 0 & g_{2,N+1} & g_{2,N+2} & \cdots & g_{2,N+1-i} & \cdots & g_{2,2N-2} & g_{2,2N-1} \\ 0 & 0 & g_{33} & \ldots & 0 & g_{3,N+1} & g_{3,N+2} & \cdots & g_{3,2N+1-i} & \cdots & g_{3,2N-2} & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 & \vdots & \vdots & \ldots & \vdots & \ldots & 0 & 0 \\ 0 & 0 & 0 & \ldots & g_{NN} & g_{N,N+1} & 0 & \ldots & 0 & \ldots & 0 & 0 \end{array} \right). \tag{3.8}$$

Note que,

$$\begin{aligned} Q_f &= G^t \\ &= \left(\begin{array}{cccc} |Q_{f1}\rangle & |Q_{f2}\rangle & \ldots & |Q_{fN}\rangle \end{array} \right) \\ &= \left(\begin{array}{cccccc} g_{11} & 0 & \ldots & 0 & \ldots & 0 \\ 0 & g_{22} & \ldots & 0 & \ldots & 0 \\ 0 & 0 & \ddots & 0 & \ldots & 0 \\ 0 & 0 & \ldots & g_{ii} & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots & \ddots & 0 \\ 0 & 0 & \ldots & 0 & \ldots & g_{NN} \\ g_{1,N+1} & g_{2,N+1} & \cdots & g_{i,N+1} & \cdots & g_{N,N+1} \\ g_{1,N+2} & g_{2,N+2} & \cdots & g_{3,N+2} & \ldots & 0 \\ g_{1,N+3} & g_{2,N+3} & \cdots & g_{i,2N+1-i} & \ldots & 0 \\ \vdots & \vdots & \ldots & 0 & \ldots & 0 \\ \vdots & \vdots & \ldots & 0 & \ldots & 0 \\ g_{1,2N-1} & g_{2,2N-1} & \cdots & 0 & \ldots & 0 \end{array} \right), \end{aligned}$$

uma matriz de dimensão $2N - 1 \times N$.

Já sabemos que os elementos de G, referentes às componentes relacionadas à base ortonormal, são obtidos resolvendo um problema de programação semidefinida sobre os estados de

entrada, ou seja, $g_{ii} = \sqrt{p_i}$, para $i = 1, \ldots, N$ (onde os valores de p_i são solucionados no Passo 3). Assim, na matriz $Q_f = G^t$, os únicos elementos desconhecidos são os coeficientes associados ao espaço de Hilbert estendido, ou seja, g_{ij}, para $i = 1, \ldots, N$ e $j = N+1, \ldots, 2N-1$. Para resolver este problema, propomos o uso de conceitos de álgebra linear, mais precisamente, o uso de matrizes em forma de blocos e a solução de sistemas lineares (esta é uma das principais diferenças do nosso trabalho, comparado com os resultados conhecidos na literatura).

Assim, reescrevendo a matriz Q_f, dada anteriormente, usando matrizes em forma de blocos, obtemos:

$$Q_f = \begin{pmatrix} g_{11} & \vdots & 0^t \\ \cdots & \cdots & \cdots \\ 0 & \vdots & \tilde{D}^t \\ \cdots & \cdots & \cdots \\ b & \vdots & B^t \end{pmatrix}, \tag{3.9}$$

onde 0 é o vetor nulo de dimensão $N-1 \times 1$, $\tilde{D} = \begin{pmatrix} g_{22} & 0 & \ldots & 0 \\ 0 & g_{33} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & g_{NN} \end{pmatrix}$ é uma matriz $N-1 \times N-1$, $B^t = \begin{pmatrix} g_{2,N+1} & g_{3,N+1} & \cdots & g_{N-1,N+1} & g_{N,N+1} \\ g_{2,N+2} & g_{3,N+2} & \cdots & g_{N-1,N+2} & 0 \\ \vdots & \vdots & \ldots & 0 & 0 \\ g_{2,2N-1} & 0 & \ldots & 0 & 0 \end{pmatrix}$ é uma matriz de ordem $N-1 \times N-1$ (semelhante à triangular superior, porém com anti-diagonal) e $b = \begin{pmatrix} g_{1,N+1} \\ g_{1,N+1} \\ \vdots \\ g_{1,2N-1} \end{pmatrix}$ é um vetor de ordem $N-1 \times 1$.

Lembramos que $Q^{esc} = C^t = \begin{pmatrix} 1 & c_{21} & c_{31} & \dots & c_{N1} \\ 0 & c_{22} & c_{32} & \dots & c_{N2} \\ 0 & 0 & c_{33} & \dots & c_{N3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}$ é uma matriz de ordem $N \times N$.
Estendendo esta matriz para $2N-1 \times N$ e também escrevendo-a na forma de blocos, obtemos:

$$Q^{\tilde{esc}} = \begin{pmatrix} Q^{esc} \\ \dots \\ \tilde{O} \end{pmatrix} = \begin{pmatrix} 1 & \vdots & s^t \\ \dots & \dots & \dots \\ 0 & \vdots & S \\ \dots & \dots & \dots \\ 0 & \vdots & O \end{pmatrix}, \tag{3.10}$$

onde $\tilde{O}$ é a matriz nula de dimensão $N-1 \times N$, $s = \begin{pmatrix} c_{21} \\ c_{31} \\ \vdots \\ c_{N1} \end{pmatrix}$ é um vetor de ordem $N-1 \times 1$,

0 é o vetor nulo de ordem $N-1 \times 1$, $S = \begin{pmatrix} c_{22} & c_{32} & \dots & c_{N2} \\ 0 & c_{33} & \dots & c_{N3} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & c_{NN} \end{pmatrix}$ é uma matriz de dimensão $N-1 \times N-1$ e O é matriz nula de ordem $N-1 \times N-1$. Dessa forma, $Q^{\tilde{esc}}$ será uma matriz de ordem $2N-1 \times N$.

Pela conservação do produto interno dos vetores coluna das matrizes Q_f e $Q^{\tilde{esc}}$, ou seja,

$$Q_f^t Q_f = {Q^{\tilde{esc}}}^t Q^{\tilde{esc}},$$

obtemos que:

$$\begin{pmatrix} g_{11} & \vdots & 0^t & \vdots & b^t \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \vdots & \tilde{D} & \vdots & B \end{pmatrix} \begin{pmatrix} g_{11} & \vdots & 0^t \\ \dots & \dots & \dots \\ 0 & \vdots & \tilde{D}^t \\ \dots & \dots & \dots \\ b & \vdots & B^t \end{pmatrix} = \begin{pmatrix} 1 & \vdots & 0^t & \vdots & 0^t \\ \dots & \dots & \dots & \dots & \dots \\ s & \vdots & S^t & \vdots & O^t \end{pmatrix} \begin{pmatrix} 1 & \vdots & s^t \\ \dots & \dots & \dots \\ 0 & \vdots & S \\ \dots & \dots & \dots \\ 0 & \vdots & O \end{pmatrix}.$$

Logo,

$$\begin{pmatrix} g_{11}^2 + b^t b & \vdots & b^t B^t \\ \cdots & \cdots & \cdots \\ Bb & \vdots & \tilde{D}\tilde{D}^t + BB^t \end{pmatrix} = \begin{pmatrix} 1 & \vdots & s^t \\ \cdots & \cdots & \cdots \\ s & \vdots & ss^t + S^t S \end{pmatrix}.$$

Fazendo a igualdade matricial, temos que:

$$\begin{cases} g_{11}^2 + b^t b = 1 \\ b^t B^t = s^t \\ Bb = s \\ \tilde{D}\tilde{D}^t + BB^t = S^t S + ss^t. \end{cases} \tag{3.11}$$

Agora, nosso problema resume-se em resolver o seguinte sistema linear:

$$\begin{cases} BB^t = S^t S + ss^t - \tilde{D}\tilde{D}^t \\ Bb = s \\ g_{11}^2 + b^t b = 1, \end{cases} \tag{3.12}$$

ou seja, encontrar a matriz B e o vetor b, pois pelo Passo 1, conhecemos a matriz S e o vetor s. A matriz $\tilde{D}$ e a componente g_{11} são definidas pela programação semidefinida. Assim, com estes resultados na Equação (3.9), obtemos a matriz Q_f, isto é, a matriz de configuração final discriminável dos estados de entrada não-ortogonais. O programa que implementa essa rotina foi formulado da seguinte forma:

```
%--- function: configuracao_final
function [Qf]=configuracao_final(pot,Qesc,N)
%--- definindo matriz diagonal D
   for i =1:N
       D(i,i)=sqrt(pot(i));
   end
%--- definindo matriz S (triang.superior) e vetor s
   S=Qesc(2:N,2:N);
   Dtil = D(2:N,2:N);
   k=2:N;
   s=Qesc(1,k)';
```

```
%--- cálculo de A = S'S +ss' -DD', ordem N-1 x N-1
   A = S'*S + s*s' - Dtil*Dtil';
%---  cálculo da matriz B
    ndim = length(A);
     for j = 1:ndim
         soma = 0.0;
         i = ndim - j + 1;
         %--- elementos da diagonal de B
          for k = 1:(j-1)
              soma = soma + B(i,k)*B(i,k);
          end
          B(i,j) = sqrt(A(i,i) - soma);
          %--- elementos acima da diagonal de B
           for l = (i-1):-1:1
               soma = 0.0;
               for k = 1:(j-1)
                   soma = soma + B(i,k)*B(l,k);
               end
               B(l,j) = ( A(i,l) - soma ) / B(i,j);
           end
     end
%--- cálculo do vetor b
        for j=1:ndim
            soma=0.0;
            i= ndim - j +1;
           for k = 1: (j-1)
               soma = soma + B(i,k)*b(k,1);
           end
           b(j,1)=(s(i,1) - soma)/B(i,j);
        end
%---  cálculo da matriz Qf
```

```
Qf=[D(1,1) zeros(1,N-1);zeros(N-1,1) Dtil; b B];
```

3.3 A distinção dos estados quânticos não-ortogonais

Agora, estamos aptos a responder a pergunta da Seção 1.2, ou seja, como fazer a distinção entre os estados não-ortogonais $|Q_1\rangle$ e $|Q_2\rangle$? De acordo com o ADQ, através da extensão do espaço de Hilbert inicial de N para $2N-1$ dimensões, transformamos os estados quânticos não-ortogonais de entrada (neste caso, $|Q_1\rangle$ e $|Q_2\rangle$) em uma configuração final discriminável, dada pela Equação (3.6). Isso significa que estamos prontos para fazer a distinção do estado final, pois teremos uma discriminação perfeita ou falha no processo para os estados quânticos não-ortogonais analisados, uma vez que, os estados nesta configuração foram escritos com componentes únicas na base ortonormal e as demais componentes na base extra. Desta forma, se a medida colapsa na base ortonormal $\{|1\rangle, \ldots, |N\rangle\}$ do espaço de Hilbert original, distinguimos o estado perfeitamente, devido a unicidade da representação dos estados nesta componente. Caso contrário, se a medida colapsa para a base extra $\{|N+1\rangle, \ldots, |2N-1\rangle\}$ do espaço de Hilbert estendido, não sabemos qual foi o estado detectado, pois os estados de interesse (neste caso, $|Q_{f1}\rangle$ e $|Q_{f2}\rangle$) possuem uma das componentes da base estendida, $|N+1\rangle, |N+2\rangle, \ldots, |2N-1\rangle$, ou elas simultaneamente. A figura abaixo ilustra este procedimento:

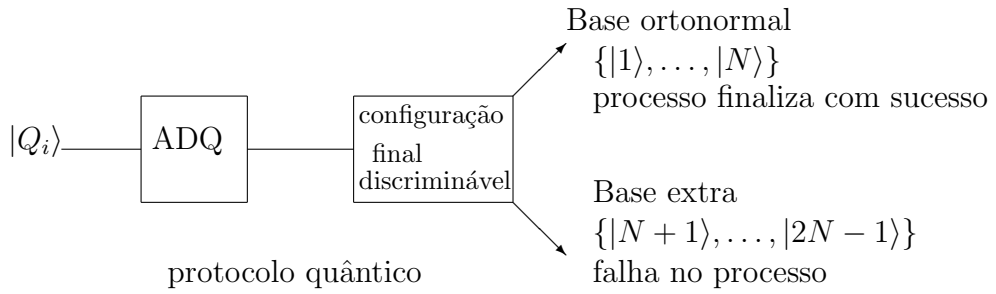


Figura 3.1 : Processo de medição para N estados quânticos não-ortogonais via ADQ.

Note que, com uma única medida projetiva (ou seja, medida na base), temos a possibilidade de discriminar os estados sem ambiguidade.

CAPÍTULO 4

Resultados computacionais

Neste capítulo, veremos aplicações do ADQ através dos dois pacotes apresentados no Capítulo 2, ou seja, ADQ via LMI ou via CSDP. Nestas versões, empregamos o algoritmo fornecendo os estados não-ortogonais associados com as probabilidades a priori.

4.1 Exemplo detalhado

Nesta seção, veremos um exemplo detalhado, de dimensão $N = 2$, para a aplicação do ADQ.

Considere o ensemble $\{(0,5; |Q_1\rangle), (0,5; |Q_2\rangle)\}$, onde

$$|Q_1\rangle = 0,9717|1\rangle + 0,2364|2\rangle,$$
$$|Q2\rangle = 0,7805|1\rangle + 0,6251|2\rangle.$$

Aplicando o ADQ, temos os seguintes passos:

- **Passo 1:** A matriz que gera os estados de entrada na forma escada é dada por:

$$Q^{esc} = \begin{pmatrix} 1 & 0,9062 \\ 0 & 0,4229 \end{pmatrix}.$$

Então,

$$|Q_1^{esc}\rangle = |1\rangle$$
$$|Q_2^{esc}\rangle = 0,9062|1\rangle + 0,4229|2\rangle.$$

- **Passo 2:** As matrizes $\mathcal{C}_1$ e $\mathcal{C}_2$, obtidas através da pseudo-inversa de Moore-Penrose da matriz Ψ, cujas colunas são os estados não-ortogonais de entrada, são dadas por

$$\mathcal{C}_1 = \begin{pmatrix} 2,1849 & -2,7280 \\ -2,7280 & 3,4062 \end{pmatrix} \text{ e } \mathcal{C}_2 = \begin{pmatrix} 0,3125 & -1,2844 \\ -1,2844 & 5,2795 \end{pmatrix}.$$

- **Passo 3:** Para encontrar o vetor ótimo $|p\rangle$, do problema (2.9) e (2.10), usamos os pacotes LMI ou CSDP através do Matlab, como apresentados anteriormente. Para ambos os pacotes, obtivemos:

$$|p\rangle = \begin{pmatrix} 0,0939 \\ 0,0937 \end{pmatrix}.$$

- **Passo 4:** A matriz de configuração final é:

$$Qf = \begin{pmatrix} 0,3064 & 0 \\ 0 & 0,3062 \\ 0,9519 & 0,9520 \end{pmatrix}.$$

Então,

$$\begin{cases} |Q_{f1}\rangle = 0,3064|1\rangle + 0,9519|3\rangle, \\ |Q_{f2}\rangle = 0,3062|2\rangle + 0,9520|4\rangle \end{cases}$$

A tabela abaixo fornece os valores das iterações e tempo de cada pacote usado.

pacote	iterações	tempo(s)
LMI	15	0,0150
CSDP	14	1,1720

Tabela 4.1. Resultados computacionais do ADQ para N=2.

4.2 Exemplos para N dimensões

Em todos os resultados descritos nesta seção, os testes foram realizados em um computador com sistema operacional Windows XP, processador Intel Core 2 Duo, T5500 @ 1.66 GHz, com 1 GB de Ram e usando a versão do MatLab 7.0.1.

Para cada dimensão fornecida, aplicamos o algoritmo a dez problemas com ensembles aleatórios, ou seja, foram fornecidos aleatoriamente estados de entrada não-ortogonais associados a probabilidades a priori também aleatórias.

Nas tabelas a seguir, fornecemos o número de iterações e o tempo, em segundos, referentes à aplicação do ADQ, para a obtenção da configuração final discriminável Q_f, considerando os pacotes LMI e CSDP para a solução do problema de PSD associado.

- **Dimensão** $N = 2$

N=2	dados	1	2	3	4	5	6	7	8	9	10
LMI	iterações	16	18	21	19	16	12	17	15	16	17
	tempo	0,0150s	0,0320s	0,0160s	0,0310s	0,0150s	0,0320s	0,0310s	0,0310s	0,0160s	0,0160s
CSDP	iterações	15	15	13	11	15	14	12	13	16	16
	tempo	1,8440s	1,4210s	1,2500s	1,5940s	1,3290s	1,2810s	1,4840s	1,1720s	1,7030s	1,2660s

Tabela 4.2. Resultados computacionais do ADQ para N=2.

	iterações			tempo		
N=2	mínimo	média	máximo	mínimo	média	máximo
LMI	12	17	21	0,0150s	0,0235s	0,0320s
CSDP	11	14	16	1,1720s	1,4344s	1,8440s

Tabela 4.3. Estatística para N=2.

- **Dimensão** $N = 3$

N=3	dados	1	2	3	4	5	6	7	8	9	10
LMI	iterações	23	29	26	27	25	26	24	18	23	22
	tempo	0,0320s	0,0160s	0,0150s	0,0310s	0,0150s	0,0160s	0,0160s	0,0150s	0,0160s	0,0320s
CSDP	iterações	13	12	11	13	11	12	15	12	13	11
	tempo	1,9530s	1,5310s	1,3900s	1,6410s	1,4060s	3,3290s	1,5320s	1,4370s	1,5310s	1,4060s

Tabela 4.4. Resultados computacionais do ADQ para N=3.

	iterações			tempo		
N=3	mínimo	média	máximo	mínimo	média	máximo
LMI	18	24	29	0,0150s	0,0204s	0,0320s
CSDP	11	12	15	1,3900s	1,7156s	3,3290s

Tabela 4.5. Estatística para N=3.

- **Dimensão** $N = 4$

N=4	dados	1	2	3	4	5	6	7	8	9	10
LMI	iterações	31	18	18	20	27	30	18	18	29	29
	tempo	0,0310s	0,0470s	0,0160s	0,0470s	0,0310s	0,0160s	0,0310s	0,0150s	0,0310s	0,0470s
CSDP	iterações	15	15	13	12	14	11	12	12	12	15
	tempo	2,0470s	1,6570s	1,7970s	2,1570s	1,6880s	1,8130s	1,6720s	1,5620s	1,5940s	1,7190s

Tabela 4.6. Resultados computacionais do ADQ para N=4.

	iterações			tempo		
N=4	mínimo	média	máximo	mínimo	média	máximo
LMI	18	24	31	0,0150s	0,0312s	0,0470s
CSDP	11	13	15	1,5620s	1,7706s	2,1570s

Tabela 4.7. Estatística para N=4.

- **Dimensão** $N = 5$

N=5	dados	1	2	3	4	5	6	7	8	9	10
LMI	iterações	32	28	33	21	24	21	24	32	21	33
	tempo	0,0630s	0,0470s	0,0310s	0,0310s	0,0150s	0,0310s	0,0310s	0,0310s	0,0310s	0,0310s
CSDP	iterações	15	15	18	14	16	13	15	18	13	15
	tempo	2,5000s	1,7660s	2,6250s	1,8290s	1,7650s	1,6100s	2,6250s	2,4530s	2,3600s	1,6410s

Tabela 4.8. Resultados computacionais do ADQ para N=5.

	iterações			tempo		
N=5	mínimo	média	máximo	mínimo	média	máximo
LMI	21	27	33	0,0150s	0,0342s	0,0630s
CSDP	13	15	18	1,6100s	2,1174s	2,6250s

Tabela 4.9. Estatística para N =5.

- **Dimensão** $N = 6$

N=6	dados	1	2	3	4	5	6	7	8	9	10
LMI	iterações	19	30	19	33	34	19	36	27	33	36
	tempo	0,0470s	0,0320s	0,0310s	0,0630s	0,0620s	0,0310s	0,0620s	0,0470s	0,0630s	0,0470s
CSDP	iterações	15	17	15	16	15	15	16	15	16	18
	tempo	2,2660s	1,6090s	2,7030s	2,0160s	1,5630s	1,5930s	1,6250s	1,7500s	2,1410s	3,0160s

Tabela 4.10. Resultados computacionais do ADQ para N=6.

	iterações			tempo		
N=6	mínimo	média	máximo	mínimo	média	máximo
LMI	19	29	36	0,0310s	0,0485s	0,0630s
CSDP	15	16	18	1,5630s	2,0282s	3,0160s

Tabela 4.11. Estatística para N=6.

- **Dimensão $N = 7$**

N=7	dados	1	2	3	4	5	6	7	8	9	10
LMI	iterações	37	21	42	23	35	37	39	45	36	36
	tempo	0,1090s	0,0780s	0,0790s	0,0630s	0,0620s	0,0780s	0,0780s	0,0780s	0,0780s	0,0780s
CSDP	iterações	16	14	21	21	16	17	27	20	16	14
	tempo	2,2660s	2,6250s	3,3900s	1,7340s	1,7500s	1,7810s	1,8120s	3,0310s	2,2030s	2,3440s

Tabela 4.12. Resultados computacionais do ADQ para N=7.

	iterações			tempo		
N=7	mínimo	média	máximo	mínimo	média	máximo
LMI	21	35	45	0,0620s	0,0781s	0,1090s
CSDP	14	18	27	1,7340s	2,2936s	3,3900s

Tabela 4.13. Estatística para N=7.

- **Dimensão $N = 8$**

N=8	dados	1	2	3	4	5	6	7	8	9	10
LMI	iterações	36	21	41	43	35	32	30	29	40	42
	tempo	0,0780s	0,0780s	0,0930s	0,0940s	0,0940s	0,0940s	0,0940s	0,0780s	0,1090s	0,0940s
CSDP	iterações	18	16	20	17	16	20	16	15	16	20
	tempo	2,3590s	3,7970s	6,0150s	2,1100s	1,8590s	1,6710s	5,3280s	2,2970s	1,9070s	2,0780s

Tabela 4.14. Resultados computacionais do ADQ para N=8.

	iterações			tempo		
N=8	mínimo	média	máximo	mínimo	média	máximo
LMI	21	35	43	0,0780s	0,0906s	0,1090s
CSDP	15	17	20	1,6710s	2,9421s	6,0150s

Tabela 4.15. Estatística para N=8.

- **Dimensão** $N = 9$

N=9	dados	1	2	3	4	5	6	7	8	9	10
LMI	iterações	48	36	42	49	29	43	23	52	40	29
	tempo	0,1720s	0,1400s	0,1410s	0,1710s	0,1090s	0,1560s	0,1090s	0,1560s	0,1560s	0,0930s
CSDP	iterações	22	20	20	16	18	22	20	20	18	20
	tempo	2,6410s	2,7810s	2,0930s	1,7350s	2,2030s	2s	2,8280s	2,2030s	1,6410s	4,7350s

Tabela 4.16. Resultados computacionais do ADQ para N=9.

	iterações			tempo		
N=9	mínimo	média	máximo	mínimo	média	máximo
LMI	23	39	52	0,0930s	0,1403s	0,1720s
CSDP	16	20	22	1,6410s	2,4860s	4,7350s

Tabela 4.17. Estatística para N=9.

- **Dimensão** $N = 10$

N=10	dados	1	2	3	4	5	6	7	8	9	10
LMI	iterações	49	55	27	36	49	34	51	44	36	39
	tempo	0,2500s	0,2810s	0,1410s	0,1880s	0,2660s	0,1720s	0,2500s	0,2350s	0,1720s	0,2030s
CSDP	iterações	15	18	16	16	15	19	21	17	17	23
	tempo	3,6250s	1,7650s	2,0160s	1,8440s	1,9530s	1,7500s	1,7810s	2,0310s	1,8900s	1,8590s

Tabela 4.18. Resultados computacionais do ADQ para N=10.

	iterações			tempo		
N=10	mínimo	média	máximo	mínimo	média	máximo
LMI	27	42	55	0,1410s	0,2158s	0,2810s
CSDP	15	18	23	1,7500s	2,0514s	3,6250s

Tabela 4.19. Estatística para N=10.

- **Dimensão** $N = 20$

N=20	dados	1	2	3	4	5	6	7	8	9	10
LMI	iterações	35	42	49	46	52	57	64	52	39	69
	tempo	3,0930s	3,7340s	4,3430s	4,0620s	4,1720s	5,0470s	5,6250s	4,6250s	3,4680s	5,5000s
CSDP	iterações	27	19	22	25	24	21	27	20	28	22
	tempo	3,0160s	2,4530s	2,9840s	2,5790s	2,5000s	2,2970s	2,4370s	2,7500s	2,7660s	3,1720s

Tabela 4.20. Resultados computacionais do ADQ para N=20.

	iterações			tempo		
N=20	mínimo	média	máximo	mínimo	média	máximo
LMI	35	51	69	3,0930s	4,3669s	5,6250s
CSDP	19	24	28	2,2970s	2,6954s	3,1720s

Tabela 4.21. Estatística para N=20.

- **Dimensão** $N = 30$

N=30	dados	1	2	3	4	5
LMI	iterações	55	56	88	64	66
	tempo	31,9530s	32,1100s	50,7340s	37,2030s	38,5000s
CSDP	iterações	23	28	25	22	26
	tempo	4,8440s	3,4680s	3,9690s	5,1560s	3,9840s

N=30	dados	6	7	8	9	10
LMI	iterações	62	54	86	59	69
	tempo	36,2030s	31,4060s	48,8440s	34,4070s	40,1100s
CSDP	iterações	62	54	86	59	69
	tempo	3,4370s	3,3750s	4,7650s	5,6870s	4,6570s

Tabela 4.22. Resultados computacionais do ADQ para N=30.

	iterações			tempo		
N=30	mínimo	média	máximo	mínimo	média	máximo
LMI	54	66	88	31,4060s	38,1470s	50,7340s
CSDP	22	28	47	3,3750s	4,3342s	5,6870s

Tabela 4.23. Estatística para N=30.

- **Dimensão** $N = 40$

N=40	dados	1	2	3	4	5
LMI	iterações	71	94	68	65	78
	tempo	2min45,1880s	3min38,9840s	2min38,0780s	2min31,1100s	3min1,5780s
CSDP	iterações	29	31	27	21	28
	tempo	5,7340s	4,9060s	6,4070s	5,7820s	4,5790s

N=40	dados	6	7	8	9	10
LMI	iterações	76	73	75	73	72
	tempo	2min56,6720s	2min49,7340s	2min54,5160s	2min49,7820s	2min47,4840s
CSDP	iterações	30	34	25	33	28
	tempo	6,0940s	5,5000s	5,3120s	5,4060s	8,8440s

Tabela 4.24. Resultados computacionais do ADQ para N=40.

	iterações			tempo		
N=40	mínimo	média	máximo	mínimo	média	máximo
LMI	65	75	94	2min31,1100s	2min53,3126s	3min38,9840s
CSDP	21	29	34	4,9530s	6,1859s	8,8440s

Tabela 4.25. Estatística para N=40.

- **Dimensão** $N = 50$

N=50	dados	1	2	3	4	5
LMI	iterações	70	68	89	70	60
	tempo	7min58,5470s	7min44,7190s	10min7,9220s	7min58,5150s	6min49,9060s
CSDP	iterações	29	29	28	29	30
	tempo	8,2810s	10,5160s	11,2500s	8,2190s	10s

N=50	dados	6	7	8	9	10
LMI	iterações	61	80	61	100	77
	tempo	6min56,9690s	9min6,6870s	6min56,8590s	11min23,5930s	8min45,9220s
CSDP	iterações	25	39	30	32	27
	tempo	7,0620s	7,9070s	7,3590s	7,2970s	7,1560s

Tabela 4.26. Resultados computacionais do ADQ para N=50.

	iterações			tempo		
N=50	mínimo	média	máximo	mínimo	média	máximo
LMI	60	74	100	6min49,9060s	8min22,9639s	11min23,5930s
CSDP	22	29	32	7,8130s	9,0860s	11,2500s

Tabela 4.27. Estatística para N=50.

- **Dimensão** $N = 60$

N=60	dados	1	2	3	4	5
LMI	iterações	88	88	98	100	65
	tempo	18min23,1s	25min4,8s	27min53,2s	28min26,9s	19min18,8s
CSDP	iterações	35	37	31	39	30
	tempo	15s	14,7500s	14,0940s	18,3130s	13,7350s

N=60	dados	6	7	8	9	10
LMI	iterações	90	100	72	100	88
	tempo	25min38,3s	28min22,7s	20min31s	28min29,2s	25min17,5s
CSDP	iterações	44	37	21	30	31
	tempo	15,2190s	13,7500s	12,5470s	13,9540s	17,0940s

Tabela 4.28. Resultados computacionais do ADQ para N=60.

	iterações			tempo		
N=60	mínimo	média	máximo	mínimo	média	máximo
LMI	65	89	100	18min23,2s	24min44,56s	28min29,2s
CSDP	21	34	44	12,5470s	14,8456s	18,3130s

Tabela 4.29. Estatística para N=60.

- **Dimensão** $N = 70$

N=70	dados	1	2	3	4	5
LMI	iterações	100	70	94	95	91
	tempo	1h13min39,1s	51min45,9s	1h9min13,9s	53min51,3s	1h7min18,9s
CSDP	iterações	30	30	36	30	44
	tempo	20,3440s	21,4220s	21,5000s	23,1250s	22,6870s

N=70	dados	6	7	8	9	10
LMI	iterações	62	65	100	59	100
	tempo	45min49,7s	48min1,6s	1h14min24,3s	44min6s	1h14min29s
CSDP	iterações	29	32	29	25	32
	tempo	18,9840s	22,4370s	19,6250s	18,5780s	19,7340s

Tabela 4.30. Resultados computacionais do ADQ para N=70.

	iterações			tempo		
N=70	mínimo	média	máximo	mínimo	média	máximo
LMI	59	84	100	44min6s	1h15,97s	1h14min29s
CSDP	25	32	44	18,5780s	20,8436s	23,1250s

Tabela 4.31. Estatística para N=70.

- **Dimensão** $N = 80$

N=80	dados	1	2	3	4	5
LMI	iterações	100	76	100	72	63
	tempo	2h11min0,6s	1h15min36,5s	2h10min16,8s	1h33min44,1s	1h21min54,7s
CSDP	iterações	36	30	34	39	34
	tempo	40,7970s	32,4370s	30,9380s	30,0160s	35,5160s

N=80	dados	6	7	8	9	10
LMI	iterações	56	63	71	67	100
	tempo	1h13min2,8s	1h22min9,5s	1h32min37,8s	1h27min31,6s	2h10min54,4s
CSDP	iterações	38	30	34	37	33
	tempo	31,9060s	28,0470s	34,7970s	31,8280s	32,1250s

Tabela 4.32. Resultados computacionais do ADQ para N=80.

	iterações			tempo		
N=80	mínimo	média	máximo	mínimo	média	máximo
LMI	56	77	100	1h13min2,8s	1h37min52,88s	2h11min0,6s
CSDP	30	35	39	28,0470s	32,8407s	40,7970s

Tabela 4.33. Estatística para N=80.

- **Dimensão** $N = 90$

N=90	dados	1	2	3	4	5
LMI	iterações	74	70	86	100	94
	tempo	3h10min39s	2h59min47s	3h41min8s	4h16min8s	3h10min12s
CSDP	iterações	31	39	36	34	29
	tempo	39,5470s	45,4060s	43,1090s	42,8600s	40s

N=90	dados	6	7	8	9	10
LMI	iterações	87	63	83	94	90
	tempo	2h55min58s	2h42min3,3s	3h32min47s	4h58s	3h50min11s
CSDP	iterações	34	31	39	42	36
	tempo	51,1560s	40,8280s	52,5310s	45,9840s	42,8280s

Tabela 4.34. Resultados computacionais do ADQ para N=90.

	iterações			tempo		
N=90	mínimo	média	máximo	mínimo	média	máximo
LMI	63	84	100	2h42min3,3s	3h26min5,13s	4h16min58s
CSDP	29	35	42	39,5470s	44,4249s	52,5310s

Tabela 4.35. Estatística para N=90.

- **Dimensão $N = 100$**

N=100	dados	1	2	3	4	5
LMI	iterações	98	100	80	86	100
	tempo	6h44min2s	6h50min29s	5h28min34s	5h53min5s	6h51min55s
CSDP	iterações	38	36	45	52	29
	tempo	1min1,2030s	1min19,5310s	1min7,1560s	1min30,5160s	55,4060s

N=100	dados	6	7	8	9	10
LMI	iterações	70	91	91	93	86
	tempo	3h40min55s	4h46min48s	8h11min5s	6h22min28s	5h53min19s
CSDP	iterações	53	45	29	38	35
	tempo	1min19,4380s	1min19,6560s	55,6410s	1min12,4060s	59,2970s

Tabela 4.36. Resultados computacionais do ADQ para N=100.

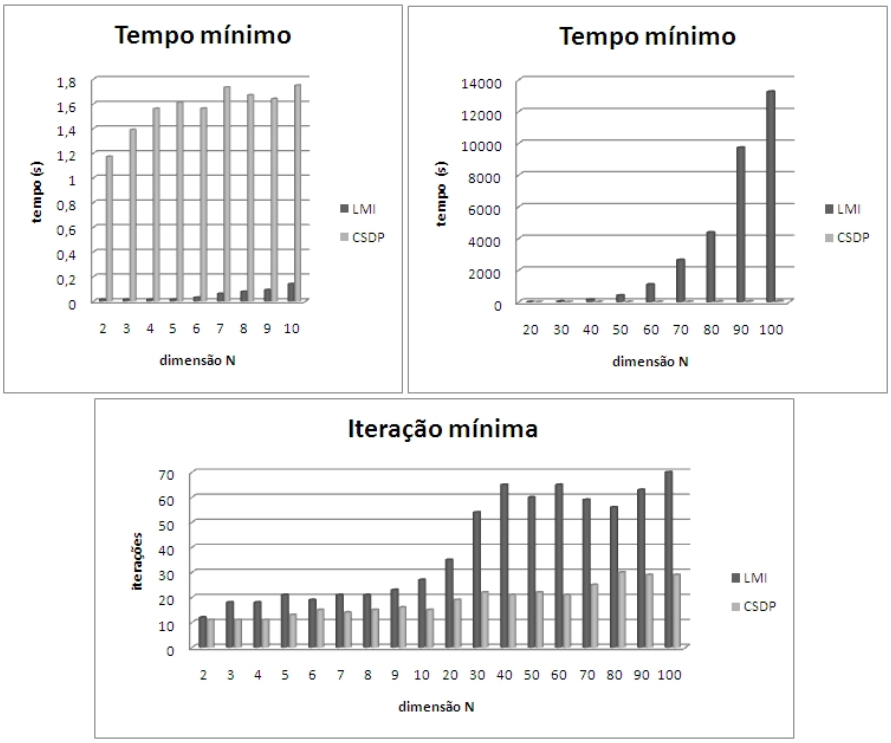
	iterações			tempo		
N=100	mínimo	média	máximo	mínimo	média	máximo
LMI	70	90	100	3h40min55s	6h4min16s	8h11min5s
CSDP	29	40	53	55,4060s	1min11,025s	1min30,5160s

Tabela 4.37. Estatística para N=100.

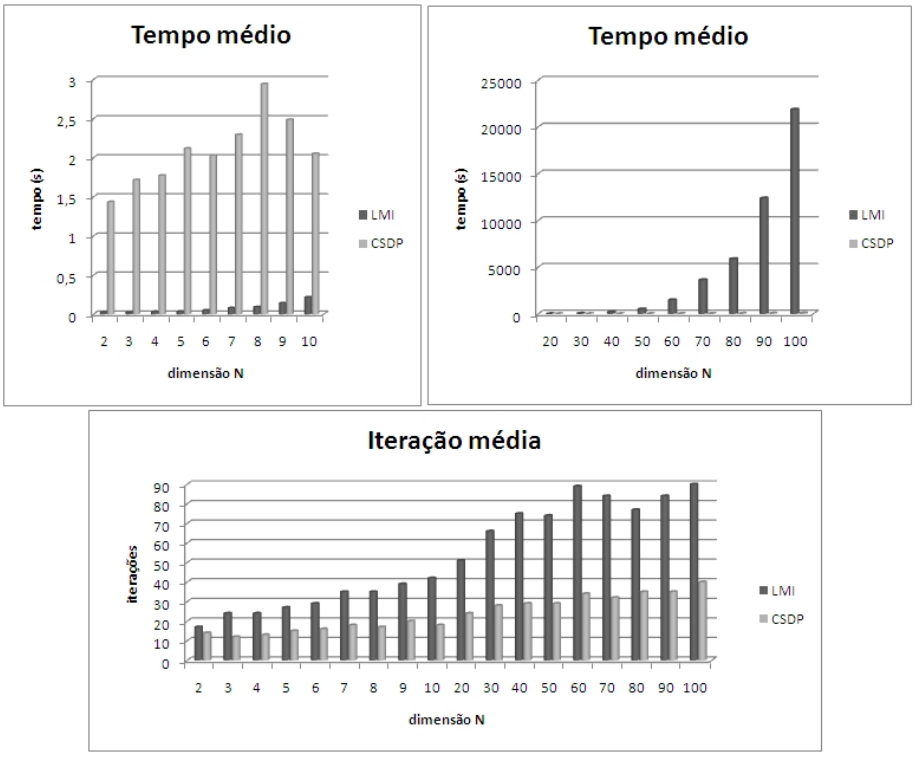
4.3 Gráficos de tempo e iterações

Nesta seção, utilizamos os dados fornecidos nas tabelas da seção anterior, para plotar gráficos de tempo e iterações, considerando os valores mínimo, médio e máximo obtidos para os dez problemas dados, associados às dimensões consideradas acima.

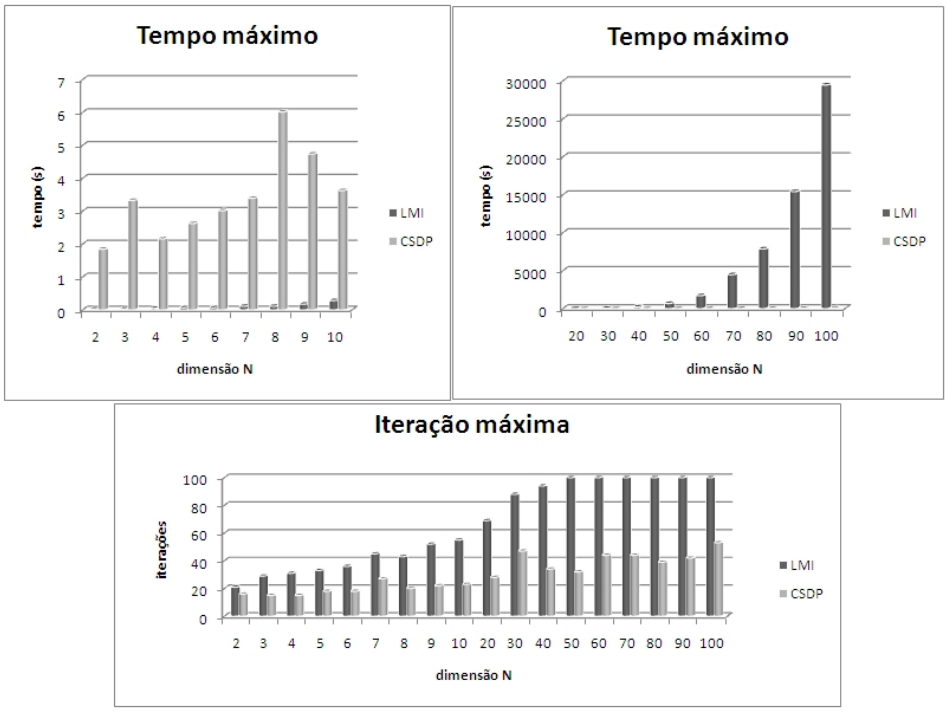
- Gráficos de tempo mínimo × iteração mínima



- Gráficos de tempo médio × iteração média



- Gráficos de tempo máximo × iteração máxima



Observando os gráficos referentes aos valores mínimos de tempo, podemos verificar que, para a dimensão $2 \leq N \leq 10$, o pacote LMI é mais rápido que o pacote CSDP. Por exemplo, para a dimensão $N = 8$, o pacote LMI opera em $0,078s$, enquanto o pacote CSDP atua no mesmo problema em $1,6710s$. No entanto, a partir de $N \geq 20$, o CSDP foi mais rápido que o outro pacote. Esta vantagem é mais evidente no gráfico a partir de $N = 60$, onde o pacote CSDP age aproximadamente em $13s$, ao passo que o pacote LMI necessita de $18min23s$. Para problemas com dimensão $N = 100$, é notável a grande diferença de tempo entre os dois pacotes, uma vez que o pacote LMI leva em torno de $3h40min55s$ e o pacote CSDP necessita de $55s$. No gráfico de iteração mínima, notamos que o CSDP sempre obteve número de iterações menores para todas as dimensões em relação ao outro pacote e não ultrapassou o número de 30 iterações. Já o pacote LMI atingiu 70 iterações para a dimensão máxima ($N = 100$).

Para os gráficos referentes aos valores médios de tempo, obtemos também um comportamento semelhante à análise anterior, pois para a dimensão $2 \leq N \leq 10$ o pacote LMI mostrou-se mais rápido que o pacote CSDP, como por exemplo, novamente considerando a dimensão $N = 8$, o pacote LMI convergiu em $0,1s$, ao passo que o outro pacote necessitou de $3s$. Entretanto, a partir de $N \geq 20$, o comportamento do pacote CSDP tende também a tornar-se mais rápido que o pacote LMI. Por exemplo, para problemas com dimensão máxima, $N = 100$, o pacote CSDP operou em $1min11s$, enquanto o pacote LMI necessitou para o mesmo problema em $6h4min16s$. O comportamento do pacote CSDP, para à análise do gráfico de iteração média, também obteve resultados semelhantes ao gráfico de iteração mínima, ou seja, este pacote sempre obteve menos iterações em relação ao pacote LMI para todas as dimensões. Neste caso, não ultrapassou o número de 40 iterações, ao passo que o pacote LMI atingiu para a dimensão máxima 90 iterações.

Finalmente, na análise dos gráficos referente aos valores máximos de tempo, os pacotes LMI e CSDP tiveram o mesmo comportamento dos gráficos de valores mínimo e médio. Curiosamente, para problemas com dimensão $N = 8$, o pacote CSDP sempre teve o mesmo comportamento: atingiu o tempo máximo. Neste caso, convergiu aproximadamente em $6s$ ao passo que o pacote LMI necessitou de $0,1s$. Para dimensão $N \geq 20$, o pacote CSDP também tornou-se mais rápido do que o outro pacote e sempre com menos iterações. Observando o gráfico de iteração máxima, notamos que o pacote CSDP não ultrapassou o número de 50 iterações, enquanto o pacote LMI atingiu o número de 100 iterações. A partir da dimensão $N = 50$, este número tornou-se constante.

Resumindo, podemos observar que, para valores de $1 \leq N \leq 10$, o pacote LMI é mais rápido do que o pacote CSDP, ao passo que para $N \geq 20$, o segundo pacote torna-se mais rápido do que o primeiro, sempre com menos iterações (isto deve-se ao algoritmo do pacote CSDP basear-se na teoria dos pontos interiores). Além disso, o algoritmo foi escrito na linguagem em C (porém, pode ser usado pelo MatLab via o toolbox yalmip), tornando o pacote CSDP mais estável que o LMI, que mostrou problemas de estabilidade para dimensões $N \geq 20$.

CAPÍTULO 5

Conclusão

Ao longo deste trabalho, abordamos um problema fundamental na área de informação e computação quântica: a discriminação de estados quânticos não-ortogonais.

Em [17, 18], foi proposto um procedimento computacional, chamado Algoritmo Ótimo Discriminador (AOD), que implementa as medidas POVM para a estratégia da distinção de N estados puros não-ortogonais, usando a programação semidefinida e a minimização da norma. Baseado nas idéias do AOD, desenvolvemos um novo algoritmo, denominado Algoritmo Discriminador Quântico (ADQ).

A principal diferença entre o AOD e ADQ é que usamos apenas conceitos de álgebra linear, substituindo o cálculo de raízes de um polinômio de grau 8 (efetuado no AOD) e tornando o ADQ mais eficiente. Acreditamos também termos conseguido reduzir bastante os conceitos físicos envolvidos no AOD, deixando o ADQ mais compreensível para a comunidade de matemática e computação.

Os resultados computacionais foram apresentados com base em dois pacotes inseridos na implementação do ADQ: LMI e CSDP. Observamos que para problemas com pequenas dimensões, a implementação que utiliza o primeiro pacote foi mais eficiente. No entanto, para problemas com dimensões maiores, a implementação que utiliza o pacote CSDP foi mais eficaz. Não pudemos fazer comparações de tempo e número de iterações entre AOD e ADQ, pois não existem tais dados para o AOD.

Como prosseguimento natural deste trabalho, citamos as aplicações, por exemplo, em criptografia quântica.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] P. Benioff, *The computer as a physical system: a microscopic quantum hamiltonian model of computers as represented by Turing machines*, Journal of Statistical Physics, Vol. 22, p. 563-591, 1980.

[2] S. Boyd and L. Vandenberghe, *Semidefinite programming*, SIAM Review, Vol. 38, p. 49-95, 1996.

[3] J. I. Cirac and P. Zoller, *Quantum computations with cold trapped ions*, Physical Review Letters, Vol. 74, p. 4091-4094, 1995.

[4] T.M. Cover and J.A. Thomas, *Elements of information theory*, Wiley, 1991.

[5] Y.C. Eldar, *A semidefinite programming approach to optimal unambiguous discrimination of quantum sates*, IEEE Transactions on Information Theory, Vol. 49, p. 446-456, 2003.

[6] P. Gahinet, A. Nemirovski, A. J. Laub and M. Chilali, *LMI control toolbox for use with Matlab*, The MathWorks Inc., 1995.

[7] G. Golub and C. Van Loan, *Matrix Computations*, Johns Hopkins University, 1996.

[8] L. K. Grover, *Quantum mechanics help in searching for a needle in a haustack*, Physical Review Letters, Vol. 79, p. 325-328, 1997.

[9] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, *An interior point method for semidefinite programming*, SIAM Journal on Optimization, Vol. 6, p. 342-361, 1996.

[10] I.D. Ivanovic, *How to differentiate between non-orthogonal states*, Physics Letters A, Vol. 123, p. 257-259, 1987.

[11] G. Jones and M.J. Jones, *Information and Coding Theory*, Springer, 2000.

[12] U. Jönsson, C.-Y.Kao, A. Megretski and A.Rantzer, *A guide to IQCβ: software for robustness analysis*, 2004.

[13] Yu Nesterov and A. Nemirovski, *Interior point polynomial methods in convex programming: theory and applications*, SIAM Books, Philadelphia, 1994.

[14] M.A. Nielsen and I.L. Chuang, *Quantum computation and quantum information*, Cambridge University Press, 2000.

[15] J. Nocedal and S.J. Wright, *Numerical optimization*, Springer, 1999.

[16] D. Peaucelle, D. Henrion, Y.Labit and K. Taitz, *Users guide for SeDuMi interface 1.04*, 2002.

[17] W.R.M. Rabelo, *Algoritmos para a informação quântica: discriminação de estados quânticos e modelo híbrido*, Tese de Doutorado em Física, UFMG, 2006.

[18] W. R. M. Rabelo, A. G. Rodrigues and R. O. Vianna, *An algorithm to perform POVMs through Neumark theorem: applications to the discrimination of non-orthogonal pure quantum states*, International Journal of Modern Physics C, Vol. 17, p. 1-16, 2006.

[19] L. Roa, J. C. Retamal and C. Saavedra, *Quantum-state discrimination*, Physical Review A, Vol. 66, p. 12103-1 a 12103-4, 2002.

[20] P.W. Shor, *Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM Journal on Computing, Vol. 26, p. 1484-1509, 1997.

[21] H. Wolkowicz, R. Saigal and L. Vandenberghe, *Handbook of Semidefinite Programming: Theory, Algorithms and Applications*, Kluwer Academic Publishers, 2000.